Abstract

Ongoing title:
1. Probabilistic user modelling for improving human-in-the-loop machine learning
2. Probabilistic user modelling methods for improving human-in-the-loop machine learning
3. Interactive user modelling for human-in-the-loop machine learning

Abstract

En puhu suomea.

Preface

To fill.

Espoo, November 12, 2019,

Pedram Daee

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 Pedram Daee, Joel Pyykkö, Dorota Głowacka, and Samuel Kaski. Interactive Intent Modeling from Multiple Feedback Domains. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*, Sonoma, California, USA, 71–75, March 2016.

II Pedram Daee*, Tomi Peltola*, Marta Soare*, and Samuel Kaski. Knowledge elicitation via sequential probabilistic inference for high-dimensional prediction. *Machine Learning*, 106, 9-10, 1599–1620, 2017.

III Iiris Sundin*, Tomi Peltola*, Luana Micallef, Homayun Afrabandpey, Marta Soare, Muntasir Mamun Majumder, Pedram Daee, Chen He, Baris Serim, Aki Havulinna, Caroline Heckman, Giulio Jacucci, Pekka Marttinen, and Samuel Kaski. Improving genomics-based predictions for precision medicine through active elicitation of expert knowledge. *Bioinformatics*, 34, 13, i395–i403, 2018.

IV Pedram Daee*, Tomi Peltola*, Aki Vehtari, and Samuel Kaski. User Modelling for Avoiding Overfitting in Interactive Knowledge Elicitation for Prediction. In *Proceedings of the 23rd International Conference on Intelligent User Interfaces*, Tokyo, Japan, 305–310, March 2018.

V Giulio Jacucci, Oswald Barral, Pedram Daee, Markus Wenzel, Baris Serim, Tuukka Ruotsalo, Patrik Pluchino, Jonathan Freeman, Luciano Gamberini, Samuel Kaski, Benjamin Blankertz. Integrating Neurophysiological Relevance Feedback in Intent Modeling for Information Retrieval. *Journal of the Association for Information Science and Technology*, 2019.

Author's Contribution

Publication I: "Interactive Intent Modeling from Multiple Feedback Domains"

The author had the main responsiblity in problem formulation and modeling. The author designed and implemented the simulation experiment. Joel Pyykkö and the author built the system for user studies and conducted them together. The author wrote the initial draft of the manuscript, after which all co-authors joined for revisions.

Publication II: "Knowledge elicitation via sequential probabilistic inference for high-dimensional prediction"

The ideas and experiments in this article were designed jointly (the first three authors contributed equally). The author had the main responsibility in the derivation of the sequential experimental design and implementation of the experiments. Dr. Tomi Peltola derived and implemented the posterior approximation. The manuscript was written jointly.

Publication III: "Improving genomics-based predictions for precision medicine through active elicitation of expert knowledge"

The author contributed on formulation of the sequential experimental design and implementation of a portion of the early version of the experiments. The author made comments to the manuscript in preparation.

Publication IV: "User Modelling for Avoiding Overfitting in Interactive Knowledge Elicitation for Prediction"

The ideas and experiments in this article were designed jointly (the first two authors contributed equally). The author designed and implemented the user study. Dr. Tomi Peltola had the main responsibility of the model formulation. The first two authors wrote the initial draft of the manuscript, after which all co-authors joined for revisions.

Publication V: "Integrating Neurophysiological Relevance Feedback in Intent Modeling for Information Retrieval"

The author had the main responsibility in design and implementation of the interactive intent modelling and information retrieval system, and writing of the correspnding sections. All the authors contributed to paper revisions.

1. Introduction

Whether it is an everyday user searching for an application in her mobile phone or a doctor working with a cancer diagnostic system, humans and machines are increasingly interacting with each other [4]. The aim of this thesis is to improve this interaction by incorporating a probabilistic model of the human user in the system they are interacting with. In particular, the thesis considers the family of problems where the human and machine interact to solve a prediction problem. Such problems can include personalized search activity or medical prediction about the response of a cancer drug. An important common factor in these scenarios is that the number of labeled data (known as training data) that the machine can use to make predictions, is usually very few compared to the dimension of search space. This is known as the "small n, large p" problem (n referring to the number of available labeled data and p the size of the dimension) which results in ill-posed statistical learning since there are limits in how low in sample size statistical methods can go [19].

There are different ways to handle the limited labeled data challenge. The most direct solution is to provide more labeled data for the model. However, due to the high cost (e.g., experimenting the drug response on a new patient in drug response prediction task) or the nature of the data provider (e.g., reluctance of users to provide many feedbacks about their interest in personalized search systems), adding more training data data may not always be possible. An alternative direction is to restrict the model family to a simpler family of solutions (for example by assuming linearity and/or sparsity [56]) to prevent the model from overfitting to the few available labeled data. A less explored direction for tackling the limited data challenge particularly in human-in-the-loop systems is to make prior assumptions about how user interaction with the system is generated and exploit this knowledge to extract more information from limited interaction. This approach is known as user modelling in human–computer interaction (see for example [21]) and it broadly studies ways to improve usability and usefulness of human–computer collaboration.

1.1 Research questions and contributions

This thesis investigates methods to tackle the limited user interaction challenge in interactive machine learning for prediction. The thesis focuses on scenarios where there is few labeled data available compared to the dimension of the problem, or when a human user is provider of the labeled data. The core idea of the thesis is to jointly model the human user, as a probabilistic user model, with the data as part of a unified probabilistic model and then perform sequential probabilistic inference on the joint model to design improved interaction. The thesis focuses on the following research questions derived from the core idea mentioned above:

RQ1 – *Can we exploit new sources of interaction as additional learning signals from human user to improve interactive intent modelling?*

Publications I and V contribute to this research question by proposing models to incorporate new types of user feedback to amend the limited feedback in exploratory information seeking tasks. The focused task is an article search scenario where a user needs to sequentially provide relevance feedback to suggested keywords in order to find the targeted article. This is modelled as a multi-armed bandit problem with the goal of finding the most relevant article with minimum interaction. In particular, Publication I couples user relevance feedback on both articles and keywords by assuming a shared underlying latent intent model connected through a probabilistic model of the relationship between keywords and articles. Thompson sampling on the posterior of the latent intent was then used to recommend new articles and keywords in each iteration. Publication V investigates the use of implicit relevance feedback from neurophysiology signals for effortless information seeking. The work contributes by demonstrating how to integrate this inherently noisy and implicit feedback source with scarce explicit interaction. A model for controlling the accuracy of the feedback given its nature (implicit or explicit) was introduced. Similar to Publication I, Thompson sampling was used to control the exploration and exploitation balance of the recommendations. Both publications were evaluated by user studies in realistic information seeking tasks.

RQ2 – *Can expert knowledge about high dimensional data models be elicited to improve the prediction performance?*

Publications II and III contribute to this research question. Publication II proposes a framework for user knowledge elicitation as a probabilistic inference process, where the user knowledge is sequentially queried to improve predictions on a "small n, large p" problem. In particular, sparse linear regression is considered as the data model with access to only few high-dimensional training data. It is assumed that there are experts who have knowledge about the relevance of the covariates, or of values of the regression coefficients and can provide this information to the data model if queried. The work contributes by an algorithm and computational approximation for fast and efficient interac-

tion, which sequentially identifies the most informative queries to ask from the user. Publication III, builds on Publication II by adding user knowledge about direction of relevance of covariates and applying the method in important application of precision medicine to predict the effects of different treatments using high-dimensional genomic measurements. Both publications were evaluated by extensive simulations and user studies. Source codes for methods presented in Publications II and III, and user study data from Publication II are available at `https://github.com/HIIT/knowledge-elicitation-for-linear-regression` and `https://github.com/AaltoPML/knowledge-elicitation-for-precision-medicine`.

RQ3 – *Is it enough to incorporate human knowledge directly in the data model as explained in RQ2, or could it be beneficial to account for rational knowledge updates that humans may undergo during the interaction?*

Publication IV contributes to this research question by modelling the knowledge provider, here the human user, as a rational agent that updates its knowledge about the underlying prediction task during the interaction. In particular, certain aspects of training data may be revealed (for example through visualizations) to the user during knowledge elicitation. The elicited knowledge from the user may then be, to some extent, dependent to the knowledge coming from the training data. This redundant knowledge can result in overfitting if the user reinforces noisy patterns in the data, since the model may not account for the dependencies between training data and user feedback. We propose a user modelling methodology that assumes that the observed user feedback is an outcome of the user's rational knowledge update. The user model can then perform the reverse of the update to extract user's tacit knowledge and then update the model. The proposed user modelling idea was evaluated in a user study. Source code and user study data are available at `https://github.com/HIIT/human-overfitting-in-IML`.

1.2 Organization of the thesis

The organization of the thesis is as follows. Chapter 2 provides an overview of probabilistic modelling and introduces our approach of modelling the user and data as a joint probabilistic model. Chapter three investigates the design of interactions and reviews different utility functions for selecting the most informative query to be asked from the user. The fourth chapter revisits the research questions and summarizes Publications I-IV and discuses the future directions for probabilistic user modelling.

2. Probabilistic modelling of data and user

This chapter provides a brief introduction to probabilistic modelling as the main statistical framework that is used through the thesis. After some preliminaries, Section 2.2 reviews the type of linear models that are used in Publication I-V for prediction. Section 2.3 introduces different types of user interaction with the linear model and explains how user knowledge about the model can be incorporated as observational feedback. The computational solutions for Bayesian inference are reviewed in 2.4.

2.1 Preliminaries

The core idea of probabilistic modelling is to describe all unobserved parameters and observed data as random variables from probability distributions. The unobserved parameters include the unknown quantity of interest or other parameters that affect the data or the quantity of interest. Bayesian inference provides a powerful framework to fit the described probabilistic model to observational data [23]. A core feature of Bayesian inference is that it provides probability distributions as the solution, compared to deterministic methods which provide a single outcome. This uncertainty quantification is of high interest in cases where few observational data are available or when the data acquisition scheme is controlled by the model. Both of these constrains are prominently present in the tasks investigated by this thesis.

We follow the notation of [23] and use $p(.)$ to denote a probability distribution and $p(. \mid .)$ a conditional distribution. Consider the case where there are a set of observations $\mathscr{D} = \{y_1, \ldots, y_n\}$ generated from a probabilistic model with an unobserved parameter of interest θ. The observational model for an observation y describes the conditional density of y given the parameter θ and is denoted as $y \sim p(y \mid \theta)$. It is usually assumed that the observations are conditionally independent given θ, enabling us to write the model for all observations as $p(\mathscr{D} \mid \theta) = \prod_{i=1}^{n} p(y_i \mid \theta)$. The observational model is called likelihood function if perceived as a function of θ with fixed observation y. One of the core questions in statistical inference is to estimate the parameter of interest θ based on

observations $\mathscr{D}$. Bayesian inference answers this question by computing the conditional distribution of θ given $\mathscr{D}$ following the Bayes rule

$$p(\theta \mid \mathscr{D}) = \frac{p(\mathscr{D} \mid \theta)p(\theta)}{p(\mathscr{D})}. \tag{2.1}$$

Where $p(\theta)$ represents the prior belief about θ and $p(\mathscr{D})$ is called the marginal likelihood and acts as a normalization factor as it does not depend on θ. The marginal likelihood can be computed using the marginalization rule, i.e., $p(\mathscr{D}) = \int_\theta p(\mathscr{D}, \theta) d\theta = \int_\theta p(\mathscr{D} \mid \theta) p(\theta) d\theta$ [1]. The conditional distribution $p(\theta \mid \mathscr{D})$ is called the posterior and it expresses the uncertainty surrounding the true value of θ, after updating the prior assumptions about θ (represented by $p(\theta)$) with the knowledge coming from the observations (represented by $p(\mathscr{D} \mid \theta)$).

