Department of Computer Science

Machine Learning Methods for Interactive Search Interfaces and Cognitive Models

Antti Kangasrääsiö

Aalto University

DOCTORAL DISSERTATIONS

Aalto University publication series
DOCTORAL DISSERTATIONS 206/2018

Machine Learning Methods for Interactive Search Interfaces and Cognitive Models

Antti Kangasrääsiö

A doctoral dissertation completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Science, at a public examination held at the lecture hall M1 of the school on 16 November 2018 at 12:00.

Aalto University
School of Science
Department of Computer Science
Probabilistic Machine Learning

Supervising professor
Professor Samuel Kaski, Aalto University, Finland

Preliminary examiners
Doctor Marc Deisenroth, Imperial College London, UK
Doctor Otto Lappi, University of Helsinki, Finland

Opponent
Professor Kristian Kersting, TU Darmstadt, Germany

Aalto University publication series
DOCTORAL DISSERTATIONS 206/2018

ISBN 978-952-60-8250-9 (printed)
ISBN 978-952-60-8251-6 (pdf)
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)
http://urn.fi/URN:ISBN:978-952-60-8251-6

Unigrafia Oy
Helsinki 2018

Finland

NORDIC ECOLABEL
441 697
Printed matter

Aalto University
Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

Abstract

Author
Antti Kangasräasiö

Name of the doctoral dissertation
Machine Learning Methods for Interactive Search Interfaces and Cognitive Models

Publisher School of Science

Unit Department of Computer Science

Series Aalto University publication series DOCTORAL DISSERTATIONS 206/2018

Field of research Machine Learning

Manuscript submitted 3 January 2018 **Date of the defence** 16 November 2018

Permission to publish granted (date) 15 March 2018 **Language** English

☐ **Monograph** ☒ **Article dissertation** ☐ **Essay dissertation**

Abstract

Computer systems that users interact with are becoming more and more driven by artificial intelligence and machine learning components. This means that the ability of the users to efficiently interact with these intelligent systems on one hand, and the ability of these intelligent systems to understand the users on the other hand, are becoming more and more important for productive human-computer interaction. This thesis proposes new methods to improve both of these aspects.

The first contribution of this thesis is to improve the ability of the users to predict the consequences of their actions, and to observe possible inconsistencies in the feedback they give, when interacting with an information retrieval system that performs interactive user modelling. The proposed solutions for improving predictability are interactive visualization of the consequences of user actions and changing the behavior of the user model to better match user expectations. The proposed solutions for detecting inconsistencies in user feedback are visualization of past user feedback and interactive modelling of the accuracy of the feedback. Experiments demonstrate that the proposed methods improve user satisfaction and the usability of the search system.

The second contribution is to develop generally applicable methods for inferring the parameter values for various types of models of the user's cognition. The inherent difficulty in estimating these parameter values is caused by the complicated relation between the parameters of these cognitive models and the observation data: the likelihood function. The proposed solution is to use likelihood-free Bayesian inference, which is applicable for various different cognitive models and also able to quantify the uncertainty of the parameter estimates. Experiments demonstrate that the proposed solution enables efficient inference of cognitive model parameter values in multiple settings, and also allows informative quantification of parameter uncertainty across the parameter space.

Keywords Bayesian inference, cognitive modelling, human-computer interaction, interactive machine learning, likelihood-free inference, probabilistic modelling, user modelling

ISBN (printed) 978-952-60-8250-9 **ISBN (pdf)** 978-952-60-8251-6

ISSN (printed) 1799-4934 **ISSN (pdf)** 1799-4942

Location of publisher Helsinki **Location of printing** Helsinki **Year** 2018

Pages 205 **urn** http://urn.fi/URN:ISBN:978-952-60-8251-6

Aalto-yliopisto

Tiivistelmä

Aalto-yliopisto, PL 11000, 00076 Aalto www.aalto.fi

Tekijä
Antti Kangasrääsiö

Väitöskirjan nimi
Koneoppimismenetelmiä interaktiivisia hakukäyttöliittymiä ja kognitiivisia malleja varten

Julkaisija Perustieteiden korkeakoulu

Yksikkö Tietotekniikan laitos

Sarja Aalto University publication series DOCTORAL DISSERTATIONS 206/2018

Tutkimusala Koneoppiminen

Käsikirjoituksen pvm 03.01.2018 **Väitöspäivä** 16.11.2018

Julkaisuluvan myöntämispäivä 15.03.2018 **Kieli** Englanti

☐ **Monografia** ☒ **Artikkeliväitöskirja** ☐ **Esseeväitöskirja**

Tiivistelmä
Interaktiiviset tietokonejärjestelmät sisältävät enenevissä määrin tekoäly- ja koneoppimiskomponentteja. Näiden järjestelmien hyödyllisyyden kannalta on tärkeä kiinnittää huomiota toisaalta niiden käytettävyyteen, ja toisaalta siihen, että ne kykenevät mallintamaan käyttäjiensä tarpeita riittävällä tarkkuudella. Tämä väitöstutkimus esittää keinoja kummankin päämäärän edistämiseksi.

Tutkimuksen ensimmäisessä osiossa esitetään menetelmiä, joilla parannetaan interaktiivisen hakukoneen käytettävyyttä lisäämällä interaktion ennustettavuutta sekä käyttäjän epäjohdonmukaisen palautteen tunnistamista. Ennustettavuutta parannettiin sekä interaktiivisesti visualisoimalla käyttäjän toimintojen mahdollisia vaikutuksia että muuttamalla käyttäjämallin dynamiikkaa vastaamaan paremmin käyttäjän odotuksia. Epäjohdonmukaisen palautteen määrää vähennettiin sekä visualisoimalla käyttäjän antamaa palautetta kokonaisuutena että mallintamalla interaktiivisesti käyttäjän antaman palautteen tarkkuutta. Kokeellisesti näytettiin, että esitetyt menetelmät parantavat käyttäjätyytyväisyyttä sekä hakujärjestelmän käytettävyyttä.

Tutkimuksen toisessa osiossa kehitettiin yleisesti sovellettavia menetelmiä kognitiivisten mallien parametriarvojen päättelemiseksi. Näiden mallien parametriarvojen päättely havaintoaineiston perusteella on yleisesti ottaen hankalaa johtuen monimutkaisesta uskottavuusfunktiosta, joka sitoo yhteen havaintoaineiston sekä mallin parametriarvot. Ratkaisuksi esitettiin uskottavuusfunktioton Bayesiläinen päättely, jonka avulla kognitiivisten mallien parametriarvot, sekä niiden epävarmuus, on mahdollista päätellä. Kokeellisesti näytettiin, että menetelmä soveltuu useille erilaisille kognitiivisille malleille, ja että se mahdollistaa parametrien epävarmuuden määrittämisen koko parametriavaruudessa.

Avainsanat Bayesiläinen tilastotiede, ihmisen ja tietokoneen vuorovaikutus, interaktiivinen koneoppiminen, kognitiivinen mallinnus, käyttäjämallinnus, todennäköisyysmallinnus, uskottavuusfunktioton päättely

ISBN (painettu) 978-952-60-8250-9 **ISBN (pdf)** 978-952-60-8251-6

ISSN (painettu) 1799-4934 **ISSN (pdf)** 1799-4942

Julkaisupaikka Helsinki **Painopaikka** Helsinki **Vuosi** 2018

Sivumäärä 205 **urn** http://urn.fi/URN:ISBN:978-952-60-8251-6

Preface

Although the thesis only has one author, there are many other people and organizations, without whom, this thesis would not exist. First and foremost, my warmest thanks go to Samuel Kaski, who was both the supervisor and instructor for this thesis, and a co-author in all of the papers. Thank you for providing me freedom to explore, guidance in scientific thinking and writing, the contact network for finding great co-authors, and for your contributions to the articles. I also thank all my other co-authors: Dorota Głowacka, Yi Chen, Kumaripaba Athukorala, Andrew Howes, Jukka Corander, Antti Oulasvirta, Jussi Jokinen, Jarno Lintusaari, Henri Vuollekoski, Kusti Skytén, Marko Järvenpää and Michael Gutmann. Thank you for your help and contributions to the articles, and for being able to learn from you good scientific practices and ways of working.

I am grateful for having received funding for the thesis from multiple sources. First, I thank the the Helsinki Doctoral Education Network in Information and Communications Technology (HICT) for providing me four-year full-time funding for my thesis, which enabled me to focus fully on my research during these years. Second, I thank the Re:Know research project, the MindSee research project and the Finnish Centre of Excellence in Computational Inference Research (COIN) for funding various international conference visits that helped me both learn and network. Third, I thank the European Institute of Innovation & Technology (EIT) Doctoral School for providing me excellent entrepreneurial education and for funding a two month research visit to Kyoto University.

I also had the great opportunity to visit different research groups abroad during the past years. My warmest thanks go to Hiroshi Mamitsuka for being my host professor in Kyoto University during my two month visit, and to all the members of his research group for the interesting conversations we had during the visit. I have also made multiple fascinating one-day visits to various research groups. Thank you for hosting me and for taking the time to discuss with me about my research, Adam Jatowt (Kyoto University), Hisashi Kashima (Kyoto University), Noriko Kando (National Institute of Informatics), Christo du Plessis (University of Tokyo), Julian McAuley (UCSD), Ciro Donalek (Caltech), John O'Donovan (UCSB), and Naoki Abe (IBM Research Yorktown).

I thank all the people in the Probabilistic Machine Learning (PML) research group at Aalto for interesting discussions over lunch and coffee, and for energetic table-fussball games. Thank you especially Homayun Afrabandpey, Paul Blomstedt, Pedram Daee, Ritabrata Dutta, Manuel Eugster, Jussi Gillberg, Eemeli Leppäaho, Pekka Marttinen, Pekka Parviainen, Jaakko Peltonen, Iiris Sundin, Marta Soare, Jonathan Strahl, and Aki Vehtari. I also want to thank all the people I met from University of Helsinki for interesting discussions, especially Oswald Barral, Giulio Jacucci, Tuomas Ruotsalo, Baris Serim, and Mats Sjöberg.

I warmly thank all my friends and family for their love and support during the past years. Thank you Elli for being my partner during all these years. Thank you, my parents Juha and Paula, and all my grandparents, for your continued support. Also, thanks to my doctoral student peer support group: Olli, Eero, Hannu and Iiris. Thanks also to Reino Keränen for being my mentor in the Aalto mentoring programme for doctoral students; discussions with you were very insightful.

Finally, thank you Samuel Kaski, Dorota Głowacka, Juha Kangasrääsiö, Iiris Sundin, Eero Laukkanen and Yi Chen for proofreading this thesis and for your helpful comments to improve its readability.

Otaniemi, October 4, 2018,

Antti Kangasrääsiö

Contents

List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.

I Antti Kangasrääsiö, Dorota Głowacka, Samuel Kaski. Improving Controllability and Predictability of Interactive Recommendation Interfaces for Exploratory Search. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*, pp. 247–251, 2015.

II Antti Kangasrääsiö, Yi Chen, Dorota Głowacka, Samuel Kaski. Interactive Modeling of Concept Drift and Errors in Relevance Feedback. In *Proceedings of the 24th Conference on User Modeling, Adaptation and Personalization*, pages 185–193, 2016.

III Antti Kangasrääsiö, Kumaripaba Athukorala, Andrew Howes, Jukka Corander, Samuel Kaski, Antti Oulasvirta. Inferring Cognitive Models from Data using Approximate Bayesian Computation. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 1295–1306, 2017.

IV Antti Kangasrääsiö, Samuel Kaski. Inverse Reinforcement Learning from Summary Data. *Machine Learning*, volume 107, pages 1517–1535, 2018.

V Antti Kangasrääsiö, Jussi Jokinen, Antti Oulasvirta, Andrew Howes, Samuel Kaski. Posterior Estimation for Cognitive Models using Approximate Bayesian Computation. Submitted to *a journal*, 2018.

VI Jarno Lintusaari, Henri Vuollekoski, Antti Kangasrääsiö, Kusti Skytén, Marko Järvenpää, Michael Gutmann, Aki Vehtari, Jukka Corander, Samuel Kaski. ELFI: Engine for Likelihood Free Inference. *Journal of Machine Learning Research*, volume 19, paper 16, pages 1–7, 2018.

Author's Contribution

Publication I: "Improving Controllability and Predictability of Interactive Recommendation Interfaces for Exploratory Search"

The author designed the new methods together with the co-authors. The author implemented the new methods. The author designed, executed and analyzed the user experiment. The author wrote the majority of the manuscript.

Publication II: "Interactive Modeling of Concept Drift and Errors in Relevance Feedback"

The author designed the new modelling-related methods. The author implemented the modelling-related methods and participated in implementing the user interface related methods as well. The author designed, executed and analyzed the simulation experiment. The author participated in designing and analyzing the user experiment. The author wrote the majority of the manuscript.

Publication III: "Inferring Cognitive Models from Data using Approximate Bayesian Computation"

The author designed the new methods together with the co-authors. The author implemented the new methods. The author designed the simulation experiment together with co-authors. The author executed and analyzed the simulation experiment. The author wrote the majority of the manuscript.

Publication IV: "Inverse Reinforcement Learning from Summary Data"

The author designed and implemented the new methods together with the co-author. The author designed the simulation experiment together with the co-author. The author executed and analyzed the simulation experiment. The author wrote the majority of the manuscript.

Publication V: "Posterior Estimation for Cognitive Models using Approximate Bayesian Computation"

The author designed the new methods together with the co-authors. The author implemented the new methods. The author designed the simulation experiments together with the co-authors. The author executed and analyzed the simulation experiments. The author wrote the majority of the manuscript.

Publication VI: "ELFI: Engine for Likelihood Free Inference"

The author designed and implemented an initial draft version of the software library, used in Publication III. The author participated in designing and implementing the final version of the library, used in Publications IV and V, together with the co-authors. Regarding the final version, contributions of the author relate mostly to the design and implementation of Bayesian optimization, Gaussian processes, experiment logging and automated testing. The author made comments to the manuscript in preparation.

1. Introduction

1.1 Motivation

One of the overarching goals of human-computer interaction (HCI) research is to improve the ability of users and computer systems to interact productively. One enabling technology in this respect is *machine learning*, or automated learning algorithms. Through improving the ability of computer systems to learn the interests and capabilities of the user – by being able to construct adequate *user models* – it has been possible to construct systems with improved capabilities for productive interaction [Fischer, 2001].

This thesis focuses on two specific issues related to the use of machine learning in user modelling. The first issue relates to the ability of users to interact efficiently with a computer system whose behavior is largely determined by a user model. The second issue relates to the ability of computer systems to learn even more accurate models of the interests and capabilities of the user based on commonly available observation data.

Regarding the first issue, the behavior of an interactive system is ultimately driven by the instructions given by the user. In order to give purposeful instructions, the user employs a *mental model* of the system for predicting the consequences of her actions; however, the user only has limited cognitive capability for this kind of mental simulation [Helander et al., 1997]. With systems whose behavior is largely determined by a user model, it may be difficult for the user to predict the consequences of her actions; this is because the user model may cause the relationship between user actions and consequences to be complicated [Tsandilas and Schraefel, 2004]. As a consequence, the user might make interaction errors, which may increase the amount of mistaken predictions made by the user model, which may then erode the user's trust to the system [Muir, 1994]. Overall, these issues may lead to less than optimal performance when performing a task with the system [Norman, 1983], reduced use of the system [Dzindolet et al., 2003] and lower user satisfaction and confidence in using the system [Kulesza et al., 2012]. Thus, it is important to provide the user adequate capabilities for comprehending how her actions affect the user model, and the behavior of the system as a whole. This should both facilitate better mental models and reduce the need for mental simulation. It is also important to provide the user adequate capabilities for noticing possible interaction errors. This should facilitate fixing those errors and also provide explanations for odd system behavior.

