Machine Learning for Pairwise Data

Applications for Preference Learning and Supervised Network Inference

Een wetenschappelijke proeve op het gebied van de
Natuurwetenschappen, Wiskunde en Informatica

Proefschrift

ter verkrijging van de graad van doctor
aan de Radboud Universiteit Nijmegen
op gezag van de rector magnificus prof. mr. S.C.J.J. Kortmann,
volgens besluit van het college van decanen
in het openbaar te verdedigen op dinsdag 18 oktober 2011
om 15:30 uur precies

door

Adriana Bîrluţiu

geboren op 18 december 1981
te Abrud, Roemenië

Promotor:

Prof. dr. T.M. Heskes

Manuscriptcommissie:

Prof. dr. L.M.C. Buydens
Prof. dr. A.P.J.M. Siebes (Universiteit Utrecht)
Prof. dr. B. Hammer (Universität Bielefeld, Germany)

SIKS Dissertation Series No. 2011-37

The research reported in this thesis has been carried out under the auspices of the Dutch Research School for Information and Knowledge Systems (SIKS), and the Institute for Computing and Information Sciences of the Radboud University Nijmegen.

This research has been funded by the Dutch Technology Foundation (STW), applied science division of the Netherlands Organization for Scientific Research and the Technology Program of the Ministry of Economic Affairs.

ISBN: 978-90-8891-330-3
Cover design: Ideaal Communicatie, Apeldoorn
Printed & Lay Out: Proefschriftmaken.nl || Printyourthesis.com
Published: Uitgeverij BOXPress, Oisterwijk

Contents

1 Introduction **1**

1.1 Outline of the Thesis 4

2 Methods for Analyzing Pairwise Data **7**

2.1 Introduction 8

2.1.1 Overview of Bayesian Methods 8

2.1.2 Bayesian Framework for Rating Players 9

2.2 Probabilistic Framework for Estimating Players' Strengths 9

2.3 Expectation Propagation 10

2.4 Experimental Evaluation 13

2.4.1 Accuracy 14

2.4.2 Confidence 15

2.5 Conclusions and Future Work 16

I Preference Learning **17**

3 Multi-Task Preference Learning **19**

3.1 Introduction 20

3.1.1 Related Work 20

3.1.2 Outline of the Chapter 21

3.1.3 Notation 21

3.2 Probabilistic Choice Models 22

3.3 Models for Utility Functions 23

3.3.1 Parametric Models 24

3.3.2 Non-Parametric Models 24

3.3.3 Semi-Parametric Models 25

3.4 Learning Utility Functions 27

3.5 Multi-Task Preference Learning 27

3.5.1 EM Algorithm for Learning the Hierarchical Prior 28

3.6 Experimental Evaluation 29

3.7 Conclusions and Future Work . . . 30
3.8 Appendix A: Equivalence of the GP Representations . . . 31
3.9 Appendix B: Methods for Approximate Inference . . . 32
3.10 Appendix C: EM Derivation . . . 34

4 Efficient Preference Learning 39
4.1 Introduction . . . 40
4.1.1 Background and Related Work . . . 41
4.1.2 Notation . . . 44
4.2 Learning Framework . . . 45
4.2.1 Probabilistic Choice Models . . . 45
4.2.2 The Utility Function . . . 46
4.2.3 Multi-task Preference Learning . . . 47
4.3 Active Preference Learning . . . 48
4.3.1 QBC for Preference Learning . . . 48
4.3.2 Average Probability . . . 51
4.3.3 Standard Criteria for Active Learning . . . 52
4.3.4 Similarities between Criteria . . . 54
4.4 Experimental Evaluation . . . 58
4.4.1 Data Sets . . . 58
4.4.2 Protocol . . . 59
4.4.3 Performance . . . 60
4.4.4 Computational Complexity . . . 63
4.4.5 Similarities between Criteria . . . 64
4.5 Conclusions and Discussions . . . 65
4.5.1 Future Work . . . 66
4.6 Appendix . . . 66

II Supervised Network Inference 71

5 Bayesian Framework for Protein-Protein Interaction Prediction 73
5.1 Introduction . . . 74
5.2 Models and Methods . . . 75
5.2.1 Topological Properties of PPI Networks . . . 75
5.2.2 Random Graph Generator . . . 76
5.2.3 Bayesian Framework for Combining Topology and Feature Information . . . 78
5.3 Results . . . 84
5.3.1 Data Sets . . . 84
5.3.2 Experimental Setup . . . 85
5.3.3 Performance . . . 85

5.3.4 Topology Learning 86
5.3.5 The Influence of the Prior 87
5.4 Discussions 88
5.5 Appendix: Bayesian Inference 89

6 Link Transfer for Protein-Protein Interaction Prediction 91
6.1 Introduction 92
6.2 Regularized Output Kernel Regression 93
6.3 Link Transfer 95
6.3.1 Converter 95
6.3.2 Link Propagation 97
6.4 Empirical Evaluation 97
6.4.1 Data 97
6.4.2 Protocol 98
6.4.3 Results 98
6.5 Conclusions 100
6.6 Appendix 101

Bibliography 104

Samenvatting (Dutch Summary) 121

Chapter 1

Introduction

Machine learning is a branch of artificial intelligence concerned with the design and development of methods that allow machines to learn. Based on observations and experience, machines can learn to make accurate predictions and take useful decisions. In the recent years, machine learning has become a highly successful discipline with applications in many different areas. Its success can largely be explained by the increasing availability of empirical data and computational power. The type of applications of machine learning range from computational biology to astronomy to robotic surgery. Successful applications of machine learning are spam filtering, speech and hand-write recognition, medical diagnosis, detecting credit card fraud, stock market analysis, to name just a few.

There are many parallels between machine learning and **human learning**. On the one hand, many techniques in machine learning derive from the efforts of psychologists to build computational models for theories of human learning. On the other hand, the concepts and techniques explored by researchers in machine learning might help biologists in a better understanding of human learning. People learn many and various things in a very efficient manner. This is a consequence, among others, of the property of human learning being highly dependent on prior experience. For example, the abilities acquired while learning to walk apply when one learns to run, knowledge gained while learning to recognize cars applies when recognizing trucks, and one can easily learn to speak Dutch if he/she already speaks German. Human learning can also be active, in the sense that the learner can select the most useful information while learning. As a consequence of these characteristics of human learning, people can learn languages, concepts, and causal relationships from far less data and much faster than any automated system.

Most of the machine learning tasks are supervised in the sense that the learning consists of inferring a function from training data. The training data is formed by a set of examples, each example being a pair made by an input object and a desired output value. In spam filtering, an automatic classifier to label emails as “spam” or “not spam” is trained using a sample of previous emails labeled by a human user. In automatic mammogram classification, a model is trained on medical images which were labeled by radiologists as “malignant” or “benign”. Machines can interpret human speech by learning from vocal recordings that have been previously annotated for words and sentences. In all these cases and many others, obtaining labeled data to train the algorithms is expensive. This is an important issue that machine learning has to address. Making the parallel with human learning, **transfer/multi-task learning** and **active learning** are areas that have been extensively explored in the machine learning community over the past few years. The idea behind multi-task learning is to utilize labeled data from other similar learning tasks in order to improve the performance on a target task. Multi-task learning is applicable to the situation in which data for a specific single scenario is scarce, but data is already available from similar scenarios. Active learning can be applied when the algorithm can interactively ask by its own choice for the labels of unlabeled examples. The motivation behind active learning is that a good classifier can be learned with a smaller training set by actively selecting the labels of informative examples, than by random selection.

The work in this thesis is an exploration of multi-task/transfer learning and active learning, for two directions: preference learning and supervised network inference.

Preference learning is a research field studying methods for modeling and predicting people’s desires. It has attracted significant attention in the machine learning community in the last years since it is a crucial aspect in modern applications such as decision support systems (Chajewska et al., 2000), recommender systems (Blythe, 2002; Blei et al., 2003), and personalized devices (Clyde et al., 1993; Heskes and de Vries, 2005). E-commerce, marketing, health care, computer games are application areas that significantly benefit from preference learning methodologies. A prototypical example for application of preference learning that we will use in this thesis is **fitting hearing-aids**, i.e., tuning of hearing-aid parameters so as to maximize user satisfaction. This is a complex task, due mainly to three reasons: *1)* high dimensionality of the parameter space, *2)* the determinants of hearing-impaired user satisfaction are unknown, and *3)* the evaluation of this satisfaction through listening tests is costly (in terms of patient burden and clinical time investment) and unreliable (due to inconsistent responses). The last point illustrates an important issue that preference learning has to address, which is the limited availability of labeled data used for model training. Obtaining appropriate training data in preference learning applications requires time and effort from the modeled user. This shortcoming can be addressed by taking advantage of two characteristics of the settings in which preference learning is usually applied. First, the training data is mostly acquired through interactions with the modeled user; and, second, preferences are modeled for multiple users, as a result multiple training data sets are available. In order for the preference learning methods to be implemented in real-world systems, they must be capable of exploiting all possible sources of information and in the most efficient way. Figure 1.1 is a schematic representation of fitting a hearing-aid. User preferences are learned from listening experiments of the type “which of the two audio samples sounds best?” Transfer learning methodologies allow to include useful information like preference data already

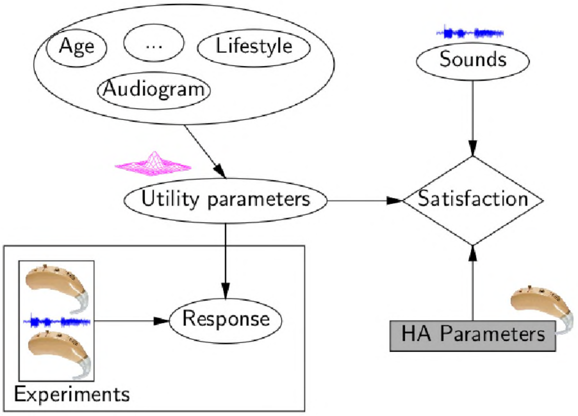

Figure 1.1: Fitting a hearing-aid: a utility model is learned for a patient based on listening experiments and background information such as age, audiogram, lifestyle parameters, etc. The utility model determines the satisfaction of the patient with a certain setting of the hearing-aid parameters and for a given sound sample.

collected from other user and background information such as auditory profile, age, etc. Active learning makes it possible to select the most informative listening experiments throughout a fitting session. As a result, this method promises to find better parameter values with the same number of responses as current methods.

The inference of biological networks is currently an active research subject with important applications ranging from basic biology to medical applications. For example, knowing the interactions between proteins would highlight key proteins that interact with many partners, which could be interesting drug targets (Barabási et al., 2011). The elucidation of gene regulatory networks would provide new insights into the complex mechanisms that allow an organism to regulate its metabolism and adapt itself to environmental changes (Kumar et al., 2008). A recent trend is to solve the problem of network reconstruction in a supervised learning setting. The network of interest is often partially known and the reconstruction process uses this partial knowledge to guide the inference of the missing edges. In the case of **protein-protein interaction (PPI) networks**, information about proteins and labels for protein pairs as interacting or not, supervise the estimation of a function that can predict whether a physical interaction exists or not between two proteins. Figure 1.2 is a schematic representation of the yeast PPI network, the nodes in the graph represent proteins and the edges represent the interactions. Obtaining labeled training data can only be done through costly and tedious laboratory experiments. Computational techniques for predicting protein interactions have become standard tools to address this problem complementing their experimental counterparts. Accurately predicting which proteins might interact can help in designing and guiding future laboratory experiments. Therefore, much effort is recently being put in develop-

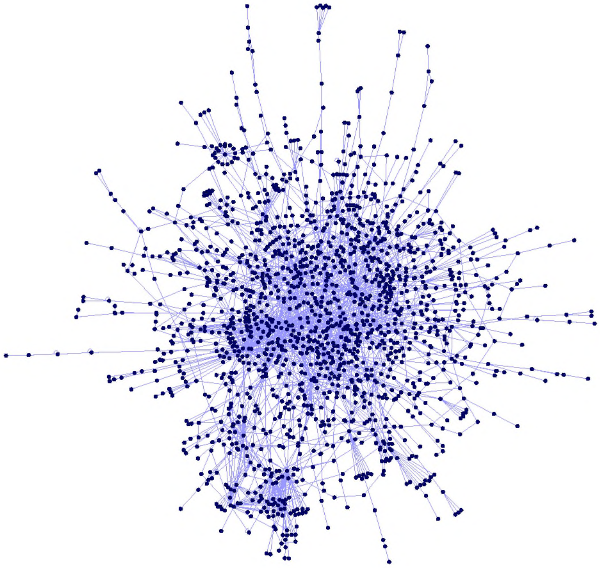

Figure 1.2: Yeast protein-protein interaction network: an undirected graph with no self-loops that encodes the proteins of an organism as nodes and two nodes are connected by an edge if the proteins they represent can physically interact.

ing computational methods that can accurately predict PPIs. Similar to the preference learning setting discussed above, the learning setting for PPI prediction is characterized by multiple data sets, in the sense that biological networks in general, and PPI networks in particular, are modeled from multiple organisms, which makes it possible to apply the multi-task/transfer learning paradigm in order to build accurate models for predicting PPIs.