In many cases, we may be more interested to make a prediction about an unknown observable data point $\tilde{y}$ rather than the parameter θ. Bayesian inference allows us to compute the conditional distribution of this unknown observable data given the observed data points by averaging over the posterior:

$$p(\tilde{y} \mid \mathscr{D}) = \int_\theta p(\tilde{y} \mid \theta) p(\theta \mid \mathscr{D}) d\theta. \tag{2.2}$$

$p(\tilde{y} \mid \mathscr{D})$ is known as the posterior predictive distribution and represents the uncertainty about a potential new observation. An example usage of this distribution could be when we want to predict the value of a test data. Since test data is unobserved, we can use posterior predictive distribution as our best guess. However, in many applications, only a value (and not a distribution) is required as the prediction. This can be handled by using some statistics of the distribution (for example mean, mode, or median) as the prediction. Still, the quantified uncertainty in the distribution can be useful as it provides knowledge about how certain we are about our estimate. In particular, this uncertainty can help us to design more efficient interaction with a user, as will be discussed in Section 3.

2.2 Modelling data for prediction

Prediction is one of the core problems in statistical analysis and supervised machine learning. Given a set of n input and output pairs, called training data, denoted by $\mathscr{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, the goal is to find a mapping from inputs to outputs. Here, $\boldsymbol{x}_i = [x_i^1, \ldots, x_i^d]^\top \in \mathbb{R}^d$ is a d-dimensional column vector representing the values of i^{th} data point[2]. The dimensions are commonly called feature, covariates, or attribute. The corresponding response (or target) variable to $\boldsymbol{x}_i$ is denoted by y_i which may take different forms depending on the underlying

[1] The integral turns to summation for discrete θ.

[2] We generally use bold font to refer to vectors, subscripts to index an specific item (e.g, one particular observation out of several), and superscripts to refer to dimensions of a variable.

problem. In this thesis, we consider the regression tasks, meaning that we model response variables by real values, i.e., $y_i \in \mathbb{R}$. The problem is called classification in supervised learning if the response variable is restricted to a set of discrete classes represented by a categorical variable.

2.2.1 Bayesian linear regression

A well-studied and widely practical type of regression, known as linear regression, assumes that the relationship between all inputs and their corresponding response variables is linear. This relationship can be described as

$$y_i = \sum_{j=1}^{d} x_i^j w^j + \epsilon_i = \boldsymbol{x}_i^\top \boldsymbol{w} + \epsilon_i, \qquad i = 1, \ldots, n,$$

where $\boldsymbol{w} \in \mathbb{R}^d$ is the regression coefficients or the model's weights and ϵ_i is the residual error between linear prediction $\boldsymbol{x}_i^\top \boldsymbol{w}$ and the response value y_i. Given the labeled data $\mathscr{D}$, a commonly used error function to measure the goodness of a weight vector $\boldsymbol{w}$, is the sum of squared of residual errors $\sum_{i=1}^{n} (\boldsymbol{x}_i^\top \boldsymbol{w} - y_i)^2$. By stacking the inputs in $\boldsymbol{X} = [x_1, \ldots, x_n]^\top \in \mathbb{R}^{n \times d}$ and outputs in $\boldsymbol{y} = [y_1, \ldots, y_n]^\top \in \mathbb{R}^n$ the error can be expressed in vector format as $(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})^\top (\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})$. The frequentist approach directly finds a point estimate for $\boldsymbol{w}$ that minimizes this error (also known as least squares solution). It is straightforward to show that $\hat{\boldsymbol{w}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}$ would be the point estimate solution given that $(\boldsymbol{X}^\top \boldsymbol{X})^{-1}$ is invertible.

However, as mentioned, we are interested in directly quantifying the uncertainty of the solution. The probabilistic way to model this problem is to explicitly describe the model assumptions (likelihood and priors) as probability distributions. In linear regression, the residual errors ϵ_i and the weights $\boldsymbol{w}$ are modelled as random variables and the inputs $\boldsymbol{x}_i$ as vector of values that are given. A customary assumption is to model the residual errors as independent zero-mean Gaussian random variables $\epsilon_i \sim \mathrm{N}(0, \sigma^2)$, where σ^2 is the variance of the Gaussian distribution which indicates the model tolerance about residual errors. It is common to also model σ^2 as another random variable with its own distribution assumption, however, for now we consider it to be a fixed hyperparameter for simplicity. The unknown quantity of interest in the linear model is the regression coefficients. To complete the Bayesian inference loop, we need to consider a prior distribution on $\boldsymbol{w}$. There are many ways to do this depending on the underlying task. One simple prior could be to assume that coefficients are independent and each come from a zero-mean Gaussian distribution, encouraging the weights to be close to zero. This simple Bayesian linear regression can be described by

$$\begin{aligned} \boldsymbol{y} &\sim \mathrm{N}(\boldsymbol{X}\boldsymbol{w}, \sigma^2 \mathbf{I}), \\ \boldsymbol{w} &\sim \mathrm{N}(\mathbf{0}, \tau^2 \mathbf{I}), \end{aligned} \tag{2.3}$$

where τ^2 is the variance of the weights and $\mathbf{I}$ is the identity matrix. For simplicity we assume that τ^2 is also a fixed hyperparameter. Therefore, The only unobserved parameters of this model is $\boldsymbol{w}$. The Bayesian rule allows us to derive the posterior for $\boldsymbol{w}$ as[3]

$$p(\boldsymbol{w} \mid \mathscr{D}) = \frac{p(\mathscr{D} \mid \boldsymbol{w})p(\boldsymbol{w})}{p(\mathscr{D})} = \frac{\mathrm{N}(\boldsymbol{y} \mid \boldsymbol{X}\boldsymbol{w}, \sigma^2\mathbf{I})\mathrm{N}(\boldsymbol{w} \mid 0, \tau^2\mathbf{I}))}{p(\mathscr{D})}. \tag{2.4}$$

Generally, the posterior in many problems cannot be analytically derived (we will discuss this issue more in Section 2.4). For this simple model, however, an analytical solution is available. We will go through the steps as an exercise with posterior inference. Before starting, it would be useful to note that a multivariate Gaussian distribution can be expressed by its mean vector $\boldsymbol{\mu} \in \mathbb{R}^d$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$ as

$$\begin{aligned} \mathrm{N}(\boldsymbol{w} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) &\propto \exp\left((\boldsymbol{w} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{w} - \boldsymbol{\mu})\right) \\ &\propto \exp\left(\boldsymbol{w}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{w} - 2\boldsymbol{w}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\right), \end{aligned} \tag{2.5}$$

after dropping all the terms that are constant with respect to $\boldsymbol{w}$.

To derive the posterior, we follow Equation 2.4

$$\begin{aligned} p(\boldsymbol{w} \mid \mathscr{D}) &\propto \exp\left(\frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})\right)\exp(\frac{1}{2\tau^2}\boldsymbol{w}^\top\boldsymbol{w}) \\ &\propto \exp\left(\sigma^{-2}(\boldsymbol{y}^\top\boldsymbol{y} - 2\boldsymbol{w}^\top\boldsymbol{X}^\top\boldsymbol{y} + \boldsymbol{w}^\top\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{w}) + \tau^{-2}\boldsymbol{w}^\top\boldsymbol{w}\right) \\ &\propto \exp\left(-2\sigma^{-2}\boldsymbol{w}^\top\boldsymbol{X}^\top\boldsymbol{y} + \boldsymbol{w}^\top(\sigma^{-2}\boldsymbol{X}^\top\boldsymbol{X} + \tau^{-2}\mathbf{I})\boldsymbol{w}\right) \end{aligned} \tag{2.6}$$

where we dropped $p(\mathscr{D})$ and all the other terms which where constant with respect to $\boldsymbol{w}$. Equation 2.6 has the same form to Equation 2.5 (with respect to $\boldsymbol{w}$) and therefore is proportional to a multivariate Gaussian distribution. This relation shows that the posterior should be multivariate Gaussian, as both equations should integrate to one, and thus its parameters can be found by matching the terms of the two equations as $\boldsymbol{\Sigma}^{-1} = (\sigma^{-2}\boldsymbol{X}^\top\boldsymbol{X} + \tau^{-2}\mathbf{I})$ and $\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} = \sigma^{-2}\boldsymbol{X}^\top\boldsymbol{y}$. Finally, the posterior of our simple linear regression can be described as

$$\begin{aligned} p(\boldsymbol{w} \mid \mathscr{D}) &= \mathrm{N}(\boldsymbol{w} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}), \text{where} \\ \boldsymbol{\Sigma}^{-1} &= (\sigma^{-2}\boldsymbol{X}^\top\boldsymbol{X} + \tau^{-2}\mathbf{I}), \\ \boldsymbol{\mu} &= \boldsymbol{\Sigma}\sigma^{-2}\boldsymbol{X}^\top\boldsymbol{y}. \end{aligned} \tag{2.7}$$

This simple Bayesian regression is used as the underlying data model in Publication I. It would be interesting to compare a point estimate of the posterior

[3] We generally use the notation $\boldsymbol{w} \sim \mathrm{N}(0, \tau^2\mathbf{I})$ to denote the random variable $\boldsymbol{w}$ and $\mathrm{N}(\boldsymbol{w} \mid 0, \tau^2\mathbf{I})$ to refer to the density function.

distribution with the frequentist solution $\hat{\boldsymbol{w}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{y}$. Maximum a posteriori (MAP) is a point estimate which represents the value in the posterior that has the highest density. for multivariate Gaussian distributions, this would be equal to the mean, which after some rearrangement, can be described as $\hat{\boldsymbol{w}}_{MAP} = (\boldsymbol{X}^\top \boldsymbol{X} + \frac{\sigma^2}{\tau^2}\mathbf{I})^{-1}(\boldsymbol{X}^\top \boldsymbol{y})$. Comparing the mean of the posterior with the frequentist solution indicates that the two estimates are very similar with the only difference that the MAP estimate has the additive term $\frac{\sigma^2}{\tau^2}\mathbf{I}$ in the covariance matrix. This added term is a direct outcome of our assumptions for the likelihood and prior of the model which were not present in our simple frequentist model. This term can also be interpreted as a regularization parameter that controls the variance of weights. The designer of the model may want to consider different types of regularization or constrains in the model, which in probabilistic modelling is usually done by incorporating them in the prior distribution. Such problems may not have an analytical posterior available. We will consider two more complex models in the following subsections.