The second issue is that the models currently employed for user modelling are still far from perfect, and they are often based on multiple simplifying assumptions. A common example of such simplification is to treat the user simply as a "static data distribution" [e.g. Marlin, 2004] without explicit cognitive capabilities such as learning or memory. More sophisticated user models, which do explicitly take such aspects of human cognition into account, are called *cognitive models*. Examples of such modelling frameworks include ACT-R (Adaptive Control of Thought – Rational) [Anderson, 1996] and computational rationality [Lewis et al., 2014; Gershman et al., 2015]; the latter are often formulated using the *reinforcement learning* formalism [Sutton and Barto, 1998]. The use of cognitive models for user modelling should allow computer systems to make more accurate predictions of user behavior and cognitive state, and thus facilitate more efficient interaction. However, existing methods for learning such models either have strict requirements for both the type of observations and underlying model structure [e.g. Ramachandran and Amir, 2007; Ziebart et al., 2008], are not efficient (e.g. brute-force optimization), or are not able to evaluate the uncertainty of the parameter estimates [e.g. Lagarias et al., 1998]. These limitations reduce the applicability of cognitive models, which may lead to simpler and possibly less accurate models to be used instead. Thus, it is important to advance our ability to estimate the parameter values of cognitive models from observation data. Optimally, efficient inference should be possible based on various different types of observations, without strict limitations for the cognitive model structure, and even from small amounts of observation data. Furthermore, as the observation data for learning these models may often be incomplete, it is important to develop methods that are able to estimate the parameter uncertainty in a principled manner.

1.2 Research Questions and Contributions

This thesis focuses on two specific research questions derived from the general motivating factors outlined above.

RQ1 – *Can the satisfaction and task performance of search engine users be improved through improving the predictability of the interaction with the search engine and through improving the capability of users to notice and correct possible interaction errors?*

Publications I and II provide contributions to RQ1. In both of these publications, improvements are proposed to an existing search engine, where the user is able to both visualize the user model and interact with it.

The effect of improved predictability when interacting with a user model was studied in Publication I. The publication proposes two complementary methods for improving the predictability of user interaction. The first method is to forward-simulate the dynamics of the user model with various possible feedback options available to the user, and visualize these predictions to the user. This way the user is able to better predict the consequences of her actions before committing to them. The second method is to explicitly optimize the effects of user interaction to be as predictable as possible. This way the dynamics of the user model are altered to improve the predictability of user actions. User experiments indicated that these new methods improved user satisfaction and ability to understand model behavior.

The effect of improved ability to notice and correct possible interaction errors was studied in Publication II. The publication proposes both a method for modelling errors in user feedback, and a user interface for visualizing these estimated errors and enabling corrections to existing feedback. The model uses a method called automatic relevance determination for estimating the reliability of each feedback item when considering the entire feedback data set as a whole. The user interface highlights feedback with low estimated reliability and allows the user to make various modifications to existing feedback. Simulation experiments indicate that the approach leads to improved information retrieval quality. User experiments indicate improved usability of the system and improved ability of users to notice and correct interaction errors.

RQ2 – *Can the parameter values of cognitive models be learned efficiently from observation data, together with principled uncertainty estimates, without strict requirements for the type of the observations or for the structure of the cognitive model?*

Publications III, IV and V provide contributions to RQ2. The main issue with inferring the parameter values of cognitive models is with the *likelihood function* of the model, which explicates the relation between model parameters and predictions. Especially if the model structure is complicated, or the type of observations is not convenient, evaluating the likelihood function is difficult. The proposed solution is to perform likelihood-free inference, which allows considerable flexibility regarding both the type of observation data and the underlying cognitive model. The chosen approach for likelihood-free inference is approximate Bayesian computation (ABC) combined with Bayesian optimization (BO), which allows both principled estimation of parameter uncertainty, due to ABC, and efficient exploration of the parameter space, due to BO. Publication III presents an initial proof-of-concept where point estimates are produced for the parameters of a reinforcement learning (RL) model based on aggregate observation data. Publication IV provides further technical justification for the use of likelihood-free inference for RL models when the observations are not in the traditional format of state-action trajectories. Publication V provides further case studies, which demonstrate that the method can be applied for the ACT-R cognitive model family as well. These studies show that the proposed method is applicable for cognitive model parameter inference, being flexible both regarding the model structure and type of observations. The method also provides a good trade-off between efficiency and interpretability of results.

Software development related to Publications III, IV and V has also contributed to the open-source likelihood-free inference framework ELFI (Engine for Likelihood-Free Inference), presented in Publication VI.

1.3 Organization of the Thesis

The second chapter gives a brief overview of relevant concepts in quantitative modelling of human behavior, which constitutes the common theoretical background. The third chapter gives further motivation and background to interactive user modelling and discusses the contributions related to RQ1. The fourth chapter gives further motivation and background to cognitive models and discusses the contributions related to RQ2. The fifth chapter discusses the answers to the research questions and directions for further research.

2. Brief Background on User Modelling

User modelling, or fitting the parameters of quantitative models to observation data collected from users, has multiple motivations. These models may be used, for example, for estimating psychometric properties of the user, such as the level of experience or personality traits [e.g. Aykin and Aykin, 1991]. User models may also be used for estimating the interests or preferences of the user, which may then be used, for example, to personalize the system to the particular user or to perform a task according to these preferences [e.g. Anand and Mobasher, 2003]; one example of this is in information retrieval, where the displayed information is filtered according to the search user model.

This chapter gives a brief theoretical outline of the modelling framework used in this thesis. First, two common user modelling contexts are described. Second, an outline of Bayesian statistical modelling is given. Third, evaluation of the quality of user models is discussed.

2.1 Modelling Contexts in Human-Computer Interaction

In human-computer interaction (HCI), models can be constructed either offline or online. Offline modelling is performed based on previously collected data, and may be done for various purposes, while online modelling is performed based on continuously arriving data, and is generally done for the purpose of helping the user perform a task.

2.1.1 Offline User Modelling

The setting of offline user modelling is defined by two common features. First, the observation data used for fitting the model are collected before the modelling takes place. Second, if further data are collected, the results of the modelling process do not affect this data collection process. This makes the process conceptually simple, as illustrated by Figure 2.1. There are just two consecutive steps in the modelling process; first, the observation data is collected, and second, the model is fit to this data.

Offline user modelling is commonly used, for example, in scientific modelling, where the purpose of the model is to discover new insights about human behavior [e.g. Fitts and Peterson, 1964]. Another example is model-based user interface (UI) design, where the model is used for designing the UI layout [e.g. Wilson et al., 1993].

Figure 2.1. Information flow chart outlining the key aspects of the offline user modelling situation. The actions performed by the user are the source of the observation data. The inference process uses these observations to estimate the user model parameters. The learned user model is then used for some purpose, such as scientific research or system design.

2.1.2 Online User Modelling in a Task Context

The setting of online user modelling is somewhat more complex than that of offline modelling. In online user modelling, the model is constructed at the same time as the data collection takes place, and the results of the modelling process generally affect the data generating process as well. In HCI, online modelling is commonly performed for the purpose of helping the user perform a task. The situation is illustrated in Figure 2.2. Essentially, the situation is analogous to closed-loop control, where the user acts as the controller, and the AI system and task together form the system under control [cf. Doyle et al., 2013].

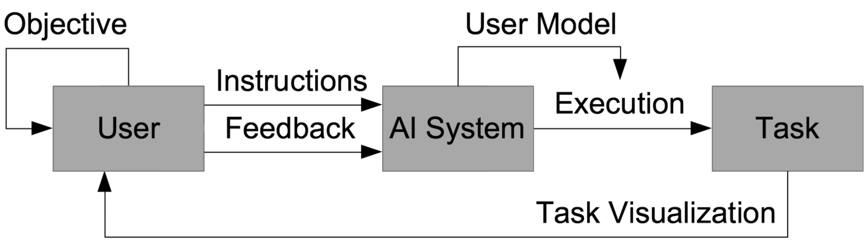

Figure 2.2. Information flow chart outlining the key aspects of the online user modelling situation in a task context. The user has an objective in mind, which corresponds to a specific task state. She uses an interactive AI system to perform the task. The interaction with the system consists of instructions, which directly affect the task execution, and feedback which affects the task execution through the user model. The user is able to visualize the current state of the task, and compare it to the current objective.

Information retrieval is a concrete example of online user modelling in a task context [Baeza-Yates and Ribeiro-Neto, 1999]. The objective of the user is to fulfill an information need. The AI system is a search engine, and the task is to retrieve a set of documents from a large collection. The feedback given by the user are the search keywords. The instructions are to find documents matching the keywords or to display the next page of results. The user model is used to predict what documents match the search keywords typed by the user.

Task execution consists of retrieving documents that the user model predicts are relevant, with possible post-processing, such as diversification of the results. The task visualization consists, for example, of a result page, displaying a set of relevant documents. After viewing the results, the user generally adjusts the feedback, or gives new instructions, until the objective is reached.

2.2 Bayesian Modelling

Introduction

Bayesian statistics is a formalism for quantifying the subjective uncertainty an observer has of the state of the world. This formalism is especially attractive when the amount of observation data is relatively small, but is compensated by a reasonable amount of prior knowledge of the *generative process* which is postulated to have caused the observations.

Incidentally, this is often the situation when modelling, for example, individual users. For example, in online user modelling, the models are often made of individual users, based on relatively small amount of observations made after the task started [e.g. Głowacka et al., 2013].

Bayesian Statistical Concepts

The *probability distribution* is probably the most fundamental concept in Bayesian statistics. A probability distribution P is a density function that precisely defines the behavior of a random variable X. The domain of the function are the possible realizations of X, denoted by x, and the range is all non-negative real numbers. To indicate that the realizations x are assumed to be drawn independently according to the distribution P the notation $x \sim P(x)$ is used.

The *generative model* is another core concept. A generative model M is a procedure which is postulated to have generated the observation data D. In the case of user modelling, M is often called the user model and D is the observation data available of the user. Generally, M is a procedure that is defined in terms of unknown model parameters θ, and defines a probability distribution over the space of possible observations. Thus the modelling assumption is $D \sim P(D|\theta; M)$, although M is generally left out of notation for brevity when only one model is considered.

The core idea in the Bayesian statistical formalism is that the knowledge an observer has about the true values of θ can be precisely quantified by a *posterior* $P(\theta|D)$, which can be computed based on Bayes' formula

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)},$$

where $P(D|\theta)$ is the *observation likelihood* (often called the *likelihood function*), $P(\theta)$ the *prior*, and $P(D)$ the *marginal likelihood*. The prior answers the question "Before observing the data D, how much more probable is it from the observer's perspective that the true parameters are θ compared to θ'?" The observation likelihood answers the question "Given that the model parameter values are θ, how likely it is that data D is observed?", while the marginal likelihood answers the question "How likely it is that data D is observed in general?" The posterior thus answers the question "Given both prior knowledge and evidence presented by the data D, how much more probable is it from the observer's perspective that the true parameters are θ compared to θ'?" The process of computing the posterior based on observation data is called *inference*, or more informally model fitting or "learning a model".

2.3 Evaluation of Model Quality

Evaluation of offline user models is often straight-forward, as standard measures are applicable, such as prediction accuracy or likelihood of observation data under model assumptions [cf. Friedman et al., 2001, ch. 7].

However, evaluating the quality of online user models generally differs from other quantitative models, for two main reasons. The first reason is that the main design goal is to help the user complete the task, not merely to make accurate predictions. The second reason is that the dynamic behavior of the system can often be reliably measured only with real users in a realistic task setting. For these reasons, *user studies* are generally required.

Occasionally, user studies are replaced by proxy measures computed on previously collected data, but the performance of these proxies may not always be reliable [Grotov et al., 2015]. The experiments may also be conducted using simulated users [Ritter and Young, 2001], but in general, it is challenging to capture the whole range of human behavior with a simulator.

One key benefit of user studies, compared to using some kind of a proxy measure or simulator, is that they take into account all of the possible side-effects that might occur. In general, human behavior depends on multiple factors in non-linear ways, which can make extrapolation based on old data unreliable. This is especially important if the model is used in an interactive setting, where changes to the model affect the data collection process. For example, if an information retrieval model is changed, this changes the search results that the user is shown, which then affects the subsequent actions the user will perform during the search session. Thus, data from old search sessions might not be a good proxy for measuring the performance of a new search user model. Another benefit of user studies is that they allow collection of various types of observation data, including usage logs, questionnaire answers and interview recordings. This rich set of both quantitative and qualitative observation data allows the effects of new methods to be evaluated in a thorough manner, based on multiple complementary data sources. Furthermore, qualitative data collection methods, such as interviews, allow the users to express their opinions freely, which might help in forming new hypotheses that did not exist when the experiment was designed.

User studies also have certain drawbacks. First, the experiments are often expensive to perform, often requiring multiple hours of researcher time per individual user. Second, the experiments often require naïve users, which means that users who have taken place in previous experiments with a similar system may not be recruited for new studies. These two drawbacks generally result in user studies with relatively small number of participants, both due to limited resources and the finite pool of easily available experiment subjects. Third, individual users often exhibit large variation in their performance. This may be due to individual differences, as the users may come from a rich set of backgrounds, and the performance of the users may also vary over time. This variability generally results in low signal-to-noise ratio, which makes it more difficult to argue for statistical significance unless there are sufficiently many subjects or the measured effect is notably large. Thus, in general, design of user studies is a trade-off between the cost of the experiments and the ability to detect small effects.

3. New Methods for Interactive User Modelling in Information Retrieval

This chapter gives an overview of the line of research related to RQ1 – *Can the satisfaction and task performance of search engine users be improved through improving the predictability of the interaction with the search engine and through improving the capability of users to notice and correct possible interaction errors?* First, the concept of *interactive user modelling* is defined. Then, the SciNet search engine [Głowacka et al., 2013; Ruotsalo et al., 2013], which is used as the research platform in this thesis, is described. Next, motivating factors for the use of interactive user modelling are briefly discussed. Finally, two new approaches to improve interactive user modelling are introduced, providing improved interaction predictability and validation of the consistency of user feedback.

3.1 Interactive User Modelling in a Task Context

Interactive user modelling is a special case of online user modelling where the user is additionally able to explicitly visualize and interact with the user model. The name interactive user modelling is derived from the fact that the user is interactively participating in the adjustment of the user model, instead of being just a passive subject of the modelling process. The term *viewable user model* [Cook and Kay, 1994] has also been used when the main purpose has been to give the user the ability to inspect the model, while the ability to adjust the model has not been equally important. The term *interactive machine learning* [Fails and Olsen Jr, 2003] has also been used for the more general setting, when the model is not necessarily a user model.

In this thesis, interactive user modelling is studied in a task context, as visualized in Figure 3.1. In comparison to the standard setting of online user modelling in a task context, now the user is able to visualize both the user model and the state of the task. Generally, this also means that the model interaction is an integral part of the AI system [e.g. Głowacka et al., 2013], instead of a separate mode of interaction [e.g. Cook and Kay, 1994]. Another change, compared to the standard setting of online user modelling, is that the user can additionally provide feedback related directly to the user model, such as the weights of different features [Bostandjiev et al., 2012].

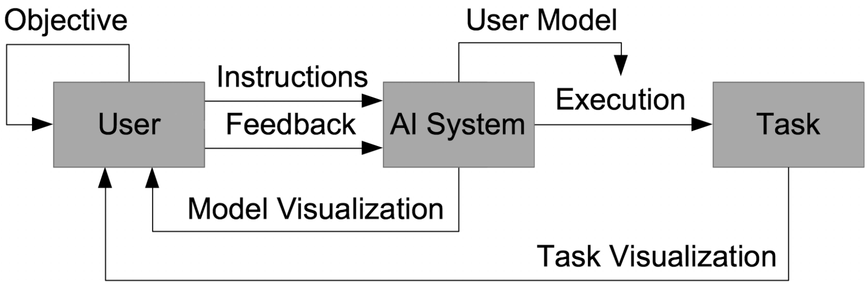

Figure 3.1. Information flow chart outlining the key aspects of the interactive user modelling situation. The setting is similar as in Figure 2.2, except that the user has the additional ability to visualize the user model.

