Information Search Behavior Profiles: Analysis of Search Activities & Behavior Driven Ranking

DISSERTATION

zur Erlangung des akademischen Grades

Doktoringenieur (Dr.-Ing.)

angenommen durch die Fakultät für Informatik
der Otto-von-Guericke-Universität Magdeburg

von M.Sc., Johannes, Schwerdt

geb. am 09.09.1987 in Halle/Saale

Gutachterinnen/Gutachter

Prof. Dr. Andreas Nürnberger
Prof. Dr. Norbert Fuhr
Prof. Dr. Anke Huckauf

Magdeburg, den 13.10.2023

OTTO-VON-GUERICKE-UNIVERSITÄT MAGDEBURG

DOCTORAL THESIS

Information Search Behavior Profiles: Analysis of Search Activities & Behavior Driven Ranking

Author:
Johannes Schwerdt
(09.09.1987, Halle/Saale)

Supervisor:
Prof. Dr. Andreas Nürnberger

Reviewers:
Prof. Dr. Andreas Nürnberger
Prof. Dr. Norbert Fuhr
Prof. Dr. Anke Huckauf

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Engineering

in the

Data & Knowledge Engineering Group
Technical and Business Information Systems

13.10.2023

OTTO-VON-GUERICKE-UNIVERSITÄT MAGDEBURG

Abstract

Faculty of Computer Science
Technical and Business Information Systems

Doctor of Engineering

Information Search Behavior Profiles:
Analysis of Search Activities & Behavior Driven Ranking

by Johannes Schwerdt

Nowadays, it has become a crucial skill to gather information from the Internet to satisfy a current information need. Information Retrieval (IR) systems provide such frameworks to represent & organize information with the aim of easy access for users interested in it. Unfortunately, the characterization of the information need of users is not a simple task since it is pervaded with subjectivity, vagueness and uncertainty. With growing information content and demands of users, it has become evident that new concepts and techniques are needed to support users in their pursuit to gather information and extract knowledge from it. For that goal, systems have been envisioned that support users via personalization mechanisms. Such so-called User Models work on collected data about users to build specialized models capable to cater to different user characteristics. This thesis aims to add new mechanisms for that evolution and create learnable user-centered models able to adjust towards aspects of subjectivity, partiality and the user's individual concept of relevance. These proposed models aim to work on aspects derived from behavior information of users. Therefore, these models will be called (User) Behavior Models. To improve support for user search sessions when working with IR systems, (User) Behavior Models will be designed for specific user search activities. By merging the IR system with such models, a combined system should be able to adapt their results towards desired information suitable for the given search activity context. In the core of this thesis, it is assumed that only the user knows what is relevant and what is not, and that the user will behave accordingly. By analyzing the user search activity, it can be assumed that the underlying relevance concept can be measured indirectly by the analysis of the user behavior. This will result in the idea of a behavior driven relevance concept, and a formal model will be derived throughout this thesis.

OTTO-VON-GUERICKE-UNIVERSITÄT MAGDEBURG

Abstract

Faculty of Computer Science
Technical and Business Information Systems

Doctor of Engineering

Information Search Behavior Profiles:
Analysis of Search Activities & Behavior Driven Ranking

by Johannes Schwerdt

Heutzutage wird es zu einer immer wichtigeren Aufgabe, Informationen aus dem Internet zu akquirieren, um zugrundelegende Informationsanliegen zu erfüllen. Dafür stellen Information Retrieval (IR) Systeme Schnittstellen in Form von Informationsorganisation und -repräsentation, mit dem Ziel einen einfachen Zugang für Nutzende zu schaffen. Unglücklicherweise ist die exakte Charakterisierung des Informationsanliegens von Nutzenden keine einfache Aufgabe, da es Aspekte von Subjektivität, Unschärfe und Ungenauigkeit umspannt. Mit dem Anstieg an Wissen und Informationen, wird es immer offensichtlicher, dass es neuer Konzepte und Techniken bedarf, Nutzende bei der Informationssuche und Wissensakquirierung zu unterstützen. Für dieses Ziel wurden Systeme vorgeschlagen, welche sich durch Personalisierung an Nutzende anpassen. Solche sogenannten User Models arbeiten auf der Grundlage von gesammelten Nutzungs-Daten, um daraus spezialisierte Modelle abzuleiten, welche eine Anpassung ermöglichen. Das Ziel dieser Arbeit ist es, diese Entwicklung zu unterstützen und lernbare Modelle zu formulieren, welche sich an die Subjektivität von Nutzenden anpasst und auf zugrundeliegende Relevanzkonzepte zu schließen. Diese vorgeschlagenen Modelle arbeiten auf Aspekten, die abgeleitet werden von dem Verhalten (engl. Behavior) von Nutzenden (engl. User) während der Informationssuche. Aus diesem Grund werden diese Modelle als (User) Behavior Model bezeichnet. Um die Unterstützung von Nutzenden in ihren Recherchen durch IR Systemen auszubauen, werden diese (User) Behavior Models erstellt für spezifische Such-Aktivitäten. Durch das Verbinden dieser Systeme mit IR Systemen entsteht ein kombiniertes adaptives System, welches seine Resultate anpasst an den passenden Such-Aktivitätskontext. Im Kern dieser Arbeit steht die Annahme, dass nur die Instanziierung des Nutzenden selbst weiß, welche Information für die Suche relevant ist und welche nicht. Das Verhalten der Instanz wird sich während der Suche demnach ausrichten. Durch die Analyse der Such-Aktivität, sollte es möglich sein, auf das zugrundeliegende Relevanzkonzept von Nutzenden zu schließen in indirekter Form durch die Analyse des Verhaltens von Nutzenden. Diese Idee führt zu einem verhaltensbasierten Relevanzkonzept und das formale Modell wird innerhalb dieser Arbeit formuliert.

Contents

List of Figures

List of Tables

Part I

Introduction

Chapter 1

Introduction

1.1 Information Retrieval

Libraries were among the first institutions to adopt *Information Retrieval* (IR) systems for retrieving information [8]. With the increasing importance and ubiquity of the Internet as an information source, the growing interest in IR moved its habitat from libraries to the web [29]. In general, IR aims at modeling, designing and implementing systems able to provide fast and effective access to a large amount of information [29] and to represent & organize information for easy access of users interested in it [8]. Unfortunately, the characterization of the user *Information Need* is not a simple task [8] because it comprises aspects of subjectivity, vagueness and uncertainty [29]. In case of information items of textual form, the text is often translated into a set of keywords (or index terms) which serve as a summarized description of the item. Such keyword-based reductions remain quite popular in IR systems, because it allows for efficient indexing and processing [129]. Besides the efficiency of such a reduction approach, the effectiveness to satisfy the information need remains an open issue. IR systems must somehow 'interpret' the contents of the information item [8]. This not only includes the difficulty to extract information, but also to measure its relevance [8]. Thus, the notion of relevance remains at the center of IR [8] and consequently the analysis of the expressed information need of users [29]. For that, research in IR includes modeling, document classification and categorization, systems architecture, user interfaces, data visualization, filtering, languages, etc. [8]. Crestani et al. [29] argued that it is becoming increasingly evident that as knowledge and information grow in complexity and content in all dimension, new concepts and techniques are needed for dealing not just with hard knowledge and hard information, but also with soft knowledge and soft information. Therefore, adaptive systems are needed that are able to consider subjectivity, partiality and the individual concept of relevance for a particular user [29].

1.2 User Modeling

With the growth of the available information on the web, the diversity of its users and the complexity of web applications, researchers started to question a 'one-size-fits-all' approach [19]. To address these deficits, the development of adaptive systems started, that tailored their appearance and behavior to each individual user or user group [19]. These adaptive systems were specifically designed for usage in different contexts and in a personalized way. To support this personalization, these systems collected data about the users by implicitly observing their interaction and explicitly requesting direct input from them, and they build *User Models* (UM) aka *Profiles* that enabled them to cater to different user characteristics [19]. UMs are an explicit representation of properties of a particular user or user group, which allows the system to adapt its performance to individual needs [9][18]. The study of the field of UMs has resulted in significant amounts of theoretical work, as well as practical experience in developing UM based applications in a variety of fields, including Artificial Intelligence, Psychology, Linguistics, Human-Computer Interaction [9] as well as Information Retrieval, Machine Learning, Cognitive Science [19] and Information Filtering, Hypermedia Presentation, Tutoring Systems, E-Commerce and Medicine [18]. Fortunately, with the spread of user modeling in everyday applications, concerns about privacy are emerging [18].

1.3 Scope of the Thesis

The dedicated scope of this work is centered on the following three aspects:

- **User Modeling**: To improve support for the search experience of users when working with *Information Retrieval* (IR) systems, a *User Model* (UM) for specific search activities is aimed to be implemented. A suitable UM combined with an IR system should support the user with adequate search activity support mechanisms. By combining the IR system with the UM, such a combined system is able to adapt result lists towards the desired information items suitable for the given search activity context. The UM aimed to be implemented will work on search behavior *Profiles* that additionally gain insights about the search process itself. The proposed UM will be called the *Information Search Behavior Profile Model*.

- **Relevance**: To improve user support when working with IR systems, a suitable relevance concept is needed that can work with aspects of subjectivity, partiality, vagueness, uncertainty and more generally can adapt towards an individual user. In the core of this thesis, it is assumed that only the user knows what is relevant and what is not, and that the user will behave accordingly. By analyzing the user search behavior via a suitable UM, it can be assumed that the relevance concept can be indirectly be measured by the behavior of the user. Therefore, a behavior driven relevance concept is aimed to be characterized within this thesis.

- **Ranking**: To improve IR systems, the retrieval process (ranking) should consider an individual and user centered estimate of relevance. The described UM should abstractly correlate with a relevance concept suitable for the contextualized search activity of the user. Therefore, a behavior driven ranking concept is aimed to be design within this thesis. The proposed ranking approach will be called the *Information Search Behavior Profile Ranker*.

The thesis postulates some very ambitious goals of quite an interdisciplinary nature. While User Modeling is rather associated to the field of Psychology, Information Retrieval is rather founded in Information Science and the combination of both via computational models relies heavily on the field of Computer Science, see Fig. 1.1. It is clear that this thesis cannot solve the task in its entirety. Nonetheless, a careful restriction towards a focused sub-field yields the potential as a proof-of-concept for the ideas postulated in this thesis. First, the user search behavior comprises a magnitude of expressions which descriptions could already fill an entire thesis itself. The reduction of search behavior towards two prototypical extremes is a promising starting point for such a needed analysis. The concept of search activities provides such candidates in the form of Exploratory Searches and Fact-Finding Searches. Second, the user search behavior manifests in a plethora of actions that comprise a diverse set of modalities. Again, one thesis cannot exhaustively describe all possible actions and strategies a user can execute or implement during the online searches. Therefore, this thesis restricts itself to modalities derived from log-files and eye tracking data. Even these modalities comprise nearly an infinite amount of possible features that can be derived. Within this thesis, a meaningful subset will be described and motivated. Third, several data modeling techniques exist to draw conclusions about underlying processes. Unfortunately, even these objective techniques are currently rather motivated by certain beliefs in the concept of Artificial Intelligence and Machine Learning. Within the dedicated sections, the illusion of such an intelligence and its ability to learn in a romanticized interpretation will be completely invalidated by exhaustive definitions, formulas and explicit modeling approaches. The proposed methodology will be centered on Bayesian Networks and statistical modeling. This will be justified by the lack of black-box behavior of these models, by their increased interpretability in comparison to other modeling paradigms and their capabilities to work on data sets of even limited sizes. To achieve the claimed goals, the thesis needs to dive deep into the respected domains, bridge the gap between the individual terminologies and eventually prove its worth with its findings.

This is the point where the reader might expect a statement about the research questions. I firmly believe that not even the best possible introduction is able to provide all the necessary details to presented well-founded research questions. I apologize to the curious reader who expects these so early on. The fundamentals are needed first. Skip ahead to Sec. 5 *Research Questions* for scanning through. Without the necessary fundamentals, research questions are merely more than structured sequences of characters created by an ego or aimed to satisfy another one.

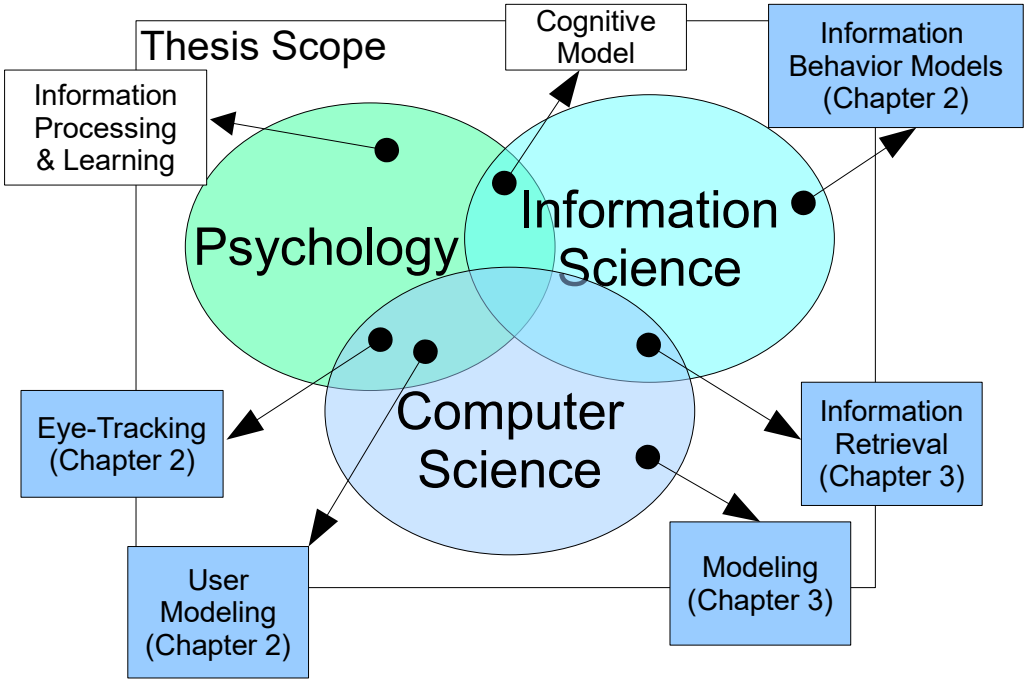


FIGURE 1.1: The scope of the thesis comprises interdisciplinary aspects ranging from: User Modeling & Psychology, Information Retrieval & Information Science and Computer Science & (Data) Modeling. Introducing chapters for the respected fields are displayed in brackets.

1.4 Structure of the Thesis

The thesis is structured in three main parts: Part I *Introduction*, Part II *Fundamentals*, Part III *Information Search Behavior Profiles* and an additional *Appendix* in Part IV for all necessary mathematical fundamentals. Each part is structured in chapters and sections, which will be described in the following.

Part II *Fundamentals* will introduce the foundation of this thesis. It is structured in three chapters: Chap. 2 *Information Behavior Models*, Chap. 3 *Modeling* and Chap. 4 *Related Work*. Chap. 2 will introduce aspects of the fields of Information Science and Psychology. It will introduce necessary concepts in respect to the user's *cognitive model* and aspects of *behavior* that can be seen as the expression of it. First, the fundamentals of *Information-Seeking Behavior* & *Information Search Behavior* will be introduced to lay the foundation of the *User Model* aimed to be implemented. Second, the fundamentals of *Eye-Tracking* will be introduced with a particular focus on *Reading and Information Processing*. Chap. 3 will introduce modeling aspects associated to the fields of Computer Science and Information Retrieval. These model will serve as the foundation to draw conclusions from the behavior level towards the cognitive model. First, the fundamentals of modeling data will be described, which will realize the mathematical implementation of such a proposed User Model. A particular focus will be centered on *Graphical Models*, especially *Bayesian Networks*, because of their flexible nature combined with their interpretability. Second, the fundamentals of *Ranking* in Information Retrieval will be described and put into perspective with Bayesian Networks. The combined approach of a (User) Behavior Model and a ranking model via one global network will produce a user-centered and behavior-driven ranking paradigm. Chap. 4 will describe the *Related Work* of this thesis, especially in respect to Information Retrieval, User Models and Information Search. While Chap. 2 & 3 already provided an overview of related work in their individual descriptions, Chap. 4 will focus on technical realizations instead of the conceptual research previously mentioned.

Part III *Information Search Behavior Profiles* will introduce the central core of the thesis itself. Chap. 5 will start with a detailed description and motivation of the *Research Questions* (RQ) of this thesis and put them in perspective with the *Fundamentals*. The following Chap. 6 *User Study - Design* will present the design of the user study that will serve as the foundation of the following analysis. This comprises a detailed description of the experimental design, the characterization of the participants of that experiment and the technical implementation of it. The analysis itself will be described during Chap. 7 *User Study - Analysis*. The first three sections will mainly focus on aspects for modeling the proposed (User) Behavior Model in respect to search activity recognition. These sections will primarily center on the navigational strategies of users during their online search and the analysis of basic actions a user executes during it. The subsequent two sections will increase the analysis via higher-level eye movement motifs with a specific focus on reading and information processing. The last three sections describe the combination approach of the (User) Behavior Model with an Information Retrieval system, which will lead to a new ranking approach. The final part in Chap. 8 will provide an overall conclusion, summary & perspectives to the scope of the thesis. All (sub)conclusions of the previous sections will be put into perspective and a global summary will be presented. Further on, potential for future development will be described and aspects of future work will be stated.

Part IV *Appendix* is a collection of useful material that describes mathematical fundamentals for *Probability Theory*, *Information Theory*, *Vector Space & Matrix Algebra*, *Probability Distributions* and *Machine Learning*.

The thesis comprises a recognizable length. I am aware that time is a precious resource and the natural question arise if the thesis is worth reading in its entirety. Because of conservative time management, I propose two alternative & distilled reading paths, e.g. for the **modeling purist** and for the **application purist**:

- **Modeling Purist**:
  Before anything of worth can be derived from data, a clear modeling methodology needs to be stated. Without, no valid conclusion can be derived. There is no need to read the details if such a methodology is missing or misleading. Therefore, the reading path 'blue' in Fig. 1.2 is the most efficient way to read the thesis under this perspective. As a modeling purist, you are aware of the *No Free Lunch Theorem*. Generally, there is no best model but given the particular constraints of a specific domain, there is the possibility of models more capable to uncover insights than others. As a modeling purist, you can accept this fact and dive into the details of the domain after initial doubts of the modeling methodology have been cleared. In case that my work does not suffice the expected standards, at least some sections could be skipped.

- **Application Purist**:
  Any model is just a reduction of a real-life phenomenon. There is no model that can capture a complex reality and there is no need to read all the details about data modeling. The application is part of our shared reality, abstract modeling is more of a philosophy. Therefore, the reading path 'green' in Fig. 1.2 is the most efficient way to read the thesis under this perspective. As an application purist, you are aware of the famous statistician George E. P. Box who stated: *all models are wrong, but some are useful*. Therefore, as an application purist, you can accept the fact that models have the potential to capture a glimpse of truth. After initial doubts about the description of the domain have been cleared, the details about modeling can be read at a later stage. In case that my work does not suffices the expected standards, at least some sections could be skipped.

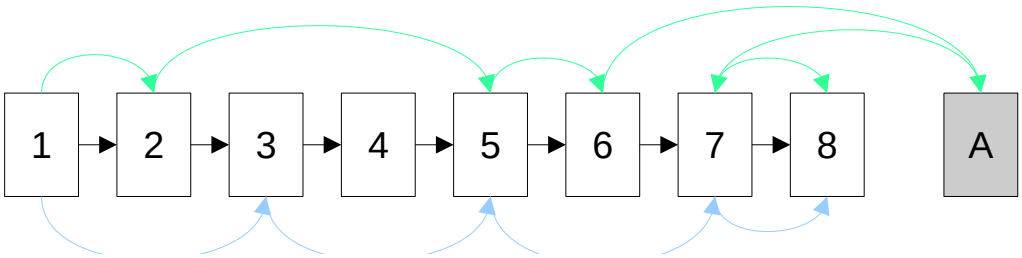


FIGURE 1.2: Three reading paths of the thesis. Black: thorough reading of the entire thesis. Distilled reading paths: modeling purists (blue) & application purists (green).

Part II

Fundamentals

Chapter 2

Information Behavior Models

A variety of models exist that address different aspects or levels of information behavior. Within the context of the following sections, the term 'model' refers to concepts of high abstraction levels. Therefore, clear 'definitions' cannot be stated but 'models' can be motivated or described 'rhetorically'. Kotzyba et al. [69] presented an overview of different models for *Information Behavior* and how to put them into perspective. In the following, the categorization approach of Kotzyba et al. [69] is strictly applied. Information behavior models are a general approach to describe users during information acquisition/exploration. It categorizes the user's attempt to satisfy an *Information Need* and includes context information about the user, possible dialog partners and/or information systems. All methods, strategies and tactics that a user implements during such information search are covered by *Information-Seeking Behavior* models. A particular instance or a class of information-seeking behavior, e.g. an exploratory search, is often categorized as *Information Activity*. The most concrete level of such models is described by the *Information Search Behavior*, which addresses interaction with an information system via mouse, keyboard, eye movement, query input and more. Based on this description, one might recognize a nesting or hierarchy of these models that is visualized in Fig. 2.1.

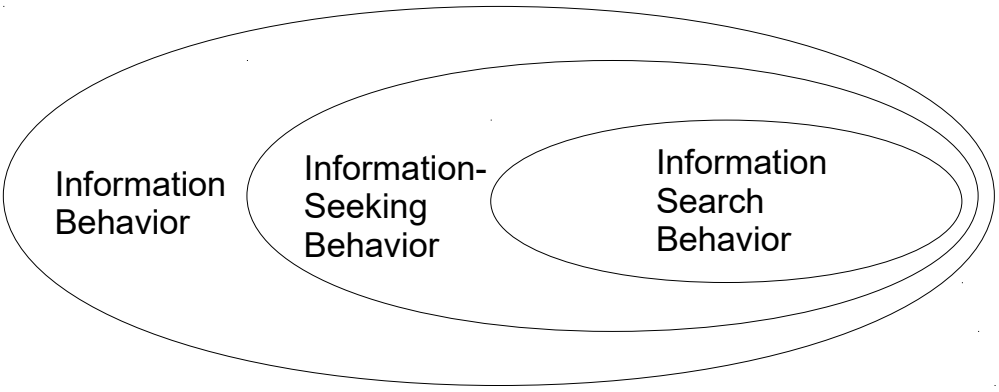


FIGURE 2.1: Adaption of [69]: Wilson's nested model of Information Behavior illustrating the hierarchy of Information Behavior, Information-Seeking Behavior and Information Search Behavior models.

The thesis aims to identify & characterize the user's information-seeking behavior during online search sessions and to exploit these findings in the Information Retrieval (IR) framework. Therefore, mathematical & computational models need to be designed which are able to recognize such behavior and to draw conclusions about its characteristics, e.g. in form of executed actions and strategies during the search. With such models given, user adaptive & context aware IR systems could provide more sophisticated and more adequate user support during search sessions. A technical system is only able to recognize measurable expressions on the level of the information search behavior. The actual user intent lies on the level of the information-seeking behavior. Unfortunately, this is not directly measurable. Nonetheless, there is strong evidence for the assumption that both levels are heavily associated with each other. To advance towards more adaptive IR systems, which recognize desired interests & needs or even anticipate them, a clear outline of information-seeking behavior & information search behavior is needed.

2.1 Information-Seeking Behavior

2.1.1 Kuhlthau's Model

Kuhlthau [72][73] proposed an information-seeking model with six *stages* and corresponding activities. A compact description of the six stages is given by Kotzyba et al. [69] and presented here with minor adaptions:

- **Initiation**:
  First awareness & recognition for a lack of knowledge creating the *Information Need*. Users might have a feeling of uncertainty and apprehensions.
- **Selection**:
  Identification of a relevant search domain that apparently leads to success. Users might have an optimistic view in case of quick and positive results, or anxiety in case of delay of any kind.
- **Exploration**:
  Investigation of general topics to specify the Information Need. Users might feel confused and uncertain.
- **Formulation**:
  Turning point which focuses the search. Users might feel confident and focused.
- **Collection**:
  Most effective and efficient stage in respect to accumulation of relevant information. Users might perceive continuing confidence.
- **Presentation**:
  Retrospective evaluation of the search process satisfaction. User might have the feeling of relief.

The stages suggest an internal ordering, but they can be traversed in a non-binding way, at least in parts. The model describes the information acquisition in respect to users feelings, thoughts and actions and hence has a phenomenological perspective [69].

2.1.2 Ellis' Model & Wilson's Aggregation

Ellis et al. [37][39][38] discussed an alternative model for information-seeking and empirically support it by studies on scientists, e.g. physicists, chemists and social scientists. The model comprises eight *features* and a compact description is given by Kotzyba et al. [69] and presented here with minor adaptions:

- **Starting**:
  Activities to initiate an information acquisition, e.g. asking others, formulating a query, identify a first document etc.
- **Chaining**:
  Building chains of relevant documents by following references, hyperlinks, citations, footnotes, etc.
- **Browsing**:
  Performing a semi-directed search or exploration of a promising domain.
- **Differentiating**:
  Differentiation between several information sources by exploiting particular characteristics as filters.
- **Monitoring**:
  Keeping awareness of development in a (search) domain.
- **Extracting**:
  Identify specific information which lead to success.
- **Verifying**:
  Checking the retrieved information.
- **Ending**:
  Activities to complete the information acquisition.

Even though, these aspects are different to *Kuhlthau's Model* [72][73], Wilson [136] suggest a relation between them and enriched the model with additional *activities*. The aggregated model of the three perspectives (*stages*, features, activities) can be seen in Fig. 2.2.

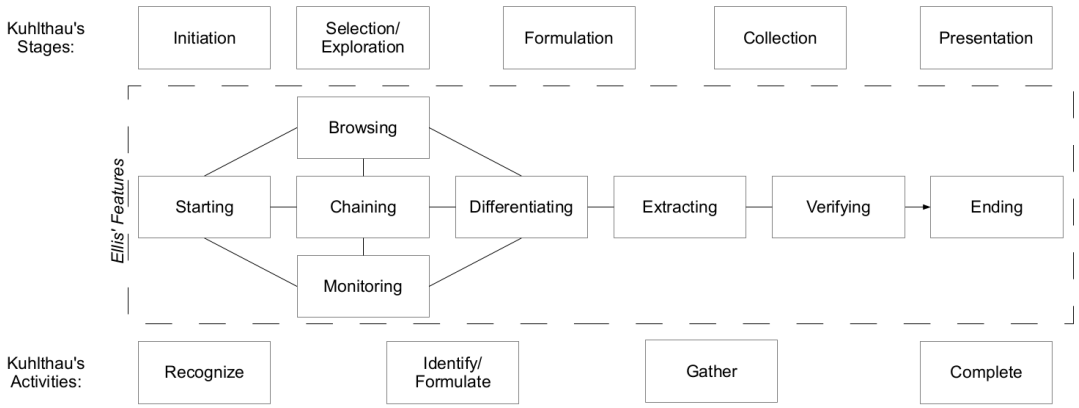


FIGURE 2.2: From [69]: Wilson's aggregation [136] of *Kuhlthau's Model* [72][73] and Ellis' models [37] [39][38].

2.1.3 Exploratory Search & Search Activities

In Information Retrieval, *exploration* or *exploratory search* are embedded in the context of a search process with vague *Information Need*. It is connected with information-seeking [69] and incorporated in *Kuhlthau's Model* [72][73] via the stages *Selection* & *Exploration*. Marchionini [79] provides a framework of different *search activities* which integrates exploratory searches. According to this framework, activities within a search decompose into *lookup*, *learn* & *investigate*. Lookup describes a standard *Fact-Finding* search with a specified query request, which is associated to Kuhlthau's Model via the *Selection* stage [69]. This can be considered as an elementary, conscious and purposeful action to satisfy the precise Information Need. Exploratory searches extend this lookup with the activities of learning & investigation. Both activities are considered to be iterative processes that involve different search strategies [69], which falls in-line with the concept of *Information-Seeking Behavior*. In conclusion, exploratory searches are often characterized as open-ended, multifaceted searches with unclear goals [134][135]. In exploratory searches, the acts of searching, browsing and navigation are often more important than the actual find and success in exploratory searches does not necessarily mean to find a certain piece of information, but to learn, investigate and to conceptualize about an initial Information Need and to build up on it.

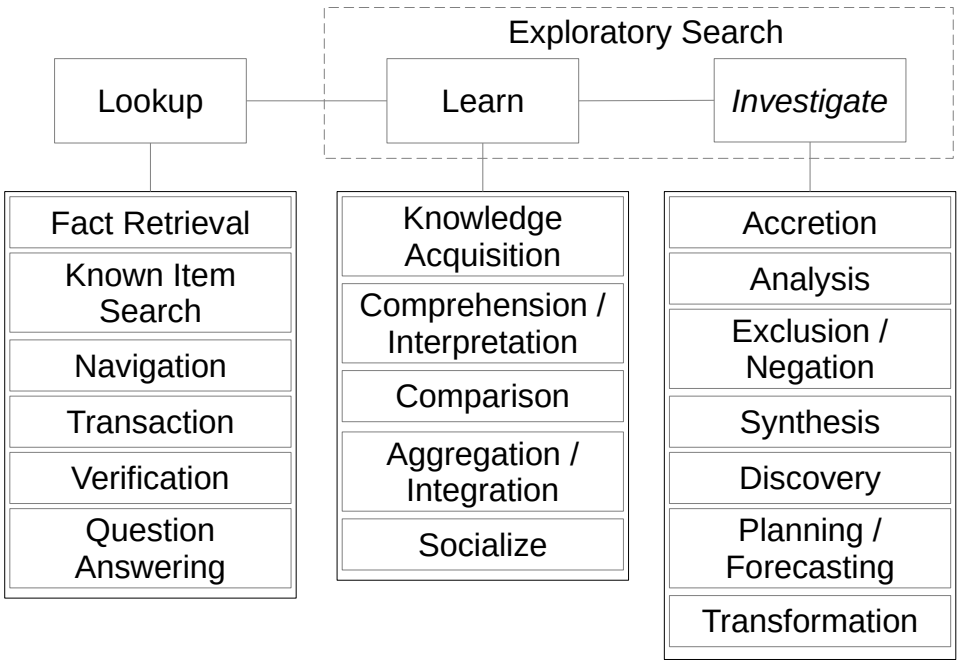


FIGURE 2.3: Adaption of [69]: Marchinonini's exploratory search embedded in search activities.

2.2 Information Search Behavior

2.2.1 Navigation & Probabilistic Regular Grammars

The analysis of navigational patterns in Internet searches have been extensively researched to extract important navigational rules of users. A quite general framework has been proposed by Borges et al. [16] which is based on *Hypertext Probabilistic Grammars* (HPG) to model user behavior in respect to navigational trails. HPG are a subclass of *Probabilistic Regular Grammars* which are grounded on a sound theoretical foundation. In essence, each navigational pattern can be seen as a (sub) sequence, and the HPG approach assigns higher probability to (sub)sequences that correspond to the preferred trails of users. Fig. 2.4 schematically illustrates such an approach. Any navigational pattern consists of an abstract start (S) and an abstract finalization (F). Within these boundaries, any abstract web page ($A.$) can be visited by multiple navigational pattern. By measuring such navigational patterns via log-files, preferred trails arise with higher probabilities, indicating more promising navigational rules of users. The authors provide an algorithm to incrementally build the HPG using log-file data without the need of rebuilding it from scratch on [16]. To mine for important navigational rules, the model uses various configurations by applying certain ad-hoc parameters.

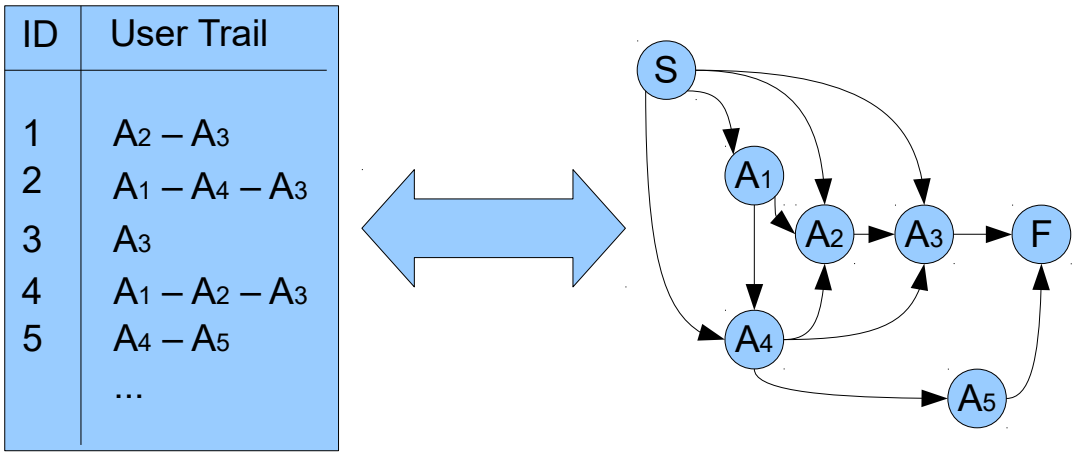


FIGURE 2.4: Adaption of [16]: A sample Hypertext Probabilistic Grammar (right) extracted from logged user trails (left) for user navigation behavior through web pages $\{A_1, ...A_5\}$ with an explicit start S and an explicit finalization F.

2.2.2 Combining Interaction & Navigation

The analysis of a combined approach of navigational pattern and the interaction with individual web pages is a promising approach to capture aspects of the *Information-Seeking Behavior*. The work of Chan [24] provides an interesting approach that explicitly models interaction with web pages via the *Page Interest Estimator* (PIE) and navigational patterns via the *Web Access Graph* (WAG). The WAG is a weighted directed graph of users web page access behavior, see Fig. 2.5. Each vertex represents a web page and stores information about it, while edges represent the traversal between pages. Therefore, the WAG represents patterns in a user web search and directly corresponds to *Navigation & Probabilistic Regular Grammars*. In addition, the PIE represents an abstract function for the interest of a user for a particular web page. Chan [24] described several functions that can approximate this interest, e.g. the frequency, duration and recency of particular page visits. The combination of the WAG and the PIE results in an abstract association between two web pages that comprises its connection via web page traversals weighted by the particular interaction scheme of the 'starting' page. This association is formulated as follows:

$$Association(Page_A \rightarrow Page_B) = P(Page_A|Page_B) \cdot Interest(Page_B)$$

It can easily be recognized that the WAG model is a special instance of the *Hypertext Probabilistic Grammar* (HPG) of Borges et al. [16] in Sec. 2.2.1 for a limited traversal context, e.g. the predecessor. In general, the WAG can be extended to a general HPG. The extension with the PIE as a weighting scheme is a promising approach to incorporate behavior aspects for individual web pages. Unfortunately, the construction of ad-hoc functions for the PIE is a laborious task and potentially prone to errors. Therefore, Chan [24] argues about the usage of Machine Learning models that intrinsically learn these functions based on measured observation in form of log-files. The combination of PIE and WAG can be used to model the web search behavior of users. It explicitly models aspects of navigation and interaction, which is well suited to combine aspects of the Information-Seeking Behavior with the *Information Search Behavior*.

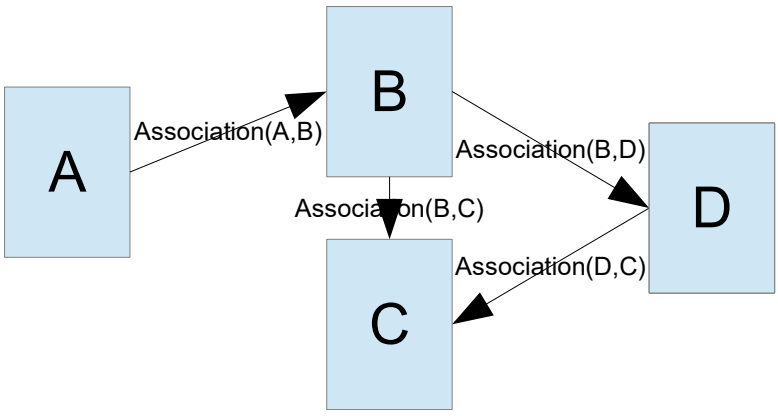


FIGURE 2.5: Adaption of [24]: A sample Web Access Graph for web pages $\{A, B, C, D\}$. Associations between pages can be modeled by a traversal function or by an additional weighting with the Page Interest Estimator.

2.2.3 Eye-Tracking

Eye-Tracking is the measurement of the eye gaze on a computer screen. The movement of the eyes is a sequence that mainly decompose into two measurable, rather atomic states, called *fixation* and *saccade*. While a fixation is a temporal steady state of the eye to focus on a stimulus, a saccade moves the visual field towards another one. Specific sequences of these states can form highly complex patterns for visual search, reading text or scanning web-pages for desired information. Fig. 2.6 illustrates such a pattern that processes a search engine result page (SERP). The illustrated pattern emerges via a heat-map of aggregated fixations on the SERP and forms an *F-shape*. A shape that can be observed quite often on SERPs. Such a pattern comprises several sub-patterns of orientational gaze over the SERP, scanning through the text and inspecting presented media such as images. An in-depth introduction to eye movements will be provided during the following sections to lay the foundation for the research work of this thesis.

FIGURE 2.6: Taken from [48]: Heat-Map of aggregated fixations to high-light the eye movement on the search engine result page of Google.

2.2.3.1 Fixations & Saccades

The previous introduction stated an oversimplification in respect to *fixations* & *saccades*. Besides saccades, three other types of eye movements can be distinguished: *pursuit*, *vergence* and *vestibular eye movements*. Pursuit eye movements are executed while following a moving target. Its velocity is slower than saccades, but saccades can be executed to catch up with the target [133]. Vergence eye movements describe an inward rotation of the eyes (toward each other) to fixate nearby objects. Vestibular eye movements compensate for head and body movements by rotation of the eyes. According to Rayner [101] saccadic eye movements are more relevant in typical information processing tasks in comparison to the previous mentioned ones. Also, the concept of a fixation as a steady state of the eyes is an oversimplification. A constant tremor of the eyes, the so-called *nystagmus*, results in a quite small movement, deeming fixations a misnomer [101]. Even though, the exact nature of that remains unclear, it is argued that this movement is related to perceptual activity and helps the nerve cells in the retina to keep firing [101]. Further on, during fixations the eyes occasionally drift in small and slow movements because of less-than-perfect control of the oculomotor system by the nervous system. These small eye movements are therefore called *drifts*. *Microsaccades* bring the eyes back to where they were by a more rapid movement. According to Rayner [101] these 3 types of small movements of the eye are assumed to be 'noise' by researchers focusing on the analysis of reading. Specific scoring procedures were design to adapt for the described effects. These smaller eye movements can therefore be unified towards one abstract concept, e.g. the fixation. Further on, an in-depth description is needed for what can be perceived within such a fixation. According to Sanders [111], the visual field can be divided into 3 regions: a stimulus can be identified without eye movements, by necessary eye movements and by necessary head movement. While looking straight ahead, the *visual field* of the eye can be divided into 3 regions according to Rayner [101]. The central 2° of vision is called the *foveal* and has very good acuity. The *parafoveal* extends up to 5° and the *peripheral* extends even further. They are characterized by decreasing acuity. A saccade is used to place the foveal on a stimulus. In addition, developmental changes of the eye movement have to be stated because eye movement in children differ in comparison to those of adults. Kowler & Martins [71] report for pre-school children more small saccades and drifts during maintained fixations, longer saccadic latency and less precise saccade accuracy when scanning a scene. In contrast, the shape of the frequency distribution of fixation durations for children, adults and infants have been reported to be similar [53]. In respect of elderly and younger adults, the fixation duration distribution show similarities, but saccade latency increases with age [2] & [93]. A brief summary of the introduced concepts can be found in Tab. 2.1.

Elements of Eye Movement	
Eye Focus:	
fixation	focused gaze & cognitive processing
nystagmus	constant tremor of the eyes
drift	small & slow drifts during fixations
micro-saccades	compensate for drifts
Eye Movements:	
pursuit	following moving targets
vergence	inward movement to fixate nearby objects
vestibular	compensate head and body movement
saccades	fast relocation of the visual field
Visual Field:	
foveal	central 2° of vision & very good acuity
parafoveal	up to central 5° of vision & decreasing acuity
peripheral	more than 5° of vision & decreasing acuity

TABLE 2.1: Brief summary for elements of eye movement

2.2.3.2 Eye Movement in Reading

In the context of reading English texts, *fixations* are reported to vary in duration in about 200-250 ms. Many words are skipped during reading. Studies of Carpenter & Just [23] and Rayner & Duffy [104] report that content words are fixated 85% of the time, while only 35% of function words are fixated. Multiple factors influence this skipping behavior, e.g. word length and sentential constraints. In respect to word length, there is a clear connection of fixation frequency and increasing word length. Words of 2-3 length are only fixated in around 25% of the time, while words of more than 7 letters are almost always fixated once and often more than once [105]. Studies on the effect of sentential constraints indicate that predictable words are more likely to be skipped than unpredictable ones because these constraints can be used in connection with the *parafoveal* information [36]. The perceptual span in reading texts of English language is rather asymmetric. It extends no more than 3-4 letters to the left and about 14-15 letter spaces to the right around the currently fixated word [82][21]. The perceptual span comprises variation in respect to other written languages, such as Hebrew [94] and Japanese [61], and depends on the level of practice [100]. *Saccades* are very fast, typically taking 30–80 ms to complete and velocities as high as 500° per second [26]. During reading, mean saccade sizes range from 7-9 letter spaces [101]. Letter spaces are reported to be appropriate measurements because of the relative invariance of traversed letters at different distances [86][87]. Most saccades follow the left-to-right direction as the text being read. Nonetheless, 10-15% of the saccades are *regressions*, a right-to-left movement along the line to previously read text units [101]. Rayner argues, that short regressions of few letters are corrections of too long saccades while short within-word regression are applied during processing problems of the currently fixated word. Regressions of more than 10 letters seem to indicate that the reader did not understand the text. In contrast to fixations, it is reported that during saccades new information can not be obtained because the eyes are moving so quickly across the stable visual stimulus that only a blur would be perceived [130]. All values reported in this section can only be seen as references because several parameters, such as age, education and

familiarity of the text, influence these metrics. All in all, it can be stated that there is a considerable variability between readers. Even differences during silent reading and oral reading have been reported with mean fixation durations of 225 ms and 275 ms, respectively [102]. Other factors, such as quality of presentation, line length and letter space, influence the eye movement. Models of eye movement control during reading can generally be classified into two categories: *processing models* and *oculomotor models*. The first one was proposed by Morrison [85] and states that the eye movement during reading is heavily influenced by lexical processing and ongoing comprehension processes. The second one was proposed by O'Regan [91] and states that movement is mainly controlled by oculomotor factors and ongoing language processes only indirectly influences the reading. O'Regan proposed that readers adopt 'strategies', such as careful or risky reading, which influences fixations and saccades. Even though, both models are supported by empirical evidence, it can be agreed that more mathematical and computational models are needed for sufficiently precise predictions and testing. A brief summary of the most important concepts and their characteristic reference values can be found in Tab. 2.2.

Eye Movement in Reading (English)	
Fixations:	
duration	200-250 ms
perceptual span	3-4 letters to the left 14-15 letters to the right
word length	fixated 25% of 2-3 length words almost always fixated more than 7 letter words
content words	fixated 85% of the time
function words	fixated 35% of the time
Saccades:	
duration	30-80 ms
mean size	7-9 letter spaces
direction	10-15% right-to-left (regression) most follow the left-to-right direction
text information	can not be obtained

TABLE 2.2: Brief summary for elements of eye movement with reference values in reading.

2.2.3.3 Reading and Information Processing

Reading plays an essential role during the information search process in online searches. Previous studies found evidence for subgroups of reading types with different aims, characteristics and underlying cognitive processes. Three major groups have been reported, namely *Scanning*, *Skimming* and thorough *Reading*. Scanning is a form of rapidly reading a text. According to Rodeghero & McMillan [109], the focus of Scanning is centered on gaining a particular piece of information of a text, rather than the general understanding of the text in its whole. The eye movement is considered to be a sweeping over the text with the aim to identify specific pieces of information such as keywords and phrases, e.g. definitions, phone numbers etc. According to White et al., the eye movement during Scanning is characterized by shorter overall reading time, fewer *fixations*, shorter first pass fixations, longer progressive *saccades* and higher skipping rates of words. Clark et al. [26] state that longer *scan-paths* can be observed in respect to duration and length. Eye movement over text considered as being irrelevant shows little indication for higher level processing in form of *regressions* for re-reading. According to Clark et al. [25] Scanning might be influenced by formatting styles, e.g. *italic* & **bold**. In general, Scanning text requires full attention and can be compared to a "mental spotlight". Skimming is also a form of rapidly reading a text. According to Rodeghero & McMillan [109] the focus of Skimming centers on understanding the general meaning and obtain a brief summary of the content for a given text. According to Clark et al. [26] it is characterized by less and shorter fixations, less saccadic regressions and long progressive saccades. Further on, it was mentioned that participants during Skimming capture the main content of the text but lack knowledge about precise details in it. Duggan & Payne [34] describe Skimming as a consequence of the reading experience after a threshold of information content is reached. After reading initial parts of a text in detail, readers switch to Skimming if the information content is low according to their intrinsic satisficing model. Reading is a clearly defined task with organized eye movements, according to Rayner & Castelhano [103]. It is characterized by moving from line to line and fixating almost every word in each line to ensure complete understanding of the text. A brief summary of the most important concepts and their aims & goals can be found in Tab. 2.3.

Reading Strategies
Scanning:
Focuses on gaining a particular piece of information (keywords and phrases) Sweeping over the text for specific snippets (definitions, phone numbers, etc.)
Skimming:
Focuses on understanding the general meaning (obtain a brief summary) Captures main content but lacks knowledge about precise details
Reading:
Focuses on each line and fixates (almost) every word Ensures complete understanding of the text

TABLE 2.3: Brief summary for elements of reading and their interpretation.

2.3 Summary

Information Search Behavior can be analyzed on multiple levels and by multiple perspectives. Nonetheless, the proposed concepts should rather be seen as complementary than contradictory. The broadest level of information search can be described by Information Behavior Models as a general approach to describe users in an attempt to satisfy an *Information Need*. The more specific level of *Information-Seeking Behavior* describes methods, strategies and tactics that a user implements during the information search. Promising models can be found in *Kuhlthau's Model* [72][73], *Ellis' Model & Wilson's Aggregation* [37][39][38][136]. These approaches can be described as rather conceptual and focused on the description of the cognitive model of the user during the search. Unfortunately, this is too abstract to be directly used in a Computer Science application. The more specific layer of *Information Search Behavior* realizes concrete interaction with an information system and is well suited to be analyzed by a Computer Science application. Throughout the previous sections, it could be observed that there is an interconnection between the conceptual strategies and the concrete interactions. Specifically designed *(User) Behavior Models* might be able to draw a conclusion based on observable interaction with an information system towards the cognitive model of the user during the search. Therefore, a clear outline between these levels need to be stated in the following.

The concept of *Exploratory Search & Search Activities*, especially for *Exploratory* and *Fact-Finding* search activities, are of specific focus during this thesis. Both search activities comprise different characteristics and properties in respect to the underlying Information Need. While the first comprises a rather *open* nature where an Information Need cannot be specified precisely, the second comprises a rather *closed* nature, that can be clearly defined. Based on such characteristics, it seems promising to analyze changes of interactions with the information system to evaluate if such interactions are indicative for different search activities. By using the binary separation in Exploratory and Fact-Finding search activities, it is straight-forward to draw conclusion via pairwise comparisons or by ratios. Navigational trails of users within online searches have extensively been researched, and a clear connection between *Navigation & Probabilistic Regular Grammars* could be stated. Further, approaches for *Combining Interaction & Navigation* are promising direction to analyze the complex interconnection between the levels of Information-Seeking and Information Search Behavior. A specific focus during this thesis is centered on the analysis of the eyes gaze because it can be assumed to be the main sensor unit of a user to extract knowledge during an online search. Such an analysis should be broader than the analysis of plain *Fixations & Saccades* to be of any value. Higher level analysis of patterns, such as *Eye Movement in Reading*, are of importance to gain understanding of the user search behavior. Especially, the analysis of *Reading and Information Processing* provide a fruitful perspective into the search behavior because of the close connection of reading and the cognitive model of users. The description of reading and variants of it, e.g. *Scanning*, *Skimming* and thorough *Reading*, provides a connection between text processing and information search.

Chapter 3

Modeling

Machine Learning is a data analytical approach that gained high interest during the last decades. Models were applied in a variety of domains such as speech recognition systems [78], computational molecular biology [49][35], data compression and areas simply called artificial intelligence [46]. From an application perspective, most common approaches in Machine Learning are either *classification* or *clustering* scenarios. In case of classification, data is assumed to be grouped into predefined classes. A classification model is trained on known data examples to estimate separating boundaries between these classes. For new data with unknown group assignments, a classification model aims to predict the unknown grouping. Fig. 3.1 illustrates a typical example of a classification approach. Data points are assigned to classes (color encoded). Both classes are densely populated but do not overlap. A classification model estimates the separating boundary between classes and assigns new data into a class by using the estimated boundary. A clustering model has no explicit knowledge about data grouping. Based on data similarity alone, a clustering model aims to predict previously unknown groupings within the data. Fig. 3.2 illustrates an example of such a clustering approach. Data point similarities are represented within a graph structure. The closer data points are connected within the graph, the more similar they are. Compact subgraph structures can be seen as data groupings (color encoded). Clustering models do not necessary estimate graph-like similarities. They can also group data into a 'flat' grouping, often simply called cluster. Such a 'flat' clustering is closely related to the described classification approach, but with a lack of knowledge about the group assignment of the data.

Machine Learning is heavily founded in Statistics, and the following section will formally introduce Machine Learning. Therefore, a unified formulation of variables will be stated and used throughout the following sections. Variables in X form represent aspects of data with: $\mathcal{X}$ being the data space, $\underline{\boldsymbol{x}} = (\boldsymbol{x}_1, ...\boldsymbol{x}_N)$ being a set of N data points within $\mathcal{X}$, all data points $\boldsymbol{x}_n = (x_{n1}, ...x_{nA})$ comprise features x_{na}. Features of index a are structurally the same. Nonetheless, data points can comprise discrete and continuous features. For simplicity, $\boldsymbol{x}$ is written for an arbitrary point $\boldsymbol{x}_n$. Consequently, x_a refers to the feature a in $\boldsymbol{x}$. Variables in Y form represent aspects of assignments for data with: $\mathcal{Y}$ being the space of assignments, $\boldsymbol{y} = (y_1, ...y_N)$ being N particular assignments for the data points in $\underline{\boldsymbol{x}}$. Assignments within the scope of this work are discrete values representing an interpretation, such as a class. For simplicity, y is written as the assignment of $\boldsymbol{x}$. Variables in C form represent aspects of predictions with: $\mathcal{C}$ being the space of predictions, $\boldsymbol{c} = (c_1, ...c_N)$ being N particular predictions for $\boldsymbol{y}$. Therefore, Y and C are closely related but not the same. For simplicity again, c is written for the prediction of the assignment y of data point $\boldsymbol{x}$.

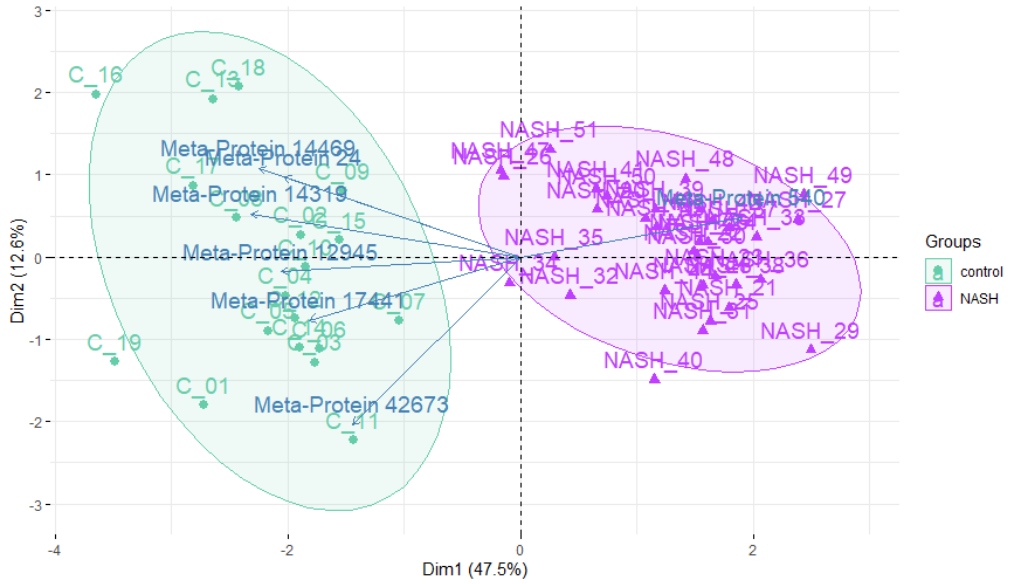


FIGURE 3.1: Supplementary Material of [127]: Visual representation of a classification scenario. Data points are assigned into groups/classes (color encoding). Classification models estimate separating boundaries between classes and use them for the prediction of new data with unknown group assignments.

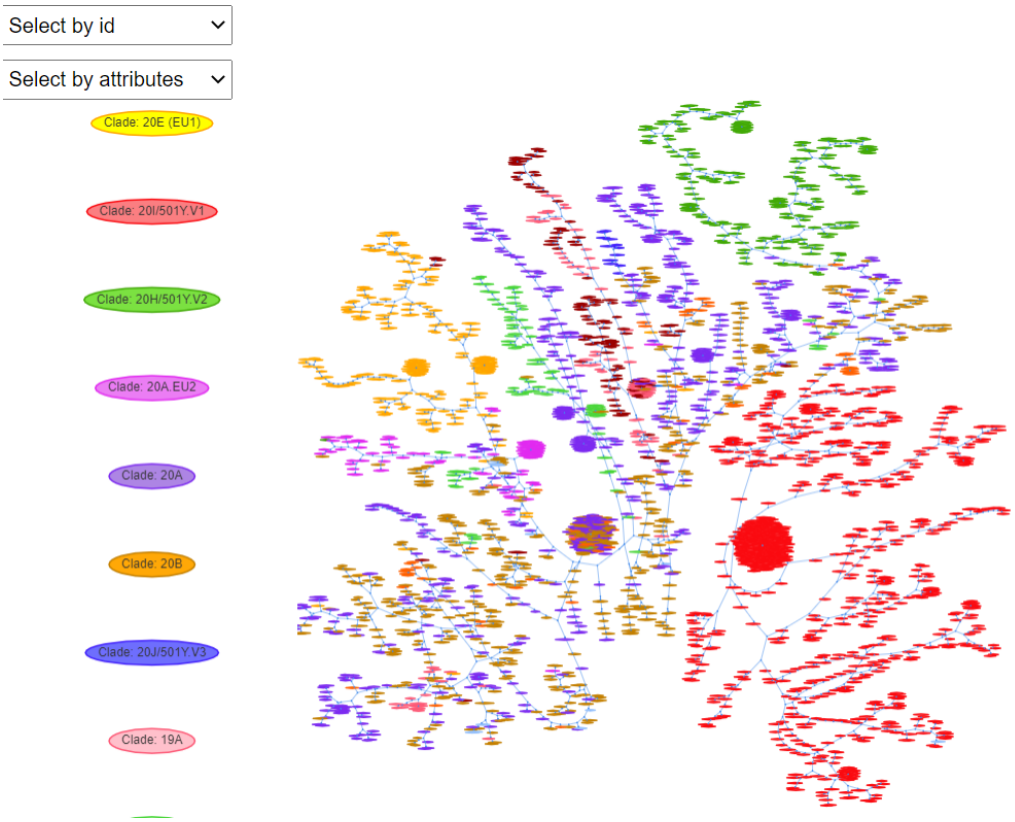


FIGURE 3.2: From [118]: Visual representation of a clustering scenario. Data point similarities are evaluated by the model to group them in a graph structure where the proximity reflects the similarity. Subgraph structures can be interpreted as groupings (color encoded).

3.1 Fundamental Statistics

Before proceeding with Machine Learning models, this section introduces some fundamental concepts of probabilities. The fundamentals of the fundamentals can be found in appendix *Probability Theory*, but the following will simply assume that Statistics works within its intended scope. Variables for a data set $\underline{\boldsymbol{x}}$ and variables for their assignments $\boldsymbol{y}$ have already been introduced. The interconnection of both can be described via a tuple and their *joint probability distribution*. All instances of that tuple are assumed to be *independent and identically distributed*:

$$\begin{aligned} &P(\underline{\boldsymbol{x}} = (\boldsymbol{x}_1, ...\boldsymbol{x}_N), \boldsymbol{y} = (y_1, ...y_N)) \\ &\quad = \prod_{n=1}^{N} P(\boldsymbol{x}_n, y_n) \end{aligned} \tag{3.1}$$

The probability of the entire set can be written as a product of probabilities in case of such an assumption. This strong assumption is followed by consequences, e.g. the independence of the data points. To show the validity of this claim, the *sum rule* [14] (sometimes referred to as *marginalization*) can be applied to the *marginal probability distribution*:

$$\begin{aligned} P(\underline{\boldsymbol{x}} = (\boldsymbol{x}_1, ...\boldsymbol{x}_N)) &= \sum_{\boldsymbol{y} \in \boldsymbol{\mathcal{Y}}} P(\boldsymbol{x}_1, ...\boldsymbol{x}_N, y_1, ...y_N) \\ &= \sum_{y_1 \in \mathcal{Y}_1} ... \sum_{y_N \in \mathcal{Y}_N} P(\boldsymbol{x}_1, ...\boldsymbol{x}_N, y_1, ...y_N) \\ &= \sum_{y_1 \in \mathcal{Y}_1} ... \sum_{y_N \in \mathcal{Y}_N} \prod_{n=1}^{N} P(\boldsymbol{x}_n, y_n) \\ &= \sum_{y_1 \in \mathcal{Y}_1} P(\boldsymbol{x}_1, y_1) \cdot ... \cdot \sum_{y_N \in \mathcal{Y}_N} P(\boldsymbol{x}_N, y_N) \\ &= P(\boldsymbol{x}_1) \cdot ... \cdot P(\boldsymbol{x}_N) \\ &= \prod_{n=1}^{N} P(\boldsymbol{x}_n) \end{aligned} \tag{3.2}$$

With the sum-rule/marginalization being a valid operation, Eq. (3.2) can be concluded as a consequence of Eq. (3.1). With y_n being discrete valued, this operation works on summations. For continuous values, it works with integrals. A more or less symmetric consequence can be stated in the following:

$$\begin{aligned} P(\boldsymbol{y} = (y_1, ...y_N)) &= \sum_{\underline{\boldsymbol{x}} \in \boldsymbol{\mathcal{X}}} P(\boldsymbol{x}_1, ...\boldsymbol{x}_N, y_1, ...y_N) \\ &= \sum_{\boldsymbol{x}_1 \in \mathcal{X}_1} ... \sum_{\boldsymbol{x}_N \in \mathcal{X}_N} P(\boldsymbol{x}_1, ...\boldsymbol{x}_N, y_1, ...y_N) \\ &= \sum_{\boldsymbol{x}_1 \in \mathcal{X}_1} ... \sum_{\boldsymbol{x}_N \in \mathcal{X}_N} \prod_{n=1}^{N} P(\boldsymbol{x}_n, y_n) \\ &= \sum_{\boldsymbol{x}_1 \in \mathcal{X}_1} P(\boldsymbol{x}_1, y_1) \cdot ... \cdot \sum_{\boldsymbol{x}_N \in \mathcal{X}_N} P(\boldsymbol{x}_N, y_N) \\ &= P(y_1) \cdot ... \cdot P(y_N) \\ &= \prod_{n=1}^{N} P(y_n) \end{aligned} \tag{3.3}$$

Assumption Eq. (3.1) has additional consequences for the *conditional probability distributions*. The *product rule* [14] (also referred to as *chain rule*) can be applied in combination with Eq. (3.2) & Eq. (3.3) to further show:

$$\begin{aligned} P(\underline{\boldsymbol{x}}|\boldsymbol{y}) &= P(\underline{\boldsymbol{x}}|\boldsymbol{y}) \cdot \frac{P(\boldsymbol{y})}{P(\boldsymbol{y})} \\ &= \frac{P(\underline{\boldsymbol{x}}, \boldsymbol{y})}{P(\boldsymbol{y})} \\ &= \frac{\prod_{n=1}^{N} P(\boldsymbol{x}_n, y_n)}{\prod_{n=1}^{N} P(y_n)} \\ &= \prod_{n=1}^{N} \frac{P(\boldsymbol{x}_n, y_n)}{P(y_n)} \\ &= \prod_{n=1}^{N} P(\boldsymbol{x}_n|y_n) \end{aligned} \tag{3.4}$$

$$\begin{aligned} P(\boldsymbol{y}|\underline{\boldsymbol{x}}) &= P(\boldsymbol{y}|\underline{\boldsymbol{x}}) \cdot \frac{P(\underline{\boldsymbol{x}})}{P(\underline{\boldsymbol{x}})} \\ &= \frac{P(\underline{\boldsymbol{x}}, \boldsymbol{y})}{P(\underline{\boldsymbol{x}})} \\ &= \frac{\prod_{n=1}^{N} P(\boldsymbol{x}_n, y_n)}{\prod_{n=1}^{N} P(\boldsymbol{x}_n)} \\ &= \prod_{n=1}^{N} \frac{P(\boldsymbol{x}_n, y_n)}{P(\boldsymbol{x}_n)} \\ &= \prod_{n=1}^{N} P(y_n|\boldsymbol{x}_n) \end{aligned} \tag{3.5}$$

3.2 Models for Unstructured Prediction

This section will introduce models for unstructured predictions. This means models exploit either assumption Eq. (3.1) or Eq. (3.5). Both assumptions result in assignments y being just dependent on their assigned data point $\boldsymbol{x}$. Therefore, predictions c for y are just dependent on the assigned data point. Because of the missing interconnection between predictions, these models are referred to as *models of unstructured prediction*.

3.2.1 Supervised Learning

This section will introduce models that have full knowledge about all tuples $(\boldsymbol{x}, y)$ to make predictions c. The described scenario is consistent with the concept of *supervised learning*. With assignments y being discrete, so are their predictions c, and the scenario is called *classification*. Consequently, assignments are called classes.

3.2.1.1 Generative Classifiers

Generative Classifiers (GC) are Machine Learning models that specify how to generate data using a class conditional probability function $P(\boldsymbol{x}|c = y, \boldsymbol{\theta})$ and a class *prior* $P(c = y|\boldsymbol{\theta})$ to make predictions for the *posterior* via the *Bayes Theorem* (also called *Bayes rule*) [14][88] as follows:

$$\begin{aligned} P(c = y|\boldsymbol{x}, \boldsymbol{\theta}) &\propto P(\boldsymbol{x}|c = y, \boldsymbol{\theta}) \cdot P(c = y|\boldsymbol{\theta}) \\ &= P(\boldsymbol{x}, c = y|\boldsymbol{\theta}) \end{aligned}$$

The variable $\boldsymbol{\theta}$ encodes the model as the description of its parameters. In a less formal interpretation, $P(c = y|\boldsymbol{\theta})$ encodes a probability function that reflects the degree of a priori certainty of an outcome y, while $P(\boldsymbol{x}|c = y, \boldsymbol{\theta})$ encodes a probability function reflecting the degree of certainty that a data point $\boldsymbol{x}$ was generated by it conditioned on y. $P(c = y|\boldsymbol{x}, \boldsymbol{\theta})$ reflects the plausibility to assume a prediction c for y based on observing $\boldsymbol{x}$. Consequently, GCs chose predictions with maximal plausibility by the *Maximum A Posteriori Prediction* (MAP):

$$\begin{aligned} \hat{c} &= \underset{y \in \mathcal{Y}}{\operatorname{argmax}}\, P(\boldsymbol{x}|c = y, \boldsymbol{\theta}) \cdot P(c = y|\boldsymbol{\theta}) \\ &= \underset{y \in \mathcal{Y}}{\operatorname{argmax}}\, P(\boldsymbol{x}, c = y|\boldsymbol{\theta}) \end{aligned}$$

A plethora of Machine Learning models work within the family of GCs. This section will describe only a sub-set, and it starts with the most simple one. *Naive Bayes Classifiers* (NBC) are a model family which is probably the most commonly used one in Machine Learning. NBC rely on the *Naive Bayes Model* (NBM), also known as the *Idiot Bayes Model*, which is perhaps the simplest example where a conditional parametrization is combined with conditional independence assumptions to produce a very compact representation of a high-dimensional probability distribution [66]. In essence, the NBM assumes features to be conditional independent given their class label, resulting in a product of one-dimensional probability functions, see Fig. 3.3 and the following:

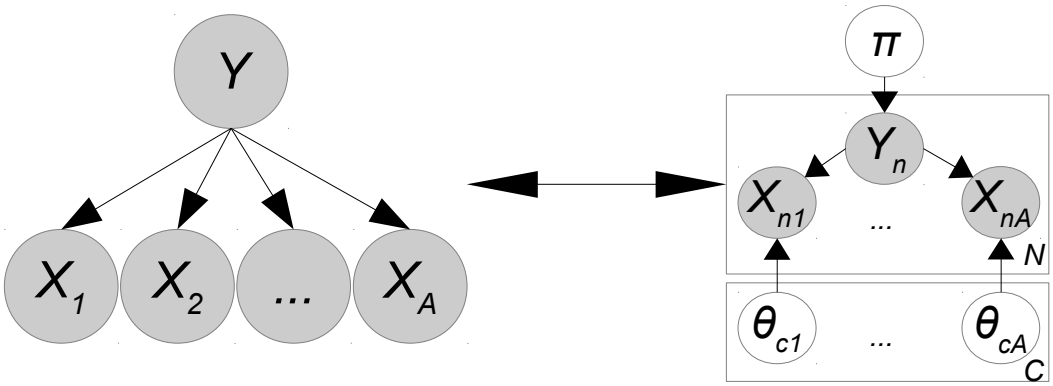


FIGURE 3.3: Left: The Bayesian Network Graph for a Naive Bayes Model [66]. Right: Naive Bayes Classifier as a Directed Graphical Model with single plates [88]. Nodes reflect model variables and edges their interaction: $A \rightarrow B = P(B|A)$. Model parameter are stated as unknown variables and are marked in white.

$$P(\boldsymbol{x}|c = y, \boldsymbol{\theta}) = \prod_{a=1}^{A} P(x_a|c = y, \boldsymbol{\theta})$$

Its simplicity combined with a reasonable classification performance makes NBC an adequate choice for baseline comparisons. This arises from two observations: even if the assumption of the NBM is not true, it often results in classifiers that work well [32] and with comparably few parameters NBC are relatively immune to overfitting [88]. An additional benefit of this easy model lies in the fact that it can handle different types of features by adapting the particular choice of class conditional probability functions. Even mixing these types is possible. A small set of common choices is listed below [88]:

- **real-valued feature**: *Gaussian Naive Bayes Model* with multiple *Gaussian/Normal Distribution*: $P(x_a|c = y, \boldsymbol{\theta}) = \mathcal{N}(x_a|\mu_{ca}, \sigma^2_{ca})$ and $\boldsymbol{\theta} = (\mu_{ca}, \sigma^2_{ca})$ being the class specific mean and variance for that feature.
- **binary feature**: *Multivariate Bernoulli Naive Bayes Model* with multiple *Bernoulli Distribution*: $P(x_a|c = y, \boldsymbol{\theta}) = Ber(x_a|p_{ca})$ and $\boldsymbol{\theta} = p_{ca}$ being the class specific proportion comprising that feature.
- **multicategorical feature**: *Multinomial Naive Bayes Model* with a *Multinomial Distribution*: $P(x_a|c = y, \boldsymbol{\theta}) = Mult(x_a|\boldsymbol{p}_c)$ and $\boldsymbol{\theta} = \boldsymbol{p}_c$ being the class specific proportion for that particular feature category.

Even though being one of the simplest models in Machine Learning, NBC remain popular because of their reasonable performance, low computational resources and easy interpretation. In general, GCs are not restricted by the NBM assumption. By applying multivariate conditional probability functions, GCs can work with correlated features. The *Multivariate Gaussian/Normal Distribution* forms a joint probability density function for continuous variables that are widely used in GCs [88]. When applied as a class conditional density function, the GCs result in a model family called *Gaussian Discriminate Analysis*. Such a GC is called *Quadratic Discriminant Analysis* (QDA) [44] and forms an inherently quadratic discriminant function [33] (see *Determinant of a Matrix* & *Inverse of a Matrix*):

$$
\begin{aligned}
P(\boldsymbol{x}|c = y, \boldsymbol{\theta}) &= \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \\
&= (2\pi)^{-\frac{A}{2}} \cdot |\boldsymbol{\Sigma}_c|^{-\frac{1}{2}} \cdot \exp\Big(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_c)^T \boldsymbol{\Sigma}_c^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_c) \Big)
\end{aligned}
$$

By restricting the class specific covariance matrix to be a *Diagonal Matrix*, the *Gaussian Naive Bayes Classifier* arises. In case of restricting the covariance matrices to be shared across classes, the GC is known as a *Linear Discriminant Analysis* (LDA) [43] which forms a linear GC [33]. Further restricting the shared covariance matrix to be diagonal, the *diagonal LDA* [88] arises as an NBC. Nonetheless, besides the model specific assumptions of the class conditional probability function, GCs comprise an overall rather implicit assumption. To state it clear and explicit, the *(complete data) Likelihood* of GCs can be described as follows:

$$
\begin{aligned}
&P(\underline{\boldsymbol{x}}, \boldsymbol{c}|\boldsymbol{y}, \boldsymbol{\theta}) \\
&=_* \prod_{n=1}^{N} \prod_{y \in \mathcal{Y}} P(\boldsymbol{x}_n, c_n = y|\boldsymbol{\theta})^{\mathcal{I}(y_n = y)} \\
&= \prod_{n=1}^{N} \prod_{y \in \mathcal{Y}} \big(P(\boldsymbol{x}_n|c_n = y, \boldsymbol{\theta}) \cdot P(c_n = y|\boldsymbol{\theta}) \big)^{\mathcal{I}(y_n = y)}
\end{aligned}
$$

The function $\mathcal{I}(\cdot)$ is the *indicator function*, being one if its expression is true and zero otherwise. The Likelihood of GCs is rigorously restricted by assumption Eq. (3.1), which is exploited in the marked line (*). As mentioned in the previous section, this assumption results in the consequences Eq. (3.2) - Eq. (3.5) implicitly. This generous assumption results in the low computational demand of GCs, but more carefully selected assumptions might yield more powerful models for class predictions.