Sparsity-inducing priors

In many prediction problems, in particular those that require human intervention in a form, the number of available training data is smaller than dimensions of the problem. If not regularized properly, a model trained in this setting may overfit to the training data (achieving very low training error) while not being able to generalize properly to unobserved data (high test data error). One way to tackle this challenge is to directly regularize the model parameters so that they would not have the flexibility to overfit to the observed data. This regularization can be done by adding penalty terms, for example l_1 norms of the weights, to the error function in the non-Bayesian models with the general idea of pushing weights that are not useful toward zero (see for example Lasso [56]). In probabilistic modelling, we can achieve the same goal by selecting sparsity-inducing priors.

There are different priors that encourage sparsity but they can be categorized to two general groups of mixture priors, such as spike-and-slab prior [24], and the continuous shrinkage priors, such as horseshoe [43] and Laplace [50] priors. The core idea of these priors is that their densities for coefficients peak at zero but at the same time have good amount of probability mass in non-zero values. In this thesis we consider the spike-and-slab prior as it introduces a binary latent variable that explicitly indicates whether a coefficient should be zero or non-zero (relevant and not-relevant groups). This explicit explanation could be very useful as it is compatible with human opinion about inclusion/exclusion of variables in a model (we will discuss this link in detail in Section 2.3).

The linear regression model with sparsity-inducing spike-and-slab prior on coefficients $\boldsymbol{w}$ can be described as

$$\boldsymbol{y} \sim \mathrm{N}(\boldsymbol{X}\boldsymbol{w}, \sigma^2\mathbf{I}), \tag{2.8}$$

$$
\begin{aligned}
\sigma^{-2} &\sim \text{Gamma}(\alpha_\sigma, \beta_\sigma), \\
w^j &\sim \gamma^j \,\text{N}(0, \tau^2) + (1-\gamma^j)\delta_0, && j = 1, \ldots, d, \\
\gamma^j &\sim \text{Bernoulli}(\rho), && j = 1, \ldots, d,
\end{aligned}
$$

where δ_0 is a Dirac delta point mass at zero, and γ^j is the binary variable that indicates whether the corresponding coefficient w^j should be excluded from the model (if $\gamma^j = 0$, then $w^j = 0$) or should it be addressed as a normal variable (if $\gamma^j = 1$, then $w^j \sim \text{N}(0, \tau^2)$). As we are also interested in the behavior of γ^js, we considered a Bernoulli prior distribution on all of them with the prior inclusion probability ρ as a fixed hyperparameter that controls the expected number of non-zero covariates. Unlike the simple model introduced before, here we considered σ^2 as an unknown parameter and assumed a prior distribution for it with α_σ and β_σ as fixed hyperparameters. for the three unobserved parameters $\boldsymbol{w}$, $\boldsymbol{\gamma}$ (as a vector of all γ^js), and σ^2 the posterior can be derived following the Bayes rule

$$
p(\boldsymbol{w}, \boldsymbol{\gamma}, \sigma^2 \mid \mathscr{D}) = \frac{p(\mathscr{D} \mid \boldsymbol{w}, \sigma^2) p(\boldsymbol{w} \mid \boldsymbol{\gamma}) p(\sigma^2) p(\boldsymbol{\gamma})}{p(\mathscr{D})}. \tag{2.9}
$$

The posterior does not have an analytical solution. We will discuss different methods to compute the posterior of these types of problems in Section 2.4. The introduced spike-and-slab Sparsity-inducing model is used as the underlying data model in Publication II, III, and IV.

Detecting outliers

In regression, there may be observations that have response values substantially different from all other data. These highly noisy observations, called outliers, can considerably affect the results if not accounted properly. A Bayesian way to handle this is to consider different variance for residual error of each observation [57, 31]. This can be implemented by changing the observational model from $y_i \sim \text{N}(\boldsymbol{x}_i^\top \boldsymbol{w}, \sigma^2)$, where there was a global parameter σ^2 for the variance of all observations, to $y_i \sim \text{N}(\boldsymbol{x}_i^\top \boldsymbol{w}, \frac{\sigma^2}{\nu_i})$, where the new parameter ν_i controls the noise variance per observation. As ν_i is an unobserved parameter, we consider a prior distribution for it to complete the Bayesian loop:

$$
\begin{aligned}
y_i &\sim \text{N}(\boldsymbol{x}_i^\top \boldsymbol{w}, \frac{\sigma^2}{\nu_i}), && i = 1, \ldots, n, \\
\nu_i &\sim \text{Gamma}(\alpha_\nu, \beta_\nu), && i = 1, \ldots, n, \\
\sigma^{-2} &\sim \text{Gamma}(\alpha_\sigma, \beta_\sigma), \\
\boldsymbol{w} &\sim \text{N}(\boldsymbol{0}, \tau^2 \mathbf{I}),
\end{aligned} \tag{2.10}
$$

where α_ν, β_ν, α_σ, β_σ, and τ^2 are fixed hyperparameters and n is the number of observations. For the unobserved parameters $\boldsymbol{w}$, σ^2, and $\boldsymbol{\nu}$ (as a vector of all

ν_is), the posterior does not have an analytical solution and can be written as

$$p(\boldsymbol{w}, \boldsymbol{\nu}, \sigma^2 \mid \mathscr{D}) = \frac{p(\mathscr{D} \mid \boldsymbol{w}, \sigma^2, \boldsymbol{\nu}) p(\boldsymbol{w}) p(\sigma^2) p(\boldsymbol{\nu})}{p(\mathscr{D})}. \tag{2.11}$$

Noisy observations can particularly happen if the data provider is a noisy source. For example in Publication V, we considered the setting where some of the response variables were generated from noisy physiological signals of a human user. This model was employed to handle the potential outliers in the data.

2.3 Modelling user interaction for prediction

A core idea in the thesis is to model user interaction with the probabilistic model as observational feedback. These feedbacks can be tied to the model parameters via conditional distributions (feedback models). The posterior inference can then learn the unobserved parameters given data observations (as introduced in the previous section) and the feedback observations (which will be discussed in this section). The feedback models are used to incorporate the user knowledge into the regression models introduced in the previous section. The following subsections explain the proposed feedback models in Publication I-V and their relations to the probabilistic models.

2.3.1 User feedback as observation about model parameters

Publication II and III target the case where the user has knowledge about the regression parameters and can provide feedback if queried. As mentioned, we consider high dimensional scenarios where few data points are available. An example task could be a drug response prediction given genomic features of patients [5]. The genomic feature size may exceed thousands of variables, but only a small number of patient data might be available and it may not be possible (or too expensive) to add new data, which make the prediction task challenging. Fortunately, experts in the field may have experience or knowledge from literature about which features, and in what ways, are relevant for this prediction task. If accounted properly, the expert knowledge can help the prediction performance. Another potential scenario is the sentiment prediction problem [35] when only few texts (e.g., textual reviews of items) with known sentiments (e.g., score of the review) are available. A classical representation of textual data known as bag-of-words, considers each data as a vector of distinct words where the feature value indicates the number of appearance of each word in the text[4]. Humans have intuitions about how individual words may be related to the sentiment (e.g., appearance of the word "awful" in a review may

[4]It is also common to represent textual data in dense forms such as [17]. We use bag-of-words due to its simplicity and interpretability of the features.

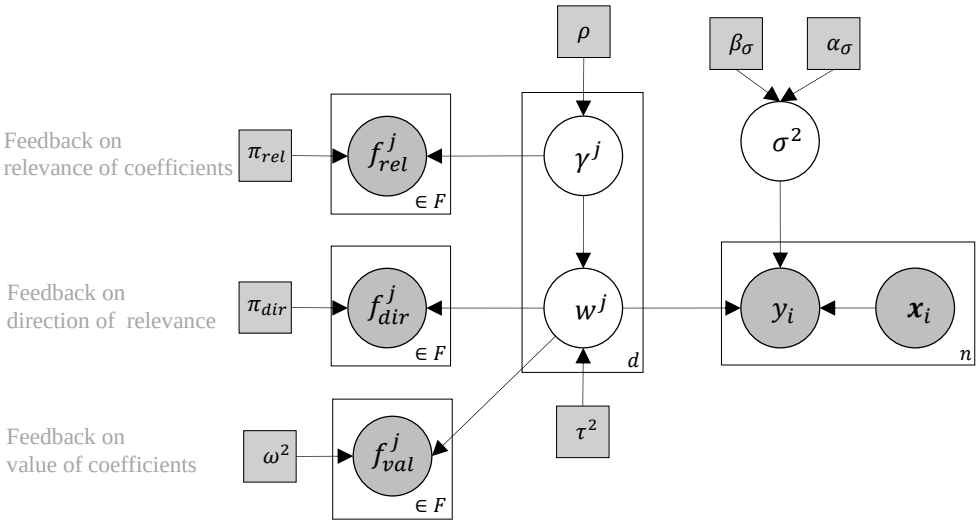

Figure 2.1. Plate notation of the prediction model (right; see Equation 2.8) and user feedback observations (left; see Equations 2.12, 2.13, and 2.14). User feedback influences the data model through observations about model parameters. Circles represent variables (observed variables are shaded), and shaded squares are the fixed hyperparameters.

indicate low review score). Our goal is to design probabilistic models that can receive these types of expert knowledge, alongside the training data, to boost the prediction performance.

We consider the linear regression model with sparsity-inducing spike-and-slab prior on coefficients introduced in Equation 2.8 as the underlying data model. The type of feedback naturally depends on the task and availability of user knowledge. We consider three simple and natural types of user interaction with the model:

- With some noise, the user can provide feedback about the value of coefficients

$$f^j_{val} \sim \mathrm{N}(w^j, \omega^2), \tag{2.12}$$

  where ω^2 models the variance of error in user feedback. Knowledge about value of coefficients is very powerful, however, only available in some specific applications (e.g., exploiting similar trained models or reported coefficient values in related literature).

- With some probability, the user can provide feedback about whether a coefficient should be included or excluded from the model

$$f^j_{rel} \sim \gamma^j \,\mathrm{Bernoulli}(\pi_{rel}) + (1-\gamma^j), \mathrm{Bernoulli}(1-\pi_{rel}). \tag{2.13}$$

  where π_{rel} indicates the probability that user is correct about the feedback. This relevant $f^j_{rel} = 1$ (or not-relevant $f^j_{rel} = 0$) feedback is the simplest form of knowledge that a user may have about a regression task.

- With some probability, the user can provide feedback about the direction of relevance of a coefficient, i.e., whether a feature is positively or negatively correlated with the response variable

$$f_{dir}^{j} \sim I(w^{j} \geq 0)\,\text{Bernoulli}(\pi_{dir}) + I(w^{j} < 0)\,\text{Bernoulli}(1-\pi_{dir}), \tag{2.14}$$

  where π_{dir} indicates the probability that user is correct about the feedback. Feedback about positive correlation of feature j is coded as $f_{dir}^{j} = 1$ and negative correlation as $f_{dir}^{j} = 0$.