1.1 Outline of the Thesis

This thesis starts with evaluating, in **Chapter 2**, methods for the analysis of pairwise data. This type of data will be used in the subsequent chapters, for preference learning and for predicting PPIs. The analysis of pairwise data is done in a Bayesian framework in which inference is intractable. Several techniques for approximate inference have been proposed in the literature, one of the most popular being Expectation propagation. Expectation propagation can be computationally expensive and in order to simplify it and adapt it to pairwise data we propose several modifications. The question that we want to answer in this chapter is: how do different variants of expectation propagation perform for the analysis of pairwise data in a Bayesian setting? These variants are evaluated by rating players in sports competitions, more precisely in tennis.

The next chapters, Chapter 3 and Chapter 4, investigate applications of machine learning for preference learning. Preference learning is a research field studying methods for modeling and predicting users desires. The work on preference learning has a focus on two directions. The first direction considers incorporating the information available from other users when learning the preferences of a new user. The second direction is concerned with the choice of experiments performed with a user in order to learn his/her preferences. **Chapter 3** investigates multi-task learning for preference learning with Gaussian processes. We use the multi-task formalism to enhance the individual training data by making use of the preference information learned from other subjects. The contribution of this chapter is the combination of multi-task learning with another learning formalism, Gaussian processes (GPs) for the task of preference learning. The advantage of using GPs is twofold, first, GPs allow for non-parametric, thus complex, modeling, and second, when identifying high-quality features is difficult, GPs allow to specify a particular kind of pairwise function between data objects, known as a kernel function, which is used by the algorithm instead of explicit features. GPs are combined with the multi-task formalism by means of a semiparametric model. By using a semiparametric representation, multi-task learning can be easily implemented by employing the theory of hierarchical modeling for parametric models while in the same time benefiting from the characteristics of GPs. We demonstrate the usefulness of our model on an audiological data set. Since preference learning is a cumbersome process it is important to make it as efficient as possible in order to reduce the costs and time involved. **Chapter 4** introduces a framework for optimizing the preference learning process. This framework considers the combination between active learning and multi-task learning. Active learning has hardly been studied in a multi-task formalism. In this chapter we offer an alternative for the standard criteria in active learning which actively chooses queries by making use of the available preference data from other subjects. The advantage of this alternative is the reduced computation costs and reduced time subjects are involved. The contribution of this chapter is a criterion for active learning designed for the multi-task setting; we show in theory and practice that this new criterion performs similar to the standard criteria from optimal experimental design. We validate empirically our approach on three real-world data sets involving the preferences of people.

The next chapters, Chapter 5 and Chapter 6, investigate applications of machine learning for supervised network inference, in particular for protein-protein interaction networks. The work on supervised network inference has a focus on two topics has a focus on two topics. First, it investigates approaches for combining information about the topological structure of biological networks together with information about each protein in order to construct accurate models for PPI prediction. Second, it investigates methods for supervised network inference in a multi-task setting, that is transferring knowledge about PPIs from several reference species to a target species using orthology information. **Chapter 5** presents a principled way, based on Bayesian inference, for combining network topology information together with observations about protein pairs as interacting or not. The goal of this combination is to improve the prediction of PPIs. We define a model for generating random graphs which gives rise to networks with topology similar to the one observed in PPI networks. We incorporate to this model the actual information from the network we investigate by treating our random graph model as a prior and define a probability model for protein features given the absence/presence of an interaction, and combine these two using Bayes rule, to finally arrive at a model incorporating both topological and feature information. We show experimentally that this resulting model improves the prediction accuracy in yeast and human PPI networks. **Chapter 6** investigates link transfer applied to PPI prediction. Bioinformatics researchers have defined strategies that consist in mapping known interactions between a reference organism onto a target organism and this for the orthologous genes: this is called the protein-protein interologs approach. Predicting using interologs is based on the theory that proteins interacting in one organism co-evolve such that their respective orthologs maintain the ability to interact in another organism. The contribution of this chapter is a framework for link prediction, we call it "link transfer", that resembles the interolog approach while remaining in the supervised learning setting. This link transfer framework is build on the theory of output kernel regression by using output kernel regression twice: first, to convert output feature vectors from a reference species to the target species and second, to learn the target network. The underlying idea of the converter is to increase the training set of the target species by converting the output space of the reference species to the output space of the target species.

Chapter 2

Methods for Analyzing Pairwise Data

Rating players in sports competitions based on game results is one example of pairwise data analysis. Since an exact Bayesian treatment is intractable, several techniques for approximate inference have been proposed in the literature. In this chapter we compare several variants of expectation propagation (EP). EP generalizes assumed density filtering (ADF) by iteratively improving the approximations that are made in the filtering step of ADF. Furthermore, we distinguish between two variants of EP: EP-Correlated, which takes into account the correlations between the strengths of the players and EP-Independent, which ignores those correlations. We evaluate the different approaches on a large tennis data set to find that EP does significantly better than ADF (iterative improvement indeed helps) and EP-Correlated does significantly better than EP-Independent (correlations do matter).[1]

[1] This work has been published as: A. Birlutiu and T. Heskes "Expectation Propagation for rating players in sports competitions." *Knowledge Discovery in Databases: PKDD 2007, 11th European Conference on Principles and Practice of Knowledge Discovery in Databases, volume 4702 of Lecture Notes in Computer Science, pages 374–381, Springer, 2007.*

2.1 Introduction

In this chapter we discuss methods for the analysis of pairwise data. We illustrate such methods by rating players in sports, in particular in tennis. In the rest of this section we will first present Bayesian methods in general (Section 2.1.1) and then give a general overview of how this applies to rating players in sports competitions (Section 2.1.2).

2.1.1 Overview of Bayesian Methods

Bayesian inference is a statistical method that provides a principled way to update hypotheses about quantities or states of certain systems by using empirical data. The hypotheses are typically expressed with the help of probabilities on the quantities or states under consideration. Let the variable θ be a quantity of interest and let the probability $p(\theta|\mathcal{M})$ express the uncertainty in θ given some background information expressed by the model $\mathcal{M}$. Based on a (new) set of observations, called it Y, the knowledge about θ can be updated. The principled way to incorporate the empirical information is by combining the uncertainties using Bayes rule (Cox, 1946). The "updated" knowledge about θ is expressed by the conditional probability $p(\theta|Y, \mathcal{M})$ computed according to

$$p(\theta|Y, \mathcal{M}) = \frac{p(Y|\theta; \mathcal{M})p(\theta|\mathcal{M})}{p(Y|\mathcal{M})} \quad \text{(Bayes' rule)}\,.$$

The assumed conditional probability $p(Y|\theta; \mathcal{M})$ expresses the uncertainty in Y given a fixed θ and the model $\mathcal{M}$. The proportionality factor $p(Y|\mathcal{M})$ is independent of θ and expresses the uncertainty in observing Y given the model $\mathcal{M}$. It is computed by averaging the probability of observing Y over all possible values of θ, i.e.,

$$p(Y|\mathcal{M}) = \int d\theta p(Y|\theta; \mathcal{M})p(\theta|\mathcal{M}) \quad \text{(evidence)}\,.$$

This quantity is called the evidence and can be used to compare two different modeling choices $\mathcal{M}_1$ and $\mathcal{M}_2$. The quantity $p(\theta|\mathcal{M})$ is called the prior distribution and $p(\theta|Y, \mathcal{M})$ is called the posterior distribution. An important property of the Bayesian updating rule is that the information from a set of observations $Y = \{y_1, \ldots, y_n\}$ can be incorporated gradually, i.e.,

$$\begin{aligned} p(\theta|y_1; \mathcal{M}) &\propto p(\theta|\mathcal{M})p(y_1|\theta; M)\,, \\ p(\theta|y_1, y_2; \mathcal{M}) &\propto p(\theta|y_1; \mathcal{M})p(y_2|\theta; \mathcal{M})\,, \ldots \end{aligned}$$

This allows us to update the probability (density) $p(\theta|\mathcal{M})$, whenever any information in terms of observations becomes available.

2.1.2 Bayesian Framework for Rating Players

In this work we present a Bayesian framework for rating players based on match outcomes. We consider the player's strength as a probabilistic variable in a Bayesian framework. Before taking into account the match outcomes, information available about the players can be incorporated in a prior distribution. Using Bayes' rule we compute the posterior distribution over the players' strengths. We take the mean of the posterior distribution as our best estimate of the players' strengths and the covariance matrix as the uncertainty about our estimation.

An exact Bayesian treatment is intractable, even for a small number of players; the posterior distribution cannot be evaluated analytically, and therefore we need approximations for it. Expectation propagation (Minka, 2001) is a popular approximation technique. We will use it in this work for approximating the posterior distribution over the players' strengths. The question that we want to answer here is: how do different variants of expectation propagation perform for this setting? In particular, does it make sense to perform backward and forward iterations for the approximations and does it help to have a more complicated (full) covariance structure?

The rest of the chapter is structured as follows: in the next section, Section 2.2, we introduce the probabilistic framework used to estimate players' strengths. In Section 2.3, we present algorithms for approximate inference and the way they apply to our setting. In Section 2.4, we show experimental results for real data based on which we compare the performance of the algorithms. In Section 2.5 we end with conclusions.

2.2 Probabilistic Framework for Estimating Players' Strengths

Let $\boldsymbol{\theta}$ be an n_{players}-dimensional probabilistic variable whose components represent the players' strengths. We define $y_{ij} = 1$ if player i wins over player j, and $y_{ij} = -1$ otherwise. For modeling the probability of y_{ij} as a function of the strengths θ_i and θ_j, we use the Bradley-Terry model (Bradley and Terry, 1952):

$$p(y_{ij}|\theta_i, \theta_j) = \frac{1}{1 + \exp[-y_{ij}(\theta_i - \theta_j)]} . \tag{2.1}$$

A straightforward method to approximate the players' strengths is to compute the likelihood of $\boldsymbol{\theta}$ given Y; where Y stands for the outcomes of all played matches. The likelihood is computed as the product of terms of the form given in Equation (2.1), with each term corresponding to a match outcome. This way of computing the likelihood is based on the assumption that the match outcomes are independent. This assumption does not totally holds in practice, but we have to make it in order to obtain a reasonable computational form of the likelihood. We can then take the maximum of the likelihood as the estimate for the strengths of the players.

The maximum likelihood approach gives a point estimate, the Bayesian approach, on the other hand, yields a distribution over the players' strengths. Furthermore, useful sources of information, like results in previous competitions and additional information about the players, can be incorporated in a prior distribution over the strengths. Using Bayes' rule we compute the posterior distribution over the players' strengths:

$$p(\boldsymbol{\theta}|Y) = \frac{1}{d}p(Y|\boldsymbol{\theta})p(\boldsymbol{\theta}) = \frac{1}{\mathcal{E}}p(\boldsymbol{\theta})\prod_{i\neq j} p(y_{ij}|\theta_i,\theta_j)\,, \tag{2.2}$$

where $p(\boldsymbol{\theta})$ is the prior, $p(y_{ij}|\theta_i, \theta_j)$ is given in Equation (2.1), and $\mathcal{E}$ is a normalization constant, the evidence.