Up to this point, the section focused on the model description and class prediction. This last paragraph will now focus on learning such models. For the sake of focus, the description restricts itself towards *Maximum Likelihood Estimation* (MLE) [42]. The MLE chooses the point estimate $\hat{\boldsymbol{\theta}}$ within all possible parameters $\boldsymbol{\theta}$ of the (valid) parameter space $\boldsymbol{\Theta}$ in the model by the maximum of its Likelihood:

$$
\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\operatorname{argmax}} P(\underline{\boldsymbol{x}}, \boldsymbol{c}|\boldsymbol{y}, \boldsymbol{\theta})
$$

Because of several valuable properties of the logarithm, one normally maximizes the Log-Likelihood. The parameter maximizing the Log-Likelihood also maximizes the Likelihood because of the monotonic property of the logarithm.

$$
\begin{aligned}
&\ln P(\underline{\boldsymbol{x}}, \boldsymbol{c}|\boldsymbol{y}, \boldsymbol{\theta}) \\
&= \ln \prod_{n=1}^{N} \prod_{y \in \mathcal{Y}} \left(P(\boldsymbol{x}_n | c_n = y, \boldsymbol{\theta}) \cdot P(c_n = y|\boldsymbol{\theta}) \right)^{\mathcal{I}(y_n = y)} \\
&= \sum_{n=1}^{N} \sum_{y \in \mathcal{Y}} \mathcal{I}(y_n = y) \cdot \ln \left(P(\boldsymbol{x}_n | c_n = y, \boldsymbol{\theta}) \cdot P(c_n = y|\boldsymbol{\theta}) \right) \\
&= \sum_{n=1}^{N} \sum_{y \in \mathcal{Y}} \mathcal{I}(y_n = y) \cdot \left(\ln P(\boldsymbol{x}_n | c_n = y, \boldsymbol{\theta}) + \ln P(c_n = y|\boldsymbol{\theta}) \right) \\
&= \sum_{n=1}^{N} \sum_{y \in \mathcal{Y}} \mathcal{I}(y_n = y) \cdot \ln P(\boldsymbol{x}_n | c_n = y, \boldsymbol{\theta}) + \mathcal{I}(y_n = y) \cdot \ln P(c_n = y|\boldsymbol{\theta})
\end{aligned}
$$

The logarithm creates a sum of components reflecting the conditional probability and the class prior. Maximization of the Log-Likelihood is done by taking the partial derivative in respect to the parameter of interest. The derivative of a sum is the sum of its derivatives. In general, the MLE of GCs maximizes the joint probability distribution $P(\underline{\boldsymbol{x}}, \boldsymbol{c} = \boldsymbol{y}|\boldsymbol{\theta})$. The direct maximization for the predictive distribution $P(\boldsymbol{c} = \boldsymbol{y}|\underline{\boldsymbol{x}}, \boldsymbol{\theta})$ is not the goal of GCs. The MLE for the QDA, LDA and GNB model is given as an example below. To declutter the notation, π_c is written instead of the class prior. Often, one will rather find the variance term in the form of the *Bessel correction*.

$$
\begin{aligned}
&\underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\operatorname{argmax}} \, P(\underline{\boldsymbol{x}}, \boldsymbol{c}|\boldsymbol{y}, \boldsymbol{\theta}) \\
\rightarrow \hat{\pi}_c &= \frac{\sum_{n=1}^{N} \mathcal{I}(c = y_n)}{\sum_{n=1}^{N} \sum_{c' \in \mathcal{C}} \mathcal{I}(c' = y_n)} \\
\rightarrow \hat{\boldsymbol{\mu}}_c &= \frac{1}{\sum_{n=1}^{N} \mathcal{I}(c = y_n)} \cdot \sum_{n=1}^{N} \mathcal{I}(c = y_n) \cdot \boldsymbol{x}_n \\
\rightarrow \hat{\boldsymbol{\Sigma}}_c &= \frac{1}{\sum_{n=1}^{N} \mathcal{I}(c = y_n)} \sum_{n=1}^{N} \mathcal{I}(c = y_n) \cdot (\boldsymbol{x}_n - \hat{\boldsymbol{\mu}}_c)(\boldsymbol{x}_n - \hat{\boldsymbol{\mu}}_c)^T \\
\rightarrow \hat{\boldsymbol{\Sigma}} &= \sum_{c=1}^{C} \frac{\sum_{n=1}^{N} \mathcal{I}(c = y_n)}{\sum_{n=1}^{N} \sum_{c' \in \mathcal{C}} \mathcal{I}(c' = y_n)} \cdot \hat{\boldsymbol{\Sigma}}_c
\end{aligned}
$$

3.2.1.2 Discriminative Classifiers

Discriminative Classifiers (DC) can be seen as the counterpart to *Generative Classifiers* (GC). While GCs create a joint probability function $P(\underline{\boldsymbol{x}}, \boldsymbol{c} = \boldsymbol{y}|\boldsymbol{\theta})$ to derive the posterior $P(\boldsymbol{c} = \boldsymbol{y}|\underline{\boldsymbol{x}}, \boldsymbol{\theta})$ via the *Bayes Theorem*, DCs directly fit the model towards the posterior [14][88]. A plethora of Machine Learning models work within this model family. DCs which are linear in their parameters are significantly simpler in model fitting [88]. This is implemented by restricting the model parameters $\boldsymbol{\theta}$ to be a linear combination with the data, e.g. $\boldsymbol{x}^T\boldsymbol{\theta}$.

The *Logistic Regression* (LR) [11] model is commonly used in Machine Learning, and it is designed for binary classification tasks. Predictions of the LR model are realized as follows:

$$P(c = 1|\boldsymbol{x}, \boldsymbol{\theta}) = \sigma(\boldsymbol{x}^T\boldsymbol{\theta})$$

The function $\sigma(\cdot)$ is known as the *logistic function* (sometimes referred to as the *sigmoid function*) and maps the whole real line to $[0, 1]$, squashing it into a probabilistic interpretation [88], see Fig. 3.4:

$$\sigma(z) = \frac{1}{1 + \exp(-z)} = \frac{\exp(z)}{\exp(z) + 1}$$

By using the symmetric property $\sigma(-z) = 1 - \sigma(z)$ of the logistic function, the LR model can be represented as a log ratio of probabilities, also known as the *log odds* [14]. By understanding the log odds, the LR model remains interpretable in respect to its parameters:

$$\ln \frac{P(c = 1|\boldsymbol{x}, \boldsymbol{\theta})}{P(c = 0|\boldsymbol{x}, \boldsymbol{\theta})} = \ln \frac{\exp(\boldsymbol{x}^T\boldsymbol{\theta})/(1 + \exp(\boldsymbol{x}^T\boldsymbol{\theta}))}{1/(1 + \exp(\boldsymbol{x}^T\boldsymbol{\theta}))} = \boldsymbol{x}^T\boldsymbol{\theta}$$

The model comprises the amount of parameter as the feature space of $\boldsymbol{x}$, say A. In comparisons, *Gaussian Naive Bayes Models* (models with rigorously restricting assumptions) would need $2 \cdot A$ parameters for both means, $2 \cdot A$ parameters for all variances and additional 2 parameters for the class prior. In respect to parameter size, there is a clear advantage in the compact parameter representation of the LR model.

The LR model can be extended to a broader model family called *Generalized Linear Models* (GLM) [83], by exchanging the logistic function $\sigma(\cdot)$ with a particular choice of functions, called *mean functions*, which are more commonly referred in its inverse form as *link functions* [88]. Different choices for these functions, results in different models such as the *Probit Regression* [41][15], *Complementary-Log-log Regression* [83] model etc. By restricting the mean function to be an invertible monotonic function, e.g. a *cumulative distribution function* (CDF), the linear combination will be mapped into $[0, 1]$ and a probabilistic interpretation is given via the particular choice of CDF. In their classical form, GLM use *canonical link functions* (CDF of the *Exponential Family*) but non-canonical links have been used as well. Common choices for canonical link functions are illustrated in Fig. 3.4 and listed with a small description as follows:

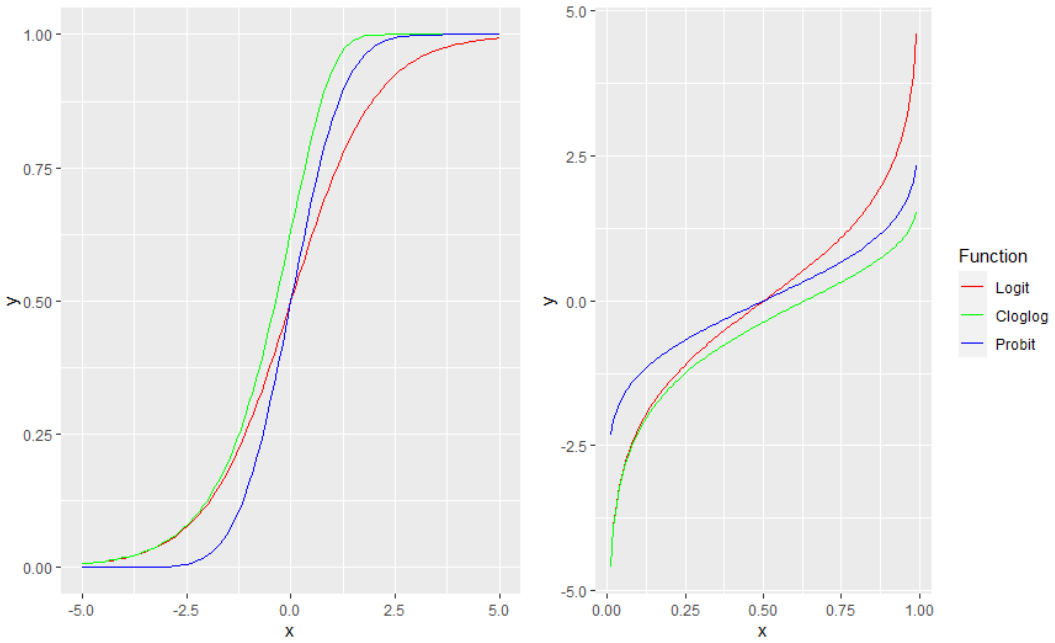


FIGURE 3.4: Overview of different mean & link functions (left & right). The particular choice of the link function will result in a specific model instance: logit (Logistic Regression [11]), probit (Probit Regression [41][15]), cloglog (Complementary-Log-Log Regression [83])

- **logit**: $\ln(\pi/(1-\pi))$ [83] for the *Logit-Model* [11] resulting in the CDF of a *Logistic Distribution*:
  $P(c=1|\boldsymbol{x},\boldsymbol{\theta}) = \sigma(\boldsymbol{x}^T\boldsymbol{\theta})$

- **probit**: $\Phi^{-1}(\pi)$ [83] for the *Probit-Model* [41][15] resulting in the CDF of a standard *Gaussian/Normal Distribution*:
  $P(c=1|\boldsymbol{x},\boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\boldsymbol{x}^T\boldsymbol{\theta}} \exp\Big(-\frac{1}{2}t^2\Big)\mathrm{d}t$

- **cloglog**: $\ln(-\ln(1-\pi))$ [83] for the *cloglog-Model* resulting in the CDF of an *extreme value distribution* (or *Gumbel Distribution*):
  $P(c=1|\boldsymbol{x},\boldsymbol{\theta}) = 1 - \exp\big(-\exp(\boldsymbol{x}^T\boldsymbol{\theta})\big)$

The current formulation of the DCs motivated them for binary classification but they can be extended to multi-categorical cases as well, e.g. the *Multinomial Logit Model* [30]. For the sake of focus, the description will remain in the binary case. With the interpretation of the canonical link functions given, a revision of the LR model is needed. The LR model can be seen as a probabilistic process [88] which is illustrated in Fig. 3.5 and described as follows:

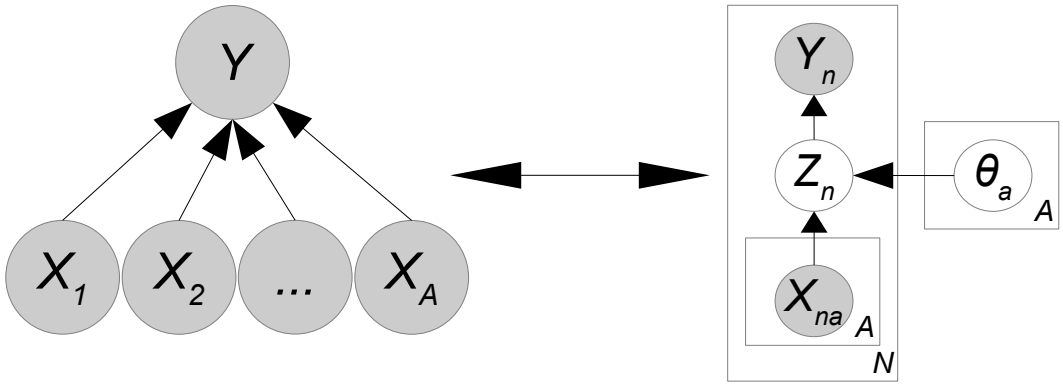


FIGURE 3.5: Left: The Bayesian Network Graph for the Logistic Regression model. Right: Logistic Regression as a Directed Graphical Model with plates. Nodes reflect model variables and edges their interaction: $A \rightarrow B = P(B|A)$. Model parameter are stated as unknown variables and are marked in white.

$$
\begin{aligned}
z_n &= \boldsymbol{x}^T\boldsymbol{\theta} + \epsilon_n \\
\epsilon_n &\sim logistic(0,1) \rightarrow E[\epsilon] = 0, Var[\epsilon] = \pi^2/3 \\
y_n &= \mathcal{I}(z_n \geq 0) \\
P(c_n = 1|\boldsymbol{x}_n, \boldsymbol{\theta}) &= \int_{-\boldsymbol{x}^T\boldsymbol{\theta}}^{\infty} f(\epsilon)\mathrm{d}\epsilon = \int_{-\infty}^{\boldsymbol{x}^T\boldsymbol{\theta}} f(\epsilon)\mathrm{d}\epsilon = F(\boldsymbol{x}^T\boldsymbol{\theta}) = \sigma(\boldsymbol{x}^T\boldsymbol{\theta})
\end{aligned}
$$

The switching integral bounds could be used because of the symmetric properties of the *Logistic Distribution*. DCs comprise some promising properties, which needs a clear comparison to the previously introduced GCs. Therefore, the *(complete data) Likelihood* [14] will be inspected:

$$
\begin{aligned}
&P(\boldsymbol{c}|\underline{\boldsymbol{x}}, \boldsymbol{y}, \boldsymbol{\theta}) \\
&= \prod_{n=1}^{N} P(c_n = 1|\boldsymbol{x}_n, \boldsymbol{\theta})^{\mathcal{I}(y_n=1)} \cdot \big(1 - P(c_n = 1|\boldsymbol{x}_n, \boldsymbol{\theta})\big)^{1-\mathcal{I}(y_n=1)}
\end{aligned}
$$

The assumption of the model is Eq. (3.5), which is less restrictive than Eq. (3.1) in the case of GCs, with its consequences Eq. (3.2) - Eq. (3.5). With less restricting assumptions, DCs have the potential to achieve better predictive performances than GCs, if their assumptions are incorrect. Empirically undermined, *Gaussian* GCs need less training data than a (Logistic) DC to achieve a certain level of performance, but if these assumptions are incorrect the DC will do better [89].

Up to this point, the section focused on the model description and class prediction. This last paragraph will now focus on learning such models. For the sake of focus, this description restricts itself towards *Maximum Likelihood Estimation* (MLE) [42]. The MLE is coupled with the parameter itself. By recursive application of this estimate on an initial starting point, the estimate will (under mild assumptions) converge towards the MLE. This procedure for maximizing the conditional Log-Likelihood is formally known as *Gradient Ascent*. The procedure is more often applied by minimizing the negative conditional Log-Likelihood (also known as the

Cross-Entropy) and it is formally known as *Gradient Descent*, see Algo. 1. Other more sophisticated algorithms exist, e.g. *Coordinate Descent* [45], *Conjugate Gradient* [58], etc., which differ in speed of convergence towards the MLE. As Fig. 3.6 high-lights, the convergence of the Likelihood needs to be analyzed carefully. All in all, learning DCs is more complicated than in the case of GCs, as a direct consequence of the less restricting assumption Eq. (3.5) instead of Eq. (3.1). Fitting a DC maximizes the conditional Likelihood $P(\boldsymbol{c} = \boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta})$ where GCs maximize the Likelihood $P(\underline{\boldsymbol{x}}, \boldsymbol{c} = \boldsymbol{y}|\boldsymbol{\theta})$ [88]. Obviously, this difference can lead to different predictions.

$$
\begin{aligned}
&\frac{\partial \ln P(\boldsymbol{c}|\underline{\boldsymbol{x}}, \boldsymbol{y}, \boldsymbol{\theta})}{\partial \theta_a} \\
&= \partial \Big(\sum_{n=1}^{N} \mathcal{I}(y_n = 1) \cdot \ln P(c_n = 1|\boldsymbol{x}_n, \boldsymbol{\theta}) \\
&\qquad + \big(1 - \mathcal{I}(y_n = 1)\big) \cdot \ln \big(1 - P(c_n = 1|\boldsymbol{x}_n, \boldsymbol{\theta})\big) \Big) / \partial \theta_a \\
&= \sum_{n=1}^{N} \mathcal{I}(y_n = 1) \cdot \partial \ln P(c_n = 1|\boldsymbol{x}_n, \boldsymbol{\theta}) / \partial \theta_a \\
&\qquad + \big(1 - \mathcal{I}(y_n = 1)\big) \cdot \partial \ln \big(1 - P(c_n = 1|\boldsymbol{x}_n, \boldsymbol{\theta})\big) / \partial \theta_a \\
&= \sum_{n=1}^{N} \frac{\mathcal{I}(y_n = 1)}{P(c_n = 1|\boldsymbol{x}_n, \boldsymbol{\theta})} \cdot \partial P(c_n = 1|\boldsymbol{x}_n, \boldsymbol{\theta}) / \partial \theta_a \\
&\qquad + \frac{\big(1 - \mathcal{I}(y_n = 1)\big)}{1 - P(c_n = 1|\boldsymbol{x}_n, \boldsymbol{\theta})} \cdot \partial \big(1 - P(c_n = 1|\boldsymbol{x}_n, \boldsymbol{\theta})\big) \partial \theta_a \\
&= \sum_{n=1}^{N} \frac{\mathcal{I}(y_n = 1)}{\sigma(\boldsymbol{x}_n^T \boldsymbol{\theta})} \cdot \partial \sigma(\boldsymbol{x}_n^T \boldsymbol{\theta}) / \partial \theta_a \\
&\qquad + \frac{\big(1 - \mathcal{I}(y_n = 1)\big)}{1 - \sigma(\boldsymbol{x}_n^T \boldsymbol{\theta})} \cdot \partial \big(1 - \sigma(\boldsymbol{x}_n^T \boldsymbol{\theta})\big) \partial \theta_a \\
&= \sum_{n=1}^{N} \frac{\mathcal{I}(y_n = 1)}{\sigma(\boldsymbol{x}_n^T \boldsymbol{\theta})} \cdot \sigma(\boldsymbol{x}_n^T \boldsymbol{\theta}) \big(1 - \sigma(\boldsymbol{x}_n^T \boldsymbol{\theta})\big) x_{na} \\
&\qquad - \frac{\big(1 - \mathcal{I}(y_n = 1)\big)}{1 - \sigma(\boldsymbol{x}_n^T \boldsymbol{\theta})} \cdot \big(1 - \sigma(\boldsymbol{x}_n^T \boldsymbol{\theta})\big) \sigma(\boldsymbol{x}_n^T \boldsymbol{\theta}) x_{na} \\
&= \sum_{n=1}^{N} \mathcal{I}(y_n = 1) \big(1 - \sigma(\boldsymbol{x}_n^T \boldsymbol{\theta})\big) x_{na} \\
&\qquad - \big(1 - \mathcal{I}(y_n = 1)\big) \sigma(\boldsymbol{x}_n^T \boldsymbol{\theta}) x_{na} \\
&= \sum_{n=1}^{N} \big(\mathcal{I}(y_n = 1) - \mathcal{I}(y_n = 1) \sigma(\boldsymbol{x}_n^T \boldsymbol{\theta})\big) x_{na} \\
&\qquad - \big(\sigma(\boldsymbol{x}_n^T \boldsymbol{\theta}) - \mathcal{I}(y_n = 1) \sigma(\boldsymbol{x}_n^T \boldsymbol{\theta})\big) x_{na} \\
&= \sum_{n=1}^{N} \big(\mathcal{I}(y_n = 1) - \sigma(\boldsymbol{x}_n^T \boldsymbol{\theta})\big) x_{na} \\
&= \nabla E[\theta_a]
\end{aligned}
$$

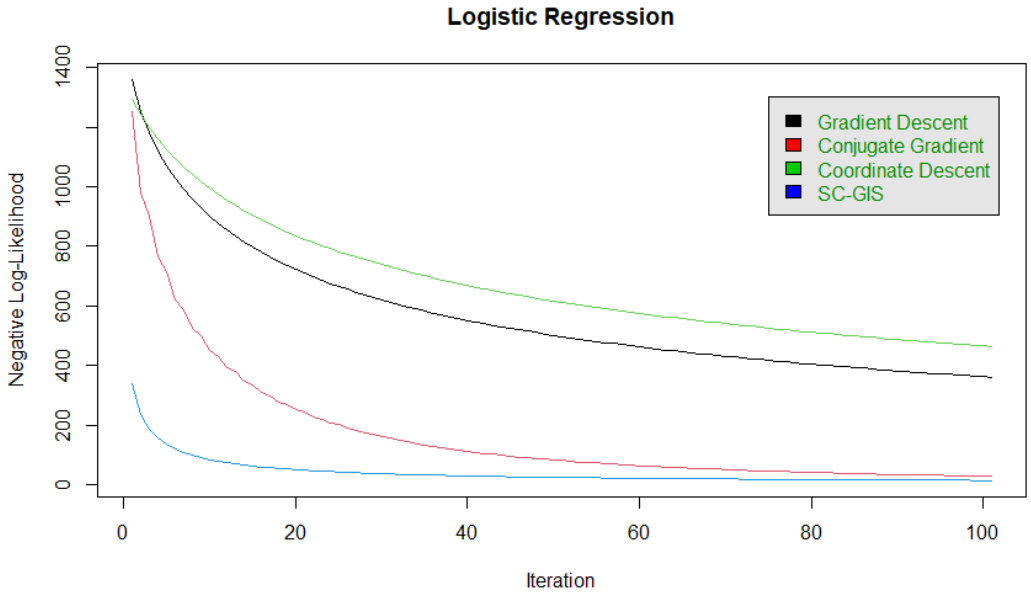


FIGURE 3.6: Overview of different optimizers for learning the Logistic Regression model. Different speeds of convergence can be observed for the different methods: Gradient Descent, Coordinate Descent [45], Conjugate Gradient [58] and Sequential Conditional Generalized Iterative Scaling (SC-GIS) [47].

Algorithm 1 Gradient Descent (GD): This algorithm naturally arises as the Maximum Likelihood Estimation [42] technique for (probabilistic) Discriminative Classifiers. The GD minimizes the negative conditional Log-Likelihood (*Cross-Entropy*).

procedure GDALGO
  Initialisation:
    initialize parameter 'randomly'
  Iteration:
    Update: $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \cdot \nabla E[\boldsymbol{\theta}^{(t)}]$
  Termination:
    Either: (i) iteration t exceeds maximum predefined number
    Or: (ii) saturation $P(\boldsymbol{c}|\underline{\boldsymbol{x}}, \boldsymbol{y}, \boldsymbol{\theta}^{(t+1)}) - P(\boldsymbol{c}|\underline{\boldsymbol{x}}, \boldsymbol{y}, \boldsymbol{\theta}^{(t)}) < \delta$

3.2.2 Unsupervised Learning

This section will introduce models that have only partial knowledge about all tuples $(\boldsymbol{x}, y)$. The missing information on y will be predicted via c. The described scenario is consistent with the concept of *unsupervised learning*. With the unknown assignments y being discrete, so are their predictions c, and the scenario is called *clustering*. Consequently, assignments are called clusters.

3.2.2.1 Expectation Maximization

Expectation Maximization (EM) [31] is a learning method for probabilistic models in Machine Learning that are most commonly used for clustering. The EM works with missing information in the form of latent variables $\boldsymbol{y}$. By *sum-rule/marginalization* over the latent variables $\boldsymbol{y}$ combined with the observable data $\underline{\boldsymbol{x}}$, the *(incomplete data) Likelihood* is defined as follows:

$$P(\underline{\boldsymbol{x}}|\boldsymbol{\theta}) = \sum_{\boldsymbol{y}\in\mathcal{Y}} P(\underline{\boldsymbol{x}}, \boldsymbol{y}|\boldsymbol{\theta})$$

Even though the assignments $\boldsymbol{y}$ are not known, its joint probability distribution with $\underline{\boldsymbol{x}}$ exist. Maximizing this distribution will result in an approximation of the unknown assignments. The basic *Maximum Likelihood Estimation* (MLE) [42] comes to its limits, but the EM technique overcomes this problem as a generalized version of it. For that, $P(\underline{\boldsymbol{x}}|\boldsymbol{\theta})$ (more precisely its logarithm) needs to be decomposed into the following:

$$\begin{aligned}
&\ln P(\underline{\boldsymbol{x}}|\boldsymbol{\theta}) \\
&= \sum_{\boldsymbol{y}\in\mathcal{Y}} q(\boldsymbol{y}) \cdot \ln P(\underline{\boldsymbol{x}}|\boldsymbol{\theta}) \\
&= \sum_{\boldsymbol{y}\in\mathcal{Y}} q(\boldsymbol{y}) \cdot \ln \frac{P(\boldsymbol{y}|\underline{\boldsymbol{x}}, \boldsymbol{\theta}) \cdot P(\underline{\boldsymbol{x}}|\boldsymbol{\theta})}{P(\boldsymbol{y}|\underline{\boldsymbol{x}}, \boldsymbol{\theta})} \\
&= \sum_{\boldsymbol{y}\in\mathcal{Y}} q(\boldsymbol{y}) \cdot \ln \frac{P(\underline{\boldsymbol{x}}, \boldsymbol{y}|\boldsymbol{\theta})}{P(\boldsymbol{y}|\underline{\boldsymbol{x}}, \boldsymbol{\theta})} \\
&= \sum_{\boldsymbol{y}\in\mathcal{Y}} q(\boldsymbol{y}) \cdot \ln \frac{P(\underline{\boldsymbol{x}}, \boldsymbol{y}|\boldsymbol{\theta})/q(\boldsymbol{y})}{P(\boldsymbol{y}|\underline{\boldsymbol{x}}, \boldsymbol{\theta})/q(\boldsymbol{y})} \\
&= \sum_{\boldsymbol{y}\in\mathcal{Y}} q(\boldsymbol{y}) \cdot \left(\ln \frac{P(\underline{\boldsymbol{x}}, \boldsymbol{y}|\boldsymbol{\theta})}{q(\boldsymbol{y})} - \ln \frac{P(\boldsymbol{y}|\underline{\boldsymbol{x}}, \boldsymbol{\theta})}{q(\boldsymbol{y})} \right) \\
&= \sum_{\boldsymbol{y}\in\mathcal{Y}} q(\boldsymbol{y}) \cdot \ln \frac{P(\underline{\boldsymbol{x}}, \boldsymbol{y}|\boldsymbol{\theta})}{q(\boldsymbol{y})} - \sum_{\boldsymbol{y}\in\mathcal{Y}} q(\boldsymbol{y}) \cdot \ln \frac{P(\boldsymbol{y}|\underline{\boldsymbol{x}}, \boldsymbol{\theta})}{q(\boldsymbol{y})} \\
&= L(q, \boldsymbol{\theta}) + KL(q||p) \\
&\geq L(q, \boldsymbol{\theta})
\end{aligned}$$

This decomposition results in two recognizable parts, which serve as a suitable approximating lower bound for the Likelihood [14]. The term on the right can be recognized as a *Kullback-Leibler Divergence* [74] and *Information Theory* [122] describes its values as non-negative. Therefore, $L(q, \boldsymbol{\theta})$ is a suitable lower bound of $\ln P(\underline{\boldsymbol{x}}|\boldsymbol{\theta})$ and, because of the monotonic property of the logarithm, of $P(\underline{\boldsymbol{x}}|\boldsymbol{\theta})$. Unfortunately,

$q(\boldsymbol{y})$ is not known and $\boldsymbol{y}$ is not observable. But it can be approximated with the expected value of the posterior from an assumed *Generative Model*. This approximation is formally known as the *E-Step* (E) [14]:

$$\begin{aligned} L(q, \boldsymbol{\theta}) &= \sum_{\boldsymbol{y} \in \mathcal{Y}} q(\boldsymbol{y}) \cdot \ln \frac{P(\underline{\boldsymbol{x}}, \boldsymbol{y} | \boldsymbol{\theta})}{q(\boldsymbol{y})} \\ &\simeq \sum_{\boldsymbol{y} \in \mathcal{Y}} P(\boldsymbol{y} | \underline{\boldsymbol{x}}, \boldsymbol{\theta}^{(t)}) \cdot \ln \frac{P(\underline{\boldsymbol{x}}, \boldsymbol{y} | \boldsymbol{\theta})}{P(\boldsymbol{y} | \underline{\boldsymbol{x}}, \boldsymbol{\theta}^{(t)})} \\ &= \sum_{\boldsymbol{y} \in \mathcal{Y}} P(\boldsymbol{y} | \underline{\boldsymbol{x}}, \boldsymbol{\theta}^{(t)}) \cdot \ln P(\underline{\boldsymbol{x}}, \boldsymbol{y} | \boldsymbol{\theta}) \\ &\quad - \sum_{\boldsymbol{y} \in \mathcal{Y}} P(\boldsymbol{y} | \underline{\boldsymbol{x}}, \boldsymbol{\theta}^{(t)}) \cdot \ln P(\boldsymbol{y} | \underline{\boldsymbol{x}}, \boldsymbol{\theta}^{(t)}) \\ &= Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) + H[\boldsymbol{y} | \underline{\boldsymbol{x}}] \\ &\geq Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) \end{aligned}$$

The term on the right can be recognized as an *Entropy* [122] and Information Theory describes its values as non-negative. Therefore, $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)})$ is a suitable lower bound of $L(q, \boldsymbol{\theta})$ which is a suitable lower bound of $\ln P(\underline{\boldsymbol{x}} | \boldsymbol{\theta})$ and $P(\underline{\boldsymbol{x}} | \boldsymbol{\theta})$. The maximization of this lower bound for $\boldsymbol{\theta}^{(t)}$ is formally known as the *M-Step* (M) [14]. At its heart, the EM algorithm iterates (E) and (M) via $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)})$ till this lower bound reaches its maximum, see Algo. 2. This maximum is the MLE. The EM is a simple iterative algorithm, often with closed-form updates in each step [88]. Unfortunately, the true underlying distribution might yield several local optima and the EM algorithm will get stuck in one local optimum. Several initializations from different starting points are a necessity, and Fig. 3.7 illustrates the convergence of exemplary EM runs.

Algorithm 2 Expectation Maximization (EM) [31]: : This algorithm naturally arises as the Maximum Likelihood Estimation [42] technique for Generative Models with latent variables. The EM maximizes the (incomplete data) Likelihood.

procedure EMALGO
    Initialisation:
        Either: (i) initialize cluster assignments $\boldsymbol{y}$ 'randomly'
        Or: (ii) initialize model parameters $\boldsymbol{\theta}^{(0)}$ 'randomly'
    Iteration:
        (E): Evaluate $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)})$
        (M): Update $\boldsymbol{\theta}^{(t+1)} = \text{argmax}_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)})$
    Termination:
        Either: (i) iteration t exceeds a maximal predefined number
        Or: (ii) saturation of increase $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t+1)}) - Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) < \delta$

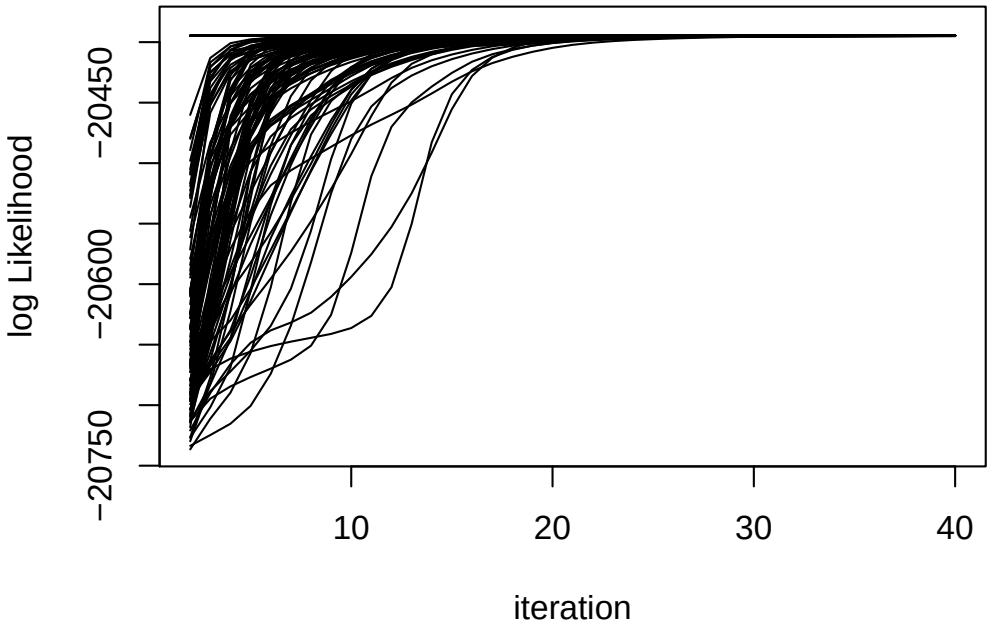


FIGURE 3.7: From [115]: Iterative optimization of $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)})$ during the Expectation Maximization [31]. Several initializations should be used to deal with multiple local optima. The best run with the highest value of $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)})$ is used to approximate the Maximum-Likelihood-Estimate [42].

3.2.2.2 Finite Mixture Models

Finite Mixture Models (FMM) are one of the most applied models in Machine Learning for clustering and are defined by their *(incomplete data) Likelihood* as follows:

$$\begin{aligned} P(\boldsymbol{x}|\boldsymbol{\theta}) &= \sum_{y \in \mathcal{Y}} P(\boldsymbol{x}, c = y|\boldsymbol{\theta}) \\ &= \sum_{y \in \mathcal{Y}} P(\boldsymbol{x}|c = y, \boldsymbol{\theta}) \cdot P(c = y|\boldsymbol{\theta}) \end{aligned}$$

The conditional probability function $P(\boldsymbol{x}|c = y, \boldsymbol{\theta})$ is referred to as the k-th *base distribution* and $P(c = y|\boldsymbol{\theta})$ is referred to as a discrete *prior* [88]. A widely used mixture model is the *Gaussian Mixture Model* (GMM), also known as *Mixture of Gaussians*. GMMs comprise base distributions of the *Multivariate Gaussian/Normal Distribution*. Given a sufficiently large number of mixture components, a GMM can be used to approximate any density defined on $\mathbb{R}^D$ [88]. Other choices for base distributions are possible to form different FMMs. Data point assignments are done via $P(c = y|\boldsymbol{x}, \boldsymbol{\theta})$, which is called the responsibility of cluster y given $\boldsymbol{x}$. The procedure is called *soft clustering*, and is identical to the computations performed when using *Generative Classifiers* (GC) [88]. Assignments via the *Maximum A Posteriori Prediction* are called *hard clustering*.

There is a strong connection between FMMs with GCs. Indeed, FMMs are just GCs by *sum-rule/marginalization* over the unknown assignments. Because of this close connection, it is clear that the same properties hold for FMMs as in the case of GCs. The Likelihood exploits assumption Eq. (3.1). This results in the same structure of the *Graphical Model* [88] in case of FMMs as in GCs, see Fig. 3.8.

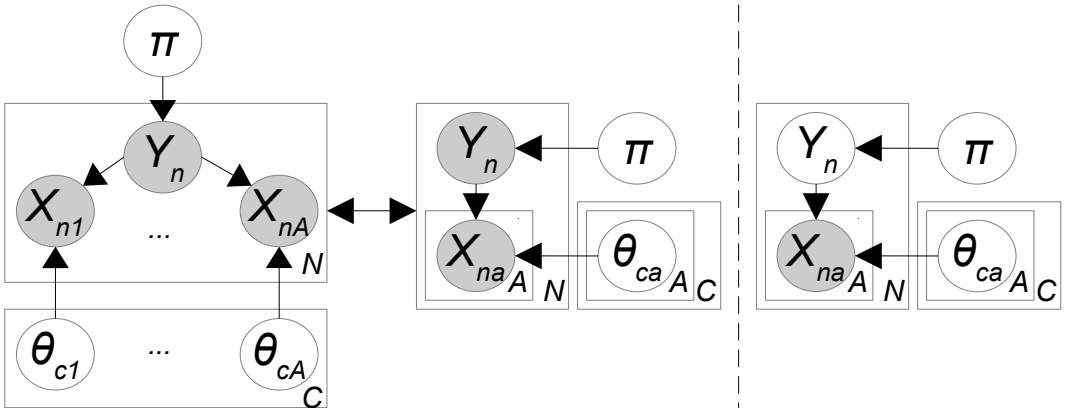


FIGURE 3.8: Comparison of the Directed Graphical Model of a Naive Bayes Classifier with single plates (left) and with multiple plates (middle). The Naive Bayes Classifier with a latent class assignment (right) resembles a Finite Mixture Model. These Graphical Models comprise a structural equivalence in case of the shared Naive Bayes assumption. Nodes reflect model variables and edges their interaction: $A \rightarrow B = P(B|A)$. Latent (unknown) variables are marked white.

Up to this point, the section focused on the model description and cluster prediction. This last paragraph will now focus on learning such models. For the sake of focus, the description restricts itself towards *Maximum Likelihood Estimation* (MLE) [42] via Algo. 2. The *Expectation Maximization* (EM) [31] comprises the sum-rule/ marginalization over the latent variable space. In its general form, this sum is of exponential magnitude. By taking full advantage of assumptions made by the FMM, e.g. Eq. (3.1) and its consequence Eq. (3.5) marked with (*), the EM algorithm reduces in complexity:

$$
\begin{aligned}
&Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) \\
&= \sum_{\boldsymbol{y} \in \mathcal{Y}} P(\boldsymbol{c} = \boldsymbol{y} | \underline{\boldsymbol{x}}, \boldsymbol{\theta}^{(t)}) \cdot \ln P(\underline{\boldsymbol{x}}, \boldsymbol{c} = \boldsymbol{y} | \boldsymbol{\theta}) \\
&=_{(*)} \sum_{y_1, \dots y_N \in \mathcal{Y}} \prod_{n=1}^{N} P(c_n = y_n | \boldsymbol{x}_n, \boldsymbol{\theta}^{(t)}) \cdot \sum_{n'=1}^{N} \ln P(\boldsymbol{x}_{n'}, c_{n'} = y_{n'} | \boldsymbol{\theta}) \\
&= \sum_{n'=1}^{N} \sum_{y_1 \in \mathcal{Y}} \dots \sum_{y_N \in \mathcal{Y}} \ln P(\boldsymbol{x}_{n'}, c_{n'} = y_{n'} | \boldsymbol{\theta}) \cdot \prod_{n=1}^{N} P(c_n = y_n | \boldsymbol{x}_n, \boldsymbol{\theta}^{(t)}) \\
&= \sum_{n'=1}^{N} \sum_{y_{n'} \in \mathcal{Y}} P(c_{n'} = y_{n'} | \boldsymbol{x}_{n'}, \boldsymbol{\theta}^{(t)}) \cdot \ln P(\boldsymbol{x}_{n'}, c_{n'} = y_{n'} | \boldsymbol{\theta}) \\
&\quad \cdot \sum_{y_1 \in \mathcal{Y}} P(c_1 = y_1 | \boldsymbol{x}_1, \boldsymbol{\theta}^{(t)}) \cdot \dots \sum_{y_{n'-1} \in \mathcal{Y}} P(c_{n'-1} = y_{n'-1} | \boldsymbol{x}_{n'-1}, \boldsymbol{\theta}^{(t)}) \\
&\quad \cdot \sum_{y_{n'+1} \in \mathcal{Y}} P(c_{n'+1} = y_{n'+1} | \boldsymbol{x}_{n'+1}, \boldsymbol{\theta}^{(t)}) \cdot \dots \sum_{y_N \in \mathcal{Y}} P(c_N = y_N | \boldsymbol{x}_N, \boldsymbol{\theta}^{(t)}) \\
&= \sum_{n'=1}^{N} \sum_{y_{n'} \in \mathcal{Y}} P(c_{n'} = y_{n'} | \boldsymbol{x}_{n'}, \boldsymbol{\theta}^{(t)}) \cdot \ln P(\boldsymbol{x}_{n'}, c_{n'} = y_{n'} | \boldsymbol{\theta})
\end{aligned}
$$

This specialized EM for FMMs is practical in its computational demand and therefore often used. EM updates are given exemplary for GMMs below. In comparison to *Gaussian* GCs, one can see a close similarity with the estimates of a GMM. The EM produces estimates as a weighted sum of cluster responsibilities. In case of Gaussian GCs, these weights were the indicator function encoding the class assignments with 100% responsibilities:

$$
\operatorname*{argmax}_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)})
$$

$$
\begin{aligned}
\rightarrow \hat{\pi}_c^{(t+1)} &= \frac{\sum_{n=1}^{N} P(c = c_n | \boldsymbol{x}_n, \boldsymbol{\theta}^{(t)})}{\sum_{n=1}^{N} \sum_{c' \in \mathcal{C}} P(c' = c_n | \boldsymbol{x}_n, \boldsymbol{\theta}^{(t)})} \\
\rightarrow \hat{\boldsymbol{\mu}}_c^{(t+1)} &= \frac{1}{\sum_{n=1}^{N} P(c = c_n | \boldsymbol{x}_n, \boldsymbol{\theta}^{(t)})} \cdot \sum_{n=1}^{N} P(c = c_n | \boldsymbol{x}_n, \boldsymbol{\theta}^{(t)}) \cdot \boldsymbol{x}_n \\
\rightarrow \hat{\boldsymbol{\Sigma}}_c^{(t+1)} &= \frac{1}{\sum_{n=1}^{N} P(c = c_n | \boldsymbol{x}_n, \boldsymbol{\theta}^{(t)})} \\
&\quad \cdot \sum_{n=1}^{N} P(c = c_n | \boldsymbol{x}_n, \boldsymbol{\theta}^{(t)}) \cdot (\boldsymbol{x}_n - \hat{\boldsymbol{\mu}}_c^{(t+1)})(\boldsymbol{x}_n - \hat{\boldsymbol{\mu}}_c^{(t+1)})^T
\end{aligned}
$$

3.3 Models for Structured Prediction

This section will introduce models for structured prediction capable to adjust for interconnected assignments. This is realized by the models via sequential adaptions of assumption Eq. (3.1) or Eq. (3.5). Depending on which assumption is exploited by the model, it is referred to as Sequential *Generative Classifiers* or Sequential *Discriminative Classifiers*.

3.3.1 Markov Models

Markov Models (MM) are probabilistic models for sequences of observations of arbitrary length [88]. These models can be used as the backbone for models for structured prediction. In its most basic form, MM work on discrete time steps that are analyzed via *transition functions*, which only considers dependencies to the previous time step. If this transition function does not consider any position information but solely the transitioning states, this model is called *homogenous*, *stationary* or *time-invariant* [88]. This is an example of parameter tying, since the same parameter is shared by multiple variables [88]. Fig. 3.9 illustrates the structure of an MM and the *(complete data) Likelihood* of this sequence model is defined as follows:

$$P(\boldsymbol{c}|\boldsymbol{y},\boldsymbol{\theta}) = \prod_{y\in\mathcal{Y}} P(c_1 = y|\boldsymbol{\theta})^{\mathcal{I}(y_1=y)} \cdot \prod_{y\in\mathcal{Y}}\prod_{n=2}^{N} P(c_n = y|y_{n-1},\boldsymbol{\theta})^{\mathcal{I}(y_n=y)}$$

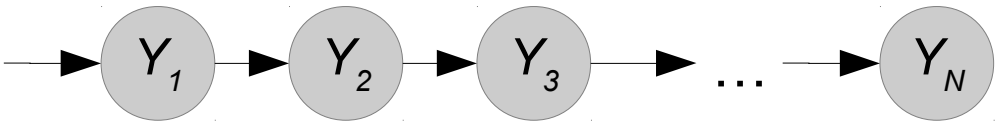


FIGURE 3.9: The Bayesian Network Graph for a Markov Model. Nodes reflect model variables and edges their interaction according: $A \rightarrow B = P(B|A)$.

The Likelihood decomposes into two parts: the *start probability* of the chain and the *transition probabilities* within the chain, defined by conditional distribution functions. This one-step-at-most kind of dependence is known as the *Markov property*, characterizing *Markov chains* [126]. For *discrete* & *finite states*, it is common to model these transitions via a *transition matrix* $\boldsymbol{A}$:

$$A_{yy'} = P(c_n = y|c_{n-1} = y', \boldsymbol{\theta})$$

Because of the probabilistic properties of conditional distribution functions, each row sums to one. Such matrices are formally called a *stochastic matrix*. By further exploiting this notation, a quite strong property of MM will be uncovered. The t-th step transition matrix of the chain will be defined as:

$$A_{yy'}(t) = P(c_{n+t} = y | c_n = y', \boldsymbol{\theta})$$

This function models the probability of getting from y' to y in t steps. The *Chapman-Kolmogorov equation* states that [88]:

$$A_{yy'}(s+t) = \sum_{z \in \mathcal{Y}} A_{yz}(t) \cdot A_{zy'}(s)$$

This realizes the same transit from y' to y within $s+t$ steps via *sum-rule / marginalization* over the unknown middle state z. This can be equivalent formulated via matrix multiplication, e.g. $\boldsymbol{A}(s+t) = \boldsymbol{A}(s) \cdot \boldsymbol{A}(t)$. A strong property about the behavior of the long term distribution of an MM arises by the following:

$$\begin{aligned} \pi_n(y) &= P(c_n = y | \boldsymbol{\theta}) \\ A_{yy'} &= P(c_n = y | c_{n-1} = y', \boldsymbol{\theta}) \\ \boldsymbol{\pi} &= \boldsymbol{\pi} \cdot \boldsymbol{A} \end{aligned}$$

For some MM, there exists a distribution $\boldsymbol{\pi}$ that remains unchanged during further transition via $\boldsymbol{A}$. This distribution is called the *stationary distribution*, *invariant distribution* or *equilibrium distribution* [88]. This property can be recognized as an instance of an *Eigenvector Equation* in respect to the transition matrix. The existence of a stationary distribution imposes a prelimiting constraint on $P(c_n = y | c_{n-1} = y', \boldsymbol{\theta})$ called *irreducibility* in the theory of Markov chains, which is that the *kernel* $P(c_n = y | c_{n-1} = y', \boldsymbol{\theta})$ allows for free moves all over the *state space* $\mathcal{Y}$, namely that, no matter the starting value y_0, the sequence $\{y_n\}$ has a positive probability of eventually reaching any region within the state space [107]. A sufficient condition for that property is any $P(c_n = y | c_{n-1} = y', \boldsymbol{\theta}) > 0$ [113]. Another major consequence resulting from the existence of a stationary distribution on the behavior of the chain $\{y_n\}$ called *recurrency*, is that any arbitrary non-negligible set is returned in an infinite number of times. In the case of recurrent chains, the stationary distribution is also a limiting distribution, in the sense that the limiting distribution of y_n is $\boldsymbol{\pi}$ for almost any initial value y_0 [107]. This property is called *ergodicity*. Variants of MM can easily be created by extending the model with additional dependencies in the conditional probability functions. The underlying Markov chain can be formulated in a more general form by *n-th order* Markov Models.

3.3.2 Dynamic Naive Bayes Models

Dynamic Naive Bayes Models (DNBM) [80] can be considered as extensions of *Generative Classifiers* (GC) that are sequentially connected via *Markov Models* to form dynamic predictions. The DNBM is illustrated in Fig. 3.10. Class assignments follow a homogenous discrete & finite state Markov chain, while GCs are connected as conditional probability functions (*emission probabilities*) for the features. The DNBM is restricted by definition to use Naive Bayes Models for the emissions. The *(complete data) Likelihood* of this sequence model is defined as follows:

$$
\begin{aligned}
P(\underline{\boldsymbol{x}}, \boldsymbol{c}|\boldsymbol{y}, \boldsymbol{\theta}) = & \prod_{y \in \mathcal{Y}} \Big(P(c_1 = y|\theta) \cdot P(\boldsymbol{x}_1|c_1 = y, \boldsymbol{\theta}) \Big)^{\mathcal{I}(y_1 = y)} \\
& \cdot \prod_{y \in \mathcal{Y}} \prod_{n=2}^{N} P(c_n = y|y_{n-1}, \boldsymbol{\theta})^{\mathcal{I}(y_n = y)} \\
& \cdot \prod_{y \in \mathcal{Y}} \prod_{n=2}^{N} P(\boldsymbol{x}_n|c_n = y, \boldsymbol{\theta})^{\mathcal{I}(y_n = y)} \\
= & \prod_{y \in \mathcal{Y}} \Big(P(c_1 = y|\theta) \cdot \prod_{a=1}^{A} P(x_{1a}|c_1 = y, \boldsymbol{\theta}) \Big)^{\mathcal{I}(y_1 = y)} \\
& \cdot \prod_{y \in \mathcal{Y}} \prod_{n=2}^{N} P(c_n = y|y_{n-1}, \boldsymbol{\theta})^{\mathcal{I}(y_n = y)} \\
& \cdot \prod_{y \in \mathcal{Y}} \prod_{n=2}^{N} \prod_{a=1}^{A} P(x_{na}|c_n = y, \boldsymbol{\theta})^{\mathcal{I}(y_n = y)}
\end{aligned}
$$

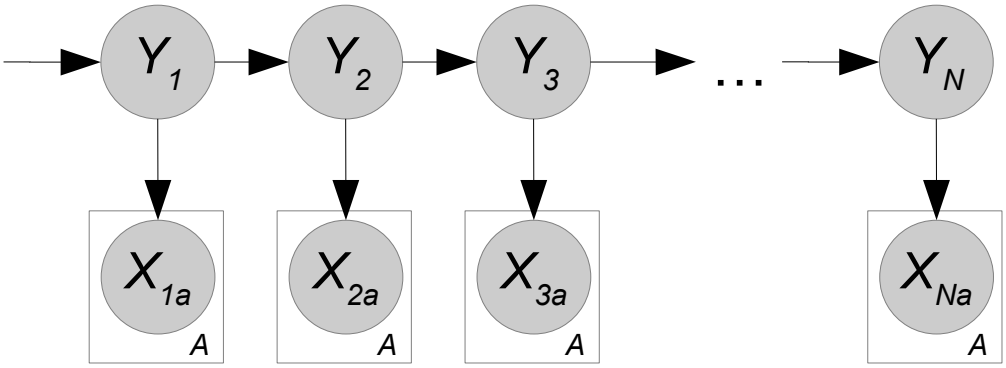


FIGURE 3.10: The Bayesian Network Graph for a Dynamic Naive Bayes Model. Nodes reflect model variables and edges their interaction: $A \rightarrow B = P(B|A)$. The variables of $Y_{\cdot}$ follow *Markov Models*, while variables of $X_{\cdot\cdot}$ are connected with them.

The Likelihood of the DNBM is less rigorously restricted as in the case of GCs via assumption Eq. (3.1). By constraining the sequence of classes to the *Markov property*, Eq. (3.3) modifies towards a Markov chain and features are conditioned on it according Eq. (3.4). Learning can be done via *Maximum Likelihood Estimation* [42] analogue to GCs. Predictions with this model are normally done via *Maximum A Posteriori Prediction* (MAP) but without the simplifying assumptions of GCs, the MAP does not

reduce towards a one-index-only prediction. The MAP in its general form is given as follows:

$$\begin{aligned}\hat{\boldsymbol{c}} &= \underset{\boldsymbol{y}\in\mathcal{Y}}{\operatorname{argmax}}\, P(\boldsymbol{c}=\boldsymbol{y}|\underline{\boldsymbol{x}},\boldsymbol{\theta}) \\ &= \underset{\boldsymbol{y}\in\mathcal{Y}}{\operatorname{argmax}}\, \frac{P(\underline{\boldsymbol{x}}|\boldsymbol{c}=\boldsymbol{y},\boldsymbol{\theta})\cdot P(\boldsymbol{c}=\boldsymbol{y}|\boldsymbol{\theta})}{P(\underline{\boldsymbol{x}}|\boldsymbol{\theta})} \\ &= \underset{\boldsymbol{y}\in\mathcal{Y}}{\operatorname{argmax}}\, \frac{P(\underline{\boldsymbol{x}},\boldsymbol{c}=\boldsymbol{y}|\boldsymbol{\theta})}{P(\underline{\boldsymbol{x}}|\boldsymbol{\theta})} \\ &= \underset{\boldsymbol{y}\in\mathcal{Y}}{\operatorname{argmax}}\, P(\underline{\boldsymbol{x}},\boldsymbol{c}=\boldsymbol{y}|\boldsymbol{\theta})\end{aligned}$$

In its general form, the MAP maximizes for predictions within a space of predictions of exponential magnitude. Equivalent to the situation in *Finite Mixture Models*, the general algorithm needs to be reduced in complexity by fully exploiting the factorization of the Likelihood or equivalent of the *Graphical Model* [88]. By using the factorization (marked with *), it is straight-forward to derive the MAP for the DNBM as a *Dynamic Programming* algorithm:

$$\begin{aligned}P_n(c_n=y) &= \max_{y_1,\dots y_{n-1}} P(\boldsymbol{x}_1,\dots\boldsymbol{x}_n,y_1,\dots y_{n-1},c_n=y|\boldsymbol{\theta}) \\ P_{n+1}(c_{n+1}=y) &= \max_{y_1,\dots y_n} P(\boldsymbol{x}_1,\dots\boldsymbol{x}_{n+1},y_1,\dots y_n,c_{n+1}=y|\boldsymbol{\theta}) \\ &= \max_{y_n}\max_{y_1,\dots y_{n-1}} P(\boldsymbol{x}_1,\dots\boldsymbol{x}_n,y_1,\dots y_n|\boldsymbol{\theta}) \\ &\qquad \cdot P(\boldsymbol{x}_{n+1},c_{n+1}=y|\boldsymbol{x}_1,\dots\boldsymbol{x}_n,y_1,\dots y_n,\boldsymbol{\theta}) \\ &= \max_{y_n} P_n(c_n=y_n) \\ &\qquad \cdot P(\boldsymbol{x}_{n+1},c_{n+1}=y|\boldsymbol{x}_1,\dots\boldsymbol{x}_n,y_1,\dots y_{n-1},c_n=y_n,\boldsymbol{\theta}) \\ &= \max_{y_n} P_n(c_n=y_n) \\ &\qquad \cdot P(\boldsymbol{x}_{n+1}|c_{n+1}=y,\boldsymbol{x}_1,\dots\boldsymbol{x}_n,y_1,\dots y_{n-1},c_n=y_n,\boldsymbol{\theta}) \\ &\qquad \cdot P(c_{n+1}=y|\boldsymbol{x}_1,\dots\boldsymbol{x}_n,y_1,\dots y_{n-1},c_n=y_n,\boldsymbol{\theta}) \\ &=_{(*)} \max_{y_n} P_n(c_n=y_n) \\ &\qquad \cdot P(\boldsymbol{x}_{n+1}|c_{n+1}=y,\boldsymbol{\theta}) \\ &\qquad \cdot P(c_{n+1}=y|c_n=y_n,\boldsymbol{\theta}) \\ &= P(\boldsymbol{x}_{n+1}|c_{n+1}=y,\boldsymbol{\theta}) \\ &\qquad \cdot \max_{y_n} P_n(c_n=y_n)\cdot P(c_{n+1}=y|c_n=y_n,\boldsymbol{\theta}) \\ P_1(c_1=y) &= P(\boldsymbol{x}_1,c_1=y|\boldsymbol{\theta}) \\ &= P(\boldsymbol{x}_1|c_1=y,\boldsymbol{\theta})\cdot P(c_1=y|\boldsymbol{\theta}) \\ P^* &= \max_y P_N(c_N=y)\end{aligned}$$

Within the community of these model types, the naming convention MAP is rather uncommon. Usually, the term *Max-Product Algorithm* [88] (MPA) is more often used, even more precise for this model: the *Viterbi Algorithm* [132][35], see Algo. 3.

Algorithm 3 Viterbi Algorithm (VA) [132]: This algorithm naturally arises as the Maximum A Posteriori Prediction of Dynamic Naive Bayes Models. The VA is a special instance of the more general Max-Product Algorithm.

procedure VITERBIALGO
  Initialisation:
    $P_1(c_1 = y) = P(\boldsymbol{x}_1|c_1 = y, \boldsymbol{\theta}) \cdot P(c_1 = y|\boldsymbol{\theta})$
  Recursion:
    $P_{n+1}(c_{n+1} = y) = \max_{y' \in \mathcal{Y}} P_n(c_n = y')$
      $\cdot P(\boldsymbol{x}_{n+1}|c_{n+1} = y, \boldsymbol{\theta}) \cdot P(c_{n+1} = y|c_n = y', \boldsymbol{\theta})$
    $tr_{n+1}(c_{n+1} = y) = \operatorname{argmax}_{y' \in \mathcal{Y}} P_n(c_n = y')$
      $\cdot P(c_{n+1} = y|c_n = y', \boldsymbol{\theta})$
  Termination:
    $\hat{c}_N = \operatorname{argmax}_{y \in \mathcal{Y}} P_N(c_n = y)$
  Traceback:
    $\hat{c}_{n-1} = tr_n(\hat{c}_n)$

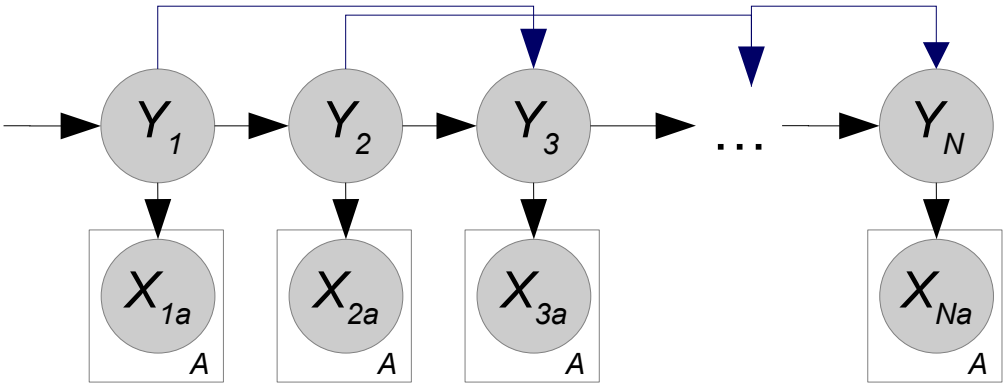


FIGURE 3.11: The Bayesian Network Graph for a Dynamic Naive Bayes Model variant. Nodes reflect model variables and edges their interaction: $A \rightarrow B = P(B|A)$. The variables of $Y_\cdot$ follow higher order *Markov Models*, while variables of $X_{\cdot\cdot}$ are connected with them.

Variants of DNBM can easily be created by extending the model with additional dependencies. Especially, the underlying Markov property can be formulated in a more general form by *n-th order* Markov Models. Fig. 3.11 shows the Graphical Model of a DNBM with a second order chain. The choices for the particular Markov Model will need adaptions in Algo. 3 for the particular dependencies the model represents.

3.3.3 Maximum Entropy Markov Models

Maximum Entropy Markov Models (MEMM) [81] can be considered as extensions of *Discriminative Classifiers* (DC) that are sequentially connected to form dynamic predictions. The MEMM is illustrated in Fig. 3.12. Models used in the MEMM are by definition either *Logistic Regression* (LR) [11] or *Multinomial Logit* [30] models. Equivalent to the LR model, predictions follow the linear combination of parameters and data. For the sequential extension, this linear combination is extended with the previous class assignment to form a chain within the network. The *(complete data) Likelihood* of this sequence model is defined as follows:

$$
\begin{aligned}
&P(\boldsymbol{c}|\underline{\boldsymbol{x}}, \boldsymbol{y}, \boldsymbol{\theta} = (\boldsymbol{\theta}_0, \boldsymbol{\theta}_1)) \\
&= \prod_{y \in \mathcal{Y}} P(c_1 = y|\boldsymbol{x}_1, \boldsymbol{\theta}_0)^{\mathcal{I}(y_1 = y)} \\
&\quad \cdot \prod_{y \in \mathcal{Y}} \prod_{n=2}^{N} P(c_n = y|\boldsymbol{x_n}, y_{n-1}, \boldsymbol{\theta}_1)^{\mathcal{I}(y_n = y)}
\end{aligned}
$$

$$
P(c_1 = y|\boldsymbol{x}_1, \boldsymbol{\theta}_0) = \begin{cases} \sigma(\boldsymbol{x}_1^T \boldsymbol{\theta}_0) & y = 1 \\ 1 - \sigma(\boldsymbol{x}_1^T \boldsymbol{\theta}_0) & y = 0 \end{cases}
$$

$$
P(c_n = y|\boldsymbol{x}_n, y_{n-1}, \boldsymbol{\theta}_1) = \begin{cases} \sigma(\boldsymbol{z}_n^T \boldsymbol{\theta}_1) & y = 1, \boldsymbol{z}_n = (\boldsymbol{x}_n, y_{n-1}) \\ 1 - \sigma(\boldsymbol{z}_n^T \boldsymbol{\theta}_1) & y = 0, \boldsymbol{z}_n = (\boldsymbol{x}_n, y_{n-1}) \end{cases}
$$

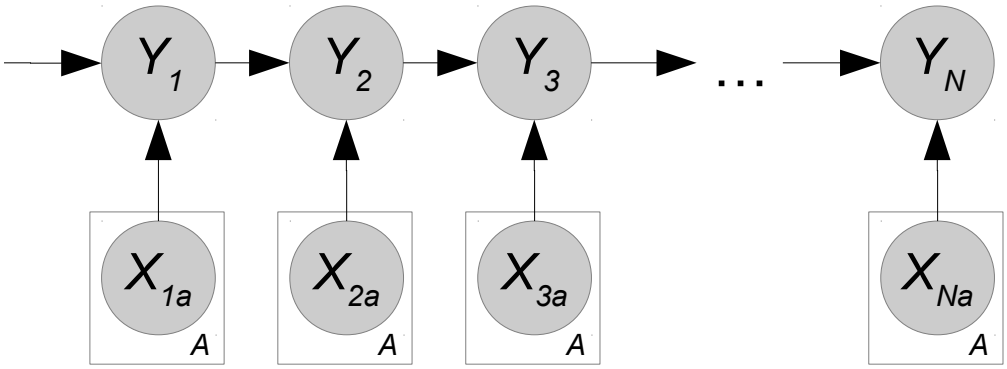


FIGURE 3.12: The Bayesian Network Graph for a Maximum Entropy Markov Model. Nodes reflect model variables and edges their interaction: $A \rightarrow B = P(B|A)$. The graph structure is structurally comparable to *Dynamic Naive Bayes Models* [80] but with partially inverted edge directions.

The Likelihood of a MEMM is less restricted as in the case of DCs via assumption Eq. (3.5) but directly extends it towards a sequential adaption. Equivalent to the relation of *Generative Classifiers* to DCs, a MEMM assume less restricting assumptions as *Dynamic Naive Bayes Models* (DNBM) [80]. Learning a MEMM via *Maximum Likelihood Estimation* [42] can be done by *Gradient Descent* Algo. 1. The MEMM can simply be learned as a DC by adding previous classes to an extended feature space. Equivalent to the *Maximum A Posteriori Prediction* (MAP) for DNBMs in Algo. 3, the general MAP needs to be reduced in complexity by fully exploiting the factorization

of the Likelihood or equivalent of the *Graphical Model* [88]. By using the factorization (marked with *), it is straight-forward to derive the MAP for the MEMM as a *Dynamic Programming* algorithm:

$$
\begin{aligned}
P_n(c_n = y) &= \max_{y_1,\dots y_{n-1}} P(y_1,\dots,y_{n-1}, c_n = y|\boldsymbol{x}_1,\dots\boldsymbol{x}_n,\boldsymbol{\theta}) \\
P_{n+1}(c_{n+1} = y) &= \max_{y_1,\dots y_n} P(y_1,\dots y_n, c_{n+1} = y|\boldsymbol{x}_1,\dots\boldsymbol{x}_{n+1},\boldsymbol{\theta}) \\
&= \max_{y_n} \max_{y_1,\dots y_{n-1}} P(y_1,\dots y_n|\boldsymbol{x}_1,\dots\boldsymbol{x}_n,\boldsymbol{\theta}) \\
&\qquad \cdot P(c_{n+1} = y|\boldsymbol{x}_1,\dots\boldsymbol{x}_n,\boldsymbol{x}_{n+1},y_1,\dots y_n,\boldsymbol{\theta}) \\
&= \max_{y_n} P_n(c_n = y_n) \\
&\qquad \cdot P(c_{n+1} = y|\boldsymbol{x}_1,\dots\boldsymbol{x}_n,\boldsymbol{x}_{n+1},y_1,\dots y_{n-1},c_n = y_n,\boldsymbol{\theta}) \\
&=_{(*)} \max_{y_n} P_n(c_n = y_n)\cdot P(c_{n+1} = y|\boldsymbol{x}_{n+1},c_n = y_n,\boldsymbol{\theta}) \\
P_1(c_1 = y) &= P(c_1 = y|\boldsymbol{x}_1,\boldsymbol{\theta}) \\
P^* &= \max_y P_N(c_N = y)
\end{aligned}
$$

Algorithm 4 Max-Product Algorithm: This algorithm arises naturally as the Maximum A Posteriori Prediction for Maximum Entropy Markov Models. This algorithm is a special instance of the general Max-Product Algorithm.

procedure MAXPRODUCTMEMM
  Initialisation:
    $P_1(c_1 = y) = P(c_1 = y|\boldsymbol{x}_1,\boldsymbol{\theta})$
  Recursion:
    $P_{n+1}(c_{n+1} = y) = \max_{y'\in\mathcal{Y}} P_n(c_n = y')$
      $\cdot P(c_{n+1} = y|\boldsymbol{x}_n, c_n = y',\boldsymbol{\theta})$
    $tr_{n+1}(c_{n+1} = y) = \operatorname{argmax}_{y'\in\mathcal{Y}} P_n(c_n = y')$
      $\cdot P(c_{n+1} = y|\boldsymbol{x}_n, c_n = y',\boldsymbol{\theta})$
  Termination:
    $\hat{c}_N = \operatorname{argmax}_{y\in\mathcal{Y}} P_N(c_N = y)$
  Traceback:
    $\hat{c}_{n-1} = tr_n(\hat{c}_n)$

3.3.4 Hidden Markov Models

Hidden Markov Models (HMM) [99] can be considered as extensions of *Finite Mixture Models* (FMM), that are sequentially connected via *Markov Models* to form dynamic predictions. The HMM is illustrated in Fig. 3.13. Latent class assignments follow a homogenous discrete & finite state Markov chain, while FMMs are connected as conditional probability functions (*emission probabilities*) for the features. The *(incomplete data) Likelihood* of this sequence model is defined as follows:

$$
\begin{aligned}
P(\underline{\boldsymbol{x}}|\boldsymbol{\theta}) &= \sum_{\boldsymbol{y}\in\mathcal{Y}} P(\underline{\boldsymbol{x}}, \boldsymbol{y}|\boldsymbol{\theta}) \\
&= \sum_{y_1,\dots y_N\in\mathcal{Y}} P(c_1 = y_1|\boldsymbol{\theta}) \cdot P(\boldsymbol{x}_1|c_1 = y_1, \boldsymbol{\theta}) \\
&\quad \cdot \prod_{n=2}^{N} P(c_n = y_n|y_{n-1}, \boldsymbol{\theta}) \cdot P(\boldsymbol{x}_n|c_n = y_n, \boldsymbol{\theta}) \\
&= \sum_{y_1,\dots y_N\in\mathcal{Y}} P(c_1 = y_1|\boldsymbol{\theta}) \cdot \prod_{a=1}^{A} P(x_{1a}|c_1 = y_1\boldsymbol{\theta}) \\
&\quad \cdot \prod_{n=2}^{N} P(c_n = y_n|y_{n-1}, \boldsymbol{\theta}) \cdot \prod_{a=1}^{A} P(x_{na}|c_n = y_n, \boldsymbol{\theta})
\end{aligned}
$$

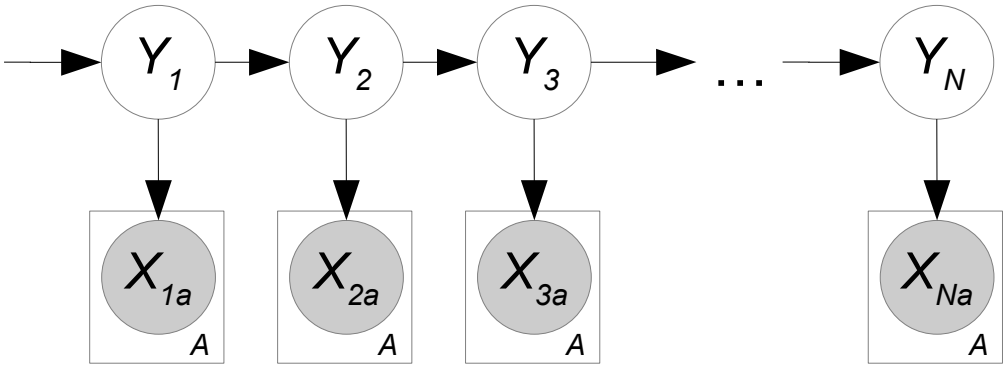


FIGURE 3.13: The Bayesian Network Graph for a Hidden Markov Model [99]. Nodes reflect model variables and edges their interaction: $A \rightarrow B = P(B|A)$. The graph structure is structurally comparable to *Dynamic Naive Bayes Models* [80] but with latent (unknown) variables marked in white.