For example, in the case of information retrieval, the visualized user model allows the user to answer the question "what does the search engine think I am interested in finding", while the visualized state of the task allows the user to answer the question "which items appear to be most relevant based on my instructions and the current user model".

3.2 Case Example: The SciNet Search Interface

One recent example that demonstrates the benefit of interactive user modelling in information retrieval is the SciNet seach interface [Głowacka et al., 2013; Ruotsalo et al., 2013].

A central motivation for using interactive user modelling in information retrieval is the fact that when retrieving information, users often prefer to use small local steps to navigate towards the objective [Teevan et al., 2004]. However, it might be challenging for the user to make these small steps when they have to be made through manually altering a textual keyword query.

To address this issue, the SciNet search interface is divided into two parts, illustrated in Figure 3.2. On the left side of the user interface is an interactive visualization of the user model, while on the right side of the user interface is a visualization of the search results. Through the radar visualization, the interface allows the user to both visualize a part of the user model and give relevance feedback to keywords by moving them to new locations on the radar. The radar displays a small number of the most relevant keywords as predicted by the user model. Keywords that are close to the center of the radar are more relevant than those that appear closer to the edge. Moving a keyword closer to center indicates high relevance and vice versa. Thus, through keyword adjustments, the user has an intuitive way to make small local adjustments to the search query.

The use of interactive user modelling leads to multiple improvements over a system without the interactive visualization of the user model [Ruotsalo

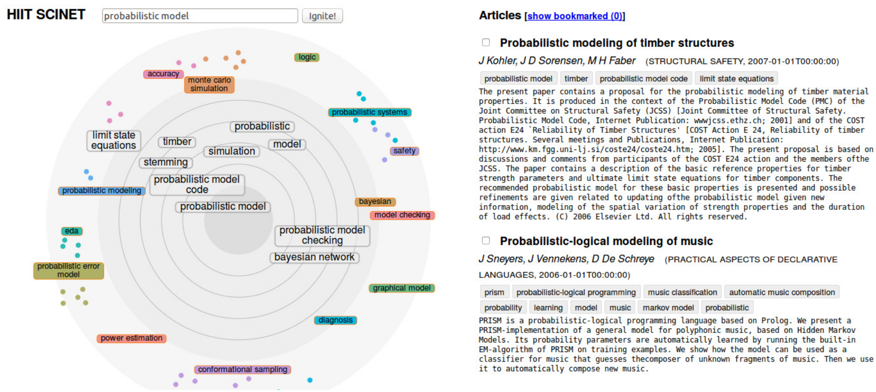

Figure 3.2. The SciNet search engine interface. **Left:** At the top left of the interface there is a text field for inputting direct keyword queries. Below the text field is a large radar visualization of the search user model. At the center of the radar, the 10 most relevant keywords are visualized using gray boxes. The location of the box illustrates both the relevance of the keyword and its similarity to other visualized keywords. The closer to the center of the radar the box is, the more relevant the keyword. The closer the keyword is to other keywords on the radar, on average the more similar these keywords are. At the edge of the radar are numerous keyword suggestions, visualized using colored boxes. The keyword suggestions are clustered based on similarity, and each cluster is denoted by a color that is distinct from the neighboring clusters. The user is able to give feedback to any of the visualized keywords through dragging them to a new place on the radar. **Right:** A scrollable list of the most relevant scientific articles retrieved by the search engine is on the right side of the interface. For each article, various pieces of information are displayed, including the title, authors, publication venue, publication time, related keywords and the abstract. The articles can also be added to a bookmark list, by clicking at a checkbox next to the article title. The user can access the list of bookmarked articles through a link at the top of the result list.

et al., 2013]. The main result by Ruotsalo et al. [2013] was that the interactive visualization improves users' task performance, which could be attributed to both the improved quality of information available to the user and to the improved interaction capabilities offered to the user.

3.3 Motivation for Interactive User Modelling

In general, there are two main motivating factors for adding model visualization to on-line modelling situations: improved trust and improved efficiency. A brief discussion on possible pitfalls is also given.

3.3.1 Improved Trust

Trust in an AI system has a significant influence on how the user actually uses it; poor trust may cause the user to abandon the system, even though using it would lead to superior results [Dzindolet et al., 2003]. Trust is built upon multiple factors, including the user's ability to predict the system behavior, depend on the correct functionality of the system and the user having faith in the continuing performance of the system [Muir, 1994].

With AI systems that rely on a user model, it is common for the model to make incorrect predictions. These errors are due to multiple reasons, including simplifying model assumptions, insufficient data and miscommunication between the user and the system. These errors can possibly erode the trust the user has towards the system, unless an explanation is provided as to why the error may have occurred [Dzindolet et al., 2003].

As interactive user modelling provides the user with means to inspect the user model, this will likely increase the trust in the system when poor performance can be attributed to errors in the user model. This is supported by multiple examples of model explanations leading to improved task performance and higher confidence when using the system [Suermondt and Cooper, 1993], allowing users to provide better feedback [Fiebrink et al., 2011], facilitating better understanding of system behavior and allowing users to correct mistakes in the model more efficiently [Kulesza et al., 2015].

3.3.2 Improved Efficiency

Interactive user modelling provides improved efficiency in two ways: through improving the accuracy of the mental model the user has of the system and through enabling new feedback options.

Mental Model Quality

A mental model is the user's internal explanation for the behavior of the AI system, and can be seen as an analogue of the user model that the system has of the user [Helander et al., 1997]. If the mental model the user has of the system behavior is less accurate, the user is then more likely to perform less efficiently with the system [Norman, 1983]. In comparison, having a better mental model has been shown to both facilitate learning to use a new system [Kieras and Bovair, 1984] and lead to improved satisfaction and confidence in using the system [Kulesza et al., 2012].

In the case of systems where the performance is largely defined by the quality of the user model, such as information retrieval systems, the behavior of the user model is naturally instrumental in explaining the behavior of the system. If the user does not have any ability to get further insight into the behavior of the user model, it is more difficult for the user to generate explanations for the system behavior [Kieras and Polson, 1985], which corresponds to a less accurate mental model. Thus, it is beneficial to make the model intelligible to the user, thus helping the user build a better mental model of the system behavior [Lim et al., 2009], and ultimately facilitating more adequate interaction with the system.

Adequate Feedback Options

In online user modelling, when the user is deciding on her next action, she essentially has to make a selection between all available feedback actions, trying to choose the one that most advances her objective. Thus, it is natural that systems with adequate feedback options are preferable; in general, the options should be effective, easy to understand, and suitable for the current task status [Rogers et al., 2011]. For example, in the case of information retrieval, if the user only has the ability to give feedback through modifying a keyword query, it might be challenging for the user to come up with a good query, due to the necessity of recalling the correct terms to use [Baeza-Yates et al., 2004]. In contrast, it has been demonstrated that when the user has adequate interaction options, such as the ability to adjust selected parts of the user model, both task performance and user satisfaction have improved [Bostandjiev et al., 2012; Ruotsalo et al., 2013].

3.3.3 Possible Pitfalls

There are of course also situations where interactive user modelling might not bring any additional advantages, and implementation issues may diminish the possible benefits.

First of all, the user will not use explanations provided by the system unless they are easy to access and unless there is a salient reason to use them [Gregor and Benbasat, 1999]. For this reason, any visualizations of the model should be easily available, and interacting with them should propose a clear advantage to the user.

Another possible pitfall is presented by poor design and interpretability of the visualized model [Vellido et al., 2012]. Examples of possible issues include too much or too little detail, misleading visualizations, and terminology that is unfamiliar to the end-user [Ward et al., 2010, chapter 12]. In general, a large amount of practical guidelines apply for the design of effective visualizations [Ware, 2012, appendix D].

3.4 Improving Predictability in Interactive User Modelling

This section discusses the significance of predictability in interactive user modelling, lists causes for poor predictability and proposes solutions for improving the predictability of the SciNet search user interface.

3.4.1 Significance of Predictability

A predictable system is such that the user is able to predict how the system will behave in reaction to user actions. The predictability of the system has a significant effect on building trust, especially for novice users [Muir, 1994], and also helps to improve user satisfaction [Gajos et al., 2008].

Muir [1994] lists three factors that affect the apparent predictability of a system:

1. The *actual predictability* of the system. This is defined by the fundamental limitations to its predictability, derived from both the design of the system and the physical limitations of the task environment.

2. The *capability of the user* to estimate the behavior of the system. This is primarily defined by the transparency of the system, which depends on the methods that are available for the user to observe and understand the system behavior. However, the practical experience of the user also has an effect, as more experienced users are generally more capable of comprehending the system behavior.

3. The *stability of the environment*. The behavior of many systems depends on the environment they operate in, so when the environment changes, the system behavior changes as well. Thus, if the environment the system is used in is subject to instability or unpredictability, this also has an effect on the apparent predictability of the system.

3.4.2 Causes of Poor Predictability in Interactive User Modelling

As noted by Tsandilas and Schraefel [2004], user models are one of the main sources of unpredictability with interactive AI systems. This is understandable, as the apparent behavior of many mathematical models may differ greatly from the intuition of the layman user. However, in a mathematical sense, the behavior of many user models is strictly predictable as a function of the user feedback. Thus, the actual predictability of the user model is not an issue with interactive user modelling; the problem is with the capability of the user to predict the behavior of the user model.

Regarding this issue, I see two main aspects that define the capability of the user to understand the behavior of the user model.

The first aspect is *comprehension of the model state*. A user who is able to comprehend the model state is able to answer questions such as “what does the user model predict my interests are?”, or “what are the features of the current model that have the largest effect on the predictions the model makes?”

The second aspect is *comprehension of the model dynamics*. A user who is able to comprehend the model dynamics is able to answer questions such as “how would the model predictions change if I would give this or that feedback?”, or “how would the most important features of the model change if I would give this or that feedback?”

In the context of interactive user modelling, the user is generally able to inspect the user model, which facilitates the ability of the user to comprehend the state of the model. However, just because the user is able to comprehend the state of the model, there is no guarantee that the system facilitates the ability of the user to comprehend or predict the model dynamics as well.

To the best of my knowledge, the lack of support for comprehension and prediction of model dynamics has not been explicitly discussed in previous research, and no general design principles exist for facilitating the comprehension of model dynamics in the context of interactive user modelling.

3.4.3 Proposed Solutions for Improving Predictability

To address the shortcomings outlined above, this thesis presents two complementary methods for improving the ability of the user to comprehend and predict how her feedback affects the user model in interactive user modelling: predictive visualization and intuitive control. I will first give a general explanation of the proposed methods, followed by an analysis of their possible limitations.

For explicitness, I use the following notation, which describes the standard setting of interactive user modelling in precise terms. Let the feedback of the user up to time t be $X_t = \{x_1, \ldots, x_t\}$ and let the model at time t be $M_t = M(X_t)$. The visualization presented to the user at time t is denoted $V_t = V(M_t)$ and the interaction options are $O_t = \{o_{t1}, \ldots, o_{tN}\} = O(V_t, M_t)$ (not to be confused with the O-notation for computational complexity), such that there is an input mapping I from each o_{ti} to a certain x_{t+1}.

Predictive Visualization
The key insight of predictive visualization, presented in Publication I, is that as a visualization V of the model state already exists in the interactive user modelling setting, predicted changes to the model state can be communicated to the user through this same visualization before the user commits to a feedback action.

In precise terms, when the user is making the choice between interaction options O_t, the search engine is able to construct, for each distinct option o_{ti}, a one-step-ahead prediction of the next user model state $M_{t+1,i} = M(X_t \cup \{I(o_{ti})\})$, which can then be visualized to the user as $V_{ti} = V(M_{t+1,i})$. In this way, the search engine will facilitate the ability of the user to comprehend how her feedback will affect the search user model, using the full capability offered by the visualization V.

Intuitive Control
The key insight of intuitive control, presented in Publication I, is that if the user model dynamics[1] are difficult for the user to understand, optimization can be used to change the model dynamics so that they become easier to understand from the user's perspective. This is similar to the set-up of control engineering, where a controller is used to change the dynamics of a system under control so that it becomes better behaved [cf. Doyle et al., 2013]. An analogous example is the power steering system used in cars, which keeps the effort needed to turn

[1]By *model dynamics*, I mean how the model changes in response to changes in the training data, such as the addition of one more instance of user feedback.

the steering wheel constant and predictable in various driving situations.

First, assume that for each interaction option o_{ti}, it is possible to define the most predictable consequence from the user's perspective. In precise terms, assume these are defined using a *requirement function* r. Corresponding to each o_{ti}, let r_{ti} be a mapping from future model states M_{t+1} to $\mathbb{R}_{\leq 0}$ such that values of $r_{ti}(M_{t+1})$ closer to zero indicate that M_{t+1} better satisfies the requirement r_{ti}. Thus, the interpretation of the interaction options can be changed, so that instead of mapping each o_{ti} to a feedback x, they are mapped to requirements $R_t = \{r_{t1}, \ldots, r_{tN}\}$ regarding the next state of the model M_{t+1}.

Now, instead of constructing the next feedback set X_{t+1} by simply aggregating the user's latest feedback to the old data by $X_{t+1} = X_t \cup \{I(o_{ti})\}$, the idea is to construct X_{t+1} using the solution for an optimization problem:

$$X_{t+1} = X_t \cup X'_{ti},$$

$$X'_{ti} = \underset{X'}{\operatorname{argmax}}\ r_{ti}(M(X_t \cup X')),$$

where the option chosen by the user was o_{ti} and sensible limitations are made for the search space of X' based on the application context. This approach will then guarantee that the latest feedback given by the user will cause optimally predictable consequences (as defined by R_t), regardless of the complexity of the underlying model dynamics, with the limitation that the existing feedback X_t should not be modified in the process.

The requirement that the existing feedback should not be modified by the update is justified by the fact that X_t defines the current information about the user, and thus if X'_{ti} were to be optimized by $\operatorname{argmax}_{X'}\ r_{ti}(M(X'))$, there would be no guarantee that this information would be preserved.

This way of mapping o_{ti} to feedback data X'_{ti} is also fundamentally different from having a static mapping between o_{ti} and feedback instances x. The reason for this is path-dependency; X'_{ti} is now a function of both r_{ti} and X_t, as the requirements are always interpreted in the context of all the existing feedback. This suggests that there is no trivial way to reduce this method to just a different static mapping between o_{ti} and x.

As an alternative, the requirement function could also be constructed using V instead of M. However, when there is a close correspondence between these two, the results are likely similar.

And lastly, this approach can be combined with predictive visualization simply by visualizing $M(X_t \cup X'_{ti})$ instead of $M(X_t \cup \{I(o_{ti})\})$ when the user is making a choice between the interaction options.

Limitations

One limitation with these methods is with responsiveness of the interaction. In general, humans have some tolerance for the responsiveness of an interactive system but in many cases the response times are expected to be within a few seconds [Shneiderman, 1984]. However, if updating the model, re-drawing the visualization or solving the optimization problem cannot be done in such a short time, it is difficult to provide the user with a satisfactory user experience.

There are multiple ways to address this issue, depending on the situation. If the delays are attributable to latency between the user interface and the computer which performs the model update, which is not unusual in online applications, the predictions might be computed in advance and transmitted to the user interface together with the rest of the task related data. If the optimization problem is challenging to solve precisely, an approximate solution might be used instead. If there is a large number of similar feedback options, say a continuous range of values, a smaller number of the possible options could be sampled from the full set and the predictions only computed on these options, while interpolation could be used for the rest. If re-drawing the visualization is time-consuming, only the most salient parts of the visualization could be updated on-line.