The connection between the three mentioned feedback models with the data model (Equation 2.8) is shown as a plate diagram in Figure 2.1. The full posterior of the unknown parameters given data observations and the user feedback can be described as

$$p(\boldsymbol{w}, \boldsymbol{\gamma}, \sigma^2 \mid \mathscr{D}, \mathscr{F}) = \frac{p(\mathscr{D} \mid \boldsymbol{w}, \sigma^2)\, p(F_{val} \mid \boldsymbol{w})\, p(F_{rel} \mid \boldsymbol{\gamma})\, p(F_{dir} \mid \boldsymbol{w})\, p(\boldsymbol{w} \mid \boldsymbol{\gamma})\, p(\sigma^2)\, p(\boldsymbol{\gamma})}{p(\mathscr{D}, \mathscr{F})}, \tag{2.15}$$

where F_{val}, F_{rel}, and F_{dir} are the sets of collected feedback corresponding to the three considered feedback types and $\mathscr{F} = (F_{val}, F_{rel}, F_{dir})$. The posterior computation is discussed in Section 2.4.

Related works

The proposed feedback models provide an intuitive and effortless way for the user to directly influence the prediction model without caring about the complications of the data model. Classical prior elicitation (see for example [40, 22]) aims at eliciting a distribution to represent the expert's knowledge by asking about summary information such as quantiles of the parameters. This is done through iterations between the expert in the related field and statisticians who design the model. Our work goes beyond pure elicitation as it directly connects the expert to the probabilistic model, without the need to the statistician. Furthermore, By exploiting the training data and available feedback, the probabilistic model can facilitate the interaction with the expert by asking the most important questions first (interaction design will be discussed in Section 3).

We have considered three intuitive types of feedback models. Studies have shown that the type of domain knowledge in prediction tasks can be summarized in a small set [15]. A different type of feedback is the case where user has knowledge about pairwise similarity of features with respect to the response variable (i.e., asking the user about which pairs of features affect the prediction output similarly) which has been investigated in [1, 2]. A large body of works have studied the exclusion/inclusion of features (also known as feature selection [16]) in different contexts, for classification [44, 20, 52], and regression [37]. These methods are different to ours from the modelling point of view (as we consider sparse models) and also the type of interaction with the user.

2.3.2 User feedback as outcome of a cognitive process

The feedback models proposed in the previous section consider the user as a passive data provider with some noise models. However, humans are more complicated than that. Publication IV investigates a similar knowledge elicitation task to the previous section, but aims at accounting for rational process that humans may undergo during the interaction to produce the feedback. In other words, we will define a more complex feedback model that accounts for the cognitive process of the user, with the idea that performance can be improved if we also model the thought process behind the feedback.

In many interaction scenarios, such as visual analytics or human in the loop machine learning systems, certain aspects of the training data, such as statistics or outputs of the machine learning method, may be revealed to the user [18, 47, 1, 39, 54, 59, 32, 34, 48, 37]. This is mainly done to provide information about important characteristics of the data to improve decision making or to guide the user to facilitate the interaction. For example, in the mentioned knowledge elicitation tasks, the system may first show what it has learned about the coefficients of the linear model from the training data and then ask for user feedback about the same coefficients. Though very common, this type of interaction is prone to overfitting, as the user feedback may not be independent of the knowledge in the training data, while the machine learning methods commonly assume independence between data and user input. Studies in cognitive science have shown that humans are unintentionally biased toward the pieces of information provided for them [58, 22]. We propose a user model that accounts for such bias. In other words, the user model assumes that the user is rational [25] and combines the information provided to her with her knowledge and then provides feedback as the outcome of such cognitive procedure. The user model can then undo the bias in the feedback by performing the inverse of the same rational update.

In particular, we consider the case where the user provides feedback about the probability of relevance of a coefficient. The belief of the model (based on data alone) about the probability of relevance of the coefficient, i.e., marginal posterior probability $p(\gamma^j \mid \mathscr{D})$, is also visualized for the user. We make the assumption that the user is rational in combining the visualized information from data with her latent knowledge. The reported feedback is then from the updated knowledge which is naturally biased as it is partially coming from the information in the training data. The model should consider this while inferring the posterior of the parameters of interest given the user feedback.

We assume that the user is rational and combines her latent knowledge about the probability of relevance of the j^{th} coefficient, i.e., $p(f^j \mid \gamma^j = 1)$, with the revealed information from the training data, i.e., $p(\gamma^j = 1 \mid \mathscr{D})$, following a Bayesian update

$$p(\gamma^j = 1 \mid \mathscr{D}, f^j) \propto p(f^j \mid \gamma^j = 1) p(\gamma^j = 1 \mid \mathscr{D}), \tag{2.16}$$

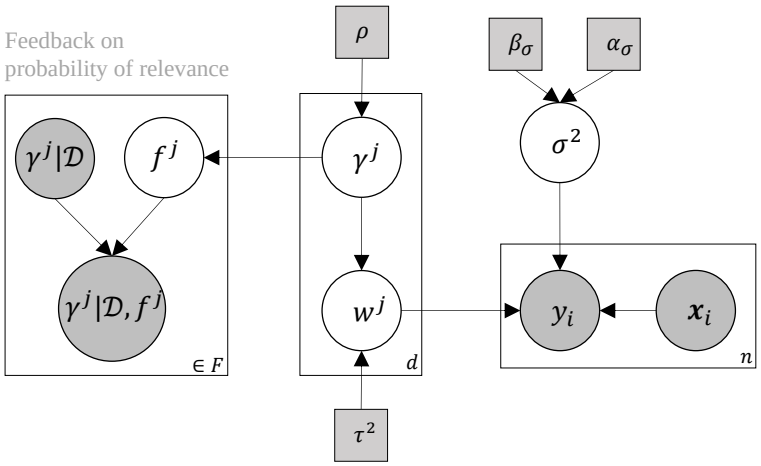

Figure 2.2. Plate notation of the prediction model (right; see Equation 2.8) and user model for feedback (left). The observed user feedback about the probability of relevance of coefficient, $p(\gamma^j \mid \mathscr{D}, f^j)$, is biased as it is generated from combination of latent user knowledge about the coefficient, i.e., $p(f^j \mid \gamma^j)$, and information visualized from the data model about probability of relevance of the coefficient, i.e., $p(\gamma^j \mid \mathscr{D})$. We are interested in inferring the unbiased, latent, version of the feedback to update the parameters of interest in the data model.

and provides as feedback the resulting posterior $p(\gamma^j = 1 \mid \mathscr{D}, f^j)$. The latent feedback model can be represented as $p(f^j \mid \gamma^j) = \pi^j \gamma^j + (1-\pi^j)(1-\gamma^j)$, with π^j being the likelihood for latent feedback when $\gamma^j = 1$. By doing the reverse of the Bayesian update in 2.16, we can infer π^j as:

$$\pi^j \propto \frac{p(\gamma^j = 1 \mid \mathscr{D}, f^j)}{p(\gamma^j = 1 \mid \mathscr{D})}. \tag{2.17}$$

We can then use the inferred user feedback $p(f^j \mid \gamma^j)$, instead of the observed one, to update the posterior for the parameters of interest. Figure 2.2 depicts a schematic of the data and user model for this problem. Publication IV shows in a simple user study that the prediction performance in a simple sentiment analysis task can be improved if the user model accounts for the rational updates that the user may undergo.

Related works

Human biases in interactive tools have been discussed in prior elicitation [22] and visual analytics [60] fields. In particular, [22] reviews the common human biases from psychological literature and provides guidelines for how to reliably extract expert knowledge about uncertain quantities. In an interactive system implemented in [60], the authors proposed a framework for measurement of biases in visual analytics tools and investigated how to inform the user of potential biases. Our work is fundamentally different as we are not limiting (by following a guideline) or interrupting (by showing a warning) the interaction, but rather, acknowledge the presence of such biases and allow the user model, as part of the system, to account for those.

2.3.3 User feedback for intent modelling

Intent modelling describes the task of learning the hidden intent of a user from user feedbacks. Unlike the previous mentioned scenarios where the user feedback was used as complementary information to the training data, in Publication I and V the task is to predict the user intent based on user feedback alone. An important instance of this scenario is the interactive personalized search systems where in each iteration the system recommends items and the user provides feedback regarding relevance of recommendations. The goal is not just to better predict relevance of items for the user but rather to find the most relevant item with minimal interaction. In this section, we discuss the probabilistic modelling of this task and later in Section 3 discuss how to design the interaction so that the most relevant item can be learned as fast as possible.

Consider the case where a user is searching for a scientific article by forming a query with keywords. The user is uncertain about the exact description of the article (e.g., the user may not know the tile), however, if the article is recommended, the user can assess its relevance. This information seeking task is usually performed in several iterations starting with an initial query, assessing the recommended results, and modifying the query with the goal of getting better results. This type of information seeking, with the user being uncertain about her information need, is known as exploratory search [36, 61] and covers around half of search behaviours of users [55] (see [6] for in depth review for exploratory and standard lookup search).

The main bottlenecks of such search systems are that (i) the amount of user feedback that can be gathered is very limited compared to the size of the information space, and (ii) the users are mostly reluctant to give more than few explicit feedback after assessing the recommended results. To tackle these challenges, Publication I, exploits the relationship between articles and their keywords and defines feedback likelihoods on both keywords and articles, providing more flexibility for the user to express her intent. Publication V investigates the modelling of noisy implicit (but effortless) feedback generated from brain activity measured through electroencephalography (EEG) and eye movements.

The intent model is defined as a function that maps all the keywords (used to express the search intent) and articles to real values, indicating the relevance of each item for the user. As the number of user feedback is few, it is reasonable to use a simple linear model to define the relation of user feedback on keywords to the hidden intent (denoted by $\boldsymbol{w}$):

$$f_i^{kw} \sim \mathrm{N}(\boldsymbol{x}_i^\top \boldsymbol{w}, \sigma_{kw}^2), \tag{2.18}$$

where σ_{kw}^2 models the noise of the keyword feedback. The keywords are represented in the keyword-document[5] matrix $\boldsymbol{X} \in \mathbb{R}^{k \times d}$ where the element (i, j)

[5]We use documents to refer to articles or any other potential textual data of interest. TODO: change everything to document (instead of article)?