The Likelihood of the HMM is less rigorously restricted as in the case of FMMs via assumption Eq. (3.1). By constraining the sequence of latent assignments to the *Markov property*, Eq. (3.3) modifies towards a Markov chain and features are conditioned on it according Eq. (3.4). Unfortunately, the computational overhead for calculating the Likelihood increases exponentially with the length of the sequence [35] because of the *sum-rule/marginalization*. Equivalent to the *Maximum A Posteriori Prediction* (MAP) for *Dynamic Naive Bayes Models* (DNBM) [80] and *Maximum Entropy Markov Models* [81], an efficient algorithm for calculating the Likelihood itself is needed. A reduction in complexity can be achieved by fully exploiting the factorization of the Likelihood or equivalent of the *Graphical Model* [88]. In that case, two *Dynamic Programming* algorithms can be derived for that case: the *Forward Algorithm*

(FA) and the *Backward Algorithm* (BA), see below. Both algorithms are an instance of the *Sum-Product Algorithm* (SPA). The application of the factorization assumptions will be marked with (*):

$$
\begin{aligned}
f_n(c_n = y) =& P(\boldsymbol{x}_1, ...\boldsymbol{x}_n, c_n = y|\boldsymbol{\theta}) \\
=& \sum_{y_{n-1}\in\mathcal{Y}} P(\boldsymbol{x}_1, ...\boldsymbol{x}_n, y_{n-1}, c_n = y|\boldsymbol{\theta}) \\
=& \sum_{y_{n-1}\in\mathcal{Y}} P(\boldsymbol{x}_1, ...\boldsymbol{x}_{n-1}, y_{n-1}|\boldsymbol{\theta}) \\
& \quad \cdot P(\boldsymbol{x}_n, c_n = y|\boldsymbol{x}_1, ...\boldsymbol{x}_{n-1}, y_{n-1}, \boldsymbol{\theta}) \\
=& \sum_{y_{n-1}\in\mathcal{Y}} f_{l-1}(c_{n-1} = y_{n-1}) \\
& \quad \cdot P(\boldsymbol{x}_n, c_n = y|\boldsymbol{x}_1, ...\boldsymbol{x}_{n-1}, c_{n-1} = y_{n-1}, \boldsymbol{\theta}) \\
=& \sum_{y_{n-1}\in\mathcal{Y}} f_{l-1}(c_{n-1} = y_{n-1}) \\
& \quad \cdot P(\boldsymbol{x}_n|\boldsymbol{x}_1, ...\boldsymbol{x}_{n-1}, c_{n-1} = y_{n-1}, c_n = y_n, \boldsymbol{\theta}) \\
& \quad \cdot P(c_n = y|\boldsymbol{x}_1, ...\boldsymbol{x}_{n-1}, c_{n-1} = y_{n-1}, \boldsymbol{\theta}) \\
=_{(*)} & \sum_{y_{n-1}\in\mathcal{Y}} f_{l-1}(c_{n-1}) \\
& \quad \cdot P(\boldsymbol{x}_n|c_n = y, \boldsymbol{\theta}) \\
& \quad \cdot P(c_n = y|c_{n-1} = y_{n-1}, \boldsymbol{\theta})
\end{aligned}
$$

$$
\begin{aligned}
b_n(c_n = y) =& P(\boldsymbol{x}_{n+1}, ...\boldsymbol{x}_N|c_n = y, \boldsymbol{\theta}) \\
=& \sum_{y_{n+1}\in\mathcal{Y}} P(\boldsymbol{x}_{n+1}, ...\boldsymbol{x}_N, y_{n+1}|c_n = y, \boldsymbol{\theta}) \\
=& \sum_{y_{n+1}\in\mathcal{Y}} P(\boldsymbol{x}_{n+2}, ...\boldsymbol{x}_N|\boldsymbol{x}_{n+1}, y_{n+1}, c_n = y, \boldsymbol{\theta}) \\
& \quad \cdot P(\boldsymbol{x}_{n+1}, y_{n+1}|c_n = y, \boldsymbol{\theta}) \\
=& \sum_{y_{n+1}\in\mathcal{Y}} P(\boldsymbol{x}_{n+2}, ...\boldsymbol{x}_N|\boldsymbol{x}_{n+1}, y_{n+1}, c_n = y, \boldsymbol{\theta}) \\
& \quad \cdot P(\boldsymbol{x}_{n+1}|c_n = y, y_{n+1}, \boldsymbol{\theta}) \\
& \quad \cdot P(y_{n+1}|c_n = y, \boldsymbol{\theta}) \\
=_{(*)} & \sum_{y_{n+1}\in\mathcal{Y}} b_{n+1}(c_{n+1} = y_{n+1}) \\
& \quad \cdot P(\boldsymbol{x}_{n+1}|c_{n+1} = y_{n+1}, \boldsymbol{\theta}) \\
& \quad \cdot P(c_{n+1} = y_{n+1}|c_n = y, \boldsymbol{\theta})
\end{aligned}
$$

Being structurally identical to DNBMs, the MAP of HMMs is the same as in Algo. 3. Further, being structurally the extension of sequential FMMs, a HMM can simply be learned as an FMM with additional conditions. The *Maximum Likelihood Estimation* [42] of HMMs will result in an *Expectation Maximization* (EM) [31] algorithm, see Algo. 2. In case of this particular model, the learning algorithm is formally known as the *Baum-Welch Algorithm* (BWA). The BWA is the realization of an EM that fully exploited the factorization of HMMs, which will be marked with (*):

$$
\begin{aligned}
& P(c_n = y | \boldsymbol{x}_1, ...\boldsymbol{x}_N, \boldsymbol{\theta}) \\
&= \frac{P(\boldsymbol{x}_1, ...\boldsymbol{x}_N | c_n = y, \boldsymbol{\theta}) \cdot P(c_n = y | \boldsymbol{\theta})}{P(\boldsymbol{x}_1, ...\boldsymbol{x}_N | \boldsymbol{\theta})} \\
&= \frac{P(\boldsymbol{x}_1, ...\boldsymbol{x}_N, c_n = y | \boldsymbol{\theta})}{P(\boldsymbol{x}_1, ...\boldsymbol{x}_N | \boldsymbol{\theta})} \\
&=_{(*)} \frac{P(\boldsymbol{x}_1, ...\boldsymbol{x}_n, c_n = y | \boldsymbol{\theta}) \cdot P(\boldsymbol{x}_{n+1}, ...\boldsymbol{x}_N | c_n = y, \boldsymbol{\theta})}{P(\boldsymbol{x}_1, ...\boldsymbol{x}_N | \boldsymbol{\theta})} \\
&= \frac{f_n(c_n = y) \cdot b_n(c_n = y)}{P(\boldsymbol{x}_1, ...\boldsymbol{x}_N | \boldsymbol{\theta})} \\
&= \frac{f_n(c_n = y) \cdot b_n(c_n = y)}{\sum_{y' \in \mathcal{Y}} P(\boldsymbol{x}_1, ...\boldsymbol{x}_N, c_n = y' | \boldsymbol{\theta})} \\
&= \frac{f_n(c_n = y) \cdot b_n(c_n = y)}{\sum_{y' \in \mathcal{Y}} f_n(c_n = y') \cdot b_n(c_n = y')}
\end{aligned}
$$

Variants of a HMM can easily be created by extending the model with additional dependencies in the form of conditional probability functions. The underlying Markov chain can be formulated in a more general form by *n-th order* Markov Models. Additional dependencies can be incorporated in the feature level. Such variant will form so-called *Autoregressive Hidden Markov Models* [40] as it can be seen in Fig. 3.14. Algo. 3 needs adaption for these higher dependencies, but it is straightforward to derive the MPA & SPA for structural extensions in regular form.

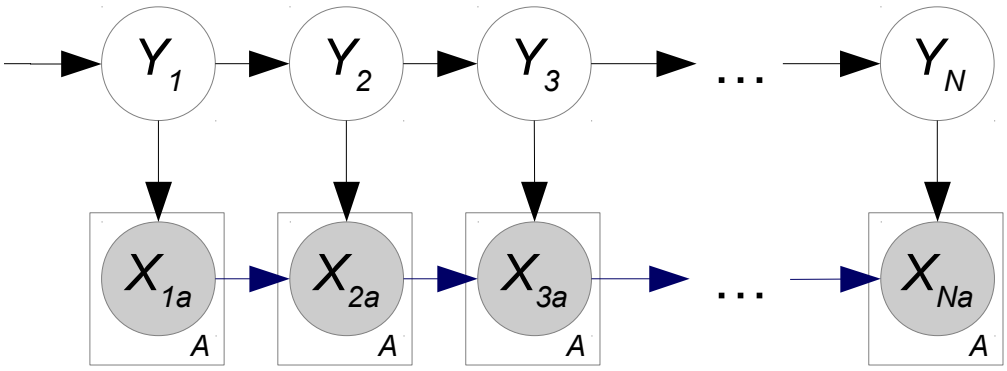


FIGURE 3.14: The Bayesian Network Graph for a Hidden Markov Model variant. Nodes reflect model variables and edges their interaction: $A \rightarrow B = P(B|A)$. Latent (unknown) variables are marked white. Additional dependencies are incorporated via the autoregressive variables in $X_{...}$. These model extensions are known as Autoregressive Hidden Markov Model [40].

3.3.5 Bayesian Networks

Bayesian Networks (BN) are a general family of models that are defined over graphs. Graphs comprises *nodes* (vertices) connected via *edges* (links, arcs) [14]. If links are interpreted symmetric, these graphs are called *undirected*. If all links follow a direction, the graph is called *directed*. Directed graphs that comprise no loops are called *directed acyclic graphs* (DAG). For the introduction of BNs, the following definitions (adapted from [88]) are necessary:

- **graph**: $G = (\mathcal{V}, \mathcal{E})$
- **nodes**: $\mathcal{V} = \{1, ...V\}$
- **edges**: $\mathcal{E} = \{(s,t) : s,t \in \mathcal{V}\}$
- **undirected:** $\forall (s,t) \in \mathcal{E} \rightarrow \exists (t,s) \in \mathcal{E}$
- **directed:** $\forall (s,t) \in \mathcal{E} \rightarrow \nexists (t,s) \in \mathcal{E}$
- **parent:** $pa(s) = \{t \in \mathcal{V} : (t,s) \in \mathcal{E}\}$

In *Probabilistic Graphical Models*, each node represents a *Random Variable* (or group of random variables), and the links express probabilistic relationships between these variables [14]. *Directed Graphical Models* (DGM) are DAGs, and they are commonly known as *Bayesian Networks*, *Belief Network* or *Causal Network* [88]. Links in DGMs represent conditional probability distributions. Therefore, the graph structure in its entirety represents a joint probability distribution that is factorized by conditional independence assumptions realized by the graph edges. Or more precise: the joint distribution defined by a graph is given by the product, over all the nodes of the graph, of a conditional distribution for each node conditioned on the variables corresponding to the parents of that node in the graph [14]. With that definition given, all previously introduced models are instances of BNs. The Likelihood of a BN is defined as follows:

$$\begin{aligned} P(Z = (\underline{\boldsymbol{x}}, \boldsymbol{y}, \boldsymbol{c})|\boldsymbol{\theta}) &= P(Z|G_{\boldsymbol{\theta}}) \\ &= \prod_{z \in \mathcal{Z}} P(Z_z|pa(Z_z), \boldsymbol{\theta}) \end{aligned}$$

It is straight-forward to recognize the connection to all the previously introduced models. Fig. 3.15 illustrates this connection with an overview of several *Graphical Models* [88]. Learning methods for specific graph structures have already been introduced. In case of *Supervised Learning*, the data tuples $(\boldsymbol{x}, y)$ are known and can be learned by *Maximum Likelihood Estimation* (MLE) [42] (in closed form, Algo. 1 or Algo. 2). In case of *Unsupervised Learning*, the data tuples $(\boldsymbol{x}, y)$ have missing information about the latent variable y and can be learned by MLE via *Expectation Maximization* [31] (Algo. 2). All inference methods have already been described as special instances of the *Max-Product Algorithm* for the *Maximum A Posteriori Prediction* or the *Sum-Product Algorithm* for the case of *sum-rule/marginalization* over latent variables. Both algorithms are *Variable Elimination Algorithms* (VEA) [88] and can be seen in their general form in Algo. 5 & Algo. 6. In general, VEA can be applied to any *commutative semi-ring*, e.g. *sum-product*, *max-product*, *min-sum* and *boolean satisfiability* [88]. This section has now generalized all previously introduced models towards a common framework. This framework comprises all aspects of supervised

& unsupervised learning, e.g. classification & clustering. Further, arbitrary BNs can be constructed, learned and inferred with. Therefore, this section ends in a description of a model family comprising countably infinite instances. Some of them have known names, most of them do not. During the end of the thesis, a particular graph structure will receive a name.

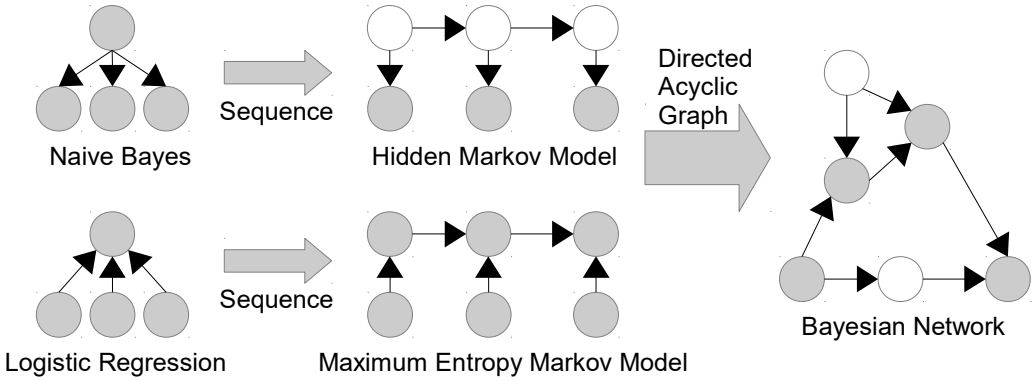


FIGURE 3.15: Overview of Bayesian Network Graphs for previously described models. *Generative Classifiers* can be extended for sequences, e.g. Naive Bayes Models towards *Hidden Markov Models* [99]. *Discriminative Classifiers* can be extended as well, e.g. Logistic Regression [11] towards *Maximum Entropy Markov Models* [81]. The broadest class of such extensions (considered in this thesis) form Directed Acyclic Graphs and are called Bayesian Networks. Nodes reflect model variables and edges their interaction: $A \rightarrow B = P(B|A)$. Latent (unknown) variables are marked white.

Algorithm 5 Sum-Product algorithm for variable elimination. Taken from [66].

procedure SUMPRODUCTALGORITHM
  Procedure SumProduct(Φ, $\boldsymbol{Z}$, $\prec$)
    // Φ: set of factors, $\prec$: ordering on $\boldsymbol{Z}$
    // $\boldsymbol{Z}$: set of variables to be eliminated

    Let $Z_1, ...Z_k$ be the ordering $Z_i \prec Z_j$ for $i < j$
    for i = 1, ... k
      $\Phi \leftarrow SumProductEliminateVar(\Phi, Z_i)$
    $\phi^* \leftarrow \prod_{\phi \in \Phi} \phi$
    return ϕ^*

  Procedure SumProductEliminateVar(Φ, Z)
    $\Phi' \leftarrow \{\phi \in \Phi : Z \in Scope[\phi]\}$
    $\Phi'' \leftarrow \Phi - \Phi'$
    $\psi \leftarrow \prod_{\phi \in \Phi'} \phi$
    $\tau \leftarrow \sum_Z \psi$
    return $\Phi'' \cup \{\tau\}$

Algorithm 6 Max-Product algorithm for variable elimination. Taken from [66].

procedure MAXPRODUCTALGORITHM
  Procedure MaxProduct(Φ, $\prec$)
    // Φ: set of factors over $\boldsymbol{X}$, $\prec$: ordering on $\boldsymbol{X}$

    Let $X_1, ...X_k$ be the ordering $X_i \prec X_j$ for $i < j$
    for i = 1, ... k
      $(\Phi, \phi_{X_i}) \leftarrow MaxProductEliminateVar(\Phi, X_i)$
    $\boldsymbol{x}^* \leftarrow TracebackMAP(\{\phi_{X_i} : i = 1, ...k\})$
    return $(\boldsymbol{x}^*, \Phi)$ // $\boldsymbol{x}^*$ = MAP, Φ = Probability of MAP

  Procedure MaxProductEliminateVar(Φ, X)
    $\Phi' \leftarrow \{\phi \in \Phi : X \in Scope[\phi]\}$
    $\Phi'' \leftarrow \Phi - \Phi'$
    $\psi \leftarrow \prod_{\phi \in \Phi'} \phi$
    $\tau \leftarrow max_X \psi$
    return $(\Phi'' \cup \{\tau\}, \psi)$

  Procedure TracebackMAP($\{\phi_{X_i} : i = 1, ...k\}$)
    for i = k, ... 1
      $\boldsymbol{u}_i \leftarrow (x^*_{i+1}, ...x^*_k) < Scope[\phi_{X_i}] - \{X_i\} >$
      $x^*_i \leftarrow \text{argmax}_{x_i} \phi_{X_i}(x_i, \boldsymbol{u}_i)$
    return $\boldsymbol{x}^*$

3.4 Summary

All in all, there is a rich set of models in Machine Learning that are well-founded in Statistics. Previous mentioned models were introduced either in *Supervised Learning* and *Unsupervised Learning* scenarios. Nonetheless, such a separation is rather superficial because it rather resembles properties of the data itself and not the models. During this section, models were introduced in families based on shared assumptions about the data and assignment tuples. The following will shortly summarize all important characteristics of the before mentioned model families.

Section *Modeling* started with *Models for Unstructured Prediction*. This family of models can be described by its shared assumptions about the missing interconnection of predictions, see *Fundamental Statistics*. Models for unstructured prediction can further be partitioned into *Generative Classifiers* and *Discriminative Classifiers*. Generative Classifiers exploit Eq. (3.1) which rigorously restricts data and assignment tuples. This subfamily comprises classical models such as *Gaussian Naive Bayes Model*, *Multivariate Bernoulli Naive Bayes Model*, *Multinomial Naive Bayes Model*, *Linear Discriminant Analysis* [43], *Quadratic Discriminant Analysis* [44] and many more. Discriminative Classifiers exploit the less restricting assumption Eq. (3.5) about data and assignment tuples. This subfamily comprises classical models such as *Logistic Regression* [11], *Probit Regression* [41][15], *Complementary-Log-log Regression* [83] and can generally be expressed via *Generalized Linear Models* [83]. The entirety of models for unstructured prediction can be described as *Bayesian Networks* without interconnected predictions on data points. All these simplistic networks can be learned via *Maximum Likelihood Estimation* either in closed-form, *Gradient Descent* (Algo. 1) or *Expectation Maximization* [31] (Algo. 2). Inference can be done consistently via *Variable Elimination Algorithms*, e.g. the *Sum-Product Algorithm* (Algo. 5) or the *Max-Product Algorithm* (Algo. 6).

Section *Models for Structured Prediction* extended the rather simplistic models of the first section. The tuples of data and assignment are less rigorously restricted but follow a chain-like dependency resulting in interconnection of predictions. In case of *Markov Models*, this results in Generative Sequence Models comprising *Dynamic Naive Bayes Models* [80] and *Hidden Markov Models* [99]. Their counterpart can be defined as Discriminative Sequence Models, e.g. in the form of *Maximum Entropy Markov Models* [81]. Both subfamilies achieve their increased complexity by sequential adaptions of Eq. (3.1) and Eq. (3.5). The entirety of the presented models can be described as Bayesian Networks with interconnected predictions via chain structures. Nonetheless, the chain's interpretation of modeling 'time' in the sequence is arbitrary. In case of text analysis, such chain structures are interpreted as 'grammar' [78]. In case of phylogenetic analysis, such chain structures are interpreted as 'ancestor relationships' [49][35]. All mentioned model instances can be learned via Maximum Likelihood Estimation either in closed-form, Gradient Descent (Algo. 1) or Expectation Maximization (Algo. 2). Inference can be done consistently via Variable Elimination Algorithms, e.g. the Sum-Product Algorithm (Algo. 5) or the Max-Product Algorithm (Algo. 6).

In general, arbitrary complex dependency structures can be modeled. If such dependencies can be represented via a *Directed Acyclic Graph*, the resulting model suffices the definition of a Bayesian Network. Such networks can be used for models for structured and unstructured prediction. Further on, any complex attribute interaction can be modeled explicitly via the network structure. In case of multimodal data, the fusion of several sub-models into a broader one can be realized via a Bayesian Network, as long as the fusion network satisfies the properties of being directed and acyclic. Network learning can be done via Maximum Likelihood Estimation either in closed-form, Gradient Descent (Algo. 1) or Expectation Maximization (Algo. 2). Inference within Bayesian Networks can be done consistently via Variable Elimination Algorithms, e.g. the Sum-Product Algorithm (Algo. 5) or the Max-Product Algorithm (Algo. 6). Fig. 3.16 illustrates the interconnection of all the described models.

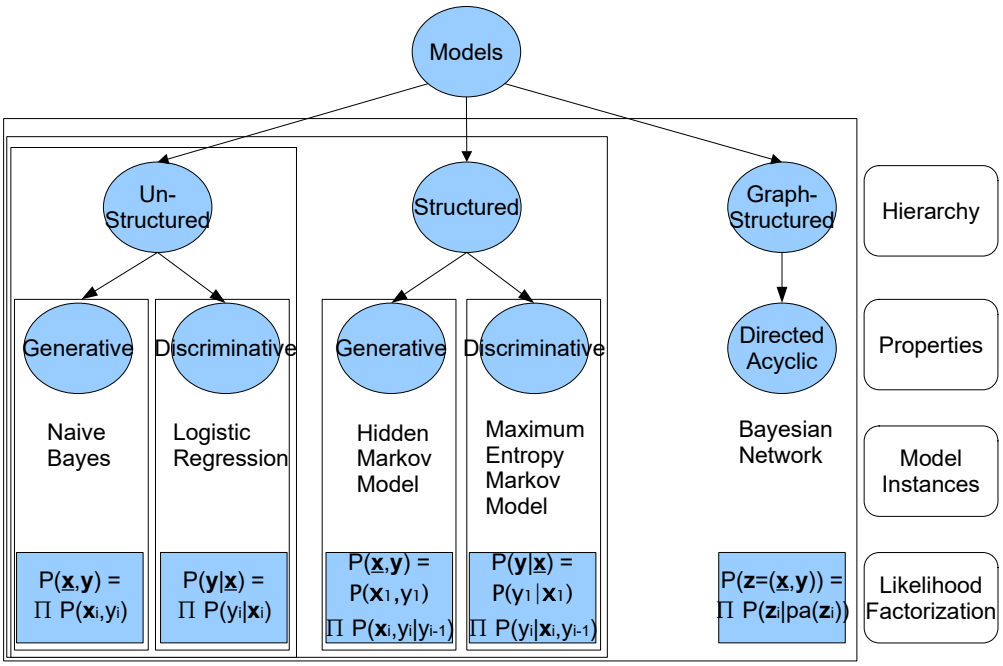


FIGURE 3.16: Hierarchy of Models. The section started the description with *Models for Unstructured Prediction* (left), extended for *Models for Structured Prediction* (middle) and generalized towards *Bayesian Networks* (right). The increasing complexity could be achieved by reduction of restricting independence assumptions.

3.5 Information Retrieval

Information Retrieval (IR) is a nice field of application for *Bayesian Networks*. In its broadest scope, IR deals with the problem of representation, storage, organization and access of information items [8]. This section will focus on the aspect of accessing information items via *ranking*. Ranking describes a *retrieval process* to order information items, e.g. documents, in respect to the users' *Information Need*, e.g. a query. The following will exclusively described information items of textual form. According Baeza-Yates & Ribeiro-Neto [8] an IR model is a quadrupel $(\boldsymbol{D}, \boldsymbol{Q}, \mathcal{F}, R(\boldsymbol{q}, \boldsymbol{d}))$ of the following form:

- $\boldsymbol{D}$: Set of documents in a 'suitable' representation.
- $\boldsymbol{Q}$: Set of queries in a 'suitable' representation (the realization of the Information Need of users).
- $R(\boldsymbol{q}, \boldsymbol{d})$: *Ranking function* to assign a real number measuring the relationship between $\boldsymbol{q} \in \boldsymbol{Q}$ and $\boldsymbol{d} \in \boldsymbol{D}$. Such rankings are used to define the ordering for all documents in respect to a given query.
- $\mathcal{F}$: Framework for modeling the 'suitable' representation of $\boldsymbol{D}$, $\boldsymbol{Q}$ and their relationship via $R(\boldsymbol{q}, \boldsymbol{d})$.

The concept 'suitable' representation was not provided in the original work [8] but incorporated here for simplicity. In general, a plethora of 'suitable' representations can be constructed, but this is not considered to be the scope of this research work here. Without (much) loss of generality, the following description will focus on the most prominent representation, e.g. *Bag-Of-Words* (BoW). Each document can be considered in a simplified form as a sequence of words. Words are elements within a finite set from a language (or an indexed sub set of this language). The BoW reduces the sequence of words within documents into a position invariant vector representation for each document. Each dimension in the vector represents the document-specific frequency of the indexed word, or sometimes a document-specific weight for it. Based on this description, the 'suitable' BoW representation can be defined in accordance to Baeza-Yates & Ribeiro-Neto [8] just as follows:

- $U = \{k_1, ...k_T\}$: The 'universe of discourse' as a set of keywords (indexed by the IR system as the *sample space*)
- $\boldsymbol{d} = (w_{d1}, ...w_{dT})$: The vector representation of a document where each dimension represents the keyword frequency in its index position or a document and keyword specific weight. Index terms not present in the document follow $w_{dt} = 0$ and $w_{dt} > 0$ otherwise.
- $g_i(\boldsymbol{d}) = w_{di}$: The index function that maps a keyword index t in $\boldsymbol{d}$ with the document keyword specific weight.
- $\boldsymbol{q} = (w_{q1}, ...w_{qT})$: The vector representation of a query where each dimension represents the keyword frequency in its index position or a query and keyword specific weight. Index terms not present in the query follow $w_{qt} = 0$ and $w_{qt} > 0$ otherwise.
- $g_t(\boldsymbol{q}) = w_{qt}$: The index function that maps a keyword index t in $\boldsymbol{q}$ with the query keyword specific weight.
- $u \subseteq U$: A subset of keywords interpreted as an abstract *concept*.

3.5.1 Boolean Model

The *Boolean Model* is a very simple model, easy to understand, but unfortunately with effectiveness problems [84]. It is founded on *Set Theory* and *Boolean Algebra*. In this model, a query consists of keywords linked by three connectives: *and*, *or* and *not* [129]. Each keyword index i in a query $\boldsymbol{q}$ will be represented as a logical literal $g_i(\boldsymbol{q})$ indicating the presence of a keyword at index i in $\boldsymbol{q}$ via the value true or false otherwise. By applying the Boolean transformation rules, e.g. *De Morgan's law*, each query can be formulated in their *disjunct normal form* q_{dnf} comprising only *conjunctive components* q_{ck} [106]. Each document $\boldsymbol{d}$ will be represented in the same literal form for all possible terms j via $g_j(\boldsymbol{d})$ as well. Retrieved documents comprise the same logical expression as the query and are assumed to be relevant, while documents contradicting the query logic can be considered as irrelevant. Analog to Ribeiro-Neto et al. [106], this similarity is defined as follows:

$$sim(\boldsymbol{d}, \boldsymbol{q}) = \begin{cases} 1 & \text{if} \quad \exists\, q_{ck} \mid q_{ck} \in q_{dnf} \wedge \forall_i\, g_i(\boldsymbol{d}) = g_i(q_{ck}) \\ 0 & \text{otherwise} \end{cases} \tag{3.6}$$

With the restriction to (Boolean) logic, the implied relevance concept, in its most simplistic form, decomposed in similarity measurements of $\{0, 1\}$. According to Micarelli et al. [84] the model appears to retrieve either too much or too little in practice. One might argue that this observation is the direct consequence of the strict dichotomous treatment of the similarity concept.

3.5.2 Vector Space Model

The *Vector Space Model* [110] works on vector spaces (see *Vector Space & Matrix Algebra*) and represents queries and documents in a high dimensional vector, while each dimension reflects a distinct keyword. These dimensions are assumed to be pairwise orthogonal, which implies that keywords are assumed to occur independently within documents and independently within queries [106]. This model does not represent keywords in a binary manner, such as the *Boolean Model*, but allows for positive weights such as keyword frequencies. A more common approach for these weights is a balance of the intra and inter-document importance of keywords [106], e.g. in the form of balancing the *term frequency* (tf) and the *inverse document frequency* (idf) according to the popular *tf-idf* weighting scheme. Based on the algebraic fundamentals of this model, similarities are simple angles between document vectors & the query vector. Analog to Micarelli et al. [84] and Ribeiro-Neto et al. [106], this similarity is defined as follows:

$$sim(\boldsymbol{d}, \boldsymbol{q}) = \cos(\boldsymbol{d}, \boldsymbol{q}) = \frac{\boldsymbol{d} \cdot \boldsymbol{q}}{|\boldsymbol{d}| \cdot |\boldsymbol{q}|} \tag{3.7}$$

This specification of similarity is quite intuitive because its values fall into [0,1] making any interpretation comparable to probabilistic distributions. Two main advantages emerge from the formulation of the Vector Space Model in comparison to the Boolean Model. First, input values do not need to be binary but can be positive real values. Second, the similarity measurement is also a real value between the binary extreme points. A drawback of the model arises from the fact that the logical connectives of the Boolean Model cannot easily be incorporated.

3.5.3 Probabilistic Model

The *Probabilistic Model* [108] relies on the idea that query and documents are generated probabilistically by an underlying relevance process. The entire ranking is calculated by the Likelihood ratio of the *relevant* (R) and *non-relevant* ($\overline{R}$) hypothesis. Analog to Micarelli et al. [84] and Ribeiro-Neto et al. [106], this similarity is defined as follows:

$$\begin{aligned} sim(\boldsymbol{d}, \boldsymbol{q}) &= \frac{P(R_q|\boldsymbol{d})}{P(\overline{R}_q|\boldsymbol{d})} \\ &= \frac{P(\boldsymbol{d}|R_q) \cdot P(R_q)/P(\boldsymbol{d})}{P(\boldsymbol{d}|\overline{R}_q) \cdot P(\overline{R}_q)/P(\boldsymbol{d})} \\ &= \frac{P(\boldsymbol{d}|R_q) \cdot P(R_q)}{P(\boldsymbol{d}|\overline{R}_q) \cdot P(\overline{R}_q)} \\ &\propto \frac{P(\boldsymbol{d}|R_q)}{P(\boldsymbol{d}|\overline{R}_q)} \end{aligned} \tag{3.8}$$

Surprisingly easy models have been used for that task, and several models have been briefly mentioned as *Generative Classifiers* in Sec. 3.2.1.1, e.g. the *Multivariate Bernoulli Naive Bayes Model*. In the IR community, this model is better known as the *Bernoulli Product Model* or *Binary Independence Model* (BIM) [138]. While binary models like the BIM follow the Boolean thought, models with keyword count information follow the vector thought. In its most basic form, the Probabilistic Model applies simple *Naive Bayes Models*. In the following, the BIM model will be described in more detail. Terms in documents and queries are assumed to be either present or absent $g_i(\cdot) \in \{0, 1\}$. The BIM model is defined as a similarity according to Ribeiro-Neto et al. [106] just as follows:

$$\begin{aligned} sim(\boldsymbol{d}, \boldsymbol{q}) &= \sum_{i=1}^{T} g_i(\boldsymbol{d}) \cdot g_i(\boldsymbol{q}) \cdot \delta_{i|R} \\ \delta_{i|R} &= \ln \frac{p_{iR}}{1 - p_{iR}} + \ln \frac{p_{i\overline{R}}}{1 - p_{i\overline{R}}} \\ p_{iR} &= P(g_i(\boldsymbol{q}) = 1|R_q) \\ p_{i\overline{R}} &= P(g_i(\boldsymbol{q}) = 1|\overline{R}_q) \end{aligned} \tag{3.9}$$

The simplex of the probabilistic framework bounds the relevance measurement between [0,1].

3.5.4 Bayesian Network Model

The three classical models *Boolean Model*, *Vector Space Model* [110] & *Probabilistic Model* [108] can all be expressed by a more general *Bayesian Network Model* (BNM). Ribeiro-Neto et al. [106] proofed that the BNM subsumes all three models. For all documents in the document collection and a query, the authors argue about a probability function $P(\boldsymbol{d}|\boldsymbol{q})$ that reflects their degree of coverage. Within their work, documents and queries comprise the 'universe of discourse' U as the whole set of keywords (the 'elementary concepts' in the U space). The factorization of the proposed BNM decomposed into the following:

$$\begin{aligned} P(\boldsymbol{d},\boldsymbol{q}) &= \sum_{u \in U} P(\boldsymbol{d},\boldsymbol{q},u) \\ &= \sum_{u \in U} P(\boldsymbol{d},\boldsymbol{q}|u) \cdot P(u) \\ &= \sum_{u \in U} P(\boldsymbol{d}|u) \cdot P(\boldsymbol{q}|u) \cdot P(u) \end{aligned} \tag{3.10}$$

The BNM is illustrated in Fig. 3.17 and its coverage function $P(\boldsymbol{d}|\boldsymbol{q})$ can be computed by the *Bayes Theorem*. The authors postulate to simply assume an equal prior about the query. During their work, it could be shown that this expression can be used to represent any of the classical models. The importance of this observation arises from the fact that all conclusion derived from the BNM will apply to all classical models as well.

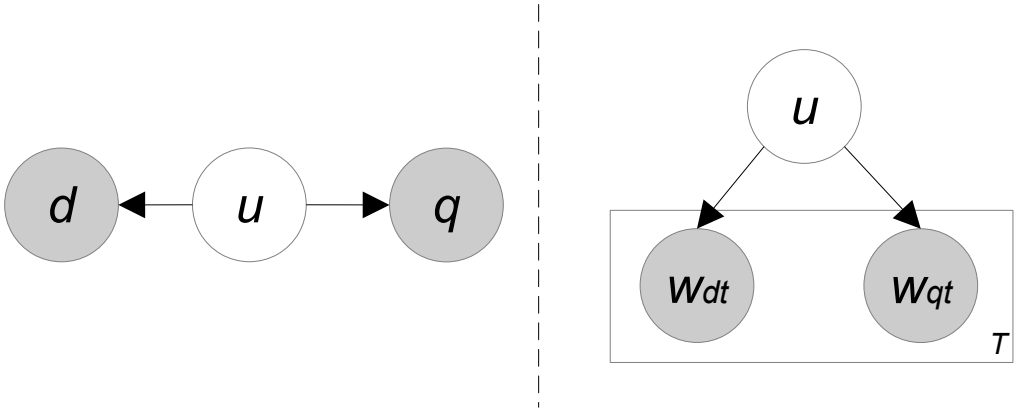


FIGURE 3.17: Left: The Bayesian Network Graph for the Bayesian Network Model [106]. Right: Binary Independence Model [138] as a Directed Graphical Model with plates. Nodes reflect model variables and edges their interaction: $A \rightarrow B = P(B|A)$. Latent (unknown) variables are marked white.

3.5.4.1 Bayesian Network Model & Boolean Model

The following proof is a direct transcript from the work of Ribeiro-Neto et al. [106]. Their proof works on the following link dependency equations:

$$\begin{aligned} P(\boldsymbol{q}|u) &= \begin{cases} 1 & \text{if} \quad \exists\, q_{ck} \,|\, (q_{ck} \in q_{dnf}) \wedge (\forall_i\, g_i(u) = g_i(q_{ck})) \\ 0 & \text{otherwise} \end{cases} \\ P(\overline{\boldsymbol{q}}|u) &= 1 - P(\boldsymbol{q}|u) \\ P(\boldsymbol{d}|u) &= \begin{cases} 1 & \text{if} \quad \forall_i\, g_i(\boldsymbol{d}) = g_i(u) \\ 0 & \text{otherwise} \end{cases} \\ P(\overline{\boldsymbol{d}}|u) &= 1 - P(\boldsymbol{d}|u) \end{aligned} \tag{3.11}$$

Lemma 3.5.1. *Equations Eq. (3.11) when applied to Eq. (3.10), define a set of relevant documents that coincides with the set of relevant documents returned by the classic Boolean Model through Eq. (3.6).*

Proof. Let S_{net} be the set of relevant documents returned by the *Bayesian Network Model* and let S_{bool} be the set of relevant documents returned by the *Boolean Model*.

$$S_{net} = S_{bool} \iff S_{net} \subseteq S_{bool} \wedge S_{bool} \subseteq S_{net}$$

Assume $S_{net} \nsubseteq S_{bool}$. Then, $\exists\, \boldsymbol{d} \,|\, \boldsymbol{d} \in S_{net} \wedge \boldsymbol{d} \notin S_{bool}$. From Eq. (3.10) and Eq. (3.11), $\boldsymbol{d} \in S_{net}$ implies $\exists\, u \,|\, P(\boldsymbol{q}|u) = 1 \wedge P(\boldsymbol{d}|u) = 1$. But then, u is a conjunctive component of q_{ndf} (i.e., $u \in q_{dnf}$) and $g_i(\boldsymbol{d}) = g_i(u)$ for all i. From Eq. (3.6) we conclude that $\boldsymbol{d} \in S_{bool}$ which contradicts our initial hypothesis. Analogously, we can prove that assuming $S_{bool} \nsubseteq S_{net}$ also leads to a contradiction. □

3.5.4.2 Bayesian Network Model & Vector Space Model

The following proof is a slight adaption of the work of Ribeiro-Neto et al. [106]. The proof works on the following link dependency equations:

$$
\begin{aligned}
P(\boldsymbol{q}|u) &= \begin{cases} \frac{w_{qi}}{|\boldsymbol{q}|} & \text{if} \quad u = u_i \text{ and } g_i(\boldsymbol{q}) = 1 \\ 0 & \text{otherwise} \end{cases} \\
P(\boldsymbol{d}|u) &= \begin{cases} \frac{w_{di}}{|\boldsymbol{d}|} & \text{if} \quad u = u_i \text{ and } g_i(\boldsymbol{d}) = 1 \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
\tag{3.12}
$$

Lemma 3.5.2. *By applying equations Eq. (3.12) to Eq. (3.10), we obtain a ranking whose document ordering is the same as the ordering provided by the Vector Space Model through equation Eq. (3.7).*

Proof. By substituting equations Eq. (3.12) to Eq. (3.10), the following can be written:

$$
\begin{aligned}
P(\boldsymbol{d}, \boldsymbol{q}) &= \sum_{u \in U} P(\boldsymbol{d}|u) \cdot P(\boldsymbol{q}|u) \cdot P(u) \\
&= \sum_{u_i \,|\, g_i(\boldsymbol{q})=1 \wedge g_i(\boldsymbol{d})=1} \frac{w_{di}}{|\boldsymbol{d}|} \cdot \frac{w_{qi}}{|\boldsymbol{q}|} \cdot P(u_i) \\
&= \frac{1}{|\boldsymbol{d}| \cdot |\boldsymbol{q}|} \cdot \sum_{u_i \,|\, g_i(\boldsymbol{q})=1 \wedge g_i(\boldsymbol{d})=1} w_{di} \cdot w_{qi} \cdot P(u_i) \\
&= \frac{1}{|\boldsymbol{d}| \cdot |\boldsymbol{q}|} \cdot \boldsymbol{d} \cdot \boldsymbol{q} \cdot P(\boldsymbol{u}) \\
&= \frac{\boldsymbol{d} \cdot \boldsymbol{q}}{|\boldsymbol{d}| \cdot |\boldsymbol{q}|} \cdot P(\boldsymbol{u})
\end{aligned}
$$

The joint probability distribution already shows the strong connection to Eq. (3.7). By applying the *Bayes Theorem*, the ranking distribution $P(\boldsymbol{d}|\boldsymbol{q})$ can be formulated. Within, $P(\boldsymbol{u})$ realizes a normalization constant which does not affect the document ordering. □

3.5.4.3 Bayesian Network Model & Probabilistic Model

The following proof is a transcript from the work of Ribeiro-Neto et al. [106] with minor adaptions. The proof works on the following link dependency equations:

$$\begin{aligned} P(\boldsymbol{q}|u) &= \begin{cases} 1 & \text{if} \quad \forall_i \, g_i(u) = g_i(\boldsymbol{q}) \\ 0 & \text{otherwise} \end{cases} \\ P(\boldsymbol{d}|u) &= \frac{\sum_{i|g_i(u)=1} g_i(\boldsymbol{d}) \cdot \delta_{i|R}}{\sum_{i|g_i(u)=1} \delta_{i|R}} \\ P(\overline{\boldsymbol{d}}|u) &= 1 - P(\boldsymbol{d}|u) \end{aligned} \tag{3.13}$$

Lemma 3.5.3. *Equations Eq. (3.13) applied to Eq. (3.10) define an ordering of relevant documents (i.e. ranking) that coincides with the ordering defined by the classical Probabilistic Model through Eq. (3.9).*

Proof. Eq. (3.13) implies that $u = \boldsymbol{q}$ is the only concept for which $P(\boldsymbol{q}|u) = 1$. Thus, Eq. (3.9) after proper substitution, can be rewritten as:

$$P(\boldsymbol{d}|\boldsymbol{q}) = \eta \cdot \frac{\sum_{i|g_i(\boldsymbol{q})=1} g_i(\boldsymbol{d}) \cdot \delta_{i|R}}{\sum_{i|g_i(\boldsymbol{q})=1} \delta_{i|R}} \tag{3.14}$$

The difference between this equation and Eq. (3.9) lies in the normalization factor η. This normalization is a consistent transformation between the joint probability to the conditional probability. This factor does not depend on the document $\boldsymbol{d}$ being ranked and thus, do not affect the ordering of the documents. Therefore, the rankings generated by Eq. (3.14) and Eq. (3.9) coincide. □

Chapter 4

Related Work

4.1 Information-Seeking & Information Search Behavior

Research to investigate differences in *Information-Seeking Behavior* was done by Marchionini [79]. He investigated the differences between user's search behavior caused by *open*, i.e. imprecise, tasks where the *Information Need* can not be specified precisely and *closed*, i.e. precise, tasks that lead to a clear derivable information need. In his studies, Marchionini was able to show that especially novices need more time and have more difficulties to specify the queries for open tasks. With the focus on easy and difficult tasks of closed nature, Aula et al. [7] presented the results of a large-scale study with the goal to detect the task difficulty based on the *Information Search Behavior*. In their work, Aula et al. described what kind of strategies users apply if they had difficulties to solve the tasks. Liu et al. [77] investigated the influence of tasks with different difficulty levels and different types on the user behavior and report that users increase the number of queries, view more results or spend more time on result pages. Their findings show that dwell time alone is not a reliable measure for prediction whether users are performing difficult tasks. Hassan et al. [55] investigated user behavior if they experienced difficulties during the search or if users are exploring and highlighted the need for advanced methods to distinguish between the user's situation. Athukorala et al. [6] conducted a user study with 32 computer science researchers and explored different parameters that can help to distinguish exploratory from *lookup* tasks. They found that the length of the first query is shorter in exploratory tasks, and users spend more time reading the documents and scroll significantly deeper than in lookup tasks. Kuhlthau [72][73] proposed an information-seeking model with six stages, see *Kuhlthau's Model*, and corresponding search activities in terms of a so-called information search process (ISP). Shah et al. [120] proposed an approach to clearly distinguish between the six stages of the ISP over two search sessions for individuals. For collaborative information-seeking, the distinction between *Exploration*, *Formulation* and *Collection* stages was vague [120]. In respect to search user interfaces (SUIs), Huurdeman and Kamps [60] investigate different exploratory features of a user interface, e.g. rapid query refinement, and suggest that there are differences in the interaction flow with search user interface features depending on the stage of an exploratory search. Not only the SUIs, but also retrieval algorithms should provide mechanisms to support *exploratory search* tasks, e.g. ensuring a diversity of search results to cover as many different aspects of the query topic as possible [134].

4.1.1 Models for Search Activities

To model the search activity of users, *Markov Models* are commonly applied in IR. For example, *Markov chains* were used by Tran and Fuhr [128] to model Amazon book searches. Ageev et al. [3] applied different Markov Models approaches to predict user's success for *Fact-Finding* search tasks. Further adaptions have been used by Hassan et al. [54] to model user search behavior with transition times. They showed that this approach performs significantly more accurate than traditional relevance-based models for predicting user search goal success. Cole et al. [28] used Markov Models to study interaction patters for complex search tasks based on users' interaction and gaze data. *Hidden Markov Models* [99] were applied by Yue et al. [139] to compare collaborative and individual *exploratory searches*. Alternative approaches to the Markov Models were used by Athukorala et al. [5] by applying a *C4.5 Decision Tree* [98] with three parameters (query length, reading time, and cumulative clicks) to determine a user's search activity while searching for scientific literature. Shah et al. [121] applied a *Support Vector Machine* [131] to forecast how well users will perform in the later stages of the exploratory search process based on the actions they are currently performing.

4.1.2 Eye Movement & Information-Seeking Behavior

Several studies have been conducted to analyze different reading strategies in respect to particular tasks. Clark et al. [25] conducted an eye tracking study for *Wikipedia* searches, with the focus on usage of structural features during search tasks. Besides several other factors, the authors analyzed participants reading strategies, e.g. *Skimming* and *Scanning*, during searches. The authors conclude that participants preferred Skimming during searches amongst very long documents to get an understanding of the article's relevance for their task [25]. Scanning was a more common behavior and focused on Wikipedia's structural components to look for keywords or phrases to match the task [25]. Holmqvist et al. [59] conducted a study to compare newspaper and net paper reading in respect to Scanning and thorough *Reading* behavior. The study gives evidence for the assumption that Scanning is more prominent in net paper reading than in newspaper. The authors state that parts of their experiment reflect the fact that Scanning a newspaper is made in search of entry points for further consumption [59]. Clark et al. [26] analyzed ocular behavior of participants for analyzing the content of e-mails. Their findings provided support for the theory that structural information, such as format and layout, plays an important role in text categorization [26]. The authors conclude that structural formatting enables participants to employ intensive Scanning behavior. Rodeghero and McMillan [109] conducted an eye movement study on programmers for source code as a summarization task. The authors showed in their specific scope that readers apply modified strategies of reading to fulfill precise summarization (and to some extent search) tasks. Their findings indicate that participants follow reading patterns, e.g. Skimming, that provide quicker understanding as opposed to a more in-depth understanding [109]. The authors conclude that their participants modify their reading strategy towards rapid techniques such as Skimming and jumping within the text.

4.2 Eye-Tracking & Reading

4.2.1 Automatic Reading Detection & Rule-Based Systems

Campbell & Maglio [22] state that the availability of eye tracking technology enables researchers to take advantage of this powerful source of information to determine user intentions and interests. The authors argue that reading detection provides the means to infer user interest based on the type of behavior, such as thorough *Reading* for high, *Skimming* for medium and *Scanning* for low interest, see *Reading and Information Processing* in Sec. 2.2.3.3 for in-depth descriptions of these reading variants. A rule-based system is proposed by the authors to make an automated reading detection with a high *Accuracy* rate of nearly 100% [22]. The eye movement sequence is tokenized and each token is assigned with an evidence value for reading, see Tab. 4.1. Once a threshold value is reached, a chunk within the sequence is detected as a reading behavior. If the threshold is crossing the boundary of 30, Reading is detected. Otherwise, the behavior is detected as Scanning. Skimming remains future work for the proposed method. Even though the simplicity and interpretability of such rule-based systems are quite convincing, the lack of adaptivity reduces the practicability of such systems. Further, the specific implementation of the selected rules heavily influences the system's performance. The proposed tokenization scheme works on static distance measures on the displayed screen. Just by zooming in and out of the text, the prediction of such a rule-based system should fall apart. Therefore, the need of more contextual, adaptive and learnable reading detection models remain.

Distance, direction, axis	Token	Points
Short right X	Read forwards	10
Medium right X	Skim forwards	5
Long right X	Scan Jump	Reset
Short left X	Regression saccade	-10
Medium left X	Skim jump	-5
Long left X	Scan jump	Reset
Short up Y	Skim jump	-5
Medium up Y	Scan jump	Reset
Long up Y	Scan jump	Reset
Short down Y	Anticipatory saccade	0
Medium down Y	Skim jump	-5
Long down Y	Scan jump	Reset
Long, medium left X and short down Y	Reset jump	5

Note: positive point values indicate evidence supporting Reading and negative numbers indicate evidence against Reading.

TABLE 4.1: Adaption of [22]: Reading detection via tokenization of eye movements and evidence values for Reading.

4.2.2 Automatic Reading Detection & Learnable Systems

Kollmorgen & Holmqvist [67] highlighted the necessity to analyze reading behavior of participants in natural conditions, such as in long text with embedded images e.g. web pages. It is argued by the authors that just assuming participants are (generally) reading when they are looking at text is a poor solution because it would include *Scanning* which is a very different process [67]. The authors propose quite simple yet esthetic *Hidden Markov Models* (HMM) [99] for an automatic reading detection approach. The eye movement is assumed to be a sequence of *Fixations & Saccades* (& blinks), that are either part of a reading activity or a non-reading activity. Further, the states are associated with attributes, such as gaze event durations and x,y coordinates of the gaze. This description naturally leads to the 6-State HMM, that can be seen in Fig. 4.1. The authors compared the model prediction on labeled data via *Supervised Learning* with 93% *Accuracy* and unlabeled data via *Unsupervised Learning* with 87% accuracy. Both settings were previously described as *Dynamic Naive Bayes Models* (DNBM) [80] and HMMs. To put the findings into perspective, the authors compared their proposed model with an *Artificial Neural Network* (ANN). DNBMs outperformed the ANN, which outperformed HMMs. Biedert et al. [12] proposed a different model type for automatic reading detection, which resulted in 86% accuracy. The described approach is based upon extensive feature engineering and expert knowledge for data preprocessing in combination with a *Support Vector Machine* (SVM) [131]. Kelton et al. [65] proposed a *Region Ranking SVM* for reading detection to achieve 82.5% accuracy. A broader overview of approaches associated with reading detection was given by Gündüz et al. [52] ranging from a *Naive Bayes Model* to a *Decision Tree* [98].

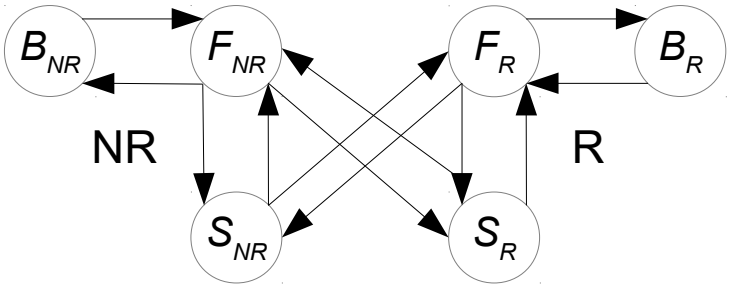


FIGURE 4.1: Adaption of [67]: Reading detection via 6-State *Hidden Markov Models* [99] partitioning *Fixations & Saccades* into reading and non-reading sequences. The state transition diagram shows non-reading states (NR) on the left and reading states (R) on the right. Fixations (F) are either followed by a saccade (S) or a blink (B). Blinks and saccades are followed by a fixation. F, S and B are either done during R or NR, resulting in 6 states.

4.2.3 Individual Factors in Reading

Many factors influence the eye movement, especially reading, which results in challenging aspects for automated detection. First, there is an anatomical variance of the eye itself. Mean pupil size varies between male & female and changes by age [13]. Second, developmental changes are factors influencing the reading behavior, with dyslexic and bi/multi-lingual readers being examples of extremes. Third, cognitive load is a factor, e.g. in the form of fast vs. slow reading, silent vs. oral reading and more. Forth, lexical and sentential constraints in the text effects the reading behavior. Rayner presented an outstanding overview of these factors and more in [101] as a summary of over 20 years of research in eye movements in reading and information processing. The entirety of factors are indeed so unique that automatic reader identification is possible. Landwehr et al. proposed *Dynamic Bayesian Networks* to identify readers with an 98.25% *Accuracy* [76]. These observations indicate a challenging task for a generalized reading detector which does not overfit towards individual factors.

Part III

Information Search Behavior Profiles

Chapter 5

Research Questions

5.1 Research Question 0

RQ0: ***What is the preexisting state of research, formalisms, models and technical solutions for information search behavior? Is there a unified way to state the problem scope? If not, is it possible to create a unified formalism or framework to create comparable findings? Is it possible to bridge the gap in this interdisciplinary field of Computer Science, Information Science and Psychology?***

The entirety of *Fundamentals* in Part. II was dedicated to **RQ0**. In particular, *Information Behavior Models* (Sec. 2) focused on the conceptual basis of what a *(User) Behavior Model* should satisfy to represent the search behavior of a user or user group. This decomposed into *Information-Seeking Behavior* (Sec. 2.1) and *Information Search Behavior* (Sec. 2.2). While Information-Seeking Behavior describes conceptual strategies of users within the search, Information Search Behavior meanly describes actions a user implements to execute the search. In the following, a (User) Behavior Model is aimed to be build, that is capable to make inference based on such actions, and to draw conclusions about underlying strategies. *Modeling* (Sec. 3) provided an overview of models capable of such inference, and therefore provide the mathematical realization of such desired models. *Related Work* (Sec. 4) provided an overview of previous research associated to that task. Even though, not all the work explicitly use the term User Model or (User) Behavior Model. With the aim to create such models and to add insights in the information search process, the following terminology is postulated to bridge the gap between the interdisciplinary aspects of this complex field. From that point on, this new terminology will be used throughout the rest of the thesis:

- *Action*:
  Is a singular action, activity or interaction of an individual to recognize, interact with or manipulate its surrounding. In the context of information search, such actions are rather described by Information Search Behavior. Examples for such actions in respect to online searches can be: a web page visit and the clicking, scrolling, dwell-time, fixation, saccade activity on that page etc.

- *Goal*:
  Is an aim, outcome, goal or achievement of an individual that is wished to be realized. In the context of information search, such goals are referred to as the *Information Need*. Such needs can be clear and precise for the individual, but also vague and hard to define. Examples for such goals can be: the knowledge about a particular fact or the overview about a particular domain of interest.

- *Strategy*:
  Is a plan, algorithm or strategy of an individual as the implementation, composition or sequence of *actions* that is expected to achieve a *goal*. In the context of information search, such strategies are rather described by Information-Seeking Behavior. Examples for such strategies can be: *Fact-Finding* or *Exploratory search activities*. Strategies can decompose into sub-strategies such as *Exploration*, *Collection*, *Selection*, etc. The lowest level of decomposition will result in a singular action.

- *Behavior*:
  Is a particular expression, combination or triple of $(goal, actions, strategy)$. The *behavior* of an individual is defined as a particular expression of *actions* that are combined into a *strategy* to achieve a *goal*. It can be assumed that a behavior can be observed as clusters within a population. This assumption is based on the idea that behavior arose in individual development and/or collective evolution, guided by the success rate as an optimization function to achieve such goals.

The described concepts will new be translated into the words of the data analysis community. *Goals* can be described by a set of desired outcomes. *Actions* can be described as a set of actions an individual can execute. *Strategies* can be defined as a set comprising all possible combinations/sequences of actions to achieve goals. *Behavior* can be defined as a joint probability function $P(goal, actions, strategy|\boldsymbol{\theta})$. It can be assumed that each individual has an own *Behavior Model* $\boldsymbol{\theta}$ that follows an individual's preference to assign certain actions in strategies to fulfill a goal. It can further be assumed that certain instances of behavior are more potent to succeed a goal than others. Through the complex optimization by enforced interaction with reality, certain behavior instances arose as (more or less successful) peaks within the joint probability space. *Bayesian Networks* can be used as a representation of this joint probability function. They will be called *(User) Behavior Model* if and only if they model the combination of actions, goals & strategies. In the following sections, experiments were designed with predefined search tasks, also implying predefined Information Needs; the goal to be achieved in the search. In a user study, participants could implement actions to achieve such goals according to their own strategy. Using data analytical models, the aim of this thesis is described by the identification of an over-represented search behavior via (User) Behavior Model to make conclusions about the characteristics of the information search process.

5.2 Research Question 1

RQ1: ***Which cognitive models for users in online searches can be used, and do they provide useful interpretations for Information Retrieval? What actions can be implemented by a user to achieve their search goals, and what strategies of users can be observed? Which aspects can we derive from that to further improve the usability of Information Retrieval systems? How can we exploit the interdependency between actions, strategies and goals?***

In the pursuit of implementing the desired *(User) Behavior Model* to analyze information search aspects, the terminology of Sec. 5.1 will be used to work within a unified framework. **RQ1** is centered on the formulation of *actions*, *goals* & *strategies* individuals are faced with during their online searches and how to draw conclusion from it in respect to the Information Retrieval scenario. For that, a precise formulation of the task is needed, otherwise derived conclusion remain in the space of unnecessary fuzziness. The following will shortly formulate and summarized the concepts.

- *Goal*
  A *goal* is the desired knowledge about an *Information Need*. Such a need can be triggered by a search task. Without any structural properties, there are as many goals as there are search tasks, potentially infinite many. Within Sec. 2.1, it could be observed that some search tasks can be seen as precise or 'formulatable', some task can be described as unprecise or vague. These tasks are referred to as *Fact-Finding* and *Exploratory* search tasks. Their goals fall into categories of either being closed in case of a Fact-Finding task and open in case of an Exploratory task. A goal is defined as *closed* if there exists an outcome of the search that fully satisfies the information need, e.g. the answer to a factual task. A goal is defined as *open* if there exist several outcomes that can only partially satisfy the information need. Even the knowledge of all outcomes that partially satisfy the information need are not sufficient to fully satisfy it, e.g. the exploration within a domain of interest.

- *Action*
  An *action* is a singular interaction of an individual within the online search. Because of the limited set of interactions an individual can implement with a computer, it is possible to clearly define this set of actions. For practical reasons, only a subset of such will be defined. Within the context of this thesis, these actions will be either derived from log-files or eye-tracking data. Singular actions derived from *Eye-Tracking* can be stated as *Fixations & Saccades*. Singular actions derived from log-files can be stated as: a *web page visit*, a *clicking* event, a *scrolling* event and the *dwell time* on a web page. In the context of Information Retrieval, it seems plausible to group these actions into chunks associated to a particular web page. For such a chunk, a profile of actions can be derived. Such a profile can be described via the mean of actions executed on it or its accumulation, e.g. in the form of a summation. Both result in singular measurements which do not reflect the composition or sequence of these actions, e.g. their *strategy*.

- *Strategy*
  A *strategy* is a composition or sequence of *actions* to achieve a *goal*. Strategies can comprise sub-strategies as long as they remain still a composition of actions. A strategy at its lowest level will decompose into a singular action. Probably the 'lowest' level of the strategy hierarchy can be described by *Eye Movement in Reading* and *Reading and Information Processing*. As a composition or sequence of *fixations* & *saccades* with a precise goal, they satisfy the condition of a strategy. Further, reading and its variants provide a reasonable interpretation that can be exploited to characterize 'higher' levels in the strategic hierarchy. Probably at the 'middle' level of the strategy hierarchy, *Navigation & Probabilistic Regular Grammars* can be stated. As a composition or sequence of *web page visits* with a goal, they satisfy the condition of a strategy. The importance of web trail analysis of users have been stated previously, and it can be assumed that patterns in the navigational trail might be indicative for the search process, e.g. rapid query refinements. Probably the 'highest' level of the strategy hierarchy can be described by the entirety of *Information-Seeking Behavior*. The thesis will solely focus on *Exploratory* and *Fact-Finding* search activities. With their connection of implicit goals being either *closed* or *open* and as a composition comprising all the 'below' levels, they satisfy the condition of a strategy. The three described strategic levels can be considered to be in accordance to Wilson's nested model of information behavior, see Sec. 2 Fig. 2.1. It simply extends the nested model with intermediate levels between Information-Seeking Behavior and Information Search Behavior.

- *Behavior*
  Behavior is a joint probability function over the triple $(goals, actions, strategies)$ measuring an individual's preference to assign *actions* into *strategies* to achieve a *goal*. Any model that realizes inference within that triple can be considered as an instance of *(User) Behavior Models*. While *Models for Unstructured Prediction* are suitable for the analysis of *actions*, *Models for Structured Prediction* are more suitable for the analysis of *strategies*. The combination of all levels together can be implemented by *Bayesian Networks*. Such a global inference network will be used as the realization of the desired *(User) Behavior Model*.

The described concepts are illustrated in Fig. 5.1 and put into perspective with the concepts of *Information-Seeking Behavior* and *Information Search Behavior*. The provided terminology will not only serve as the foundation for implementing the desired *(User) Behavior Model*, but also will serve as the foundation of any interpretation drawn from that model. In the following, (User) Behavior Models will be used with the aim to identify and characterize *Fact-Finding* and *Exploratory* search activities. By exploiting the mathematical characteristics of these models, these profiles are aimed to described actions and strategies a user implements during their respected search activity. A well-defined user study will serve as the basis of this approach.

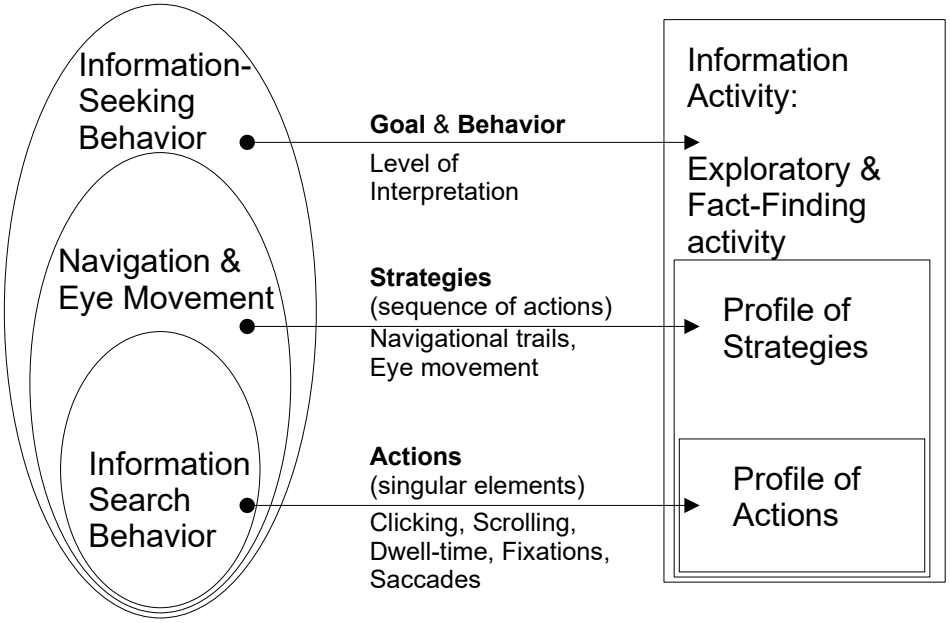


FIGURE 5.1: Conceptual representation of the thesis scope. Provided definitions of Sec. 5.1 are put into perspective with concepts of *Information-Seeking Behavior* and *Information Search Behavior* in Chap. 2. (User) Behavior Models are used to characterize the profile of actions and strategies during online searches in respect to search activities of well-defined goals.

5.3 Research Question 2

RQ2: ***Which mathematical & computational (User) Behavior Model can be used to draw a conclusion about the information search behavior of users? What is their potential in respect to inference, and what are their limitations? How can they be applied to gain knowledge about the search process?***

In respect to the fundamentals of **RQ2**, it can be stated that *Bayesian Networks* provide a flexible and efficient way to model abstract data. They can generally be applied in all Machine Learning settings, e.g. *classification* (*Supervised Learning*) & *clustering* (*Unsupervised Learning*). This model family ranges from *Models for Unstructured Prediction* to *Models for Structured Prediction*. Further on, arbitrarily complex networks can be created by combining them with sub-models, as long as the global network remain *directed* & *acyclic*. In general, Bayesian Networks can contextualize any information flow by suitable factorization assumptions, and they can naturally fuse multi-modal data sources. Inference fully decomposes into the *Max* & *Sum Product Algorithm* (Algo. 5 & 6) and remain statistical consistent within the network.

In the pursuit of implementing the desired *(User) Behavior Model* to analyze information search aspects, the terminology of Sec. 5.1 will be used to work within a unified framework. **RQ2** will also decompose into the field of application. Bayesian Networks realize a complex joint probability distribution which can be considered as the realization of the *behavior* measurement when the network comprises nodes for *actions*, *goals* & *strategies*. Singular actions will be modeled via 'leaf' nodes in the network, while strategies can be modeled via 'inner' nodes as compositions or sequences of the 'leaf' nodes. A dedicated 'root' node in the network will realize the goal. The proposed (User) Behavior Model for search activities is illustrated in Fig. 5.2. During an online search session, a user visits a sequence of web pages: pages providing an option to formulate a query (*Query*), pages providing a search engine result page (*SERP*) and page presenting content (*Page*). In each web page, the user executes actions in the form of log-file derived feature such as *Clicking*, *Scrolling* and simply remaining on that page by *Dwell-Time*. In addition, the user will inspect the presented web page by *Eye Gaze* measured with an eye-tracker. The eye movement will form complex strategies for *Reading*, *Orientation* or inspecting *Images*. These patterns comprise *Fixations* and *Saccades*. After a web page has sufficiently been inspected by the user, the search will proceed towards another web page. The *Navigation* itself will form again a complex strategy as a sequence of visited web pages. By observing multiple search sessions, the network is populated with the measured behavior. After observing multiple individual search sessions (model training), the network can be used to draw conclusions about the desired characteristics. In the following, Bayesian Networks will be used to identify and characterize *Fact-Finding* & *Exploratory* search activities based solely on the measurement of actions and (sub)strategies, such as reading & navigation. Therefore, the potential of Bayesian Networks will be explored as realizations of mathematical & computational (User) Behavior Models.

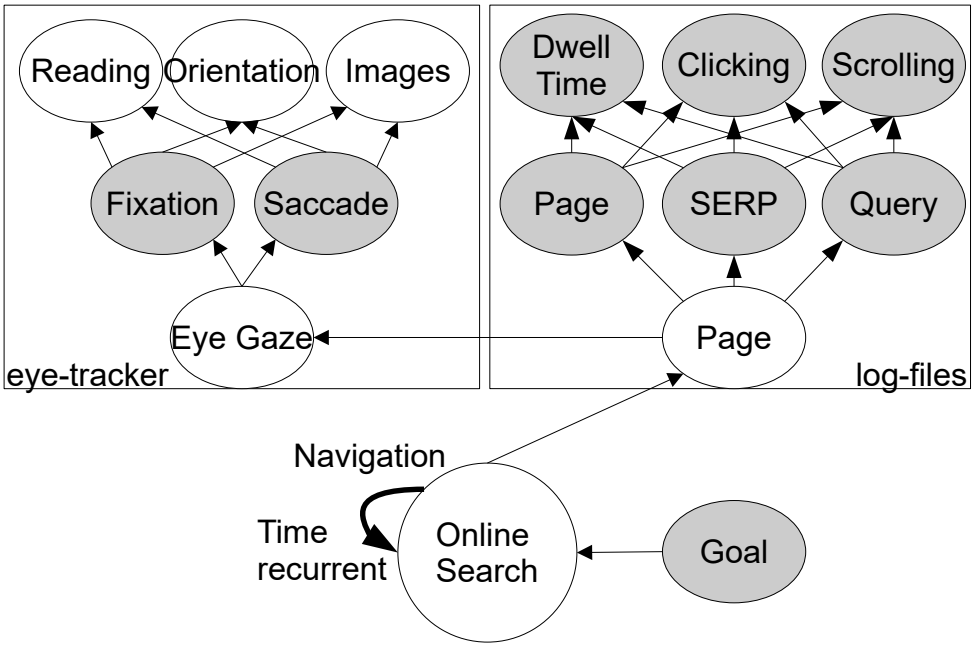


FIGURE 5.2: Graphical representation of an online search session. With a *Goal* in mind, a user navigates through the Internet by visiting web pages, either by *Query*, *SERP* & *Page*. Within each web page, a user will implement actions, either measured by log-files (*Clicking*, *Scrolling*, *Dwell-Time*) or by an eye-tracker (*Fixation*, *Saccade*). The eye movement itself will form complex pattern for *Reading*, *Orientation* & *Images*. This representation coincides with the structure of the Bayesian Network postulated for the (User) Behavior Model, namely the Information Search Behavior Profile Model.

5.4 Research Question 3

RQ3: ***In case that mathematical & computational (User) Behavior Models provide reliable information to draw conclusions about the Information Behavior: How can this information be exploited in the setting of Information Retrieval (IR)? Is it possible to use such (User) Behavior Models in rankings by an IR system? Can a static machine adapt to changed behavior in users? What does this mean in the context of the Human-Machine-Interaction?***

In the pursuit to use *(User) Behavior Models* in combination with *Information Retrieval* (IR) systems, the terminology of Sec. 5.1 will be used to work within a unified framework. **RQ3** is mainly centered around the potential to incorporate (User) Behavior Models in rankings. The *Bayesian Network Model* [106] can be considered to be the most promising ranking family. As a generalization for the classical ranking models, e.g. *Boolean Model*, *Vector Space Model* [110] and *Probabilistic Model* [108], findings based on it will generalize for the classical models as well. Additionally, the Bayesian Network Model is an instance of *Bayesian Networks* and the proposed (User) Behavior Models are Bayesian Networks as well. The combination of both approaches seems more than plausible and will result in a fully consistent modeling setting.

- *Goal* & *Behavior*
  A *goal* is the desired knowledge about an *Information Need*. Goals have been classified as either *open* in case of *Exploratory* search activities or *closed* in case of *Fact-Finding* search activities. The most ambitious aim of IR systems can be postulated as the characterization of individualized relevance concepts of users. Because the *behavior* of a user is affected by their underlying goals, the analysis of the search via *(User) Behavior Models* can directly be used to approximate this relevance concept. Therefore, the combination of (User) Behavior Models with the ranking of IR systems will result in a contextualized, behavior-driven and user-centered ranking that is goal oriented.

- *Action* & *Strategy*
  Actions and *strategies* are the measurable expressions of users during their search. Actions and strategies in combination with a *goal* form a *behavior* that can be analyzed by *(User) Behavior Models*. Within the scenario of IR, the goal of the user is unfortunately either not known or simply not measurable. By exploiting the functional interdependency of $(goal, action, strategy)$ with behavior, (User) Behavior Models can be used to approximate the goal by measuring actions and strategies.

Within this thesis, it is planned derive a behavior-driven ranking paradigm. For that, classical ranking approaches are combined with *(User) Behavior Models* with the aim to adapt towards a goal oriented ranking by measuring behavior aspects. The reasoning behind was previously explained. The behavior-driven ranking approach will be design with the focus on *Exploratory* and *Fact-Finding* search activities.

Chapter 6

User Study - Design

6.1 Search Tasks

In pursuit to analyze search activities of users in complex search sessions, these search activities need to be recorded. For that, a well-defined experiment needs to be created which is able to trigger the desired activities. Founded on the description in Sec. 2.1 *Exploratory* and *Fact-Finding* search activities emerged as promising candidates for further analysis. As dichotomous concepts, it is easy to draw conclusion in the form of pairwise comparisons and/or by ratios. Further, both tasks can be associated with properties of the *cognitive model* that a user might have when confronted with such tasks. Exploratory search activities will be triggered by Exploratory tasks, which can be described as a rather *open* task. In that context, an open task is defined by an *Information Need* which cannot be specified precisely. This results in a rather fuzzy expectation by the user that is aimed to be further investigated during the search session. This expectation manifests in the user's cognitive model, and the search activity should be affected by it. In contrast, a Fact-Finding search activity will be triggered by Fact-Finding tasks, which can be described as a *closed* task. In this context, a closed task is defined by an Information Need that can be more or less defined clearly. This results in a clear expectation about this fact that is aimed to be found during the search session by the user. This expectation manifests in the user's cognitive model, and the search activity should be affected by it. Both concepts can complement each other in certain aspects. Both search activities can be compared against and interpreted.

In total, twelve Fact-Finding tasks were designed, with six task being assumed to be easy and six to be assumed to be hard. The individual tasks can be found in table 6.1. Search tasks are considered to be *easy* if the answer can be found in the first search engine result page (SERP), either in the form of a high-lighted text snippet or a presented page that comprises the desired information. A task is considered to be *hard*, if the first SERP does not provide sufficient information to fully answer the task. This means that either more than the first SERP needs to be inspected, more than one web pages needs to be inspected, or one promising web page needs to be inspected in great detail by reading one long document and derive a complex answer from it. The task categorization were tested by two people before the experiment.

ID	Level	Task description
1	easy	*In what year did the Google search engine went online for the first time?*
2	easy	*How old is Mickey Mouse today?*
3	easy	*According to current information, how many rooms are in the Buckingham Palace in London?*
4	easy	*What are sciaphobs afraid of?*
5	easy	*How many men were on the moon until June 2004?*
6	easy	*What is the name of the largest passenger aircraft?*
7	hard	*In what period of the Paleozoic era the first reptiles appeared?*
8	hard	*What percentage of German men aged between 70 and 79 suffer from diabetes (data for the year 2011)?*
9	hard	*Which word is coded in Morse code as "− . − . − − − − − − . − ..″?*
10	hard	*Which actor won the same year the Golden Raspberry Award for Worst Actor, Worst Supporting Actor and Worst Supporting Actress?*
11	hard	*What is the largest known planet in the binary star system Kepler-47?*
12	hard	*How many years passed between the first flight of the Kitty Hawk Flyers and Neil Armstrong's moon landing?*

TABLE 6.1: Fact-Finding tasks of the user study with assigned complexity levels, e.g. easy or hard. All tasks are translated from German.