One possible pitfall with these methods is related to stochastic model updates. If the state of the model is not a deterministic function of the user feedback, but instead the state is drawn from a distribution $M_t \sim P(M_t|X_t)$, the predicted state that is visualized to the user may not correspond to the resulting state if the actual model update and the prediction are made separately. This issue could be remedied either by storing the prediction and using it directly as the resulting model, by making the model deterministic by storing the random number generator seed, or by using, for example, an expectation $E[M_t]$ as the prediction visualized to the user.

3.4.4 Implementation

Both of the proposed methods were implemented using the existing SciNet search engine developed by Głowacka et al. [2013] and Ruotsalo et al. [2013].

Predictive Visualization

Predictive visualization was implemented so that when the user starts dragging a keyword on the radar, for the purpose of giving relevance feedback to that keyword, the locations of the rest of the central keywords are changed to correspond to their predicted new relevance scores, given that the user would drop the dragged keyword to the current location. The new behavior is visualized in Figure 3.3.

Intuitive Control

Intuitive control was implemented so that moving a keyword on the radar to a certain location would optimize the resulting user model in such a way that the predicted relevance of the latest moved keyword would be as close as possible to the given feedback value.

To explain why this behavior is not guaranteed by the model by itself, consider that it is standard for a model to treat all user feedback equally, as if each of the

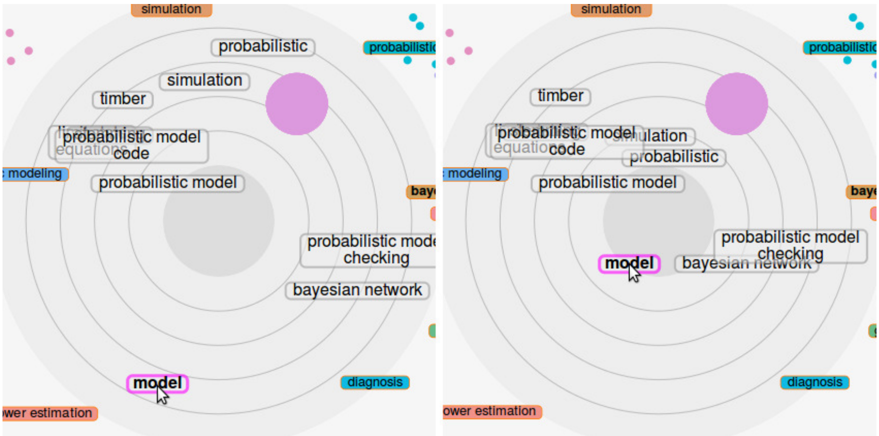

Figure 3.3. Illustration of the predictive visualization behavior. When the user starts dragging a keyword over the radar, the original location of the keyword is highlighted with a purple circle and the box of the dragged keyword is also highlighted with same color. When the user drags the keyword over different locations on the radar, the distances of other central keywords from the radar center change simultaneously according to their predicted relevance scores. The user also has the opportunity to drag the keyword back to its original location to cancel the initiated feedback action. Reprinted from Publication I by permission, © 2015 ACM.

feedback values would have been generated in an independent fashion and carry equal weight. Now, as the size of the feedback set X_t grows, it is clear that the *additional effect* of each new feedback on average diminishes in proportion to $|X_t|^{-1}$. This fundamental property, common to a wide set of user models, makes the model dynamics difficult to predict, as they are by design conditional not only on the most recent feedback, but also on all past feedback. An example of how this property can cause problems with predictability is illustrated in Figure 3.4.

3.4.5 Evaluation

A user study was conducted in Publication I to evaluate the effects of these two proposed improvements over the baseline SciNet search engine [Ruotsalo et al., 2013]. A balanced study design was used, where 12 users each performed 2 search tasks, one with each variant of the search engine. One of the tasks was a focused task that required the user to search for specific answers, while the other one was broad, and had multiple possible answers.

The performance of the systems was measured in four dimensions. The task performance was measured by an expert who scored the performance of the users. The general usability of the system was measured with the SUS (System Usability Scale) usability questionnaire [Brooke, 1996]. The user satisfaction on information retrieval performance was measured with a modified version of the ResQue recommender system performance questionnaire [Pu et al., 2011]. The modifications were related to adapting the questionnaire from recommendations

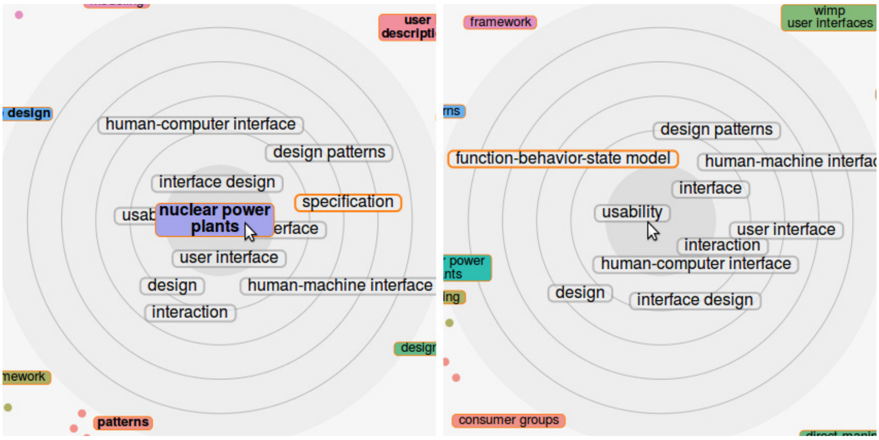

Figure 3.4. Illustration of the diminishing effect of feedback and apparent lack of control over the user model, manifesting in the baseline system. In this example, the user has previously given multiple instances of relevance feedback related to the topic “human-computer interaction”, and now wants to direct the search towards the topic “human-computer interaction in nuclear power plants”. In this situation, although the user gave maximum relevance feedback to the keyword “nuclear power plants”, shown in the left figure, this keyword was not predicted to be among the most relevant keywords in the resulting model, shown on the right figure. Instead, the user model appears to have remained largely the same, regardless of the additional feedback given by the user. Reprinted from Publication I by permission, © 2015 ACM.

to information retrieval by removing unrelated questions and slightly re-wording some of the existing questions. The user experiences were also measured using structured interviews during and after the experiment.

3.4.6 Results

There are multiple conclusions that can be made based on the interviews. First, based on the user comments, the proposed improvements did improve the predictability of the interaction. The largest contributor to improved predictability in this case was the predictive visualization, which was mentioned most often by the users as the reason for being able to predict the consequences of feedback actions. Second, the improved predictability due to the predictive visualization was reported by the users to be helpful in completing the search task. However, the expert evaluations did not indicate a corresponding difference in task performance. Thus, it is not clear whether this was only a subjective feeling or also led to objectively improved task performance. Third, the predictive visualization also contributed to the building of mental models, as the users stated that simultaneous movement of the keywords over the radar gave them further insight to the behavior of the system. Fourth, the users preferred and were more satisfied with the system with the improvements, which indicates that users are more willing to use system variants which are more predictable.

There was also an indication that the proposed system resulted in better performance in the focused search task, but surprisingly, worse performance in the broad search task.

3.4.7 Discussion

The fact that task performance effects appear to be opposite for the two task types was not expected in advance. My post hoc hypothesis for this behavior is as follows. In the focused search task, the users had very specific requirements for the type of information they needed to find. This made it easier for the users to determine how the user model should optimally be altered for the relevant information to be found. In contrast, in the broad search task, the requirements for the found information were broad, which made determining the optimal changes more challenging. Thus, although interaction with the baseline system may have caused unpredictable changes in the user model, these changes may also have led to serendipitous information discovery. This means that the users may have been propelled to directions that are objectively relevant but may have felt irrelevant to the user *a priori*, due to the user being relatively unfamiliar with the search topic.

Based on this post hoc analysis, I believe it is plausible that there are tasks where serendipitous discoveries play a significant part in completing the task successfully. One such example could be search tasks with broad information requirements, as was the case in this experiment. In such situations, it is possible that just maximizing predictability is not optimal, if it also leads to fewer serendipitous discoveries.

3.5 Improving Feedback Consistency in Interactive User Modelling

This section discusses the significance of the quality of the observation data in user modelling, lists causes for inconsistencies in the observation data and proposes solutions for improving the quality of the observation data in interactive user modelling, in the context of the SciNet search user interface.

3.5.1 Significance of Observation Quality in Modelling

From the perspective of modelling, it is vital that the observation data contains correct information about the modelled behavior. Aberrations and inconsistencies in the observation data have many names: noise, errors, misclassifications, drift, among others. Although introducing small amounts of noise to the observations has been shown to sometimes improve the robustness of the model [Dwork and Lei, 2009], the general rule is that when the amount of inconsistencies in the observation data increases, the performance of the model decreases [Zhu and Wu, 2004].

3.5.2 Causes for Feedback Inconsistency in User Modelling

There are two general reasons why inconsistencies appear in observation data collected from users. The first reason for inconsistencies is that the interests, task or decision criteria of the user change over time. For example, if the user starts by searching for vacations in a ski resort, but later decides that she actually prefers beach resorts, there has been a drift in the interests of the user. In modelling terms, this can be seen as *concept drift* [Widmer and Kubat, 1996; Tsymbal, 2004] in the context of user modelling.

The second reason for inconsistencies are errors made by the user when interacting with the system. Reason [1990] has presented a comprehensive classification of different types of user errors. Regarding planned actions made by the user, there are two main classes of errors that may happen. The first is called *unintentional errors*. For example, if the user makes a manual error in giving feedback, such as indicates that a keyword is relevant when the opposite is true, this can be seen as an unintentional error. The second class of errors is called *mistaken actions*. For example, when searching for healthy foods, if the user gives high relevance feedback to the search term "Apple", which the user interprets as the name of a fruit, while it actually signifies the name of an IT company, this can be seen as a mistaken action. The difference between these two types of errors is that given the description of the performed action alone, the user can spot unintentional errors but not mistaken actions. In order to spot mistaken actions, the user has to also observe whether the consequences of the action were as intended. From the perspective of the learning algorithm, both types of user errors are generally treated similarly, as spurious random events that occur randomly among the true feedback. Separating these errors from the correct feedback can be seen as *outlier detection* [Hodge and Austin, 2004] in the context of user modelling.

Out of these two reasons for feedback inconsistency in user modelling, concept drift has generally been seen as the more critical issue [Webb et al., 2001]. One reason for this is that a drift may cause large portions of the feedback data to become invalid, while errors often only constitute a constant fraction of the feedback. Another reason may be the similarity between modelling errors and user errors. As the model is in every case an approximation, even if there were no user errors it can be expected that part of the observation data will not agree with the model predictions. Many models even make explicit assumptions about the noisiness of the observation data, for example, by assuming that the feedback given by the user is corrupted with Gaussian noise. As both random modelling errors and user errors can be captured with this same noise model, a distinction between these two is often not made in user modelling.

3.5.3 Detecting Feedback Inconsistency

Traditional Approaches

Hodge and Austin [2004] list three fundamental approaches to identifying inconsistencies in observation data. In *unsupervised clustering* the outliers are determined without using any prior knowledge of the data, simply by the remoteness to other observations [Brito et al., 1997]. In *supervised abnormality detection* the outliers are explicitly modelled as part of the data distribution, which is generally more efficient than unsupervised clustering but requires labelled data [Laskov et al., 2005]. *Semi-supervised abnormality detection* is between these two approaches, meaning that some labelled data exists regarding which points are outliers, but generally not enough to train an adequate supervised classifier [Zhang et al., 2005].

Klinkenberg [2004] further divides the methods for dealing with inconsistent observation data into two categories. In *example selection* the approach is to use a selection rule for choosing the observations which are used for fitting the model. Common example selection approaches include windowing [Widmer and Kubat, 1996] and data selection [Fan, 2004]. In *example weighting* the approach is to use a weighting rule for altering the impact of different observations. Common example weighting approaches include linear [Koychev, 2000] and exponential weighting [Klinkenberg, 2004].

Shortcomings With Traditional Approaches

The existing methods for dealing with concept drift have mostly been studied in the context of *data streams* [Muthukrishnan, 2005]. A data stream is a source of observation data that provides new examples at such a high rate that there may be sufficient resources to read each data item only once, after which it has to be discarded. This is different from user modelling, and especially interactive user modelling, where it is more often the case that only a relatively small amount of data is available about the user's interests [Webb et al., 2001]. In the data stream setting it is often acceptable to be able to adapt to a new concept within some hundreds of observations after the drift takes place [e.g. Gama et al., 2004]. In contrast, in interactive user modelling the entire dataset might only be some tens or hundreds of samples. Thus, in the context of interactive user modelling, in order to be able to quickly adapt to changes in the learned concept, the adaptation would need to take place almost instantly after the concept changes, which is quite a tall order for many of the existing methods developed for the data stream setting.

Unsupervised clustering approaches for outlier detection suffer mostly from the same issue, which is the assumption of abundant observation data. These methods naturally work best when there is ample data, as with small amount of data it is challenging to distinguish between outliers and small clusters of data.

Supervised abnormality detection is challenging to apply in user modelling, as there generally is no labelled data available of outliers in user feedback. Furthermore, due to the subjectivity of user feedback, it might even be impossible to objectively determine what precisely is an outlier as the correct classification might be user-dependent.

Conclusion: Feasible Approaches for Interactive User Modelling
Many of the traditional approaches for drift and outlier detection do not seem to be applicable to user modelling with low amount of observation data. This is due to the modelling assumptions which presume abundant data and often emphasize fast model computation time over detection accuracy.

The only approach for outlier detection which seems to be applicable for user modelling with low amount of observation data is semi-supervised abnormality detection. This means that some part of the data needs to be labelled as inliers or outliers, but it cannot be assumed that all of the data is labelled.

In the interactive user modelling setting, the low amount of available observation data is compensated by the relatively large amount of resources available for performing inference, compared to the data stream setting. This means that by using more sophisticated models that are computationally feasible in the interactive user modelling setting, but not necessarily in the data stream setting, it might be possible to adapt quickly enough to changes in user interests.

Another opportunity in the interactive user modelling setting is the ability to ask for further clarification from the user, who is also the source of the observation data. This would not be possible when the source of the data is not available for interaction during the modelling process.

3.5.4 Interactive Detection of Data Inconsistency

As discussed earlier, the ability to ask for further clarification to the observation data directly from the user is a special feature of the interactive user modelling setting. However, there is little previous research of how this type of interaction should be designed.

In general, multiple methods exist for on-line outlier and drift detection [Gupta et al., 2014]. Some of these are even paired with interactive visual tools, designed to help the user, for example, in detecting outliers from large [Chaudhary et al., 2002] and high-dimensional datasets [Wilkinson et al., 2006] or for interactive cleaning of data [Guyon et al., 1996]. However, the vast majority of existing methods have been designed for situations where the user and the data generating process are completely distinct.

The setting of interactive user modelling is fundamentally different from those explored earlier, for three main reasons. First, the user is also the source of the modelled data, so the system is presenting critique of user actions, rather than the behavior of some external process. This means that the method of giving critique should be designed so that it is not offensive or unpleasant to the user. Second, the detection of outliers is a corrective process that should only be noticeable when something goes wrong. This means that the user is primarily focusing on a separate main task, not on finding inconsistencies in her own actions. Third, the user has also the option of altering her past feedback data on-line. This means that the system should also support the user in this task, for example, by giving suggestions for changes and explanations as to why the system believes the data is inconsistent.

To the best of my knowledge, no published methods exist for interactive data inconsistency detection in the context of interactive user modelling. Perhaps the closest analogue is spell-checking, where the text written by the user is checked for inconsistencies at the same time when the user is writing it [e.g. Macdonald et al., 1982]. Possible issues are highlighted gently to the user for further scrutiny and possible corrections to misspelled words may be presented. However, as the text written by the user in a word processor does not constitute data used for modelling, and as the approaches for spotting misspelled words are different from those for spotting inconsistencies in observation data, the setting is not quite the same.