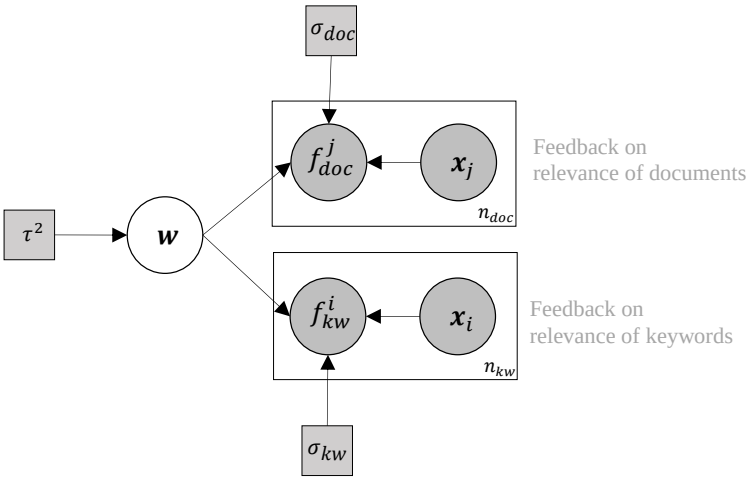

Figure 2.3. Plate notation of the intent modelling from feedback on recommended documents and keywords. The feature transformations are not shown. The feature for the i^{th} keyword (shown as $\boldsymbol{x}_i$) is the i^{th} row of the keyword-document matrix $\boldsymbol{X}$, and for the j^{th} document (shown as $\boldsymbol{x}_j$) is the j^{th} row of $\hat{\boldsymbol{X}}^\top \boldsymbol{X}$.

describes the tf-idf weighting of keyword i (out of k total keywords) in document j (out of d total documents), and $\boldsymbol{x}_i^\top$ indicates the i^{th} row of the matrix. As mentioned, we are interested to also model potential user feedback on documents. We make the simplifying assumption that the expected relevance of a document can be represented as a weighted sum of the expected relevance of keywords that appear in it

$$\mathbb{E}[f_j^{doc}] = \sum_{i=1}^{k} p_{(i|j)} \mathbb{E}[f_i^{kw}], \tag{2.19}$$

where f_j^{doc} indicates the relevance of the j^{th} document, $\mathbb{E}[.]$ denotes the expected value and $p_{(i|j)}$ is the likelihood of the i^{th} keyword being present in the j^{th} document. The likelihood $p_{(i|j)}$ is not available but can be approximated by normalizing $\boldsymbol{X}$ such that its columns sum up to one. Denoting the resulting matrix as $\hat{\boldsymbol{X}}$, writing Equation 2.19 in vector format, and following 2.18, the feedback for relevance of documents can be described as

$$\boldsymbol{F}^{doc} \sim \mathrm{N}(\hat{\boldsymbol{X}}^\top \boldsymbol{X}\boldsymbol{w}, \sigma_{doc}^2 \mathbf{I}), \tag{2.20}$$

where $\boldsymbol{F}^{doc} = [f_1^{doc}, \ldots, f_d^{doc}]$ is the vector representation of relevance feedback for all documents, and σ_{doc}^2 models the noise of the document feedback.

Given user feedback on keywords and documents the posterior of the hidden intent can be learned by assuming a Gaussian prior on the intent weights, i.e., $\boldsymbol{w} \sim \mathrm{N}(\mathbf{0}, \tau^2 \mathbf{I})$. As both likelihoods follow a Gaussian distribution, the posterior has a closed form solution and can be derived following the same derivations

as the simple linear model in 2.7. Figure 2.3 illustrates the schematic of user feedback models and its connection to the latent intent.

Implicit feedback model

The proposed model is very simple as the noise of feedback is assumed to be known and the only unknown parameter is the intent vector $\boldsymbol{w}$. As mentioned, we are interested to consider feedbacks from implicit sources such as neurophysiologic signals gathered from brain activities[6]. The implicit feedback is extremely cheap, as it requires no physical effort from the user, but at the same time extremely noisy. Our modelling solution is to use the automatic relevance detection model proposed in 2.10 to endow the feedback likelihood handle the inherent noise in the implicit feedback when used in combination of the explicit feedbacks. For simplicity, let's assume that only the keyword feedback is generated from an implicit source (it is straightforward to consider a similar feedback source for documents or to consider both explicit and implicit feedback for keywords or documents). The implicit feedback likelihood for keywords can be expressed as

$$f_i^{kw} \sim \mathrm{N}(\boldsymbol{x}_i^\top \boldsymbol{w}, \frac{\sigma_{kw}^2}{\nu_i}), \tag{2.21}$$

where the parameters follow the model 2.10. The model can then infer the accuracy of feedback ν_i based on the observed feedback and handle the high uncertainty in implicit feedback. The resulting model for a case of availability of explicit feedback on documents and implicit feedback on keywords is depicted in Figure 2.4. The resulting posterior is analytically intractable.

Related works

Intent modelling for article search has been studied in [27, 45, 46]. These works are different to ours as they only define the intent on keywords and then use the learned intent as an input to a separate retrieval model (based on a standard language model) to retrieve ranked documents, while document retrieval is a natural outcome of the document feedback likelihood in our work. Furthermore, the previous methods did not consider the option of user providing feedback on documents. We borrowed the same user interface for providing feedback on keywords which is based on the idea of visualizing the top relevant keywords in a radar-like interface where the distance to center for each keyword is proportional to the predicted relevance from the intent model. The user then can provide feedback by dragging a keyword toward the center (indicating positive relevance) or further away from the center (indicating negative feedback). We added document feedback through clicks or bookmarking.

[6]Note that here we assume that the implicit feedbacks, EEG signals, are already converted to a continuous relevance value (see Publication V for details about this conversion).

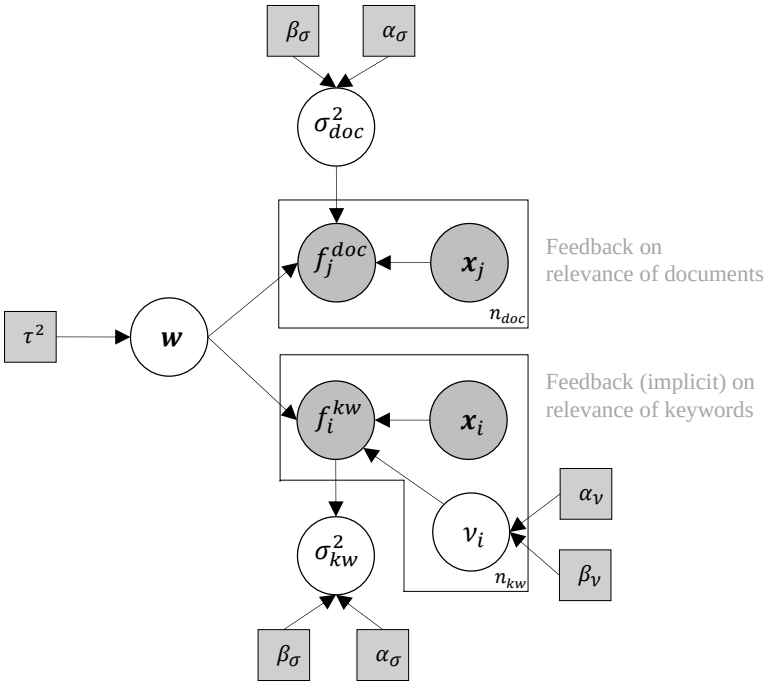

Figure 2.4. Plate notation of the intent modelling from feedback on recommended documents and noisy, implicit feedback on keywords. The data model is based on 2.10 with the feedbacks as the observational data. The feature vector of documents follows the transformation introduced in equation 2.20

2.4 Posterior inference

Other than simple models (like the linear regression introduced in 2.7) the models proposed in this thesis do not have a closed-form posterior and posterior predictive distributions. For low dimensional unknown parameters, it is still practical to approximate the posterior distribution by simulation methods such as numerical computation on girds or Monte Carlo simulations (see [23, Chapter 10] for a review). However, as mentioned, we are interested on high-dimensional problems (e.g., sentiment analysis and search with textual data, or drug response prediction with genomic data), where simple simulation methods cannot scale well due to the curse of dimensionality (the number of evaluations grows exponentially with respect to the dimension). There are two general family of methods to approach such intractable posteriors –namely Markov chain simulation and deterministic posterior approximation methods.

Markov chain simulation methods

Markov chain simulation methods are general methods to draw samples from the target posterior, i.e., $\theta^s \sim p(\theta \mid \mathscr{D})$, by starting from an initial sample in the parameter space and sequentially updating the sample toward the target posterior distribution. The sequence of dependent samples forms a chain which should have the Markov property, i.e., the updating rule to get to θ^{t+1} at time

$t+1$ should only depend on the previous sample θ^t, and not the whole chain. It can be shown that under some assumptions for the updating rule and the Markov chain, by increasing the chain size toward infinity, the last sample in the chain would be a sample from the target distribution [23, Chapter 11]. Recent advances in Markov chain simulation methods has resulted in probabilistic programming languages, such as Stan [13] and Pyro [8], that make the computation of Bayesian inference straightforward for end users. The user of such softwares needs to declare the model, and the probabilistic programming language performs the inference by providing samples from the posterior (for example see [49] for a step-by-step guides to do Bayesian modelling and inference using Stan). The posterior samples can be used to compute predictive distributions, summaries (such as expected value or other estimates), or utilities for decision making. A bottleneck of iterative simulation methods is that they can sometimes be slow. This is particularly important in the works investigated in this thesis as all of them require real-time performance with a user.

Deterministic posterior approximation

The idea of deterministic approximation methods is to approximate the target distribution, such as posterior distribution $p(\theta \mid \mathscr{D})$, with a simpler distribution $q(\theta)$ (e.g., from exponential family), i.e., $q(\theta) \approx p(\theta \mid \mathscr{D})$. The resulting approximation does not fully represent the target distribution, however, in many cases it can be efficient or accurate enough for the targeted task. There are different ways, e.g., different objective functions or assumptions about the approximated distribution, to handle the approximation. Here we briefly review the methods that have been also employed in this thesis.

Variational Bayes (VB) [10] considers $q(\theta)$ from a tractable family and refines it to be similar to the target by minimizing the Kullback–Leibler (KL) divergence between the approximation and the posterior. KL-divergence is a popular asymmetric measure of how two distributions are different from each other and for continuous θ is defined as[7] $\mathrm{KL}[q(\theta) \parallel p(\theta \mid \mathscr{D})] \stackrel{\text{def}}{=} \int q(\theta) \log \frac{q(\theta)}{p(\theta \mid \mathscr{D})} d\theta$. This objective is intractable as it requires the computation of the marginal likelihood $\log p(\mathscr{D})$. Considering $p(\theta \mid \mathscr{D}) = \frac{p(\theta, \mathscr{D})}{p(\mathscr{D})}$, the objective can be written as $\mathrm{KL}[q(\theta) \parallel p(\theta \mid \mathscr{D})] = \log p(\mathscr{D}) - \int q(\theta) \log \frac{p(\theta, \mathscr{D})}{q(\theta)} d\theta$. Since KL is non-negative and $\log p(\mathscr{D})$ is constant with respect to θ, the minimization of the KL-divergence would be equivalent to maximizing $L(q(\theta)) = \int q(\theta) \log \frac{p(\theta, \mathscr{D})}{q(\theta)} d\theta$. By appropriate choice for the model family of $q(\theta)$, the maximization of $L(q(\theta))$, also known as evidence lower bound, becomes tractable (unlike the initial objective). In particular, for the case where θ can be partitioned into disjoint group $\theta_1, \ldots, \theta_m$, by assuming factorization $q(\theta) = \prod_{j=1}^{m} q(\theta_j)$, a general expression for optimal $q(\theta_j)$, that maximizes $L(q(\theta))$, can be achieved. This approximation is known as

[7] for discrete θ the integral turns to summation.

the mean-field VB approximation which is implemented by iteratively updating the approximated factors [9, Chapter 10].