In respect to Exploratory tasks, the design of the task is a bit more challenging because such tasks need to suffice more complex properties. Exploratory activities are characterized by aspects of learning & investigation, they involve a level of uncertainty and generally are described by being ill-structured & open-ended problems, see Sec. 2.1.3. Fortunately, the related work provided two examples for Exploratory tasks that this user study work with. Both tasks can be found in the following.

[Expl$_1$, adapted from [51]] *Your friends are planning to build a new house and have heard that using solar energy panels for heating can save a lot of money. Since they do not know anything about home heating and the issues involved, they have asked for your help. You are uncertain as well, and do some research to identify some issues that need to be considered in deciding between more conventional methods of home heating and solar panels. Afterwards you want to discuss this topic with your friends and, therefore, make some notes.*

[Expl$_2$, adapted from [137]] *You are flying to Moscow next month. During the travel arrangements you learn that body scanners are being used in many airports as part of routine security procedures. You start thinking about health issues related to their use. Your friends want to calm you down and say that people are exposed to different kind of radiation every day. You want to learn more and start a research to gather a range of information about radiation and health. Afterwards you want to discuss this topic with your friends and, therefore, make some notes.*

The lab experiment was designed for users to perform search tasks in blocks. Two of them were Exploratory tasks ($Expl_1$ & $Expl_2$) and the 12 Fact-Finding tasks were combined into one *multitasking search* [124] block. The 12 tasks within the multitasking search block were randomized. Each block was limited by a time restriction of 20 minutes. Therefore, the experiment for each participant took at maximum one hour time. A *latin square* study design was implemented to vary the order of task blocks for each participant. The applied design can be found in Fig. 6.1.

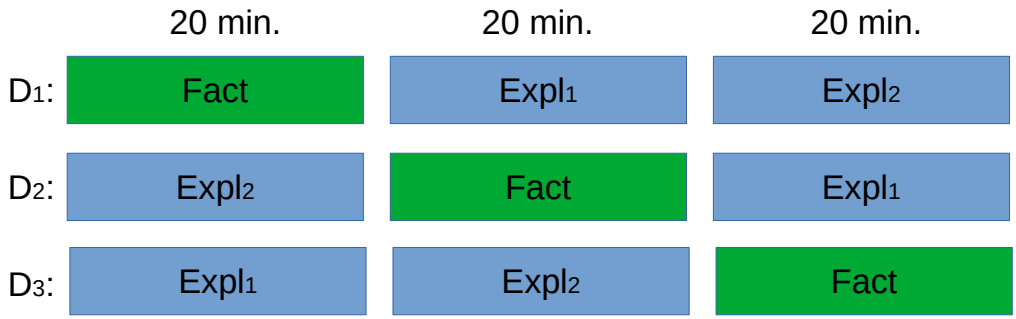
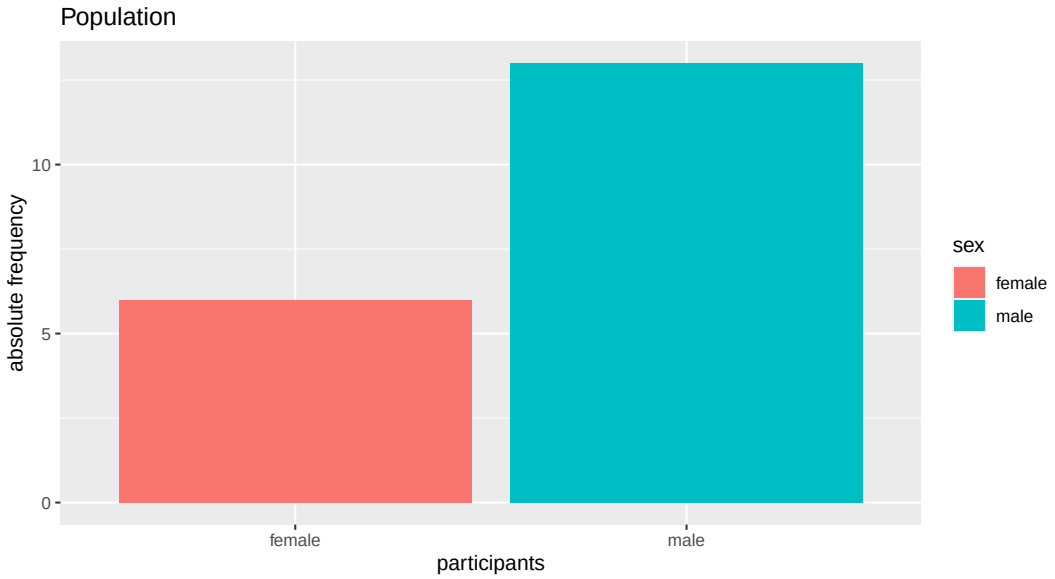


FIGURE 6.1: The study design comprises three blocks, with each being restricted for 20 minutes. Two blocks comprise an Exploratory task ($Expl_1$ & $Expl_2$) and one block a multitasking search with randomized Fact-Finding tasks. The user study varied the order of blocks between participants, e.g. by D_1, D_2 or D_3.

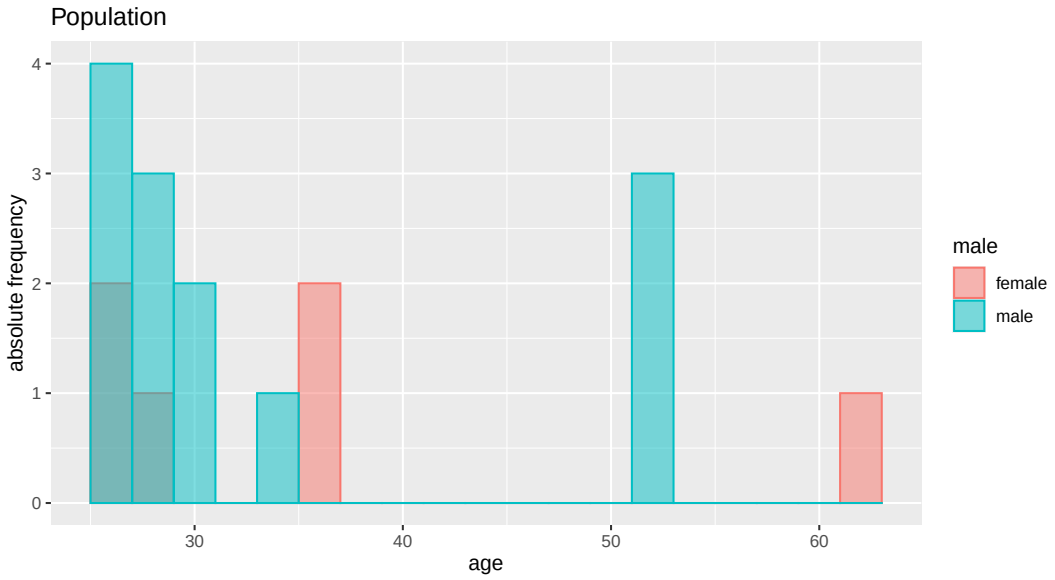
6.2 Participants

The user study comprises measurements from 19 participants. All of them were recruited via mailing lists, and no one received any monetary reward for the lab study. The distribution of the participant's sex was heavily skewed, with 13 male and 6 female subjects, see Fig. 6.2a. With a mean age of 34.5 the participants are representative for an adult cohort. Nonetheless, there is no homogenous distribution within the participant's age, as it can be seen in Fig. 6.2b. The majority of the subjects are in their mid-twenties to mid-thirties. Four participants are older than fifty and could be seen as outliers. This factor results in a rather inflated standard deviation of 11.33 years in age.

In addition to the biological metadata, further aspects characterizing the participant's familiarity to online search tasks were recorded. The cohort consists of rather high educated individuals, as it can be seen in Fig. 6.3a. Most of them are associated to the STEM field and the majority of the participants have a computer science background as PhD students. This was to be expected. As a user study from the faculty of computer science, colleagues showed solidarity to join the experiment and heavily skewed this distribution. All participants reported using the Internet on the daily basis, using it to search for information, checking/writing emails and searching with Google, see Fig. 6.3b for a detailed overview. Participants also described Internet activities associated towards the private sector, such as connecting with friends and gaming. The majority of 17 participants use the Internet for work. Therefore, the cohort is also characterized by highly trained users that presumably have evolved skills in online search sessions. Participants report an estimated average search time of ca. 10 minutes for an abstract but usual search task, see Fig. 6.3c. Nonetheless, extremes for search durations ranging from 1 minute to more than 1 hour have been reported as well. Overall, the average search time is non-symmetric and heavily tailed towards longer search times.

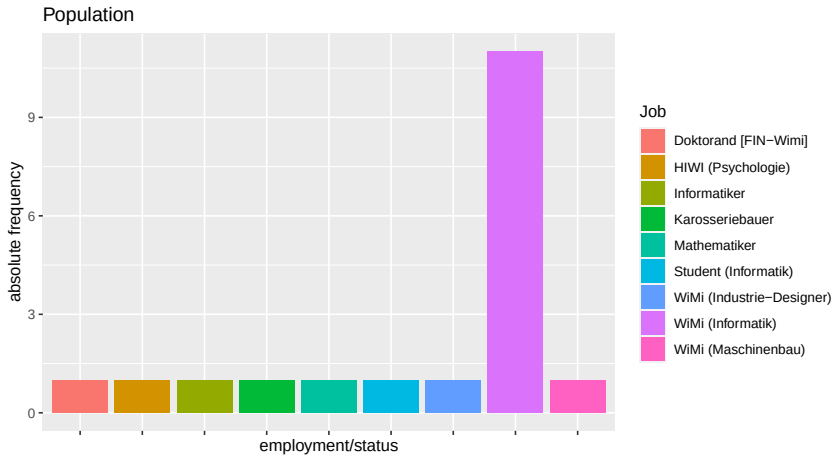


(A) Participants sex

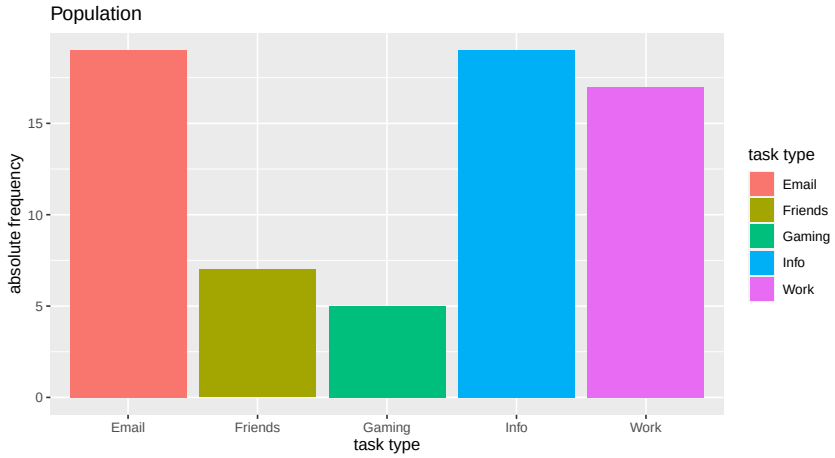


(B) Participants age

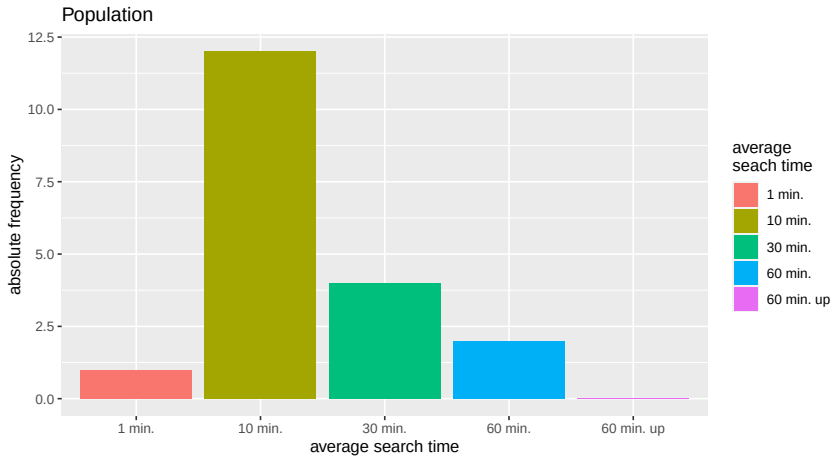
FIGURE 6.2: Participant metadata is illustrated in respect to their biological aspects, e.g. sex (up) & age (below)



(A) Participants job assignments



(B) Participants Internet activity



(C) Participants average search time

FIGURE 6.3: Participant metadata is illustrated in respect to their search familiarity, e.g. by their job assignment (up), their Internet activities (middle) & their reported average search time (below).

6.3 Logger & Eye-Tracker

A complex multi-modal recording set-up was designed to further analyze participants search activities triggered by search tasks. During search sessions, participant interactions with the browser are stored in log-files and their individual gaze is analyzed by an eye-tracker.

The experiment was carried out via the *Firefox* browser and the *Google* search engine. Interactions were recorded using a logger developed in the Data & Knowledge Engineering group as a browser add-on. It enables recording of search engine interactions in form of clicks, scrolling, durations, the search engine result page (SERP) number and tab & window activation. Search tasks were presented as a quiz which was placed & fixed in the first tab of the browser. The interface of the quiz tab is illustrated in Fig. 6.4. For each task, participants were asked to identify if they already knew the answer in case of *Fact-Finding* tasks or how much expertise they had in case of an *Exploratory* task. Users had to search for the answer even if they knew the answer. It was explained that the quality/correctness of the answer is a priority over the number of tasks solved. The limit of maximum twelve Fact-Finding tasks were not mentioned. It was assumed this would result in a more natural search behavior. Once a participant submitted an answer, they got the next question and could not correct the answer later. After each participant, the browser history was deleted to avoid highlighting of previously clicked search results and personalization.

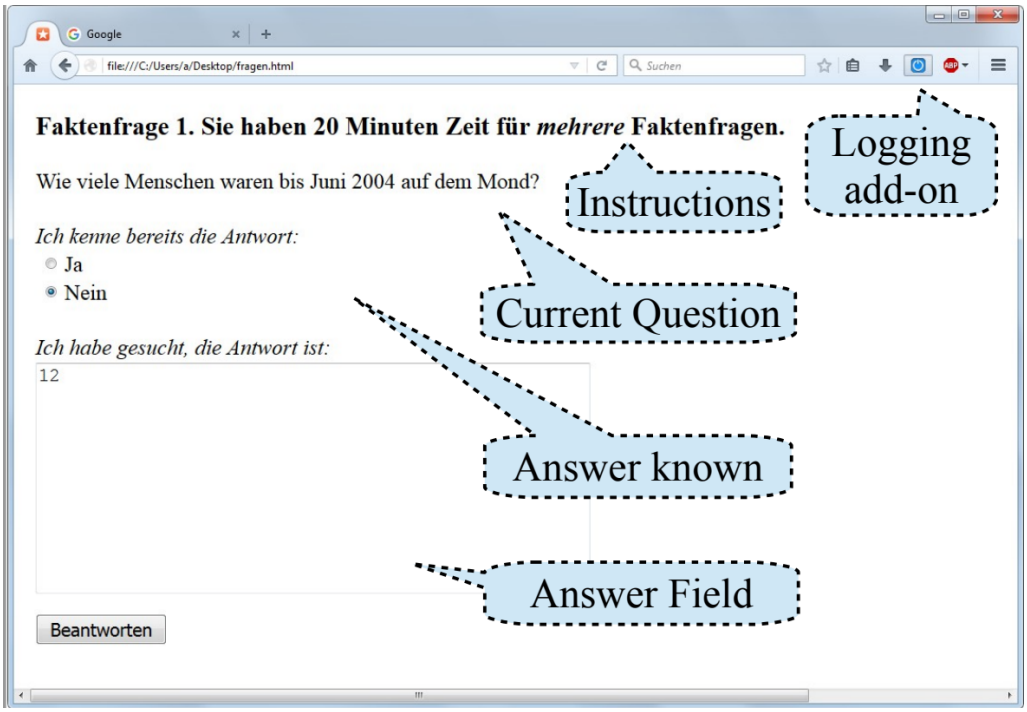


FIGURE 6.4: Taken from [68]: The design menu that was shown to participants to trigger search sessions. The text marked in bold provided general instructions for the search task block and its time limitation. The subsequent line provides the current search task. In case of a Fact-Finding task, participants were asked if they already knew the answer to the task. Below was the answer box to write into. The button below finalizes the search task.

The eye-tracker in use was the *Tobii X2-60* with the software *Tobii-Studio* (ver. 3.4.2.). It allows a 60Hz data-sampling rate. The monitor comprises 24″ with 1920x1200 px and *USB-CAM-152H* from *Phytec* with 1280x960 px is used to record the participants during their search sessions. Eye gaze analysis by the eye-tracker was done via default settings, such as the fixation filter *I-VT filter* (Velocity-Threshold Identification (I-VT) fixation classification algorithm) [1].

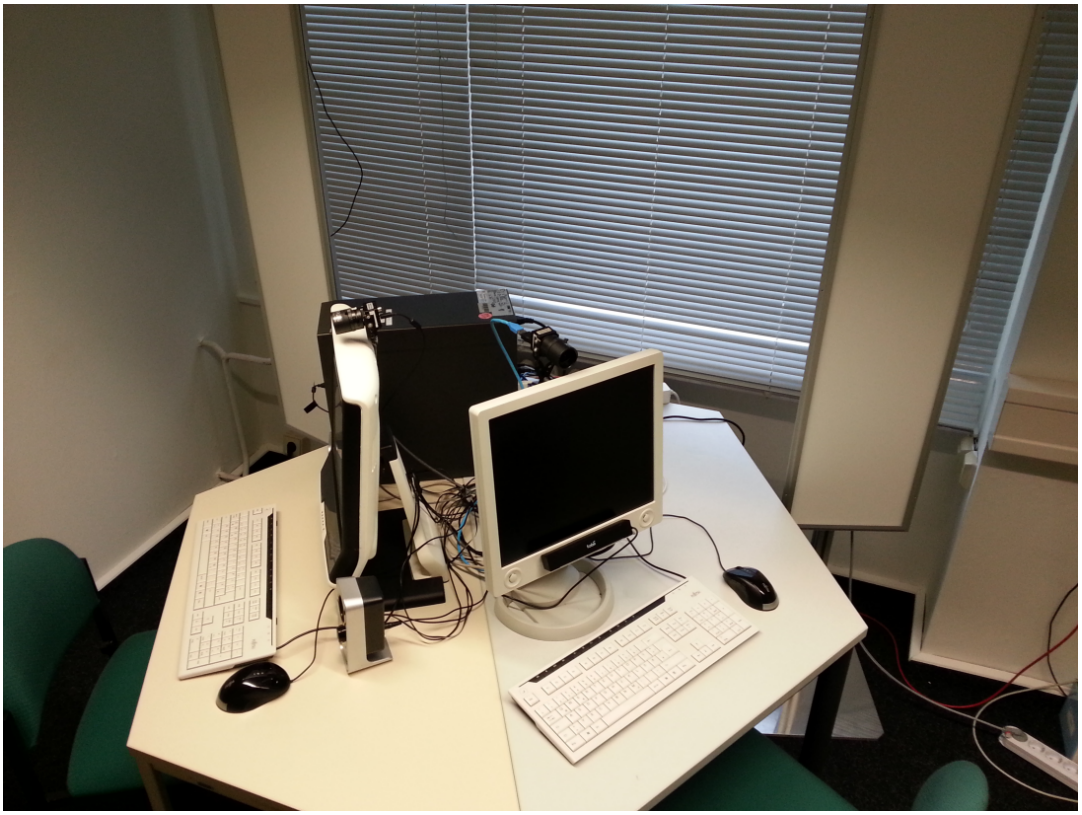


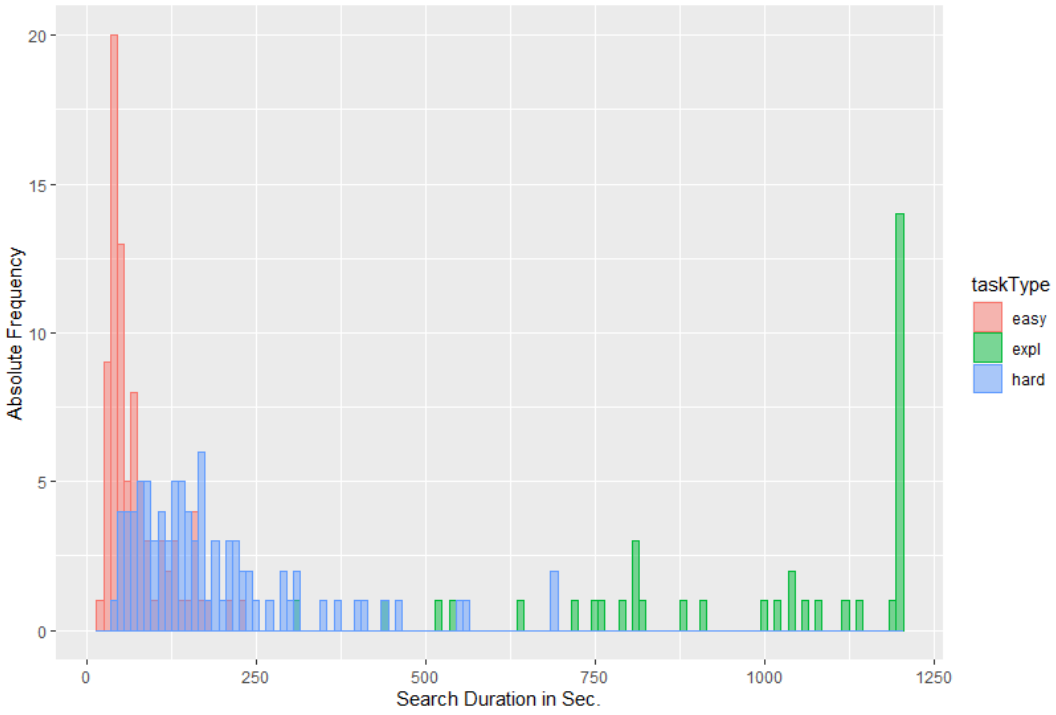
FIGURE 6.5: Taken from [70]: Set-up of the lab study. A camera recorded the participants while performing their search sessions. The eye-tracker mounted on the computer screen records and analyze the participant's eye gaze.

Chapter 7

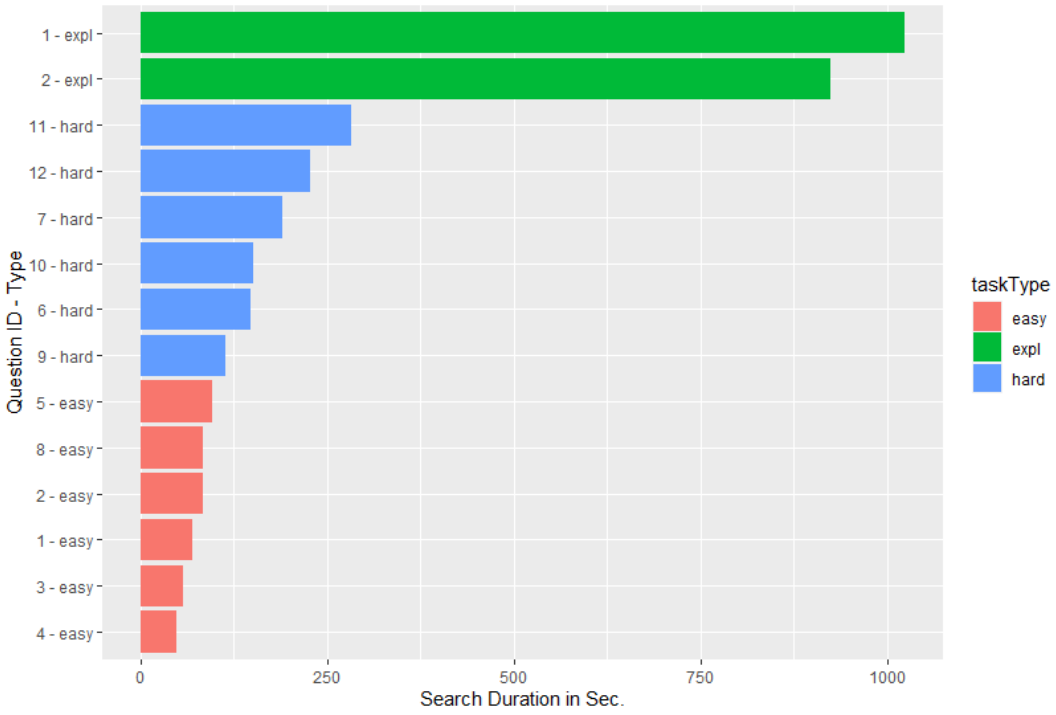
User Study - Analysis

7.1 Search Sessions

The experimental design and the recorded data set described in Sec. 6 needs to be verifying for a certain level of plausibility before further analysis can be applied. In total 171 search tasks, which should trigger *Fact-Finding* search activities, have been recorded. The tasks were recorded in blocks of consecutive Fact-Finding tasks, resulting in 19 *multitasking searches*. To incorporate different levels of complexity, half of these Fact-Finding tasks were designed to be of *easy* and *hard* difficultly. This resulted in recordings of 83 easy and 88 hard search sessions. Participants spent on average 128 seconds (min = 18, max = 693, median = 93) to answer the questions. It took users on average 2.6 times longer to answer hard tasks (min = 43, max = 693, median = 147) than easy ones (min = 18, max = 229, median = 51). Participants answered on average 8.8 factual questions (min = 4, max = 12) within the time restrictions of 20 minutes. In total, 80% of the answers were correct. Participants made more errors answering hard tasks (28.6% errors) than easy ones (10% errors). The gradient between easy & hard in respect to time duration and error rate imply that the experimental design is valid for difficulty assignments. In addition, 38 sessions were recorded, assigned to tasks that should trigger *Exploratory* search activities. Participants spend on average 16 min. (min = 5.21, max = 20, median = 17.38) to finalize their exploration. With Exploratory search tasks being of a rather *open* search goal, the analysis of being correct is conceptually not possible or at least misleading. Fig. 7.1 illustrates the distribution of time durations for each task individually in Fig. 7.1a and averaged participant-wise for each task in Fig. 7.1b. As the time durations follow a clear trend for task assignments, it can be concluded that the given search tasks might have triggered the intended search activity. Even though, the described statistics are no indicator for being 100% correct, this surface level analysis justifies a certain level of plausibility in the experimental design. The task types described in Sec. 6.1 have been carefully designed and mentioned statistics are fully in-line with the expectation about search sessions properties. Therefore, the recorded data set can be considered as reasonable trustworthy to further proceed with a more in-depth analysis.



(A) Time Duration of Search Sessions & Task Types



(B) Mean Search Session Durations by Task

FIGURE 7.1: Time duration of search sessions for all tasks (top) and mean session duration per task (below). Coloring encoding indicate the task assignments for Exploratory and Fact-Finding tasks (easy & hard).

7.2 Navigational & Interaction Model

To gain understanding of the search behavior of users, a mathematical & computational *(User) Behavior Model* is aimed to be build that explicitly models aspects of navigational *strategies* and *actions* that users execute during their online search. Within the following sections, this model will increase its complexity to eventually implement the *Information Search Behavior Profile Model* in Fig. 5.2. Before starting any analysis, it is necessary to define the task at hand. In the case of data analysis, this decomposes into the definition of the data and the applied models. This section (and all following) will start with **Data Definition** & **Model Definition**. Even though, the *User Study - Design* does not change, each section will just define the specific aspects needed for the individual task. This task description starts with very few and simple concepts. Following sections will incrementally add new aspects of the data set in use. This section can be seen as a heavily reduced summary of my research work in *Exploration or Fact-Finding: Inferring User's Search Activity Just in Time* [68]. The aim of this research work is directly motivated by **RQ2**, and initial findings for **RQ1** will be reported.

7.2.1 Task Description

A first model candidate will be introduced that should be considered as the baseline approach. This baseline model will introduce basic models capable of representing aspects of navigation and interaction through & with web pages during online searches. For that, simple time-series models will realize the *navigational model* and standard probability distributions will realize the *interaction model*.

7.2.1.1 Data Definition

User search sessions can be thought of as a sequence of interactions with a search engine and a web browser. These sequences consist of navigational pattern within the online search. Each element within this sequence can be represented as a discrete state, which will be described by the following:

- *Main* (M):
  A searcher is viewing the menu/quiz tab of the experimental design. Users either read and/or answer the presented search task. At any time, a user can revisit the menu/quiz tab for re-reading or to make notes during the search session.
- *Query* (Q):
  A searcher is formulating a search query on *Google*'s search engine by entering the query or using auto-complete suggestions.
- *SERP* (S):
  A searcher is examining *Google*'s search engine result page (SERP). Usually *SERP* occurs directly after *Query*, but it is also accessible by changing the active tab.
- *Page* (P):
  A searcher is inspecting a web page not categorized by the previous states.

Hence, the navigational aspect of the data set $\underline{\boldsymbol{s}}$ consists of N search sessions indexed by $1 \leq n \leq N$ via $\boldsymbol{s}_n$ comprising a search session of length L_n. Each element at position $1 \leq l \leq L_n$ in session n is defined as the visited state s_{nl} from the *state space* $\{M, Q, S, P\}$.

Further on, each state consists of a set of actions, a user executes on it. In this work, just two simple actions derived from log-files are taken into account:

- *State.Duration*:
  A searcher is visiting the state in a specific time interval measured in seconds.
- *State.Scrolling*:
  A searcher's accumulated scrolling time within a state, measured in milliseconds.

Hence, the action aspect of the data set $\underline{\underline{\boldsymbol{x}}}$ is associated to N search sessions indexed by $1 \leq n \leq N$ via $\underline{\boldsymbol{x}}_n$ up to the search session length L_n. Each vector at position $1 \leq l \leq L_n$ in session n is defined as a set of actions $\boldsymbol{x}_{nl}$. The set comprises a size of A many actions indexed by $1 \leq a \leq A$ via x_{nla}. With x_{nla} representing measurements of time, its values are positive real-valued.

Each online search is triggered by a particular *Information Need*. In the described experimental design, this need is triggered by search tasks on *Main*. The search tasks are grouped into two categories:

- *Fact-Finding* (*Fact*):
  Search Tasks that can be assumed to be *closed*. Search sessions, trigger by that tasks, are assumed to be a *Fact-Finding* search activity.
- *Exploratory* (*Expl*):
  Search tasks that can be assumed to be *open*. Search sessions, triggered by that tasks, are assumed to be an *Exploratory* search activity.

7.2.1.2 Model Definition

Abstractly, search sessions are assumed to be *independent* because of the missing interaction between participants. Further, the *behavior* within a specific search activity is assumed to be *identical* for all participants. This overall assumption about the *independent and identical distribution* for all task blocks will be justified by the reasoning that a user can not really gain information within a task block and exploit this in another one. Further on, the user's search history was regularly cleared, so search meta information could not be exploited by the search engine. Additionally, the experiment was conducted in a reasonable amount of time which limits effects of fatigue, developmental changes etc.

In respect to the precise model formulation, the *navigational model* will be introduced in the following. The sequence of states within an individual search session forms a joint probability distribution, which can be transformed without loss of generality via the *product rule/chain rule* in the following:

$$\begin{aligned} P(\underline{\boldsymbol{s}}|\boldsymbol{\theta}) &= \prod_{n=1}^{N} P(\boldsymbol{s}_n|\boldsymbol{\theta}) \\ &= \prod_{n=1}^{N} P(s_{n1}, ...s_{nL_n}|\boldsymbol{\theta}) \\ &= \prod_{n=1}^{N} P(s_{n1}|\boldsymbol{\theta}) \cdot \prod_{l=2}^{L_n} P(s_{nl}|s_{n1}, ...s_{nl-1}, \boldsymbol{\theta}) \end{aligned}$$

Therefore, the entire joint probability distribution comprises an amount of parameters linear in the magnitude of the sequence length. With a data set of limited size, the entire representation of this distribution seems quite exhaustive. A user will execute navigational patterns within the search session, but it seems reasonable to assume that such navigational pattern will only use a window of limited memory (of the user). *Markov Models* will be used to approximate the entire distribution by exploiting the Markov property (marked with *):

$$\begin{aligned} P(\underline{\boldsymbol{s}}|\boldsymbol{\theta}) &= \prod_{n=1}^{N} P(s_{n1}|\boldsymbol{\theta}) \cdot \prod_{l=2}^{L} P(s_{nl}|s_{n1}, ...s_{nL_n}, \boldsymbol{\theta}) \\ &=_* \prod_{n=1}^{N} P(s_{n1}|\boldsymbol{\theta}) \cdot \prod_{l=2}^{L} P(s_{nl}|s_{nl-1}, \boldsymbol{\theta}) \end{aligned}$$

$P(s_{n1}|\boldsymbol{\theta})$ is the *start probability* of state s_{n1} and $P(s_{nl}|s_{nl-1}\boldsymbol{\theta})$ is the *transition probability* from state s_{nl-1} to state s_{nl}. The amount of approximating errors can be decreased by an increase of the amount of transitions to be considered in the transition probabilities. This context will be referred to as the *order* of the Markov chain, e.g. by *n-th order* Markov Models. In addition to the state sequence, the data set comprises a set of *actions* a user implements on that state, e.g. by *State.Duration* & *State.Scrolling*. All actions form a sequence of features $\underline{\boldsymbol{x}}_n$ parallel to the state sequence $\boldsymbol{s}_n$, resulting in the entire data set $\underline{\boldsymbol{z}}$. The set of actions is assumed to be strictly associated to the state only. Any interdependency between actions is neglected, and the baseline model is described as follows:

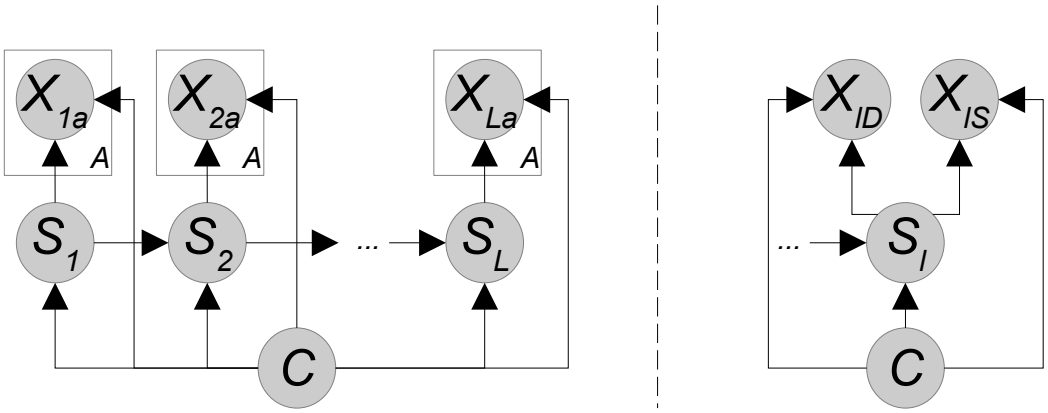


FIGURE 7.2: The Graphical Model of the proposed approach is structurally equivalent to *Hidden Markov Models* [99]. A navigational model realizes the transition probabilities through the states $\{S_l\}$ and an interaction model the emission probabilities for actions: State.Duration $\{X_{lD}\}$ & State.Scrolling $\{X_{lS}\}$. The Naive Bayes assumption is used in the interaction model, and the entire model is conditioned on the search activity C.

$$
\begin{aligned}
P(\underline{\boldsymbol{z}} = (\underline{\boldsymbol{s}}, \underline{\boldsymbol{x}})|\boldsymbol{\theta}) &= \prod_{n=1}^{N} P(\boldsymbol{z}_n = (\boldsymbol{s}_n, \underline{\boldsymbol{x}}_n)|\boldsymbol{\theta}) \\
&= \prod_{n=1}^{N} P(s_{n1}|\boldsymbol{\theta}) \cdot \prod_{l=2}^{L_n} P(s_{nl}|s_{n(l-1)}, \boldsymbol{\theta}) \cdot \prod_{l'=1}^{L_n} P(\boldsymbol{x}_{nl'}|s_{nl'}, \boldsymbol{\theta}) \\
&= \prod_{n=1}^{N} P(s_{n1}|\boldsymbol{\theta}) \cdot \prod_{l=2}^{L_n} P(s_{nl}|s_{n(l-1)}, \boldsymbol{\theta}) \cdot \prod_{l'=1}^{L_n} \prod_{a=1}^{A} P(x_{nl'a}|s_{nl'}, \boldsymbol{\theta})
\end{aligned}
$$

The *navigational model* is now combined with an *interaction model*, and the resulting Likelihood resembles the one of *Hidden Markov Models* [99]. Therefore, *Generative Classifiers* $\boldsymbol{\theta}_c$ can be trained for $c \in \{Expl, Fact\}$. By using the *Bayes Theorem*, inference about search activities can be done via the *Maximum A Posteriori Prediction*:

$$
P(\boldsymbol{\theta}_c|\boldsymbol{z}_n) = \frac{P(\boldsymbol{z}_n|\boldsymbol{\theta}_c) \cdot P(\boldsymbol{\theta}_c)}{P(\boldsymbol{z}_n)} = \frac{P(\boldsymbol{z}_n|\boldsymbol{\theta}_c) \cdot P(\boldsymbol{\theta}_c)}{\sum_{c'} P(\boldsymbol{z}_n|\boldsymbol{\theta}_{c'}) \cdot P(\boldsymbol{\theta}_{c'})}
$$

$P(\boldsymbol{z}_n|\boldsymbol{\theta}_c)$ denotes the Likelihood of the proposed model given the search activity c and $P(\boldsymbol{\theta}_c)$ the *prior* associated to that activity $\boldsymbol{\theta}_c$. The factorization of the Likelihood is illustrated as its *Graphical Model* [88] in Fig. 7.2.

7.2.2 Towards a Baseline Model

A reasonable baseline model will be created to be compared against during later approaches. In case of the *navigational model*, first-order Markov Models are suitable instances because they are the least complex models that still consider context. In case of the *interaction model*, the set of baseline candidates remain less trivial. There is a huge amount of statistical distributions that could model *State.Duration* and *State.Scrolling*. Both features can only be positive real-valued measurements. A one parameter model (low complexity) that works on that *support* is the *Exponential Distribution*. Fig. 7.3 illustrates the fit for *State.Duration*. Overall, this easy distribution provided a surprisingly good fit. Further on, it can be seen that state specific distributions are a necessary choice. With such suitable proposal functions fitted, reasonable baseline models can be generated. The first baseline model classifies only by using the *navigational model*, while a second baseline extends the first with an *interaction model* purely on *State.Scrolling* and a third purely on *State.Duration*. The results of this approach can be seen in Tab. 7.1, which represents the *Confusion Matrix* for the model prediction. Performance measurements were derived by a 5-fold *Cross-Validation* [125] averaged over 2,000 repeats. The *navigational model* alone (marked as *None*) resulted in an *Accuracy* of 73.6%. The second approach of combining the *navigational model* with an *interaction model* on *State.Scrolling* resulted in a slight increase of accuracy, up to 75.4% (marked as *Scrolling*). The third approach of using *State.Duration* drastically increased the accuracy to 89.4% (marked as *Duration*). It can be stated, that the *navigational model* already provide meaningful information about search activity classification, but suitable models for additional actions boost the predictive performance. The combination of both feature in the *interaction model*, on the other hand, resulted in a drop of performance. A possible reason for this observation is the Naive Bayes assumption on the feature. Violations of this assumption lead to the decrease, and future development should consider meaningful feature selection because of this.

Design	Prediction None		Scrolling		Duration	
	Fact	*Expl*	*Fact*	*Expl*	*Fact*	*Expl*
Fact	13	6	13	6	16	3
Expl	9	29	8	30	3	35

TABLE 7.1: Adapted from [68]: *Confusion Matrix* of the baseline models: None, Scrolling, Duration. Evaluation was done by averaging results of 2,000 repeated 5-fold *Cross-Validation* [125]. For visual clarity, the nearest integer values are reported.

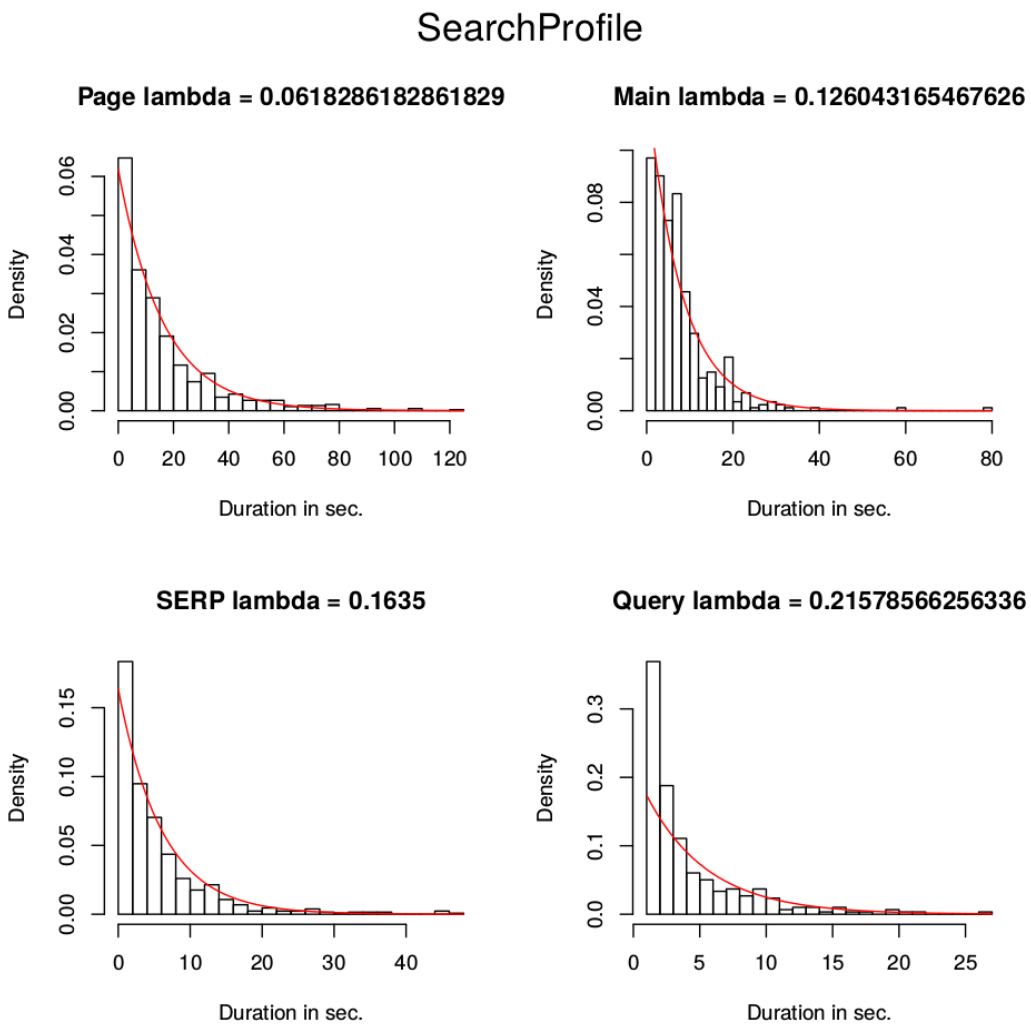


FIGURE 7.3: Taken from [68]: Histogram of *State.Duration* for Fact-Finding search activities. Red lines represent the fit of an *Exponential Distribution*.

7.2.3 Fine-Tuning the Model

The described baseline can be improved by many factors. The *navigational model* was implemented with first-order Markov Models. The order can be increased to consider a broader context in the navigational trails. This has the potential for increased predictive performance, but also increase the risk of over-fitting by the increase of the parameter space. Fig. 7.4 illustrates the accuracy progression of different orders. For that, a comparison of two prominent estimation techniques are provided, e.g. *Maximum Likelihood* (ML) [42] and *Maximum A Posteriori* (MAP) estimation. In case of ML estimates, the increasing order results in a decrease of the predictive performance. In contrast, such an effect can not be observed in case of the MAP estimates. The MAP shows a clear plateau in case of order 2 & 3, with the first being the best on a 92.1% accuracy. ML estimates are known for their tendency to overfit. In contrast, MAP estimates can be seen as a regularization approach that stabilizes the estimate and therefore are less prone to overfitting tendencies.

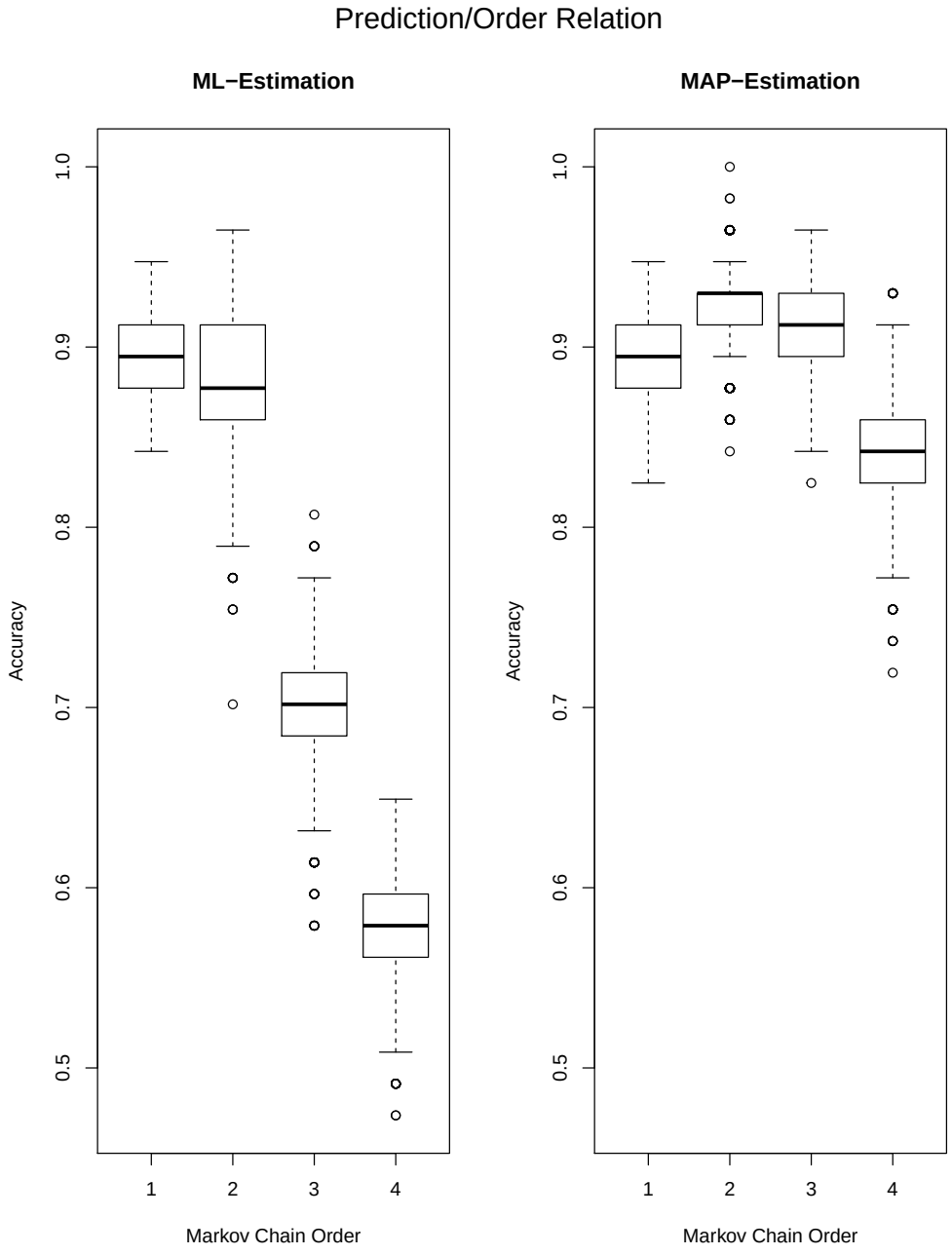


FIGURE 7.4: Taken from [68]: *Accuracy* progression for increasing order of the navigational model. Left: performance of a model learned by Maximum Likelihood (ML) Estimation [42]. Right: performance learned by Maximum A Posteriori (MAP) Estimation. Evaluation was done by averaging a 5-fold *Cross-Validation* [125] over 2,000 repeats.

7.2.4 Addressing Experimental Limitations

It was decided to enable the user to read and process all tasks digitally via the web browser in the question tab (*Main*: M) during the user study. This decision has some advantages. First, a user does not need to switch between different media, e.g. sheet of paper and screen. Second, a user can copy & paste text fragments directly in a query field (*Query*: Q). Third, users can copy & paste information from web pages (*Page*: P) directly to the answer field in the quiz tab. Forth, a user is always focused on the screen, which will be important in subsequent eye-tracking studies. Nevertheless, M is an artifact of the user study and influences natural behavior with the search system. Thus, M was erased from all search sessions to train models on the remaining data set. If M was between two different states, M and its associated feature could be erased easily, e.g. $(P - M - S \mapsto P - S)$. If M was between states of the same kind, M was erased as well but feature for the remaining state were accumulated when the state was from the same instance, e.g. $(P - M - P \mapsto P)$. In case, M separated different instances from the same state, feature accumulation was not needed, e.g. $(P - M - P' \mapsto P - P')$

7.2.5 First Model & Initial Findings

After resolving for experimental artifacts, models are trained again on the remaining data and its results are depicted in Tab. 7.2. The accuracy drops from 92.1% to 87.7% but now represents more realistic estimates for natural search behavior. Nonetheless, the predictive performance seems reasonable enough to draw some initial conclusion about characteristics derived from the model. In case of the *interaction model*, it can be stated that the profile of *actions* on web pages is indicative for search activities. The relevance of features follows the same ordering as in Sec. 7.2.2, namely $None \leq Scrolling \leq Duration$. In case of the *navigational model*, it provides some indicative information for search activities. By exploiting the properties of the Markov chains in the *navigational model*, insights about the navigational trails of the user can be derived. Fig. 7.5 illustrates the *stationary distribution* for each search activity in a comparison. This distribution represents the relative proportion of each state in a long-run behavior. In general, users implement different navigational *strategies* during different search activities. In $Expl$ search activities, users focus the search heavily towards *Page*. Only in a few cases, *Query* will be implemented. *SERP*s will be inspected more often than *Query* is executed, but the major focus in the search lies on the detailed inspection of *Page*. The overall profile of the stationary distribution is heavily skewed. In *Fact* search activities, the overall profile is less skewed. Even though, *Page* is still the most dominant navigational state during the search, *SERP* and *Query* are way more often implemented. This indicates that in case of *Fact* search activities, users are able to re-formulate queries more often by inspecting the provided search engine results than they are able in $Expl$ search activities.

Design	Prediction	
	Fact	*Expl*
Fact	17	2
Expl	5	33

TABLE 7.2: Adapted from [68]: *Confusion Matrix* for the interaction & navigational model. Evaluation was done by averaging results of a 2,000 repeated 5-fold *Cross-Validation* [125]. For visual clarity, the nearest integer values are reported.

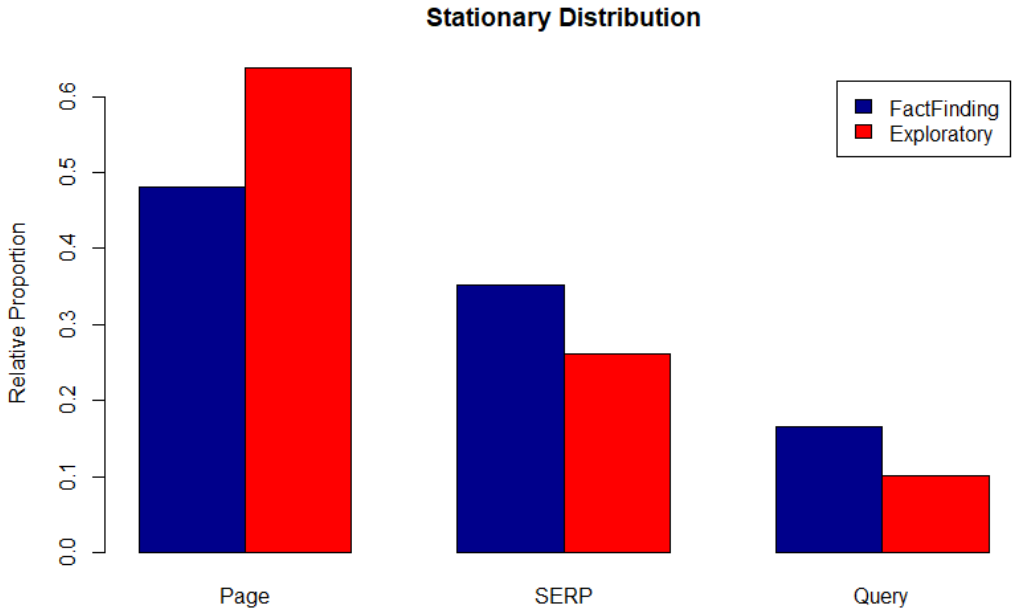


FIGURE 7.5: Characteristics of the navigational model for Exploratory (red) and Fact-Finding (blue) search activities derived from the stationary distribution of the *Markov Models*.

7.2.6 Conclusion

An initial and simplistic *(User) Behavior Model* was proposed to identify search activities in online search sessions. The proposed model resembles a classical and well researched model, namely a Hidden Markov Model (HMM). Within the interpretation of the given task, such HMMs naturally comprise a *navigational model* in form of their transition probabilities and an *interaction model* in form of their emission probabilities. The proposed model is able to differentiate between Exploratory and Fact-Finding search activities with a reasonable accuracy of 87.7%. The parametric form of the HMM provides possibilities to draw some limited conclusion from the model about characteristics of search activities. Within search sessions, users follow navigational *strategies*. The initial analysis indicate that second-order Markov Models are suitable approximations to evaluate these strategies. Exploratory search activities are heavily focused towards the analysis of individual web pages, while Fact-Finding search activities show an increased orientation towards querying and the inspection of search engine results. The profile of *actions* that a user implements during the search is indicative of their search activity. Not all of these actions are equally indicative for search activities, and the dwell-time on a web page remains the most indicative one. A systematic way of feature selection will be needed to construct more sophisticated models. The described brute-force approach for combining features resulted in a decrease of predictive performance.

7.3 Combining Eye Tracking and Navigation

The proposed model of the previous section showed some promising results. Nonetheless, its construction was rather ad-hoc and the model was restricted by a very limited set of *actions* a user could execute during a search. Therefore, this section aims to build upon the previous model to create a more promising *(User) Behavior Model* and derive further insights about search behavior. In this section, a broader field of user *actions* will be considered, while some of them are derived from log-files and others from eye-tracking data. By combining both data sources, the model evolves further towards the *Information Search Behavior Profile Model* in Fig. 5.2. The following section can be seen as a reduced summary of my research work in *Inferring User's Search Activity Using Interaction Logs and Gaze Data* [116] with minor adaptions. The aim of this research work is directly motivated by **RQ1** & **RQ2**.

7.3.1 Task Description

To build upon the previous model, several challenges need to be addressed. First, combining data from different sources is a challenging task in its own rights. Second, a meaningful combination of features from different sources needs to be identified via a systematic approach for feature selection. Third, a meaningful model selection needs to be implemented, so an advanced model can be compared to baselines. Finally, the model needs to be capable to draw conclusions from it and derive information about user search activities.

7.3.1.1 Data Definition

The *Data Definition* in Sec. 7.2 remains valid, but will be extended for an additional modality in respect to the set of *actions*. This work will focus on the following, while the former three are derived from log-files and the last three from eye-tracking data:

- *State.Duration* (Dur):
  A searcher is visiting the state in a specific time interval measured in seconds.
- *State.Scrolling* (Scroll):
  A searcher's accumulated scrolling time within a state, measured in milliseconds.
- *State.Clicking* (Click):
  The sum of clicking events of a searcher during the state.
- *State.Fixation-Count* (Fix):
  The occurrence of fixations during the visit of the state.
- *State.Fixation-Duration* (FixDur):
  A searcher's accumulated duration of fixations during the state in milliseconds.
- *State.Mean-Fixation-Duration* (FixDurMean):
  A searcher's mean duration of fixations while visiting the state.

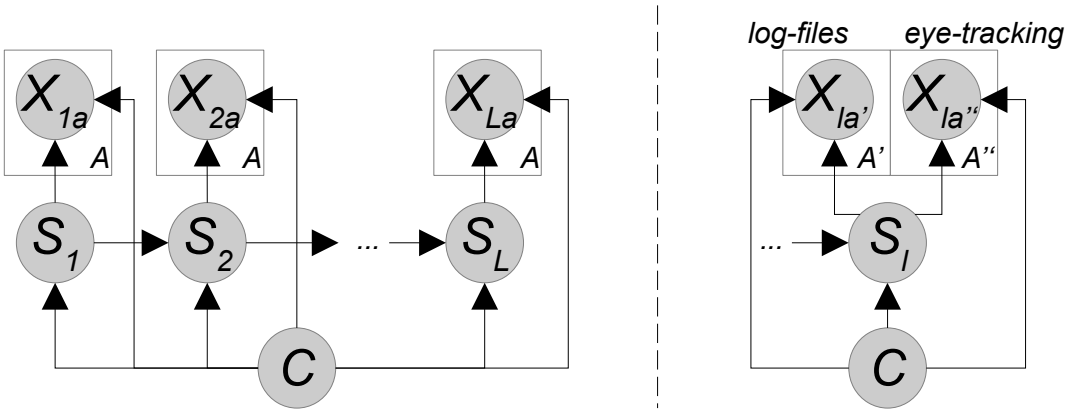


FIGURE 7.6: The Graphical Model of the proposed approach is structurally equivalent to *Hidden Markov Models* [99]. A navigational model realizes the transition probabilities through the states $\{S_l\}$ and an interaction model the emission probabilities for actions derived from log-files $\{X_{la'}\}$ & eye-tracking data $\{X_{la''}\}$. The Naive Bayes assumption is used for both modalities, and the entire model is conditioned on the search activity C.

7.3.1.2 Model Definition

The *Model Definition* of the *Navigational & Interaction Model* remains equal, but the Likelihood will be extended for the new feature by simply increasing the product over that feature. The *Graphical Model* [88] in Fig. 7.6 can easily visualize the mentioned extension.

7.3.2 Model Selection

Model selection relates to the model in respect to selecting the order of the *navigational model*. While *Fine-Tuning the Model* in Sec. 7.2.3 already addressed this issue, it should rather be considered as an ad-hoc approach. Within this section, a more theoretically founded approach will be implemented that also considers more possible combinations of selected models. The factorization of the Likelihood of the proposed model clearly shows that even though the *interaction models* are dependent to the *navigational model*, the chain in the navigation is not affected by the *interaction model*. Therefore, selecting the best model reduces to selecting the order of the Markov Models. In the following, two well studied model selection approaches are used, namely the *Akaike Information Criterion* [4] & the *Bayesian Information Criterion* [112]. To adequately model characteristics in the navigational *strategy*, it is necessary to limit the potentially infinite range of candidate orders. In this setting, a range from 1 up to 4 appeared to be sufficient, as it can be seen in Tab. 7.3. Both selection criteria show their minimal values (indicating the best model) on order 2 for $Expl$ and $Fact$ search activities. This finding complements the results of the previous approach in Sec. 7.2.3, based solely on the predictive performance as a measurement for model selection, see Fig. 7.4. In all settings, the second-best model has a noticeable difference of $\Delta_i > 10$ to the best model, implying that alternatives have essentially no support according [20]. This can be seen as an indicator that patterns of 3 consecutive trails (a second-order Markov Model) suffices to capture main navigational characteristics in search activities. This further implies an equally complex structure in the navigational *strategy* of both activities.

Markov Chain Order	AIC AIC_{Fact} / AIC_{Expl}	BIC BIC_{Fact} / BIC_{Expl}
1	5042.78 / 3995.37	5046.95 / 4009.94
2	**4708.33 / 3920.90**	**4725.84 / 3982.07**
3	5033.79 / 4313.11	5104.62 / 4560.70
4	6774.96 / 6064.45	7059.12 / 7057.70

TABLE 7.3: Adapted from [116]: Model selection using the *Akaike Information Criterion* (AIC) [4] & the *Bayesian Information Criterion* (BIC) [112] applied to the navigational model for different orders in Fact-Finding (Fact) and Exploratory (Expl) search activities. Selection of the best model instances are marked in bold.

7.3.3 Feature Selection

Feature selection is the task to choose relevant features from a data set. On one side, relevant could relate to the usefulness in prediction settings. On the other side, it could relate to features that facilitate data understanding. Hence, it is necessary to decide exactly what it means for a feature to be relevant [90]. In the proposed setting features originate from different sources, log-file and eye-tracker data. Initial analysis *Towards a Baseline Model* in Sec. 7.2.2 indicated that simply combining all features into a model might even down-grade its performance. Therefore, a structured approach to identify feature combinations is needed. In the following the terminology is used that a feature is *relevant* if it holds a certain significance threshold for a *goodness-of-fit* test, see *Statistical Hypothesis Testing*, and a feature is *useful* if it increases predictive performance.

7.3.3.1 Feature Selection via Filtering

The features at hand are positive real-valued measurements in two classes. Without any further assumption about the data distribution, the two sample *Kolmogorov-Smirnov-Test* [62] can be applied to check the *null hypothesis* H_0 that both samples are representations of the same distribution. It can be assumed that a feature can only hold limited usefulness under H_0. Each attribute can now be associated to a *p-value* and feature selection can be done via thresholds on it. Tab. 7.4 illustrates the complete feature set and their associated p-values.

Feature	P-Value	Feature	P-Value	Feature	P-Value
***P*.Dur**	**0.0013**	*P*.Fix	0.0575	*P*.Scroll	0.4123
M.Dur	0.0720	***M*.Fix**	**3.2e-05**	*M*.Scroll	0.4487
S.Dur	0.9533	*S*.Fix	0.9956	*S*.Scroll	N/A
Q.Dur	0.2743	*Q*.Fix	0.9973	*Q*.Scroll	N/A
***P*.FixDur**	**1.7e-06**	*P*.FixDurMean	0.6262	*P*.Click	1
***M*.FixDur**	**3.1e-06**	*M*.FixDurMean	0.9054	*M*.Click	0.9525
S.FixDur	0.5787	*S*.FixDurMean	0.1281	*S*.Click	N/A
Q.FixDur	0.3647	*Q*.FixDurMean	0.6664	*Q*.Click	N/A

TABLE 7.4: Adapted from [116]: Overview of the statistical significance [1] of multiple feature derived from the data of the user study. P-values were calculated by the *Kolmogorov-Smirnov-Test* [62] and N/A indicates no data measurements. The naming convention of features were introduced in *Data Definition* Sec. 7.2 & *Data Definition* Sec. 7.3.

In accordance to previous findings, the p-values indicate that not all *actions* during the search are equally indicative of search activities. Features such as *State.Scroll* & *State.Click* comprise no reasonable differences between search activities on all states. This indicates that these *actions* are so fundamental in the search process that adaption in behavior cannot be realized within the search. It can be assumed that both are such integral interactions with the computer that users can implement them just as they are, independent of their search activity. In contrast, *State.Dur*, *State.Fix* & *State.FixDur* are indicative with statistical significance [1]. Further on, even the indicative feature are not informative overall but only in association to a particular state. Especially *actions* on *Main* and *Page* are indicative, but not on *SERP* and *Query*. This implies a lack of support by *SERPs* in respect to individual search activities and that users struggling to formulate queries suitable for their $Expl$ search activity rely on the same mechanism as in case of $Fact$ search activities.

In respect to the profile of *actions* on *Page* (*P*), users spend considerable more time on web pages (*P.Dur*) and fixate them considerable more (measured by *P.FixDur*) in case of $Expl$ search activities. This indicates that processing of the presented content and extracting the relevant information from the web page comprise an increased cognitive load for the user in case of explorations. In case of dwell times in *Page*, an *Exponential Distribution* was fitted for both search activities, see Fig. 7.7. By analyzing the ratio between both activities, the shift between both modi can be located on 10 seconds. This implies that a user infers the usefulness of a web page in average under 10 seconds within a $Fact$ search activity, but takes longer than 10 second on average for $Expl$ search activities. The same ratios can be derived for the fixation activity, but for now it is sufficient to simply state an increased fixation activity in case of the $Expl$ search activity.

[1] A p-value is considered to be significant if it is less than a threshold α. Different communities use different values, but as an example, let $\alpha = 0.05$. Multiple testing leads to alpha error accumulation, deeming individual test results over-optimistic. A correction for that is implemented in the *Bonferroni Correction* (BC). To remain at the mentioned error level ($\alpha = 0.05$), the interpretation of statistical significance of p-values holds only true, if they are less than their adjusted threshold ($\alpha_{adjusted} = \alpha/n$ in case of the BC). In this example, $\alpha_{adjusted} = 0.05/20 = 0.0025$ (there are $n = 20$ tests because four features had no measurement). I am thankful that this was pointed out to me recently, so I have the chance to learn from my fault in [116] and to adapt Tab. 7.4 to the best of my current knowledge.

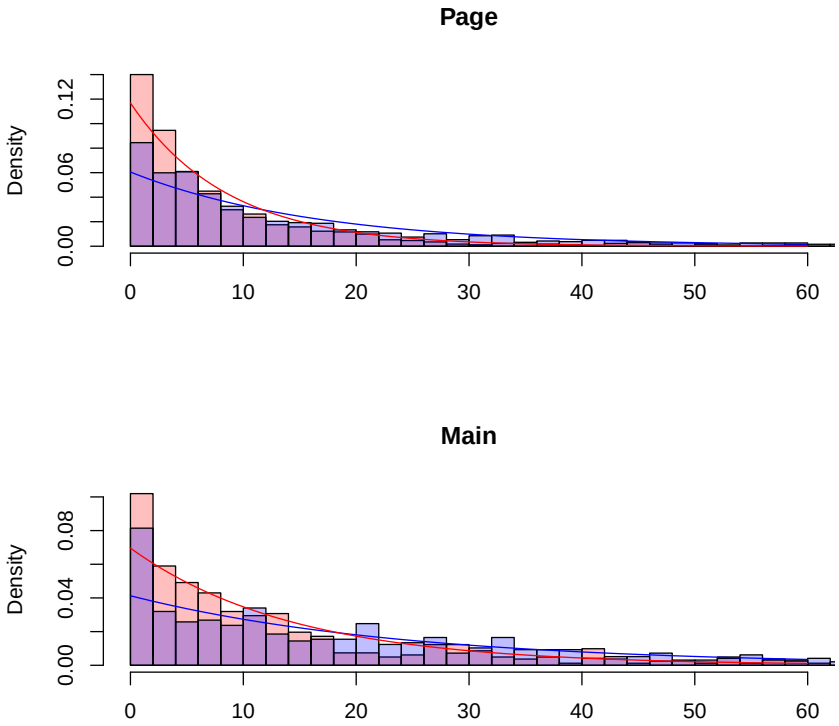


FIGURE 7.7: Adapted from [116]: Histogram of the feature State.Duration measured in seconds. Lines represents the fit of an *Exponential Distribution* for visual guidance. Color-Coding indicate Fact-Finding (red) and Exploratory (blue) search activities.

In respect to the profile of *actions* on *Main* (*M*) (the menu/quiz tab), users tend to spend less time (*M.Dur*) on it and fixate it considerable less (measured by *M.FixDur*) in case of *Fact* search activities. This indicates that these tasks might be easier to understand, but *Fact* tasks are also short in nature, and they can be read faster. Because copy&paste'ing was supported and encouraged, it can be argued that the time consumption for longer answers are negligible. Therefore, visiting *Main* was used to understand and clarify the exact task description and the time consumption for answering can be neglected. In its entirety, this implies that the cognitive load of users is increased in case of *Expl* tasks. In case of dwell times in *Main*, an Exponential distribution was fitted for both search activities, see Fig. 7.7. By analyzing the ratio between both activities, the shift between both modi can be located at 20 seconds. This implies that a user takes on average under 20 seconds to digest *Fact* tasks, but takes longer than 20 second on average for an *Expl* tasks. The same ratios can be derived for the fixation activity, but for now it is sufficient to simply state an increased fixation activity in case of *Expl* tasks.

7.3.3.2 Feature Selection via Wrappers

To find an adequate feature combination with good predictive performance, the construction of a classification model is necessary. For all as *relevant* selected features (up to the generously chosen p-value of 0.1), a probability distribution needs to be found that the classifier makes use of. With different normalizations, graphical inspections and goodness-of-fit tests, one can find candidate distributions, e.g. the Exponential distribution. After selecting the distributions for the classification model, one can use the predictive performance to identify a combination of *useful* features. In the filter approach, features were evaluated independently, and it is to be assumed that they are at least partially redundant and/or highly correlated. Therefore, the wrapper approach is used for selecting features. In its most general formulation, the wrapper methodology consists in using the prediction performance of a given learning machine to assess the relative usefulness of subsets of variables [50]. Measurements of performance are normally chosen by *Accuracy*, *Precision*, *Recall* etc. A 5-fold *Cross-Validation* [125] with 2,000 repeats was used to estimate such performance measurements. Tab. 7.5 illustrates the results of this approach. Using *State.Duration* as an *interaction model*, the best accuracy was achieved with 94.53%. In combination with *State.Fixation-Count* the best precision was achieved with a 93.10% accuracy. The best recall was observed with an accuracy of 92.12% with the combination of all features in the *interaction model*. The overall variance of the prediction was lowest in the *State.Duration* only model. Because of this fact, in combination with the highest predictive performance, this model was chosen as 'best'. It can be concluded that interaction of log-files and gaze data capture main aspects to discriminate search activities individually and when used in combination.

Design	Prediction			
	Fact	Expl	Fact	Expl
	Duration (D)		Fixation-Count (F)	
Fact	**17.199/0.479**	1.799/0.479	16.896/1.137	2.103/1.137
Expl	1.315/0.686	**36.685/0.686**	4.281/1.269	33.718/1.269
	Fixation-Duration (FD)		D + F	
Fact	16.931/1.120	2.068/1.120	17.189/0.634	1.810/0.634
Expl	4.831/1.403	33.168/1.403	**1.272/0.873**	36.728/0.873
	D + FD		F + FD	
Fact	17.152/0.620	1.847/0.620	17.117/0.975	1.882/0.974
Expl	1.349/0.982	36.651/0.982	5.159/1.139	32.839/1.139
	D + F + FD			
Fact	17.503/0.733	**1.497/0.732**		
Expl	2.207/1.303	35.792/1.303		

TABLE 7.5: Adapted from [116]: Comparison of different feature subsets. *Confusion Matrix* comprises values of a 2,000 times repeated 5-fold *Cross-Validation* [125]. Value pairs μ/σ represents the mean & standard deviation of the repeats.