3.5.5 Proposed Solutions for Improving Feedback Consistency

In Publication II, two methods were proposed to improve the capability of the user to notice drift and errors in her feedback, and make corrections to it: a model for estimating the accuracy of user feedback and a timeline visualization of the past feedback. I will first give a general explanation of the proposed methods, followed by an analysis of their possible limitations.

Modelling Accuracy of Feedback

In Bayesian modelling, automatic relevance determination (ARD) [Li et al., 2002] is a traditional approach to feature selection [Guyon and Elisseeff, 2003]. The general idea is to introduce auxiliary variables w_j which scale the impact of each (normalized) observation data feature j. Thus when w_j is small, the effect of feature j is small, and vice-versa. Based on the assumed amount of significant features, a prior distribution $P(w)$ is chosen. The values w_j are learned at the same time with the model parameters, and their values are affected both by the evidence presented by the data and the prior.

However, it is also possible to formulate a similar ARD approach for weighting the individual observations themselves, as demonstrated by Tipping and Lawrence [2003]. In technical terms, this constitutes an example weighting approach to inconsistency detection, where the weights are learned based on the mutual agreement between observation data points, conditional on the assumed generative model. Further technical justification of this approach was presented recently by Wang et al. [2017], who used the term *reweighted probabilistic models*.

The proposed method, presented in Publication II, is based on a Bayesian linear regression model for estimating the relevance of different keywords that occur in the searchable documents.[2] Using a Bayesian linear model is justified by the fact that there are more coefficients to be learned than user feedback to learn them from. Thus, this is an under-determined inference problem, which requires further regularization for the solution to be well-defined; this is provided by the structured Bayesian prior distributions. A similar approach has been shown to be applicable to outlier detection in robotics in the context of a large amount of low-dimensional observation data [Ting et al., 2007]. The hypothesis was that with the use of more restrictive priors, the same approach should in principle also work with low amount of high-dimensional observation data.

In technical terms, let each keyword i have a feature vector x_i. The user gives relevance feedback y_i to a subset of these keywords. The modelling assumption is that there is a linear relation between y_i and x_i, corrupted with Gaussian noise. However, the variance of the noise is dependent on i; specifically, it is assumed to be σ^2/w_i, where σ^2 is the overall variance and w_i an keyword-dependent weight. Altogether, this can be written as a generative model

$$y_i \sim Normal(y_i|x_i^T\theta, \frac{\sigma^2}{w_i}),$$

where θ are the linear model coefficients. For the model parameters, common prior distributions were used:

$$\theta_j \sim Normal(\theta_j|\mu_\theta, \lambda_\theta),$$

$$\sigma^2 \sim InverseGamma(\sigma^2|\alpha_{\sigma^2}, \beta_{\sigma^2}),$$

$$w_i \sim Gamma(w_i|\alpha_w, \beta_w),$$

where μ_θ, λ_θ, α_{σ^2}, β_{σ^2}, α_w and β_w constitute the hyper-parameters of the model, which are fixed to reasonable constant values based on initial tests.

In addition to this, a semi-supervised approach to outlier detection was used. The assumption was that it is known with certainty that part of the feedback is accurate, meaning non-outliers. For such feedback a different prior distribution is used:

$$w_i \sim Delta(w_i|1),$$

which forces the weights of those feedback items to be 1. The most recent feedback given by the user is always assumed to be accurate. This is a very reasonable assumption in the context of concept drift [cf. Koychev, 2000; Klinkenberg, 2004]. In the case of a user error, this assumption will only delay the detection of the error by one cycle of iterative feedback. In addition to the most recent feedback being assumed to be accurate, the user can also explicitly inform the model about other feedback which she insists is accurate. There is no special prior for feedback that is certainly inaccurate, as it was assumed that upon noticing such feedback, the user will remove or correct it.

[2]The feature vector of a keyword is constructed based on the documents where the keyword appears in.

It should be noted that in general, this approach would not be applicable for outlier detection with data streams, as inferring the w_i requires considering the entire history of observation data. However, with small data sets the approach is feasible.

Regarding inference, the unnormalized posterior does not belong to any standard family of distributions due to the observation likelihood being non-conjugate with the prior distributions. This means that it is more difficult, for example, to find the maximum of this distribution as no closed-form solution is known. In general, an iterative solution approach has to be used for optimizing model parameter values.

The chosen approach was mean-field variational Bayesian inference [Attias, 1999]. The idea of this approach is to approximate the complicated posterior distribution with a set of simple distributions multiplied together, called the *variational distribution*. Variational inference was chosen instead of, for example, Markov Chain Monte-Carlo methods [cf. Gilks et al., 1995], as it is computationally much faster in practice, which is crucial in interactive systems.

Interactive Visualization of Feedback Reliability

For allowing the user to interact with this new model, an interactive timeline component to the SciNet user interface was proposed in Publication II. This interface was used for visualizing the feedback given by the user so far, indicating if some of the feedback was estimated to be inconsistent, and allowing the user to alter the previously given feedback. The timeline component is illustrated in detail in Figure 3.5 and shown in context of the full user interface in Figure 3.6.

The timeline consists of two parts: the upper part, displaying the feedback given during the current search sessions, and the lower part, displaying keywords used in previous search sessions.

The upper part contains one row per keyword that the user gave feedback to. The rows are ordered such that most recent feedback appears at the top. Each row has the name of the keyword on the left, and the relevance score given to the keyword on the right. The relevance score is visualized as a green bar, with larger width indicating higher relevance score. The names of keywords with inconsistent feedback are highlighted with yellow color, such that the saturation of the color is more intense when the value of w_i is smaller. In other words, the more confident the system is that the feedback is inconsistent, the more salient it becomes in the interface. When the user hovers the mouse pointer on top of the name of a keyword, the background color of the keyword changes to blue, and two new interaction options are displayed to the user. By clicking the x-symbol next to the keyword, the user removes the feedback item altogether. By clicking the lock-symbol, the user indicates that she is confident that this feedback is accurate. When the user hovers the mouse on top of the green bar, she can change the relevance score given to the keyword by clicking at the location which corresponds to the desired new relevance.

The lower part initially contains one row per previous search session per-

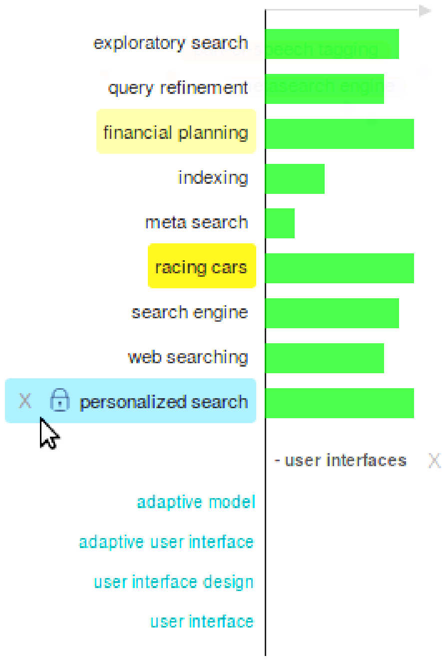

Figure 3.5. Illustration of the layout of the timeline user interface component. Explanation in main text. Reprinted from Publication II by permission, © 2016 ACM.

formed during the search task. The purpose of this part is to allow the user to conveniently re-find keywords used previously, in case the user has recurring interests. On each row, the initial keyword query for that search session is displayed as the title for that session, on the right hand side of the timeline. When the user clicks on the title, one row is added below it for each keyword that was given feedback to in that session, with the keyword name displayed on the left-hand side. When the keywords are visible, the user can give feedback to any of the previously used keywords in the same manner as for the keywords belonging to the current session. The user can also hide the keywords belonging to a previous session by clicking on the session title again. In addition, if there are irrelevant previous search sessions, the user can remove them by clicking the x-symbol next to the session title.

Limitations

One limitation of both of these methods is that they are not practical if the amount of feedback is very large. First of all, the timeline interface may become cumbersome to use if it contains a large number of keywords, as the user will need to manually scroll the timeline to find a keyword of interest. Also, inference becomes more expensive as the number of past feedback grows, which might slightly degrade the responsiveness of the interactive system. However, as the amount of feedback given in this particular case example was relatively small for each search session, these limitations did not appear to cause any issues.

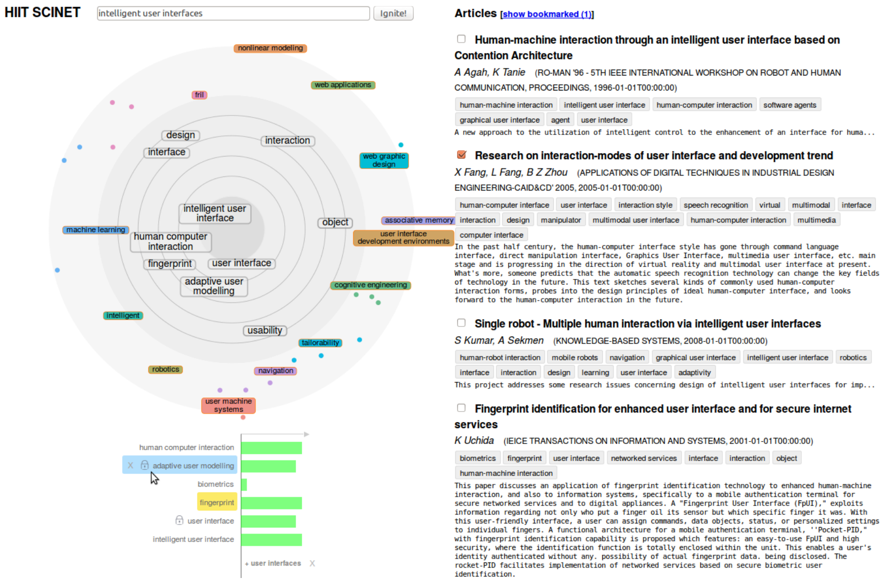

Figure 3.6. Illustration of the full user interface layout, with the timeline placed below the radar visualization.

3.5.6 Evaluation

Simulation Study

The proposed approach was first evaluated in a simulation study, presented in Publication II. In the study, a simulated user searched for relevant documents among the 20 Newsgroups dataset [Rennie and Lang, 2008]. The study compared the proposed method to a baseline approach with the difference that all feedback was assumed to be equally accurate, so that $w_i = 1$ for all i. In addition to this, an oracle was used which knew precisely which of the user feedback was correct and which was not. The performance of the methods was measured with information retrieval F1-score, as a function of the amount of feedback.

At each round of the simulation, the user first gave relevance feedback to two random documents[3]. With certain probability the feedback given by the user would be significantly altered from the correct value. After this, one of the previously given feedback items was highlighted to the user. The simulated user operated in four modes, reacting differently to the highlighted feedback. When the highlighted feedback was incorrect, the user either corrected or did not correct the feedback. When the shown feedback item was correct, the user either indicated or did not indicate that the feedback was accurate.

[3]For simplicity, the regression model was slightly altered to predict relevance of documents based on document feedback in this experiment.

User Study

In Publication II, the performance of the proposed system was also compared against a baseline system in a user study. The baseline system used a similar baseline model as in the simulation study, and in addition, the timeline interface was hidden. The study included 18 users who completed two search tasks, one with each interface, using a balanced study design. Both of the tasks were similar in format, asking the user to explore the subtopics of a certain larger topic.

The prevalence of drift and errors was promoted in two ways. First, the search tasks were formulated to promote moving across different subtopics during search, which was hypothesized to cause drift. Second, the length of the viewable results was limited, which forced the user to search for relevant documents through giving more feedback, instead of just manually scrolling down a long list of search results. This was hypothesized to cause more interactions with the system, and thus more user errors as well.

The performance of the systems was measured in five dimensions. The task performance was measured by an expert who scored the performance of the users. The general usability of the system was measured with the SUS usability questionnaire [Brooke, 1996] and the information retrieval performance with the same ResQue recommender system performance questionnaire [Pu et al., 2011] as used in Publication I, but with four additional questions related to usability after the user had made an error. The user experiences were also measured using semi-structured interviews after the experiment. The keywords seen and manipulated by the users during the experiments were logged, together with the documents found by the users; the quality of the keywords and documents was rated by an expert.

3.5.7 Results

In the simulation experiment, the proposed model was able to achieve performance comparable to the oracle with the help of further interaction with the user. This demonstrates the advantage of asking additional clarification from the user to her feedback. The simulation experiment also demonstrates that highlighting feedback based on the estimated accuracy works much better than selecting them randomly, as was done by the baseline system.

The user experiment partially confirms that the performance of the search engine was improved thanks to the use of the proposed user model and the interactive timeline. This is demonstrated by the 10 % higher ResQue score received by the proposed system, compared to the baseline. However, the effect in terms of objective quality of displayed information was small, as the expert evaluation found no significant differences between the relevance of information displayed to the users by the two systems.

The user experiment also demonstrated that the users performed more interaction with the proposed system. This demonstrates that the users found the new interaction options useful, with the ability to delete and revise feedback being the most used options. Also, as the proportion of changes and deletions of all the keyword interactions was notable, roughly 30 %, this indicates that either there was drift in the search intents of the users, or that the users had made errors in giving feedback.

Both the interview results and answers to individual questions in the two questionnaires demonstrate that the proposed system made it easier for the users to make corrections to previously given feedback and to form better mental models of the system behavior. These improvements can mostly be attributed to the presence of the timeline interface and the interaction options it offers.

3.5.8 Discussion

I have two post-hoc hypothesis for the lack of difference between the two systems, regarding task performance, keyword quality and document quality.

The first hypothesis is that even though the significance of giving feedback was emphasized for finding good results, through limiting the number of results available for viewing, this did not increase the amount of feedback enough. Based on the logs of user activity, the users gave generally less than 10 instances of feedback per search query. However, when compared to the simulation experiment, the effects of the proposed model seem to be more clearly noticeable only after roughly 20 instances of feedback. Thus, as expected, no significant difference in performance was found between the two systems when the users only gave this low amount of feedback. To remedy this, a more efficient approach for promoting the amount of feedback given should be used. However, just reducing the amount of viewable results even further would likely have caused frustration among users, as one user already commented that she would have wanted to scroll through more results. This means that some other complementary approaches would have to be used as well.

The second hypothesis is that as the users were both made explicitly aware of the feedback they had given, and were notified by the system of the consistency of their feedback, they were able to deal with the concept drift preemptively, instead of relying on the system to alert them when drift happens. Support for this hypothesis is given by the user who commented that she made the core keywords stay accurate before exploring the various subtopics, which indicates that the user was able to anticipate which keywords might be seen as outliers by the search engine, before giving possibly conflicting feedback.

4. Likelihood-Free Inference for Cognitive Models

This chapter gives an overview of the line of research related to RQ2 – *Can the parameter values of cognitive models be learned efficiently from observation data, together with principled uncertainty estimates, without strict requirements for the type of the observations or for the structure of the cognitive model?* First, a definition of a cognitive model is given with examples, outlining both the simpler traditional cognitive models and more complicated modern models. After these, challenges related to inference of cognitive model parameter values are discussed and existing methods for inference are described. Finally, approximate Bayesian computation based on Bayesian optimization with a Gaussian process surrogate model is introduced, and its applicability to inference of cognitive model parameter values is evaluated.

4.1 Cognitive Models

Cognitive models are a subset of user models distinguished by the explicit assumptions they make regarding the cognitive processes or capabilities of the user. Examples include assumptions about the user's capabilities to perform manual operations [Card et al., 1980], visual perception [Lohse, 1993], working memory capacity [Byrne and Bovair, 1997] and skill level [Desmarais and Baker, 2012]. One of the main reasons why cognitive models are used in HCI is their ability to predict how users will interact with various proposed designs of the interactive system [Olson and Olson, 1990]. Another reason is the possibility of personalizing the user interface based on a model of the user [Langley, 1999; Duric et al., 2002]. In addition, computational models of cognition are also used in the field of cognitive science for explaining various cognitive behaviors [Lazer et al., 2009].