The mean or the mode of the posterior can be taken as the best estimate for the players' strengths. While computing the mean of the posterior distribution is computationally intractable, its mode (MAP) can be determined using optimization algorithms. For the MAP estimate the computation time is linear in the number of matches, and the number of iterations needed to obtain convergence. Typically, with a state-of-the-art optimization method such as conjugate gradient, the number of iterations needed scales linearly with the number of players.

For making accurate predictions and estimating the confidence of these predictions, using the entire posterior distribution over the players' strengths is a better option than only focusing on point estimates of this distribution. The posterior distribution obtained in Equation (2.2) using Bayes' rule cannot be computed analytically, hence we need to make approximations for it. For this task, sampling methods are very costly because of the high-dimensionality of the sampling space: the dimension is equal to the number of players. Therefore, for rating players, we here focus on deterministic approximation techniques, in particular expectation propagation and variants of it.

2.3 Expectation Propagation

Expectation propagation (EP) (Minka, 2001) is an approximation technique which tunes the parameters of a simpler approximate distribution to approximate the exact posterior distribution of the model parameters given the data. The typical choice for the simpler approximate distribution is the Gaussian distribution.

The Gaussian distribution, also known as the normal distribution, is a continuous probability distribution that is often used as a first approximation to describe real-valued random variables that tend to cluster around a single mean value. The multivariate Gaussian distribution of a random variable $\boldsymbol{\theta}$ of dimension d is defined as follows, i.e.,

$$p(\boldsymbol{\theta}|\boldsymbol{\mu},\Sigma) \equiv \frac{1}{(2\pi)^d\sqrt{|\Sigma|}} \exp\left(-\frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{\theta}-\boldsymbol{\mu})\right)\,,$$

where $\boldsymbol{\mu}$ and Σ are the parameters of the distribution.

Assumed Density Filtering

ADF is an approximation technique in which the terms of the posterior distribution are added one at a time, and in each step the result of the inclusion is projected back into the assumed density. As the assumed density we take the Gaussian, to which we will refer further as q.

The first term which is included is the prior, $q(\boldsymbol{\theta}) = p(\boldsymbol{\theta})$. Next, terms are added one at a time $\tilde{p}(\boldsymbol{\theta}) = \Psi_{ij}(\theta_i, \theta_j)q(\boldsymbol{\theta})$, with $\Psi_{ij}(\theta_i, \theta_j) = p(y_{ij}|\theta_i, \theta_j)$, and at each step the resulting distribution is approximated as closely as possible by a Gaussian $q^{\text{new}}(\boldsymbol{\theta}) = \text{Project}\{\tilde{p}(\boldsymbol{\theta})\}$. Using the Kullback-Leibler (KL) divergence as the measure between the non-Gaussian $\tilde{p}$ and the Gaussian approximation, projection becomes moment matching: the result q^{new} of the projection is the Gaussian that has the first two moments, mean and covariance, the same as $\tilde{p}$. After we add a term and project, the Gaussian approximation changes. We call the quotient between the new and old Gaussian approximation a term approximation, and we use it for it the notation $\tilde{\Psi}_{ij}(\theta_i, \theta_j)$. The steps from above are formalized below.

Iterative Improvement

EP generalizes ADF by performing backward-forward iterations to refine the term approximations until convergence. The final approximation will be independent of the order of incorporating the terms. The algorithm performs the following steps.

1. Initialize the term approximations $\tilde{\Psi}_{ij}(\theta_i, \theta_j)$, e.g., by performing ADF; and compute the initial approximation

$$q(\boldsymbol{\theta}) = p(\boldsymbol{\theta}) \prod_{i \neq j} \tilde{\Psi}_{ij}(\theta_i, \theta_j)\,.$$

2. Repeat until all $\tilde{\Psi}_{ij}$ converge:

   (a) Remove a term approximation $\tilde{\Psi}_{ij}$ from the approximation, yielding

$$q^{\setminus ij}(\boldsymbol{\theta}) = \frac{q(\boldsymbol{\theta})}{\tilde{\Psi}_{ij}(\theta_i, \theta_j)}\,.$$

   (b) Combine $q^{\setminus ij}(\boldsymbol{\theta})$ with the exact factor $\Psi_{ij} = p(y_{ij}|\theta_i, \theta_j)$ to obtain

$$\tilde{p}(\boldsymbol{\theta}) = \Psi_{ij}(\theta_i, \theta_j) q^{\setminus ij}(\boldsymbol{\theta})\,. \tag{2.3}$$

   (c) Project $\tilde{p}(\boldsymbol{\theta})$ into the approximation family

$$q^{\text{new}}(\boldsymbol{\theta}) = \operatorname*{argmin}_{q \in Q} KL[\tilde{p}||q]\,.$$

(d) Recompute the term approximation through the division

$$\tilde{\Psi}_{ij}^{\text{new}}(\theta_i, \theta_j) = \frac{q^{\text{new}}(\boldsymbol{\theta})}{q^{\setminus ij}(\boldsymbol{\theta})} \,.$$

where $KL[\cdot||\cdot]$ is the Kullback-Leibler divergence (Kullback and Leibler, 1951). $KL[P||Q]$ is a non-symmetric measure of the difference between two probability distributions P and Q, more precisely it measures the expected number of extra bits required to code samples from P when using a code based on Q. Typically P represents the "true" distribution of data while Q represents an approximation of P. For continuous random variables, it is defined as

$$KL[P||Q] = \int d\theta p(\theta) \log \frac{p(\theta)}{Q(\theta)} \,.$$

When minimizing the KL divergence in step (c) we can take advantage of the locality property of EP (Seeger, 2002). From Equation (2.3), because the term Ψ_{ij} does not depend on $\boldsymbol{\theta}^{\setminus ij}$, we can rewrite $\tilde{p}$ as:

$$\tilde{p}(\boldsymbol{\theta}) = \tilde{p}(\boldsymbol{\theta}_{\setminus ij}|\theta_i, \theta_j)\tilde{p}(\theta_i, \theta_j) = \tilde{p}(\theta_i, \theta_j) q^{\setminus ij}(\boldsymbol{\theta}_{\setminus ij}|\theta_i, \theta_j) \,.$$

Furthermore we obtain:

$$\begin{aligned} KL[\tilde{p}(\boldsymbol{\theta})||q(\boldsymbol{\theta})] &= KL[\tilde{p}(\theta_i, \theta_j)||q(\theta_i, \theta_j)] \\ &\quad + E_{\tilde{p}(\theta_i, \theta_j)}[KL[q^{\setminus ij}(\boldsymbol{\theta}_{\setminus ij}|\theta_i, \theta_j)||q(\boldsymbol{\theta}_{\setminus ij}|\theta_i, \theta_j)]] \,. \qquad (2.4) \end{aligned}$$

The two terms on the right-hand side can be minimized independently. Minimization of the second term gives:

$$q^{\text{new}}(\boldsymbol{\theta}_{\setminus ij}|\theta_i, \theta_j) = q^{\setminus ij}(\boldsymbol{\theta}_{\setminus ij}|\theta_i, \theta_j) \,. \qquad (2.5)$$

Minimizing the KL divergence for the first term in the right-hand side in Equation (2.4) reduces to matching the moments, mean and covariance, between the 2-dimensional distributions $\tilde{p}(\theta_i, \theta_j)$ and $q(\theta_i, \theta_j)$.

Computational Complexity

Exploiting this locality property, we managed to go from n_{players}-dimensional integrals to 2-dimensional integrals, which can be further reduced to 1 dimension, by rewriting them in the following way (see e.g., the appendix of (Barber and Bishop, 1998)):

$$\langle \Psi(\theta_i, \theta_j) \rangle_{\mathcal{N}(\boldsymbol{m}, \boldsymbol{C})} = \langle F(\boldsymbol{a}\boldsymbol{\theta}_{ij}) \rangle_{\mathcal{N}(\boldsymbol{m}, \boldsymbol{C})} = \langle F(\theta\sqrt{\boldsymbol{a}^T \boldsymbol{C} \boldsymbol{a}} + \boldsymbol{a}^T \boldsymbol{m}) \rangle_{\mathcal{N}(0,1)} \,,$$

where $\boldsymbol{a}$ is the vector $[-1,\ 1]$ if player i is the winner, or $\boldsymbol{a} = [1,\ -1]$ if player j is the winner, $\boldsymbol{\theta}_{ij} = [\theta_i,\ \theta_j]$, F is defined through Equation (2.1), and $\mathcal{N}(\boldsymbol{m}, \boldsymbol{C})$ stands for a

Gaussian with mean $\boldsymbol{m}$ and covariance matrix $\boldsymbol{C}$. Substituting the solution (2.5), we see that the term approximation, in step (d) of the algorithm, indeed only depends on θ_i and θ_j.

We can simplify the computations by using the canonical form of the Gaussian distribution. Because, when projecting, we need the moment form of the distribution, we go back and forth between distributions in terms of moments and in terms of canonical parameters. For a Gaussian, this requires computing the inverse of the covariance matrix, which is of the order n^3_{players}. Since the covariance matrix, when refining the term corresponding to the game between players i and j, changes only for the elements corresponding to players i and j, we can use the Woodbury formula (Press et al., 1992) to reduce the cubic complexity of the matrix inversion to a quadratic one. Thus, the complexity of EP is of the order:

$$\text{Complexity}(\text{EP}) = \mathcal{O}(n_{\text{iterations}} \times n^2_{\text{players}} \times n_{\text{matches}})\,,$$

where $n_{\text{iterations}}$ is the number of iterations back and forth in refining the term approximations. In practice, the number of iterations to converge seems largely independent of the number of players or matches. In our experiments, we needed $n_{\text{iterations}} \approx 5$ to converge.

We will refer to this version of EP as EP-Correlated: by projecting into a non-factorized Gaussian, it takes into account the correlations between the players' strengths.

EP-Independent

The complexity of the EP algorithm can be reduced further if we keep track only of the diagonal elements of the covariance matrix, ignoring the correlations. The matrix inversion has in this case linear complexity. The algorithm is faster and requires less memory.

2.4 Experimental Evaluation

We applied the approximation algorithms, presented in the previous section, to the analysis of a real data set. The data set consists of results of 38538 tennis matches played on ATP events among 1139 players between 1995 and 2006. The goal was to compute ratings for the players based on the match outcomes. The methods described yield a Gaussian distribution of the players' strengths; the mean of the distribution represents our estimate of the players' strengths, the rating, and the variance relates to the uncertainty. Furthermore, we predict results of future games, and estimate the confidence of our predictions. We take as the prior a Gaussian distribution with mean zero and covariance equal to the identity matrix.

Figure 1 shows the empirical distribution of the players' strengths (means of the posterior distribution) in comparison with the average width of the posterior for an individual

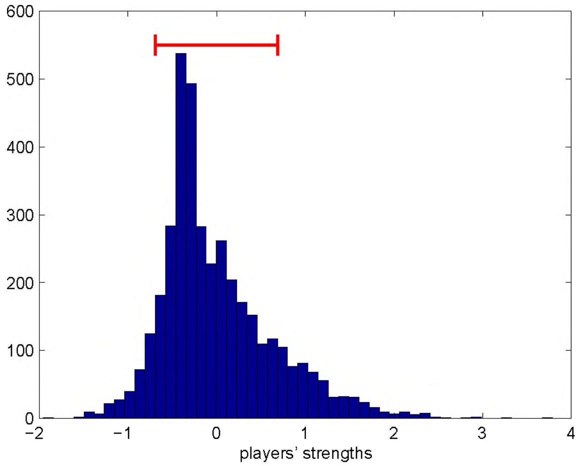

Figure 2.1: A histogram of the players' strengths (means of the posterior distribution) for all years. The bar indicates the average width of the posterior distribution for each of the individual players. The results shown are for EP-Correlated.

player. It can be seen that the uncertainty for individual players is comparable to the diversity between players.