7.3.4 Final Model

Some experimental limitations have already been addressed. *Main* is an artifact which was used to enable users to process tasks completely digital via the web browser without switching between different media (sheet of paper and screen). This results in less natural search behavior, and the approach for *Addressing Experimental Limitations* was described in Sec. 7.2.4. After resolving for experimental artifacts, the model is trained again and the accuracy drops from 94.53% to 89.32%. This was done to represent a more realistic estimate for natural search behavior. In comparison to the previous *Navigational & Interaction Model*, the more advanced model achieved nearly a 2% increase in accuracy by adequate model and feature selection. All in all, it can be argued that the model captures main characteristics of search activities. This statement can be justified by its reasonable predictive performance and plausible explanations about the search activities that can be deduced by the parametric form of the model itself.

An additional experimental limitation will be addressed in the following. The described experimental design comprises only two $Expl$ search tasks, and the generalizability of the model remains questionable. Therefore, a new sub design will be implemented to counter the expected criticism. One model is trained on ($Expl_1$, $Fact$) and tested on ($Expl_2$, $Fact$) (*approach A*) while another model is trained on exchanged $Expl_x$'s (*approach B*). If the model can learn on just one $Expl$ search activity but generalize its prediction for the other and vice versa, one could state its generalizability. Especially, if the performance values are comparable to the entire approach. Tab. 7.6 illustrates the results of the classification approach. The difference in accuracy varies just slightly, with 89.47% in *approach A* and 92.10% in *approach B*. Even though the data set is small, and the results only have limited statistical support, it can be argued that the model generalizes adequately over $Expl$ search activities.

Design	Prediction			
	$Fact$	$Expl_2$	$Fact$	$Expl_1$
$Fact$	15	4	18	1
$Expl_x$	0	19	2	17

TABLE 7.6: Adapted from [116]: *Confusion Matrix* comparison of the hold-out designs: approach A (left) and approach B (right).

7.3.5 Conclusion

A methodology and a model have been proposed as an instance of *(User) Behavior Models* with the aim to identify and characterize Exploratory and Fact-Finding search activities. To realize that goal, two different data sources, e.g. log-files and eye-tracking, were synchronized and merged into a broader analysis pipeline. *Model Selection* inferred an equally complex structure in the navigational *strategy* of both search activities, and the results confirmed initial findings in Sec. 7.2. By applying *Feature Selection*, a detailed inspection of the individual feature could be realized. First, not all *actions* within an online search are indicative of search activities. Especially, interactions such as clicking and scrolling comprises barely measurable differences between search activities. Statistical significant differences could be observed in the fixation activity of users and in dwell times on web page visits. It can be hypothesized that processing presented content on web pages and extracting the relevant information comprise an increased cognitive load for the user in case of explorations. Further, it can be argued that the cognitive load of users is increased when processing Exploratory tasks. Yet, the profile of *actions* comprises only significant differences on web pages but not on *SERPs* or during querying. This indicates a lack of support by *SERPs* in respect to individual search activities. Further, this implies, that users struggle to formulate queries suitable for their search activity, and they rely on the same mechanism. All in all, feature derived from log-files and eye-tracking data are useful for predicting search activities individually and in combination. The proposed model is capable to distinguish both activities with an accuracy of 89.32%. Therefore, it provides a reliable basis to gain insights into search activities of users. Nonetheless, it remains still a very simplistic approach that needs further extension to uncover further insights. Initial steps towards the *Information Search Behavior Profile Model*, see Fig. 5.2, have been made. Current choices for the *navigational model* and the *interaction model* result in an adequate predictive performance, but honestly lack a reasonable interpretation for what the user is actual doing on a web page besides 'being' and 'fixating'. A finer grained *interaction model* specifically designed for the evaluation of the eye movement will gain additional insights.

7.4 Reading Strategies in User's Search Activities

The last section showed that information derived from eye movement is indicative for the search activities of users. Nonetheless, the predictive performance of the proposed model was not highly convincing when working on plain fixation information. Additionally, low-level eye gaze events and derived characteristics such as fixation counts, duration, etc. provide only limited potential for further interpretation. Therefore, this section will focus on the analysis of high-level eye movement patterns. These findings will serve as the foundation for a more advanced *interaction model* specifically design for eye movement patterns. This particular *interaction model* will be referred to as the *eye gaze model* within the *Information Search Behavior Profile Model*, see Fig. 5.2. The following section can be seen as a summary of my research work in *Fact-Finding or Exploration: Characterizing Reading Strategies in User's Search Activities* [114] but it also extends it with unpublished content. The aim of this research work is directly motivated by **RQ1**, but fortunately provides a foundation to further extend aspects of **RQ2**.

7.4.1 Task Description

The data recording described in *Logger & Eye-Tracker* during Sec. 6.3 comprises recordings of eye movement. Nonetheless, an eye-tracker basically records low-level information in form of *Fixations & Saccades* and derived characteristics of them, such as positioning, duration, pupil-size, etc. A higher form of contextualization for *Eye Movement in Reading* is missing and needs to be inferred by higher-order analysis modules. Even higher structured patterns for *Reading and Information Processing* are not provided by a standard eye-tracker. In pursuit to implement such needed analysis models, a high quality data set for structured eye movement patterns will be collected. This work extends the plain eye-tracker data with an additional layer of annotated gaze behavior.

7.4.1.1 Data Definition

The *Data Definition* in Sec. 7.2 & Sec. 7.3 remains valid. The following data description will focus on the specifics of the eye-tracking aspects. The experimental approach has been described in respect to the logger & eye-tracker during Sec. 6.3, the specifics of the eye-tracker has been clearly stated and the specific implementation of the lab experiment can be seen in Fig. 6.5. Nonetheless, the actual output of the eye-tracker has only been described on the surface level. Fig. 7.8 illustrates how the eye-tracker works. An illuminator emits a near-infrared light towards the participant. By analyzing the reflection patterns of the participants eyes, a complex pipeline of image processing algorithms calculates the positioning of the eyes gaze on the computer screen. All in all, this results in two sources of information when working with eye-tracking data, one being the video recording of the participant and one being the eye-tracker output itself.

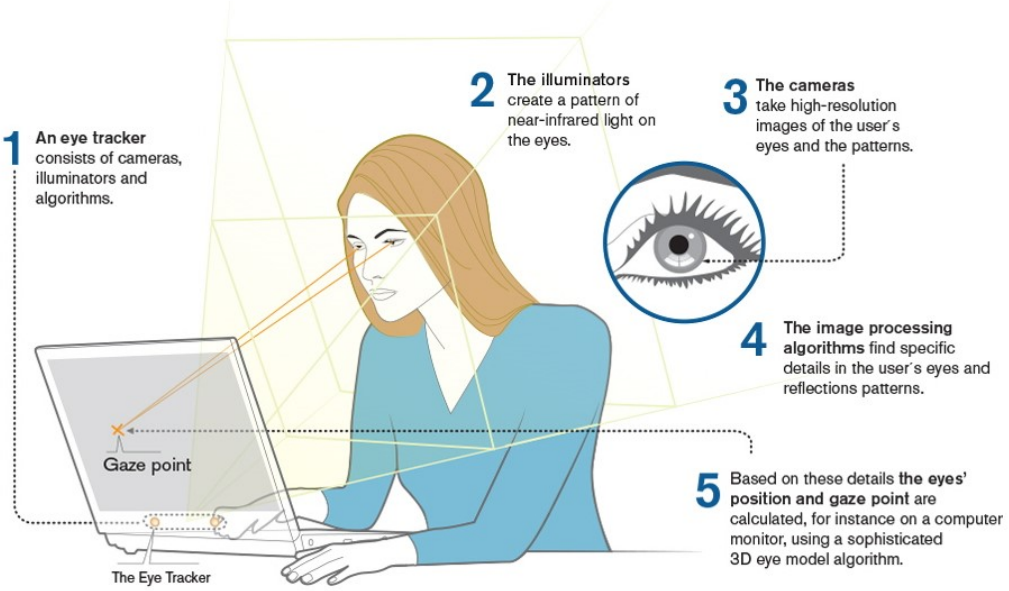


FIGURE 7.8: Taken from [96]: Screen based eye-tracker with the description of all components.

7.4.1.2 Model Definition & Annotation Guidelines

With video recordings of the participant's face and the recorded eye gaze on the display, enough information is provided to work with annotators to enrich the data with supplementary information about the eye movement pattern. In the context of information search, variations of reading strategies, namely *Scanning*, *Skimming* and thorough *Reading* are of particular interest. Further on, additional annotation states for other gaze pattern are needed, such as *MediaViews* or other (less defined) activities. Especially definitions of *Scanning* and *Skimming* may differ slightly amongst the related work and therefore annotation assumptions need to be stated explicitly. *Others* is a special rejection state in the annotation that describes a participant's gaze on keyboard, mouse, investigator, etc. Participants might adjust their glasses, chair, screen, etc. or simply implement a gaze pattern that is not captured by any of the following definitions. *MediaView* describes a state when the participant's gaze is not focused on textual information but graphs, tables, videos, images, etc. that are displayed on the screen. *Reading* describes a full and thorough reading behavior of a participant. The gaze of a participant is characterized by a rather slow eye movement with lots of fixations, few skips, increased saccadic *regressions* and short saccades on textual units. The participant is rather concentrated and the gaze behavior is focused on each aspect and/or word within the text. *Scanning* is a technique that is used when a reader is looking for something, such as a keyword or phrase. Readers move their eyes over the text mostly horizontally in the pursuit to identify the desired snippet. This process, in essence, demands the full attention of the reader. Scan-paths will be increased in duration and length. Participants are rather concentrated while looking at the screen. The gaze behavior is orientated towards keywords, phrases or towards textual structures, e.g. lists, references, titles, information boxes, etc. The individual gaze might be orientated towards formatting styles, e.g. *italic*, **bold**, color-highlights, etc. In general, this reading strategy is rather fast

and comprises fewer fixations, lots of skips, short/medium saccades, few regressions and/or re-reading. *Skimming* is a technique which a reader uses to identify the main points or essence of a text without consciously taking in every word. This process requires less attention and can be described by less fixations in form of counts & durations and negligible saccadic regressions. This gaze might be indicated by more vertical movement rather than horizontal ones. Participants appear rather less concentrated. The eye movement might even follow an F-shape of fixations within the reading sequence. The gaze is orientated towards textual sections such as abstracts and captions. In general, this reading strategy is rather fast and comprises fewer and shorter fixations, less saccadic regressions and long progressive saccades. All definitions of annotation states are founded and based on previous research already described during *Reading and Information Processing* in Sec. 2.2.3.3 but also follows an individual interpretation. The annotation process was entirely done by my research assistants Jessica Paul and Lena Bayer from the Psychology Faculty. I can not state enough that I had incredible luck to have found such disciplined and hard-working assistants that watched hours of video recordings for a high-quality annotation.

7.4.2 Inter-Annotation-Agreement

The experiment recorded sequences of low-level gaze events consisting of fixations and saccades. These sequences are manually annotated by two independent experts using recordings of the participant's face and their gaze tracked on the display. The annotation divides the sequences into chunks according to the *Model Definition & Annotation Guidelines*. Admittedly, the task is blurry, fuzzy and vague by nature. Therefore, a quality check of the annotation is necessary. In respect to its global quality, a consistency check for the annotation is needed, namely the *Inter-Annotation-Agreement*. The straight forward approach is the agreement in percentage between both annotations. A slightly more advanced approach, namely *Cohen's Kappa* [27], takes advantage of underlying statistical properties to quantify such an agreement. Further on, this measurement provides thresholds to group the annotation in interpretable categories of quality, see Tab. 7.7. Both methods were applied to the data and their results can be seen in Fig. 7.9. The majority of the annotation suffices a *moderate agreement* and two instances have a *substantial agreement*. Out of the 17 annotated sequences in total, 5 instances comprise a *fair agreement*. This observation was taken as a justification to exclude these data for subsequent analysis. The ambiguity in the annotation would introduce noise, deeming following estimates less trustworthy. In Fig. 7.10 one can see the color encoded alignment of 3 annotations: the best agreement ($\kappa = 0.66$), a borderline agreement ($\kappa = 0.42$) and the worst agreement ($\kappa = 0.28$). For data points above the acceptance threshold, it can be argued that the rather low value of agreement originates from the time delays during the annotation process and ambiguities in the human decision-making process for the fuzzy boundary between thorough *Reading* and *Scanning*.

Agreement's Quality	Threholds
poor agreement	$\kappa < 0$
slight agreement	$0 < \kappa < 0.20$
fair agreement	$0.21 < \kappa < 0.40$
moderate agreement	$0.41 < \kappa < 0.60$
substantial agreement	$0.61 < \kappa < 0.80$
(almost) perfect agreement	$0.81 < \kappa < 1.00$

TABLE 7.7: Taken from [114]: Cohen's Kappa (κ) [27] and its quality according [75].

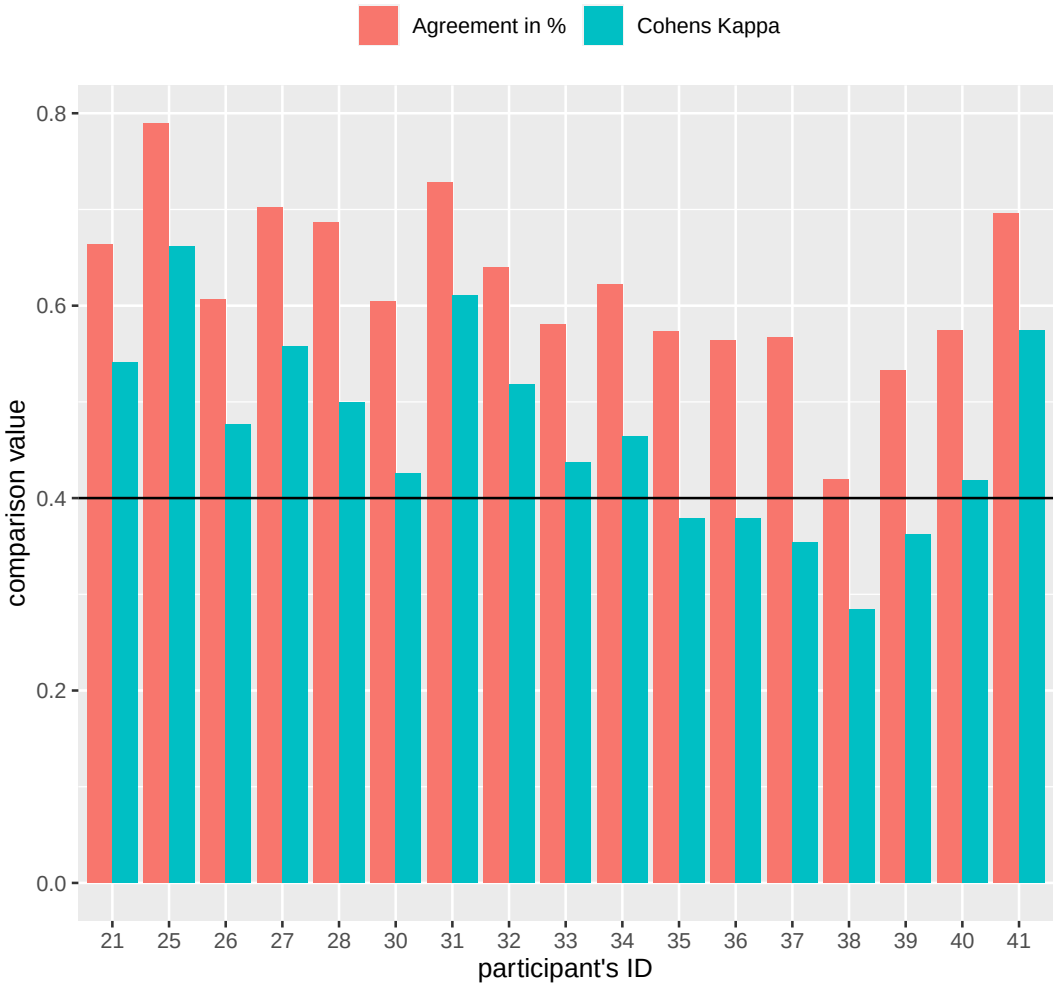
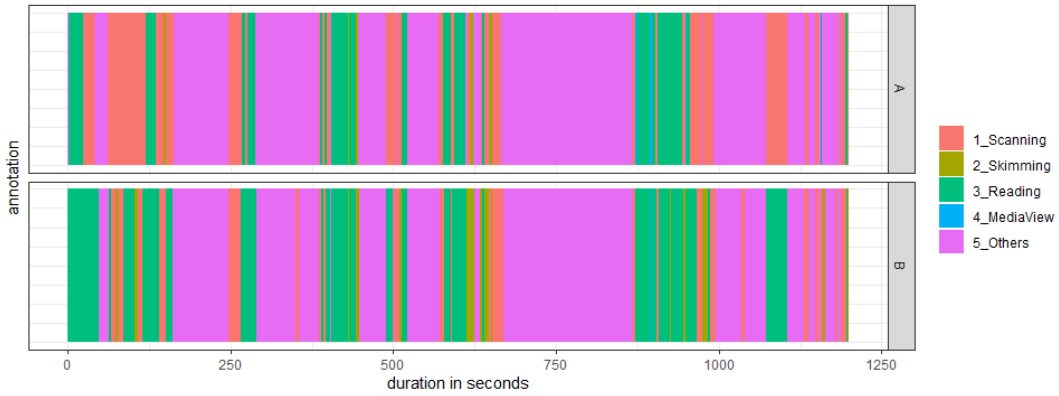
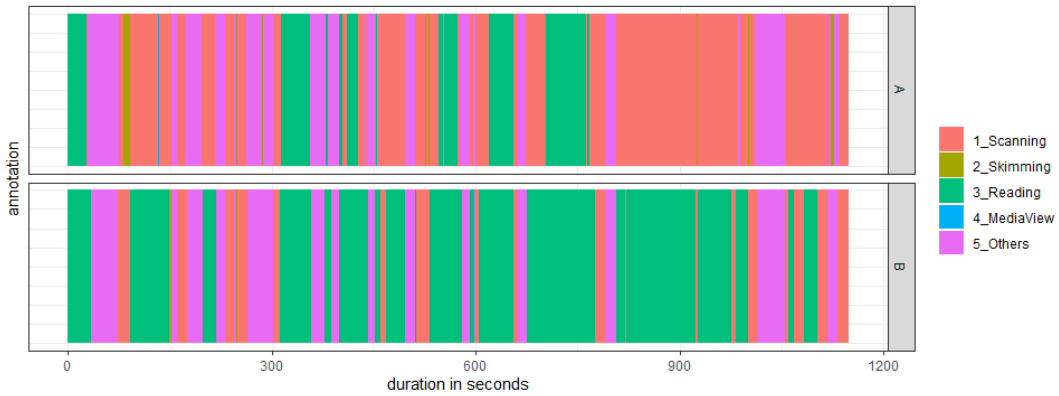


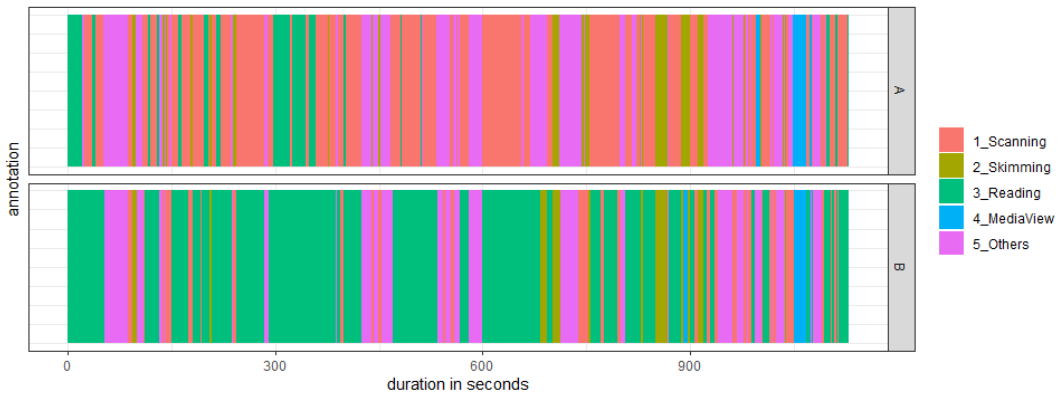
FIGURE 7.9: Taken from [114]: Inter-Annotation-Agreement in form of percentage and Cohen's Kappa [27]. Threshold for a moderate agreement according [75] as a horizontal line.



(A) Best agreement in ID 25 with $\kappa = 0.66$



(B) Agreement at threshold in ID 40 with $\kappa = 0.42$

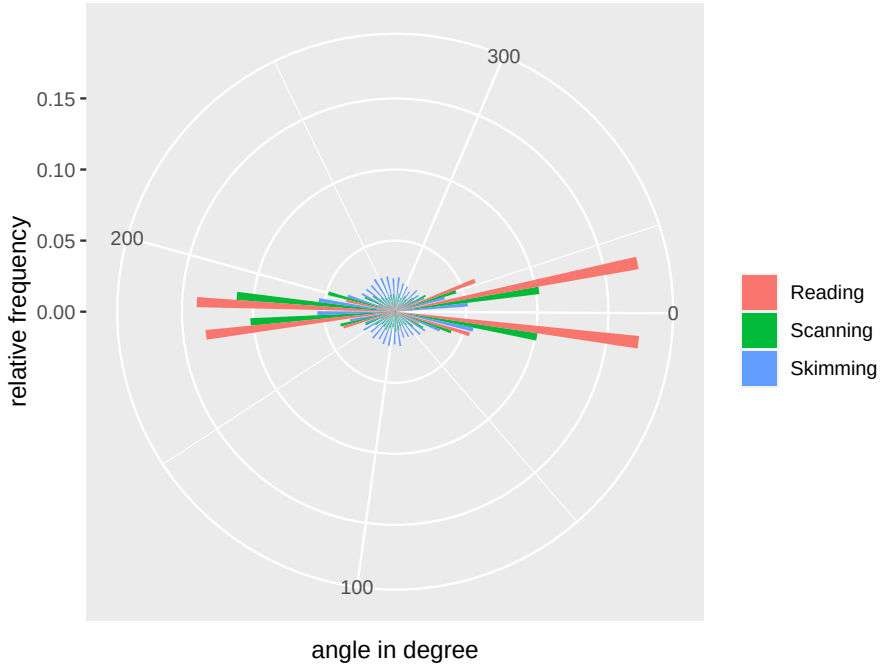


(C) Worst agreement in ID 38 with $\kappa = 0.28$

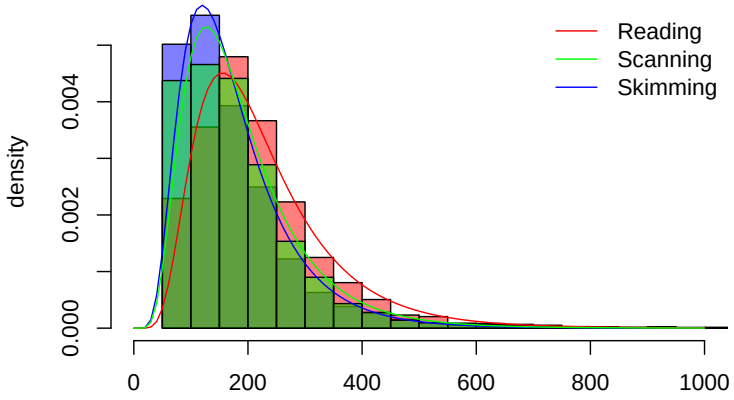
FIGURE 7.10: Adapted from [114]: Participants annotated gaze behavior during search sessions for different levels of annotation agreement.

7.4.3 Characteristics of Reading Strategies

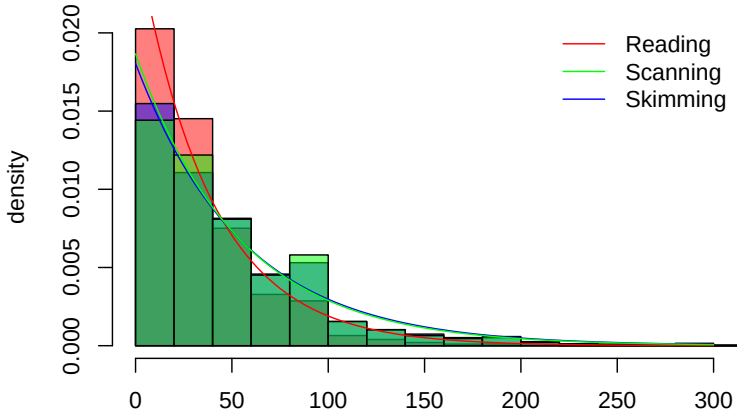
After verifying a reasonable consistency of the annotation globally, it is now necessary to check for the consistency locally. Therefore, the characteristics of the gaze events need to be inspected as well. The annotation of *MediaView* and *Others* is comparably trivial for the annotators, given the guidelines. Hence, further analysis will focus on characteristics specific to *Scanning*, *Skimming* and thorough *Reading*. This comprises saccadic directions, saccadic durations (implying big skips in case of long durations) and fixation durations (implying speed in case of short durations). In Fig. 7.11a, the *Relative Saccadic Direction* (as defined according [1]) of the annotation is visualized by a radial histogram. As expected, thorough *Reading* is dominantly horizontal orientated (0° and 180°). This includes an assumed forward reading activity at 0° but also an assumed re-reading activity at 180°, e.g. by regressions. Line jumps and other vertical orientated gaze (90° and 270°) are not often executed. *Scanning* is comparable in profile, but less dominant in its horizontal orientation. This indicates the execution of more jumps in other directions, skips and fewer regressions. Nonetheless, the text seems to be treated as a coherent chunk while searching for the precise snippet of text satisfying the individual's interest. *Skimming* is a gaze behavior that seems to take more advantage of the radial space. This includes a stronger vertical orientation, even though horizontal orientation is still more prominent. *Reading* comprises comparable longer fixation durations with a mean of 227.78 ms, as it can be seen in Fig. 7.11b. *Scanning* and *Skimming* show shorter fixation durations, with a mean of 190.50 and 179.16 ms, respectively. This implies a faster gaze behavior than the slower thorough *Reading*. *Skimming* is a bit faster than *Scanning* but comparable to it, nonetheless. To put it in perspective, Rayner [101] states that during the reading process, the fixation duration averages between 225 and 250 ms. Saccadic durations can be seen in Figure 7.11c. *Reading* has the shortest saccadic duration, with a mean of 38.04 ms. Therefore, the gaze behavior is focused on close proximity and longer skips are rather absent. *Scanning* and *Skimming* show longer saccadic durations, with a mean of 53.59 and 55.41 ms, respectively. This implies longer skips and distances. Clark et al. [26] states that saccades are very fast, typically taking 30-80 ms to complete. All in all, the local characteristics of the annotation guidelines in respect to gaze event durations, speed, proximity, saccadic directions, regressions and skips can be verified.



(A) Relative Saccadic Direction



(B) Fixation Duration in milliseconds



(C) Saccadic Duration in milliseconds

FIGURE 7.11: Taken from [114]: Gaze event durations of *Fixations & Saccades* (*Log-Normal Distribution* and *Exponential Distribution* as visual support) and Relative Saccadic Direction [1] (right (0°), left (180°), up (270°) and down (90°)).

7.4.4 Search Activities & Reading Strategies

With the consistency analysis of the annotation done globally and locally, further advantage of the experimental design can be exploited. As stated previously, the search sessions are assigned to either Exploratory or Fact-Finding search activities. By combining both, a contingency table can be created, see Tab. 7.8. The eye-tracker produced measurements in a sampling rate of several milliseconds. Therefore, the amount of annotated gaze events is on a huge scale, deeming a test for statistical independence less informative. Nonetheless, they can be reported with $\chi^2 = 25818, df = 4, p - value < 2.2e - 16$ using Pearson's χ^2-Test [92]. The normalized gaze behavior profile in Fig. 7.12 will give more reasonable insights. The most common gaze behavior in both kind of search activities is the *Others* type, with 40%. As a rejection class, in contrast to the other more precisely defined classes, it seems plausible to be that dominantly distributed. The least occurring gaze activity is *MediaView* with 1-3%. During an online search, textual content seems to be the most prominent source of information. Images, graphics and tables might be easier or faster to understand or simply are less often present during online searches. Both gaze behavior seemed to be equally distributed between search activities. In respect to the reading strategies, namely *Scanning*, *Skimming* and thorough *Reading*, some trends for search activities can be observed. A discriminative trend can be seen in thorough *Reading* with 25% to 17% and in *Skimming* with 8% to 13% in Exploratory and Fact-Finding search activities, respectively. In contrast, *Scanning* seems to be equally present in search activities, with 23% to 25%. The trends of reading strategies during search activities seem plausible given their underlying cognitive processes and are fully consistent to the initial expectation. An Exploratory search activity comprises a rather *open* task where the *Information Need* cannot be specified precisely. Therefore, an individual cannot rely on strategies to fasten up reading without the risk to lose information during the reading process. For that reason, the individual is forced to rely on thorough *Reading*. During Fact-Finding activities, an individual has a rather *closed* task with a more or less clear Information Need. Based on this clear need, the individual forms an expectation that is aimed to be found within the text. Therefore, faster strategies can be applied during reading. *Scanning* is a fast reading strategy with a precise concept of what is expected to be read, e.g. keyword or key phrases. Therefore, in the initial expectation, it was assumed to be heavily associated with Fact-Finding search activities. Nonetheless, one can see just a minor trend. It can be hypothesized that *Scanning* is used during both search activities rather for orientation and fast initial validations of presented web pages to justify further in-depth reading. In contrast, *Skimming* as a fast reading strategy is associated with Fact-Finding search activities. It can be hypothesized that the individual takes advantage of this strategy, presumably because a clear Information Need enables the option to select or skip textual subsections that the individual deems unworthy of fully reading.

Search Task	*MediaView*	*Reading*	*Scanning*	*Skimming*	*Others*
Fact-Finding	19431	95332	141207	71809	219953
Exploratory	13464	249102	225866	83668	391896

TABLE 7.8: Taken from [114]: Contingency table of annotated gaze behavior and search activities.

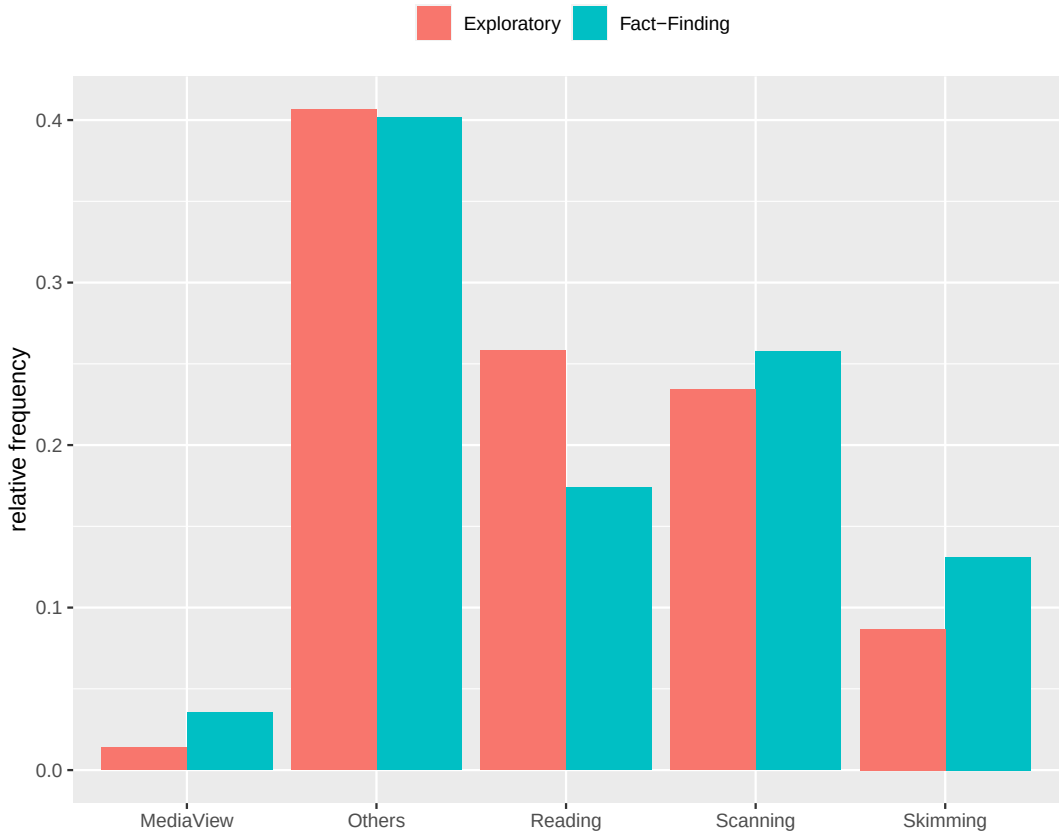


FIGURE 7.12: Adapted from [114]: Normalized profile of gaze behavior annotation for Exploratory (red) and Fact-Finding (blue) search activities.

7.4.5 Search Activities & Eye Movement Strategies

Further advantages can be exploited by combining the experimental design and the work of previous sections. Online search sessions of users are inherent contextual and multi-modal. The provided annotation of this section can directly be combined with the navigational trails of the user to create insights about their interconnection. Therefore, a prototypical Exploratory and Fact-Finding search activity will be selected via the maximum value of the respected *Maximum A Posteriori Predictions*. To analyze eye movement in a rather natural search condition, the menu/quiz tab was removed by *Addressing Experimental Limitations* described in Sec. 7.2.4.

7.4.5.1 Exploratory Search Activity & Eye Movement

As being stated in previous findings, Exploratory search activities heavily focus on the detailed and thorough inspection of web pages, as it can be seen in Fig. 7.13. The navigational *strategy* is mainly focused towards web pages and users spend a considerable amount of time on it. In only a few instances, a query will be executed and the inspections of SERPs comprise only a fraction of the entire search session. The search session mainly comprises web page visits. In respect to the eye movement *strategy*, the combination with the navigational trail uncovers a valuable interrelation. While entering the web page, a user starts the eye movement of *Scanning*. After an initial evaluation of the first impression seemingly has been satisfied, the user proceeds with a detailed inspection of its content via thorough *Reading*. A majority of the time spend on that page, the user is thoroughly *Reading*. This process is only shortly interrupted by *Scanning* for presumably more desired content on that same page. A fraction of the time spend on that web page is used to orientate or navigate the eye gaze on it by the gaze event *Others*. All in all, this prototypical search session can be considered to be quite coherent in respect to the eye movement behavior. Exploratory search activities seemingly comprise a rather consistent behavior in respect to eye movement and navigation. Given the interpretation of the rather *open* nature of Exploratory search activities where the *Information Need* cannot be specified precisely, it seems plausible that the user is unable to incorporate switches of strategies to speed up the search. Without a clear expectation of the search *goal*, the user cannot create decision criteria to identify undesired content and use it to skip sections without the risk to miss the desired information. For that reason, the individual is forced to rely on thorough inspection of entire web pages.

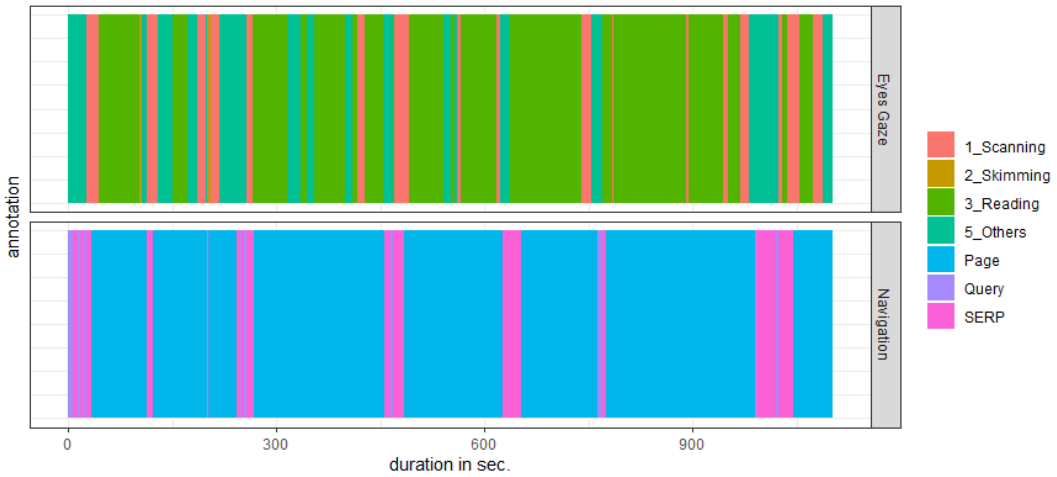


FIGURE 7.13: Prototypical search session for an Exploratory search activity comprising navigational trails (bottom) and eye movement pattern (top). Navigation comprises three states: *Query*, *SERP* and (web) *Page*, see Sec. 7.2. Eye movement pattern comprise the strategies: *Scanning*, *Skimming*, thorough *Reading*, *MediaView* and *Others*

7.4.5.2 Fact-Finding Search Activity & Eye Movement

As being stated in previous findings, Fact-Finding search activities rely less on the detailed and thorough inspection of web pages as compared to Exploratory search activities. The navigational *strategy* comprises an increased activity of inspecting SERPs and query formulations, as it can be seen in Fig. 7.14. Web pages are inspected as well, but show a statistical significant reduction in the time spent on it. In respect to the eye movement *strategy*, the combination with the navigational trail uncovers a valuable interrelation. While inspecting the web page, the user heavily relies on *Scanning* or *Skimming* through its content. Thorough *Reading* is only implemented in a fraction of the time during the search session, presumably in cases when the presented content was harder to understand and a detailed inspection is considered necessary. All in all, this prototypical search session can be considered to be characterized by fast switches in respect to the eye movement behavior. Fact-Finding search activities seemingly comprise fast switches or adaptions of behavior in respect to eye movement and navigation. Given the interpretation of the rather *closed* nature of Fact-Finding search activities where the *Information Need* can be specified precisely, it seems plausible that the user is able to incorporate switches of strategies to speed up the search. With a clear expectation of the search *goal*, the user can create decision criteria to identify undesired content and use it to skip sections without the risk to miss the desired information. For that reason, the individual can abandon web pages that are deemed unworthy of detailed inspection. Further, the user can do *Scanning* through SERPs and identify promising results early on in the search. In case the entire SERP is deemed unworthy, the adaption of the associated query can be done via such a fast inspection by *Scanning*.

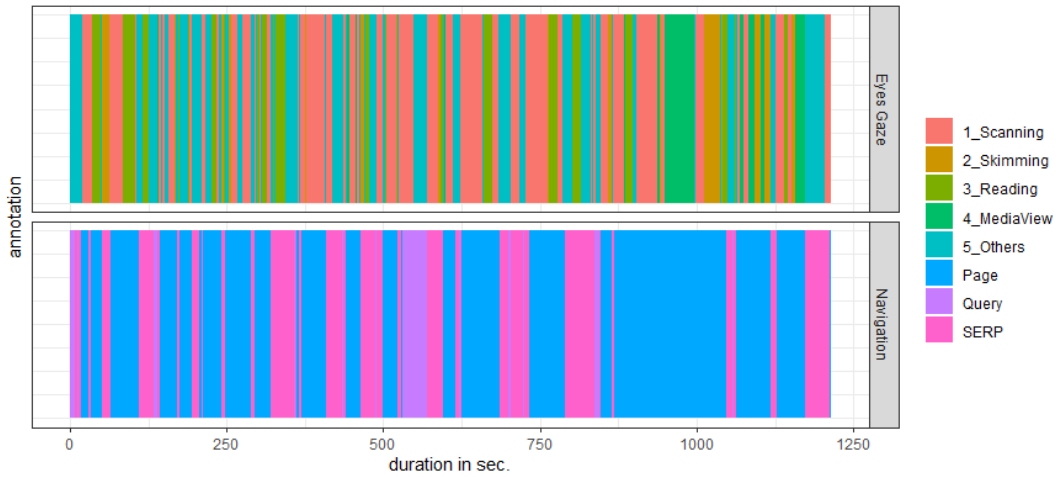


FIGURE 7.14: Prototypical search session for a Fact-Finding search activity comprising navigational trails (bottom) and eye movement pattern (top). Navigation comprises three states: *Query*, *SERP* and (web) *Page*, see Sec. 7.2. Eye movement pattern comprise the strategies: *Scanning*, *Skimming*, thorough *Reading*, *MediaView* and *Others*

7.4.6 Conclusion

In this section, different reading strategies in Exploratory and Fact-Finding search activities were investigated. Eye-tracker data of 17 participants were analyzed from search sessions in blocks of size up to 20 minutes. (Two participants of the entire study were excluded. The data of one participant was used for internal communication, and the data of another participant was used to train the annotators. Consequently, these two were removed from the following analysis.) Annotation guidelines were created, so human experts could enrich eye tracking data by supplementary information layers with the focus on reading strategies, such as *Scanning*, *Skimming* and thorough *Reading*. These strategies imply cognitive processes of intent and planing. Hence, it can be argued that these reading strategies enable researchers to derive more sophisticated conclusion about search activities than the analysis of low-level eye tracking feature. The data were analyzed for consistent annotation and their characteristics were put in perspective with previous research. Fact-Finding search activities can be described as a rather *closed* task with a clearly defined *Information Need*. This need is manifested in an individual's expectation that is aimed to be found during the search session. In respect to reading, this enables the individual the ability of faster reading strategies. Such an expectation can be mapped to the text for a fast decision of further 'detailed' text processing. *Skimming* is a fast reading strategy with the aim to identify main points or the essence of a text without thoroughly reading the text word-by-word. A positive correlated trend of this reading strategy towards Fact-Finding search activities can be observed, and it can be considered logical in its causal interpretation. In contrast, Exploratory search activities can be described as a rather *open* task with an Information Need that cannot be specified precisely. Therefore, this need cannot be manifested clearly in an individual's expectation. In the context of reading, individuals cannot rely on fast reading strategies to skip text without the risk to miss the desired information. A positive correlated trend of thorough *Reading* towards Exploratory search activities can be observed, and it can be considered logical in its causal interpretation. *Scanning* is a fast reading strategy to identify desired keywords or phrases. Both search activities make use of this reading strategy, with a minor trend towards Fact-Finding search activities. The conclusion is based on empirical findings, and can be justified by causal logic. Therefore, it can be assumed that the annotation enriches the interpretation and understanding of user search activities in online search sessions. In the context of this work, this section enriches the eye-tracker data used in previous experiments. Initial findings when *Combining Eye Tracking and Navigation* in Sec. 7.3 indicated that eye tracking data is a valuable source of information to draw conclusions about user search activities. Nonetheless, simplistic analysis on plain information derived from fixations & saccades is not sufficient to either boost the predictive performance of previous models nor to gain a deeper understanding of search activities. Within this section, evidence was provided for the claim that complex eye movement pattern, e.g. *Scanning*, *Skimming* & *Reading*, are associated to search activities and enrich the understanding of search behavior. Therefore, the *Information Search Behavior Profile Model* should consider these patterns via a specific *interaction model* (or more specific, an *eye gaze model*).

7.5 Automatic Reading Detection

The last section showed that eye movement pattern are quite indicative for search activities. Especially, reading and its variants provide certain interpretation for the cognitive state of users during their search sessions. The previous section provided a data set with eye-tracking data and annotations for the eye movement *strategy*. Based on this data set, an *interaction model* is aimed to be designed specifically for eye movement *strategies*, namely the *eye gaze model*. With that model implemented, the *Information Search Behavior Profile Model* in Fig. 5.2 is nearly finalized. The following section can be seen as a summary of my research work in *Automatic Reading Detection during Online Search Sessions* [117]. The aim of this research work is directly motivated by **RQ2** and partially adds perspectives to **RQ1**.

7.5.1 Task Description

In pursuit of a deeper understanding of search activities, the detailed analysis of *Reading Strategies in User's Search Activities* provided enough evidence to further explore the potential of a high-level analysis of the eye gaze. It is safe to say that there is an association between eye movement *strategies* and search activities, especially in the form of *Reading and Information Processing*. Unfortunately, eye-tracker with higher-order analysis modules for complex eye movement pattern are missing or are rudimentary designed by over-simplifications. For that reason, this section presents an approach for automatic reading detection based on the plain output of a standard eye-tracker. With a high quality data set provided that encodes annotations for defined eye movement *strategies* such as reading, there is a clear foundation given to work with a mathematical & computational *(User) Behavior Model* for eye gaze information. In this section, an approach for automatic reading detection models will be presented. By using such models, one is able to automatically identify strategic eye movement patterns and their characteristics during the information search process.

7.5.1.1 Data Definition

The *Data Definition* in Sec. 7.4 remains valid. In the following, the data description will be extended for the specifics of the eye-tracker output. Fig. 7.8 illustrates nicely the workflow of the eye-tracker in use. The following analysis considers the following feature from the eye-tracker output:

- *GazeEventType*:
  The type of eye movement event categorized as *Fixation*, *Saccade* or *Unclassified*.
- *GazeEventDuration*:
  The duration of an eye movement event measured in milliseconds.
- *FixationPointX(MCSpx)*:
  Coordinate of the fixation point on the horizontal axis.
- *FixationPointY(MCSpx)*:
  Coordinate of the fixation point on the vertical axis.
- *SaccadicAmplitude*:
  This is the distance between the previous fixation location and the current fixation location in visual degrees.

- *AbsoluteSaccadicDirection*:
  It shows the offset in degrees from the horizontal axis to the current fixation point, with the previous fixation point as the origin.
- *RelativeSaccadicDirection*:
  An angle calculated by comparing the absolute saccadic directions of the current and previous saccades.
- *DistanceLeft*:
  The distance between the left eye and the eye tracker.
- *DistanceRight*:
  The distance between the right eye and the eye tracker.
- *PupilLeft*:
  The estimated size of the left pupil.
- *PupilRight*:
  The estimated size of the right pupil.

Every gaze event in the output of the eye-tracker has been annotated for a specific eye movement pattern using the *Model Definition & Annotation Guidelines* in Sec. 7.4. The following pattern or *strategies* have been considered:

- *Scanning*:
  An eye movement pattern that can be considered as a special kind of reading. Scanning is a *strategy* that is used when a reader is looking for something, such as a keyword or phrase.
- *Skimming*:
  An eye movement pattern that can be considered as a special kind of reading. Skimming is a *strategy* which a reader uses to identify the main points or essence of a text without consciously taking in every word.
- *Reading*:
  An eye movement pattern that describes a full and thorough reading behavior of a participant. To differentiate it from *Scanning* & *Skimming*, it will be referred to as thorough *Reading*.
- *MediaView*:
  An eye movement pattern when the participant's gaze is not focused on textual information but graphs, tables, videos, images, etc. that are displayed on the screen.
- *Others*:
  All eye movement pattern that have not been described by the previous states.

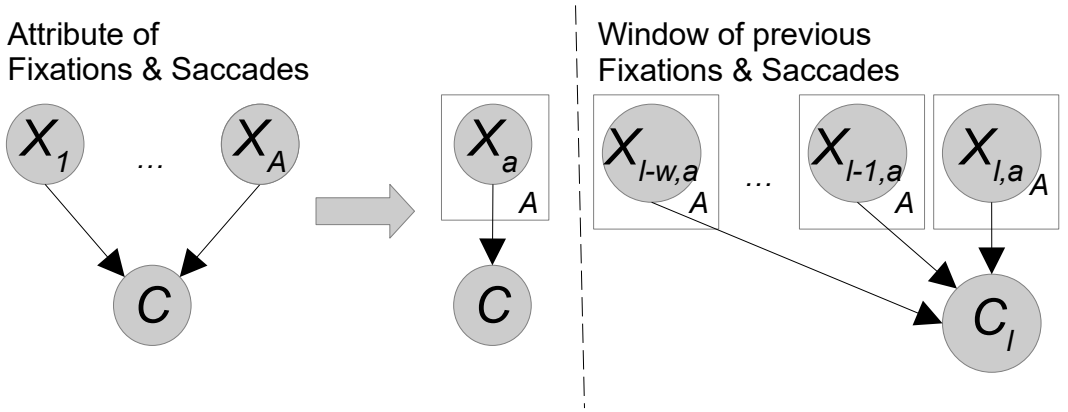


FIGURE 7.15: The Graphical Model of the proposed approach in the case of *Discriminative Classifiers*. Left: Models can classify gaze events based solely on its features. Right: Classifiers can be extended for a window of previous features.

7.5.1.2 Model Definition

Abstract *Bayesian Networks* (BN) can be created for the *eye gaze model*. During the following section, a set of multiple models will be applied, but stating every individual Likelihood will overblow this section. In the following, abstract *Discriminative Classifiers* will be stated exemplary. The introduced formalism of BNs will be used to keep the Likelihood as general as possible:

$$P(\boldsymbol{c}|\underline{\boldsymbol{x}}, \boldsymbol{\theta}) = \prod_{l=1}^{L} P(c_l|pa(c_l), \boldsymbol{\theta})$$

Each gaze point within the data set will be described by a distribution conditioned on an abstract *parent function* within BNs. Some reasonable parent functions are listed in the following:

- $pa(c_l) = \boldsymbol{x}_l$:
  The prediction of the n-th gaze point is conditioned on all features at this n-th time point.
- $pa(c_l) = (\boldsymbol{x}_{l-w}, ...\boldsymbol{x}_{l-1}, \boldsymbol{x}_l)$:
  The prediction of the n-th gaze point is conditioned on all features within a time window of size w up to the n-th time point.
- $pa(c_l)$ The prediction of the n-th gaze point is conditioned on dependencies stored within a graph structure of general BNs, e.g. $G = (\mathcal{V}, \mathcal{E})$

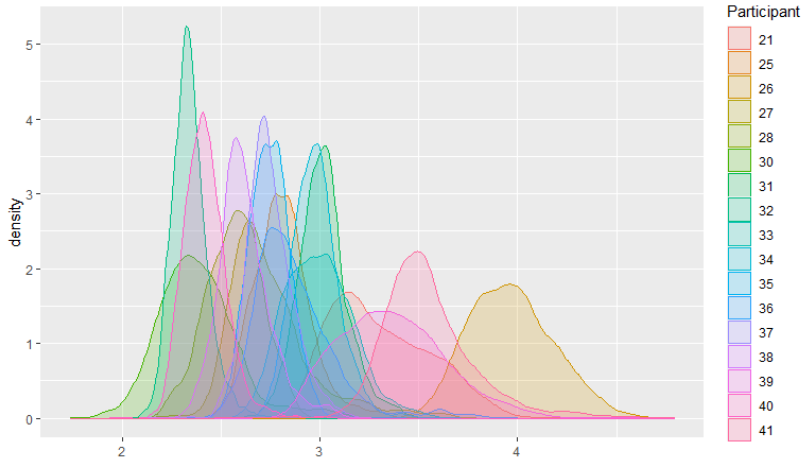
The first choice of parent function will result in models such as *Logistic Regression* [11], *Probit Regression* [41][15], etc. The second choice will result in the same models, but the model is able to consider a window in a time dependent way. The third choice of parent function can be considered as an instance of a (probabilistic) *Decision Tree* [98].

7.5.2 Models & Evaluation

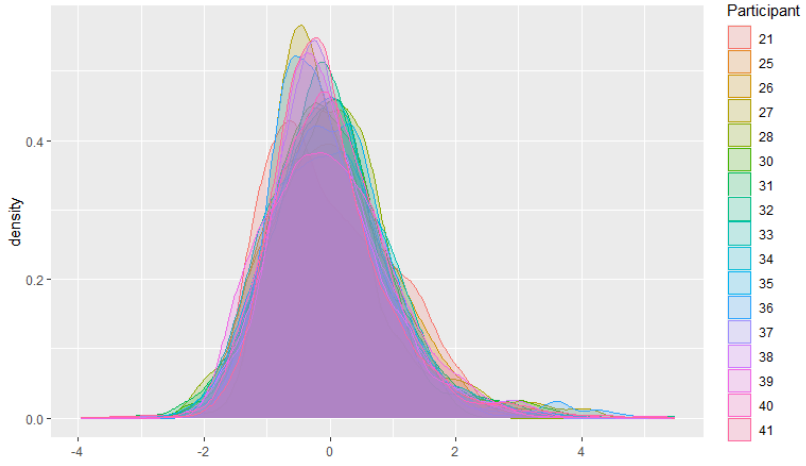
The data set consists of several annotation states, comprising variants of reading and gaze activities not related to reading. Within this initial approach, all reading variants, such as *Scanning*, *Skimming* & thorough *Reading*, are subsumed towards one general *Reading* category. All non-reading related activities, such as *MediaViews* and *Others* are subsumed towards one category, *Non-Reading*. This results in a dichotomous design or a 2-class classification scenario. A plethora of Machine Learning (ML) models can be used for that task. *Generative Classifiers* in Sec. 3.2.1.1 and Discriminative Classifiers in Sec. 3.2.1.2 already provided an in-depth description of the most commonly used approaches. Rather classical models are the *Linear Discriminant Analysis* [43] and the *Quadratic Discriminant Analysis* [44]. Both models are valuable baseline models of well described properties, such as their linear and quadratic decision functions. *Logistic Regression* (LR) [11] models are often used in ML and several 'adaptions' exists in form of *Probit Regression* [41][15], *Complementary-Log-Log Regression* [83] and *Cauchit Regression* [64]. All of them use linear decision functions in their most basic form. The *Support Vector Machine* (SVM) [131] gained much interest in the ML community because they gave rise to the idea of classification via *kernels*. While a linear SVM is 'comparable' to an LR, a kernelized SVM gains its strength via kernels to deform decision boundaries. Another ML model with flexible decision functions is the *Random Forest* [17]. Additional benefits of Random Forests arise from the built-in logic for *Feature Selection*, called *Feature Importance*, which is used to identify the feature's relevance in respect to the classification. Evaluation of classification tasks are standardized in ML (e.g. *Model Evaluation by Data Partitioning*, *Confusion Matrix*, *Accuracy*, *Precision*, *Recall*, etc.). Based on the description of *Individual Factors in Reading* influencing eye movements, it seems more than plausible to adapt the evaluation towards *Leave-One-Out Cross-Validation*, a.k.a. *Jack-Knife* [97]. One participant is used only for evaluation, while the remaining ones are used for model training. An overall performance measurement can be reported by averaging all Leave-One-(Participant)-Out samples, while the analysis of the individual samples give insights about the inter-participant (error) variance.

7.5.3 Data Preprocessing & Normalization

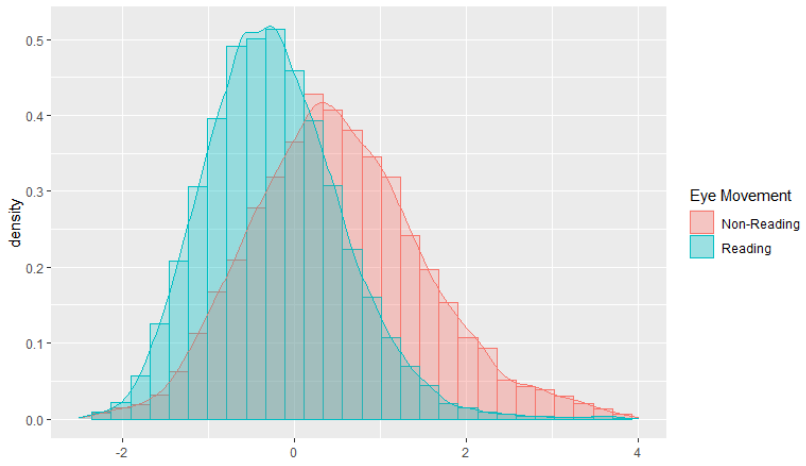
Based on the description of *Individual Factors in Reading* influencing the eye movement, it is plausible to normalize the eye tracker data. Such normalization aims to counter the effect of overfitting towards individual readers instead of general reading activities. Therefore, one might apply a participant-wise standardization via mean-centering and variance-scaling for all aspects in *Fixations & Saccades*. Fig. 7.16 illustrates this approach on the feature *Pupil Size* (defined as the average of *PupilLeft* and *PupilRight*). While Fig. 7.16a highlights the variability of this feature between participants, Fig. 7.16b illustrates the effect of the standardization. Fig. 7.16c indicates a clear trend in the normalized feature space by a decreased *Pupil Size* in case of *Reading*, presumable to focus on the text. Participant-wise standardization of this feature should normalize for physiological factors such as sex and age. With the same scheme, the feature for gaze event durations are also standardized because of the effect of different reading speeds. After normalization, faster fixation show a clear trend towards *Reading* while longer fixations are associated with *Non-Reading* activities. On the other hand, saccades are so fast in their nature that normalization does not change the data distribution much. Even though, all participants were advised to remain relatively still during the experiment, posture adjustments are natural for humans. After normalization of the feature *Distance to Screen* (defined as the average of *DistanceLeft* and *DistanceRight*), a small trend of adjustments towards the screen can be stated in case of *Reading* activities. Feature associated with the gaze's x & y coordinates on the screen are omitted completely. The eye movement in *Reading* should be treated independently of the exact text positioning on the screen. Gradients between fixations, on the other hand, are important parameters for reading detection and were realized in several variants. After data preprocessing, the entire feature space comprises the following standardized features: *Pupil Size*, *GazeEventDuration*, *Distance to Screen* and the following non-standardized ones: saccadic directions in form of *SaccadicAmplitude*, *AbsoluteSaccadicDirection* and *RelativeSaccadicDirection*.



(A) Pupil Sizes in millimeters



(B) Normalized Pupil Sizes



(C) Normalized Pupil Sizes & Gaze Activity

FIGURE 7.16: Taken from [117]: Data Normalization: Feature such as *Pupil Size* comprise a high inter-participant variance (7.16a). Participant-wise standardization reduces this individuality (7.16b). Normalized feature show trends in gaze activities for *Reading* and *Non-Reading* (7.16c).

7.5.4 Hyperparameter Tuning

Complex eye movements, such as reading, form complex pattern in a time-dependent way. Reading does not comprise a strict line-wise movement from left-to-right but also new-line jumps, re-reading of words/phrases via *regressions* and skip-jumps of small words/phrases. Often regressions and skip-jumps are executed 'unconsciously'. Automated reading detectors need a certain time window to put these events into context. Such windows were defined as successive fixations & saccades considered directly before the prediction of a particular time point. While predictions on single gaze events result in 28% error-rates, they decrease towards 21% with increasing window size. The error-rate starts to saturate at the size of 6 and the size of 10 resulted in the lowest error. Reported rates were generated by a Random Forest exemplary, but the overall trend was consistent in all models. Fig. 7.17 illustrates the effect of the increased window size on the prediction within its time context. Predictions on a window size of one are characterized by fast switches and high error-rates. The increased size of 10 smooths out these switches in predicted gaze activities while simultaneously decreasing the error.

FIGURE 7.17: Adapted from [117]: Effect of increasing window sizes on reading detection. Comparison of gold standard (top) and predictions (below) for reading activities of participant ID=25 (annotation A, see Fig. 7.10a). Center: Small window size of one results in a high error-rate and fast switches in predicted activities. Bottom: Increasing the window size to 10 results in longer and smoother predictions with a reduced error-rate.

Further on, reading comprises a diverse set of feature. To identify relevant feature for the detection task, feature selection was applied to reduce the entire set only to the relevant ones. Therefore, the built-in logic of a Random Forest was used to rank the feature for their relevance. Fig. 7.18 illustrates this ranking in respect to the *Reading* category. The overall (mean) feature importance follows the ordering (normalized) *GazeEventDuration*, (normalized) *Pupil Size*, *SaccadicAmplitude*, (normalized) *Distance to Screen*, *AbsoluteSaccadicDirection* (ASD) and *RelativeSaccadicDirection* (RSD). Features close to the point of prediction are assigned the highest importance while features before the point of prediction gradually lose importance. This is fully consistent with the saturating error-rate during window size adjustment. ASD and RSD are more-or-less redundant measurements of the *SaccadicAmplitude*. Because of their limited importance and redundancy, these feature will be removed from the following analysis. Also, the *Distance to Screen* feature, which measures posture adjustments, comprises a rather mediocre importance. Even though, this feature is not clearly underperforming, it can be considered a potential source for overfitting towards 'more active posture adjusters', an act not quintessential to 'actual' reading. All in all, it can be concluded that only the following features within a time window of 10 gaze events are relevant for automatic reading detection models: *Pupil Size*, *GazeEventDuration* and *SaccadicAmplitude*. The importance of *GazeEventDuration* is by no means surprising. Related work in respect to *Eye Movement in Reading* in Sec. 2.2.3.2 mentioned the research done to describe its characteristic values. The importance of *Pupil Size* might be explained by the necessary pupil adjustments to focus and fixate the text. *SaccadicAmplitude* is also reasonable because it reflects the direction in which the eyes moving on the screen in the form of the distance in visual degrees between the previous fixation location and the current fixation location as defined by the fixation filter.

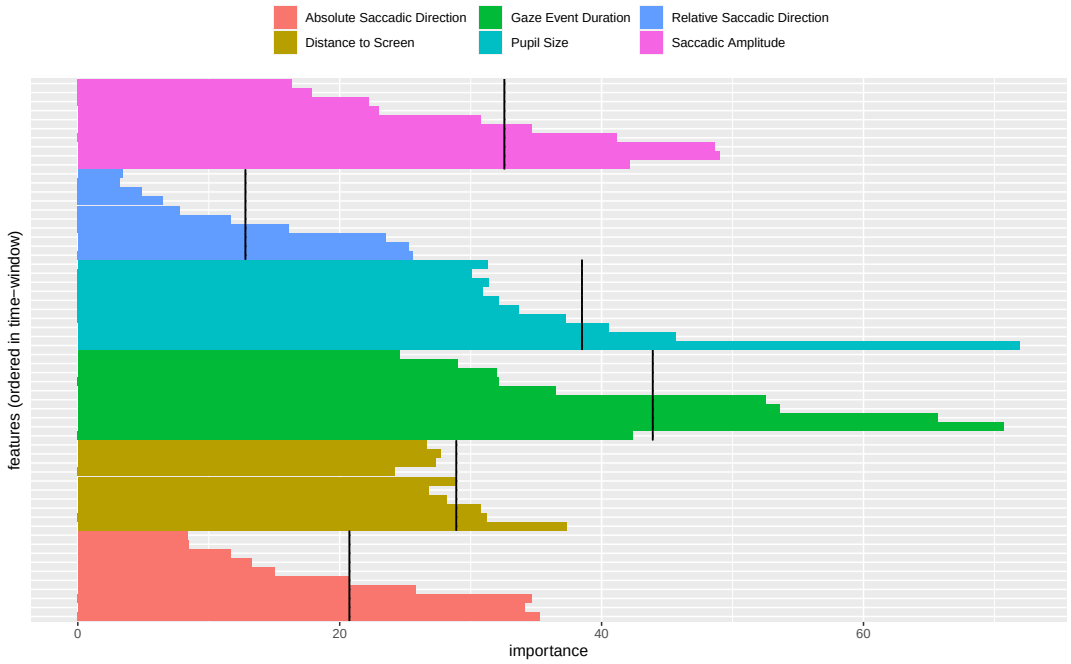


FIGURE 7.18: Adapted from [117]: Feature Importance is color encoded by groups. Mean group importance is marked with a black line. Within a group, features are ordered in respect to their position in the window of size 10. Features closest to the prediction point are most important, while features further apart gradually lose relevance.

7.5.5 Comprehensive Evaluation

After suitable data normalization, model and hyperparameter tuning, only the comprehensive evaluation of the candidate models remains open. Leave-One-(Participant)-Out Cross-Validation, a.k.a. Jack-Knife is used to capture the inter-participant (error) variance and to approximate the model's generalizability. Fig. 7.19 illustrates the error-rate of all individual participants. Despite the varying model assumptions (e.g. linear, quadratic, kernelized, tree), all models perform at a comparable level. Overall, the Random Forest performs best with an 20.77% error-rate, followed by a *Radial Basis Function* [88] kernalized SVM with an 22.52% error-rate. The *Quadratic Discriminant Analysis* performs worst with 25.01%, while the remaining models perform around 24%. The global error-rate of the models is somewhat misleading because of the high inter-participant (error) variance, ranging from 7.36% (best) to 42.43% (worst). Such a high variance seems reasonable in respect to the *Individual Factors in Reading* influencing the eye movement. Unfortunately, the proposed data normalization was not sufficient to fully generalize the data towards a common reading activity. Nonetheless, it is the best performing processing pipeline that was observed during the work.

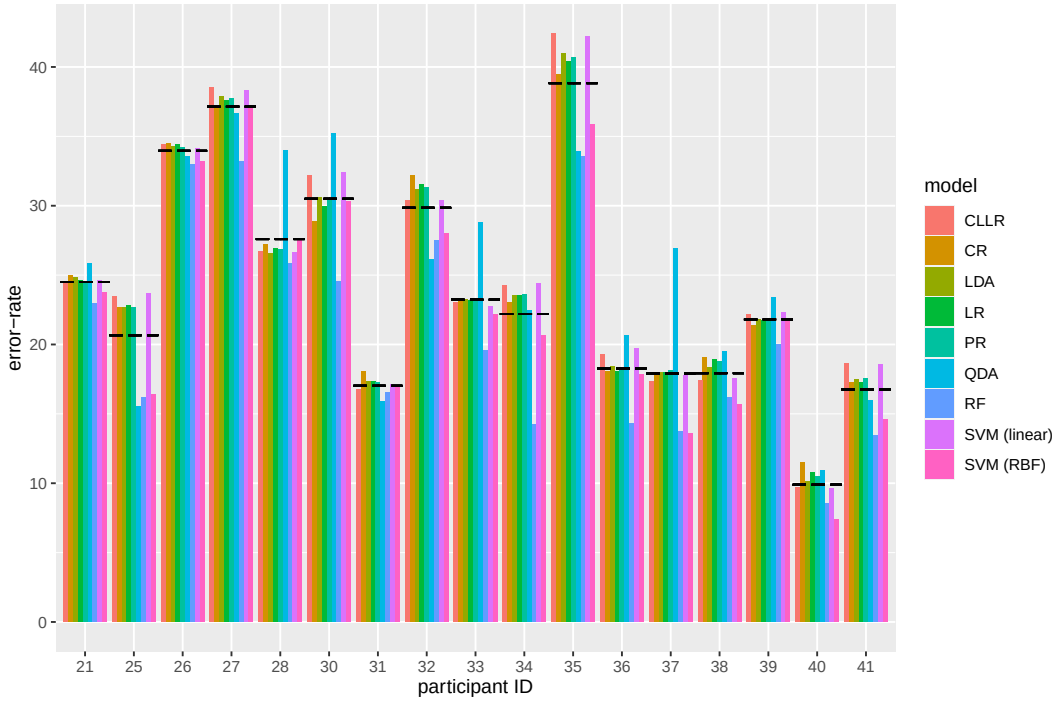


FIGURE 7.19: Adapted from [117]: Participant-wise error-rates for automated reading detection indicate a high variance between the reading activities of individuals. Comparison of Linear Discriminant Analysis (LDA) [43], Quadratic Discriminant Analysis (QDA) [44], Logistic Regression (LR) [11], Probit Regression (PR) [41][15], Cauchit Regression (CR) [64], Complementary-Log-Log Regression (CLLR) [83], *Support Vector Machine* (SVM) [131] (linear [88] & RBF-kernel [88]) and *Random Forest* [17]. Mean (model) error-rates per participant are marked with black lines.

7.5.6 Conclusion

The proposed approach indicates that automatic reading detection under natural conditions is possible with an error-rate of 20.77% (or 79.23% *Accuracy*) on a data set of highly educated participants of young to medium age. The work comprises a detailed description of data processing, normalization, model & hyperparameter adjustments and the evaluation of 9 candidate models. Error-rates range from 20.77% to 25.01% indicating a rather small variance between models. The evaluation indicates that models with flexible assumptions, such as tree-based and kernelized models, perform slightly better compared to the parametric ones using linear or quadratic decision functions. Nonetheless, it can be concluded that particular model choices might not have the biggest impact on solving the task. The analysis of the inter-participant (error) variance shows a range from 7.36% up to 42.43%. This indicates that *Individual Factors in Reading* are more influential to the results. Therefore, the report of just global error-rates/accuracies might be misleading metrics for the evaluation of automatic detection. The applied Leave-One-(Participant)-Out Cross Validation, a.k.a. Jack-Knife seems to be a reasonable approach to analyze such effects. Further on, the huge inter-participant variance highlights the need for adequate data processing and normalization. All in all, the described approach slightly underperforms in respect to the reference work of [22], [67], [12] & [65]. Nonetheless, the results are plausible and justifiable. The high inter-participant variance indicates that the specific cohort (data set) has a high impact on the evaluation metric. The description of this cohort, data collection and annotation guidelines implemented by 2 independent human experts is fully described in [114]. This study added additional layers to the analysis. The entirety from data collection to model evaluation is fully transparent (and reasonably reproducible). Automatic reading detection under natural conditions, e.g. complex web-pages with text, pictures, videos, graphics, etc., can effectively been used to monitor online searches. It can be stated, that users who are actively reading web-pages invest cognitive resources in that page. It can be argued that higher user investment equals higher page relevance. On one hand, this can become a valuable information source to increase understanding of users search behavior. On the other hand, reading activities can serve as relevance measurements for web-pages during online searches. Either way, automatic reading detection is a fruitful component for user modeling in Information Retrieval systems to anticipate user interests and provide support by analyzing the user behavior within the search session. In the context of this work, the described framework can be directly combined with previous models to form even broader Bayesian Networks. This combination will result in the *Information Search Behavior Profile* model in Fig. 5.2. The simplistic *Navigational & Interaction Model* in Sec. 7.2 provided promising results, but *Combining Eye Tracking and Navigation* in Sec. 7.3 by using plain fixations & saccades needed more complex *eye gaze models*. Automated reading detection (ARD) models realize a promising extension for that. Therefore, the *eye gaze model* was constructed in addition to the previous *navigational & interaction model*. In the context of search sessions, ARD models can be considered as a plausible instance of an *eye gaze model* to make inference about search activities.

7.6 Information Search Behavior Profile Model

All the previous sections provided sub-models to work towards the implementation of the *Information Search Behavior Profile Model* (ISBP model) in Fig. 5.2. The ISBP model comprises three components that are designed to analyze aspects of behavior in user search activities. The *navigational model* evaluates the navigational *strategy* of a user during the search session. The *interaction model* evaluates *actions* a user executes while being in an individual state during the search session. These *actions* are either derived from log-files, e.g. clicking, scrolling and dwell times, or derived from eye-tracker data, e.g. fixations and saccades. The *interaction model* was extended with an additional *eye gaze model*, which provides high-level eye movement *strategies* derived from low-level eye-tracker information. Even though, all sub-models have been finalized, the connection into one global multi-modal *(User) Behavior Model* remains open. This section will combine all sub-models, and the aim of this research work is directly motivated by **RQ2**.

7.6.1 Task Description

The ISBP model has been constructed by the careful evaluation of several rather independent sub-models. During the following, all sub-models will be consistently connected with each other. This global & multi-modal model will be called the *Information Search Behavior Profile Model* and will serve as an instance of (User) Behavior Models suitable for the analysis of search activities.

7.6.1.1 Data Definition

All the previous data definitions remain valid, but they will be re-grouped to enhance their semantic interpretation. The data set comprises aspects derived from log-files, low-level eye-tracking data and high-level eye movement pattern:

- *Log-File Interaction* ($\boldsymbol{x}^{LF}$):
  All *actions* a user implements within the search that can be derived from log-files. Example features have been presented in *Data Definition* in Sec. 7.2 and *Data Definition* in Sec. 7.3. In the following, *Log-File Interaction* will not comprise any feature derived from eye-tracking information.

- *Eye-Tracker Interaction* ($\boldsymbol{x}^{ET}$):
  All *actions* a user implements within the search that can be derived from an eye-tracker. Example features have been presented in *Data Definition* in Sec. 7.3 and *Data Definition* in Sec. 7.4. In the following, *Eye-Tracker Interaction* will only include low-level eye-tracking data, not high-level eye movement pattern.