One of the key features of a cognitive model is the behavioral *policy* of the user, which describes what actions the user will perform in any given situation. Next, I will briefly describe how the approaches for constructing these policies have evolved over time.

4.1.1 Traditional Cognitive Models

Many of the early cognitive models were formulated for situations where it was assumed obvious what actions the user will perform in order to complete a task [e.g. Card et al., 1980]. For example, in the GOMS formalism[1], the modeller explicitly lists all the actions the user will perform to achieve a certain goal [John and Kieras, 1996]. Such models have been used, for example, for predicting how long it will take a user to perform certain kinds of text editing task [John and Kieras, 1996]. For example, the actions required for selecting a certain word from a short body of text can be stated explicitly as: (1) look at the word on the screen, (2) move the cursor on top of the word, (3) double-click the left mouse button. More complex tasks can be described with the help of simpler sub-tasks; for example, the text selection task can be used as a sub-task when describing a copy-and-paste task.

However, this traditional approach to modelling the policy of the user is generally only applicable for tasks that are rather trivial or easily decompose into a sequence of trivial sub-tasks. There are two reasons why more complicated tasks might require a different approach. The first reason is the bounded rationality and cognitive capabilities of the user [Simon, 1990; Gigerenzer and Goldstein, 1996]. In other words, when performing a complicated task, the user is not able to precisely observe, consider or remember all the facts that are relevant for performing the task. However, taking all this uncertainty into account may make the policy complicated and laborious to describe manually. The second reason is that certain tasks are inherently complex and require policies that are difficult for humans to explicate. In general, many tasks performed by humans require considerable expertise for good task performance [Ericsson and Lehmann, 1996]. Regarding these kinds of tasks, it is unlikely that a non-expert could describe the behavioral policy of an expert user, and furthermore, it might be that even the expert user herself cannot explicate the precise reasons for her behavior [e.g. Maxwell et al., 2000], for example, due to unconscious adaptation. For example, in the case of word selection, if the size of the document is large, the sequence of steps needed for determining the location of the word is much more complicated compared to a small document. The user might remember only approximately where the word is located, which requires the policy to be formulated as a function of this probabilistic knowledge. The policy may also be very sophisticated if the user is an experienced user of text editors, as there are multiple alternative ways to perform this search operation, each being suitable in certain contexts.

[1] Definition of the Goals of the agent, Operations to interact with the environment, Methods for achieving goals, and Selection rules for choosing between alternative methods.

4.1.2 Modern Cognitive Models

Currently, there are two main approaches for modelling user behavior in situations where a complicated policy is necessary for adequately predicting the behavior of the user. One approach is to make the language for describing the policies more flexible. Perhaps the most widely used framework in this respect is ACT-R [Anderson, 1996]. In ACT-R, the complicated policy emerges from interaction between two model components: the procedural knowledge and the declarative knowledge of the user. The procedural knowledge consists of a set of *production rules*, which describe how various tasks are performed in terms of the parameters of the task and the state of the declarative knowledge. The declarative knowledge consists of a set of *chunks*, which describe various facts related to the environment of the user in terms of ontological relations, and may be altered by the production rules. The ACT-R model simulates the behavior of the user by first initializing the declarative knowledge based on the initial state of the environment, and then by repeatedly applying the production rules until a terminal state is reached.

Another approach is to have a completely implicit definition of the policy, by defining it to be a solution to an optimization problem. Perhaps the most widely used approach in this respect is computational rationality [Lewis et al., 2014; Gershman et al., 2015]. The idea of computational rationality is to model the complex behavior of the user as an optimal adaptation to the present goals and environmental limitations. One popular way to formulate this is by using the *reinforcement learning* (RL) formalism [Sutton and Barto, 1998]. In this formalism, the environment of the user is often assumed to be a Markov decision process (MDP), consisting of a set of actions A, states S, state-transition probabilities T, reward function R, temporal discount γ and initial state distribution $P(s_0)$. The policy π is a distribution $P(a|s)$, where a is an action and s is a state. Performing an action a in state s results in the following state s' being drawn from $P(s'|a,s)$ as defined by T. An optimal policy π^* is defined to be such that it maximizes the expected cumulative discounted sum of rewards,

$$\pi^* := \operatorname*{argmax}_{\pi} E\Big[\sum_{s_t \in \xi_\pi} \gamma^t R(s_t)\Big],$$

where ξ_π is a trajectory $(s_0, a_1, s_1, \ldots)$ generated by starting from an any initial state s_0 and acting according to π thereafter. Thus, given a description of the environment, a RL model first solves the corresponding optimal policy and then uses this policy for predicting the behavior of the user in any given situation.

4.2 Inference of Cognitive Model Parameter Values

Although in some special cases it might be feasible to manually tune the parameter values of cognitive models, in general, the applicability of these models is greatly enhanced when they are paired with an efficient automated method for inferring the model parameters based on observation data. This section first outlines reasons why this is not trivial with modern cognitive models, discusses existing methods for performing inference and lists shortcomings of existing methods, especially related to the situation where the type of the observations varies.

4.2.1 Challenges in Cognitive Model Parameter Inference

There are two main challenges in inferring parameters values of modern cognitive models, such as ACT-R or RL models, based on observation data. The first challenge is with the complexity of the model structure, and the second with the unavailability of sufficient observation data. Both of these ultimately lead to difficulties with evaluating the observation likelihood $P(D|\theta)$.

Complexity of Model Structure
In general, the more faithfully a cognitive model attempts to mimic the structure of a human cognitive system, the more complex the structure of the model becomes. This is especially salient with the ACT-R model family. For example, the ACT-R 7.0 software library contains eight core modules, a variety of additional modules and can be even extended by custom user-defined modules [Bothell, 2017]. Each of these modules contains functionality related to a particular aspect of the cognitive system, such as auditory processing or motor control, and the final cognitive model is constructed by composing functionality from multiple modules. However, in practice, this means that there are no guarantees that the observation likelihood $P(D|\theta)$ can be written in closed form. From a practical perspective, this is because the model is essentially a software executable, instead of an analytical mathematical model, which makes it difficult to construct the likelihood function. Furthermore, the model structure may be very complicated, which leads to a possibly intractable likelihood.

One method for dealing with this issue was proposed by Said et al. [2016], who replaced the Lisp library traditionally used for defining ACT-R models with an approximate analytic definition of the model structure which guaranteed that the likelihood and its derivatives exist. While this approach was shown to work for a very simple ACT-R model, this approach still has multiple issues. First, existing ACT-R models cannot be used as such, but need to be re-written in this analytic format from ground-up. Second, writing the model in purely analytic form may be cumbersome for more complicated models. Third, there are no guarantees to how the model predictions change when approximations are introduced. Fourth, there are no guarantees as to how computationally expensive the likelihood or its derivatives are to evaluate.

Unavailability of Sufficient Observation Data
The second challenge relates to the unavailability of sufficient observation data. By sufficient observation data, I mean a large enough amount of observations of

suitable type, measured with sufficient accuracy, such that the model parameter values can be estimated reliably and conveniently based on this data.

With cognitive models, the main reason for the unavailability of sufficient observation data is the difficulty of observing the states of the actual cognitive processes that occur in the mind of the user. In many cases it is only possible to make *proxy measurements*, or substitute observations, that somehow relate to these processes; one example of such a proxy measurement is the time required to complete a task. In some cases, it is also possible to make measurements more closely related to cognitive processes, for example, by measuring the visual fixations of the user with an eye-tracking camera. However, for example, it is unlikely that one could conveniently measure the state of the user's short- or long-term memory, at least with modern equipment. Although some progress has been made recently in using fMRI (functional magnetic resonance imaging) measurements for validating cognitive models [Borst and Anderson, 2015], in the vast majority of cases various proxy measurements have to be used instead. There are of course other possible reasons for insufficient observation data as well, such as physical occlusion, faulty measurement devices, censored or missing observations, and so on.

Due to the fact that the modelled cognitive behavior and the observation data are separated by one or multiple steps of transmutation and filtering, the observation likelihood $P(D|\theta)$ is generally complicated. To give a concrete example, with RL models the observation likelihood can be quite simple when the observation data is available as state-action trajectories ξ:

$$P(\xi|\theta) = P(s_0) \prod_{t=0}^{T-1} \left[\pi^*_\theta(s_t, a_t) P(s_{t+1}|s_t, a_t)\right],$$

and there are multiple existing *inverse reinforcement learning* (IRL) methods for parameter inference in this case [e.g. Ramachandran and Amir, 2007; Ziebart et al., 2008]. However, if one is only able to observe a proxy measure which can be seen as the true trajectory being filtered through a function σ, the available observation is $\xi_\sigma = \sigma(\xi)$. In this case, the observation likelihood has the form

$$P(\xi_\sigma|\theta) = \sum_{\xi \in \Xi_{ap}} \left[P(\xi_\sigma|\xi) P(\xi|\theta)\right],$$

where Ξ_{ap} is the set of all plausible trajectories that could have caused the observation ξ_σ which may be arbitrarily large and thus make the likelihood arbitrarily expensive to evaluate. Some steps have been taken to alleviate this, for example, regarding probabilistic state observations [Kitani et al., 2012] and observations missing from the trajectory [Bogert et al., 2016]. However, to the best of my knowledge, no existing IRL method is able to deal with the more general case where the observation data of the user's behavior is given in arbitrary format, such as in the form of durations between certain measurable events.

4.2.2 Current Methods for Likelihood-Free Parameter Inference

The key problem caused by a complicated observation likelihood is the difficulty of applying many of the existing parameter inference methods, as they rely on being able to repeatedly evaluate the likelihood function in order to find parameter regions that correspond to large likelihood.

For example, if the derivatives of the likelihood function can be easily computed or estimated, various gradient-based optimization methods [cf. Bertsekas, 1999] can be used for finding the maximum likelihood (ML) solution. If the derivatives cannot be easily computed or estimated but the (unnormalized) likelihood function itself can be evaluated, Markov-Chain Monte Carlo methods [cf. Gilks et al., 1995] can be applied. However, if the likelihood function is computationally expensive or impossible to be evaluated, these likelihood-based inference methods become impossible to apply. Unfortunately, the likelihood functions of modern cognitive models often end in the latter class.

The most common approach has thus been to perform likelihood-free inference. The core idea of likelihood-free inference is to estimate the likelihood values empirically through simulating data from the model and comparing it to the observed data. This way, it is possible to avoid evaluating the likelihood function altogether but the trade-off is that multiple simulated datasets need to be generated from the model instead.

The most rudimentary likelihood-free approach has been to set the parameter values manually [e.g. Lewis and Vasishth, 2005; Chen et al., 2015], which might be feasible in some special cases but is not a scalable or accurate method for inference in general. Another approach has been to optimize the estimated prediction error using a black-box optimization method. Examples of this includes brute-force optimization [e.g. Blouw et al., 2016; Lee et al., 2016] and derivative-free local optimization methods [e.g. Vandekerckhove and Tuerlinckx, 2007; Logačev and Vasishth, 2016], such as Nelder-Mead simplex optimization [Lagarias et al., 1998]. However, there are two problems with these commonly used likelihood-free approaches.

The first problem relates to the inefficiency of brute-force optimization. In general, brute-force optimization requires evaluating the optimized function a large number of times. The number of evaluations usually grows exponentially with the number of parameters. Thus, especially when the parameter space is high-dimensional, brute-force optimization is no longer a practical approach. Furthermore, brute-force optimization is not able to guide the sampling process in an efficient manner, which may lead to a large amount of simulations with parameter values that are almost certainly not optimal.

The second problem relates to the ability to estimate the uncertainty of the parameter estimates. Such estimates are often not provided by point-estimation methods, such as Nelder-Mead simplex optimization. However, especially due to the fact that in general there is insufficient observation data as discussed in the previous subsection, it is likely that there will remain considerable uncertainty regarding the true parameter values of the cognitive model. Intuitively, when the posterior distribution of the model parameters is concentrated in a small region of the parameter space, a point-estimate is a reasonable summarization of the current knowledge regarding the true parameter values. However, when the posterior distribution is wide or multi-modal, this is no longer the case. In the case of wide distributions, it would be informative to estimate the region of plausible parameter values instead of a single point, and in the case of multi-modal distributions it would be informative to estimate the locations of each of the modes and their relative plausibilities.

4.3 Proposed Solution for Inferring Cognitive Model Parameter Values

In Publications III, IV and V, approximate Bayesian computation (ABC) [cf. Sunnåker et al., 2013] combined with Bayesian optimization (BO) [cf. Snoek et al., 2012] using Gaussian processes (GP) [Rasmussen and Williams, 2006] is proposed for inferring the parameters of cognitive models. ABC is a likelihood-free inference method based on the Bayesian statistical formalism, which allows for principled estimation of parameter uncertainty. BO using GPs is a widely used black-box optimization method that is able to estimate the global optimum of a function with a low number of samples, which is important when simulations from the model are expensive, which is often the case with modern cognitive models. I will first give a general introduction to the proposed method, followed by an analysis of its predicted benefits and possible limitations, and a discussion of the implementation.

4.3.1 Introduction

The core idea of ABC is that given a *discrepancy function* δ for quantifying the difference between model predictions and observation data, the posterior distribution of the model parameters can be estimated without the use of the observation likelihood function. First, let $\delta(D,D')$ be a deterministic non-negative function that is zero if and only if $D = D'$, and with slight abuse of notation, let $M(\theta)$ be a stochastic function that generates simulated observation data D_{sim} from model M with parameters θ. Now the probability of generating observed data D_{obs} from model M with parameters θ is

$$P(D_{obs}|\theta) \equiv P\big(D_{obs} = M(\theta)|\theta\big),$$

because $M(\theta)$ is the assumed generative process. This can also be stated as

$$P\big(\delta(D_{obs},D_{sim}) = 0|\theta\big).$$

What this means is that the value of the observation likelihood function equals the probability of simulating the observation data from the model given the parameter values. This probability can be estimated empirically using simulations from the generative model. However, this empirical estimate may have high variance as the probability of precisely simulating D_{obs} from $M(\theta)$ is often very small. This is the motivation for the ABC approximation, which is to relax this requirement by defining an ε-approximate likelihood

$$\tilde{P}_{\varepsilon}(D_{obs}|\theta) := P\big(\delta(D_{obs}, D_{sim}) < \varepsilon|\theta\big),$$

which is easier to estimate empirically with a sufficiently large ε. A further common relaxation is to allow δ to be zero also when the two datasets are "sufficiently similar", for example, when they have equal descriptive statistics.

One of the main limitations for performing ABC has been the need to perform a large number of simulations using the generative model. For example, a naïve approach would be to simulate in a brute-force manner multiple times with each plausible θ for estimating the value of $\tilde{P}_{\varepsilon}$ near every point of the parameter space. One way forward was presented recently by Gutmann and Corander [2016], who proposed both to use a Gaussian process (GP) [Rasmussen and Williams, 2006] surrogate model of the discrepancy function for approximating the ABC likelihood and to use Bayesian optimization [cf. Snoek et al., 2012] for choosing the locations to use for fitting the GP surrogate. This method resulted in significant speedups in estimating the posterior of a bacterial infection model [Gutmann and Corander, 2016], which was one of the core motivations for using this approach for cognitive models as well.

Bayesian optimization is a family of optimization methods where the optimization process is guided by a probability distribution over possible true underlying functions. There are generally two parts to the BO algorithm: a surrogate model and an acquisition function.

The surrogate model provides the probability distribution $P(f|\{(\theta,\delta)\})$, where f is the underlying function to be optimized and $\{(\theta,\delta)\}$ are realizations from the function: $\delta \sim f(\theta)$. In the current approach, a GP regression model is used as the surrogate. The GP model assumes that *a priori* the function has mean $\mu(\theta)$ and that the covariance between any two locations θ and θ' is given by the kernel function $k(\theta,\theta')$. After observing actual realizations $\{(\theta_1,\delta_1),\ldots,(\theta_N,\delta_N)\}$, the mean and kernel estimates are updated to better approximate the actual behavior of δ as a function of θ. An attractive property of the GP regression model is that the posterior predictive distribution for the value of δ at any individual θ is a Gaussian distribution with simple analytic expressions for both mean and standard deviation. This means that after fitting the GP model, it is simple to estimate the ε-approximate likelihood at any θ by using the cumulative distribution for the Gaussian distribution.