2.4.1 Accuracy

We computed the ratings for the players at the end of each year, based on the matches from that year. Furthermore, based on these ratings we made predictions for matches in the next year: in a match we predicted the player with the highest rating to win.

EP-Correlated versus ADF

We compared the accuracy of the predictions based on EP-Correlated ratings with the ones based on ADF ratings. We divided all joint predictions into 4 categories as shown in Table 2.1. We applied a binomial test on the matches for which the two algorithms gave different predictions to check the significance of the difference in performance (Salzberg, 1997). The p-value obtained for this one-sided binomial test is 3×10^{-14}, which indicates that the difference is highly significant: EP-Correlated performs significantly better than ADF.

EP-Correlated versus EP-Independent

The same type of comparison was performed between EP-Correlated and EP-Independent, the results are shown in Table 2.1. As for the previous comparison, the

Table 2.1: Comparison between EP-Correlated, ADF and EP-Independent based on the number of matches correctly/incorrectly predicted. The EP-C in the table below stands for EP-Correlated.

	ADF		EP-Independent	
	true	false	true	false
EP-C				
true	16636 (54.48%)	2395 (7.81%)	17857 (58.46%)	1174 (3.83%)
false	1902 (6.21%)	9620 (31.50%)	945 (3.09%)	10577 (34.62%)

p-value is very small, 3×10^{-7}: the binomial test suggests that the difference between the two algorithms is again highly significant.

EP-Correlated versus Laplace and ATP Rating

We compared Laplace's method and EP-Correlated to find out that EP-Correlated does slightly, but not significantly better (p-value is 0.3). They disagree on only 0.2% of all matches.

We also compared the accuracy of the predictions based on the EP ratings with the accuracy of the predictions obtained using the ATP ratings at the end of the year. The ATP rating system gives points to players according to the type of the tournament and how far in the tournament they reached. Averaged over all the years, both EP and ATP ratings, give similar accuracy of predictions for the next, about 62%.

2.4.2 Confidence

With a posterior probability over the players' strengths we can compute the confidence of the predictions.

The algorithms presented perform about the same in estimating the confidence. However, they all tend to be overconfident, in the sense that the actual fraction of correctly predicted matches is smaller than the predicted confidence, as indicated by the solid line in the left plot of Figure 2.2. We can correct this by adding noise to the players' strengths, to account for the fact that a player's strength changes over time:

$$\theta_{t+1} = \theta_t + \epsilon$$

where ϵ has mean zero and variance σ^2. To evaluate the confidence estimation, we plot on the right side of Figure 2.2 the Brier score (Brier, 1950) for different values of σ. The optimum is obtained for $\sigma = 1.4$, which then yields the dashed line in the left plot of

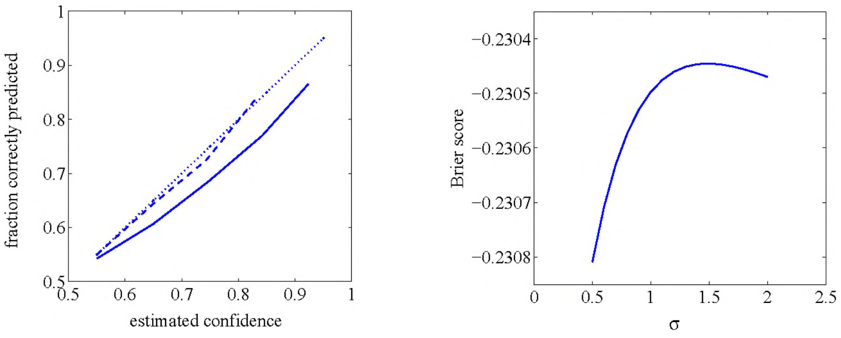

Figure 2.2: Left: the actual fraction of correctly predicted matches as a function of the predicted confidence; without added noise (solid line) and with noise of standard deviation 1.4 added (dashed line); the dotted line represents the ideal case and is drawn for reference. Right: the Brier score for the confidence of the predictions as a function of the standard deviation of the noise added to each player's strength.

Figure 2.2.

2.5 Conclusions and Future Work

Based on the experimental results reported in this study we draw the conclusion that EP-Correlated performs better in doing predictions for this type of data set than its modified versions, ADF and EP-Independent. Further experiments should reveal whether this also applies to other types of data.

Our results are generalizable to more complex models, e.g. including dynamics over time, which means that a players rating in the present is related to his performance in the past (Glickman, 1993); and team effects: a player's rating is inferred from team performance (Herbrich et al., 2007; Huang et al., 2005). Specifically for tennis, the more complex models should also incorporate the effect of surface because the performance of tennis players in a match is influenced by the type of surface they play on (grass, clay, hard court, indoor). In this work we considered the most basic probabilistic rating model; this model performs as good as the ATP ranking system. We would expect that the more complex models could outperform ATP.

Acknowledgments

The statistical information contained in the tennis data set has been provided by and is being reproduced with the permission of ATP Tour, Inc., who is the sole copyright owner of such information. We would like to thank Franc Klaassen for pointing us in the right direction.

Part I

Preference Learning

Chapter 3

Multi-Task Preference Learning with an Application to Hearing-Aid Personalization

We present an EM-algorithm for the problem of learning preferences with semi-parametric models derived from Gaussian processes in the context of multi-task learning. We validate our approach on an audiological data set and show that predictive results for sound quality perception of hearing-impaired subjects, in the context of pairwise comparison experiments, can be improved using a hierarchical model.[1]

[1] This work has been published as: A. Birlutiu, P. Groot and T. Heskes "Multi-Task Preference Learning with an Application to Hearing-Aid Personalization" *Neurocomputing, vol 73, issues 7-9, pages 1177–1185, 2010.* It is based on the work which appeared as: A. Birlutiu, P. Groot and T. Heskes "Multi-Task Preference Learning with Gaussian Processes" *Proceedings of the 17th European Symposium on Artificial Neural Networks, pages 123–128, 2009.*

3.1 Introduction

There has been a wide interest in learning the preferences of people within artificial intelligence research in the last years (Doyle, 2004). Preference learning is a crucial aspect in modern applications such as decision support systems (Chajewska et al., 2000), recommender systems (Blythe, 2002; Blei et al., 2003), and personalized devices (Clyde et al., 1993; Heskes and de Vries, 2005).

It is important to optimize the preference learning process in terms of cost/time invested. Many machine learning techniques especially designed for optimizing the learning process, such as multi-task learning, have been little explored in the context of preference learning. Multi-task learning is especially suited to the situation in which data for a specific single scenario is scarce, but data is already available from similar scenarios. An example is evaluating sound quality with hearing aids: we have gathered sound evaluations for quite some subjects, information that we would like to exploit when learning a model for a new subject.

The aim of this work is to apply multi-task learning to the context of preference learning. We consider the problem of learning people's preferences not as an individual problem, but in the context of learning from similar tasks with multiple subjects. In this way, the model of different subjects can regularize and influence each other. We demonstrate the usefulness of our model on an audiological data set. We show that the process of learning preferences can be significantly improved by using a hierarchical semi-parametric model based on Gaussian processes.

3.1.1 Related Work

In this section we review some studies from preference learning and multi-task learning related to the work presented in this chapter.

Preference Learning

Preference learning has recently received much attention in the machine learning community (Fürnkranz and Hüllermeier, 2005). In the literature, two approaches are mainly used for representing preference information: *i)* binary preference predicates and *ii)* scoring methods (utility functions). The first category aims at obtaining a ranking of instances from a set of pairwise preferences by solving an augmented binary classification problem (Herbrich et al., 1998; Aiolli and Sperduti, 2004) or by decomposing the problem into multiple binary classification problems (Fürnkranz and Hüllermeier, 2003). The second category uses regression to map instances to target valuations for direct ranking (Caruana et al., 1996; Crammer and Singer, 2001; Chu and Ghahramani, 2005a; Brochu et al., 2008). We focus on the second approach by modeling utility functions using Gaussian processes (GPs). By formulating the preference elicitation process as a probabilistic Bayesian learning problem, one can deal with inconsistencies in subjects' responses as well as learn biases the subjects may have. GPs have been around for quite some time (Kimmeldorf and Wahba, 1970; Blight and Ott, 1975), nevertheless, their applications have increased considerably over the recent years and is still the focus of much research (Rasmussen and Williams, 2006). Only recently, GP models have been applied to the problem of eliciting people's preferences (Chu and Ghahramani, 2005a; Brochu et al., 2008) or eliciting probability distributions from expert's opinions (Gosling, 2005; Gosling et al., 2007; Moala and O'Hagan, 2009).

Multi-Task Learning

The basic idea in multi-task learning is that models learned on different scenarios have parts in common. In a Bayesian framework this often boils down to sharing a hierarchical prior (Bakker and Heskes, 2003; Evgeniou et al., 2005; Xue et al., 2007). Typical application scenarios for multi-task learning are in the area of recommender systems (Blei et al., 2003; Marlin, 2003), which combine content information (e.g., features of items) with collaborative information (data from other subjects) (Chapelle and Harchaoui, 2005; Yu et al., 2003). Multi-task learning with Gaussian processes has recently received attention (Schwaighofer et al., 2005; Yu et al., 2005; Bonilla et al., 2008; Platt et al., 2002). This work is an extension of the semi-parametric multi-task models for regression based on Gaussian processes introduced by (Schwaighofer et al., 2005; Yu et al., 2005) to the case of learning from qualitative preference statements.

3.1.2 Outline of the Chapter

The rest of this chapter is organized as follows. Section 3.2 introduces the probabilistic choice model which represents how subjects choose among a finite set of alternatives. The model assumes a latent utility function that represents a person's preferences. Section 3.3 presents three representations for utility functions: *i)* A parametric representation in which multi-task learning can be easily implemented; *ii)* A non-parametric Gaussian process representation; *iii)* A semi-parametric dual representation based on Gaussian processes. Section 3.4 describes Bayesian learning of the individual utility function. Section 3.5 presents multi-task preference learning. We introduce a hierarchical extension to the Bayesian framework and use the Expectation Maximization algorithm for learning a hierarchical prior. Section 3.6 reports experimental results with the hierarchical model for learning subject preferences in an audiological context. Section 3.7 presents our conclusions and directions for future work. The appendices give details about the algorithms developed in this chapter.

3.1.3 Notation

We denote vectors and matrices with boldface type and their components with regular type, i.e., vector $\boldsymbol{x}$ with components x_n, and matrix $\boldsymbol{K}$ with elements K_{ij}. The notation

$\mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is used for a multivariate Gaussian with mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\Sigma}$. The transpose of a matrix $\boldsymbol{M}$ is denoted by $\boldsymbol{M}^T$. The zero vector and identity matrix are denoted by 0 and $\boldsymbol{I}$, respectively.

3.2 Probabilistic Choice Models

This section introduces the probabilistic choice model which gives a representation of how people choose among a finite set of alternatives. Let $X = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_I\}$ be a set of I distinct inputs. Typically every input is represented by an N-dimensional vector of features ($\boldsymbol{x}_i \in \mathbb{R}^N$). We consider M subjects. Let D_m be a set of J_m observed preference comparisons over instances in X corresponding to subject $m = 1, \dots, M$

$$D_m = \{(\boldsymbol{x}_{i_1^m(j)}, \dots, \boldsymbol{x}_{i_A^m(j)}, a_j^m) | 1 \le j \le J_m, a_j^m \in \{1, \dots, A\}\}, \tag{3.1}$$

with $i_1^m, \dots, i_A^m : \{1, \dots, J_m\} \to \{1, \dots, I\}$ index functions such that $i_1^m(j)$ represents the first input presented in the jth preference comparison to subject m, and $a_j^m = a$ means that alternative $\boldsymbol{x}_{i_a^m(j)}$ is preferred from the A inputs presented to subject m in the jth comparison. We consider a version of this setup in which the preference data of each subject uses the same set of inputs X, which is known beforehand and remains fixed. This is the standard setup in marketing applications of preference modeling where the same choice panel questions are given to many individual consumers, each subject provides his/her own preferences, and we assume that there is some similarity among the preferences of the subjects in the general sense that people have some common preferences.