- *Eye-Movement Interaction* ($\boldsymbol{x}^{EM}$):
  All *strategies* a user implements within the search that can be considered high-level eye movement pattern. Example feature have been presented in *Data Definition* in Sec. 7.4 and *Data Definition* in Sec. 7.5. In the following, *Eye-Movement Interaction* will only include high-level information derived from low-level eye-tracking data.

- *Interaction* ($\boldsymbol{x} = (\boldsymbol{x}^{LF}, \boldsymbol{x}^{ET}, \boldsymbol{x}^{EM})$):
  The set of *actions* & eye movement *strategies* a user implements within the search defined as the union of all previous mentioned interaction.

7.6.1.2 Model Definition

The *Model Definition* in 7.3 and *Model Definition* in 7.5 remain valid and will serve as the foundation for further modifications of the model:

$$P(\underline{\boldsymbol{z}}|\boldsymbol{\theta}_c) = \prod_{n=1}^{N} P(s_{n1}|\boldsymbol{\theta}_c) \cdot \prod_{l=2}^{L_n} P(s_{nl}|s_{n(l-1)}, \boldsymbol{\theta}_c) \cdot \prod_{l'=1}^{L_n} P(\boldsymbol{x}_{nl'}|s_{nl'}, \boldsymbol{\theta}_c)$$

7.6.2 Information Search Behavior Profile Model

With the provided *Data Definition* & *Model Definition* the model can be formulated as follows:

$$P(\underline{\boldsymbol{z}}|\boldsymbol{\theta}_c) = \prod_{n=1}^{N} P(s_{n1}|\boldsymbol{\theta}_c) \cdot \prod_{l=2}^{L_n} P(s_{nl}|s_{n(l-1)}, \boldsymbol{\theta}_c) \cdot \prod_{l'=1}^{L_n} P(\boldsymbol{x}^{LF}_{nl'}, \boldsymbol{x}^{ET}_{nl'}, \boldsymbol{x}^{EM}_{nl'}|s_{nl'}, \boldsymbol{\theta}_c)$$

The *product rule/chain rule* can be applied to the joint probability distribution of all *Interaction*:

$$P(\boldsymbol{x}^{LF}, \boldsymbol{x}^{ET}, \boldsymbol{x}^{EM}|s, \boldsymbol{\theta}_c) = P(\boldsymbol{x}^{LF}|\boldsymbol{x}^{ET}, \boldsymbol{x}^{EM}, s, \boldsymbol{\theta}_c) \cdot P(\boldsymbol{x}^{EM}|\boldsymbol{x}^{ET}, s, \boldsymbol{\theta}_c) \cdot P(\boldsymbol{x}^{ET}|s, \boldsymbol{\theta}_c)$$

This results in the decomposition into three parts. By applying further conditional independence assumptions, this will simplify into the following:

$$P(\boldsymbol{x}^{LF}, \boldsymbol{x}^{ET}, \boldsymbol{x}^{EM}|s, \boldsymbol{\theta}_c) = P(\boldsymbol{x}^{LF}|s, \boldsymbol{\theta}_c) \cdot P(\boldsymbol{x}^{EM}|\boldsymbol{x}^{ET}, \boldsymbol{\theta}) \cdot P(\boldsymbol{x}^{ET}|s, \boldsymbol{\theta}_c)$$

The decomposition results in three familiar parts. The first part $P(\boldsymbol{x}^{LF}|s, \boldsymbol{\theta}_c)$ is the (log-file) *interaction model*, just as in Sec. 7.2. The second part $P(\boldsymbol{x}^{EM}|\boldsymbol{x}^{ET}, \boldsymbol{\theta})$ is the (eye movement) *eye-gaze model*, just as in Sec. 7.5. The third part $P(\boldsymbol{x}^{ET}|s, \boldsymbol{\theta}_c)$ is the (low-level eye-tracking) *interaction model*, just as in Sec. 7.3. For sure, the conditional independence assumptions seem a bit ad-hoc at first. The *eye-gaze model* evaluates high-level eye movement pattern. It can be assumed that processes such as reading are quite conserved, and reading as a process will work the same way on a SERP as it will work on a web page. Also, the inspection of an image will be the same process independent of that image being present on a SERP or a web page. This assumption applies the independence of the proposed factorization, and it can be considered to be meaningful. The same reasoning will be used for the (log-file) *interaction model*: a clicking event is a clicking event, independent of the eye gaze during that click. The proposed ISBP model is visualized with its *Graphical Model* [88] in Fig. 7.20.

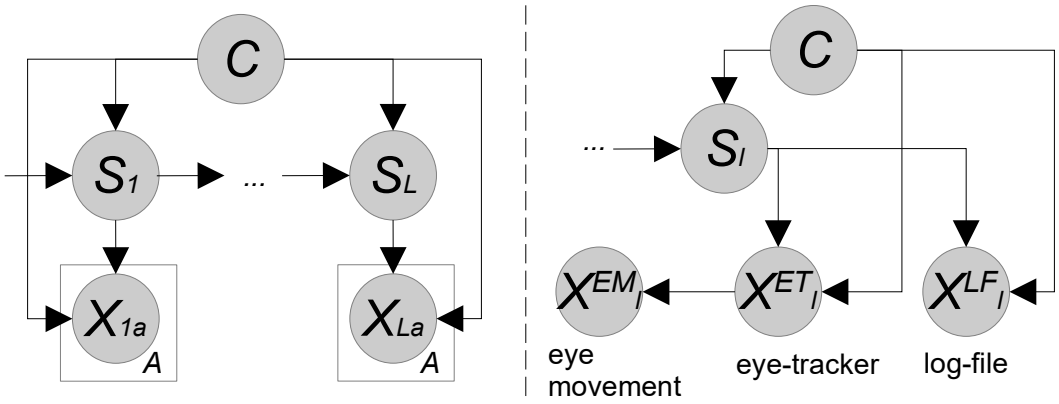


FIGURE 7.20: The Graphical Model for the Information Search Behavior Profile Model. The previous interaction & navigational model (Fig. 7.6) is extended with an additional eye gaze model. While the navigational model evaluates transitions through the states $\{S_l\}$, the interaction model evaluates actions derived from log-files $\{X_l^{LF}\}$ & eye-tracking data $\{X_l^{ET}\}$. The eye gaze model evaluates simple actions from eye-tracking data to conclude higher-level eye-moment strategies $\{X_l^{EM}\}$. The global model is conditioned on the search activity C except the eye gaze model. Nodes reflect model variables and edges their interaction: $A \rightarrow B = P(B|A)$.

In an abstract way, the overall Likelihood of the ISBP model is defined as follows:

$$P(\underline{z}|\boldsymbol{\theta}) = \prod_{n=1}^{N} naviModel(n, context(1)) \cdot \prod_{l=2}^{L_n} naviModel(n, context(l)) \cdot \prod_{l'=1}^{L_n} interModel(n, l') \cdot eyeModel(n, l')$$

- $naviModel(n, context(l))$:
  A model to evaluate the navigational *strategy* within a given context in the search session. Second-order *Markov Models* have been shown to be reasonable choices to model search activities by multiple ways of model selection (Sec. 7.2 & Sec. 7.3).

- $interModel(n, l)$:
  A model to evaluate *actions* within the search session in respect to *Log-File Interaction* (Sec. 7.2) & *Eye-Tracker Interaction* (Sec. 7.3). The *Naive Bayes Assumption* has been shown to be a reasonable choice and the individual feature could be modeled sufficiently by the *Exponential Distribution*, the *Log-Normal Distribution* etc. Several feature comprise statistical significance to differentiate for search activities.

- $eyeModel(n, l)$:
  A model to evaluate the eye movement *strategy* within the search session. Several *autoregressive Discriminative Classifiers* have been shown to capture eye movement pattern such as reading. The interrelation of eye movement pattern and search activities have been sufficiently described (Sec. 7.4).

The presented notation was maximally simplified for clarity. The ISBP model is a special instance of *Bayesian Networks*. Major parts follow the architecture of *Hidden Markov Models* (HMM) [99]. While the *navigational model* realizes *transition probabilities*, the *interaction model* realizes the *emission probabilities*. The *eye gaze model* follows autoregressive dependencies in a discriminative way. Therefore, the entire ISBP model represents a hybrid form of an *autoregressive HMM* [40] and a standard HMM.

7.6.3 Conclusion

The *Information Search Behavior Profile Model* (ISBP model) has been fully formulated. Online search sessions are inherent contextual and multi-modal. During such sessions, users implement search activities which comprise *actions* and *strategies*. To capture these multi-modal perspectives, the entire model decomposes into the *eye gaze model*, the *interaction model* and the *navigational model*. While the *navigational model* evaluates the navigational *strategy* of the search, the *eye gaze model* evaluates the eye movement *strategy* during it. The *interaction model* additionally evaluates *actions* a user implements during the search. In its entirety, the ISBP model forms a Bayesian Network. More precise, it resembles a hybrid of an *autoregressive Hidden Markov Model* (HMM) and a standard HMM. Nonetheless, the model was quite abstractly formulated. Therefore, the ISBP model can be considered a model family or a model architecture. Individual choices for the *eye gaze model*, the *interaction model* and the *navigational model* can be exchanged for as long as the entire network remains directed and acyclic. Further, the network can be extended for additional modalities in respect to *actions* and *strategies*. Respected models can easily be incorporated into this model architecture.

7.7 Ranking with Information Search Behavior Profiles

The previous sections focused on modeling suitable *(User) Behavior Models* for user online search sessions, specifically in respect to modeling *Exploratory* and *Fact-Finding* search activities. The resulting models were introduced as the *Information Search Behavior Profile Model* (ISBP Model). In essence, the ISBP Model decompose into *eye gaze*, *interaction* and *navigational model*. The question arises of how to make use of these model in the Information Retrieval (IR) setting to realized contextualized, behavior-driven and user-centered ranking. The following section will address this question and fully focus on the aspects of **RQ3**.

7.7.1 Task Description

In the pursuit to realize a contextualized, behavior-driven and user-centered ranking, a suitable combination of the proposed (User) Behavior Model with the ranking model of the IR system needs to be identified. The *Bayesian Network Model* [106] generalizes the classical ranking approaches, e.g. *Boolean Model*, *Vector Space Model* [110] & *Probabilistic Model* [108]. Fortunately, the Bayesian Network Model is an instance of *Bayesian Networks* and the proposed ISBP Model is a Bayesian Network as well. Both models can be combined into a broader Bayesian Network to consistently realize a unified ranking guided by behavior-driven aspects. In the following, a suitable representation of such a combination of (User) Behavior Models and ranking functions will be identified.

7.7.1.1 Data Definition

The *Data Definition* in Sec. 7.6 remains valid but will be slightly extended. User search sessions can be thought of as a sequence of interactions with a search engine and a web browser. These sequences form a navigational pattern that are used within an online search. Each singular element in this pattern can be represented as a discrete state. Within each state, users can execute certain actions or sub-strategies:

- *Query* (Q):
  A searcher is formulating a query on a search engine, which will be encoded in a suitable representation $\boldsymbol{q}$. This process takes a certain amount of time (*Q.Duration*). Being in a real search session, the user might be confronted with text from a previous search request, which might be fixated (*Q.Fixation-Count*) or scanned for some snippets to formulate an adequate query (*Q.Scanning*). While $\boldsymbol{q}$ is the textual representation of the query, Q represents a state in the search itself with all associated *actions* and *strategies* implemented by the user.

- *Page* (P):
  A searcher is viewing a web page and this page is called a document, which will be encoded in a suitable representation $\boldsymbol{d}$. The examination of the web page takes a certain amount of time (*P.Duration*) and a user will interact with it by scrolling (*P.Scrolling*) and clicking (*P.Clicking*). The user will fixate the web page (*P.Fixation-Count*) and process the presented information by thorough reading (*P.Reading*) or scanning through it (*P.Scanning*). While $\boldsymbol{d}$ is the textual representation of the document itself, P represents a state in the search itself with all associated *actions* and *strategies* implemented by the user.

7.7.1.2 Model Definition

Ribeiro-Neto et al. [106] postulated the *Bayesian Network Model* (BNM) Eq. (7.1) as a general approach for ranking in Information Retrieval. The coverage function of the BNM can be used to order the ranking of document query pairs. A detailed description of this model, especially in respect to its connection to the *Boolean Model*, *Vector Space Model* [110] & *Probabilistic Model* [108] can be found in Sec. 3.5. The model can be defined as follows:

$$\begin{aligned} P(\boldsymbol{d},\boldsymbol{q}|\boldsymbol{\theta}) &= \sum_{u\in U} P(\boldsymbol{d},\boldsymbol{q},u|\boldsymbol{\theta}) \\ &= \sum_{u\in U} P(\boldsymbol{d},\boldsymbol{q}|u,\boldsymbol{\theta})\cdot P(u|\boldsymbol{\theta}) \\ &= \sum_{u\in U} P(\boldsymbol{d}|u,\boldsymbol{\theta})\cdot P(\boldsymbol{q}|u,\boldsymbol{\theta})\cdot P(u|\boldsymbol{\theta}) \end{aligned} \tag{7.1}$$

The *Information Search Behavior Profile Model* (ISBP Model) was formally described in Sec. 7.6 and decomposes into the *eye gaze*, *interaction* and *navigational model*. An in-depth analysis of the *Navigational & Interaction Model* can be found in Sec. 7.2 & Sec. 7.3, while the eye gaze model has been described for *Reading Strategies in User's Search Activities* in Sec. 7.4 & Sec.7.5. The entire fusion into the global network is based on the individual components and can be described by the following Likelihood:

$$P(\boldsymbol{s},\underline{\boldsymbol{x}}|\boldsymbol{\theta}) = P(s_1|\boldsymbol{\theta})\cdot\prod_{l=2}^{L_n} P(s_l|s_{l-1},\boldsymbol{\theta})\cdot\prod_{l'=1}^{L_n} P(\boldsymbol{x}_{l'}^{LF},\boldsymbol{x}_{l'}^{ET},\boldsymbol{x}_{l'}^{EM}|s_{l'},\boldsymbol{\theta}) \tag{7.2}$$

$$P(\boldsymbol{x}^{LF},\boldsymbol{x}^{ET},\boldsymbol{x}^{EM}|s,\boldsymbol{\theta}) = P(\boldsymbol{x}^{LF}|s,\boldsymbol{\theta})\cdot P(\boldsymbol{x}^{EM}|\boldsymbol{x}^{ET},\boldsymbol{\theta})\cdot P(\boldsymbol{x}^{ET}|s,\boldsymbol{\theta})$$

7.7.2 Sequential Ranking for Search Sessions

The Bayesian Network Model follows an independence assumption about the document query pairs. This assumption makes search sessions permutation invariant. The entire work of the previous sections assumed that sequential information matters, and all findings supported evidence for that claim. It is safe to state, that sequential information needs to be considered during the ranking. Therefore, Eq. (7.1) will be adapted for such context. The original Likelihood can be formulated via its entire joint probability distribution. For a paired sequence $(\underline{\boldsymbol{d}}, \underline{\boldsymbol{q}})$ of documents $\underline{\boldsymbol{d}}$ and queries $\underline{\boldsymbol{q}}$ of length L, the Likelihood can be formulated by using a sequence $\boldsymbol{u}$ over the 'universe of discourse':

$$P(\underline{\boldsymbol{d}}, \underline{\boldsymbol{q}}|\boldsymbol{\theta}) = \sum_{\boldsymbol{u}\in\boldsymbol{U}} P(\underline{\boldsymbol{d}}, \underline{\boldsymbol{q}}, \boldsymbol{u}|\boldsymbol{\theta})$$

Without loss of generality, the *product rule/chain rule* can be applied. This first triple of $(\boldsymbol{d}_1, \boldsymbol{q}_1, u_1)$ remains equal in distribution as in Eq. (7.1), while all following triples are conditioned on the entire history of triples:

$$P(\underline{\boldsymbol{d}}, \underline{\boldsymbol{q}}|\boldsymbol{\theta}) = \sum_{\boldsymbol{u}\in\boldsymbol{U}} P(\boldsymbol{d}_1, \boldsymbol{q}_1, u_1|\boldsymbol{\theta}) \prod_{l=2}^{L} P(\boldsymbol{d}_l, \boldsymbol{q}_l, u_l|\boldsymbol{d}_{l-1}, ...\boldsymbol{d}_2, \boldsymbol{q}_{l-1}, ...\boldsymbol{q}_2, u_{l-1}, ...u_2, \boldsymbol{\theta})$$

By further re-formulating with the *product rule/chain rule*, the Likelihood decomposes into the following:

$$P(\underline{\boldsymbol{d}}, \underline{\boldsymbol{q}}|\boldsymbol{\theta}) = \sum_{\boldsymbol{u}\in\boldsymbol{U}} P(\boldsymbol{d}_1, \boldsymbol{q}_1, u_1|\boldsymbol{\theta}) \cdot \prod_{l=2}^{L} P(\boldsymbol{d}_l, \boldsymbol{q}_l|\boldsymbol{d}_{l-1}, ...\boldsymbol{d}_2, \boldsymbol{q}_{l-1}, ...\boldsymbol{q}_2, u_l, u_{l-1}, ...u_2, \boldsymbol{\theta}) \cdot P(u_l|\boldsymbol{d}_{l-1}, ...\boldsymbol{d}_2, \boldsymbol{q}_{l-1}, ...\boldsymbol{q}_2, u_{l-1}, ...u_2, \boldsymbol{\theta})$$

The conditional independence assumption of the original model in Eq. (7.1) can be applied to further decompose the Likelihood:

$$P(\underline{\boldsymbol{d}}, \underline{\boldsymbol{q}}|\boldsymbol{\theta}) = \sum_{\boldsymbol{u}\in\boldsymbol{U}} P(\boldsymbol{d}_1, \boldsymbol{q}_1, u_1|\boldsymbol{\theta}) \cdot \prod_{l=2}^{L} P(\boldsymbol{d}_l, \boldsymbol{q}_l|u_l, \boldsymbol{\theta}) \cdot P(u_l|\boldsymbol{d}_{l-1}, ...\boldsymbol{d}_2, \boldsymbol{q}_{l-1}, ...\boldsymbol{q}_2, u_{l-1}, ...u_2, \boldsymbol{\theta})$$

The *Markov property* can be assumed over sequences in the 'universe of discourse' and a conditional independence in respect to document query pairs:

$$P(\underline{\boldsymbol{d}}, \underline{\boldsymbol{q}}|\boldsymbol{\theta}) = \sum_{\boldsymbol{u}\in\boldsymbol{U}} P(\boldsymbol{d}_1, \boldsymbol{q}_1, u_1|\boldsymbol{\theta}) \cdot \prod_{l=2}^{L} P(\boldsymbol{d}_l, \boldsymbol{q}_l|u_l, \boldsymbol{\theta}) \cdot P(u_l|u_{l-1}, \boldsymbol{\theta})$$

The *product rule/chain rule* can be applied for further decomposition:

$$P(\underline{\boldsymbol{d}}, \underline{\boldsymbol{q}}|\boldsymbol{\theta}) = \sum_{\boldsymbol{u}\in\boldsymbol{U}} P(\boldsymbol{d}_1, \boldsymbol{q}_1|u_1, \boldsymbol{\theta}) \cdot P(u_1|\boldsymbol{\theta}) \cdot \prod_{l=2}^{L} P(\boldsymbol{d}_l, \boldsymbol{q}_l|u_l, \boldsymbol{\theta}) \cdot P(u_l|u_{l-1}, \boldsymbol{\theta})$$

Finally, the conditional independence assumption of the original model in Eq. (7.1) can be applied again to formulate a quite familiar Likelihood:

$$P(\underline{\boldsymbol{d}}, \underline{\boldsymbol{q}}|\boldsymbol{\theta}) = \sum_{\boldsymbol{u}\in\boldsymbol{U}} P(\boldsymbol{d}_1|u_1, \boldsymbol{\theta}) \cdot P(\boldsymbol{q}_1|u_1, \boldsymbol{\theta}) \cdot P(u_1|\boldsymbol{\theta}) \cdot \prod_{l=2}^{L} P(\boldsymbol{d}_l|u_l, \boldsymbol{\theta}) \cdot P(\boldsymbol{q}_l|u_l, \boldsymbol{\theta}) \cdot P(u_l|u_{l-1}, \boldsymbol{\theta}) \tag{7.3}$$

The resulting Likelihood Eq. (7.3) is a sequential extension for the Likelihood Eq. (7.1) and it is visualized as a *Graphical Model* [88] in Fig. 7.21. In comparison to the original approach of Ribeiro-Neto et al. [106], rankings are not independent in their sequence but are connected via the 'universe of discourse' by a *Markov chain*. Simple first-order *Markov Models* were used just for clarity, but any higher-order Markov Model can be used for the argumentation as well. In general, this sequential ranking method can be described by *Hidden Markov Models* [99] and the classical Bayesian Network Model realizes the *emissions*. A user search session should be treated as a natural sequence where the 'universe of discourse' might change over time, but within a contextual manner. Such changes can be represented by a *transition model*, a *navigational model*. The ranking itself is done by a state specific *emission model*, an *interaction model* and potentially an *eye gaze model*.

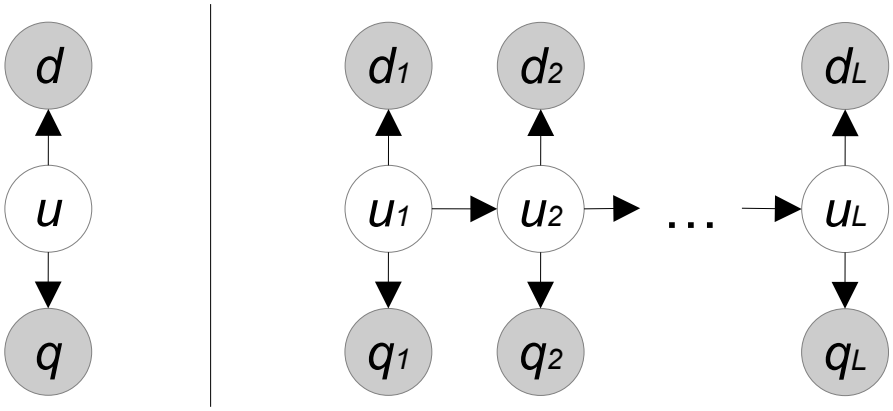


FIGURE 7.21: Left: The Graphical Model for the *Bayesian Network Model* (BNM) [106] (an instance of *Finite Mixture Models*). Right: The extension of the BNM towards a sequential ranking approach (an instance of *Hidden Markov Models* [99] with the BNM as an emission model). Nodes reflect model variables and edges their interaction: $A \rightarrow B = P(B|A)$. Latent (unknown) variables are marked white.

7.7.3 ISBP Models & Sequential Ranking

The connection between the sequential adaption of the Bayesian Network Model (BNM) in Eq. (7.3) and the ISBP Model in Sec. 7.6 will now be fully closed. The 'universe of discourse' U was initially described as consisting of a set of keywords. This will be extended to $U = (U_k, U_b)$, while U_k being the initial as a set of keywords and U_b being an additional as a set of *behavior keywords*. The entire previous analysis in Chap. 7, described *behavior keywords* without explicitly stating it, e.g. *Reading*, *Scanning* & *Skimming* in case of eye movement *strategies* or *State.Clicking*, *State.Scrolling* & *State.Duration* in case of log-file *actions*. This extension will be applied for all documents and queries, e.g. $\boldsymbol{d} = (\boldsymbol{d}_k, \boldsymbol{d}_b)$ & $\boldsymbol{q} = (\boldsymbol{q}_k, \boldsymbol{q}_b)$ with $\boldsymbol{d}_k$ & $\boldsymbol{q}_k$ being the set of keywords and $\boldsymbol{d}_b$ & $\boldsymbol{q}_b$ being the set of *behavior keywords* the document and query was exposed to by users. The Likelihood in Eq. (7.3) will be revisited under this perspective:

$$P(\underline{\boldsymbol{d}}, \underline{\boldsymbol{q}}|\boldsymbol{\theta}) = \sum_{\boldsymbol{u} \in \boldsymbol{U}} P(\boldsymbol{d}_1|u_1, \boldsymbol{\theta}) \cdot P(\boldsymbol{q}_1|u_1, \boldsymbol{\theta}) \cdot P(u_1|\boldsymbol{\theta}) \cdot \prod_{l=2}^{L} P(\boldsymbol{d}_l|u_l, \boldsymbol{\theta}) \cdot P(\boldsymbol{q}_l|u_l, \boldsymbol{\theta}) \cdot P(u_l|u_{l-1}, \boldsymbol{\theta})$$

- by applying the *Naive Bayes Model* on $U = (U_k, U_b)$ with BNM & ISBP:
  $\rightarrow P(u_l|u_{l-1}, \boldsymbol{\theta}) = P(u_{k,l}|\boldsymbol{\theta}) \cdot P(u_{b,l} = s_l|u_{b,(l-1)} = s_{l-1}, \boldsymbol{\theta})$

  Represents a *navigational model* in combination with the measurement of the original 'universe of discourse' U_k as a set of keywords. The original ranking model Eq. (7.1) assumes an (uninformative) *Uniform Distribution*. In respect to the interpretation of *behavior keywords*, it represents the navigational trails or *strategies* the user implements during a search session. Previous sections provided an in-depth analysis of *strategies* indicative for search activities. This resulted in (User) Behavior Models that evaluate characteristic profiles.

- by applying the *Naive Bayes Model* on $U = (U_k, U_b)$ with BNM & ISBP:
  $\rightarrow P(\boldsymbol{q}|u, \boldsymbol{\theta}) = P(\boldsymbol{q}_k|u_k, \boldsymbol{\theta}) \cdot P\big(\boldsymbol{q}_b = (\boldsymbol{x}^{LF}, \boldsymbol{x}^{ET}, \boldsymbol{x}^{EM})|u_b = Q, \boldsymbol{\theta}\big)$
  $\rightarrow P(\boldsymbol{d}|u, \boldsymbol{\theta}) = P(\boldsymbol{d}_k|u_k, \boldsymbol{\theta}) \cdot P\big(\boldsymbol{d}_b = (\boldsymbol{x}^{LF}, \boldsymbol{x}^{ET}, \boldsymbol{x}^{EM})|u_b = P, \boldsymbol{\theta}\big)$

  Represents the *eye gaze* & *interaction model* specific to the state *Query/Page* in combination with the keyword analysis described in Sec. 3.5. In respect to the interpretation of *behavior keywords*, it represents the profiles of *actions* & *strategies* the user implements during *Query/Page*. Previous sections provided an in-depth analysis of aspects indicative for search activities. This resulted in (User) Behavior Models that evaluate characteristic behavior profiles.

In this perspective, one can recognize the Likelihood of Eq. (7.3) as a realization of an abstract ISBP Model, which complements the classical approach with additional *behavior keywords*. Therefore, the combined approach extends Eq. (7.1), which works purely on explicit keywords, for behavior aspects in form of implicit *behavior keywords*. This is the foundation of what will be introduced as the *ISBP Ranking* method.

7.7.4 ISBP Ranking: Proof of Concept

The immediate question arises, if such a *ISBP Ranking* method works. Therefore, a special case for ranking with (User) Behavior Model, namely the (independent or *naive*) *ISBP Ranking*, will be analyzed in full detail. The classical ranking approach in Eq. (7.1) is used independently with the proposed (User) Behavior Model. For that, the joint probability distribution of both models needs to be stated. With all assumptions given in Eq. (7.3), the document-query space needs to be extended with the search history for the *navigational model*:

$$P(\boldsymbol{d} = (\boldsymbol{d}_k, \boldsymbol{d}_b), \boldsymbol{q} = (\boldsymbol{q}_k, \boldsymbol{q}_b), u_{b,l-1}) = \sum_{u \in U} P(\boldsymbol{d}, \boldsymbol{q}, u, u_{b,l-1})$$

The same *sum-rule/marginalization* over the 'universe of discourse' will be applied as in the case of the original ranking model. The 'universe of discourse' was extended into two subspaces, e.g. $U = (U_k, U_b)$, and therefore the probability distribution can be extended by explicitly stating the subspace by using its respected indices:

$$P(\boldsymbol{d}, \boldsymbol{q}, u_{b,l-1}) = \sum_{u_k \in U_k} \sum_{u_b \in U_b} P(\boldsymbol{d}_k, \boldsymbol{d}_b, \boldsymbol{q}_k, \boldsymbol{q}_b, u_k, u_b, u_{b,l-1})$$

By exploiting the independence assumption, the distribution decomposes into their respected sub-model distributions:

$$\begin{aligned} P(\boldsymbol{d}, \boldsymbol{q}, u_{b,l-1}) &= \sum_{u_k \in U_k} \sum_{u_b \in U_b} P(\boldsymbol{d}_k, \boldsymbol{q}_k, u_k) \cdot P(\boldsymbol{d}_b, \boldsymbol{q}_b, u_b, u_{b,l-1}) \\ &= \sum_{u_k \in U_k} P(\boldsymbol{d}_k, \boldsymbol{q}_k, u_k) \cdot \sum_{u_b \in U_b} P(\boldsymbol{d}_b, \boldsymbol{q}_b, u_b, u_{b,l-1}) \end{aligned}$$

The Likelihood of both models applied independently, decompose into the Likelihood of both models applied separately. The term on the left side is identically to Eq. (7.1). The additional index k simply highlights the separation of the 'universe of discourse' for U_k. The term on the right side is the proposed (User) Behavior Model. In comparison to the term on the left, the (User) Behavior Model uses previous information indicated by the index $l-1$ via a *navigational model*. The index b highlights the separation of the 'universe of discourse' for U_b.

The ranking of documents $\boldsymbol{d}$ follows the sorted probabilities of the distribution $P(\boldsymbol{d}|\boldsymbol{q}, u_{b,l-1})$ given a provided query $\boldsymbol{q}$ and previous search session information (indicated by the index $l-1$). To gain some insights about this ranking distribution, a reformulation of it is needed via the *Bayes Theorem* and the previous joint probability distribution:

$$P(\boldsymbol{d}|\boldsymbol{q}, u_{b,l-1}) = \frac{P(\boldsymbol{d}, \boldsymbol{q}, u_{b,l-1})}{P(\boldsymbol{q}, u_{b,l-1})}$$

By sum-rule/marginalization over all missing information, the postulated joint distribution can be further reformulated:

$$P(\boldsymbol{d}|\boldsymbol{q}, u_{b,l-1}) = \frac{\sum_{u \in U} P(\boldsymbol{d}, \boldsymbol{q}, u, u_{b,l-1})}{\sum_{\boldsymbol{d}' \in \boldsymbol{D}, u' \in U} P(\boldsymbol{d}', \boldsymbol{q}, u', u_{b,l-1})}$$

By further exploiting the independence assumption, the distribution can be further simplified:

$$P(\boldsymbol{d}|\boldsymbol{q}, u_{b,l-1}) = \frac{\sum_{u_k \in U_k} P(\boldsymbol{d}_k, \boldsymbol{q}_k, u_k)}{\sum_{d'_k \in \boldsymbol{D}_k, u'_k \in U_k} P(\boldsymbol{d}'_k, \boldsymbol{q}_k, u'_k)} \cdot \frac{\sum_{u_b \in U_b} P(\boldsymbol{d}_b, \boldsymbol{q}_b, u_b, u_{b,l-1})}{\sum_{d'_b \in \boldsymbol{D}_b, u'_b \in U_b} P(\boldsymbol{d}'_b, \boldsymbol{q}_b, u'_b, u_{b,l-1})}$$

By resolving the sum-rule/marginalization, all terms increase in visual clarity:

$$P(\boldsymbol{d}|\boldsymbol{q}, u_{b,l-1}) = \frac{P(\boldsymbol{d}_k, \boldsymbol{q}_k)}{P(\boldsymbol{q}_k)} \cdot \frac{P(\boldsymbol{d}_b, \boldsymbol{q}_b, u_{b,l-1})}{P(\boldsymbol{q}_b, u_{b,l-1})}$$

To finalize the reformulation, the Bayes Theorem is applied as follows:

$$P(\boldsymbol{d}|\boldsymbol{q}, u_{b,l-1}) = P(\boldsymbol{d}_k|\boldsymbol{q}_k) \cdot P(\boldsymbol{d}_b|\boldsymbol{q}_b, u_{b,l-1})$$

The ranking distribution $P(\boldsymbol{d}|\boldsymbol{q}, u_{b,l-1})$ has nice properties. The term on the left is the ranking distribution of the classical model Eq. (7.1). The term on the right is the inference of the proposed (User) Behavior Model. It represents the expected search activity on document $\boldsymbol{d}_b$ given the observed behavior during the querying process $\boldsymbol{q}_b$ via the *interaction model* (& *eye gaze model*) in combination with the previous behavior history within the search $u_{b,l-1}$ via the *navigational model*. The independence assumption between both models results in the neat interpretation that the classical ranking approach is weighted by the (User) Behavior Model. To answer the question if the *ISBP Ranking* works: yes. In case of the classical approach in Eq. (7.1), an (uninformative) Uniform distribution is assumed. This directly transforms into a uniform (User) Behavior Model. Therefore, the classical approach already is a special instance of the proposed *ISBP Ranker*. It can be assumed that search activity aware ranking approaches could increase the user support because search activities comprise a specific *goal*. In case of Exploratory and Fact-Finding search activities, such goals are either categories as being *open* or *close*. A ranking weighted by the respected search activity will therefore be guided towards the contextualized *goal* indirectly inferred by the *behavior* of the user during the search session.

It is possible to further exploit aspects of the proposed (User) Behavior Model within the *ISBP Ranking* method. The classical model takes advantage of knowing the keywords of all documents in form of $\boldsymbol{d}_k$. Based on this knowledge, the probability can be calculated by the ranker for $P(\boldsymbol{d}_k|\boldsymbol{q}_k)$. The (User) Behavior Model can take advantage of the knowledge about search activities from previous users within that document in the form of $\boldsymbol{d}_b$. Instead of solely weighting the ranking by the expected search activity of the current user $P(\boldsymbol{d}_b|\boldsymbol{q}_b, u_{b,l-1})$, the ranking could be weighted by the observed *behavior keywords* of previous users on these documents. This approach can be seen as the equivalent scenario as in case of knowing the textual keywords. In case such a document was never exposed to previous search activities, then the expected search activity is an educated guess. In case the document was multiply exposed to search sessions, then this exposure could be used in a page specific model $P(\boldsymbol{d}_b|\boldsymbol{q}_b, u_{b,l-1})$, such as they are use in the model $P(\boldsymbol{d}_k|\boldsymbol{q}_k)$. This will enable the model to fully exploit the capabilities of the *ISBP Ranking*.

7.7.5 ISBP Ranking: Just in Words and Pictures

The classical ranking approach in Eq. (7.1) is an effective computer science approach. Given an explicit formulated input, the machine reacts with a sorted list of explicitly formulated output. All manifested in form of 'hard' information derived by keywords. A user, a human being, unfortunately is not a machine. Whenever a human explicitly formulates or expresses such 'hard' information, a human transfers this information embedded within multiple other channels. Implicitly, a human has a particular way on how this information is formulated and a particular context in which this information is expressed in. These additional channels are referred to as behavior within that thesis. The *ISBP Ranking* tries to capture a glimpse of this additional information in the form of *behavior keywords*.

Whenever a query is formulated, this query is not a singular entity. This query is formulated in the context of the entire search. Within that search, the user implements several *actions* and *strategies* that are evaluated by respected (User) Behavior Models. Within the *Query* state, a user implements more *actions* than simply formulating the query itself. A user takes a certain amount of time for that formulation, an *action* that takes a certain amount of time. Longer time duration might indicate a higher cognitive load. During this formulation, the user might already be on a search engine result page and the query formulation might be inspired by aspects read on it. Such *actions* have been observed, and specific eye movement *strategies* have been reported to be associated to specific search activities. All in all, this results in a plethora of additional information provided by a multitude of multi-modal channels the user implements during the search. The mentioned channels will be analyzed by the proposed mathematical & computational (User) Behavior Model. The *ISBP Ranking* builds upon these models to weight the ranking according to such behavior aspects. Fig. 7.23 illustrates the described method. The bottom part of the graphics represents the classical ranking approach. A user provides 'hard' information derived by keywords as a query to a ranking machine. The ranker uses its ranking distribution, e.g. Eq. (7.1), to create a sorted list of suitable documents for that query. In addition, the *ISBP Ranker* provides (User) Behavior Models to weight the ranking according to certain 'soft' information derived from *behavior keywords*. These *behavior keywords* are abstractly correlated with search activities. This analysis is implemented by the combination of the *eye gaze model*, the *interaction model* & the *navigational model*. In a certain kind of perspective, one could state that the classical model evaluates the *explicit* information transfer by textual keywords, while the *ISBP Ranker* complements this evaluation by the *implicit* information transfer by behavior measured by *behavior keywords*.

The entire potential of this idea is not yet fully exploited. Further, measurements of behavior information derived from previous search sessions of users, e.g. derived from logging files and/or eye-tracking data, can be exploited for such rankings, see Fig. 7.22. The *ISBP Ranker* could additionally exploit this knowledge. Some web pages might be more suitable for Fact-Finding search activities, while others are more suited for Exploratory search activities. The importance of certain web pages might be just appreciated by the user given a certain context within the search. The entire (behavior) history of previous user searches could be exploited by the *ISBP Ranker* to guide the current user by the analysis of suitable behavior similarities. Fully exploiting this knowledge from past users and search sessions could effectively boost the capabilities and potential of the *ISBP Ranker*.

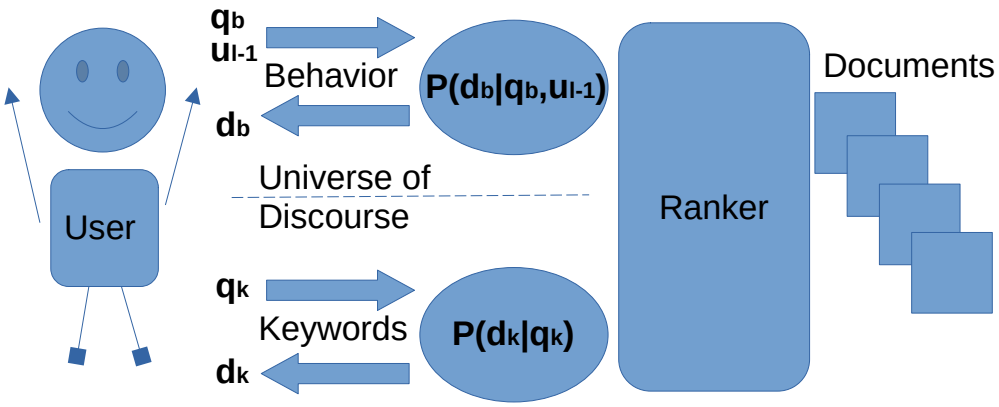


FIGURE 7.22: ISBP Ranking: A user provides a query q to a ranking system, which returns a sorted list of documents d suited to the query. While the bottom part realizes the classical ranking approach purely on 'textual' keywords (q_k, d_k). The ISBP Ranking complements this ranking with (q_b, d_b) additional behavior keywords within the previous search session context u_{l-1}.

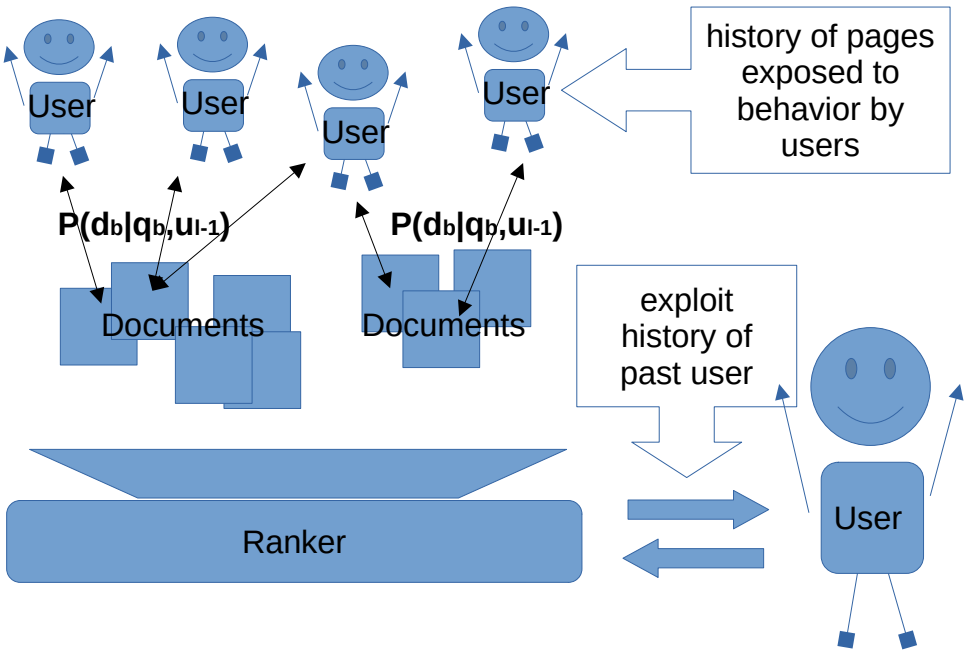


FIGURE 7.23: ISBP Ranking: With observed behavior keywords from previous users in past search sessions, the history of web pages exposed to user behavior can be exploited in the ISBP Ranking. For a current query event, the search session can be represented by the current behavior model of the user. In addition to classical ranking approaches purely working on keyword level, the ISBP Ranking complements the ranking with behavior keywords. Therefore, the ranking can adapt towards the user's search activity.

7.7.6 Conclusion

A theoretical framework for ranking in Information Retrieval was proposed that fully exploits the potential of the previously introduced *(User) Behavior Models*. This framework is called the *Information Search Behavior Profile ranking*, e.g. *ISBP Ranker*. The *ISBP Ranker* builds upon the classical ranking approach via the Bayesian Network Model, which is the generalization of ranking with the Boolean Model, the Vector Space Model & the Probabilistic Model. The *ISBP Ranker* does not re-invent ranking, but extends it with new perspectives. Classical ranking approaches work on query & document pairs without considering any sequential information, and evaluation of these pairings is purely done on the keyword level. This new approach extends the analysis for sequential information and *behavior keywords* during search sessions. For that, the introduced (User) Behavior Models are applied to model the search sessions and extend the ranking. All in all, the *ISBP Ranker* considers *behavior keywords* in form of the search session history via the *navigational model* and *actions* & eye movement *strategies* executed by the user via the *interaction model* & the *eye gaze model*. The theoretical analysis shows that this approach realizes a behavior-driven re-weighting of the classical ranking approach via the Bayesian Network Model. This weighting scheme realizes a natural re-weighting according to the behavior aspect derived by the proposed *(User) Behavior Models*. These models were specifically design to infer Exploratory and Fact-Finding search activities. With a reasonable high detection rate of 89.32% *Accuracy*, it can be assumed that the ranking can be weighted adequately in favor of the suitable search activity context and their underlying *goals*. The full potential of the *ISBP Ranker* can be exploited, when an additional database can be accessed. Such a database comprises the entire history of web pages exposed to user search behavior during previous searches. The *ISBP Ranker* can exploit this knowledge by taking advantage of the history of user search behavior on web pages and adequately guide the current user given the individual search activity and search session context.

7.8 Proceed with Unlabeled Data

This section aims to further explore the potential of the proposed *(User) Behavior Models*, namely the *Information Search Behavior Profile Model* (ISBP Model). A particular focus in the following section lies on the open issue to work with unlabeled data. Especially in the context of *Ranking with Information Search Behavior Profiles* (ISBP Ranker), this remains a limiting factor towards its practical implementation. The work was designed to achieve two goals simultaneously. First, the potential of the ISBP Model is explored and evaluated in the scenario of unlabeled data. Labeled data are usually not given, and the process of annotating them is time and cost consuming. With models capable of clustering data into clusters representing search activities, the applicability of such models increases. Second, the models will be applied on a new and independent data set to derive characteristics for search activities. If characteristics derived from both data sets coincide or are at least comparable, it can be argued that the ISBP Model indeed captures preserved behavior aspects in user search activities. Additionally, these findings will complement previous ones and significantly enhance their plausibility and evidence. If and only if both goals are reached, an implementation of the ISBP Ranker seems plausible. The entire following section can be seen as a summary of my research work in *Fact-Finding or Exploration: Identifying Latent Behavior Clusters in User's Search Activities* [115]. The aim of this research work is directly motivated by **RQ2** and **RQ3**. Further, it aims to support findings for **RQ1**.

7.8.1 Task Description

Even though, most of the definitions remain the same as in the previous sections, minor adaptions will be used because of experience gain during the initial study in Sec. 6 and its analysis in Sec. 7.2 & 7.3.

7.8.1.1 User Study

A user study of bigger magnitude was conducted to complement the results of the previous one. In essence, the study design remains comparable to the one in Sec. 6. The same two *Exploratory* (*Expl*) tasks were used but the *Fact-Finding* (*Fact*) tasks were extended to 117 tasks from different domains, such as sports, natural science, geography, technology, literature, history, movies and music. A user was presented with up to 16 randomly chosen *Fact* tasks and encouraged to answer them. Therefore, the user study increased in size with 717 *Fact* search sessions and 226 *Expl* search sessions. A broader range of participants could be collected, with 76 women and 39 men. The mean age is 26.78 (min = 17, max = 63) and the majority of 73 participants are students, 16 reported to have jobs in a variety of fields, 6 are still in schooling, 2 are in retirement, 6 are unemployed and 12 refused to give information about their current status. The entire user study comprises more experimental designs, which were used by other research parties, and this description focuses just on a subsection of it. Minor differences to the previous user study arise from the fact that participants received a small monetary reward for their participation. Further, participant recruiting was realized via social networks, bulletin boards and at supermarkets.

7.8.1.2 Data Definition

The *Data Definition* in Sec. 7.6.1.1 remains valid. During the initial study in Sec. 6 experience was gained and minor adaptions in the data recordings were implemented. The sequence of user interactions with a search engine and a web browser were reduced as a result of *Addressing Experimental Limitations*, and the following were applied:

- *Query* (Q):
  A searcher is formulating a search query on *Google*'s search engine by entering the query or using auto-complete suggestions.
- *SERP* (S):
  A searcher is examining *Google*'s search engine result page (SERP). Usually *SERP*s occur directly after *Query*, but they can also be accessed by changing the active tab.
- *Page* (P):
  A searcher is viewing a web page not categorized by the previous states.

Further on, each state consists of a set of *actions* and *strategies* a user executes on it. Even though, the data collection comprises all previously mentioned aspects, results of previous work while *Combining Eye Tracking and Navigation* indicated that not all associated features are equally important to identify search activities. With *State.Duration* being a statistically significant indicative *action* on the state *Page*, the following approach will exclusively work with that feature. Because of the varying quality in the distributional fit during previous approaches, the feature space is binned into sub-regions. Therefore, more parameters can be used to adapt to certain sub-regions. The following binning scheme has been used to measure the time 0, 1, 2, ..., 10, 20, 30, ..., 60 and > 60 in seconds. Measurements in such bins result in only one component being 1 while the rest is 0. This procedure is also known as the *one-in-hot encoding*. The previous modeling approach in Sec. 7.3 allowed *Page.Duration* being modeled by an *Exponential Distribution*, e.g. a one parameter model. The described binning allows the *Page.Duration* being modeled by a *Multinomial Distribution*, where each bin will be described by one parameter. This approach has the potential to increase the quality of the distributional fit because of the increasing parameter space.

7.8.1.3 Model Definition

Up to this point, the model family of *Hidden Markov Models* (HMM) [99] have been used extensively for the analysis of search activities. Nonetheless, its primary strength, namely working with unlabeled data (*Unsupervised Learning*), has not been exploited yet. In the classical way, the underlying *Markov Models* in HMMs are not observable. In the described scenario here, it is. In case of unlabeled search sessions, the entire sequence comprises a missing assignment. With HMM being *Generative Models*, they can easily be combined with *Finite Mixture Models* (FMM), to infer latent assignments. The learning algorithm of FMM in combination with *base distributions* of HMMs does not result in the *Baum-Welch Algorithm*, see Sec. 3.3.4, but to the *Expectation Maximization* (EM) [31] algorithm for entire sequences. Therefore, the Likelihood of a *Mixture of Hidden Markov Models* (MHMM) can easily be defined as follows:

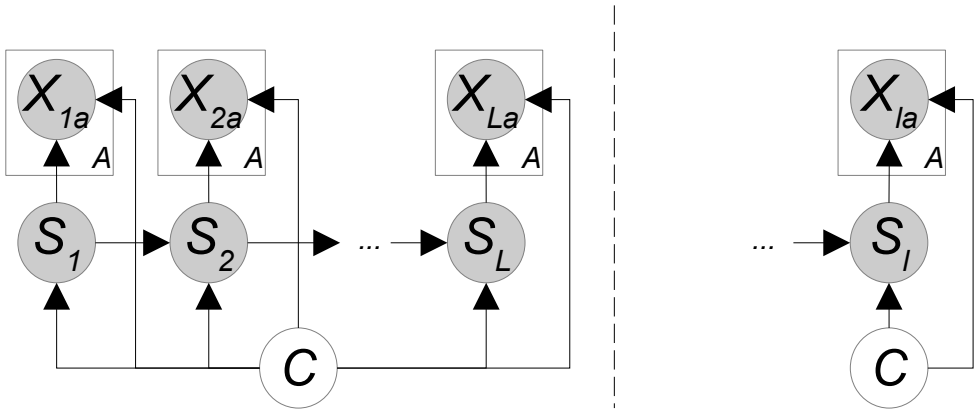


FIGURE 7.24: The Graphical Model of the proposed approach is structurally equivalent to the ones in Sec. 7.2 Fig. 7.2 and Sec. 7.3 Fig. 7.6 except of the (unknown) latent search activity assignments.

$$
\begin{aligned}
P(\underline{z}|\boldsymbol{\theta}) &= \prod_{n=1}^{N} P(\boldsymbol{z}_n|\boldsymbol{\theta}) \\
&= \prod_{n=1}^{N} \sum_{c \in C} P(\boldsymbol{z}_n, c|\boldsymbol{\theta}) \\
&= \prod_{n=1}^{N} \sum_{c \in C} P(c|\boldsymbol{\theta}) \cdot P(\boldsymbol{z}_n|c, \boldsymbol{\theta}) \\
&= \prod_{n=1}^{N} \sum_{c \in C} P(\boldsymbol{\theta}_c) \cdot P(\boldsymbol{z}_n|\boldsymbol{\theta}_c) \\
&= \prod_{n=1}^{N} \sum_{c \in C} P(\boldsymbol{\theta}_c) \cdot P(\boldsymbol{s}_n, \underline{\boldsymbol{x}}_n|\boldsymbol{\theta}_c) \\
&= \prod_{n=1}^{N} \sum_{c \in C} P(\boldsymbol{\theta}_c) \cdot P(s_{n1}|\boldsymbol{\theta}_c) \cdot \prod_{l=2}^{L_n} P(s_{nl}|s_{n(l-1)}, \boldsymbol{\theta}_c) \cdot \prod_{l=1}^{L_n} P(\boldsymbol{x}_{nl}|s_{nl}, \boldsymbol{\theta}_c)
\end{aligned}
$$

where c encodes the cluster assignment, e.g. a postulation for search activities $c := \{Expl, Fact\}$. $P(\boldsymbol{\theta}_c)$ denotes the mixture coefficients regarding such assignments, and $P(\boldsymbol{z}_n|\boldsymbol{\theta}_c)$ represents the HMM generating a certain sequence $\boldsymbol{z}_n$ by the given assignment model $\boldsymbol{\theta}_c$. The *Posterior Distribution* can be used for cluster assignments as follows:

$$
P(\boldsymbol{\theta}_c|\boldsymbol{z}_n) = \frac{P(\boldsymbol{z}_n|\boldsymbol{\theta}_c) \cdot P(\boldsymbol{\theta}_c)}{P(\boldsymbol{z}_n)} = \frac{P(\boldsymbol{z}_n|\boldsymbol{\theta}_c) \cdot P(\boldsymbol{\theta}_c)}{\sum_{c'} P(\boldsymbol{z}_n|\boldsymbol{\theta}_{c'}) \cdot P(\boldsymbol{\theta}_{c'})}
$$

With all these definitions provided, the EM for MHMM is fully defined. The factorization of the Likelihood is illustrated in the *Graphical Model* [88] in Fig. 7.24 and can directly been used in the EM, see Algo. 2:

$$
Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) = \sum_{\boldsymbol{c} \in \boldsymbol{C}} P(\boldsymbol{\theta}_{\boldsymbol{c}}^{old}|\underline{\boldsymbol{z}}) \cdot \ln P(\underline{\boldsymbol{z}}, \boldsymbol{\theta}_{\boldsymbol{c}})
$$

7.8.2 Classification via ISBP Models

First, the classification results are needed as a baseline comparison for the subsequent clustering approach. Therefore, the HMM classifier with given search activity assignments is evaluated. The performance was measured via *Cross-Validation* [125] averaged over 2,000 repeats and the resulting *Confusion Matrix* can be seen in Tab. 7.9. The classifier achieves an *Accuracy* of 88.58%. Although working on another data set, the observed performance is nearly identical to the approach in Sec. 7.3 with 89.32% on another data set. With the class imbalance of 717 *Fact* and 226 *Expl* search activity assignments, the *F-Score* was additionally calculated. For *Fact* search activities, the model produces a score of 0.925. In contrast, the score of 0.754 in *Expl* search activities is comparably low. However, it can be argued that the model recognizes search activities sufficiently good to draw conclusions from it. Fig. 7.25 visualizes the characteristics of the model, namely the *stationary distribution* of the underlying Markov chain, see Sec. 3.3.1, and the duration distribution of *Page* dwell-times. The stationary distribution represents the relative proportion of each state in a long-run behavior. Accordingly, users predominantly interact with *Page* during *Expl* search activities. In contrast, users in *Fact* search activities interact more homogeneous with all states but with a preference towards *SERP*. This finding complements the results of the *navigational model* in Sec. 7.2.5 (Fig. 7.5) in a new and independent user study with increased size. Participants in both user studies realize comparable navigational pattern measured by the stationary distribution. Further on, by analyzing the ratio between both activities in respect of *Page* dwell-times, the shift between both modi can be located on 7 seconds. This implies that a user infers the usefulness of a web page in average under 7 seconds within a *Fact* search activity, but takes longer than 7 seconds on average for *Expl* search activities. This finding complements results of the *interaction model* in respect to *Page* dwell-times in Sec. 7.3 (Fig. 7.7) with a cut point of ca. 10 seconds in the other reference user study. The resulting characteristics derived from this ISBP model are quite comparable between both user studies.

Design	Prediction	
	Fact	*Expl*
Fact	670.192	46.808
Expl	60.862	165.138

TABLE 7.9: Adapted from [115]: *Confusion Matrix* comprising the averaged results of 2,000 repeated 5-fold *Cross-Validation* [125] for the *Hidden Markov Models* [99] classifier.

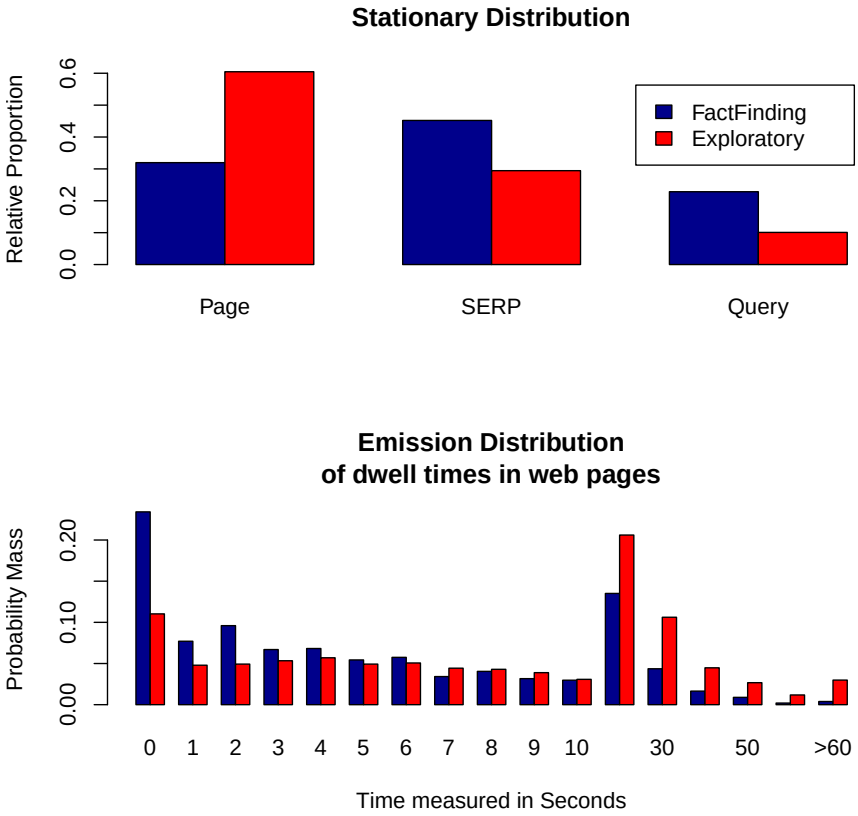


FIGURE 7.25: Adapted from [115]: Model properties for Exploratory (red) and Fact-Finding (blue) search activity assignments. The upper plots illustrate the stationary distributions of the *Markov Models*, see Sec. 3.3.1, while the lower ones represent the web page dwell-times.

7.8.3 Clustering via ISBP Models

With a baseline given, the clustering approach can be evaluated in that reference. Therefore, the MHMM model is used in a clustering setting with two components to identify search activity clusters without any session-to-assignment information, i.e. the class label information for the corresponding search session is omitted. Fig. 7.26 illustrates the iterative clustering progression of the EM algorithm, see Algo. 2, for the MHMM with two components. The Likelihood progression depends on the model's initialization, therefore 100 random starting points have been used to observe the convergence. The *(incomplete-data) Likelihood* is maximized until the incremental increase dropped under a predefined threshold. Instead of the *Maximum Likelihood Estimate* [42] of the EM, the *Maximum A Posteriori* version of the EM was used. The reasoning for that choice was already described in Sec. 7.2, see Fig. 7.4. As all runs converge to the same plateau, it can be concluded that there is little evidence for other cluster separations on 2 components. The MAP estimate was selected via the maximum of multiple repeats. For each search session, the expectation of the latent variable was used for the assignment to its hypothesized search activity. The *contingency table* illustrated in Tab. 7.10 shows the experimental assignment and the cluster prediction. The agreement of 79.1% with the experimental design allows inference about implicit semantics from the proposed activity clusters.

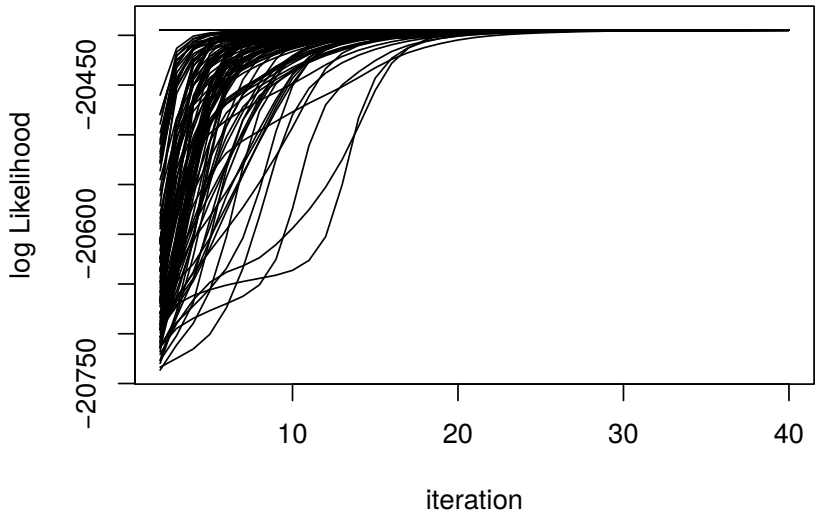


FIGURE 7.26: Adapted from [115]: *Expectation Maximization* [31] progression on 100 random initializations for the Mixture of Hidden Markov Model of two cluster components.

$Cluster_1$ consists of 90% *Fact* search activities, making it arguably the cluster representation of this search activity. $Cluster_2$ is less than a half of $Cluster_1$'s size and contains 45% *Fact* search activities. Although there is a lack of significant evidence, this cluster can be postulated as the representation of the $Expl$ search activity. The same characteristics as in Sec. 7.8.2 are visualized for the MHMM in Fig. 7.27. A striking similarity between both approaches can be observed. In $Cluster_2$ ($Expl$) users have an increased orientation towards *Page*, while in $Cluster_1$ (*Fact*) users interact more homogenous but with a preference towards *SERP*. Nearly the same similarity holds true for the duration distribution. In $Cluster_2$ users have the tendency to spend more than 8 seconds on *Page* while users in $Cluster_1$ spend less than 5 seconds on *Page*. The resulting characteristics derived from ISBP model are nearly similar as in

case of *Supervised Learning* & *Unsupervised Learning*. Based on this analysis, it can be argued that search activities can be identified by clustering ISBP models, even when search activity assignments are missing. The possibility to identify $Expl$ and $Fact$ search activities purely on the data itself indicates highly conserved search activity pattern in participants that can be captured by the *navigational model* and the *interaction model*. The abstract correlation in the clusters and the experimental assignments indicate that in case of missing search activity assignments, the described clustering approach is a reasonable option when either no experimental design is given or manual annotation is impracticable.

Design	Prediction	
	$Cluster_1$	$Cluster_2$
$Fact$	585	132
$Expl$	65	161

TABLE 7.10: Adapted from [115]: Contingency table representing the cluster assignments to the experimental design (the 'true' classes) for the Mixture of Hidden Markov Model clustering approach.

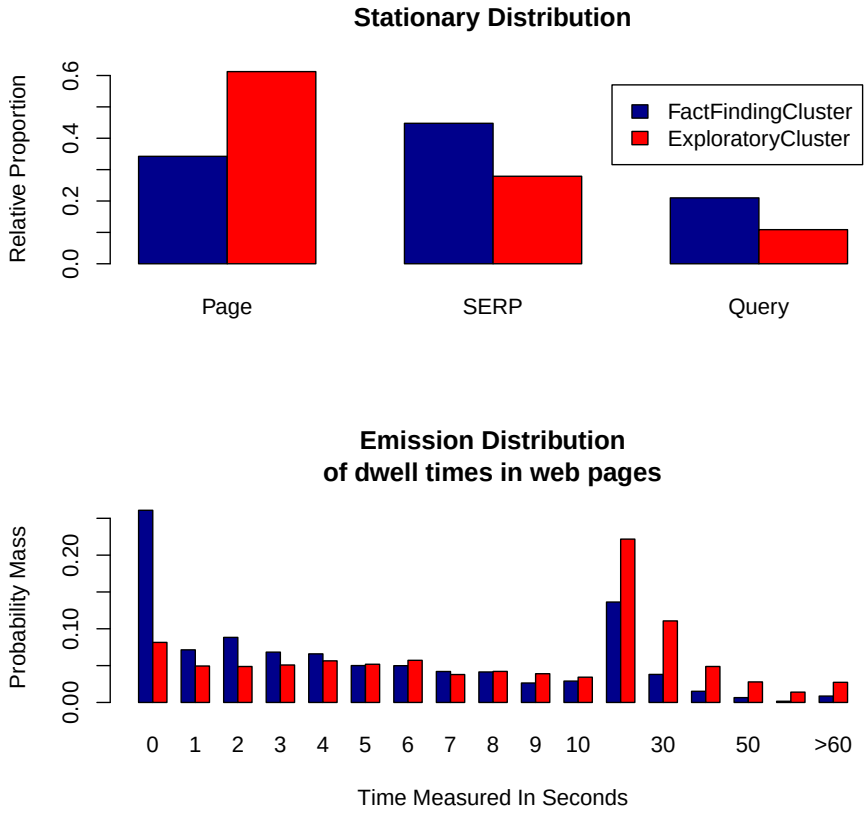


FIGURE 7.27: Adapted from [115]: Model properties for Exploratory (red) and Fact-Finding (blue) search activity clusters. The upper plots illustrate the stationary distributions of the *Markov Models*, see Sec. 3.3.1, while the lower ones represent the web page dwell-times.

7.8.4 Identification of latent ISBP Models

In the previous two experiments, a two group scenario were investigated. However, such prior assumptions might not always be given, or even worse, not even be true. The following analysis will therefore abandon any restrictions in the experimental design and purely work on the data itself. For that, a reasonable grid of latent cluster, e.g. hypothesized search activities, will be investigated. Different MHMM's for components ranging from 1 to 6 are trained, and model selection is applied via the *Bayesian Information Criterion* [112]. According to the selection score, the model with three components has the most support, see Tab. 7.11. Furthermore, the second-best model has a noticeable difference of $\Delta_i > 10$ to the best model, implying that alternatives have essentially no support according [20]. Fig. 7.28 illustrates the clustering progression for three components. The Likelihood progression provides little evidence for other cluster separations on 3 components, indicated by a convergence to the same *(incomplete-data) Likelihood* plateau. The resulting *contingency table* between experimental design and cluster assignments is illustrated in Tab. 7.12.

Components	Parameters	Log-Likelihood ($P(\underline{z}\|\boldsymbol{\theta})$)	BIC
1	30	-20667.509	41540.491
2	60	-20343.942	41098.829
3	**90**	**-20223.716**	**41063.849**
4	120	-20159.884	41141.656
5	150	-20114.927	41257.215
6	180	-20073.690	41380.213

TABLE 7.11: Adapted from [115]: Model selection via the *Bayesian Information Criterion* (BIC) [112] for cluster components from 1 to 6. Model parameter and its Log-Likelihood is provided as well.

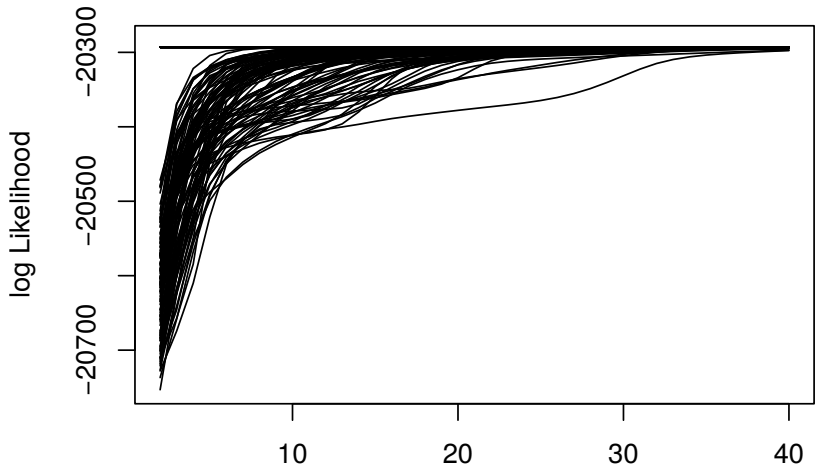


FIGURE 7.28: Adapted from [115]: *Expectation Maximization* [31] progression on 100 random initializations for the Mixture of Hidden Markov Model for three cluster components.

The same method of reasoning to characterize the clusters was used as in the previous experiment. $Cluster_1$ consists of 95.726% *Fact* search activities and is relatively homogeneous. $Cluster_2$ is less than half of $Cluster_1$'s size and contains 45.544% *Fact*

search activities. $Cluster_3$ has more of half the size of $Cluster_1$ and 64.835% *Fact* activity assignments. Therefore, $Cluster_1$ will be postulated as the representation of *Fact* search activities, $Cluster_2$ as *Expl* search activities and $Cluster_3$ as a hybrid (or borderline) search activity based on the evidence in the contingency table. The same characteristics as in the previous experiments are visualized in Fig. 7.29 to interpret the clusters. In respect to its characteristics, $Cluster_1$ & $Cluster_2$ (*Fact* & *Expl*) are nearly identical to the previous approaches. $Cluster_3$ seems to represent a hybrid of both. While in its *navigational model* it is comparable to an *Expl* search activity, its *interaction model* is more representative for a *Fact* search activity. Even though, there is a lack of clear evidence, it seems reasonable to assume that this cluster is populated by either users with highly advanced search skills or simply quite fast ones.

Design	Prediction		
	$Cluster_1$	$Cluster_2$	$Cluster_3$
Fact	448	92	177
Expl	20	110	96

TABLE 7.12: Contingency table of the experimental design and cluster assignments for the Mixture of Hidden Markov Model clustering approach.

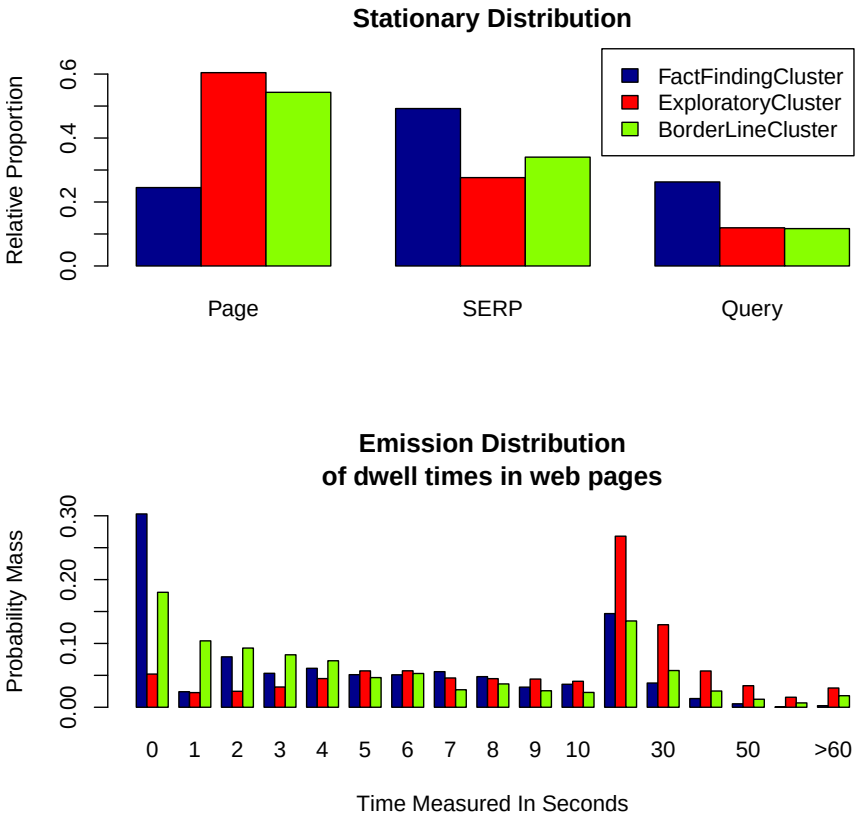


FIGURE 7.29: Taken from [115]: Model properties for Exploratory (red), Fact-Finding (blue) and borderline (green) search clusters. The upper plots illustrate the stationary distributions of the *Markov Models*, see Sec. 3.3.1, while the lower ones represent the web page dwell times.