The process of choosing the locations where δ is evaluated for fitting the GP surrogate is directed by the acquisition function Acq. At each step of the optimization process, the optimum of Acq is found as a function of θ and these parameter values are then used for evaluating δ. As derivatives for Acq often exist, it is generally straightforward to find a local optimum with standard gradient-based optimization methods [e.g. Liu and Nocedal, 1989].

The value of the acquisition function depends on both the current state of the surrogate model and on the value of θ. The acquisition function is generally formulated so that large values of Acq correspond to a preferred balance between exploration and exploitation; exploration generally corresponds to sampling regions of the parameter space with high uncertainty of the function value, while exploitation corresponds to sampling regions with low estimated function value. The acquisition function used in this thesis is the lower confidence bound (LCB) acquisition rule [Srinivas et al., 2010], which can be written as

$$Acq_{LCB}(\theta) := \mu(\theta) - c\sigma(\theta),$$

where $\mu(\theta)$ and $\sigma(\theta)$ are the GP posterior predictive mean and standard deviation at θ, and c is an "exploration rate" parameter. Reasons for choosing this acquisition function are the simplicity and speed of evaluation. For parallel acquisition, two different approaches were used; either multiple stochastic draws near the optimum based on the acquisition function curvature [cf. Gutmann and Corander, 2016] or local penalization [González et al., 2016].

One additional feature of ABC is the ability to use prior distributions for the parameter values. This allows the approximate posterior distribution to be a balance between the prior knowledge and the evidence of model fit presented by the approximate likelihood function.

4.3.2 Predicted Benefits

The use of efficient likelihood-free inference should provide multiple benefits over alternative approaches to cognitive model parameter inference.

First, the only requirement for the cognitive model structure and for the type of observations is that similar simulated data can be repeatedly generated from the cognitive model using different parameter values. This is, in general, a reasonable requirement for any predictive model, as the main purpose of these models is precisely to predict the behavior of the user under varying assumptions. This also means that it is guaranteed that the inference process can be executed, as the bottleneck is with being able to repeatedly simulate observations from the model, which is assumed to be feasible.

Second, the inference process is theoretically well-founded in Bayesian statistics. This means that it provides principled estimates of parameter uncertainty.

Third, the method aims for optimal efficiency through the use of Bayesian optimization. This means that the method is likely very sample-efficient, which is important if simulations from the cognitive model are expensive.

4.3.3 Limitations

One limitation of the method is inherent to all likelihood-free inference; as the inference is performed in model-agnostic manner, this also means that the structure of the model is not explicitly used to make the inference process more efficient. For example, if it would be possible to also approximate the derivatives of the likelihood surface, this could likely be used to improve the efficiency of BO. However, to the best of my knowledge, the ABC formalism does not allow these derivatives to be approximated efficiently.

Another limitation of the method is that a number of hyperparameter values need to be decided before the method can be used. These hyperparameters include, for example, the assumed lengthscale of the discrepancy function, and the balance between exploration and exploitation in the acquisition rule. If no information about reasonable values is known beforehand, initial tests may be required to estimate suitable values for these parameters.

A third limitation is that the basic GP regression model does not scale well in high dimensional parameter spaces. However, there are extensions that alleviate this issue, such as the use of random embeddings [Wang et al., 2016].

One limitation of this line of research in general is that out of all possible likelihood-free inference methods only ABC was considered. One example of recently developed likelihood-free methods are generative adversarial networks (GANs) [Goodfellow et al., 2014], where the idea is to train one artificial neural network to reproduce the observation data distribution and another one for classifying data sets as similar or different from the original observation data. However, a major limitation of the GAN approach is that it effectively results in black-box explanations of the studied behavior. In comparison, ABC infers a posterior for the parameters of a model defined by the researchers, which can be designed to be easily interpretable. Another example between the GAN and traditional ABC approaches is the classifier ABC approach [Gutmann et al., 2014, 2018], which is otherwise similar to standard ABC, except that the discrepancy function is replaced by a binary classifier, similarly as with GANs. This approach could be beneficial in cases where designing the ABC discrepancy function by hand is difficult.

4.3.4 Implementation

The applied method was implemented as part of a general software library for likelihood-free inference, presented in Publication VI. The library was implemented in Python and allowed the proposed method to be used for generative models implemented either directly in Python or as external executable programs. In this thesis, both ACT-R and RL models were used. The RL models were implemented in Python, while the ACT-R model was implemented in Common Lisp and compiled to run as an independent executable program.

4.4 Evaluation

The performance of the proposed method was evaluated both with an ACT-R model and with different reinforcement learning models.

4.4.1 ACT-R Model

The proposed method was used for estimating the posterior distribution of four parameters of an ACT-R model in Publication V. The model in question was proposed by Tenison et al. [2016] and its purpose is to model the phases of skill acquisition. The model was fit to aggregated observation data from 40 users collected by Tenison et al. [2016].

Three baselines were used for comparison. The first baseline was grid search, which evaluated the discrepancy function over a mesh of points in the parameter space and used the best found point as the final estimate. The second baseline was Nelder-Mead simplex optimization, which starts from a random position in the parameter space and iteratively progresses to locally better positions. The third baseline was Bayesian optimization of the discrepancy function without the additional computation of the ABC posterior distribution.

The first measure for comparison was efficiency, measured by the prediction accuracy with best found parameters, as a function of the used computational resources. Efficiency is important for saving limited computational resources, especially in online user modelling. The second measure was informativeness, measured by the ability of the methods to visualize how the prediction accuracy or posterior probability varies as a function of the parameter values. Informativeness is a relevant measure as it allows estimating the reliability of the parameter estimates and reasonable regions for plausible parameter values when the remaining uncertainty is high.

The performance of the proposed method versus manual tuning was also compared in Publication V, by comparing the predictions made with the mean value of the ABC posterior distribution to those made with parameters fit by hand in Tenison et al. [2016].

4.4.2 Reinforcement Learning Models

The proposed method was used for estimating the maximum a-posteriori (MAP) parameter values of an RL model in Publication III, and the full posterior in Publication IV and Publication V. The model in question was proposed originally by Chen et al. [2015] and its purpose is to model the visual search process of an user in the context of a vertical list of items. Multiple minor changes to the model were presented in Publication III to explore ways to improve the predictive accuracy. A slightly simplified version of the model was used in Publication IV and Publication V to reduce the duration of solving the optimal behavioral policy. In all of the cases, the model was fit to task completion times from multiple users collected by Bailly et al. [2014]. Both in Publication III and Publication V, models were also fit to individual users to demonstrate that likelihood-free inference is applicable also for small datasets collected, for example, from an individual user. In Publication V, the proposed method was compared to two baselines similar to the ACT-R model discussed above. The performance of the proposed method was compared to manual tuning in Publication III by comparing the predictions made with the MAP estimate to those made with parameters fit by hand in Chen et al. [2015].

In Publication IV, the proposed method was also used for estimating the parameters of a toy RL model, where an agent navigated in a small grid world environment. The method was compared to an exact solution and a Monte-Carlo estimate of the exact solution.

4.5 Results

The proposed method was able to infer reasonable approximate posterior distributions for parameters of two modern cognitive model families: ACT-R and RL. This demonstrates that the method is applicable to various modern cognitive models. The proposed method was able to perform inference based on realistic aggregate observation data. In the case of the ACT-R model, the observations were durations of various learning phases, while with the RL model, the observations were task completion times. This demonstrates that the method is applicable to various types of observations.

Compared to Nelder-Mead optimization, the proposed method was significantly more informative, somewhat less efficient but, on the other hand, demonstrated less susceptibility to over-fitting. Compared to grid search, the proposed method was more informative and significantly more efficient. The method also achieved significantly better results compared to manual tuning. Compared to using just Bayesian optimization, ABC allowed the use of prior distributions, which improved the overall credibility of the results due to incorporation of prior knowledge of reasonable parameter values. The ABC posterior was slightly more informative about the parameter regions with high posterior probability, compared to just the GP model of prediction error. Furthermore, as the output of ABC inference is a distribution, it can be used in many ways that a plain GP model cannot, for example, as a prior distribution for further inference.

Related to RL models, the proposed method also reached equally good inference performance compared to an exact but computationally expensive solution and an RL specific approximate method (Monte-Carlo approximation of the exact solution).

4.6 Discussion

One explanation of the performance gap between Nelder-Mead optimization and the proposed method is the fact that as a local optimization method, Nelder-Mead makes the implicit assumption that the optimization surface is unimodal, while the proposed method does not make this assumption. Because of this assumption and the fact that in the two examples the posteriors indeed were unimodal, Nelder-Mead was able to make immediately efficient process towards the optimum while the proposed method, due to the use of global Bayesian optimization, needed to initially sample all across the parameter space to be certain that there were no other better optima. However, if the posterior would have had multiple local optima, it is likely that the proposed method would have found the global optimum, while Nelder-Mead could have only found one of the local optima. Through starting Nelder-Mead optimization from multiple random locations in the parameter space similar guarantees could be given but this would then close the performance gap between these two methods and the proposed method would still be significantly more informative.

5. Conclusions

5.1 Research Question 1

RQ1 was "Can the satisfaction and task performance of search engine users be improved through improving the predictability of the interaction with the search engine and through improving the capability of users to notice and correct possible interaction errors?"

The ability of the proposed methods to improve user satisfaction was studied in Publication I and Publication II through questionnaires and interviews. In both cases, the users preferred the improved systems over the baselines, which indicates higher satisfaction of the system as a whole. The users also mentioned multiple reasons that explain this satisfaction of system performance; these include increased ability to understand system behavior, predict the consequences of user actions, and availability of convenient interaction options.

The ability of the proposed methods to improve task performance was studied in Publication I and Publication II through questionnaires, expert evaluations and simulation experiments. In the experiments conducted in Publication II, the proposed methods led to improved information retrieval performance in simulation experiments, and in questionnaires the users ranked the performance of the proposed system higher than the baseline. However, the expert evaluations conducted both in Publication I and Publication II remain inconclusive regarding objective improvement in the quality of retrieved information or in task performance.

Thus, I conclude that the satisfaction of search engine users of the system performance can be increased through the interactive machine learning interface methods presented in this thesis. I contribute this increase in satisfaction to three factors.

1. The improved ability of users to comprehend and reason about system behavior. In this thesis, this was improved through predictive visualization, the timeline interface and the automatic relevance determination (ARD) model. Predictive visualization allowed the user to gain insight about how her incremental

feedback causes changes to the user model. The timeline interface, together with the feedback accuracy predictions made by the ARD model, allowed the user to gain insight into how the complete set of user feedback affects the user model.

2. The improved ability of users to predict the consequences of their actions. In this thesis, this was improved through predictive visualization and intuitive control. Predictive visualization gave the user explicit cues that indicated how the model will change, allowing the user to adjust her feedback before committing to it. Intuitive control was complementary to predictive visualization, making the user model easier to control by adjusting its dynamic behavior.

3. The availability of convenient interaction options that match the needs of the user. In this thesis, this was increased through the timeline interface. The interface allowed the user to adjust previously given feedback, which was useful when the search intent of the user was volatile.

Regarding task performance, I conclude that based on user interviews, there is some indication that the proposed methods improve it. However, the size of the effect is likely small, and thus further experiments with more users would be required to reliably estimate the size of this effect.

These advances are important for improving the usability of interactive AI systems whose behavior is directed by a user model. The proposed methods are quite general, and can be applied for a variety of different systems.

5.2 Research Question 2

RQ2 was "Can the parameter values of cognitive models be learned efficiently from observation data, together with principled uncertainty estimates, without strict requirements for the type of the observations or for the structure of the cognitive model?"

Approximate Bayesian computation based on Bayesian optimization with a Gaussian process surrogate model was used to infer the parameters of modern cognitive models in Publications III, IV and V. In Publication III, an initial proof-of-concept was demonstrated. In Publication IV, it was demonstrated that the proposed method yields inference results comparable to likelihood-based methods for RL models but considerably faster. In Publication V, it was demonstrated that the method is applicable equally to ACT-R and RL models, and that it is efficient and informative regarding parameter uncertainty compared to alternative ways of optimizing prediction accuracy. In Publications III and V it was demonstrated that in case studies the method yields notable improvement over manual tuning of parameters and that it can also be used with small datasets from individual users.

I conclude that the proposed method is feasible for learning the parameter values of modern cognitive models. The method does not place any unreasonable requirements either on the cognitive model or the type of observations, which is attributable to the likelihood-free approach. The method is relatively efficient in learning the parameter values, which is attributable to the use of Bayesian optimization. The method is also able to provide principled estimates of parameter uncertainty, also taking into account prior knowledge, which is attributable to the use of approximate Bayesian computation. In addition, the method was demonstrated to be usable also with small amounts of observation data, which allows it to be used, for example, for modelling the behavior of individual users.

These advances are important in the scientific research of human cognition and in the practical development of cognitive models. Efficiency makes the modelling process faster, which allows the parameters of currently used models to be inferred reliably with fewer resources, and makes inference feasible also for computationally expensive models. Informativeness gives researchers more insight to the behavior of these models, by allowing the reliability of the parameter estimates to be computed. This allows alternative hypotheses to be compared and possible issues with parameter inference to be uncovered.

5.3 Discussion

The aim of this thesis has been to improve the capability of mutual understanding between users and AI systems. The first part of the thesis focused on improving the ability of users to build mental models and interact efficiently with AI systems, while the second part focused on enabling the inference of more advanced user models. Overall, I see it important to keep on working towards bridging this gap in mutual understanding from both directions, so that truly "symbiotic interaction" [Jacucci et al., 2014] could be achieved one day.

One interesting avenue of research is to further improve the efficiency of methods for inferring cognitive model parameters. Although the proposed method is applicable to a wide range of cognitive models and types of observation data, it is generally not yet fast enough for real-time interaction or online modelling. With reinforcement learning models, one possible way forward is to use transfer learning to speed up the policy learning process [e.g. Ramachandran and Amir, 2007].

After cognitive model inference is fast enough, another interesting direction of research is to use these models in interactive settings. In this situation, the proposed methods for improving user interaction, discussed in the first part of this thesis, could be applied to the cognitive models discussed in the second part of this thesis.

References

Anand, S. S. and Mobasher, B. (2003). Intelligent techniques for web personalization. In *Proceedings of the International Conference on Intelligent Techniques for Web Personalization*.

Anderson, J. R. (1996). ACT: A simple theory of complex cognition. *American Psychologist*, 51(4).

Attias, H. (1999). Inferring parameters and structure of latent variable models by variational Bayes. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*.

Aykin, N. M. and Aykin, T. (1991). Individual differences in human-computer interaction. *Computers & Industrial Engineering*, 20(3).

Baeza-Yates, R., Hurtado, C. A., and Mendoza, M. (2004). Query recommendation using query logs in search engines. In *EDBT Workshops*.

Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*.

Bailly, G., Oulasvirta, A., Brumby, D. P., and Howes, A. (2014). Model of visual search and selection time in linear menus. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*.

Bertsekas, D. P. (1999). *Nonlinear Programming*.

Blouw, P., Solodkin, E., Thagard, P., and Eliasmith, C. (2016). Concepts as semantic pointers: A framework and computational model. *Cognitive Science*, 40(5).

Bogert, K., Lin, J. F.-S., Doshi, P., and Kulic, D. (2016). Expectation-maximization for inverse reinforcement learning with hidden data. In *Proceedings of the International Conference on Autonomous Agents & Multiagent Systems*.