Mean-field VB was used in Publication V for the posterior approximation (Equation 2.11) and partly for the approximation of the residual variance in the sparse linear model used in Publication II, III, and IV (σ^2 in Equation 2.15).

Expectation propagation (EP) [38] aims at finding $q(\theta)$ that minimizes the Kullback–Leibler (KL) divergence between the posterior and the approximation, i.e., $\mathrm{KL}[p(\theta \mid \mathscr{D}) \parallel q(\theta)]$. EP differs from VB from the objective function (as the inputs of KL-divergence are reversed) and also the algorithmic solution of how it is computed. The minimization is usually intractable. The posterior distribution can be written as a product of terms $p(\theta \mid \mathscr{D}) = \prod_j t_j(\theta)$ (for example the posterior in 2.15 can be viewed as product of seven terms). It it is sensible to consider the same structure for the approximated posterior $q(\theta) = \prod_j \tilde{t}_j(\theta)$. EP approximates each term $t_j(\theta)$ in the posterior by a simpler exponential family form $\tilde{t}_j(\theta)$ such that $\prod_j t_j(\theta) \approx \prod_j \tilde{t}_j(\theta)$. As the exponential family is closed under product, $q(\theta)$ also follows a simple and tractable exponential form. To optimize the terms, EP iteratively refines the parameters of $\tilde{t}_j(\theta)$ to minimize $\mathrm{KL}[t_j(\theta)q^{\setminus j}(\theta) \parallel \tilde{t}_j(\theta)q^{\setminus j}(\theta)]$, where $q^{\setminus j}(\theta)$ denotes the approximated posterior after removing the j^{th} term, i.e., $q^{\setminus j}(\theta) \propto \frac{q(\theta)}{\tilde{t}_j(\theta)}$. This is optimized by matching the sufficient statistics of $t_j(\theta)q^{\setminus j}(\theta)$ with $\tilde{t}_j(\theta)q^{\setminus j}(\theta)$. After refining each $\tilde{t}_j(\theta)$, an updated approximation is achieved as $q(\theta) \propto q^{\setminus j}(\theta)\tilde{t}_j(\theta)$. This step is repeated until all the terms converge. [28, 29, 41]

EP can provide good estimate to uncertainty (e.g., approximation for the posterior covariance of the weights in the linear model) that is desirable for designing the interaction (experimental design methods; see Section 3.1) [28]. Furthermore, for sequential interaction, the inference can be sped up by only performing few iterations of parameter updates (instead of waiting for full convergence) as suggested by [50]. EP was used in Publication II, III, and IV for fast approximation of the posterior of the linear regression model with sparsity-inducing spike-and-slab prior on coefficients with data and feedback observations (Equation 2.15). The regression weights ($\boldsymbol{w}$) are approximated by a multivariate Gaussian distribution.

3. User interaction with the probabilistic model

The previous chapter introduced how the user input to a high-dimensional probabilistic model can jointly be modelled with observational data and how the resulting probabilistic inference can be computed. However, the number of possible feedbacks that a human users can provide is usually limited due to the user's reluctance (e.g., in personalized system the user is usually willing to provide only a couple of feedback) or the time and workload constrains (as there is usually the possibility to provide thousands of feedbacks). To reduce the burden of the user and to achieve the best results faster, the interaction with the probabilistic model needs to be carefully designed.

We consider sequential interaction where at each iteration the probabilistic model selects the next query to receive feedback from the user, and the user provides the feedback. The design of the query should be based on the history of observed feedbacks (along with training data, if available) and target of the interaction. Figure 3.1 depicts a schematic of this interaction. The key components of such interactive system can be summarized as

1. A data observation model $p(\mathscr{D} \mid \theta, \phi_D)$, where the data are in the form $\mathscr{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, θ is the unknown parameter of interest, and ϕ_D are model parameters related to data model. The observational data may not be available in some applications (see Section 2.2).

2. A user feedback model $p(f \mid \theta, \phi_F)$, where ϕ_F are model parameters related to the user model (see Section 2.3).

3. A prior model $p(\theta, \phi_D, \phi_F)$ for the hierarchical model description. The data and user model are connected through the shared latent parameter θ.

4. A query algorithm that facilitate gathering feedback f iteratively from the user (will be discussed in this chapter).

5. Bayesian inference for updating the model after user interaction (see Section 2.4).

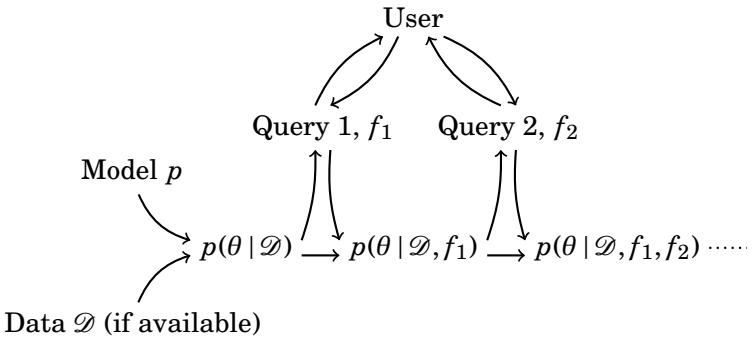

Figure 3.1. Sequential interaction with the user. The probabilistic model uses the history of interaction (and training data, if available) to query the next question from the user and the user provides the corresponding feedback. For brevity, the user and data related parameters, ϕ_D and ϕ_F, are omitted from the posterior. The figure is adapted from Publication II.

This chapter studies two general family of query algorithms for interaction. Section 3.1 reviews the methods that aim to query the most informative query from the user to maximize the predictive performance of the probabilistic model. Section 3.2 reviews the methods that are used to find the most relevant (rewarding) item with minimum number of interactions.

3.1 Sequential experimental design

In many problems, performing new experiments is expensive. Sequential experimental design [14, 50, 51] (also known as active learning [51]) are a family of methods that allow the underlying data model to select the next experiment that would maximally improve the model with respect to a utility measure. The experimental design methods are usually employed at the data domain, where the model asks the expert to label a new data point (e.g., identify the the objects in a picture) with the goal to maximally improve the prediction performance of the model. In Section 2.3.1, we considered the interactions at a novel level where the user could provide feedback about model parameters (such as direction of relevance of features). Here we propose experimental design scheme for interaction with the expert on non-data observations.

The aim of our work is to improve prediction on data. The prediction on an unobservable data given the observed data and also feedbacks form the user is expressed by the posterior predictive distribution of data

$$p(\tilde{y} \mid \tilde{\boldsymbol{x}}, \mathscr{D}, \mathscr{F}) = \int_{\theta} p(\tilde{y} \mid \tilde{\boldsymbol{x}}, \theta) p(\theta \mid \mathscr{D}, \mathscr{F}) d\theta, \tag{3.1}$$

where $\tilde{y}$ is the unobserved response variable, $\tilde{\boldsymbol{x}}$ is the corresponding feature vector, and $\mathscr{F}$ contains the gathered feedback till now. The user and data related parameters (ϕ_f and ϕ_D) are omitted for brevity. The goal is to query the user about the feature, let's say j^*, that the corresponding feedback, i.e., f^{j^*},

would maximally improve this predictive distribution. The improvement can be measured by comparing the posterior predictive distribution after and before observing the feedback using KL-divergence, i.e., $\mathrm{KL}[p(\tilde{y} \mid \tilde{\boldsymbol{x}}, \mathscr{D}, \mathscr{F}, f^{j^*}) \parallel p(\tilde{y} \mid , \tilde{\boldsymbol{x}}, \mathscr{D}, \mathscr{F})]$. This utility is known as the information gain and it measures the impact of the new feedback on the posterior predictive distribution of the data. Information gain cannot be computed as (i) the unobserved data $(\tilde{\boldsymbol{x}}, \tilde{y})$ is not available, and (ii) the user feedback f^{j^*} is only observed only after querying j^* from the user. Rather than computing the utility on unobserved data, we use the available training data set. Furthermore, unobserved feedback can be handled by computing the expected feedback that the user may provide after the query. The expectation would be taken with respect to the posterior predictive of user feedback (the distribution that the system believes the feedback would be generated from), i.e., $p(\tilde{f}^j \mid \mathscr{D}, \mathscr{F}) = \int_\theta p(\tilde{f}^j \mid \theta) p(\theta \mid \mathscr{D}, \mathscr{F}) d\theta$. The best query can then be selected as

$$j^* = \underset{j \notin \mathscr{F}}{\arg\max}\, \mathbb{E}_{p(\tilde{f}^j \mid \mathscr{D}, \mathscr{F})} \left[\sum_{i=1}^{n} \mathrm{KL}[p(\tilde{y} \mid \boldsymbol{x}_i, \mathscr{D}, \mathscr{F}, \tilde{f}^j) \parallel p(\tilde{y} \mid \boldsymbol{x}_i, \mathscr{D}, \mathscr{F})] \right]. \tag{3.2}$$

This new utility (known as expected information gain) is expensive as it requires the computation of the posterior predictive distribution before and after the new feedback (which itself requires the computation of the corresponding posterior and solving the integral in Equation 3.1), computation of the KL-divergence, and computation of the posterior predictive of the feedback necessary for deriving the expectation.

Publications II and III used expected information gain to sequentially extract expert's knowledge about model parameters (see Section 2.3.1 for a summary about the types of knowledge extracted). As mentioned, the posterior distribution of the linear regression model with sparsity-inducing spike-and-slab prior with data data and feedback observations was approximated by a multivariate Gaussian distribution (see Section 2.4). It is straightforward to show that the posterior predictive distribution of data also follows a Gaussian distribution which gives an analytical form to the KL-divergence. The posterior predictive distribution of feedback observations follow Gaussian (for feedback on value of coefficients) and Bernoulli (for feedback on relevance and direction of relevance of coefficients) distributions, which make the computation of the expectation straightforward.

3.2 Multi-armed bandits and Bayesian optimization

Consider the problem of finding the argument that maximizes an objective function in a design space $\mathscr{X}$, i.e., $x^* = \arg\max_{x \in \mathscr{X}} f(x)$. Now consider that the function of interest $f(x)$ is unknown and the only way to gain information about it is to sequentially query it, i.e., ask the function value about a point of

interest, such as $x_q \in \mathcal{X}$, and observe the corresponding function value $f(x_q)$ (or in the general case with some added noise). The natural goal for this black box optimization problem could be to find x^* with minimum number of queries. This problem has been extensively investigated in the multi-armed bandit [12] and Bayesian optimization [11, 53] literatures. A core characteristic of these approaches is that they need to make a compromise between asking queries that would provide new information about the hidden objective function (for example in areas that have not been explored), versus exploiting the current guess about where the maximum is and querying it. This is known as the exploration exploitation trade-off and methods in the literature propose different query algorithms, known as acquisition functions, to balance it. Bayesian optimization and multi-armed bandits methods have been used in many applications such as clinical trials (finding the best treatment of a patient out of several alternatives) [12], ad placement (finding the ad that would have the highest user click chance) [26], Automatic machine learning (searching in space of machine learning models) [30], reinforcement learning (finding the best action of an agent) [11], and personalized search systems [45].