The preference observations from the comparisons described above can be modeled using probabilistic choice models. The main idea behind probabilistic choice models is to assume a latent utility function value $U_m(\boldsymbol{x})$ associated with each input $\boldsymbol{x}$ which captures the preference of subject m for $\boldsymbol{x}$. In the ideal case the latent function values are consistent with the preference observations, which in probabilistic terms can be written as $p(a_j = a|\boldsymbol{x}_{i_1(j)}, \dots, \boldsymbol{x}_{i_A(j)}, U_m) = 1$ if $U_m(\boldsymbol{x}_{i_a(j)}) \ge U_m(\boldsymbol{x}_{i_{a'}(j)})$ when $a' \neq a$. In practice, however, subjects are often inconsistent in their responses. A very inconsistent subject will have a high uncertainty associated with the utility function; this uncertainty is directly taken into account in the probabilistic framework. A standard modeling assumption (Bradley and Terry, 1952; Kanninen, 2002; Glickman and Jensen, 2005) is that the subject's decision in such a forced-choice comparison follows a multinomial logistic model, which is defined as

$$p(a_j = a|\boldsymbol{x}_{i_1(j)}, \dots, \boldsymbol{x}_{i_A(j)}, U_m) = \frac{\exp\left[U_m(\boldsymbol{x}_{i_a(j)})\right]}{\sum_{a'=1}^{A} \exp\left[U_m(\boldsymbol{x}_{i_{a'}(j)})\right]}. \tag{3.2}$$

For pairwise comparisons, i.e., the subject choosing between one of two presented alternatives, Equation (3.2) is known as the Bradley-Terry model (Bradley and Terry, 1952).

The Bradley-Terry model using a dichotomous response scale $\{$*worse, better*$\}$ can be extended to a polytomous response scale, such as, for instance, $\{$*much worse, worse, equal, better, much better*$\}$. The polytomous response scale results in more information from a comparison than a dichotomous response scale and can be modeled using a polytomous Rasch model (Alagumalai et al., 2005). The optimal response scale, however, depends on the application domain. The polytomous scale cannot be applied in some domains. For example, in the audiological domain that we consider in this work it is standard practice to use forced-choice pairwise comparisons using a response scale with two or three items; more alternatives or a larger response scale would make the preference elicitation process too tiresome for the subject.

An alternative to the model from Equation (3.2) is the multinomial probit model, which has been used before to learn from pairwise comparisons, for example in (Chu and Ghahramani, 2005a; Brochu et al., 2008). The two models, logistic and probit, give similar predictions, however, for $A \geq 3$ the probit model is more difficult to handle (Kropko and Rabinowitz, 2008). For this study we will use the multinomial logistic model.

In this probabilistic framework, learning the preferences of a subject m reduces to learning the corresponding utility function U_m. The goal of this work is to learn the utility functions corresponding to different subjects jointly by sharing information between them.

3.3 Models for Utility Functions

This section discusses three representations for the utility functions:

1. A parametric representation in which multi-task learning is naturally obtained by introducing a joint prior over parameters (Section 3.3.1).

2. A non-parametric representation using Gaussian processes (Section 3.3.2). Multi-task learning is in this case arguably more complicated since here one has to consider a joint prior over functions.

3. A semi-parametric dual representation of the utility function based on Gaussian processes (Section 3.3.3). This dual representation has a parametric form on which multi-task learning can be easily implemented by employing the theory of hierarchical modeling for parametric models. We show in Appendix 3.8 that this representation preserves properties of the non-parametric Gaussian process representation.

For simplicity of notation we omit, in this section and the next one, the superscript m when referring to the individual utility function.

3.3.1 Parametric Models

The utility function in the parametric representation is a fixed model $U(\boldsymbol{x}, \boldsymbol{\theta})$ in which the vector of parameters $\boldsymbol{\theta}$ captures the preferences of the subject. To learn a subject's preferences we need to learn the parameter $\boldsymbol{\theta}$. Multi-task learning is implemented by introducing a prior distribution over $\boldsymbol{\theta}$. This prior is learned from the data available from all subjects. The parametric representation is rather limited as the model $U(\boldsymbol{x}, \boldsymbol{\theta})$ is predefined.

3.3.2 Non-Parametric Models

The main advantage of using the Gaussian process formalism in our framework is that it allows modeling the utility function in a non-parametric way, allowing more flexibility than a fixed parametric model. Furthermore, the computational complexity of GPs is independent of the dimension of the data points but dependent on the number of them; this is an advantage when the number data points is less than the feature dimension.

A Gaussian process (GP) (Rasmussen and Williams, 2006) is a collection of random variables any finite number of which have a joint Gaussian distribution. In our case the random variables are the output values of the utility function and we identify the utility function U with a finite vector $\boldsymbol{U}$. Following the approach of (Chu and Ghahramani, 2005a) for learning preferences with GPs, we define a GP prior over the utility function, i.e., given $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_I\}$, the joint distribution over the utility function values is a multivariate Gaussian distribution

$$\boldsymbol{U} = \{U(\boldsymbol{x}_1), \ldots, U(\boldsymbol{x}_I)\} \sim \mathcal{N}(\boldsymbol{m}, \boldsymbol{K}) \,. \tag{3.3}$$

The covariance matrix $\boldsymbol{K}$ is generated by a kernel function κ, $\boldsymbol{K}_{ij} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Possible choices for κ are, for example, the linear kernel κ_{Linear} or the Gaussian kernel κ_{Gauss} defined as follows

$$\begin{aligned} \kappa_{\text{Linear}}(\boldsymbol{x}, \boldsymbol{x}') &= \sum_{n=1}^{N} x_n x'_n \,, \\ \kappa_{\text{Gauss}}(\boldsymbol{x}, \boldsymbol{x}') &= \exp\left(-\frac{1}{2\ell^2} \sum_{n=1}^{N} (x_n - x'_n)^2 \right) \,. \end{aligned} \tag{3.4}$$

where ℓ in Equation (3.4) is a length-scale parameter.

The choice of the kernel function depends on our assumptions about properties of the actual utility function, where actual utility function refers to how the people evaluate utilities in reality. In some domains a linear kernel can be good enough; in other domains when a more complex form of the utility function is needed a Gaussian kernel is more suited.

A Gaussian process is in fact equivalent to a Bayesian interpretation of linear regression (Rasmussen and Williams, 2006). Let

$$U(\boldsymbol{x}) = \boldsymbol{\phi}(\boldsymbol{x})^T\boldsymbol{\alpha} = \sum_i \alpha_i \phi_i(\boldsymbol{x})$$

be a linear combination of (a possibly infinite number of) basis functions $\phi_i(\cdot)$. If the weight vector $\boldsymbol{\alpha}$ is drawn from a Gaussian distribution this induces a probability distribution over functions $U(\cdot) = \boldsymbol{\phi}(\cdot)^T\boldsymbol{\alpha}$. This distribution is a Gaussian process. From this analogy it follows that a linear kernel is essentially equivalent to a linear parametric model.

A graphical representation of preference learning using the GP representation of the utility function, for the case of pairwise comparisons, is given on the left-hand side of Figure 3.1. What is inside the plate corresponds to the utility model of one subject. The response a_1 given by a subject to the comparison $\{\boldsymbol{x}_1, \boldsymbol{x}_2\}$ depends on the values $U(\boldsymbol{x}_1)$ and $U(\boldsymbol{x}_2)$ of the subjects' utility function. The goal is to learn the latent utility function in order to predict the outcomes of the unobserved comparisons (a_2) based on the observed ones (a_1 and a_3). The utility function values $U(\boldsymbol{x}_1)$, $U(\boldsymbol{x}_2)$, and $U(\boldsymbol{x}_3)$ (corresponding to the same subject) are correlated in the GP formalism since they depend on each other through the kernel (illustrated by the solid bar between them). Furthermore, the utility models for each subject depend on the same prior estimates $\boldsymbol{m}$ and $\boldsymbol{K}$.

3.3.3 Semi-Parametric Models

Inspired by the representer theorem (Schölkopf et al., 2001) — that links the GP to a semi-parametric model — we use a dual representation for the utility function. The dual representation has a semi-parametric form on which multi-task learning can be easily implemented by employing the theory of hierarchical modeling for parametric models. In the dual representation the utility function $U(\boldsymbol{x})$, $\boldsymbol{x} \in X$ is defined as follows

$$U(\boldsymbol{x}) = \sum_{i=1}^{I} \alpha_i \kappa(\boldsymbol{x}, \boldsymbol{x}_i)\ , \tag{3.5}$$

where $\boldsymbol{x}_i \in X$, κ is the kernel function, and $\boldsymbol{\alpha} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Equation (3.5) expresses the utility function as a linear combination of basis functions defined by a kernel centered on the data points. The vector of parameters $\boldsymbol{\alpha}$ with dimension I — the number of inputs — captures the information collected from the data set related to a subject. Even though $\boldsymbol{\alpha}$ is a parameter, it does not induce a fixed form for the utility function — as the representation of the utility function in Equation (3.5) is data dependent. The parameter $\boldsymbol{\alpha}$ can give further insights about the importance of each data point and can be used to obtain sparseness and detect outliers (Gestel et al., 2002). The representation of the utility function from Equation (3.5) is similar to the Relevance Vector Machine (RVM) (Tipping, 2001); the vector of parameters $\boldsymbol{\alpha}$ can give information about which data points (if any)

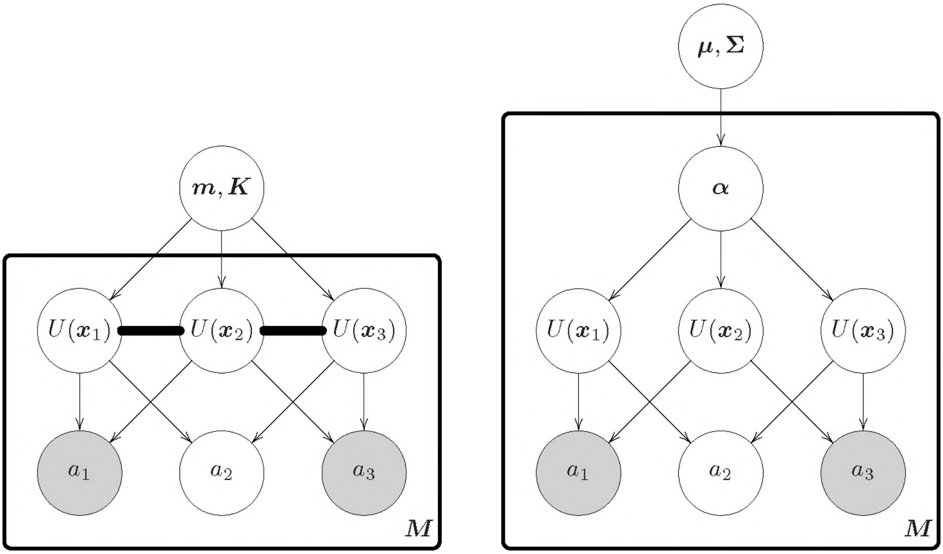

Figure 3.1: Preference learning based on two representations of the utility function. What is inside the plate corresponds to the utility model of one subject. Left: non-parametric Gaussian process (cf. Section 3.3.2). Right: semi-parametric model derived from a Gaussian process (cf. Section 3.3.3). The observation a_1 of the comparison $\{\boldsymbol{x}_1, \boldsymbol{x}_2\}$ depends on the values $U(\boldsymbol{x}_1)$ and $U(\boldsymbol{x}_2)$ of the subjects' utility function. The goal is to learn the latent utility function U in order to predict the outcomes of the unseen comparisons (a_2) based on the observed ones $(a_1$ and $a_3)$.

are relevant / prototypes. Furthermore, based on $\boldsymbol{\alpha}$ we can decide which data points to query for labeling next such as to obtain maximum information in an experimental design / active learning approach. When the number of data points is large sparsity may be desired for the parameter $\boldsymbol{\alpha}$. In that case a Laplacian rather than a Gaussian prior may be more suited.