Borst, J. P. and Anderson, J. R. (2015). Using the ACT-R cognitive architecture in combination with fMRI data. In *An Introduction to Model-Based Cognitive Neuroscience*.

Bostandjiev, S., O'Donovan, J., and Höllerer, T. (2012). TasteWeights: A visual interactive hybrid recommender system. In *Proceedings of the ACM Conference on Recommender Systems*.

Bothell, D. (2017). *ACT-R 7 Reference Manual*. http://act-r.psy.cmu.edu/actr7/reference-manual.pdf.

Brito, M. R., Chavez, E. L., Quiroz, A. J., and Yukich, J. E. (1997). Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection. *Statistics & Probability Letters*, 35(1).

Brooke, J. (1996). SUS – A quick and dirty usability scale. In *Usability Evaluation in Industry*.

Byrne, M. D. and Bovair, S. (1997). A working memory model of a common procedural error. *Cognitive Science*, 21(1).

Card, S. K., Moran, T. P., and Newell, A. (1980). The keystroke-level model for user performance time with interactive systems. *Communications of the ACM*, 23(7).

Chaudhary, A., Szalay, A. S., and Moore, A. W. (2002). Very fast outlier detection in large multidimensional data sets. In *Proceedings of the ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*.

Chen, X., Bailly, G., Brumby, D. P., Oulasvirta, A., and Howes, A. (2015). The emergence of interactive behavior: A model of rational menu search. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*.

Cook, R. and Kay, J. (1994). The justified user model: A viewable, explained user model. In *In Proceedings of the International Conference on User Modeling*.

Desmarais, M. C. and Baker, R. S. (2012). A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction*, 22(1-2).

Doyle, J. C., Francis, B. A., and Tannenbaum, A. R. (2013). *Feedback Control Theory*.

Duric, Z., Gray, W. D., Heishman, R., Li, F., Rosenfeld, A., Schoelles, M. J., Schunn, C., and Wechsler, H. (2002). Integrating perceptual and cognitive modeling for adaptive and intelligent human-computer interaction. *Proceedings of the IEEE*, 90(7).

Dwork, C. and Lei, J. (2009). Differential privacy and robust statistics. In *Proceedings of the ACM Symposium on Theory of Computing*.

Dzindolet, M. T., Peterson, S. A., Pomranky, R. A., Pierce, L. G., and Beck, H. P. (2003). The role of trust in automation reliance. *International Journal of Human-Computer Studies*, 58(6).

Ericsson, K. A. and Lehmann, A. C. (1996). Expert and exceptional performance: Evidence of maximal adaptation to task constraints. *Annual Review of Psychology*, 47(1).

Fails, J. A. and Olsen Jr, D. R. (2003). Interactive machine learning. In *Proceedings of the International Conference on Intelligent User Interfaces*.

Fan, W. (2004). Systematic data selection to mine concept-drifting data streams. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Fiebrink, R., Cook, P. R., and Trueman, D. (2011). Human model evaluation in interactive supervised learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

Fischer, G. (2001). User modeling in human-computer interaction. *User Modeling and User-Adapted Interaction*, 11(1).

Fitts, P. M. and Peterson, J. R. (1964). Information capacity of discrete motor responses. *Journal of Experimental Psychology*, 67(2).

Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The Elements of Statistical Learning*.

Gajos, K. Z., Everitt, K., Tan, D. S., Czerwinski, M., and Weld, D. S. (2008). Predictability and accuracy in adaptive user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

Gama, J., Medas, P., Castillo, G., and Rodrigues, P. (2004). Learning with drift detection. In *Brazilian Symposium on Artificial Intelligence*.

Gershman, S. J., Horvitz, E. J., and Tenenbaum, J. B. (2015). Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. *Science*, 349(6245).

Gigerenzer, G. and Goldstein, D. G. (1996). Reasoning the fast and frugal way: Models of bounded rationality. *Psychological Review*, 103(4).

Gilks, W. R., Richardson, S., and Spiegelhalter, D. (1995). *Markov chain Monte Carlo in practice*.

Głowacka, D., Ruotsalo, T., Konuyshkova, K., Athukorala, K., Kaski, S., and Jacucci, G. (2013). Directing exploratory search: Reinforcement learning from user interactions with keywords. In *Proceedings of the International Conference on Intelligent User Interfaces*.

González, J., Dai, Z., Hennig, P., and Lawrence, N. (2016). Batch Bayesian optimization via local penalization. In *Artificial Intelligence and Statistics*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems*.

Gregor, S. and Benbasat, I. (1999). Explanations from intelligent systems: Theoretical foundations and implications for practice. *MIS Quarterly*, 23(4).

Grotov, A., Chuklin, A., Markov, I., Stout, L., Xumara, F., and de Rijke, M. (2015). A comparative study of click models for web search. In *International Conference of the Cross-Language Evaluation Forum for European Languages*.

Gupta, M., Gao, J., Aggarwal, C. C., and Han, J. (2014). Outlier detection for temporal data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 26(9).

Gutmann, M. U. and Corander, J. (2016). Bayesian optimization for likelihood-free inference of simulator-based statistical models. *Journal of Machine Learning Research*, 17.

Gutmann, M. U., Dutta, R., Kaski, S., and Corander, J. (2014). Likelihood-free inference via classification. *ArXiv e-prints*, arXiv:1407.4981v1.

Gutmann, M. U., Dutta, R., Kaski, S., and Corander, J. (2018). Likelihood-free inference via classification. *Statistics and Computing*, 28(2).

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3.

Guyon, I., Matic, N., and Vapnik, V. (1996). Discovering informative patterns and data cleaning. In *Advances in Knowledge Discovery and Data Mining*.

Helander, M., Landauer, T., and Prabhu, P. (1997). Mental models and user models. In *Handbook of Human-Computer Interaction*.

Hodge, V. and Austin, J. (2004). A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2).

Jacucci, G., Spagnolli, A., Freeman, J., and Gamberini, L. (2014). Symbiotic interaction: A critical definition and comparison to other human-computer paradigms. In *International Workshop on Symbiotic Interaction*.

John, B. E. and Kieras, D. E. (1996). The GOMS family of user interface analysis techniques: Comparison and contrast. *ACM Transactions on Computer-Human Interaction*, 3(4).

Kieras, D. and Polson, P. G. (1985). An approach to the formal analysis of user complexity. *International Journal of Man-Machine Studies*, 22(4).

Kieras, D. E. and Bovair, S. (1984). The role of a mental model in learning to operate a device. *Cognitive Science*, 8(3).

Kitani, K. M., Ziebart, B. D., Bagnell, J. A., and Hebert, M. (2012). Activity forecasting. In *European Conference on Computer Vision*.

Klinkenberg, R. (2004). Learning drifting concepts: Example selection vs. example weighting. *Intelligent Data Analysis*, 8(3).

Koychev, I. (2000). Gradual forgetting for adaptation to concept drift. In *Proceedings of ECAI Workshop on Current Issues in Spatio-Temporal Reasoning*.

Kulesza, T., Burnett, M., Wong, W.-K., and Stumpf, S. (2015). Principles of explanatory debugging to personalize interactive machine learning. In *Proceedings of the International Conference on Intelligent User Interfaces*.

Kulesza, T., Stumpf, S., Burnett, M., and Kwan, I. (2012). Tell me more?: The effects of mental model soundness on personalizing an intelligent agent. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

Lagarias, J. C., Reeds, J. A., Wright, M. H., and Wright, P. E. (1998). Convergence properties of the Nelder-Mead simplex method in low dimensions. *SIAM Journal on Optimization*, 9(1).

Langley, P. (1999). User modeling in adaptive interface. In *Proceedings of the International Conference on User Modeling*.

Laskov, P., Düssel, P., Schäfer, C., and Rieck, K. (2005). Learning intrusion detection: Supervised or unsupervised? In *Proceedings of the International Conference on Image Analysis and Processing*.

Lazer, D., Pentland, A., Adamic, L., Aral, S., Barabasi, A. L., Brewer, D., Christakis, N., Contractor, N., Fowler, J., Gutmann, M., Jebara, T., King, G., Macy, M., Roy, D., and Van Alstyne, M. (2009). Life in the network: The coming age of computational social science. *Science*, 323(5915).

Lee, H. S., Betts, S., and Anderson, J. R. (2016). Learning problem-solving rules as search through a hypothesis space. *Cognitive Science*, 40(5).

Lewis, R. L., Howes, A., and Singh, S. (2014). Computational rationality: Linking mechanism and behavior through bounded utility maximization. *Topics in Cognitive Science*, 6(2).

Lewis, R. L. and Vasishth, S. (2005). An activation-based model of sentence processing as skilled memory retrieval. *Cognitive Science*, 29(3).

Li, Y., Campbell, C., and Tipping, M. (2002). Bayesian automatic relevance determination algorithms for classifying gene expression data. *Bioinformatics*, 18(10).

Lim, B. Y., Dey, A. K., and Avrahami, D. (2009). Why and why not explanations improve the intelligibility of context-aware intelligent systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

Liu, D. C. and Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1).

Logačev, P. and Vasishth, S. (2016). A multiple-channel model of task-dependent ambiguity resolution in sentence comprehension. *Cognitive Science*, 40(2).

Lohse, G. L. (1993). A cognitive model for understanding graphical perception. *Human-Computer Interaction*, 8(4).

Macdonald, N., Frase, L., Gingrich, P., and Keenan, S. (1982). The writer's workbench: Computer aids for text analysis. *IEEE Transactions on Communications*, 30(1).

Marlin, B. M. (2004). Modeling user rating profiles for collaborative filtering. In *Advances in Neural Information Processing Systems*.

Maxwell, J. P., Masters, R. S. W., and Eves, F. F. (2000). From novice to no know-how: A longitudinal study of implicit motor learning. *Journal of Sports Sciences*, 18(2).

Muir, B. M. (1994). Trust in automation: Part I. Theoretical issues in the study of trust and human intervention in automated systems. *Ergonomics*, 37(11).

Muthukrishnan, S. (2005). Data streams: Algorithms and applications. *Foundations and Trends in Theoretical Computer Science*, 1(2).

Norman, D. A. (1983). Some observations on mental models. *Mental Models*, 7(112).

Olson, J. R. and Olson, G. M. (1990). The growth of cognitive modeling in human-computer interaction since GOMS. *Human-Computer Interaction*, 5(2-3).

Pu, P., Chen, L., and Hu, R. (2011). A user-centric evaluation framework for recommender systems. In *Proceedings of the ACM Conference on Recommender Systems*. ACM.

Ramachandran, D. and Amir, E. (2007). Bayesian inverse reinforcement learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*.

Reason, J. (1990). *Human Error*.

Rennie, J. and Lang, K. (2008). The 20 Newsgroups dataset. `http://qwone.com/~jason/20Newsgroups/`.

Ritter, F. E. and Young, R. M. (2001). Embodied models as simulated users: Introduction to this special issue on using cognitive models to improve interface design. *International Journal of Human-Computer Studies*, 55(1).

Rogers, Y., Sharp, H., and Preece, J. (2011). *Interaction Design: Beyond Human-Computer Interaction*.

Ruotsalo, T., Peltonen, J., Eugster, M., Głowacka, D., Konyushkova, K., Athukorala, K., Kosunen, I., Reijonen, A., Myllymäki, P., Jacucci, G., and Kaski, S. (2013). Directing exploratory search with interactive intent modeling. In *Proceedings of the ACM International Conference on Information & Knowledge Management*.

Said, N., Engelhart, M., Kirches, C., Körkel, S., and Holt, D. V. (2016). Applying mathematical optimization methods to an ACT-R instance-based learning model. *PLoS ONE*, 11(7).

Shneiderman, B. (1984). Response time and display rate in human performance with computers. *ACM Computing Surveys*, 16(3).

Simon, H. A. (1990). Bounded rationality. In *Utility and Probability*.

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*.

Srinivas, N., Krause, A., Seeger, M., and Kakade, S. M. (2010). Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the International Conference on Machine Learning*.

Suermondt, H. J. and Cooper, G. F. (1993). An evaluation of explanations of probabilistic inference. *Computers and Biomedical Research*, 26(3).

Sunnåker, M., Busetto, A. G., Numminen, E., Corander, J., Foll, M., and Dessimoz, C. (2013). Approximate Bayesian computation. *PLoS Computational Biology*, 9(1).

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*.

Teevan, J., Alvarado, C., Ackerman, M. S., and Karger, D. R. (2004). The perfect search engine is not enough: A study of orienteering behavior in directed search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

Tenison, C., Fincham, J. M., and Anderson, J. R. (2016). Phases of learning: How skill acquisition impacts cognitive processing. *Cognitive Psychology*, 87.

Ting, J.-A., D'Souza, A., and Schaal, S. (2007). Automatic outlier detection: A Bayesian approach. In *Proceedings of the IEEE International Conference on Robotics and Automation*.

Tipping, M. E. and Lawrence, N. D. (2003). A variational approach to robust Bayesian interpolation. In *Proceeding of the IEEE Workshop on Neural Networks for Signal Processing*.

Tsandilas, T. and Schraefel, M. C. (2004). Usable adaptive hypermedia systems. *New Review of Hypermedia and Multimedia*, 10(1).

Tsymbal, A. (2004). The problem of concept drift: Definitions and related work. Technical report, Computer Science Department, Trinity College Dublin.

Vandekerckhove, J. and Tuerlinckx, F. (2007). Fitting the Ratcliff diffusion model to experimental data. *Psychonomic Bulletin & Review*, 14(6).

Vellido, A., Martín-Guerrero, J. D., and Lisboa, P. J. (2012). Making machine learning models interpretable. In *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*.

Wang, Y., Kucukelbir, A., and Blei, D. M. (2017). Robust probabilistic modeling with Bayesian data reweighting. In *Proceedings of the International Conference on Machine Learning*.

Wang, Z., Hutter, F., Zoghi, M., Matheson, D., and de Feitas, N. (2016). Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55.

Ward, M. O., Grinstein, G., and Keim, D. (2010). *Interactive Data Visualization: Foundations, Techniques, and Applications*.

Ware, C. (2012). *Information Visualization: Perception for Design*.

Webb, G. I., Pazzani, M. J., and Billsus, D. (2001). Machine learning for user modeling. *User Modeling and User-Adapted Interaction*, 11(1).

Widmer, G. and Kubat, M. (1996). Learning in the presence of concept drift and hidden contexts. *Machine Learning*, 23(1).

Wilkinson, L., Anand, A., and Grossman, R. (2006). High-dimensional visual analytics: Interactive exploration guided by pairwise views of point distributions. *IEEE Transactions on Visualization and Computer Graphics*, 12(6).

Wilson, S., Johnson, P., Kelly, C., Cunningham, J., and Markopoulos, P. (1993). Beyond hacking: A model based approach to user interface design. In *Proceedings of the British Conference on Human-Computer Interaction*.

Zhang, D., Gatica-Perez, D., Bengio, S., and McCowan, I. (2005). Semi-supervised adapted HMMs for unusual event detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Zhu, X. and Wu, X. (2004). Class noise vs. attribute noise: A quantitative study. *Artificial Intelligence Review*, 22(3).

Ziebart, B. D., Maas, A. L., Bagnell, J. A., and Dey, A. K. (2008). Maximum entropy inverse reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Machine learning and human-computer interaction share an interesting boundary regarding user modelling and usability of machine learning powered systems.

Methods introduced in this thesis enable inference of user's goals and capabilities based on incomplete observation data, and improve the usability of systems that perform online modelling of the user's interests. These methods pave the way for more fluent and productive interaction between humans and intelligent systems.

Aalto-DD 206/2018

9789526082509

ISBN 978-952-60-8250-9 (printed)
ISBN 978-952-60-8251-6 (pdf)
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)

Aalto University
School of Science
Department of Computer Science
www.aalto.fi

BUSINESS +
ECONOMY

ART +
DESIGN +
ARCHITECTURE

SCIENCE +
TECHNOLOGY

CROSSOVER

DOCTORAL
DISSERTATIONS