7.8.5 Conclusion

The applicability of the *Information Search Behavior Profile Model* (ISBP Model) was shown in supervised and unsupervised approaches. Within the classification setting, an accuracy of 88.58% could be achieved, and this value is fully comparable to the previous user study with 87.7% in Sec. 7.2 and with 89.32% in Sec. 7.3. Further, the extension of the classifier to Finite Mixture Models was demonstrated, as well as its potential to extract knowledge from the cluster model using the exact same model assumptions as the classifier. It can be observed that cluster assignments abstractly correlated with search activity assignments of the lab experiment with a 79.1% agreement. This indicates that the ISBP Model derives information purely from the data that is otherwise constraint by experimental designs via the task assignment. Hence, the inference of search activities based on unlabeled data seems reasonable. During the analysis, model characteristics were derived for Exploratory and Fact-Finding search activities. These characteristics confirmed that a Fact-Finding activity is more oriented on using search engine result pages (SERPs) and an Exploratory search activity is more associated to web pages. The analysis of the *navigational model* in this study showed strong similarities with the one of the previous study in Sec. 7.2 (Fig. 7.5 & Fig. 7.25). Additionally, the tendency of different time durations could be confirmed as well. The analysis of the *interaction model* in this study showed strong similarities with the one in the previous study in Sec. 7.3 (Fig. 7.7 & Fig. 7.25). Labeled data are often absent, and the proposed clustering approach might be a reasonable solution to infer search activities on massive data sets that cannot be assigned manually. With the promising result of this clustering approach provided, a major limiting factor for *ISBP Ranker* could be resolved. Therefore, the practical implementation in a real-life setting seems plausible.

Chapter 8

Conclusion, Summary & Perspectives

8.1 Research Question 1

RQ1: ***Which cognitive models for users in online searches can be used, and do they provide useful interpretations for Information Retrieval? What actions can be implemented by a user to achieve their search goals, and what strategies of users can be observed? Which aspects can we derive from that to further improve the usability of Information Retrieval systems? How can we exploit the interdependency between actions, strategies and goals?***

8.1.1 Search Activities & Search Goals

All in all, *Information-Seeking Behavior* should be considered a highly relevant research scope for Information Retrieval (IR) systems. Previous research in the form of *Kuhlthau's Model* [72][73], *Ellis' Model & Wilson's Aggregation* [37][39][38][136] provide solid conceptual perspectives on the search process of users. Usually in IR, a user search is reduced to singular query responses and the search is not treated in a contextualized manner, that would be more suited for the user's *cognitive model* within substages of the entire search process. A good starting point for contextualized user support in IR can be realized via a better understanding of *Exploratory Search & Search Activities*. According to Marchionini [79], Exploratory searches decompose into three activities: *lookup*, *learn* & *investigate*. Within this thesis, *Fact-Finding* (*Fact*) search activities were considered to be a realization of *lookup*, while *Exploratory* (*Expl*) search activities realized the entirety. Both search activities were compared against to draw conclusions about their underlying nature. A user study was conducted to trigger these search activities. For that, *Fact-Finding* tasks (also referred with *Fact*) were design to trigger *Fact* search activities and *Exploratory* (also referred with *Expl*) tasks for *Expl* search activities. By using such a binary separation, it is straight-forward to draw conclusions in the form of a pairwise comparison or by ratios. Further on, both tasks can be associated with properties of the cognitive model of users that are confronted with such tasks. *Expl* tasks can be described as a rather *open* task, where an *Information Need* cannot be specified precisely. In contrast, *Fact* tasks can be described as being rather *closed* with a more or less clear Information Need. This results in a clear vs. a fuzzy expectation by the user that is aimed to be found within the search session. With such an expectation about the search outcome, a user can implement different search activities for a *Fact* task by a *Fact* search activity and for an *Expl* task by *Expl* search activity. Therefore, both concepts can be used for the interpretation of the user's search process, the user's cognitive model and to a certain degree the cognitive load of the user. It can be assumed that a *Fact*

task will trigger a *Fact* search activity and an *Expl* task will trigger an *Expl* search activity. In general, this must not hold true. A fact to be found in a search session by one user might trigger a whole exploration by another user because of different *world knowledge* in respect to the nature of this fact. Also, every exploration could be reduced to a singular answer if the amount of over-simplification within the answer seems plausible for that individual. Therefore, it is more than necessary to design experiments adequately and sanity-check the experimental outcome for plausibility, such as in *Search Sessions* Sec. 7.1. The data set analyzed in this work reasonably satisfies these plausibility aspects, and the majority of search activities should follow its task assignments. Nonetheless, some kind of hybrid search activities could be observed within the user study as well.

8.1.2 Search Activities & Navigational Strategies

User search sessions are sequences and should not be treated by Information Retrieval (IR) systems as singular query response tasks. Conceptually, such search sessions can be contextualized by *Information-Seeking Behavior* models. Within the scope of this thesis, such models were too abstractly defined to be realized in a Computer Science application. Nonetheless, the concept of *Exploratory Search & Search Activities* provided suitable instances to be further explored, e.g. *Fact-Finding* (*Fact*) and *Exploratory* (*Expl*) search activities. In respect to the navigational *strategy* within these activities, it can be observed that search activities follow a navigational context while traversing through the information space (of the Internet). For the analysis of such traversals, a *navigational model* has to consider such context and IR systems should make use of such models. Naturally, *Navigation & Probabilistic Regular Grammars* are closely connected with each other. The navigational *strategy* of *Expl* search activities are mainly oriented towards inspecting web pages and users can not really exploit much information on search engine result pages (SERP). Also, when executing queries, users seemingly have only limited options for reformulating queries towards more promising ones. In case of *Fact* search activities, users exploit knowledge extracted from SERPs more often and users reformulate queries in a more efficient way. This is a direct result of the *closed* nature of *Fact* search activities in comparison to the *open* ones in *Expl* search activities. This *closed* nature results in a clear expectation which is aimed to be satisfied during the search. Based on this expectation, fast inspections of SERPs are possible and more suitable query reformulations can be derived. In case of *Expl* search activities, such precise expectations can not be created because of their *open* nature. Search engines should provide more user support in such cases to guide users towards desired search goals, lower their cognitive load and ease the search experience. IR systems should dynamically diversify their search results in case of detecting *Expl* search activities. A diversified result list will not only provide the bigger picture in respect to the query, it supports the intrinsic characteristics of investigation and learning within *Expl* search activities. Further, it counters the effect of users struggling with reformulating queries because the diversification might intrinsically solve this problem. Additionally, automatic query suggestions could support the user during *Expl* search activities to further diversify the search result and therefore potentially reduce the cognitive load of the user.

8.1.3 Search Activities & User Interaction

During search sessions, a user interacts with an Information Retrieval (IR) system via a multi-modal set of *actions*. For sure, a user does more during a search than sending a query to receive a sorted list of documents suited for that input. Conceptually, the set of *actions* a user implements during the search can be described by *Information Search Behavior* models. Within the scope of this thesis, such *actions* were derived from two sources: log-files and eye-tracking data. The profile of such *actions* were compared against for *Exploratory* (*Expl*) and *Fact-Finding* (*Fact*) search activities. In respect to the user behavior in form of *actions* during individual page visits, it can be observed that such a multi-modal approach is needed to analyze search activities. Even though, information derived from log-files alone are indeed quite effective to make predictions about the user's search activity, they do not provide enough information to clearly understand what the user is actually doing. Only by combining complementary data sources, e.g. by eye-tracking, a deeper understanding of the user's search seems plausible. Therefore, the analysis of search activities should implement *interaction models* that consider multi-modal approaches. Models capable to work with different data sources are faced with the challenge to meaningfully combine the different modalities into one global model pipeline. Brute-force approaches that simply combine every source of data without any kind of fine-tuning, might even lower the predictive performance of the global model. A sophisticated approach for *Feature Selection* and *Model Selection* should be used to achieve the best performance instead of ad-hoc approaches. The profile of *actions* users execute on web pages are predictive, but not necessarily facilitates data understanding in respect to search activity recognition when working on features derived from log-files alone. Aspects of clicking and scrolling are of limited usefulness to correctly predict the user's search activity. In contrast, web page dwell-times are differently distributed in both search activities with statistical significance. Even though, dwell-times are useful predictors, they lack a clear description of what the user is doing while just being on the visited web page. In respect to low-level eye-tracking features, users fixate more and longer on the presented content in case of *Expl* search activities with statistical significance. This might be a direct result from the more *open* nature of the *Expl* search activity in comparison to the *Fact* one. This *open* nature results in a rather fuzzy expectation for the domain to be explored. With such fuzzy expectations given, a complete and thorough inspection of the web page content is necessary. The increased time duration in combination with the increased fixation activity implies a high cognitive load of the user during the exploration. In contrast, the *closed* nature of *Fact* search activities results in a clear expectation that enables the user a fast evaluation of presented information. In case of potential support of *Expl* search activities by IR systems, external data sources could be used to provide the user with a better understanding of the presented content. Databases and ontologies could high-light meaningful named entities or provide synonym collections for future query adjustments. In case of ontologies, hypernym and hyponyms could be provided as well. Unfortunately, low-level eye-tracking data suffices barely enough information to provide meaningful interpretation for search activities. High-level eye movement analysis is a necessity to gain insights about the search process. Higher-level eye movement patterns via *Reading and Information Processing* comprise a richer interpretation and are more closely connected with the *cognitive model* of the user. Therefore, higher-level eye movements are much better suited to approximate the user's search intent than plain information derived from *Fixations & Saccades*.

8.1.4 Search Activities & Eye Movement Strategies

During search sessions, a user processes information by reading the presented text provided by Information Retrieval (IR) systems. The eye movement of the user can be measured by *Eye-Tracking* via the analysis of the users *Fixations & Saccades*. Higher-level analysis modules for *Reading and Information Processing* are not meaningfully supported by the current state of technology. Within this thesis, a module specifically designed for *Automatic Reading Detection* was implemented to derive conclusions about changing *Reading Strategies in User's Search Activities*. In respect to behavior in the form of these higher-level eye movement *strategies*, it can be observed that reading and its variants provide appropriate descriptions of the user's search process and additionally provide suitable interpretation for the user's *cognitive model*. This statement is justified by the observation that reading and its variants realize *strategies* of intent and planning. Hence, these high-level eye movement pattern enable researchers to derive sophisticated conclusions about the search process of users. Eye movement *strategies* considered in this thesis are *Scanning*, *Skimming* and thorough *Reading*. Skimming is a fast reading *strategy* with the aim to identify main points or the essence of a text without thoroughly reading the text word-by-word. Scanning is a fast reading *strategy* to identify desired keywords or phrases. Thorough reading describes a full reading of the text, nearly word-by-word. Both fast reading *strategies* are abstractly correlated with *Fact-Finding* (*Fact*) search activities, and this connection can be justified by their respected interpretation. *Fact* search activities can be described as *closed* in their nature, with a clearly defined *Information Need*. This need is manifested in a clear expectation about what to find during the search. In respect to reading, this enables the individual the ability to adapt towards faster reading *strategies*. Such an expectation can be mapped to the text as decision criteria to either inspect the text in detail or by skipping it. In contrast, *Exploratory* (*Expl*) search activities can be described with an *open* nature and an Information Need that cannot be specified precisely. Therefore, this need cannot be manifested clearly in an individual's expectation and users cannot map this expectation on the text as decision criteria to skip text without the risk to miss the desired information. For such cases, authors of web pages might support users with structural information about the web page content or with graphical enrichment in the form of color and/or style high-lights. For example, structural information in the form of table-of-contents (TOC), abstracts, sections & navigational links to subsections might lower the cognitive load of the user and ease the search experience. Especially, the Skimming *strategy* aims for a general understanding of the presented text. A meaningful abstract of the web page's content and an additional TOC would support the user efficiently. Graphical elements, such as style fonts (*italic*, **bold**) and colors, could be used to high-light important keywords or phrases, e.g. for definitions, proofs, measurements, etc. Especially, the Scanning *strategy* would be guided by graphical high-lights to better capture relevant units. This thesis incorporated this approach. On one hand, to adapt to the findings of the thesis itself and on the other hand, this thesis will serve in a follow-up study in respect to fast text understanding & style fonts. Automatic summarization of the page content might be a suitable support by an IR system to reduce the cognitive load of users. Based on such summaries, users might have the chance to decide early on if a detailed inspection of the content is worth their time and effort.

8.1.5 Information Search Behavior Profile Model

The core of this thesis is the *Information Search Behavior Profile Model* (ISBP Model). The model is a mathematical & computational *(User) Behavior Model* to draw conclusions about search activities and to gain knowledge about potential user support for Information Retrieval systems. The ISBP Model is not only a mathematical & computational approach to model data, the model presents a unified formalism for a rather complex field of application. This field comprises aspects of information search, user modeling and behavior analysis. All in all, this results in an interdisciplinary field associated to Computer Science, Information Science and Psychology. Unfortunately, this results in a major problem. As a Bioinformatician myself, I studied courses in Biology, Mathematics and Computer Science. The most important lesson that I learned during that time was, that interdisciplinary problems can only be approached in an interdisciplinary way; in a combined effort, using combined knowledge, expertise & passion. Any of such exchange is usually inhibited by the different terminology used between these fields. Even worse, sometimes even hindered by the ego of elitists. *Research Question 0* in Sec. 5.1 provided a terminology to bridge the gap between these fields, aimed to design a shared vocabulary. I have no illusion about this approach being free of criticism. In the contrary, I expect and even wish for that. Criticism is healthy, vitalizing, a potential enrichment and aims for interaction. As I was pleasantly surprised by the rich treasure of knowledge from the field of Information Science & Psychology, I am convinced that the inverse will hold true for the rich treasure of knowledge that can be found in the field of Computer Science. Interdisciplinary work must always be founded in bi/multi-directional respect, and I hope that my work suffices my own claim. At least, I tried my best. Even if my unification approach does not turn out as I intend it to be, I hope at least one or two readers will broaden their perspective by looking into fields outside their expertise with enhanced interest, respect or even gratitude.

8.2 Research Question 2

RQ2: ***Which mathematical & computational (User) Behavior Model can be used to draw a conclusion about the information search behavior of users? What is their potential in respect to inference, and what are their limitations? How can they be applied to gain knowledge about the search process?***

8.2.1 Bayesian Networks

All in all, *Bayesian Networks* (BN) provide a flexible and efficient way to model abstract data. BNs can generally be applied in all Machine Learning (ML) settings, e.g. *classification*, *clustering* & *regression*. This model family is suitable to represent a plethora of data types via *Models for Unstructured Prediction* and *Models for Structured Prediction*. Further on, arbitrarily complex BNs can be created by combining them with sub-models, as long as the global network remain directed & acyclic. Therefore, specific sub-models, e.g. the *navigational*, *interaction* and *eye gaze model*, can be combined into a broader network, e.g. the *Information Search Behavior Profile Model*. A clear interpretation can be drawn from sub-models given the graph structure, which results in less black-box-like behavior as in comparison to other ML models. In general, BNs are able to contextualize any information flow via their factorization, and they can naturally fuse multi-modal data sources. Being probabilistic models by definition, *Feature Selection* and *Model Selection* can be done consistently. Feature selection can be done straight-forward by *Statistical Hypothesis Testing* as well as model selection by information criteria, e.g. the *Akaike Information Criterion* [4] & the *Bayesian Information Criterion* [112], or by statistical hypothesis testing. Results can easily be visualized in the BN graph structure. While vertices at the leaf level in the graph represent the selected feature, edges represent their interconnection. Vertices in inner nodes represents conclusions drawn by the network. Such inner nodes could resemble *Eye Movement in Reading* concluded by leaf level eye-tracking data in the form of measurements derived from plain *Fixations & Saccades* at the leaf level. Arbitrarily complex modules and submodules can be stacked towards more complex or more multi-modal analysis models. Learning & inference in BNs fully decompose into the *Max* & *Sum Product Algorithm* and remain statistical consistent within the model family. Additionally, BNs can be learned with different estimation techniques, e.g. standard point estimation techniques such as the *Maximum Likelihood Estimation*, *Maximum A Posterior Estimation* or *Posterior Mean Estimation*. Fully Bayesian estimation techniques could be used as well, but they were not the scope of this thesis.

8.2.2 Bayesian Networks as (User) Behavior Models

In respect to modeling search sessions, *Bayesian Networks* provide a rich framework to analyze the search process in its entirety. A broad search-session-network can naturally decompose into specific subnetworks. Search session navigation can easily be modeled by a specific *navigational model*. Search session interactions on web pages can easily be modeled by a specific *interaction model*. In general, a multitude of models can be used for specific aspects, ranging from simple (mouse) *click models* to highly complex *bio-signal models*, such as *Brain-Computer-Interfaces*. This thesis provided a framework for just two multi-modal sources: log-files and eye-tracker data. One *interaction model* centered on information derived from log-files while a second

model, the *eye gaze model*, centered on the analysis of higher-level eye movement pattern. The proposed framework can and should be built upon. For that goal, it is necessary to work within a unified formalism. The *Information Search Behavior Profile Model* refers to itself as being a *(User) Behavior Model*. Unfortunately, any model can claim these words for itself. Therefore, it is necessary to state what this term means. During this thesis a particular formalism & definitions were provided, that defines a model as being a *behavior model* if and only if it realizes predictions via a triple of $(actions, strategies, goals)$ just as follows:

- *Action*:
  Is a singular action, activity or interaction of an individual to recognize, interact with or manipulate its surrounding. In the context of information search, examples of such actions are described by *Information Search Behavior* models. Particular examples can be stated by: a web page visit & its dwell-time and clicking, scrolling, fixating activities on that page etc.

- *Goal*:
  Is an aim, outcome, goal or achievement of an individual that is wished to be realized. In the context of information search, such goals are referred to as the *Information Need*. Such needs can be categorized by their underlying *Search Tasks* as being clear & precise, e.g. *closed* as in the case of *Fact-Finding* search activities, or being vague & hard to define, e.g. *open* as in the case of *Exploratory* search activities. In general, other *goals* can be considered as well, but they were not the scope of this thesis.

- *Strategy*:
  Is a plan, algorithm or strategy of an individual, where the individual has a composition or sequence of *actions* that is expected to achieve a *goal*. The lowest level of decomposition will result in a singular *action*. In the context of information search, *Exploratory Search & Search Activities*, e.g. *Fact-Finding* and *Exploratory* search activities, realize the highest level in the strategy hierarchy of searches that is considered in this thesis. Finer grained strategies comprise the navigational strategy during the search or the eye movement strategy to extract information from web pages. In general, other *strategy* layers can be considered as well, but they were not the scope of this thesis.

- *Behavior*:
  Is a particular expression, combination or triple of $(goal, actions, strategies)$. The *behavior* of an individual is defined as a particular expression of *actions* that are combined into a *strategy* to fulfill a *goal*. Any model that makes inference within all three concepts simultaneously is considered to be a *(User) Behavior Model*.

With that definition provided, it is clear that vertices at the leaf level of Bayesian Networks realize *actions*. Vertices as inner nodes represent *strategies* over *actions* via their interconnected edges. If one dedicated 'root' node within these Bayesian Networks realize the *goal*, these networks suffice the definition of *(User) Behavior Models*. The *Information Search Behavior Profile Model* comprises a *navigational model* for the navigational *strategy*, an *eye gaze model* for the eye movement *strategy* and an *interaction model* for the *actions* a user executes. The proposed model was always learned given the *goal* specification. Therefore, the model follows the proposed definition of (User) Behavior Models. Any future development, should consider if the models suffices for all mentioned aspects.

8.3 Research Question 3

RQ3: ***In case that mathematical & computational (User) Behavior Models provide reliable information to draw conclusions about the Information Behavior: How can this information be exploited in the setting of Information Retrieval (IR)? Is it possible to use such (User) Behavior Models in rankings by an IR system? Can a static machine adapt to changed behavior in users? What does this mean in the context of the Human-Machine-Interaction?***

8.3.1 Behavior-Driven Ranking

Information Retrieval systems provide query responses to users implementing online search sessions. Classical but static ranking approaches realized by the *Boolean Model*, the *Vector Space Model* [110] & the *Probabilistic Model* [108] can be generally expressed via the *Bayesian Network Model* [106]. Any of such user searches is guided by an underlying *Information Need* that aims to be satisfied during the search. Such needs can be categorized by their underlying *Search Tasks* as being clear & precise, e.g. *closed* as in the case of *Fact-Finding* search activities, or being vague & hard to define, e.g. *open* as in the case of *Exploratory* search activities. Therefore, a suitable ranking should consider the underlying *goal* of the search. The *Information Search Behavior Profile Model* (ISBP Model) provides a suitable recognition by the analysis of the user search activity with an *Accuracy* of up to 89.32%. Classical ranking approaches can be guided by the ISBP Model to realize suitable rankings in respect to the current search activity of the user. Fortunately, the Bayesian Network Model is an instance of *Bayesian Networks* and the ISBP Model is an instance of Bayesian Networks as well. Therefore, both models can easily be combined into a broader network, e.g. the *ISBP Ranker*. The unified nature of both models and their combination result in a consistent information flow in the ranking itself. The in-depth analysis of *Ranking with Information Search Behavior Profiles* provided nice mathematical properties of the proposed approach. The *ISBP Ranker* naturally re-weights the ranking with measured behavior aspects of the user during the search. This re-weighting follows the direction of the current search activity of the user. Abstractly, the ISBP model extends the classical ranking with *behavior keywords* described in *Search Activities & Navigational Strategies*, *Search Activities & User Interaction* and *Search Activities & Eye Movement Strategies*. Because the ISBP Model provides an instance of *(User) Behavior Models* and the *ISBP Ranker* provides a ranking guided by it, this ranking approach can be considered as a *behavior-driven ranking*.

8.3.2 Implications for Human-Machine-Interaction

The implication of *Behavior-Driven Ranking* for Human-Machine-Interaction should not be underestimated because of the task's restriction to the field of Information Retrieval. Conceptually, ranking can be considered as an ordered list of responses suitable for a provided input by a user. In general, the ranking can be described as a *reaction model*. Currently, such *reaction models* comprise a rather static nature. In this thesis, a *behavior recognition model* was implemented and combined with a *reaction model*. This resulted in a contextualized combination of behavior-driven responses for an estimated *goal* of the user. Conceptually, this approach realizes a human-centered or user-adaptive model. All in all, three aspects needed to be considered by such models: *recognition*, *reaction* & *goal*. The proposed model extends previous approaches working on *explicit* user input in textual form (queries) with *implicit* user input via *behavior keywords* (according to the *(User) Behavior Models*). Based on the provided definition of *behavior* in *Research Question 0* in Sec. 5.1, any *behavior* is always driven by an underlying *goal*. Therefore, the analysis and exploitation of information derived from *behavior* will be suitable for a machine to support the human's *goal*. I believe that the provided framework & definitions in Sec. 5.1 comprise suitable ideas that should be further exploited by the community of Human-Machine-Interaction, Robotics and others.

Unfortunately, this results in a broader implication. Such a proposed model framework actually needs additional evaluation by experts from Philosophy, Ethics, Sociology & Psychology. For as long as this evaluation is missing, unfortunately, the Computer Science community needs to consider these aspects. As a computer scientist myself, I dare to express that computer people are not necessary good human people. The structural and functional thinking in Computer Science does not mix well with the soft and latent constraints characteristic for human thinking. In respect to certain implication of this thesis, I recognize a potential for something useful. In others, I recognize a potential that I do not want to describe here in detail. Whoever reads this thesis and aims to adapt or apply some thoughts of it, think about the implication. As a Bioinformatician myself, I appreciate ethics committees. I have by far no idea about ethics, but I see the responsibility here to express a small rule of thumb: **especially in the pursuit towards something good, one should always consider *the right of anonymity*, *the right of individuality*, *the right of imperfection*, *the right of not-knowing-everything* and *the right to not-want-to-know-everything* otherwise this pursuit will lead in the opposite direction.**

Part IV

Appendix

Appendix A

Notation

A.1 Variables, Symbols, and Operations

Glyphs & Symbols	
$0, 1, \infty$	*zero*, *one* and *infinity*.
$\forall, \exists, \nexists$	*for all*, *exists* and *not exists*
$:, \mid$	condition on
$\wedge$	logical *and*
$\in$	*is element* in
...	contextual *indicator for a sequence*
$\times$	*cross product*
$\mathbb{N}$	*Natural Numbers*
$\mathbb{R}, \mathbb{R}^+$	*Real Numbers* and *positive Real Numbers*
$=, \neq >, <, \geq, \leq$	*equal to*, *not equal to*, *greater*, *lesser* *greater or equal* and *lesser or equal*

Variables & Spaces	
$x \in \mathbb{N}$	a variable x representing a scalar value in $\mathbb{N}$
$x \in \mathbb{R}$	a variable x representing a scalar value in $\mathbb{R}$
$x \in [a, b]$, $x \in (a, b)$	a variable x falls in the interval between a and b (two sided exclusive
$x \in (a, b]$, $x \in [a, b)$	& inclusive both boundaries, and one sided combinations).
$\boldsymbol{x} \in \mathbb{R}^m$	a variable $\boldsymbol{x}$ representing vector $\boldsymbol{x} = (x_1, ...x_m)$ with $\forall_{i \in \{1,...m\}} x_i \in \mathbb{R}$
$\mathbf{0}, \mathbf{1}$	a vector comprising only of zeros, or ones
$\boldsymbol{X} \in \mathbb{R}^{n \times m}$	a variable $\boldsymbol{X}$ representing a Matrix with $\forall_{i \in \{1,...n\}} \forall_{j \in \{1,...m\}} x_{ij} \in \mathbb{R}$, see *Vector Space & Matrix Algebra*

Variabels & Sets	
$\mathcal{X} = \{x_1, ...x_N\}$	a set $\mathcal{X}$ comprising elements x_1, ... x_N
$x \in \mathcal{X}$	x is an element in $\mathcal{X}$
$x \notin \mathcal{X}$	x is not an element in $\mathcal{X}$
$\mathcal{X} \setminus x$	a set $\mathcal{X}$ without the element x
$\mathcal{X} \cup \mathcal{Y}$	union of two sets, that is, the set containing all elements in either $\mathcal{X}$ or $\mathcal{Y}$
$\mathcal{X} \cap \mathcal{Y}$	intersection of two sets, that is, the set containing all elements that are in both $\mathcal{X}$ and $\mathcal{Y}$
$\mathcal{X} \subseteq \mathcal{Y}$	$\mathcal{X}$ is a subset of $\mathcal{Y}$, that is, all elements in $\mathcal{X}$ are also elements in $\mathcal{Y}$

A.2 Functions

This section provides a small summary of selected functions and their properties. Some properties imply prelimitary constraints on the function itself, and they will be described in more detail in *Probability Theory*, *Information Theory* and *Vector Space & Matrix Algebra*.

Functions and their Properties/Characteristics	
$f(x) = y$	*function* f on x equals y
$f : \mathcal{X} \to \mathcal{Y}$	*signature* of a function f
$\max_{x \in \mathcal{X}} f(x)$, $\min_{x \in \mathcal{X}} f(x)$	*maximum, minimum value* of f
$\operatorname{argmax}_{x \in \mathcal{X}} f(x)$	*value leading to the maximum* of f
$\lim_{x \to \infty} f(x)$	*value of* f *in the limit* as x approaches ∞
$\sup_x f(x)$	*smallest upper bound* of f
$\int_a^b f(x)\mathrm{d}x$	*definite integral* of f
$\int f(x)\mathrm{d}x$	*indefinite integral* of f
$\partial f(x)/\partial x$	*partial derivative* of f in respect to x
$\nabla f(x)$	*gradient operator* as the partial derivative
$E_{\mathsf{X}}[f(\mathsf{X})]$	*Expectation* of f_{X}
$Var_{\mathsf{X}}[f(\mathsf{X})]$	*Variance* of f_{X}

Functions on Functions	
$f \sim g$	f is *distributed* as g
$f \simeq g$	f is *approximately equal* to g
$f \propto g$	f is *proportional* to g
$KL(f_{\mathsf{X}} \| g_{\mathsf{X}})$	*Kullback-Leibler Divergence* between f_{X} and g_{X}
$H[f_{\mathsf{X}} \| g_{\mathsf{X}}]$	*Conditional Entropy* of f_{X} given g_{X}

Specific Functions	
$\ln(x)$	*natural logarithm* of x
$\exp(x)$	*exponential* of x
$\sqrt{x}$	*square root* of x
$\sum_{i=1}^N x_i$	*sum* $x_1 + ...x_N$
$\prod_{i=1}^N x_i$	*product* $x_1 \cdot ...x_N$
$\hat{\theta}$	*estimate* for θ
$\mathcal{I}(x_1 = x_2)$	*identity function* equals 1 if $x_1 = x_2$ and 0 otherwise

Functions on Vectors & Matrices	
$\boldsymbol{X}^T$	*Transpose of a Matrix*
$\lvert\boldsymbol{X}\rvert$	*Determinant of a Matrix*
$\boldsymbol{X}^{-1}$	*Inverse of a Matrix*
$\boldsymbol{x}^T$	*transposed vector*
$\boldsymbol{x}^T\boldsymbol{y}$	*(inner) vector product* $\sum_{i=1}^N x_i \cdot y_i$
$\boldsymbol{x}\boldsymbol{y}^T$	*(outer) vector product*

$$\boldsymbol{x}\boldsymbol{y}^T = \begin{pmatrix} x_1 \cdot y_1 & \cdots & x_1 \cdot y_{A'} \\ \vdots & \ddots & \vdots \\ x_A \cdot y_1 & \cdots & x_A \cdot y_{A'} \end{pmatrix}$$

Appendix B

Probability Theory

All definitions provided in this section are extracted with minor adaptions from ***Learning Kernel Classifiers - Theory and Algorithms*** by Ralf Herbrich [57].

B.1 σ-algebra

Given a set $\mathcal{X}$, a collection $\mathfrak{X}$ of sets $X \subseteq \mathcal{X}$ is called a *σ-algebra* over $\mathcal{X}$ if and only if:

1. If a set $X \in \mathfrak{X}$ so is its complement $X^c = \mathcal{X} \setminus X$.
2. If $X_i \in \mathfrak{X}, i = 1, ...\infty$ is any countable collection of sets in $\mathfrak{X}$, then also their union $\cup_{i=1}^{\infty} X_i \in \mathfrak{X}$ and intersection $\cap_{i=1}^{\infty} X_i \in \mathfrak{X}$ belong to $\mathfrak{X}$.

B.2 Borel Sets

Given $\mathcal{X} = \mathbb{R}^n$, the Borel sets $\mathcal{B}_n$ are the smallest *σ-algebra* that contains all open intervals for all $a_i, b_i \in \mathbb{R}$:

$$\{(x_1, ...x_n) \in \mathbb{R}^n | \forall i \in \{1, ...n\} : x_i \in (a_i, b_i)\}$$

B.3 Measurable Space

A *Measurable Space* is defined by the tuple $(\mathcal{X}, \mathfrak{X})$ and a real-valued function $g : \mathcal{X} \to \mathbb{R}$ if and only if:

$$\forall z \in \mathbb{R} : \{x \in \mathcal{X} | g(x) \leq z\} \in \mathfrak{X}$$

B.4 Probability Space

Given a *Measurable Space* $(\mathcal{X}, \mathfrak{X})$, the sample space $\mathcal{X}$ and a *σ-algebra* $\mathfrak{X}$ over $\mathcal{X}$, then a *Probability Space* is defined by the triple $(\mathcal{X}, \mathfrak{X}, P)$, where P is a probability measure on $\mathcal{X}$, i.e. $P : \mathfrak{X} \to [0, 1]$ such that $P(\mathcal{X}) = 1$ and for all countable collections of non-overlapping sets $X_i \in \mathfrak{X}, i = 1, ...\infty$:

$$P(\cup_{i=1}^{\infty} X_i) = \sum_{i=1}^{\infty} P(X_i)$$

B.5 Random Variable

Given a *Measurable Space* $(\mathcal{X}, \mathcal{X})$, then a *Random Variable* is a measurable real-valued function $f : \mathcal{X} \to \mathbb{R}$. Thus a random variable $\mathsf{Y} = f(\mathsf{X})$ induces a measure P_{Y} which acts on the real line, i.e. $\mathcal{Y} = \mathbb{R}$ and for which the *σ-algebra* $\mathcal{Y}$ contains at least the intervals $\{(-\infty, z] | z \in \mathbb{R}\}$. The measure P_{Y} is induced by the measure P_{X} and f by:

$$\forall Y \in \mathcal{B}_1 : P_{\mathsf{Y}}(Y) = P_{\mathsf{X}}(\{x \in \mathcal{X} | f(x) \in Y\})$$

B.6 Probability Density Function

Given a *Random Variable* X and the distribution function $F_{\mathsf{X}} : \mathbb{R} \to [0, 1]$ defined by $F_{\mathsf{X}}(x) = P_{\mathsf{X}}(\mathsf{X} \leq x)$, then the function $f_{\mathsf{X}} : \mathbb{R} \to \mathbb{R}$ is called the *Probability Density Function* if:

$$\forall z \in \mathbb{R} : F_{\mathsf{X}}(z) = \int_{x \leq z} f_{\mathsf{X}}(x) \mathrm{d}x$$

B.7 Expectation

Given a measurable function $f : \mathcal{X} \to \mathbb{R}$, then the *Expectation* $E_{\mathsf{X}}[f(\mathsf{X})]$ (written in short as $E_{\mathsf{X}}[\mathsf{X}]$) of f over the random draw of x is defined by:

$$E_{\mathsf{X}}[f(\mathsf{X})] = \int_{\mathbb{R}} f(x) \mathrm{d}F_{\mathsf{X}}(x)$$

B.8 Variance

Given a *Random Variable* X, then the *Variance* $Var_{\mathsf{X}}[\mathsf{X}]$ is defined by:

$$Var_{\mathsf{X}}[\mathsf{X}] = E_{\mathsf{X}}[(\mathsf{X} - E_{\mathsf{X}}[\mathsf{X}])^2] = E_{\mathsf{X}}[\mathsf{X}^2] - E_{\mathsf{X}}[\mathsf{X}]^2$$

B.9 Product Space

Given two *Measurable Space*s $(\mathcal{X}, \mathcal{X})$ and $(\mathcal{Y}, \mathcal{Y})$, then the *Product Space* is defined by $(\mathcal{X} \times \mathcal{Y}, \mathcal{X} \times \mathcal{Y})$ with $\mathcal{X} \times \mathcal{Y}$ being the smallest *σ-algebra* which contains the sets $\{X \times Y | X \in \mathcal{X}, Y \in \mathcal{Y}\}$.

B.10 Joint, Conditional and Marginal Measure

Given a *Joint Probability* space $(\mathcal{X} \times \mathcal{Y}, \mathcal{X} \times \mathcal{Y}, P_{\mathsf{XY}})$, then the *Marginal Probability* measure P_{X} is defined by:

$$\forall X \in \mathcal{X} : P_{\mathsf{X}}(X) = P_{\mathsf{XY}}(X \times \mathcal{Y})$$

Given $Y \in \mathcal{Y}$ and $P_{\mathsf{Y}}(Y) > 0$, then the *Conditional Probability* measure $P_{\mathsf{X}|\mathsf{Y} \in Y}$ is defined by:

$$\forall X \in \mathcal{X} : P_{\mathsf{X}|\mathsf{Y} \in Y}(X) = \frac{P_{\mathsf{XY}}(X \times Y)}{P_{\mathsf{Y}}(Y)}$$

B.11 Independence

Given two *Random Variable*s X, Y and a joint probability measure P_{XY}, then the *Random Variable*s are called *Independent* if and only if:

$$\forall X \in \mathcal{X}, Y \in \mathcal{Y} : P_{\mathsf{XY}}(X \times Y) = P_{\mathsf{X}}(X) \cdot P_{\mathsf{Y}}(Y)$$

B.12 Expectation (n-dimensional)

Given n *Random Variable*s $\mathbf{X} = (\mathsf{X}_1, ...\mathsf{X}_n)$ with a joint measure $P_{\mathbf{X}}$, then the *Expectation* $E_{\mathbf{X}}[\mathbf{X}]$ is defined by:

$$E_{\mathbf{X}}[\mathbf{X}] = (E_{\mathsf{X}_1}[\mathsf{X}_1], ...E_{\mathsf{X}_n}[\mathsf{X}_n])$$

B.13 Covariance and Covariance Matrix

Given two *Random Variable*s X, Y and a joint probability measure P_{XY}, then the *Covariance* $Cov[\mathsf{X}, \mathsf{Y}]$ is defined by:

$$Cov[\mathsf{X}, \mathsf{Y}] = E_{\mathsf{XY}}[(\mathsf{X} - E_{\mathsf{X}}[\mathsf{X}]) \cdot (\mathsf{Y} - E_{\mathsf{Y}}[\mathsf{Y}])]$$

Given n *Random Variable*s $\mathbf{X} = (\mathsf{X}_1, ...\mathsf{X}_n)$, m *Random Variable*s $\mathbf{Y} = (\mathsf{Y}_1, ...\mathsf{Y}_m)$ and a joint measure $P_{\mathbf{XY}}$, then the $n \times m$ *Covariance Matrix* $\boldsymbol{Cov}[\mathbf{X}, \mathbf{Y}]$ is defined by:

$$\boldsymbol{Cov}[\mathbf{X}, \mathbf{Y}] = \begin{pmatrix} Cov[\mathsf{X}_1, \mathsf{Y}_1] & \cdots & Cov[\mathsf{X}_1, \mathsf{Y}_m] \\ \vdots & \ddots & \vdots \\ Cov[\mathsf{X}_n, \mathsf{Y}_1] & \cdots & Cov[\mathsf{X}_n, \mathsf{Y}_m] \end{pmatrix}$$

B.14 Bayes Theorem

Given a joint probability space $(\mathcal{X} \times \mathcal{Y}, \mathcal{X} \times \mathcal{Y}, P_{\mathsf{XY}})$, then for all $X \in \mathcal{X}$, $P_{\mathsf{X}}(X) > 0$ and $Y \in \mathcal{Y}$, $P_{\mathsf{Y}}(Y) > 0$:

$$P_{\mathsf{X}|\mathsf{Y} \in Y}(X) = \frac{P_{\mathsf{Y}|\mathsf{X} \in X}(Y) \cdot P_{\mathsf{X}}(X)}{P_{\mathsf{Y}}(Y)}$$

Appendix C

Information Theory

C.1 Entropy

Within the framework of *Information Theory*, information is motivated as the process of transmitting the value of a *Random Variable* by a sender towards a receiver. The average amount of transmitted information can be formulated as the *Expectation* of that information in respect to its probability distribution. This quantity is called the *Entropy* [122] and defined by:

$$H[\mathsf{X}] = \begin{cases} -\sum_{x \in \mathcal{X}} P_{\mathsf{X}}(x) \cdot \ln P_{\mathsf{X}}(x) & \mathcal{X} \text{ - discrete} \\ -\int_{x \in \mathcal{X}} P_{\mathsf{X}}(x) \cdot \ln P_{\mathsf{X}}(x) \, \mathrm{d}x & \mathcal{X} \text{ - continuous} \end{cases}$$

C.2 Conditional Entropy

Given two *Random Variable*s X, Y and a joint probability measure P_{XY}, then the *Conditional Entropy* is motivated to be the information needed to describe one by knowing the other and defined by:

$$H[\mathsf{Y}|\mathsf{X}] = \begin{cases} -\sum_{x \in \mathcal{X}, y \in \mathcal{Y}} P_{\mathsf{XY}}(x,y) \cdot \ln \frac{P_{\mathsf{XY}}(x,y)}{P_{\mathsf{X}}(x)} & \mathcal{X}, \mathcal{Y} \text{ - discrete} \\ -\int_{x \in \mathcal{X}, y \in \mathcal{Y}} P_{\mathsf{XY}}(x,y) \cdot \ln \frac{P_{\mathsf{XY}}(x,y)}{P_{\mathsf{X}}(x)} \, \mathrm{d}x\mathrm{d}y & \mathcal{X}, \mathcal{Y} \text{ - continuous} \end{cases}$$

C.3 Joint Entropy

Given two *Random Variable*s X, Y and a joint probability measure P_{XY}, then the *Joint Entropy* is motivated to be the information needed to describe both and defined by:

$$H[\mathsf{X},\mathsf{Y}] = \begin{cases} -\sum_{x \in \mathcal{X}, y \in \mathcal{Y}} P_{\mathsf{XY}}(x,y) \cdot \ln P_{\mathsf{XY}}(x,y) & \mathcal{X}, \mathcal{Y} \text{ - discrete} \\ -\int_{x \in \mathcal{X}, y \in \mathcal{Y}} P_{\mathsf{XY}}(x,y) \cdot \ln P_{\mathsf{XY}}(x,y) \, \mathrm{d}x\mathrm{d}y & \mathcal{X}, \mathcal{Y} \text{ - continuous} \end{cases}$$

C.4 Kullback-Leibler Divergence

Let P_X and Q_X be probability measures, then the *Kullback-Leibler Divergence* [74], also known as the *Relative Entropy*, is defined by:

$$KL(P_X||Q_X) = \begin{cases} -\sum_{x\in\mathcal{X}} P_X(x)\cdot \ln Q_X(x) - \\ \quad \big(-\sum_{x\in\mathcal{X}} P_X(x)\cdot \ln P_X(x)\big) & \mathcal{X} \text{ - discrete} \\ -\int_{x\in\mathcal{X}} P_X(x)\cdot \ln Q_X(x)\mathrm{d}x - \\ \quad \big(-\int_{x\in\mathcal{X}} P_X(x)\cdot \ln P_X(x)\mathrm{d}x\big) & \mathcal{X} \text{ - continuous} \end{cases}$$

C.5 Cross-Entropy

Let P_X and Q_X be probability measures, then the *Cross-Entropy* is defined as the *Entropy* ($H_P[X]$) plus the *Kullback-Leibler Divergence* ($KL(P_X||Q_X)$). With basic algebra, one can easily see the name giving property being Cross-*Entropy* and its interpretation as an *Expectation* (for the continuous case the sum needs to be exchanged with the integral):

$$\begin{aligned} H[P_X, Q_X] &= H_P[X] + KL(P_X||Q_X) \\ &= H_P[X] - \sum_{x\in\mathcal{X}} P_X(x)\cdot \ln Q_X(x) - \big(-\sum_{x\in\mathcal{X}} P_X(x)\cdot \ln P_X(x)\big) \\ &= H_P[X] - \sum_{x\in\mathcal{X}} P_X(x)\cdot \ln Q_X(x) - H_P[X] \\ &= -\sum_{x\in\mathcal{X}} P_X(x)\cdot \ln Q_X(x) \\ &= -E_{P_X}[\ln Q_X] \end{aligned}$$

Appendix D

Vector Space & Matrix Algebra

All definitions provided in this section are extracted with minor adaptions from ***Learning Kernel Classifiers - Theory and Algorithms*** by Ralf Herbrich [57].

D.1 Vector Space

A set $\mathcal{X}$ is a *Vector Space* if addition and multiplication by scalar are defined such that $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$ and $c \in \mathbb{R}$:

$$\begin{aligned} \boldsymbol{x} + \boldsymbol{y} &\in \mathcal{X} \\ c \cdot \boldsymbol{x} &\in \mathcal{X} \\ 1 \cdot \boldsymbol{x} &= \boldsymbol{x} \\ 0 \cdot \boldsymbol{x} &= \mathbf{0} \end{aligned}$$

and the operator $\boldsymbol{x} + \boldsymbol{y}$ satisfies properties such as *commutativity*, *associativity*, existence of a *null & one element* and an *inverse element* as well as the *distributivity*, for all $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathcal{X}$:

$$\begin{aligned} \boldsymbol{x} + \boldsymbol{y} &= \boldsymbol{y} + \boldsymbol{x} \\ (\boldsymbol{x} + \boldsymbol{y}) + \boldsymbol{z} &= \boldsymbol{x} + (\boldsymbol{y} + \boldsymbol{z}) \\ \exists \mathbf{0} \in \mathcal{X} : \boldsymbol{x} + \mathbf{0} &= \mathbf{0} \\ \exists - \boldsymbol{x} \in \mathcal{X} : \boldsymbol{x} + (-\boldsymbol{x}) &= \mathbf{0} \\ c \cdot (\boldsymbol{x} + \boldsymbol{y}) &= c \cdot \boldsymbol{x} + c \cdot \boldsymbol{y} \\ (c + d) \cdot \boldsymbol{x} &= c \cdot \boldsymbol{x} + d \cdot \boldsymbol{x} \end{aligned}$$

D.1.1 Normed Space

A *Normed Space* of the *Vector Space* $\mathcal{X}$ is defined by the tuple $(\mathcal{X}, ||\cdot||)$, where $||\cdot|| : \mathcal{X} \to \mathbb{R}^+$ is called a norm for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$ and $c \in \mathbb{R}$ when:

$$\begin{aligned} ||\boldsymbol{x}|| \geq 0 \text{ and } ||\boldsymbol{x}|| = 0 &\leftrightarrow \boldsymbol{x} = \mathbf{0} \\ ||c\boldsymbol{x}|| &= |c| \cdot ||\boldsymbol{x}|| \\ ||\boldsymbol{x} + \boldsymbol{y}|| &\leq ||\boldsymbol{x}|| + ||\boldsymbol{y}|| \end{aligned}$$

D.1.2 Cauchy Sequence

A sequence $(\boldsymbol{x}_i)_{i\in\mathbb{N}}$ in a *Normed Space* is a *Cauchy Sequence* if:

$$\lim_{n\to\infty}\sup_{m\geq n}||\boldsymbol{x}_n-\boldsymbol{x}_m||=0$$

D.1.3 Inner Product Space

An *Inner Product Space* of a *Vector Space* $\mathcal{X}$ is defined by the tuple $(\mathcal{X},\langle\cdot,\cdot\rangle)$, where $\langle\cdot,\cdot\rangle:\mathcal{X}\times\mathcal{X}\to\mathbb{R}$ is called an inner product when it satisfies for all $\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}\in\mathcal{X}$ and $c,d\in\mathbb{R}$ the following:

$$\begin{aligned}\langle\boldsymbol{x},\boldsymbol{x}\rangle&\geq 0\\ \langle\boldsymbol{x},\boldsymbol{x}\rangle&=0\leftrightarrow\boldsymbol{x}=\boldsymbol{0}\\ \langle c\boldsymbol{x}+d\boldsymbol{y},\boldsymbol{z}\rangle&=c\langle\boldsymbol{x},\boldsymbol{z}\rangle+d\langle\boldsymbol{y},\boldsymbol{z}\rangle\\ \langle\boldsymbol{x},\boldsymbol{y}\rangle&=\langle\boldsymbol{y},\boldsymbol{x}\rangle\end{aligned}$$

D.1.4 Hilbert Space

A space is called *complete* if every *Cauchy Sequence* converges. A *complete Inner Product Space* is called a *Hilbert Space* $\mathcal{H}$.

D.1.5 Linear Operator

Given two *Hilbert Space*s $\mathcal{H}$ and $\mathcal{F}$, a mapping $T:\mathcal{H}\to\mathcal{F}$ is called a *Linear Operator* if and only if:

1. For all $\boldsymbol{x},\boldsymbol{y}\in\mathcal{H}$: $T(\boldsymbol{x}+\boldsymbol{y})=T\boldsymbol{x}+T\boldsymbol{y}$
2. For all $\boldsymbol{x}\in\mathcal{H}$ and $c\in\mathbb{R}$: $T(c\boldsymbol{x})=c\cdot T\boldsymbol{x}$

D.1.6 Eigenvector Equation

Let $T:\mathcal{H}\to\mathcal{H}$ be a *Linear Operator* on a *Hilbert Space* $\mathcal{H}$. If there is a vector $\boldsymbol{x}\in\mathcal{H}$, $\boldsymbol{x}\neq\boldsymbol{0}$, such that:

$$T\boldsymbol{x}=\lambda\boldsymbol{x}$$

for some scalar λ, then λ is an *eigenvalue* of T with corresponding *eigenvector* $\boldsymbol{x}$. The equation will be referred to as the *Eigenvector Equation*.

D.2 Matrix Algebra

A *Matrix* $\boldsymbol{A}\in\mathbb{R}^{n\times m}$ holds entries $\boldsymbol{A}_{ij}\in\mathbb{R}$ for $i=1,...n$ and $j=1,...m$ in the following form:

$$\boldsymbol{A}=\begin{pmatrix}\boldsymbol{A}_{11}&\cdots&\boldsymbol{A}_{n1}\\ \vdots&\ddots&\vdots\\ \boldsymbol{A}_{1m}&\cdots&\boldsymbol{A}_{nm}\end{pmatrix}$$

D.2.1 Transpose of a Matrix

For any matrix $\boldsymbol{A}$, its *Transpose* $\boldsymbol{A}^T$ is defined by:

$$\boldsymbol{A}_{ij} = \boldsymbol{A}_{ji}^T$$

D.2.2 Square Matrix

A matrix $\boldsymbol{A}$ is a *Square Matrix* if and only if $m = n$.

D.2.3 Trace of a Matrix

The *Trace* of a *Square Matrix* $tr(\boldsymbol{A})$ is defined as:

$$tr(\boldsymbol{A}) = \sum_{i=1}^{N} A_{ii}$$

D.2.4 Diagonal Matrix

A *Square Matrix* $\boldsymbol{A}$ is a *Diagonal Matrix* if and only if $\boldsymbol{A}_{ij} = 0$ for all $i \neq j$.

D.2.5 Determinant of a Matrix

The *Determinant* $|\boldsymbol{A}|$ of a *Square Matrix* $\boldsymbol{A}$ is defined by:

$$|\boldsymbol{A}| = \boldsymbol{A}, \qquad \text{if } n = 1$$
$$|\boldsymbol{A}| = \begin{cases} \sum_{i=1}^{n} \boldsymbol{A}_{ij} \cdot |\boldsymbol{A}_{[ij]}| \cdot (-1)^{i+j} & \text{for any } j \in \{1, ...n\} \\ \sum_{j=1}^{n} \boldsymbol{A}_{ij} \cdot |\boldsymbol{A}_{[ij]}| \cdot (-1)^{i+j} & \text{for any } i \in \{1, ...n\} \end{cases}$$

The matrix $\boldsymbol{A}_{[ij]}$ is a $(n-1) \times (n-1)$ matrix obtained by deleting the i-th row and j-th column from $\boldsymbol{A}$.

D.2.6 Symmetric Matrix

A *Square Matrix* $\boldsymbol{A}$ is a *Symmetric Matrix* if and only if $\boldsymbol{A}^T = \boldsymbol{A}$.

D.2.7 Inverse of a Matrix

The *Inverse* $\boldsymbol{A}^{-1}$ of a *Square Matrix* $\boldsymbol{A}$ is defined by:

$$\boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{A}\boldsymbol{A}^{-1} = \boldsymbol{I}$$

The inverse exists if and only if $|\boldsymbol{A}| \neq 0$

D.2.8 Positive Semi-Definite Matrix

A *Symmetric Matrix* $\boldsymbol{A}$ is *Positive Semi-Definite* if and only if:

$$\forall \boldsymbol{c} \in \mathbb{R}^n, \boldsymbol{c} \neq \boldsymbol{0} : \boldsymbol{c}^T \boldsymbol{A} \boldsymbol{c} \geq 0$$

Appendix E

Probability Distributions

E.1 Univariate Probability Distributions

E.1.1 Bernoulli Distribution

For the *support* $x \in \{0,1\}$ and $p \in [0,1]$, the *Bernoulli Distribution* will be described as a short summary of its characteristics, e.g. *Probability Mass Function*, *Cumulative Distribution Function*, *Expectation*, *Variance*, *Mode* and *Entropy*:

$$f(x;p) = p^x \cdot (1-p)^{1-x}$$

$$F(x;p) = \begin{cases} 1-p & x = 0 \\ 1 & x = 1 \end{cases}$$

$$E_{\mathsf{X}}[\mathsf{X}] = p$$

$$Var_{\mathsf{X}}[\mathsf{X}] = p \cdot (1-p)$$

$$\max f(x;p) = \begin{cases} 0 & \text{if } p < 0.5 \\ 1 & \text{otherwise} \end{cases}$$

$$H[\mathsf{X}] = -p \cdot \ln p - (1-p) \cdot \ln(1-p)$$

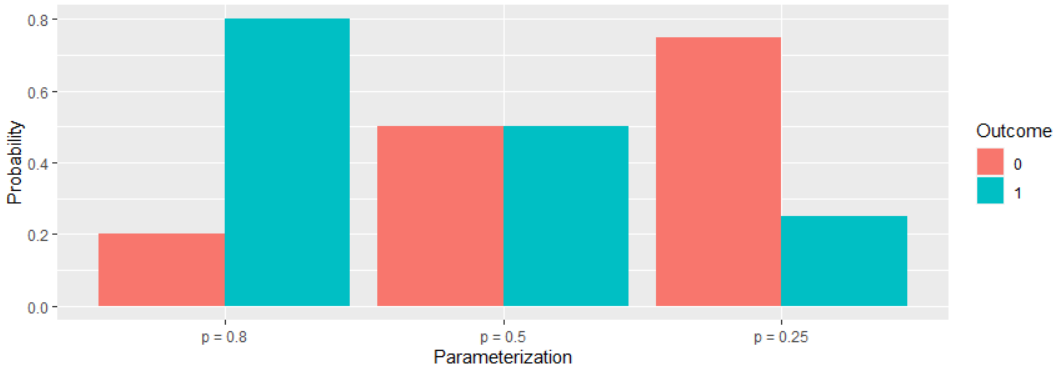


FIGURE E.1: Graphical representation of the Bernoulli distribution via its probability mass function for 3 parameterizations: $p = 0.8$ (left), $p = 0.5$ (middle) and $p = 0.25$ (right)

.

E.1.2 Binomial Distribution

For the *support* $x \in \mathbb{N}$, $p \in [0,1]$ and $n \in \mathbb{N}$ with $n > x$, the *Binomial Distribution* will be described as a short summary of its characteristics, e.g. *Probability Mass Function*, *Cumulative Distribution Function*, *Expectation*, *Variance* and *Mode*:

$$f(x;n,p) = \binom{n}{x} p^x \cdot (1-p)^{n-x}$$

$$F(x;n,p) = I_{1-p}(n-x, 1+x)$$

$$E_X[X] = np$$

$$Var_X[X] = np(1-p)$$

$$\max f(x;n,p) = \lfloor (n+1)p \rfloor$$

With $I_q(\cdot)$ being the *regularized incomplete beta function*, which is defined via the *incomplete beta function* and the *beta function* as follows:

$$I_q(a,b) = \frac{B(q;a,b)}{B(a,b)} = \frac{\int_0^q t^{a-1}(1-t)^{b-1}\mathrm{d}t}{\int_0^1 t^{a-1}(1-t)^{b-1}\mathrm{d}t}$$

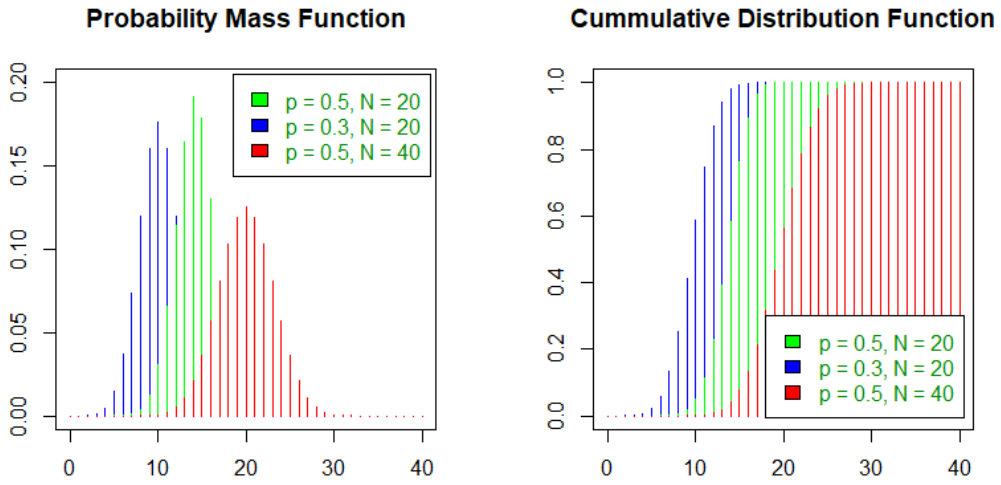


FIGURE E.2: Graphical representation of the Binomial distribution via its probability mass function (left) and its cumulative distribution function (right) for 3 parameterizations.

E.1.3 Uniform Distribution

For the *support* $x \in [a, b]$ within the boundaries $a, b \in \mathbb{R}$ and $a < b$, the *Uniform Distribution* will be described as a short summary of its characteristics, e.g. *Probability Density Function*, *Cumulative Distribution Function*, *Expectation*, *Variance*, *Mode* and *Entropy*:

$$f(x; a, b) = \frac{1}{b-a}$$

$$F(x; a, b) = \frac{x-a}{b-a}$$

$$E_{\mathsf{X}}[\mathsf{X}] = \frac{1}{2}(a+b)$$

$$Var_{\mathsf{X}}[\mathsf{X}] = \frac{1}{12}(b-a)^2$$

$$\max f(x; a, b) = \text{any value in} (a, b)$$

$$H[\mathsf{X}] = \ln(b-a)$$

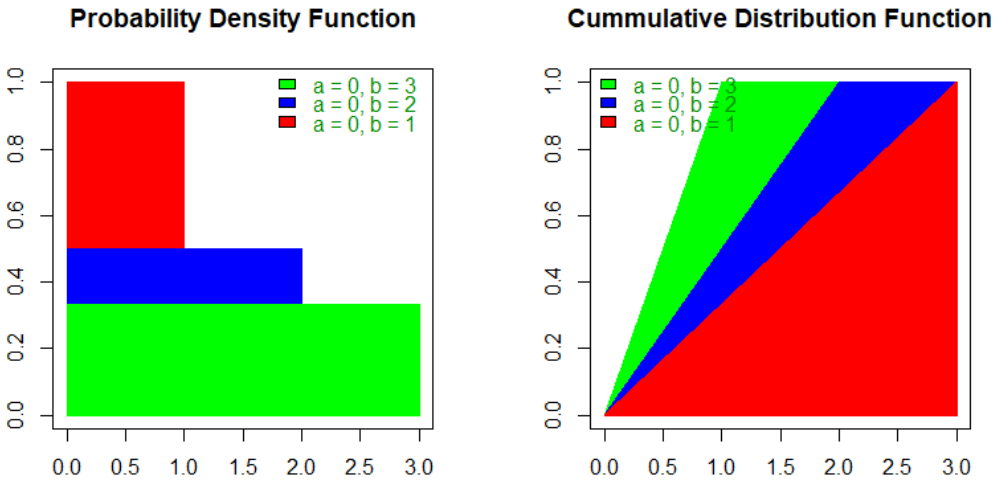


FIGURE E.3: Graphical representation of the Uniform distribution via its probability density function (left) and its cumulative distribution function (right) for 3 parameterizations/intervals $[a, b]$.

E.1.4 Exponential Distribution

For the *support* $x \in \mathbb{R}^+$ and $\lambda \in \mathbb{R}^+$ being the *rate* parameter, the *Exponential Distribution* will be described as a short summary of its characteristics, e.g. *Probability Density Function*, *Cumulative Distribution Function*, *Expectation*, *Variance*, *Mode* and *Entropy*:

$$f(x;\lambda) = \lambda \cdot \exp(-\lambda x)$$

$$F(x;\lambda) = 1 - \exp(-\lambda x)$$

$$E_{\mathsf{X}}[\mathsf{X}] = \frac{1}{\lambda}$$

$$Var_{\mathsf{X}}[\mathsf{X}] = \frac{1}{\lambda^2}$$

$$\max f(x;\lambda) = 0$$

$$H[\mathsf{X}] = 1 - \ln \lambda$$

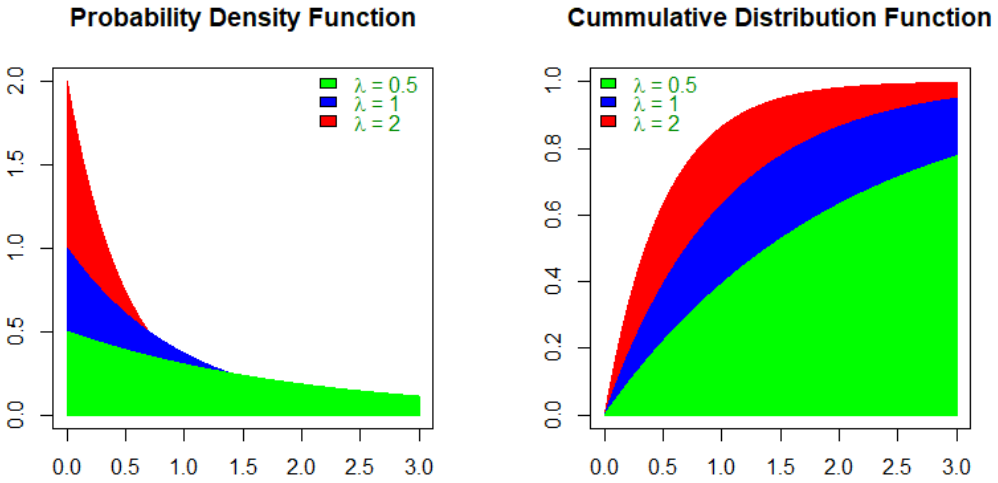


FIGURE E.4: Graphical representation of the Exponential distribution via its probability density function (left) and its cumulative distribution function (right).

E.1.5 Log-Normal Distribution

For the *support* $x \in \mathbb{R}^+$, $\mu \in \mathbb{R}$ and $\sigma \in \mathbb{R}^+$, the *Log-Normal Distribution* will be described as a short summary of its characteristics, e.g. *Probability Density Function*, *Cumulative Distribution Function*, *Expectation*, *Variance*, *Mode* and *Entropy*:

$$f(x;\mu,\sigma^2) = \frac{1}{x\sqrt{2\pi\sigma^2}} \exp\Big(-\frac{(\ln x-\mu)^2}{\sigma^2}\Big)$$

$$F(x;\mu,\sigma^2) = \frac{1}{2}\Big(1+\operatorname{erf}\big(\frac{\ln x-\mu}{\sigma\sqrt{2}}\big)\Big)$$

$$E_{\mathsf{X}}[\mathsf{X}] = \exp\Big(\mu+\frac{\sigma^2}{2}\Big)$$

$$Var_{\mathsf{X}}[\mathsf{X}] = \big(\exp(\sigma^2)-1\big)\cdot\exp(2\mu+\sigma^2)$$

$$\max f(x;\mu,\sigma^2) = \exp(\mu-\sigma^2)$$

$$H[\mathsf{X}] = \log_2(\sigma\exp(\mu+\frac{1}{2})\sqrt{2\pi})$$

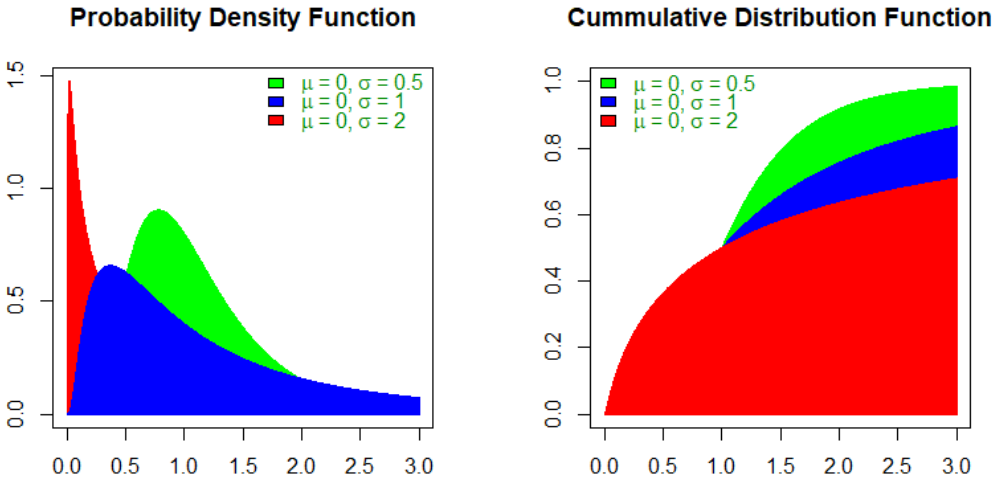


FIGURE E.5: Graphical representation of the Log-Normal distribution via its probability density function (left) and its cumulative distribution function (right).

E.1.6 Gumbel Distribution

The *Gumbel Distribution* is also known as the *log-Weibull* or *double exponential distribution*. For the *support* $x \in \mathbb{R}$, $\mu \in \mathbb{R}$ being the *location* parameter and $\beta \in \mathbb{R}^+$ being the *scale* parameter, the *Gumbel Distribution* will be described as a short summary of its characteristics, e.g. *Probability Density Function*, *Cumulative Distribution Function*, *Expectation*, *Variance*, *Mode* and *Entropy*:

$$f(x;\mu,\beta) = \frac{1}{\beta} \cdot \exp\Big(-\big(\frac{x-\mu}{\beta} + \exp(-\frac{x-\mu}{\beta})\big)\Big)$$

$$F(x;\mu,\beta) = \exp\Big(-\exp\big(-\frac{x-\mu}{\beta}\big)\Big)$$

$$E_X[X] = \mu + \beta\gamma$$

$$Var_X[X] = \frac{\pi^2}{6} \cdot \beta^2$$

$$\max f(x;\mu,\beta) = \mu$$

$$H[X] = \ln\beta + \gamma + 1$$

With γ being *Euler's constant* also known as *Euler–Mascheroni constant*.

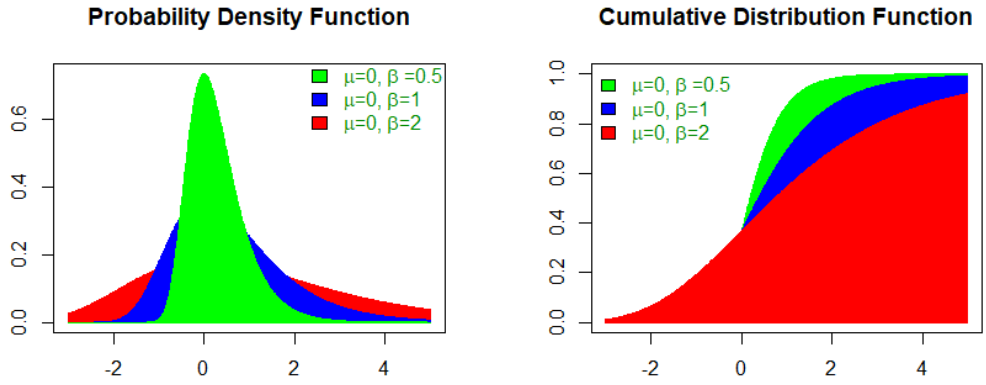


FIGURE E.6: Graphical representation of the Gumbel distribution via its probability density function (left) and its cumulative distribution function (right).

E.1.7 Logistic Distribution

For the *support* $x \in \mathbb{R}$, $\mu \in \mathbb{R}$ being the *location* parameter and $s \in \mathbb{R}^+$ being the *scale* parameter, the *Logistic Distribution* will be described as a short summary of its characteristics, e.g. *Probability Density Function*, *Cumulative Distribution Function*, *Expectation*, *Variance*, *Mode* and *Entropy*:

$$f(x;\mu,s) = \frac{\exp(-\frac{x-\mu}{s})}{s\big(1+\exp(-\frac{x-\mu}{s})\big)^2}$$

$$F(x;\mu,s) = \frac{1}{1+\exp(-\frac{x-\mu}{s})}$$

$$E_{\mathsf{X}}[\mathsf{X}] = \mu$$

$$Var_{\mathsf{X}}[\mathsf{X}] = \frac{s^2 \cdot \pi^2}{3}$$

$$\max f(x;\mu,s) = \mu$$

$$H[\mathsf{X}] = \ln s + 2$$

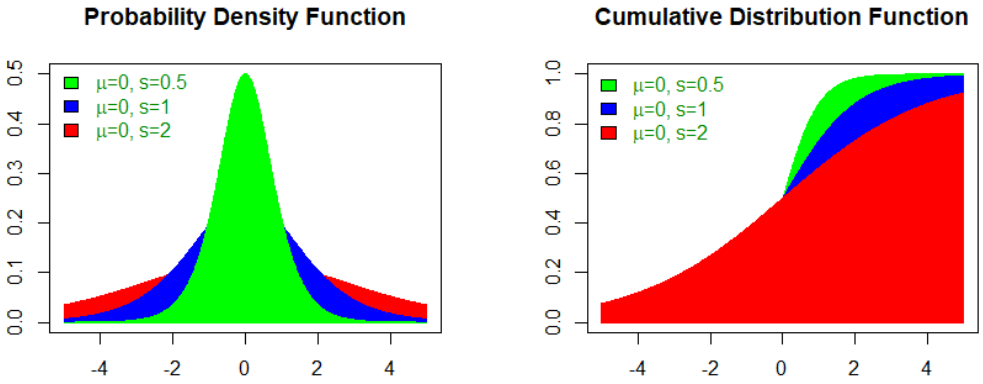


FIGURE E.7: Graphical representation of the Logistic distribution via its probability density function (left) and its cumulative distribution function (right).

E.1.8 Gaussian/Normal Distribution

For the *support* $x \in \mathbb{R}$, $\mu \in \mathbb{R}$ being the *location* parameter and $\sigma \in \mathbb{R}^+$ being the *scale* parameter, the *Gaussian* or *Normal Distribution* will be described as a short summary of its characteristics, e.g. *Probability Density Function*, *Cumulative Distribution Function*, *Expectation*, *Variance*, *Mode* and *Entropy*:

$$f(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\Big(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\Big)$$

$$F(x;\mu,\sigma^2) = \frac{1}{2}\Big(1 + \operatorname{erf}\big(\frac{x-\mu}{\sigma\sqrt{2}}\big)\Big)$$

$$E_{\mathsf{X}}[\mathsf{X}] = \mu$$

$$Var_{\mathsf{X}}[\mathsf{X}] = \sigma^2$$

$$\max f(x;\mu,\sigma^2) = \mu$$

$$H[\mathsf{X}] = \frac{1}{2}\ln(2\pi\sigma^2) + \frac{1}{2}$$

With $\operatorname{erf}(\cdot)$ being the *error function* defined as $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp -t^2 \mathrm{d}t$.

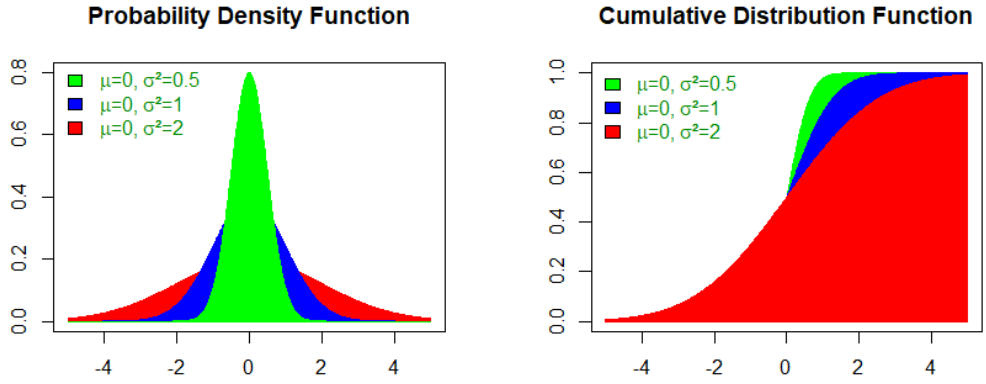


FIGURE E.8: Graphical representation of the Gaussian (or Normal) distribution via its probability density function (left) and its cumulative distribution function (right).