A graphical representation of preference learning using the dual representation of the GP, for the case of pairwise comparisons, is given on the right-hand side of Figure 3.1. Analogous to the left-hand side of the figure, what is inside the plate corresponds to the utility model of one subject. The difference with the left-hand side is that the utility function of one subject is determined by the parameter $\boldsymbol{\alpha}$. Note that in this representation the utility function values $U(\boldsymbol{x}_1), \ldots, U(\boldsymbol{x}_I)$ are conditionally independent given $\boldsymbol{\alpha}$. Furthermore, the parameters corresponding to the utility models of different subjects depend on the hierarchical prior estimates $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$.

The correspondence between the dual representation considered in this section and the primal formulation considered in Section 3.3.2 is discussed in the Appendix 3.8.

3.4 Learning Utility Functions

In order to learn a subject's preferences, we treat the vector of parameters $\boldsymbol{\alpha}$ as a random variable. After performing experiments and observing their outcomes, the posterior distribution over $\boldsymbol{\alpha}$ is computed using Bayes' rule,

$$\begin{aligned} p(\boldsymbol{\alpha}|D, \boldsymbol{\mu}, \boldsymbol{\Sigma}) &\propto p(\boldsymbol{\alpha})p(D_A|D_X, \boldsymbol{\alpha}) \\ &= p(\boldsymbol{\alpha}) \prod_{j=1}^{J} p(a_j|\boldsymbol{x}_{i_1(j)}, \ldots, \boldsymbol{x}_{i_A(j)}, \boldsymbol{\alpha}) \end{aligned}$$

where $D = \{D_X, D_A\}$ represents the inputs D_X and the outputs D_A, with inputs $D_X = \{(\boldsymbol{x}_{i_1(j)}, \ldots, \boldsymbol{x}_{i_A(j)}), j = 1, \ldots, J\}$, preference observations $D_A = \{a_j, j = 1, \ldots, J\}$, and likelihood terms of the form given in Equation (3.2). We make the common assumption of a Gaussian prior distribution. The entire posterior distribution over $\boldsymbol{\alpha}$ is needed in the multi-task learning framework presented in the next section. As the exact posterior distribution is intractable we approximate it with a Gaussian distribution. The Gaussian approximation is a good approximation of the posterior distribution because with few data points the posterior is close to a Gaussian due to the prior, and with many data points the posterior approaches again a Gaussian as a consequence of the central limit theorem (Bishop, 2006). Two types of approaches exist for approximating the posterior distribution *i)* deterministic methods for approximate inference (e.g., Laplace's method (Mackay, 2002), Expectation Propagation (Minka, 2001)); *ii)* methods based on sampling. Since the sampling methods are computationally expensive and the deterministic methods are known to be very accurate for these types of models (Glickman and Jensen, 2005) we focus on deterministic methods. In Appendix 3.9 we present two methods for approximate inference in the probabilistic choice models described in Section 3.2.

3.5 Multi-Task Preference Learning

In this section we consider learning the utility function in a multi-task setting. Consider M tasks, each task corresponding to one subject. Let D_m be the data set of subject m of the form given in Equation (3.1). The goal is the joint learning of the latent utility functions U_m for $m = 1, \ldots, M$ by sharing information between tasks. We implement the multi-task learning using Bayesian hierarchical modeling. We derive a method for gathering data from previous subjects into a single distribution that is used as a prior distribution for a new subject.

The utility function U_m is parameterized in terms of $\boldsymbol{\alpha}_m$. The inference problems for all the tasks are coupled by having the same prior over the parameters $\boldsymbol{\alpha}_m$, i.e., we set $p(\boldsymbol{\alpha}_m) = \mathcal{N}(\boldsymbol{\alpha}_m|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ a Gaussian prior with the same $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ for all subjects. The posterior distribution for each task is assumed to be (close to) a Gaussian with mean $\boldsymbol{\mu}_m$

and variance $\boldsymbol{\Sigma}_m$. A penalized version of the maximum likelihood values for the prior mean $\boldsymbol{\mu}$ and the prior variance $\boldsymbol{\Sigma}$, can be obtained by specifying a hyper prior distribution $p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. We assume a normal-inverse-Wishart distribution as the hyper prior since it is the conjugate prior for the multivariate distribution

$$p(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathcal{N}(\boldsymbol{\mu}|\boldsymbol{\mu}_0, \frac{1}{\pi}\boldsymbol{\Sigma})\,\mathcal{IW}(\boldsymbol{\Sigma}|\tau, \boldsymbol{\Sigma}_0)\,.$$

The normal-inverse-Wishart distribution is specified by means of the scale matrix $\boldsymbol{\Sigma}_0$ with precision τ and mean $\boldsymbol{\mu}_0$ with precision π. We assume that $\boldsymbol{\mu}_0 = 0$ and $\boldsymbol{\Sigma}_0 = \boldsymbol{I}$.

3.5.1 EM Algorithm for Learning the Hierarchical Prior

The hierarchical prior is obtained by maximizing the penalized loglikelihood of all data. This optimization is performed by applying the Expectation Maximization algorithm (Gelman et al., 2003; Yu et al., 2005) which reduces to the iteration (until convergence) of the following two steps.

E-step: For each subject m estimate the sufficient statistics (mean $\boldsymbol{\mu}_m$ and covariance matrix $\boldsymbol{\Sigma}_m$) of the posterior distribution over $\boldsymbol{\alpha}_m$ given the current estimates $\boldsymbol{\mu}^{(t)}$ and $\boldsymbol{\Sigma}^{(t)}$ of the hierarchical prior. The E-step is performed using one of the inference techniques mentioned in Appendix 3.9.

M-step: Re-estimate the parameters of the hierarchical prior:

$$\begin{aligned}
\boldsymbol{\mu}^{(t+1)} &= \frac{1}{M}\sum_{m=1}^{M}\boldsymbol{\mu}_m\,,\\
\boldsymbol{\Sigma}^{(t+1)} &= \frac{1}{\tau+M}\Bigg[\pi\boldsymbol{\mu}^{(t+1)}\boldsymbol{\mu}^{(t+1)^T} + \sum_{m=1}^{M}\boldsymbol{\Sigma}_m +\\
&\qquad \boldsymbol{I} + \sum_{m=1}^{M}(\boldsymbol{\mu}_m - \boldsymbol{\mu}^{(t+1)})(\boldsymbol{\mu}_m - \boldsymbol{\mu}^{(t+1)})^T\Bigg]\,,
\end{aligned} \tag{3.6}$$

where $\boldsymbol{\mu}_m$ and $\boldsymbol{\Sigma}_m$ are the posterior mean and variance for subject m computed based on the previous prior mean $\boldsymbol{\mu}^{(t)}$ and variance $\boldsymbol{\Sigma}^{(t)}$. The update equation for the variance relates to the variance of a mixture model: the last term on the right-hand side of Equation (3.6) computes the variance between the individual means and the second term the individual variances in the mixture components.

In each E-step the distribution over $\boldsymbol{\alpha}_m$ is approximated with a multivariate Gaussian. Therefore, in our hierarchical framework each utility function U_m can still be interpreted as an (approximate) Gaussian process (cf. the equivalence stated in Appendix 3.8). The derivation of the EM algorithm is given in Appendix 3.10.

3.6 Experimental Evaluation

We validated our approach for hierarchical preference learning on an audiological data set. The audiological data set consists of evaluations of sound quality from 14 normal-hearing and 18 hearing-impaired subjects which we considered as two separate data sets. Each person was subjected to 576 pairwise comparison listening experiments of the form $(\boldsymbol{x}_1, \boldsymbol{x}_2, a)$, where $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ represent two output sounds obtained by processing the same input sound using two different parameter settings of the hearing aid and $a \in \{1, 2\}$ denotes which of the two alternatives was preferred by the subject. The preference data collected in the audiological experiment is related to the overall evaluation of the quality of the sound stimulus presented. Research in audiology (Arehart et al., 2007) shows that intelligibility is an important factor in the perceptual judgment of sound quality by subjects. In order to increase intelligibility the sound stimulus is being processed, e.g., by reducing noise or increasing the volume in some frequency bands. Sound processing adds, however, different kinds of distortions to the output signal listened to by a subject, thus degrading the comfort and as a result the overall sound quality. The way in which people perceive the quality of the processed sound stimulus varies, even for normal-hearing subjects. A detailed description of the data set can be found in (Arehart et al., 2007).

The goal of the validation was to check whether the preferences of a new subject can be learned more accurately by using the available preferences from other subjects. To answer this question we compared the hierarchical model with a method which assumes no prior information and a pooling method. In each simulation one subject was left-out (the test subject). The data set for the test subject was split into training (used for learning preferences) and testing (the accuracy of the predictions on the test data was used as a measure of how much we learned about subject's preferences). Each subject was characterized by a utility function which describes his/her preferences. The utility function was parameterized by the vector $\boldsymbol{\alpha}$ as discussed in Section 3.3.

The hierarchical model uses the EM algorithm described in the previous section to gather data from the group of subjects into a probability distribution over $\boldsymbol{\alpha}$, which was used as the starting prior for the test subject. The values of the hyper-parameters of the hierarchical prior were set to $\pi = 0$ and $\tau = 1$. The non-informative prior method uses a flat prior (a Gaussian distribution with a large variance) which assumes no information about the test subject's preferences. The pooling method pools all data together and a single model is learned based on all but the test subject after which data from the test subject is added one by one.

The plots in Figure 3.2 compare the accuracy obtained using the hierarchical model with a Gaussian kernel versus the non-informative method and the pooling method. The data for each test subject was split into 5 equal sized blocks, one block was considered for training and the rest for testing. The results were averaged over the splits and within each group of normal-hearing and hearing-impaired subjects. The pooling method works good for normal-hearing subjects but performs bad for the hearing-impaired subjects. This is

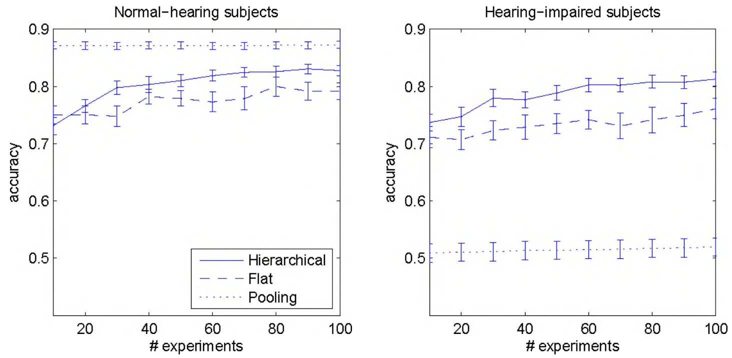

Figure 3.2: Accuracy for the hierarchical model, pooling method and the non-informative method as a function of the number of data-points included for the test subject for normal-hearing subjects (left-hand side plot) and for hearing-impaired subjects (right-hand side plot). A Gaussian kernel with $\ell = 1$ was used. The bars indicate the standard error of the mean.

as expected since hearing-impaired subjects have large variations in their preferences for speech quality, whereas for normal-hearing subjects the variations are smaller. There is no change in the accuracy of the pooling method as a function of the number of experiments / data points because the few extra data points of the test subject, compared with all the data points from the other subjects, do not really affect the estimate. The variance is higher within the hearing-impaired group due to variations in the audiological conditions.

Furthermore, in order to determine which of the kernels for the hierarchical model is more suited for this data set we compared the accuracy obtained with a Gaussian kernel versus a linear kerne. We used the same setup as in the previous comparison. The results are shown in Figure 3.3 The Gaussian kernel appears to be better overall than the linear kernel for this data set.

3.7 Conclusions and Future Work

We have introduced a hierarchical modeling approach for learning related functions of multiple subjects performing similar tasks using Gaussian processes. A hierarchical prior was used which enforces a similar structure for the utility function of each individual subject.