A focused application in this thesis is the personalize search system, where the goal is to find the most relevant item for a user by minimum interaction. As the user relevance profile over items is unobserved and can only be queried through recommendations and observing user feedback, one can can map this problem to the black-box optimization problem introduced above. In particular, the problem can be formulated as finding (recommending) the item $\boldsymbol{x}_{j^*}$ that has the highest expected relevance (or equivalently highest expected feedback value). As explained in 2.3.3, we consider the cases where the expected relevance of items have linear relationship with the unknown parameter $\boldsymbol{\theta}$[1], the optimization can then be described as:

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \mathbb{E}[f \mid \boldsymbol{x}] = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \boldsymbol{x}^\top \boldsymbol{\theta}. \tag{3.3}$$

Given a history of interaction at time t, $\mathcal{F}_t = \{(\boldsymbol{x}_i, f_i)\}_{i=1}^t$, the loss of the query algorithm can be measured by the expected cumulative regret defined as:

$$\text{regret}_t = t\boldsymbol{x}^{*\top}\boldsymbol{\theta} - \sum_{i=1}^{t} \boldsymbol{x}_i^\top \boldsymbol{\theta}. \tag{3.4}$$

Note that the cumulative regret is minimized if the most relevant item is recommended to the user. The relevance profile of the user over items is determined by $\boldsymbol{\theta}$. Since we are using Bayesian statistics, we have an ongoing update of the posterior given the history of user feedback $p(\boldsymbol{\theta} \mid \mathcal{F}_t)$ which can help us to better design the next query to ask from the user due to having an immediate expression of the uncertainty about $\boldsymbol{\theta}$. There are different forms of acquisition functions that exploit the model uncertainty to design the next

[1]Note that we have changed the notation for the unknown weight vectors $\boldsymbol{w}$ in Section 2.3 to $\boldsymbol{\theta}$ for consistency.

query. To commons ones are upper confidence bound (UCB) [7, 33] methods and Thompson sampling [3], which both enjoy theoretical guarantees (for some family of problems) regarding how well they control the regret.

Publications I and V used Thompson sampling to balance the exploration and exploitation of the recommendations. Thompson sampling follows the Bayesian idea and selects the next query according to its posterior probability of being the best query:

$$\Pr(\boldsymbol{x}_q) = \int_{\boldsymbol{\theta}} I(\arg\max_{\boldsymbol{x}\in\mathcal{X}} \boldsymbol{x}^\top \boldsymbol{\theta} = \boldsymbol{x}_q | \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathcal{F}_t) d\boldsymbol{\theta}, \tag{3.5}$$

where $\Pr(\boldsymbol{x}_q)$ is the Thompson probability for selecting query point $\boldsymbol{x}_q$, and $I(.)$ is the indicator function (returns 1 if the condition in front of it holds and zero otherwise). Selection of the query at time $t+1$ according to the Thompson probabilities can be realized by the following steps:

1. Draw a sample from the posterior: $\boldsymbol{\theta}^s \sim p(\boldsymbol{\theta} \mid \mathcal{F}_t)$.

2. Find the query that has the highest expected relevance given the posterior sample: $\boldsymbol{x}_q = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \boldsymbol{x}^\top \boldsymbol{\theta}^s$.

3. Ask the query from the user, observe user feedback, and update the posterior given the new observation: $p(\boldsymbol{\theta} \mid \mathcal{F}_{t+1})$, where $\mathcal{F}_{t+1} = \mathcal{F}_t \cup \{(\boldsymbol{x}_q, f_q)\}$.

In Publications I and V we considered feedback models for items (keywords and documents) that their expectation was also linear with respect to the unknown parameter (see likelihoods 2.18, 2.20, and 2.21 in Section 2.3.3.). Thus, the same algorithm can be applied on them with the difference that at each iteration different query types (documents and keywords) are selected to be represented to the user.

4. Conclusions and discussion

This chapter briefly summarizes the contributions of Publications I-V with emphasize on answering the research questions of the thesis.

4.1 Interactive intent modelling from multiple feedback domains (Publications I and V)

RQ1 was "Can we exploit new sources of interaction as additional learning signals from human user to improve interactive intent modelling?"

Publications I and V investigated the ways to improve interactive intent modelling by considering new feedback domains. The application considered in these papers was a personalized article search problem where at each iteration the system recommends a list of articles and keywords, given the history of user interaction, and the user provides feedback to them with the goal to find the most relevant articles faster.

In particular, Publication I exploited the relationship between articles and keywords and proposed a joint probabilistic model that ties the user feedback on both keywords and documents (see Figure 2.3 for model description) to the latent user intent. Previous works [27, 45] only defined the user intent on keywords and had a separate language model to retrieve documents. The joint probabilistic model allows to learn the user intent with fewer interaction rounds and at the same time provides a unified way to recommend keywords and articles by using Thompson sampling on the posterior of the latent intent (see Section 3.2 for details). The model was evaluated in a simulated study and a user study with 10 participants using the exploratory search system SciNet [27].

Publication V investigated the usage of noisy neurophysiologic signals gathered from brain activities (EEG signals) along with scarce explicit interactions (mouse clicks) for a the same interactive article search task. Compared to the previous work, the feedback is still in the same domain of items but its type is more implicit. The integration of feedbacks is challenging due to different nature of noise. To account for the inherit noise we considered a per feedback prior parameter that controls the noise on the observed feedback, given the feedback

source. The posterior of that parameter allows the system to correct some of the noises and to detect outlier feedback which is common in neurophysiologic feedback sources (see Figure 2.4 for model description). We used the same approach as Publication I to connect the feedback on documents and keywords and to recommend new items. The model was evaluated in a fully integrated information retrieval system that used the real-time generated feedback from brain activities (EEG) and eye movements tracking (to map the brain signal to visualized items), and scarce mouse click feedback.

The conclusion of the thesis to RQ1 is that, interactive intent modelling can be improved by adding new domains of feedback as long as the uncertainty of the feedback source and connection of the feedback to the latent intent is properly accounted for.

4.2 Expert knowledge elicitation for high-dimensional prediction (Publications II and III)

RQ2 was "Can expert knowledge about high dimensional data models be elicited to improve the prediction performance?"

Publications II and III studied the ways user knowledge about the parameters of a model can be encapsulated as likelihood functions and how they can sequentially be queried to improve the performance of a prediction task. Expert knowledge is particularly important in "small n, large p" scenarios where the number of available data points (n) is fewer than the dimension of the problem (p). An example application of such scenario that was investigated in Publication III is the drug response prediction problem given few patient profiles, where it is very costly, or some cases impossible, to add new data to the training data set but expert knowledge is available.

Both publications contributed to the research question by providing models to add different types of expert knowledge to the prediction model (see Figure 2.1 for the three suggested feedback types) and designing the interaction with sequential experimental design to query the most informative expert knowledge earlier (see Section 3.1 for details). Both publications were evaluated by extensive simulated experiments and studies with real users/experts.

The conclusion of the thesis to RQ2 is that, even though the expert knowledge is not as powerful as new labeled data points, if connected properly to the data model, the knowledge can still significantly help to improve the prediction performance. This increase in performance is noticeably fast if the queries are designed sequentially by the model to ask the most informative questions first.

4.3 User modelling for avoiding overfitting in knowledge elicitation (Publication IV)

RQ3 was "Is it enough to incorporate human knowledge directly in the data model as explained in RQ2, or could it be beneficial to account for rational knowledge updates that humans may undergo during the interaction?"

Publication IV investigates the complementary question to RQ2 of what if we acknowledge that the knowledge elicitation is being performed on a human user which comes with its own preset of biases and cognitive characteristics. In particular, we consider anchoring bias as one of the well-studied cognitive biases [58] and ask what if the knowledge elicitation system explicitly model this potential bias when gathering user feedback. The bias can particularly be harmful when extracting user knowledge after revealing certain aspects of training data, such as statistics or scatter plots. If not accounted properly, the user knowledge can be influenced by the information in the training data which may result in overfitting in the prediction model (as the user knowledge and training data are to some degree dependent). We proposed a probabilistic model that acknowledges the bias by assuming that the user updates her latent knowledge with the information in the training data that has been revealed and provides the biased feedback. The probabilistic model than accounts for the bias by performing the inverse of the knowledge update and extracting the tacit knowledge to be used in the model (see Figure 2.2 for model description). The proposed model was evaluated on a simple user study with 49 participants and the results showed that modelling the user bias (or thinking behaviour) in the knowledge elicitation model improves the prediction.

The conclusion of the thesis to RQ2 is that interactive models can gain more from an observed user feedback if it correctly models the unobserved thinking process that the user may undergo to produce the feedback. The finding is striking as it opens a new horizon for user modelling in human-in-the-loop machine learning methods systems where the model acknowledges how the human user interacts and is able to condition on the thinking process to have more informative updates from an observed interaction.

4.4 Discussion

The thesis has focused on modelling the user interaction with a machine learning system as a unified probabilistic model of both the user model, i.e., model of user interaction (in the form of feedback) and intention, and the data model. The user interaction to the model has then designed as a sequential probabilistic inference problem where the most informative query from the user is selected at each iteration (see figure Figure 3.1). We believe that probabilistic user modelling would be of great importance in human-in-the-loop machine learning systems as it allows to (i) infer more information from an observed user feedback and

(ii) design better interactions by exploiting the quantified uncertainty. User modelling, can provide the machine with a model of how users perceive the world, and the works in the thesis show that exploiting this knowledge can improve the interaction. Recent works have shown the benefits of such user modelling. In [42], we showed that the performance of a personalized search system can be improved if the user model acknowledges that the user is exploiting a simplified model of the system they are interacting with.

References

[1] Homayun Afrabandpey, Tomi Peltola, and Samuel Kaski. Interactive prior elicitation of feature similarities for small sample size prediction. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, UMAP '17, pages 265–269, New York, NY, USA, 2017. ACM.

[2] Homayun Afrabandpey, Tomi Peltola, and Samuel Kaski. Human-in-the-loop active covariance learning for improving prediction in small data sets. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 1959–1966. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[3] Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, pages 127–135, 2013.

[4] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. Power to the people: The role of humans in interactive machine learning. *AI Magazine*, 35(4):105–120, 2014.

[5] Muhammad Ammad-ud din, Suleiman A. Khan, Disha Malani, Astrid Murumägi, Olli Kallioniemi, Tero Aittokallio, and Samuel Kaski. Drug response prediction by inferring pathway-response associations with kernelized Bayesian matrix factorization. *Bioinformatics*, 32(17):i455–i463, 08 2016.

[6] Kumaripaba Athukorala, Dorota Głowacka, Giulio Jacucci, Antti Oulasvirta, and Jilles Vreeken. Is exploratory search different? a comparison of information search behavior for exploratory and lookup tasks. *Journal of the Association for Information Science and Technology*, 67(11):2635–2651, 2016.