E.2 Multivariate Probability Distributions

E.2.1 Multinomial Distribution

For the *support* $\boldsymbol{x} = (x_1, .., x_A)$ and $x_a \in \mathbb{N}$ with $\sum_a x_a = n$ and $\boldsymbol{p} = (p_1, .., p_A)$ with $p_a \in (0, 1)$ & $\sum_a p_a = 1$, the *Multinomial Distribution* will be described as a short summary of its characteristics, e.g. *Probability Mass Function*, *Expectation*, *Variance*, *Mode* and *Entropy*:

$$
\begin{aligned}
f(\boldsymbol{x}; n, \boldsymbol{p}) &= \binom{n}{x_1, ...x_A} \prod_{a=1}^{A} p_a^{x_a} \\
E_{\mathbf{X}}[\mathbf{X}] &= (E_{\mathsf{X}_1}[\mathsf{X}_1], ..., E_{\mathsf{X}_A}[\mathsf{X}_A]) \\
&= (n \cdot p_1, ..., n \cdot p_A) \\
Var_{\mathsf{X}_a}[\mathsf{X}_a] &= n \cdot p_a \cdot (1 - p_a) \\
Cov[\mathsf{X}_a, \mathsf{X}_b] &= -n \cdot p_a \cdot p_b \text{ for all } a \neq b \\
H[\mathbf{X}] &= -\sum_{a=1}^{A} p_a \ln p_a
\end{aligned}
$$

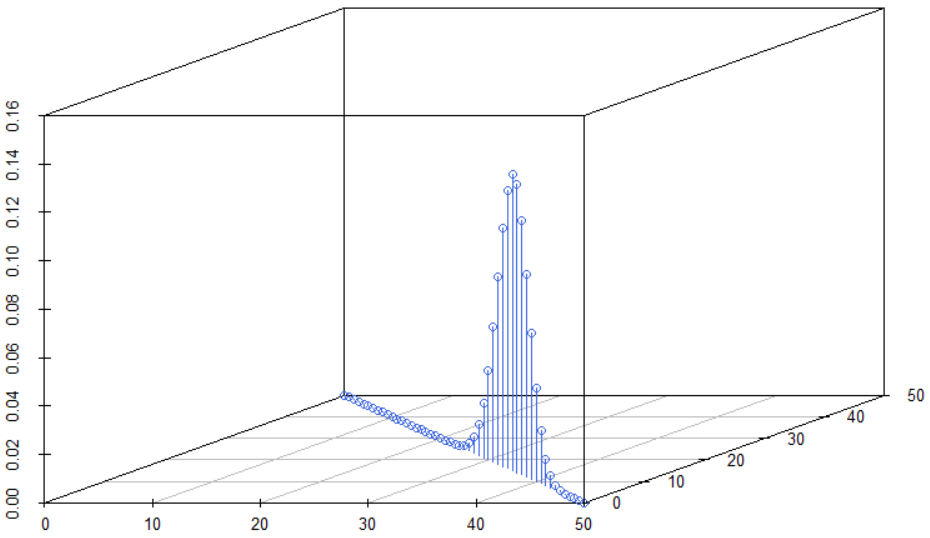


FIGURE E.9: Graphical representation of the Multinomial distribution via its probability mass function. In the 2-dimensional case, the *Binomial Distribution* emerges.

E.2.2 Multivariate Gaussian/Normal Distribution

For the *support* $\boldsymbol{x} \in \mathbb{R}^A$, $\boldsymbol{\mu} \in \mathbb{R}^A$ being the *location* parameter and $\boldsymbol{\Sigma} \in \mathbb{R}^{A \times A}$ being a *Positive Semi-Definite Matrix*, the *Multivariate Gaussian* or *Multivariate Normal Distribution* will be described as a short summary of its characteristics, e.g. *Probability Density Function*, *Expectation (n-dimensional)*, *Variance*, *Mode* and *Entropy*:

$$f(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-\frac{A}{2}} \cdot |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \cdot \exp\Big(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\Big)$$

$$E_{\mathbf{X}}[\mathbf{X}] = \boldsymbol{\mu}$$

$$Var_{\mathbf{X}}[\mathbf{X}] = \boldsymbol{\Sigma}$$

$$\max f(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \boldsymbol{\mu}$$

$$H[\mathbf{X}] = \frac{1}{2}\ln|\boldsymbol{\Sigma}| + \frac{A}{2}(1 + \ln(2\pi))$$

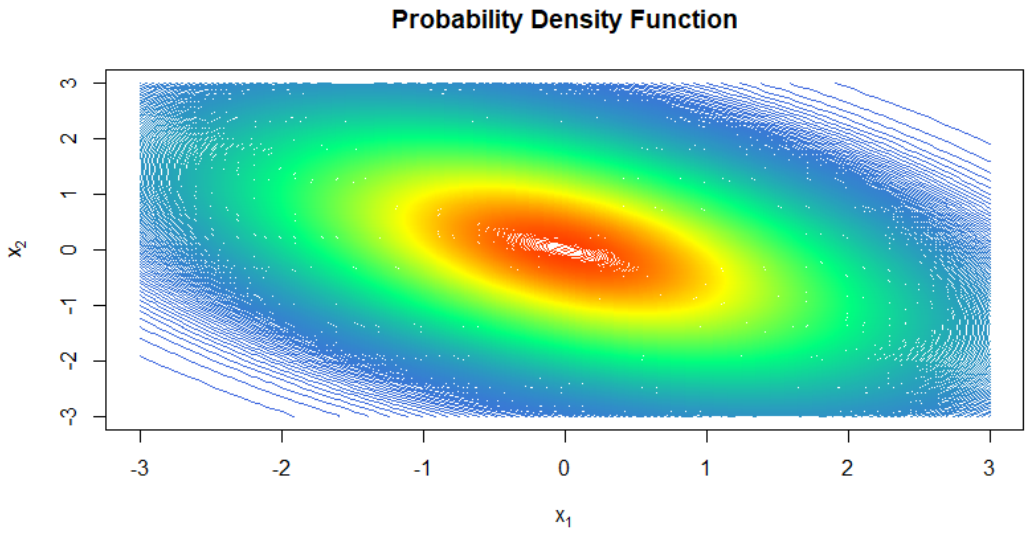


FIGURE E.10: Graphical representation of the multivariate Normal distribution via its probability density function as a heatmap.

E.3 Exponential Family

A probability measure $P_{\mathbf{X}}$ has an exponential representation if its density $f_{\mathbf{X}}(\boldsymbol{x})$ or probability mass function $P_{\mathbf{X}}(\boldsymbol{x})$ at $\boldsymbol{x} \in \mathcal{X}$ can be written as:

$$P(\boldsymbol{x}) = a_0(\boldsymbol{\theta}) \cdot \tau_0(\boldsymbol{x}) \cdot \exp\left(\boldsymbol{\theta}^T\left(\boldsymbol{\tau}(\boldsymbol{x})\right)\right)$$

for $\boldsymbol{\theta} \in \boldsymbol{\Theta} \subseteq \mathbb{R}^A$, $\tau_0 : \mathcal{X} \to \mathbb{R}$, $\boldsymbol{\tau} : \mathcal{X} \to \mathbb{R}^A$ and:

$$a_0(\boldsymbol{\theta}) = \left(\int_{\mathcal{X}} \tau_0(\boldsymbol{x}) \cdot \exp\left(\boldsymbol{\theta}^T\left(\boldsymbol{\tau}(\boldsymbol{x})\right)\right) \mathrm{d}\boldsymbol{x}\right)^{-1}$$

The entire set of all probability measure $P_{\mathbf{X}}$ that have an exponential representation are defined as the *Exponential Family*. Fig. E.1 illustrates the parametrization for a limited set of examples.

Distribution	$\tau_0(\boldsymbol{x})$	$\boldsymbol{\theta}$	$\boldsymbol{\tau}(\boldsymbol{x})$	$a_0(\boldsymbol{\theta})$
Bernoulli (p)	1	$\ln(\frac{p}{1-p})$	x	$(1+\exp(\theta))^{-1}$
Binomial (n,p)	$\binom{n}{x}$	$\ln(\frac{p}{1-p})$	x	$(1+\exp(\theta))^{-n}$
Uniform (a,b)	1	-	-	$(b-a)^{-1}$
Exp (λ)	1	$-\lambda$	x	$-\theta$
Normal (μ, σ^2)	1	$(\frac{\mu}{\sigma^2}; -\frac{1}{2\sigma^2})$	$(x; x^2)$	$\sqrt{-\frac{\theta_2}{\pi}} \cdot \exp(-\frac{\theta_1}{2})$

TABLE E.1: Adaption of [57]: Exponential representation for the *Bernoulli Distribution*, *Binomial Distribution*, *Uniform Distribution*, *Exponential Distribution* and *Gaussian/Normal Distribution*.

Appendix F

Machine Learning

F.1 Learning Problem

The learning problem is defined as the task to find the unknown functional relationship $f \in \mathcal{Y}^{\mathcal{X}}$ between objects $x \in \mathcal{X}$ and targets $y \in \mathcal{Y}$ based on samples $\mathbf{Z} = \{x_i, y_i\}_1^N \in (\mathcal{X}, \mathcal{Y})^N$ drawn independently and identically from the distribution P_{XY} [57]. In case of $\mathcal{Y}$ containing discrete values, it is called a *Classification Learning Problem*. In the classification scenario, the *Bayes Theorem* is often applied as follows:

$$P_{\mathsf{Y}|\mathsf{X}=x}(y) = \frac{P_{\mathsf{X},\mathsf{Y}}(x,y)}{P_{\mathsf{X}}(x)} = \frac{P_{\mathsf{X},\mathsf{Y}}(x,y)}{\sum_{y' \in \mathcal{Y}} P_{\mathsf{X},\mathsf{Y}}(x,y')}$$

For a postulated model f for $P_{\mathsf{Y}|\mathsf{X}=x}(y)$ (in short P_{Z}), the question arises of how much do they diverge from another. The following section will shortly describe approximate measurements of these discrepancies.

F.1.1 Model Evaluation by Data Partitioning

After a postulated model f for the classification task has been learned, its quality has to be measured. In the definition of the *Learning Problem*, the data set was defined as *independently and identically distributed*. Therefore, any arbitrary partitioning of the data set remains identically distributed and any postulated model for P_{Z} should perform comparable on each partitioning. *Evaluation by Data Partitioning* follows this thought by partitioning the entire data set into a *Training Set* for (model-)learning and a *Test Set* for (model-)evaluation. A model is considered to be acceptable, if it generalizes over this partitioning with an adequate performance measurement. Examples for these performance measures are listed below.

F.1.1.1 Confusion Matrix

A *Confusion Matrix* is a *Contingency Table* of the model's prediction for the *Test Set* data in respect to their assigned targets. Several model performance measurements can be derived from a confusion matrix.

Prediction	Assigned Targets	
	$target_1$	$target_2$
$target_1$	TP	FN
$target_2$	FP	TN

TABLE F.1: Confusion Matrix for dichotomous predictions or a binary classification problem.

F.1.1.1.1 Accuracy Derived from a *Confusion Matrix*, the *Accuracy* is defined as follows:

$$\text{Acc} = \frac{TP + TN}{TP + TN + FN + FP}$$

The multivariate extension of an accuracy can be seen as a *Trace of a Matrix* normalized by the sum of all entries, in the case of the matrix being a confusion matrix.

F.1.1.1.2 Precision Derived from a *Confusion Matrix*, the *Precision* or *Positive Predictive Value* is defined as follows:

$$\text{Prec} = \frac{TP}{TP + FP}$$

The multivariate extension of a precision arise as column specific precisions. Each individual precision is just a column normalized measurement for the specific index.

F.1.1.1.3 Recall Derived from a *Confusion Matrix*, the *Recall*, *Sensitivity*, *Hit Rate* or *True Positive Rate* is defined as follows:

$$\text{Rec} = \frac{TP}{TP + FN}$$

The multivariate extension of a recall arise as row specific recalls. Each individual recall is just a row normalized measurement for the specific index.

F.1.1.1.4 F-Score Derived from a *Confusion Matrix*, the *F-score*, *F-measure* or $F_1 - score$ is defined as follows:

$$\begin{aligned}\text{F-score} &= \frac{2}{\text{Rec}^{-1} + \text{Prec}^{-1}} \\ &= 2 \cdot \frac{\text{Prec} \cdot \text{Rec}}{\text{Prec} + \text{Rec}} \\ &= \frac{2 \cdot TP}{2 \cdot TP + FP + FN}\end{aligned}$$

The F-score represents the *harmonic mean* of *Precision* and *Recall*. The multivariate extension can be created by combining the respected multivariate extensions.

F.1.1.2 Jack-Knife

In cases of limited data, partitioning data into two fixed groups, e.g. *Training* and *Test Set*, is a wasteful treatment of the entire set because it reduces the amount of learning examples leading to a reduction of the effectiveness of the learning approach. The approach known as *Jack-Knife* [97] works on dynamic partitioning of the data set, while still making full use of it. Let $k : \{1, ...N\} \to \{1, ...N\}$ be a mapping function to encode the index identity, then $-k(n)$ encodes its complement (all the data except the one with index n). Further, let $f^{-k(n)}(\cdot)$ be a model trained on all data except the one with index n and $L(y_n, f^{-k(n)}(x_n))$ be any performance measure derived from a *Confusion Matrix* for the partitioning on n, than the Jack-Knife is defined as the average over all hold-outs:

$$\text{Jack-Knife} = \frac{1}{N}\sum_{n=1}^{N} L(y_n, f^{-k(n)}(x_n))$$

The method is illustrated in Fig. F.1. Jack-Knife is approximately unbiased, but can have high variance [56]. Additionally, the computational demand increases with the data size because models need to be learned N times.

F.1.1.3 Cross-Validation

The approach known as *Cross-Validation* (CV) [125] is a generalization of the *Jack-Knife* [97] specially designed to lower the computational demand. The mapping function $k : \{1, ...N\} \rightarrow \{1, ...K\}$ is changed to map any data point into one of K partitioning, often called *Folds*. For $K = N$, Cross-Validation is exactly the Jack-Knife but for $K < N$, the computational demand reduces to the factor K. The method is illustrated in Fig. F.1. In most cases, the value of K is set to 5 or 10 without any justification. This approach is often referred to as *5-fold CV*, in case of $K = 5$. CV has lower variance compared to Jack-Knife, but bias could be a problem, depending on how the performance of the learning method varies with the size of the training set [56].

$$\text{Cross-Validation} = \frac{1}{N}\sum_{n=1}^{N} L(y_n, f^{-k(n)}(x_n))$$

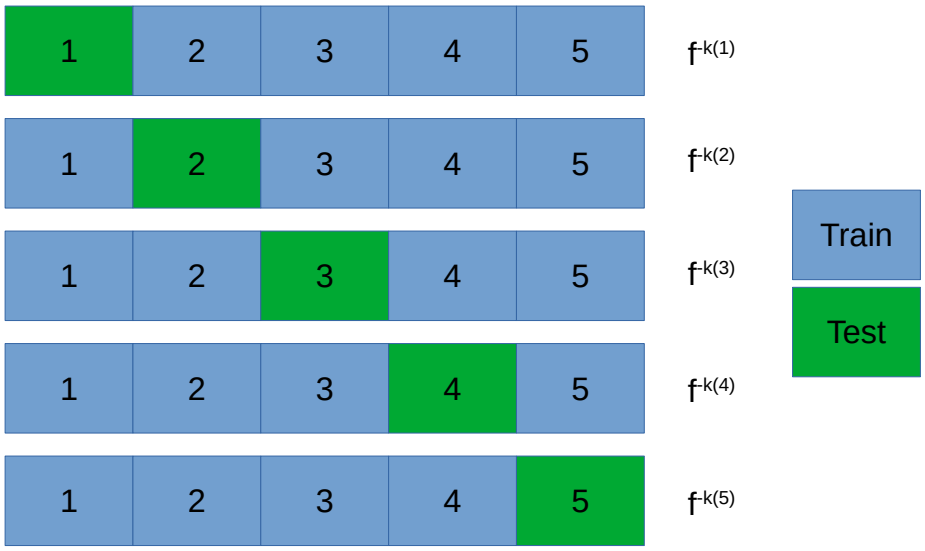


FIGURE F.1: Dynamic partitioning of data into training set and test set. Each partition of the data set is used once for testing, while the remaining data set is used for model training. This technique is used by Jack-Knife [97] and Cross-Validation [125].

F.1.2 Model Evaluation by Statistical Properties

After a postulated model for the classification task has been learned, its quality has to be measured. The definition of the *Learning Problem* follows a probabilistic interpretation. Therefore, the reliability of (probabilistic) models can be addressed via underlying statistics. Classically for a set of candidate models, it is assumed that there is a single correct (or even true) or, at least, best model, and that this model suffices as the sole model for making inferences from the data [20]. In the following, there will be a small discussion of prominent frameworks.

F.1.2.1 Akaike Information Criterion

A model is selected as 'best' if its predictive quality is as close as possible to the empirical samples for the given set of candidate models, while penalizing its complexity. The *Kullback-Leibler Divergence* [74] is a measure between a conceptual reality P_{Z}, and an approximating model f, and this divergence is defined for continuous functions as the following integral:

$$KL(P_{\mathsf{Z}}||f_{\mathsf{Z}}) = -\int_{z\in\mathcal{Z}} P_{\mathsf{Z}}(z)\cdot\ln\frac{P_{\mathsf{Z}}(z)}{f_{\mathsf{Z}}(z)}dz$$

where P_{Z} and f_{Z} are n-dimensional probability distributions [20]. Akaike found a formal relationship between K-L information and Likelihood theory [4]. This relation is defined with an estimator of expected, relative K-L information based on the maximized Likelihood function L, corrected for an asymptotic bias d specified as the number of estimable parameter. The definition of the *Akaike Information Criterion* (AIC) for a model m is as follows:

$$\mathrm{AIC}_m = -2\cdot\ln(L_m) + 2\cdot d_m$$

Given a set of candidate models, the model with the lowest value of the AIC is can be chosen as the best model within that set.

F.1.2.2 Bayesian Information Criterion

First introduced by Schwartz as the *Schwartz criterion* [112], it is more often found in its slightly adapted form as the *Bayesian Information Criterion* (BIC). The BIC arises from the Bayesian approach of model selection. Given a set of candidate models $\{\mathcal{M}_m\}_1^M$, their corresponding parameters $\{\boldsymbol{\theta}_m\}_1^M$ and a data set $\mathbf{Z} = \{x_i, y_i\}_1^N$, the posterior probability for the models is given as follows:

$$\begin{aligned} P(\mathcal{M}_m|\mathbf{Z}) &\propto P(\mathcal{M}_m)\cdot P(\mathbf{Z}|\mathcal{M}_m) \\ &\propto P(\mathcal{M}_m)\cdot\int_{\boldsymbol{\theta}_m\in\boldsymbol{\Theta}_m} P(\mathbf{Z},\boldsymbol{\theta}_m|\mathcal{M}_m) \qquad \mathrm{d}\boldsymbol{\theta}_m \\ &\propto P(\mathcal{M}_m)\cdot\int_{\boldsymbol{\theta}_m\in\boldsymbol{\Theta}_m} P(\mathbf{Z}|\boldsymbol{\theta}_m,\mathcal{M}_m)\cdot P(\boldsymbol{\theta}_m|\mathcal{M}_m) \qquad \mathrm{d}\boldsymbol{\theta}_m \end{aligned}$$

The solution of such integrals often do not exist in a closed form, but approximations have been postulated. The so-called Laplace approximation (and additional simplifications) have been applied as follows:

$$\ln P(\mathbf{Z}|\mathcal{M}_m) = \ln P(\mathbf{Z}|\hat{\boldsymbol{\theta}}_m,\mathcal{M}_m) - \ln N\cdot\frac{d_m}{2} + \mathcal{O}(1)$$

This approximation makes use of the *Maximum Likelihood Estimate* [42] $\hat{\boldsymbol{\theta}}_m$ and the number of free parameters d_m in the model $\mathcal{M}_m$. With the maximized Likelihood $L_m = \ln P(\mathbf{Z}|\hat{\boldsymbol{\theta}}_m, \mathcal{M}_m)$, the generic BIC arise as follows:

$$\text{BIC}_m = -2 \ln L_m + \ln N \cdot d_m$$

Further these approximations can be used to approximate the posterior probability of the models using the *Boltzmann distribution*:

$$\frac{\exp(-\frac{1}{2} \cdot BIC_m)}{\sum_{l=1}^{M} \exp(-\frac{1}{2} \cdot BIC_l)}$$

The BIC is an asymptotically consistent selection criterion, which is not the case if the *Akaike Information Criterion* [4] is applied that tends to choose overly complex models [56]. For finite samples, however, the BIC often chooses models that are too simple [56]. Both information criteria differ only in the coefficient multiplied with the number of parameters and the BIC will penalize complex models harder for large N. In general, models chosen by the BIC will be more parsimonious than those chosen by the AIC [63].

F.1.2.3 Statistical Hypothesis Testing

Complementary to the approach of *Information Theory* using information criteria, classical *Statistical Hypothesis Testing* can be used as well to address the evaluation of models. For that, a *Null Hypothesis* H_0 is constructed, as well as its complementary *Alternative Hypothesis* H_1. The null hypothesis is rejected at predefined level α, if the associated *Test Statistic* exceeds this level. In this abstract description, statistical testing can be characterized by the *contingency table* in Tab F.2. Models are accepted as statistically significant if they suffice a justifiable *Type I error* (measured by α for the *false positives*) or (less commonly used) the *Type II error* (measured by β for the *false negatives*). The following section will provide one example for that framework.

Null Hypothesis H_0 is ...		
	true	not true
accepted	**Right** decision	**Type II error**
rejected	**Type I error**	**Right** decision

TABLE F.2: Contingency table for the dichotomous outcome for H_0 being accepted or rejected in respect to H_0 being true or not true. Such a contingency table comprises the similar interpretation as the *Confusion Matrix* in Tab. F.1

F.1.2.3.1 Kolmogorov-Smirnov-Test Andrei N. Kolmogorov and Nikolai W. Smirnov proposed the *Kolmogorov-Smirnov-Test* [62] (KST) as a measurement to address if the *Random Variable* X and the *Random Variable* Y follow a shared and common distribution. This results in the formulation of the *Null Hypothesis* H_0 and its alternative H_1 as follows:

$$H_0 : F_X(x) = F_Y(x)$$
$$H_1 : F_X(x) \neq F_Y(x)$$

The empirical *cumulative distribution functions* (CDFs) will be compared for their absolute difference as the *Test Statistic*:

$$d_{n,m} = \|F_{X,n} - F_{Y,m}\| = \sup_x |F_{X,n}(x) - F_{Y,m}(x)|$$

The null hypothesis is rejected at level α if:

$$\sqrt{\frac{nm}{n+1}} d_{n,m} > \sqrt{\frac{\ln \frac{2}{\alpha}}{2}} \tag{F.1}$$

Fig. F.2 illustrates the working of the KST in a graphical way. Measurements of data are represented in the histogram on the right. The CDF on the left is an equivalent representation of this measurement. The KST evaluates the maximal difference between the CDFs (indicated by the red dotted line) to address a measurement for $H_0 : F_X(x) = F_Y(x)$.

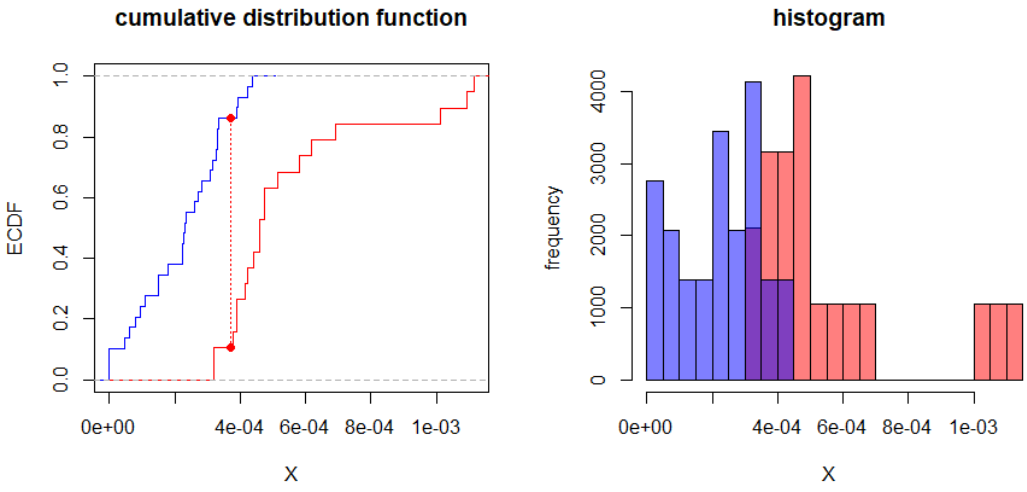


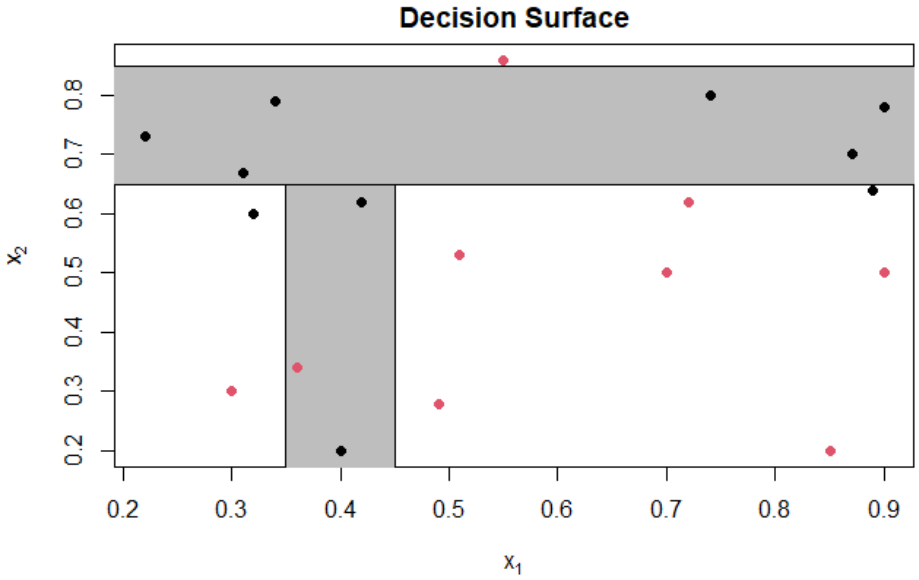
FIGURE F.2: Working of the Kolmogorov-Smirnov-Test (KST). For a given measurement of 2 'classes' (histogram on the right), the KST evaluates the maximal difference of the cumulative distribution functions (indicated by the red dotted line left) to address a measurement for $H_0 : F_X(x) = F_Y(x)$

F.2 Non-Probabilistic Models?

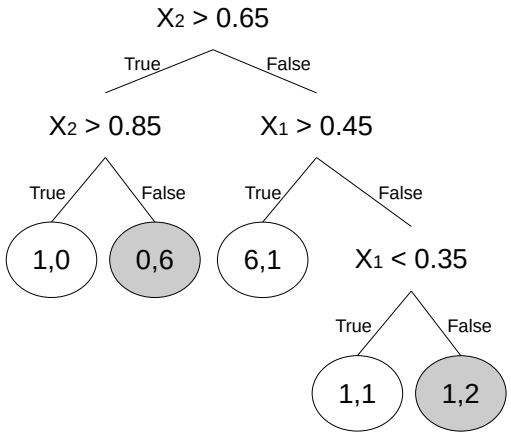
This section will provide a small description of models mentioned in the thesis but not introduced explicitly in Chap. *Modeling*.

F.2.1 Decision Tree

Decision Trees [98] are models of recursively applied rule sets of dichotomous outcome to make inference. An exemplary tree as a sequential rule set is illustrated in Fig. F.3b. For predicting a data point, the sequence starts at the (top) root node and traverses down to the leaves for the prediction. Quite often learned Decision Trees are regularized (e.g. by pruning), so leaves do not necessarily provide perfect data separations but a class distribution, which is used for the *Maximum A Posteriori* prediction. The resulting *Decision Surface* of the Decision Tree in Fig. F.3b is illustrated in Fig.F.3a.



(A) Decision Surface of a Decision Tree



(B) Decision Tree Structure

FIGURE F.3: Up: Decision Surface of the Decision Tree in Fig. F.3b. Bottom: Graphical representation of the recursive rule set of the Decision Tree.

The graphical representation of a Decision Tree, looks structurally comparable to *Bayesian Networks*. Indeed, every path from root to leaves can be represented by n-th order *Markov Models*, with n being the length of the path (-1). All Markov Models within the entire network comprise a variable length. Therefore, these models are called *Variable Order Markov Models* (VOM models) [10]. The *Posterior Predictive Distribution* of such a VOM model can be seen in Fig. F.4 for the scenario in Fig. F.3.

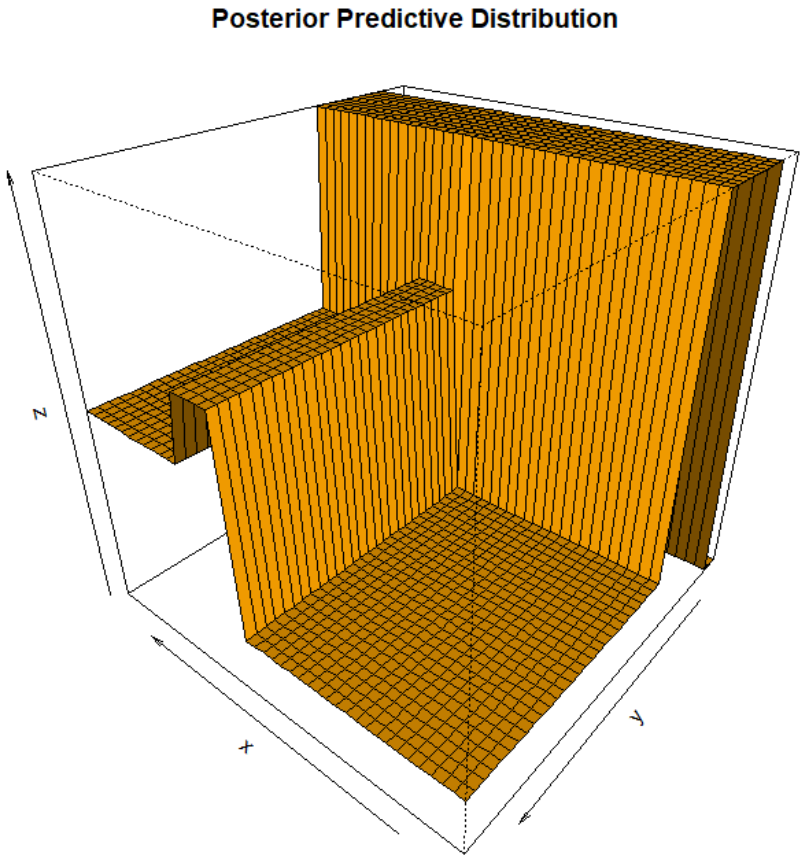


FIGURE F.4: Visualization of the Posterior Predictive Distribution of the scenario in Fig. F.3

F.2.2 Random Forest

In its core *Random Forests* [17] combine three ideas, e.g. *Bootstrap*, *Decision Tree* [98], *Bootstrap Aggregation* (*Bagging*), into one powerful model. For keeping the description as short as possible, *Random Forests* will be described in the *regression* scenario instead of *classification*. The entirety of this section is a reduced transcript of [14]. Given M bootstrap data sets, M different models can be learned ($f_m(\boldsymbol{x})$). The prediction of the committee (COM) will be defined as the bootstrap aggregation (bagging):

$$y_{\text{COM}} = \frac{1}{M} \sum_{m=1}^{M} f_m(\boldsymbol{x})$$

With the true regression function given as $h(\boldsymbol{x})$ and the model specific error $\epsilon_m(\boldsymbol{x})$, then the output of the model follows:

$$f_m(\boldsymbol{x}) = h(\boldsymbol{x}) + \epsilon_m(\boldsymbol{x})$$

The sum-of-squares error will then follow:

$$E_{\mathbf{X}}\left[\left(f_m(\boldsymbol{x}) - h(\boldsymbol{x})\right)^2\right] = E_{\mathbf{X}}\left[\epsilon_m(\boldsymbol{x})^2\right]$$

The average error by the models acting individually follows:

$$E_{\text{AV}} = \frac{1}{M} \sum_{m=1}^{M} E_{\mathbf{X}}[\epsilon_m(\boldsymbol{x})^2]$$

The expected error from the committee is given by:

$$\begin{aligned} E_{\text{COM}} &= E_{\mathbf{X}}\left[\left(\frac{1}{M} \sum_{m=1}^{M} f_m(\boldsymbol{x}) - h(\boldsymbol{x})\right)^2\right] \\ &= E_{\mathbf{X}}\left[\left(\frac{1}{M} \sum_{m=1}^{M} \epsilon_m(\boldsymbol{x})\right)^2\right] \end{aligned}$$

By assuming uncorrelated errors with zero mean, this results in:

$$E_{\text{COM}} = \frac{1}{M} E_{\text{AV}}$$

This observation implies that the average error of a model can be reduced with the factor M by averaging M model versions of it [14]. Typically, the errors are not uncorrelated and the reduction of errors is smaller. For a Decision Tree this approach is quite beneficial, and this approach is formally known as the *Random Forest*.

F.2.3 Support Vector Machine

Support Vector Machines (SVM) [131] are powerful models in Machine Learning (ML) and gave rise to the idea of working with *Kernels*. Nowadays, nearly all existing ML models can be extended towards working with kernels [119] [140]. Within the scope of this thesis, kernels were barely mentioned and therefore this description will not focus towards them. This section will just describe linear SVMs and motivate them as *Discriminative Classifiers*. Basic understanding of Sec. 3.2.1.2 is necessary for the following. Linear Discriminative Classifier are restricted to be a function of the linear combination of their parameters ($\boldsymbol{\theta}$) and the data ($\boldsymbol{x}$), e.g. $f(\boldsymbol{x}^T\boldsymbol{\theta})$. The *Logistic Regression* (LR) [11] model uses the *logistic function* $\sigma(\boldsymbol{x}^T\boldsymbol{\theta}) = f(\boldsymbol{x}^T\boldsymbol{\theta})$ and their super family *Generalized Linear Models* uses any *cumulative distribution function* of the *Exponential Family*, also known as *canonical link functions*. SVMs uses not canonical link functions but the *hinge loss*. Nonetheless, a surprising similarity between both can be stated. The close connection of the LR model and the SVM is shown via their loss functions in Tab. F.3 and Fig. F.5.

Loss Function		Minimizing Function
(-)Binomial Log-Likelihood	$\ln(1+\exp(-Yf(X)))$	$f(X) = \ln \frac{P(Y=1\|X)}{P(Y=-1\|X)}$
Hinge-Loss	$[1-Yf(X)]_+$	$f(X) = \begin{cases} 1 & \text{if } P(Y=1\|X) \geq \frac{1}{2} \\ -1 & \text{otherwise} \end{cases}$

TABLE F.3: Adapted from [56]: Comparison of loss functions between the Logistic Regression [11] model with the Binomial Log-Likelihood and the Support Vector Machine [131] with the Hinge Loss.($[\cdot]_+$ indicates the positive part)

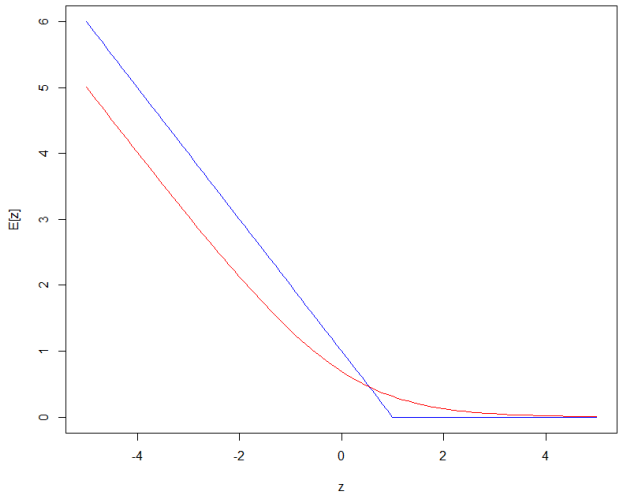


FIGURE F.5: Adapted from [14]: Comparison of loss functions between the Support Vector Machine [131] and the Logistic Regression [11] (rescaled by $\frac{1}{\ln 2}$).

With this striking connection to a probabilistic Discriminative Classifier, it is obvious to question if the SVM comprise a probabilistic nature. For that, an in-depth analysis of the SVM on its formal level is needed. The SVM is defined with regularization constant $C \in \mathbb{R}^+$ as follows:

$$\min_{\boldsymbol{\theta}} \frac{1}{2}||\boldsymbol{\theta}||^2 + C \sum_{n=1}^{N} [1 - y_n f(\boldsymbol{x}_n)]_+$$
$$\min_{\boldsymbol{\theta}} \frac{1}{2}||\boldsymbol{\theta}||^2 + C \sum_{n=1}^{N} max(0, 1 - y_n f(\boldsymbol{x}_n))$$

With the max term being non-differentiable, an equivalent formulation via the so-called *Slack Variable* ξ, is more commonly used:

$$\min_{\boldsymbol{\theta}} \frac{1}{2}||\boldsymbol{\theta}||^2 + C \sum_{n=1}^{N} \xi_n$$
$$\text{s.t. } \xi_n \geq 0, \qquad y_n f(\boldsymbol{x}_n)) \geq 1 - \xi_n, \qquad n = 1, ...N$$

For the probabilistic interpretation $C[1 - yf(x)]_+$, must be interpretable as a negative likelihood, e.g. $P(y = 1|f) = \exp(-C[1 - yf(x)]_+)$ and $P(y = -1|f) = \exp(-C[1 - y(-f(x))]_+)$ by summing over values of y [88]. However, this turns out to be not possible for any $C > 0$ [123]. By relaxing the sum-to-one condition towards a *Pseudo-Likelihood* [95], the probabilistic interpretation of the hinge loss can be derived as:

$$\exp(-2[1 - y_n f(\boldsymbol{x}_n)]_+) = \int_0^\infty \frac{1}{\sqrt{2\pi\lambda_n}} \exp\Big(- \frac{(1 + \lambda_n - y_n f(\boldsymbol{x}_n))}{2\lambda_i} \Big) \mathrm{d}\lambda_n$$

Thus, the exponential of the negative hinge loss can be represented as *Gaussian Scale Mixture* [88]. With that formulation given, SVMs can be learned via *Expectation Maximization* [31] over the latent variables λ_n.

F.2.4 Artificial Neural Network

Artificial Neural Networks (ANN) are powerful models in Machine Learning (ML) that became popular because of their proud community and their advertisement. This section will just describe vanilla *Feed-Forward Neural Networks* and motivate them as *Bayesian Networks*. Basic understanding of Sec. 3.3.5 is necessary for the following. For a K-class classification problem, K units at the top of the network model the K probabilities for each predicted class. Several layers from the data (input) layer through the middle (hidden) layers up to the top (output) layer propagate features in the form of linear combinations through the network. These linear combinations are modeled as follows:

$$\begin{aligned} z_{m,t} &= \sigma(\boldsymbol{x}_t^T \boldsymbol{\theta}_m) \\ \boldsymbol{x}_{t+1} &= \boldsymbol{z}_t \\ f_k(\boldsymbol{x}_1) &= g_k(\boldsymbol{x}_T) \end{aligned}$$

Each unit in the hidden layer uses an *activation function* $\sigma(\cdot)$ on the current input ($\boldsymbol{x}_t$) linear combined with its parameters ($\boldsymbol{\theta}_m$). Each successive unit takes this propagated output as its input ($\boldsymbol{x}_{t+1}$) and propagate it recursively further. The output layer at a predefined depth T ends this recursion with the *softmax function* (the generalization of the *logistic sigmoid*, closely related to the *Boltzmann distribution*):

$$g_k(\boldsymbol{x}_T) = \frac{\exp(x_{k,T})}{\sum_{k'=1}^{K} \exp(x_{k',T})}$$

Common choices for activation functions are the *logistic sigmoid*, *tangens hyperbolicus*, etc. Without conflicts, any *cumulative distribution function* or *probability density function* can be used as an activation function. For such probabilistic activation functions, the described networks are *Bayesian Networks*. Being *Discriminative Classifiers*, the *Maximum Likelihood Estimation* (MLE) [42] for the *Boltzmann distribution* maximizes the *Conditional Log-Likelihood* (CLL), but this estimate follows a recursive approach. Nowadays, implementation rather minimize the negative CLL (*Cross-Entropy*). The *Graphical Model* [88] of the presented approach can be found in Fig. F.6. Obviously, the height of such *Bayesian Networks* can be set arbitrary by incorporating more and more *latent variables* in the network. Fortunately, by being a probabilistic model in nature, adequate statistical testing and model selection will identify overly complex designs.

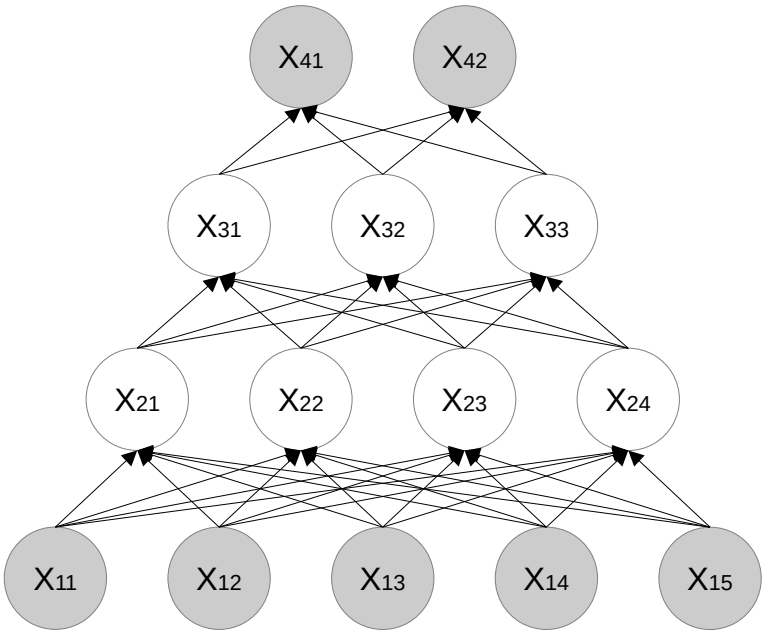


FIGURE F.6: The Graphical Model of an Artificial Neural Network. In case of activation functions being cumulative distribution function or probability density function, such networks are *Bayesian Networks*. Nodes reflect random variables and edges their interaction: $A \rightarrow B = P(B|A)$. Latent (unknown) variables are marked white.

Bibliography

[1] Tobii Ab. *User's manual Tobii Studio*. Version 3.4.7. 2016.

[2] Larry A. Abel, B. Todd Troost, and Louis F. Dell'Osso. "The effects of age on normal saccadic characteristics and their variability". In: *Vision Research* 23.1 (1983), pp. 33–37. DOI: 10.1016/0042-6989(83)90038-x.

[3] Mikhail Ageev et al. "Find it if you can: a game for modeling different types of web search success using interaction data". In: *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. 2011, pp. 345–354. DOI: 10.1145/2009916.2009965.

[4] Hirotogu Akaike. "A new look at the statistical model identification". In: *IEEE Transactions on Automatic Control* 19.6 (1974), pp. 716–723. DOI: 10.1109/TAC.1974.1100705.

[5] Kumaripaba Athukorala et al. "Beyond Relevance: Adapting Exploration / Exploitation in Information Retrieval". In: *Proceedings of the 21st International Conference on Intelligent User Interfaces*. 2016, pp. 359–369. DOI: 10.1145/2856767.2856786.

[6] Kumaripaba Athukorala et al. "Is Exploratory Search Different? A Comparison of Information Search Behavior for Exploratory and Lookup Tasks". In: *Journal of the Association for Information Science and Technology* 67.11 (2016), 2635–2651. DOI: 10.1002/asi.23617.

[7] Anne Aula, Rehan M. Khan, and Zhiwei Guan. "How Does Search Behavior Change As Search Becomes More Difficult?" In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2010, pp. 35–44. DOI: 10.1145/1753326.1753333.

[8] Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999. ISBN: 9780201398298.

[9] Mathias Bauer, Piotr J. Gmytrasiewicz, and Julita Vassileva, eds. *User Modeling 2001*. 2001. ISBN: 978-3-540-42325-6. DOI: 10.1007/3-540-44566-8.

[10] Ron Begleiter, Ran El-Yaniv, and Golan Yona. "On Prediction Using Variable Order Markov Models". In: *Journal of Artificial Intelligence Research* 22 (2011), pp. 385–421. DOI: 10.1613/jair.1491.

[11] Joseph Berkson. "Application of the Logistic Function to Bio-Assay". In: *Journal of the American Statistical Association* 39.227 (1944), pp. 357–365. DOI: 10.2307/2280041.

[12] Ralf Biedert et al. "A Robust Realtime Reading-Skimming Classifier". In: *Proceedings of the Symposium on Eye Tracking Research and Applications*. 2012, pp. 123–130. DOI: 10.1145/2168556.2168575.

[13] James E. Birren, Roland C. Casperson, and Jack Botwinick. "Age Changes in Pupil Size". In: *Journal of Gerontology* 5.3 (1950), pp. 216–221. DOI: 10.1093/geronj/5.3.216.

[14] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007. ISBN: 978-0-387-31073-2.

[15] Chester I. Bliss. "The Method of Probits". In: *Science* 79.2037 (1934), pp. 38–39. DOI: 10.1126/science.79.2037.38.

[16] José Borges and Mark Levene. "Data Mining of User Navigation Patterns". In: *Web Usage Analysis and User Profilin*. 2000, pp. 92–111. DOI: 10.1007/3-540-44934-5_6.

[17] Leo Breiman. "Random Forests". In: *Machine Learning* 45 (2001), pp. 5–32. DOI: 10.1023/A:1010933404324.

[18] Peter Brusilovsky, Albert T. Corbett, and Fiorella de Rosis, eds. *User Modeling 2003*. 2003. ISBN: 3-540-40381-7. DOI: 10.1007/3-540-44963-9.

[19] Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, eds. *The Adaptive Web, Methods and Strategies of Web Personalization*. 2007. ISBN: 978-3-540-72078-2. DOI: 10.1007/978-3-540-72079-9.

[20] Kenneth P. Burnham and David R. Anderson. "Multimodel Inference: Understanding AIC and BIC in Model Selection". In: *Sociological Methods & Research* 33.2 (2004), pp. 261–304. DOI: 10.1177/0049124104268644.

[21] Rudy Den Buurman, Theo Roersema, and Jack F. Gerrissen. "Eye Movements and the Perceptual Span in Reading". In: *Reading Research Quarterly* 16.2 (1981), pp. 227–235. DOI: 10.2307/747557.

[22] Christopher S. Campbell and Paul P. Maglio. "A Robust Algorithm for Reading Detection". In: *Proceedings of the Workshop on Perceptive User Interfaces*. 2001, 1–7. DOI: 10.1145/971478.971503.

[23] Patricia A. Carpenter and Marcel A. Just. "What your eyes do while your mind is reading". In: *Eye movements in reading: Perceptual and language processes*. 1983, pp. 275–307. DOI: 10.1016/B978-0-12-583680-7.50022-9.

[24] Philip K. Chan. "Constructing Web User Profiles: A Non-Invasive Learning Approach". In: *Web Usage Analysis and User Profiling*. 2000, pp. 39–55. DOI: 10.1007/3-540-44934-5_3.

[25] Malcolm Clark et al. "Looking for genre: the use of structural features during search tasks with Wikipedia". In: *Proceedings of the 4th Information Interaction in Context Symposium*. 2012, pp. 145–154. DOI: 10.1145/2362724.2362751.

[26] Malcolm Clark et al. "You have e-mail, what happens next? Tracking the eyes for genre". In: *Information Processing & Management* 50.1 (2014), pp. 175–198. DOI: 10.1016/j.ipm.2013.08.005.

[27] Jacob Cohen. "A Coefficient of Agreement for Nominal Scales". In: *Educational and Psychological Measurement* 20.1 (1960), pp. 37 –46. DOI: 10.1177/001316446002000104.

[28] Michael J. Cole et al. "Discrimination between tasks with user activity patterns during information search". In: *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. 2014, pp. 567–576. DOI: 10.1145/2600428.2609591.

[29] F. Crestani and G. Pasi. *Soft Computing in Information Retrieval: Techniques and Applications*. Physica-Verlag HD, 2013. ISBN: 9783790818499.

[30] John N. Darroch and Douglas Ratcliff. "Generalized Iterative Scaling for Log-Linear Models". In: *The Annals of Mathematical Statistics* 43.5 (1972), pp. 1470–1480. DOI: 10.1214/aoms/1177692379.

[31] Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. "Maximum Likelihood from Incomplete Data Via the EM Algorithm". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1 (1977), pp. 1–22. DOI: 10.1111/j.2517-6161.1977.tb01600.x.

[32] Pedro Domingos and Michael J. Pazzani. "On the Optimality of the Simple Bayesian Classifier under Zero-One Loss". In: *Machine Learning* 29 (1997), pp. 103–130. DOI: 10.1023/A:1007413511361.

[33] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. 2nd ed. Wiley, 2001. ISBN: 978-0-471-05669-0.

[34] Geoffrey B. Duggan and Stephen J. Payne. "Skim Reading by Satisficing: Evidence from Eye Tracking". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2011, pp. 1141–1150. DOI: 10.1145/1978942.1979114.

[35] Richard Durbin et al. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998. ISBN: 978-0521629713.

[36] Susan F. Ehrlich and Keith Rayner. "Contextual effects on word perception and eye movements during reading". In: *Journal of Verbal Learning and Verbal Behavior* 20.6 (1981), pp. 641–655. DOI: 10.1016/S0022-5371(81)90220-6.

[37] David Ellis. "A Behavioral Approach to Information Retrieval System Design". In: *Journal of Documentation* 45.3 (1989), pp. 171–212. DOI: 10.1108/eb026843.

[38] David Ellis, Deborah Cox, and Katherine Hall. "A comparison of the information seeking patterns of researchers in the physical and social sciences". In: *Journal of Documentation* 49.4 (1993), pp. 356–369. DOI: 10.1108/eb026919.

[39] David Ellis and Merete Haugan. "Modelling the information seeking patterns of engineers and research scientists in an industrial environment". In: *Journal of Documentation* 53.4 (1997), pp. 384–403. DOI: 10.1108/EUM0000000007204.

[40] Yariv Ephraim, David Malah, and Biing-Hwang Juang. "On the application of hidden Markov models for enhancing noisy speech". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 37.12 (1989), pp. 1846–1856. DOI: 10.1109/29.45532.

[41] Gustav T. Fechner. *Elemente der Psychophysik*. Vol. 2. Breitkopf und Härtel, 1860. URL: https://archive.org/details/elementederpsych02fech/mode/2up.

[42] Ronald A. Fisher. "On an Absolute Criterion for Fitting Frequency Curves". In: Messenger of Mathematics 41 (1912), pp. 155–160.

[43] Ronald A. Fisher. "The Use of Multiple Measurements in Taxonomic Problems". In: *Annals of Eugenics* 7.2 (1936), pp. 179–188. DOI: 10.1111/j.1469-1809.1936.tb02137.x.

[44] Jerome H. Friedman. "Regularized Discriminant Analysis". In: *Journal of the American Statistical Association* 84.405 (1989), pp. 165–175. DOI: 10.2307/2289860.

[45] Wenjiang J. Fu. "Penalized Regressions: The Bridge versus the Lasso". In: *Journal of Computational and Graphical Statistics* 7.3 (1998), pp. 397–416. DOI: 10.2307/1390712.

[46] Zoubin Ghahramani. "An Introduction to Hidden Markov Models and Bayesian Networks." In: *International Journal of Pattern Recognition and Artificial Intelligence* 15.1 (2001), pp. 9–42. DOI: 10.1142/S0218001401000836.

[47] Joshua Goodman. "Sequential Conditional Generalized Iterative Scaling". In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. 2002, 9–16. DOI: 10.3115/1073083.1073086.

[48] Tatiana Gossen, Juliane Höbel, and Andreas Nürnberger. "Usability and Perception of Young Users and Adults on Targeted Web Search Engines". In: *Proceedings of the 5th Information Interaction in Context Symposium*. 2014, pp. 18–27. DOI: 10.1145/2637002.2637007.

[49] Dan Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, 1997. ISBN: 0521585198.

[50] Isabelle Guyon and André Elisseeff. "An Introduction to Variable and Feature Selection". In: *Journal of Machine Learning Research* 3 (2003), 1157–1182.

[51] Jacek Gwizdka and Irene Lopatovska. "The role of subjective factors in the information search process". In: *Journal of the American Society for Information Science and Technology* 60.12 (2009), pp. 2452–2464. DOI: 10.1002/asi.21183.

[52] Akın Gündüz and Tarek Najjar. "Analysis Eye Movements During Reading by Machine Learning Algorithms: A Review Paper". In: *IEEE Symposium Series on Computational Intelligence*. 2018, pp. 1069–1075. DOI: 10.1109/SSCI.2018.8628799.

[53] Louise Hainline et al. "Characteristics of saccades in human infants". In: *Vision Research* 24.12 (1984), pp. 1771–1780. DOI: 10.1016/0042-6989(84)90008-7.

[54] Ahmed Hassan, Rosie Jones, and Kristina Lisa Klinkner. "Beyond DCG: user behavior as a predictor of a successful search". In: *Proceedings of the third ACM international conference on Web search and data mining*. 2010, pp. 221–230. DOI: 10.1145/1718487.1718515.

[55] Ahmed Hassan et al. "Struggling or Exploring?: Disambiguating Long Search Sessions". In: *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*. 2014, pp. 53–62. DOI: 10.1145/2556195.2556221.

[56] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference and prediction*. 2nd ed. Springer, 2009. ISBN: 978-0-387-84858-7.

[57] Ralf Herbrich. *Learning Kernel Classifiers - Theory and Algorithms*. MIT Press, 2002. ISBN: 978-0262083065.

[58] Magnus R. Hestenes and Eduard Stiefel. "Methods of conjugate gradients for solving linear systems". In: *Journal of Research of the National Bureau of Standards* 49.6 (1952), pp. 409–435. DOI: 10.6028/jres.049.044.

[59] Kenneth Holmqvist et al. "Chapter 30 - Reading or Scanning? A Study of Newspaper and Net Paper Reading". In: *The Mind's Eye*. 2003, pp. 657–670. DOI: 10.1016/B978-044451020-4/50035-9.

[60] Hugo C Huurdeman and Jaap Kamps. "From multistage information-seeking models to multistage search systems". In: *Proceedings of the 5th Information Interaction in Context Symposium*. 2014, pp. 145–154. DOI: 10.1145/2637002.2637020.

[61] Mitsuo Ikeda, Shinya Saida, and Takashi Sugiyama. "Visual field size necessary for length comparison". In: *Perception & Psychophysics* 22 (1977), pp. 165–170. DOI: 10.3758/BF03198750.

[62] Frank J. Massey Jr. "The Kolmogorov-Smirnov Test for Goodness of Fit". In: *Journal of the American Statistical Association* 46.253 (1951), pp. 68–78. DOI: 10.2307/2280095.

[63] Joseph B. Kadane and Nicole A. Lazar. "Methods and Criteria for Model Selection". In: *Journal of the American Statistical Association* 99.465 (2004), pp. 279–290. DOI: 10.1198/016214504000000269.

[64] K. Rao Kadiyala and Kambhampati S. R. Murthy. "Estimation of Regression Equations with Cauchy Disturbances". In: *Canadian Journal of Statistics* 5.1 (1977), pp. 111–120. DOI: 10.2307/3315088.

[65] Conor Kelton et al. "Reading Detection in Real-Time". In: *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*. 2019, pp. 1–5. DOI: 10.1145/3314111.3319916.

[66] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009. ISBN: 978-0262013192.

[67] Sepp Kollmorgen and Kenneth Holmqvist. *Automatically Detecting Reading in Eye Tracking Data*. Tech. rep. 144. Lund University Cognitive Studies, 2009. URL: https://www.lucs.lu.se/fileadmin/user_upload/project/lucs/LUCS-pub/LUCS-144.pdf.

[68] Michael Kotzyba et al. "Exploration or Fact-Finding: Inferring User's Search Activity Just in Time". In: *Proceedings of the Conference on Human Information Interaction and Retrieval*. 2017, 87–96. DOI: 10.1145/3020165.3020180.

[69] Michael Kotzyba et al. "Model-Based Frameworks for User Adapted Information Exploration: An Overview." In: *Companion Technology*. 2017, pp. 37–56. DOI: 10.1007/978-3-319-43665-4_3.

[70] Michael Kotzyba et al. "The Effect of Motivational Goals on Information Search for Tasks of Varying Complexity Levels". In: *IEEE International Conference on Systems, Man, and Cybernetics*. 2018, pp. 2602–2607. DOI: 10.1109/SMC.2018.00445.

[71] Eileen Kowler and Albert J. Martins. "Eye Movements of Preschool Children". In: *Science* 215.4535 (1982), pp. 997–999. DOI: 10.1126/science.7156979.

[72] Carol C. Kuhlthau. "Inside the Search Process: Information Seeking from the User's Perspective." In: *Journal of the American Society for Information Science* 42.5 (1991), pp. 361–371. DOI: 10.1002/(SICI)1097-4571(199106)42:5<361::AID-ASI6>3.0.CO;2-%23.

[73] Carol C. Kuhlthau. *Seeking Meaning: A Process Approach to Library and Information Services*. Libraries Unlimited, 2004. ISBN: 9781591580942.

[74] Solomon Kullback and Richard A. Leibler. "On Information and Sufficiency". In: *Annals of Mathematical Statistics* 22.1 (1951), pp. 79–86. DOI: 10.1214/aoms/1177729694.

[75] J. Richard Landis and Gary G. Koch. "The Measurement of Observer Agreement for Categorical Data". In: *Biometrics* 33.1 (1977), pp. 159–174. DOI: 10.2307/2529310.

[76] Niels Landwehr et al. "A Model of Individual Differences in Gaze Control During Reading". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2014, pp. 1810–1815. DOI: 10.3115/v1/D14-1192.

[77] Jingjing Liu et al. "Can Search Systems Detect Users' Task Difficulty?: Some Behavioral Signals". In: *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2010, pp. 845–846. DOI: 10.1145/1835449.1835645.

[78] Christopher Manning and Hinrich Schutze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999. ISBN: 9780262133609.

[79] Gary Marchionini. "Exploratory search: from finding to understanding". In: *Communications of the ACM* 49.4 (2006), pp. 41–46. DOI: 10.1145/1121949.1121979.

[80] Miriam Martínez and Luis Sucar. "Learning Dynamic Naive Bayesian Classifiers." In: *Proceedings of the 21th International Florida Artificial Intelligence Research Society Conference*. 2008, pp. 655–659.

[81] Andrew McCallum, Dayne Freitag, and Fernando C. N. Pereira. "Maximum Entropy Markov Models for Information Extraction and Segmentation". In: *Proceedings of the Seventeenth International Conference on Machine Learning*. 2000, 591–598.

[82] George W. McConkie and Keith Rayner. "Asymmetry of the perceptual span in reading". In: *Bulletin of the psychonomic society* 8 (1976), pp. 365–368. DOI: 10.3758/BF03335168.

[83] P. McCullagh and John A. Nelder. *Generalized Linear Models*. 2nd ed. Chapman & Hall, 1989. ISBN: 9780412317606. DOI: 10.1201/9780203753736.

[84] Alessandro Micarelli, Filippo Sciarrone, and Mauro Marinilli. "Web Document Modeling". In: *The Adaptive Web: Methods and Strategies of Web Personalization*. 2007, pp. 155–192. DOI: 10.1007/978-3-540-72079-9_5.

[85] Robert E. Morrison. "Manipulation of stimulus onset delay in reading: Evidence for parallel programming of saccades". In: *Journal of Experimental Psychology: Human Perception and Performance* 10.5 (1984), pp. 667–682. DOI: 10.1037/0096-1523.10.5.667.

[86] Robert E. Morrison. "Retinal image size and the perceptual span in reading". In: *Eye movements in reading: Perceptual and language processes*. 1983, pp. 31–40. DOI: 10.1016/B978-0-12-583680-7.X5001-2.

[87] Robert E. Morrison and Keith Rayner. "Saccade size in reading depends upon character spaces and not visual angle". In: *Perception & Psychophysics* 30.4 (1981), pp. 395–396. DOI: 10.3758/bf03206156.

[88] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012. ISBN: 9780262018029.

[89] Andrew Y. Ng and Michael I. Jordan. "On Discriminative vs. Generative Classifiers: A Comparison of Logistic Regression and Naive Bayes". In: *Advances in Neural Information Processing Systems*. Vol. 14. 2001, 841–848.

[90] R. Nilsson and kemi och biologi Linköpings universitet. Institutionen för fysik. *Statistical Feature Selection: With Applications in Life Science*. Department of Physcis, Chemistry and Biology, Linköping University, 2007. ISBN: 9789185715244.

[91] J. Kevin O'Regan. "Optimal Viewing Position in Words and the Strategy-Tactics Theory of Eye Movements in Reading". In: *Eye Movements and Visual Cognition: Scene Perception and Reading*. 1992, pp. 333–354. DOI: 10.1007/978-1-4612-2852-3_20.

[92] Karl Pearson. "X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling". In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 50.302 (1900), pp. 157–175. DOI: 10.1080/14786440009463897.

[93] Francis J. Pirozzolo and Edward C. Hansch. "Oculomotor Reaction Time in Dementia Reflects Degree of Cerebral Dysfunction". In: *Science* 214.4518 (1981), pp. 349–351. DOI: 10.1126/science.7280699.

[94] Alexander Pollatsek et al. "Asymmetries in the perceptual span for Israeli readers". In: *Brain and Language* 14.1 (1981), pp. 174–180. DOI: 10.1016/0093-934X(81)90073-0.

[95] Nicholas G. Polson and Steven L. Scott. "Data augmentation for support vector machines". In: *Bayesian Analysis* 6 (2011), pp. 1–23. DOI: 10.1214/11-BA601.

[96] Tobii Pro. *Tobii Pro. How do tobii eye trackers work?* URL: https://www.tobiipro.com/learn-and-support/learn/eye-tracking-essentials/how-do-tobii-eye-trackers-work/.

[97] Maurice H. Quenouille. "Notes on Bias in Estimation". In: *Biometrika* 43.3/4 (1956), pp. 353–360. DOI: 10.1093/biomet/43.3-4.353.

[98] John R. Quinlan. "Induction of Decision Trees". In: *Machine Learning* 1 (1986), pp. 81–106. DOI: 10.1023/A:1022643204877.

[99] Lawrence R. Rabiner. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition". In: *Readings in Speech Recognition*. 1990, 267–296. DOI: 10.1109/5.18626.

[100] Keith Rayner. "Eye movements and the perceptual span in beginning and skilled readers". In: *Journal of Experimental Child Psychology* 41.2 (1986), pp. 211–236. DOI: 10.1016/0022-0965(86)90037-8.

[101] Keith Rayner. "Eye movements in reading and information processing: 20 years of research". In: *Psychological Bulletin* 124.3 (1998), pp. 372–422. DOI: 10.1037/0033-2909.124.3.372.

[102] Keith Rayner. "Visual selection in reading, picture perception, and visual search: A tutorial review". In: *Attention and performance* 10 (1984), pp. 67–96.

[103] Keith Rayner and Monica Castelhano. "Eye movements during reading, scene perception, visual search, and while looking at print advertisements". In: *Visual Marketing: From Attention to Action* 1.1 (2007), pp. 9–42. DOI: 10.4324/9780203809617.

[104] Keith Rayner and Susan A. Duffy. "On-line comprehension processes and eye movements in reading". In: *Reading research: Advances in theory and practice*. 1988, pp. 13–66.

[105] Keith Rayner and George W. McConkie. "What guides a reader's eye movements?" In: *Vision Research* 16.8 (1976), pp. 829–837. DOI: 10.1016/0042-6989(76)90143-7.

[106] Berthier Ribeiro-Neto, Ilmério Silva, and Richard Muntz. "Bayesian Network Models for Information Retrieval". In: *Soft Computing in Information Retrieval: Techniques and Applications*. 2000, pp. 259–291. DOI: 10.1007/978-3-7908-1849-9_11.

[107] Christian P. Robert and George Casella. *Introducing Monte Carlo Methods with R*. 1st. Springer, 2009. ISBN: 1441915753.

[108] Stephen Robertson and K. Sparck Jones. "Relevance weighting of search terms". In: *Journal of the American Society for Information Science* 27.3 (1976), pp. 129–146. DOI: 10.1002/asi.4630270302.

[109] Paige Rodeghero and Collin McMillan. "An Empirical Study on the Patterns of Eye Movement during Summarization Tasks". In: *2015 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. 2015, pp. 1–10. DOI: 10.1109/ESEM.2015.7321188.

[110] Gerard Salton, A. Wong, and Chung-Shu Yang. "A Vector Space Model for Automatic Indexing". In: *Communications of the ACM* 18.11 (1975), pp. 613–620. DOI: 10.1145/361219.361220.

[111] Andries F. Sanders. "Processing information in the functional visual field". In: *Perception and Cognition: Advances in eyemovement research*. 1993, pp. 3–22.

[112] Gideon Schwarz. "Estimating the Dimension of a Model". In: *The Annals of Statistics* 6.2 (1978), pp. 461–464. DOI: 10.1214/aos/1176344136.

[113] Johannes Schwerdt. *Introducing a Gibbs Sampling Algorithm for De-novo Motif Discovery in the Transcriptional Hourglass Pattern in Embryogenesis*. Master Thesis. 2013.

[114] Johannes Schwerdt, Michael Kotzyba, and Andreas Nürnberger. "Fact-Finding or Exploration: Characterizing Reading Strategies in User's Search Activities". In: *2021 IEEE 2nd International Conference on Human-Machine Systems*. 2021, pp. 1–6. DOI: 10.1109/ICHMS53169.2021.9582460.

[115] Johannes Schwerdt, Michael Kotzyba, and Andreas Nürnberger. "Fact-Finding or Exploration: Identifying Latent Behavior Clusters in User's Search Activities". In: *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. 2019, pp. 1465–1471. DOI: 10.1109/SMC.2019.8914225.

[116] Johannes Schwerdt, Michael Kotzyba, and Andreas Nürnberger. "Inferring user's search activity using interaction logs and gaze data". In: *2017 International Conference on Companion Technology*. 2017, pp. 1–6. DOI: 10.1109/COMPANION.2017.8287075.

[117] Johannes Schwerdt and Andreas Nürnberger. "Automatic Reading Detection during Online Search Sessions". In: *Adjunct Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization*. 2022, 13–17. DOI: 10.1145/3511047.3536418.

[118] Johannes Schwerdt et al. "An Explorative Tool for Mutation Tracking in the Spike Glycoprotein of SARS-CoV-2". In: *2021 IEEE 2nd International Conference on Human-Machine Systems*. 2021, pp. 1–6. DOI: 10.1109/ICHMS53169.2021.9582636.

[119] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. "Nonlinear Component Analysis as a Kernel Eigenvalue Problem". In: *Neural Computation* 10.5 (1998), pp. 1299–1319. DOI: 10.1162/089976698300017467.

[120] Chirag Shah and Roberto González-Ibáñez. "Exploring information seeking processes in collaborative search tasks". In: *Proceedings of the American Society for Information Science and Technology* 47.1 (2010), pp. 1–7. DOI: 10.1002/meet.14504701211.

[121] Chirag Shah, Chathra Hendahewa, and Roberto González-Ibáñez. "Rain or shine? forecasting search process performance in exploratory search tasks". In: *Journal of the Association for Information Science and Technology* 67.7 (2015). DOI: 10.1002/asi.23484.

[122] Claude E. Shannon. "A mathematical theory of communication". In: *The Bell System Technical Journal* 27.3 (1948), pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[123] Peter Sollich. "Bayesian Methods for Support Vector Machines: Evidence and Predictive Class Probabilities". In: *Machine Learning* 46 (2002), pp. 21–52. DOI: 10.1023/A:1012489924661.

[124] Amanda Spink et al. "Multitasking during Web search sessions". In: *Information Processing & Management* 42.1 (2006), pp. 264–275. DOI: 10.1016/j.ipm.2004.10.004.

[125] Mervyn Stone. "Cross-Validatory Choice and Assessment of Statistical Predictions". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 36.2 (1974), pp. 111–133. DOI: 10.1111/j.2517-6161.1974.tb00994.x.

[126] Eric A. Suess and Bruce E. Trumbo. *Introduction to Probability Simulation and Gibbs Sampling with R*. Springer, 2010. ISBN: 978-0-387-40273-4.

[127] Svenja Sydor et al. "Discovering Biomarkers for Non-Alcoholic Steatohepatitis Patients with and without Hepatocellular Carcinoma Using Fecal Metaproteomics". In: *International Journal of Molecular Sciences* 23.16 (2022). DOI: 10.3390/ijms23168841.

[128] Vu Tuan Tran and Norbert Fuhr. "Using eye-tracking with dynamic areas of interest for analyzing interactive information retrieval". In: *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. 2012, pp. 1165–1166. DOI: 10.1145/2348283.2348521.

[129] Howard Turtle and W. Bruce Croft. "Evaluation of an Inference Network-Based Retrieval Model". In: *ACM Transactions on Information Systems* 9.3 (1991), pp. 187–222. DOI: 10.1145/125187.125188.

[130] William R. Uttal and Pamela Smith. "Recognition of alphabetic characters during voluntary eye movements". In: *Perception & Psychophysics* 3 (1968), pp. 257–264. DOI: 10.3758/BF03212741.

[131] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Second. Springer, 1999. ISBN: 978-0-387-98780-4.

[132] Andrew Viterbi. "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm". In: *IEEE Transactions on Information Theory* 13.2 (1967), pp. 260–269. DOI: 10.1109/TIT.1967.1054010.

[133] Charles W. White. "Visual masking during pursuit eye movements." In: *Journal of Experimental Psychology: Human Perception and Performance* 2.4 (1976), pp. 469–478. DOI: 10.1037//0096-1523.2.4.469.

[134] Ryen W White and Resa A Roth. *Exploratory search: Beyond the query-response paradigm*. Vol. 1. 1. 2009, pp. 1–98. DOI: 10.2200/S00174ED1V01Y200901ICR003.

[135] Barbara M. Wildemuth and Luanne Freund. "Assigning Search Tasks Designed to Elicit Exploratory Search Behaviors". In: *Proceedings of the Symposium on Human-Computer Interaction and Information Retrieval*. 2012, pp. 1–10. DOI: 10.1145/2391224.2391228.

[136] Thomas D. Wilson. "Models in information behaviour research". In: *Journal of Documentation* 55.3 (1999), pp. 249–270. DOI: 10.1108/EUM0000000007145.

[137] Wan-Ching Wu, Diane Kelly, and Kun Huang. "User evaluation of query quality". In: *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. 2012, pp. 215–224. DOI: 10.1145/2348283.2348315.

[138] Clement T. Yu and Gerard Salton. "Precision Weighting—An Effective Automatic Indexing Method". In: *Journal of the ACM* 23.1 (1976), pp. 76–88. DOI: 10.1145/321921.321930.

[139] Zhen Yue, Shuguang Han, and Daqing He. "Modeling search processes using hidden states in collaborative exploratory web search". In: *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*. 2014, pp. 820–830. DOI: 10.1145/2531602.2531658.

[140] Ji Zhu and Trevor Hastie. "Kernel Logistic Regression and the Import Vector Machine". In: *Journal of Computational and Graphical Statistics* 14.1 (2005), pp. 185–205. DOI: 10.1198/106186005X25619.