We are interested in further improvements of the model. Particularly, we plan to investigate how to select, in an active way, the most informative experiments in order to learn subjects' preferences. Furthermore, it might be interesting to automatically cluster, either beforehand or as an integral part of the algorithm, the subjects into groups with

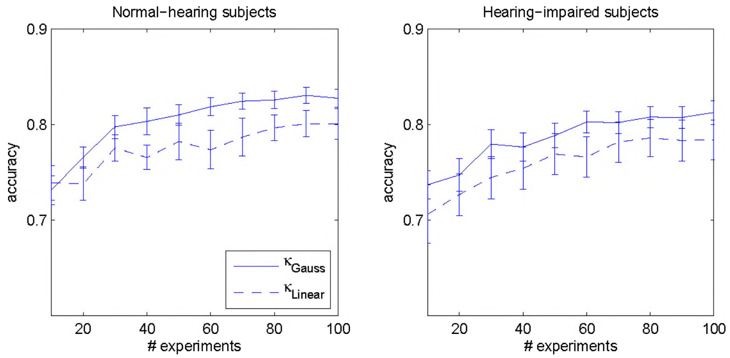

Figure 3.3: Accuracy of the hierarchical model with a Gaussian kernel (solid line) versus a linear kernel (dashed line) for normal-hearing subjects (left-hand side plot) and for hearing-impaired subjects (right-hand side plot). The bars indicate the standard error of the mean.

similar behavior. For the audiological data set used in this study we manually clustered the data into two sets of normal-hearing and hearing-impaired subjects since the plots of the maximum-likelihood estimates of the subjects' parameters did not show the need for further subclustering. For other data sets one could consider replacing the Gaussian prior with a Dirichlet prior (Xue et al., 2007). This would lead to automatically clustering of the subjects and would enable the algorithm to identify relatedness among the subjects. In this way the hierarchical prior is learned using those subjects that are more related to the test subject. Another alternative for future research is to compare our approach to other multi-task learning approaches, for example, (Micchelli and Pontil, 2005) and (Bonilla et al., 2008).

3.8 Appendix A: Equivalence of the GP Representations

We analyze the relation between the non-parametric Gaussian process representation and the semi-parametric dual representation of the utility function given in Sections 3.3.2 and 3.3.3, respectively. We show below that the two representations induce the same Gaussian distribution over the utility function for any subset $Z \subseteq X$.

Let $\boldsymbol{U}_Z$ be the vector $\boldsymbol{U}$ restricted to the index set Z, and let $\boldsymbol{\alpha} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ be a Gaussian distributed variable. From Equation (3.5) follows that $\boldsymbol{U}_Z$ is a linear combination of Gaussian distributed variables and has therefore a multivariate Gaussian distribution.

The distribution over $\boldsymbol{\alpha}$ induces the following distribution over $\boldsymbol{U}_Z$

$$\boldsymbol{U}_Z \sim \mathcal{N}(\boldsymbol{K}(Z,X)\boldsymbol{\mu},\, \boldsymbol{K}(Z,X)\boldsymbol{\Sigma}\boldsymbol{K}(Z,X)^T)\,. \tag{3.7}$$

The two Gaussian distributions from Equations (3.7) and (3.3) restricted to $Z \subseteq X$ are the same when

$$\begin{aligned}\boldsymbol{K}(Z,X)\boldsymbol{\mu} &= \boldsymbol{m}_Z\,,\\ \boldsymbol{K}(Z,X)\boldsymbol{\Sigma}\boldsymbol{K}(Z,X)^T &= \boldsymbol{K}(Z,Z)\,,\end{aligned}$$

with $\boldsymbol{m}_Z$ the vector $\boldsymbol{m}$ restricted to the index set Z. This leads to the following result.

Theorem 3.8.1 (Primal-Dual Equivalence) *The utility model* $U(\boldsymbol{x}) = \sum_{i=1}^{I} \alpha_i \kappa(\boldsymbol{x}, \boldsymbol{x}_i)$ *with* $\boldsymbol{\alpha} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ *and* $\boldsymbol{x} \in X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_I\}$ *is equivalent to the standard GP formulation* $\boldsymbol{U} \sim \mathcal{N}(\boldsymbol{m}, \boldsymbol{K})$ *when*

$$\boldsymbol{K}\boldsymbol{\mu} = \boldsymbol{m}\,, \tag{3.8}$$

$$\boldsymbol{\Sigma} = \boldsymbol{K}^+\,, \tag{3.9}$$

with $\boldsymbol{K}^+$ *the pseudo-inverse of* $\boldsymbol{K}$.

Proof: Equation (3.9) follows directly from the definition of the pseudo-inverse,

$$\boldsymbol{K}\,\boldsymbol{K}^+\,\boldsymbol{K} = \boldsymbol{K}\,.$$

If $\boldsymbol{K}$ is invertible, for any $\boldsymbol{m}$ there exists a $\boldsymbol{\mu}$ that satisfies Equation (3.8). This property does not necessarily hold if $\boldsymbol{K}$ is not invertible. □

The equivalence between the primal and the dual representation holds when we apply the model in a transductive setting, i.e., only to inputs $\boldsymbol{x} \in X$. The two representations are not equivalent anymore when we apply the model to a new test point $\boldsymbol{x}' \notin X$.

3.9 Appendix B: Methods for Approximate Inference

We present two methods for approximate inference for the probabilistic choice models described in Section 3.2.

Laplace's method

In the Laplace approximation (Mackay, 2002) the posterior distribution is approximated by a Gaussian with mean equal to the maximum a posteriori solution

$$\boldsymbol{\theta}^* \equiv \operatorname*{argmax}_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$$

where

$$L(\boldsymbol{\theta}) = \sum_{j=1}^{J} \log p(a_j | \boldsymbol{x}_{i_1(j)}, \ldots, \boldsymbol{x}_{i_A(j)}, \boldsymbol{\theta}) - \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta} - \boldsymbol{\mu})$$

and variance equal to the inverse of the Hessian, the second derivative of $L(\boldsymbol{\theta})$.

ADF and EP

Assumed Density Filtering and Expectation Propagation (Opper and Winther, 2001; Minka, 2001) are approximation techniques in which the terms of the likelihood corresponding to the observed data are added in a sequential way. At each step the result of the inclusion is projected back into the assumed density. We choose for the assumed density a Gaussian distribution. The projection is done by minimizing the Kullback-Leibler divergence between the real posterior and the approximate density. For assumed densities in the exponential family this reduces to moment matching, i.e., the new approximate posterior is the Gaussian which has the same mean and variance as the real posterior.

For a linear utility model $U(\boldsymbol{x}, \boldsymbol{\theta}) = \Phi(\boldsymbol{x})^T\boldsymbol{\theta}$ the computation of the posterior approximation can be simplified from N dimensions (where N is the dimension of $\boldsymbol{\theta}$) to 1 dimension. The likelihood function depends on $\boldsymbol{\theta}$ only through its projection onto a particular direction defined by the input $\Phi(\boldsymbol{x})$. The key idea is then to decompose $\boldsymbol{\theta}$ such that one of the components of the decomposition is perpendicular to $\boldsymbol{\Sigma}^{1/2}\Phi(\boldsymbol{x})$. The computations needed for the normalization constant can be simplified as follows

$$\left\langle g\left(\Phi(\boldsymbol{x})^T\boldsymbol{\theta}\right)\right\rangle_{\mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu},\boldsymbol{\Sigma})} = \left\langle g\left(\eta\sqrt{\Phi(\boldsymbol{x})^T\boldsymbol{\Sigma}\Phi(\boldsymbol{x})} + \Phi(\boldsymbol{x})^T\boldsymbol{\mu}\right)\right\rangle_{\mathcal{N}(\eta|0,1)}$$

where g is the logistic function

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

and

$$\left\langle g\left(\Phi(\boldsymbol{x})^T\boldsymbol{\theta}\right)\right\rangle_{\mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu},\boldsymbol{\Sigma})} = \int g\left(\Phi(\boldsymbol{x})^T\boldsymbol{\theta}\right)\mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}, \Sigma)\, \mathrm{d}\boldsymbol{\theta}\,.$$

Similarly, computing the mean and covariance of the real posterior can be reduced to 1 dimension. For a more detailed description of the method used here see (Seeger, 2002; Barber and Bishop, 1998). The same idea of efficiently updating the posterior distribution is extended to generalized linear models in (Lewi et al., 2007) using the Laplace approximation.

Which approximation technique performs better depends on the real posterior distribution. If the posterior distribution has a form close to a Gaussian, the simple Laplace's method gives good results. For more complex posterior distributions ADF or EP give in general better approximations (Minka, 2001). In the setting presented in this study the product between a logistic function and a Gaussian results in a posterior close to a Gaussian, thus the approximation is very accurate and the choice of the approximation method does not have a big influence on the result. In the experimental evaluation we used ADF.

3.10 Appendix C: EM Derivation

The basic idea in Bayesian hierarchical modeling is to assume that the parameters for individual models are drawn from the same hierarchical prior distribution.

We will first state the algorithm and then its derivation. We make the common assumption of a Gaussian prior distribution $p(\boldsymbol{\alpha}_m) = \mathcal{N}(\boldsymbol{\alpha}_m|\boldsymbol{\mu}, \Sigma)$ with the same $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ for all models. This prior is updated using Bayes' rule based on the observations from each scenario resulting in a posterior distribution for each individual model. Because the posterior is intractable we approximate it with a Gaussian. The hierarchical prior is obtained by maximizing the log-likelihood of all data in a so-called type-II maximum likelihood approach. This optimization is performed by applying the EM algorithm (Gelman et al., 2003; Yu et al., 2005), which reduces to the iteration (until convergence) of the following two steps.

E-step: Estimate the sufficient statistics (mean $\boldsymbol{\mu}_m$ and covariance matrix Σ_m) of the posterior distribution corresponding to each individual model m, given the current estimates ($\boldsymbol{\mu}^{(t)}$ and $\Sigma^{(t)}$) of the hierarchical prior. The E-step is performed using one of the inference techniques mentioned in Appendix 3.9.

M-step: Re-estimate the parameters of the hierarchical prior:

$$\boldsymbol{\mu}^{(t+1)} = \frac{1}{M} \sum_{m=1}^{M} \boldsymbol{\mu}_m \,, \tag{3.10}$$

$$\begin{aligned}\boldsymbol{\Sigma}^{(t+1)} = \frac{1}{\tau + M} \Bigg[& \tau \boldsymbol{\mu}^{(t+1)} \boldsymbol{\mu}^{(t+1)^T} + \sum_{m=1}^{M} \Sigma_m + \\ & \boldsymbol{I} + \sum_{m=1}^{M} (\boldsymbol{\mu}_m - \boldsymbol{\mu}^{(t+1)})(\boldsymbol{\mu}_m - \boldsymbol{\mu}^{(t+1)})^T \Bigg] \,. \end{aligned} \tag{3.11}$$

The term $\sum_{m=1}^{M} (\boldsymbol{\mu}_m - \boldsymbol{\mu}^{(t+1)})(\boldsymbol{\mu}_m - \boldsymbol{\mu}^{(t+1)})^T$ in Equation (3.11) measures the variance between the most probable estimates for different subjects; the term $\sum_{m=1}^{M} \Sigma_m$ measures the variance of the probabilities $p(\boldsymbol{\alpha}_m)$ around these most probable estimates.

In very high dimensions, some of the eigenvalues of the covariance matrix $\boldsymbol{\Sigma}$ may tend to infinity. For numerical stability, we therefore add a small constant β to the diagonal of $\mathbf{\Sigma}^{-1}$, and set

$$\mathbf{\Sigma} \leftarrow \left(\mathbf{\Sigma}^{-1} + \beta \boldsymbol{I}\right)^{-1} \tag{3.12}$$

after each update (3.11). With the update proposed in (3.12), the eigenvalues of $\mathbf{\Sigma}$ remain finite and we never observed problems with numerical stability.