[7] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

[8] Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D. Goodman. Pyro: Deep Universal Probabilistic Programming. *Journal of Machine Learning Research*, 2018.

[9] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[10] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[11] Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

[12] Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

[13] Bob Carpenter, Andrew Gelman, Matthew Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of Statistical Software, Articles*, 76(1):1–32, 2017.

[14] Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304, 08 1995.

[15] In Kwon Choi, Taylor Childers, Nirmal Kumar Raveendranath, Swati Mishra, Kyle Harris, and Khairi Reda. Concept-driven visual analytics: An exploratory study of model- and hypothesis-based reasoning with visualizations. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 68:1–68:14, New York, NY, USA, 2019. ACM.

[16] Alvaro H. C. Correia and Freddy Lécué. Human-in-the-loop feature selection. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, 2019.

[17] Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. In *Advances in neural information processing systems*, pages 3079–3087, 2015.

[18] S. Das, D. Cashman, R. Chang, and A. Endert. Beames: Interactive multimodel steering, selection, and inspection for regression tasks. *IEEE Computer Graphics and Applications*, 39(5):20–32, Sep. 2019.

[19] David Donoho and Jared Tanner. Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367(1906):4273–4293, 2009.

[20] Gregory Druck, Burr Settles, and Andrew McCallum. Active learning by labeling features. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 81–90, 2009.

[21] Gerhard Fischer. User modeling in human–computer interaction. *User Modeling and User-Adapted Interaction*, 11(1-2):65–86, 2001.

[22] Paul H Garthwaite, Joseph B Kadane, and Anthony O'Hagan. Statistical methods for eliciting probability distributions. *Journal of the American Statistical Association*, 100(470):680–701, 2005.

[23] Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*. Chapman & Hall/CRC, 3rd edition, 2014.

[24] Edward I George and Robert E McCulloch. Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889, 1993.

[25] Samuel J Gershman, Eric J Horvitz, and Joshua B Tenenbaum. Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. *Science*, 349(6245):273–278, 2015.

[26] Dorota Glowacka. Bandit algorithms in information retrieval. *Foundations and Trends® in Information Retrieval*, 13(4):299–424, 2019.

[27] Dorota Głowacka, Tuukka Ruotsalo, Ksenia Konuyshkova, kumaripaba Athukorala, Samuel Kaski, and Giulio Jacucci. Directing exploratory search: Reinforcement learning from user interactions with keywords. In *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, IUI '13, pages 117–128, New York, NY, USA, 2013. ACM.

[28] Daniel Hernández-Lobato, José Miguel Hernández-Lobato, and Pierre Dupont. Generalized spike-and-slab priors for bayesian group feature selection using expectation propagation. *Journal of Machine Learning Research*, 14(1):1891–1945, 2013.

[29] José Miguel Hernández-Lobato, Daniel Hernández-Lobato, and Alberto Suárez. Expectation propagation in linear regression models with spike-and-slab priors. *Machine Learning*, 99(3):437–487, 2015.

[30] Matthew Hoffman, Bobak Shahriari, and Nando Freitas. On correlation and budget constraints in model-based bandit optimization with application to automatic machine learning. In *Artificial Intelligence and Statistics*, pages 365–374, 2014.

[31] Antti Kangasrääsiö, Yi Chen, Dorota Glowacka, and Samuel Kaski. Interactive modeling of concept drift and errors in relevance feedback. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*, UMAP '16, pages 185–193, New York, NY, USA, 2016. ACM.

[32] Ashish Kapoor, Bongshin Lee, Desney Tan, and Eric Horvitz. Interactive optimization for steering machine classification. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 1343–1352, New York, NY, USA, 2010. ACM.

[33] Emilie Kaufmann, Olivier Cappe, and Aurelien Garivier. On bayesian upper confidence bounds for bandit problems. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*, pages 592–600, La Palma, Canary Islands, 21–23 Apr 2012. PMLR.

[34] Josua Krause, Adam Perer, and Enrico Bertini. INFUSE: interactive feature selection for predictive modeling of high dimensional data. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1614–1623, 2014.

[35] Bing Liu. *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge University Press, 2015.

[36] Gary Marchionini. Exploratory search: From finding to understanding. *Communications of the ACM*, 49(4):41–46, 2006.

[37] Luana Micallef, Iiris Sundin, Pekka Marttinen, Muhammad Ammad-ud din, Tomi Peltola, Marta Soare, Giulio Jacucci, and Samuel Kaski. Interactive elicitation of knowledge on feature relevance improves predictions in small data sets. In *Proceedings of the 22Nd International Conference on Intelligent User Interfaces*, IUI '17, pages 547–552, New York, NY, USA, 2017. ACM.

[38] Thomas P Minka. Expectation propagation for approximate Bayesian inference. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 362–369, 2001.

[39] Thomas Mühlbacher and Harald Piringer. A partition-based framework for building and validating regression models. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):1962–1971, 2013.

[40] Anthony O'Hagan, Caitlin E. Buck, Alireza Daneshkhah, J. Richard Eiser, Paul H. Garthwaite, David J. Jenkinson, Jeremy E. Oakley, and Tim Rakow. *Uncertain Judgements. Eliciting Experts' Probabilisties*. Wiley, Chichester, England, 2006.

[41] Tomi Peltola. Sparse bayesian linear models: Computational advances and applications in epidemiology; harvuutta suosivat bayesilaiset lineaarimallit: laskennallisia menetelmiä ja sovelluksia epidemiologiassa, 2014.

[42] Tomi Peltola, Mustafa Mert Çelikok, Pedram Daee, and Samuel Kaski. Machine teaching of active sequential learners. In *Advances in neural information processing systems*, 2019.

[43] Juho Piironen and Aki Vehtari. Sparsity information and regularization in the horseshoe and other shrinkage priors. *Electron. J. Statist.*, 11(2):5018–5051, 2017.

[44] Hema Raghavan, Omid Madani, and Rosie Jones. Active learning with feedback on features and instances. *Journal of Machine Learning Research*, 7(Aug):1655–1686, 2006.

[45] Tuukka Ruotsalo, Giulio Jacucci, Petri Myllymäki, and Samuel Kaski. Interactive intent modeling: Information discovery beyond search. *Communications of the ACM*, 58(1):86–92, 2015.

[46] Tuukka Ruotsalo, Jaakko Peltonen, Manuel J. A. Eugster, Dorota Głowacka, Patrik Floréen, Petri Myllymäki, Giulio Jacucci, and Samuel Kaski. Interactive intent modeling for exploratory search. *ACM Trans. Inf. Syst.*, 36(4):44:1–44:46, October 2018.

[47] Dominik Sacha, Michael Sedlmair, Leishi Zhang, John A Lee, Jaakko Peltonen, Daniel Weiskopf, Stephen C North, and Daniel A Keim. What you see is what you can change: Human-centered machine learning by interactive visualization. *Neurocomputing*, 268:164–175, 2017.

[48] Advait Sarkar, Mateja Jamnik, Alan F Blackwell, and Martin Spott. Interactive visual machine learning in spreadsheets. In *Visual Languages and Human-Centric Computing (VL/HCC), 2015 IEEE Symposium on*, pages 159–163. IEEE, 2015.

[49] Daniel J Schad, Michael Betancourt, and Shravan Vasishth. Toward a principled bayesian workflow in cognitive science. *arXiv preprint arXiv:1904.12765*, 2019.

[50] Matthias W Seeger. Bayesian inference and optimal design for the sparse linear model. *Journal of Machine Learning Research*, 9:759–813, 2008.

[51] Burr Settles. Active learning literature survey. Computer sciences technical report 1648, University of Wisconsin, Madison, 2010.

[52] Burr Settles. Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1467–1478, 2011.

[53] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, Jan 2016.

[54] Justin Talbot, Bongshin Lee, Ashish Kapoor, and Desney S. Tan. Ensemblematrix: Interactive visualization to support machine learning with multiple classifiers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1283–1292, New York, NY, USA, 2009. ACM.

[55] Jaime Teevan, Christine Alvarado, Mark S. Ackerman, and David R. Karger. 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*, CHI '04, pages 415–422, New York, NY, USA, 2004. ACM.

[56] Robert Tibshirani. Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(3):273–282, 2011.

[57] J. Ting, A. D'Souza, and S. Schaal. Automatic outlier detection: A bayesian approach. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 2489–2494, April 2007.

[58] Amos Tversky and Daniel Kahneman. Judgment under uncertainty: Heuristics and biases. *Science*, 185(4157):1124–1131, 1974.

[59] Stef Van Den Elzen and Jarke J van Wijk. BaobabView: Interactive construction and analysis of decision trees. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 151–160. IEEE, 2011.

[60] E. Wall, L. M. Blaha, L. Franklin, and A. Endert. Warning, bias may occur: A proposed approach to detecting cognitive bias in interactive visual analytics. In *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 104–115, Oct 2017.

[61] Ryen W White and Resa A Roth. Exploratory search: Beyond the query-response paradigm. *Synthesis lectures on information concepts, retrieval, and services*, 1(1):1–98, 2009.

Publication I

Pedram Daee, Joel Pyykkö, Dorota Głowacka, and Samuel Kaski. Interactive Intent Modeling from Multiple Feedback Domains. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*, Sonoma, California, USA, 71–75, March 2016.

© 2016 ACM.

Reprinted with permission.

Publication II

Pedram Daee*, Tomi Peltola*, Marta Soare*, and Samuel Kaski. Knowledge elicitation via sequential probabilistic inference for high-dimensional prediction. *Machine Learning*, 106, 9-10, 1599–1620, 2017.

© 2017 Copyright belongs to the authors.

Reprinted with permission.

Publication III

Iiris Sundin*, Tomi Peltola*, Luana Micallef, Homayun Afrabandpey, Marta Soare, Muntasir Mamun Majumder, Pedram Daee, Chen He, Baris Serim, Aki Havulinna, Caroline Heckman, Giulio Jacucci, Pekka Marttinen, and Samuel Kaski. Improving genomics-based predictions for precision medicine through active elicitation of expert knowledge. *Bioinformatics*, 34, 13, i395–i403, 2018.

© 2018 Copyright belongs to the authors.

Reprinted with permission.

Publication IV

Pedram Daee*, Tomi Peltola*, Aki Vehtari, and Samuel Kaski. User Modelling for Avoiding Overfitting in Interactive Knowledge Elicitation for Prediction. In *Proceedings of the 23rd International Conference on Intelligent User Interfaces*, Tokyo, Japan, 305–310, March 2018.

© 2018 ACM.

Reprinted with permission.

Publication V

Giulio Jacucci, Oswald Barral, Pedram Daee, Markus Wenzel, Baris Serim, Tuukka Ruotsalo, Patrik Pluchino, Jonathan Freeman, Luciano Gamberini, Samuel Kaski, Benjamin Blankertz. Integrating Neurophysiological Relevance Feedback in Intent Modeling for Information Retrieval. *Journal of the Association for Information Science and Technology*, 2019.

© 2019 Copyright belongs to the authors.

Reprinted with permission.