It is common practice to make approximations in the E-step (see e.g., (Frey et al., 2001; Minka and Lafferty, 2002)). In theory convergence can then no longer be guaranteed, but in practice, in particular when the approximations are known to be very accurate (as it is our case, see above) it usually works fine.

In the following we give the derivation of the M-step. Let D_m denote the data obtained from subject m, $D = \{D_1, \ldots, D_M\}$ denote the data obtained from all subjects, $\mathcal{A} = \{\boldsymbol{\mu}_m, \mathbf{\Sigma}_m;\ m = 1, \ldots, M\}$ denote all parameters for all subjects, and $\Lambda^{(t)} = \{\boldsymbol{\mu}^{(t)}, \Sigma^{(t)}\}$ denote the parameters of the hierarchical prior at the tth iteration. In order to obtain the estimates of the hierarchical prior in the $(t+1)$th iteration, we maximize the penalized log likelihood of all data

$$\log[p(D|\Lambda^{(t+1)})p(\Lambda^{(t+1)})] = \log p(D|\Lambda^{(t+1)}) + \log p(\Lambda^{(t+1)})\,.$$

We note that

$$\log p(D|\Lambda^{(t+1)}) = \log\left[\frac{p(\mathcal{A}, D|\Lambda^{(t+1)})}{p(\mathcal{A}|D, \Lambda^{(t+1)})}\right] ,\forall \mathcal{A}$$

and thus

$$\begin{aligned}
&\log P(D|\Lambda^{(t+1)}) + \log p(\Lambda^{(t+1)}) \\
&\quad = \int p(\mathcal{A}|D, \Lambda^{(t)}) \log\left[\frac{p(\mathcal{A}, D|\Lambda^{(t+1)})}{p(\mathcal{A}|D, \Lambda^{(t+1)})}\right] \mathrm{d}\mathcal{A} + \log p(\Lambda^{(t+1)}) \\
&\quad = Q(\Lambda^{(t+1)}, \Lambda^{(t)}) + \log p(\Lambda^{(t+1)}) - \int p(\mathcal{A}|D, \Lambda^{(t)}) \log p(\mathcal{A}|D, \Lambda^{(t+1)})\, \mathrm{d}\mathcal{A}
\end{aligned} \tag{3.13}$$

with the "full data loglikelihood"

$$Q(\Lambda^{(t+1)}, \Lambda^{(t)}) = \int p(\mathcal{A}|\Lambda^{(t)}, D) \log p(\mathcal{A}, D|\Lambda^{(t+1)})\, \mathrm{d}\mathcal{A}\,. \tag{3.14}$$

The EM algorithm that iteratively maximizes $Q(\Lambda^{(t+1)}, \Lambda^{(t)}) + \log p(\Lambda^{(t+1)})$ is guaranteed to converge to a local maximum of the data likelihood since the negative term in Equation (3.13) can only make things better when $\Lambda^{(t+1)} \neq \Lambda^{(t)}$.

Different subjects are only coupled through their joint prior, i.e., we have

$$p(\mathcal{A}, D|\Lambda^{(t+1)}) = \prod_{m=1}^{M} p(D_m|\boldsymbol{\alpha}_m)p(\boldsymbol{\alpha}_m|\Lambda^{(t+1)}) .$$

Plugging this into Equation (3.14) we get

$$\begin{aligned} &Q(\Lambda^{(t+1)}, \Lambda^{(t)}) \\ &\quad= \int p(\mathcal{A}|D, \Lambda^{(t)}) \sum_{m=1}^{M} \log \left[p(D_m|\boldsymbol{\alpha}_m)p(\boldsymbol{\alpha}_m|\Lambda^{(t+1)})\right] \mathrm{d}\mathcal{A} \\ &\quad= \sum_{m=1}^{M} \int p(\boldsymbol{\alpha}_m|D_m, \Lambda^{(t)}) \log p(\boldsymbol{\alpha}_m|\Lambda^{(t+1)}) \, \mathrm{d}\boldsymbol{\alpha}_m + \\ &\qquad \text{constants independent of } \Lambda^{(t+1)} . \end{aligned}$$

Ignoring these constants, noting that we can skip the index of the integration variable, and dropping the superscript notation for $\Lambda^{(t+1)}$, we obtain

$$Q(\Lambda, \Lambda^{(t)}) = M \int \left[\frac{1}{M} \sum_{m=1}^{M} p(\boldsymbol{\alpha}|D_m, \Lambda^{(t)})\right] \log p(\boldsymbol{\alpha}|\Lambda) \, \mathrm{d}\boldsymbol{\alpha} .$$

The prior over Λ is a normal-inverse-Wishart distribution

$$p(\Lambda) = p(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathcal{N}(\boldsymbol{\mu}|0, \frac{1}{\pi}\boldsymbol{\Sigma})\mathcal{IW}(\boldsymbol{\Sigma}|\tau, \boldsymbol{\Sigma}_0)$$

using the inverse Wishart distribution with scale matrix $\boldsymbol{\Sigma}_0^{-1}$ defined as

$$\mathcal{IW}(\boldsymbol{\Sigma}|\tau, \boldsymbol{\Sigma}_0^{-1}) \propto \det(\boldsymbol{\Sigma}_0^{-1})^{\frac{\tau}{2}} \det(\boldsymbol{\Sigma})^{-\frac{\tau+d+1}{2}} \exp\left[-\frac{1}{2}\mathrm{Tr}(\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\Sigma}^{-1})\right]$$

At each step the following function is maximized with respect to Λ

$$\begin{aligned} &Q(\Lambda,\Lambda^{(t)}) + \log p(\Lambda) \\ &\quad= M \int \left[\frac{1}{M} \sum_{m=1}^{M} p(\boldsymbol{\alpha}|D_m, \Lambda^{(t)})\right] \log p(\boldsymbol{\alpha}|\Lambda) \, \mathrm{d}\boldsymbol{\alpha} + \log p(\Lambda) . \end{aligned} \tag{3.15}$$

The maximum is found by computing the gradients of Equation (3.15) from above with respect to $\Lambda = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ and setting these to zero. Below we compute the gradient of Equation (3.15) with respect to $\boldsymbol{\Sigma}$, but first we write down only those terms that depend on $\boldsymbol{\Sigma}$, i.e.,

$$
\begin{aligned}
\mathcal{QP}(\boldsymbol{\Sigma}) = & \int \sum_{m=1}^{M} p(\boldsymbol{\alpha}|D_m, \Lambda^{(t)}) \left[-\log\det(\boldsymbol{\Sigma})^{1/2} - \frac{1}{2}(\boldsymbol{\alpha}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\alpha}-\boldsymbol{\mu}) \right] \mathrm{d}\boldsymbol{\alpha} \\
& - \frac{\tau+d+1}{2}\log\det(\boldsymbol{\Sigma}) - \frac{1}{2}\mathrm{Tr}(\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\Sigma}^{-1}) - \frac{1}{2}\log\det(\boldsymbol{\Sigma}) - \frac{\pi}{2}\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} \\
= & \left(-\frac{1}{2}\sum_{m=1}^{M}\int p(\boldsymbol{\alpha}|D_m, \Lambda^{(t)})\,\mathrm{d}\boldsymbol{\alpha} - \frac{\tau+d+1}{2} - \frac{1}{2} \right) \log\det(\boldsymbol{\Sigma}) - \frac{1}{2}\mathrm{Tr}(\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\Sigma}^{-1}) \\
& - \frac{1}{2}\int \sum_{m=1}^{M} p(\boldsymbol{\alpha}|D_m, \Lambda^{(t)})(\boldsymbol{\alpha}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\alpha}-\boldsymbol{\mu})\,\mathrm{d}\boldsymbol{\alpha} - \frac{\pi}{2}\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} \\
= & - \frac{\tau+d+2+M}{2}\log\det(\boldsymbol{\Sigma}) - \frac{1}{2}\mathrm{Tr}(\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\Sigma}^{-1}) \\
& - \frac{1}{2}\int \sum_{m=1}^{M} p(\boldsymbol{\alpha}|D_m, \Lambda^{(t)})(\boldsymbol{\alpha}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\alpha}-\boldsymbol{\mu})\,\mathrm{d}\boldsymbol{\alpha}\frac{\pi}{2}\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} - \frac{\pi}{2}\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\,.
\end{aligned}
$$

Taking the derivatives with respect to $\boldsymbol{\Sigma}$ for each of these terms we get

$$
\begin{aligned}
\frac{\partial \log\det(\boldsymbol{\Sigma})}{\partial \boldsymbol{\Sigma}} &= \det(\boldsymbol{\Sigma})\boldsymbol{\Sigma}^{-T}\frac{1}{\det(\boldsymbol{\Sigma})} = \boldsymbol{\Sigma}^{-1}\,, \\
\frac{\partial \mathrm{Tr}(\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\Sigma}^{-1})}{\partial \boldsymbol{\Sigma}} &= -\boldsymbol{\Sigma}^{-T}\boldsymbol{\Sigma}_0^{-T}\boldsymbol{\Sigma}^{-T} = -\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\Sigma}^{-1}\,, \\
\frac{\partial \boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}}{\partial \boldsymbol{\Sigma}} &= \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}^T\boldsymbol{\mu}\boldsymbol{\Sigma}\,, \\
\frac{\partial}{\partial \boldsymbol{\Sigma}}\int \sum_{m=1}^{M} & p(\boldsymbol{\alpha}|D_m, \Lambda^{(t)})(\boldsymbol{\alpha}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\alpha}-\boldsymbol{\mu})\,\mathrm{d}\boldsymbol{\alpha} \\
&= -\int \sum_{m=1}^{M} p(\boldsymbol{\alpha}|D_m, \Lambda^{(t)})\boldsymbol{\Sigma}^{-1}(\boldsymbol{\alpha}-\boldsymbol{\mu})(\boldsymbol{\alpha}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}\,\mathrm{d}\boldsymbol{\alpha}\,.
\end{aligned}
$$

Collecting the terms from above and setting the derivative to zero we obtain

$$
\boldsymbol{\Sigma} = \frac{1}{\tau+d+2+M}\left[\pi\boldsymbol{\mu}\boldsymbol{\mu}^T + \boldsymbol{\Sigma}_0^{-1} + \int \sum_{m=1}^{M} p(\boldsymbol{\alpha}|D_m, \Lambda^{(t)})(\boldsymbol{\alpha}-\boldsymbol{\mu})(\boldsymbol{\alpha}-\boldsymbol{\mu})^T\,\mathrm{d}\boldsymbol{\alpha} \right].
$$

For each subject m, $p(\boldsymbol{\alpha}|D_m, \Lambda^{(t)})$ is the posterior distribution resulting from the hierarchical prior with the previous estimates $\Lambda^{(t)}$. This posterior is approximated to a Gaussian $\mathcal{N}(\boldsymbol{\alpha}|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$ in the previous E-step.

It follows that

$$\int \sum_{m=1}^{M} p(\boldsymbol{\alpha}|D_m, \Lambda^{(t)})(\boldsymbol{\alpha}-\boldsymbol{\mu})(\boldsymbol{\alpha}-\boldsymbol{\mu})^T \, \mathrm{d}\boldsymbol{\alpha}$$

$$= \sum_{m=1}^{M} \int \mathcal{N}(\boldsymbol{\alpha}|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) \left(\boldsymbol{\alpha}\boldsymbol{\alpha}^T - \boldsymbol{\alpha}\boldsymbol{\mu}^T - \boldsymbol{\mu}\boldsymbol{\alpha}^T + \boldsymbol{\mu}\boldsymbol{\mu}^T\right) \mathrm{d}\boldsymbol{\alpha}$$

$$= \sum_{m=1}^{M} \boldsymbol{\Sigma}_m + \sum_{m=1}^{M} \boldsymbol{\mu}_m(\boldsymbol{\mu}_m)^T - \sum_{m=1}^{M} \boldsymbol{\mu}_m\boldsymbol{\mu}^T - \sum_{m=1}^{M} \boldsymbol{\mu}(\boldsymbol{\mu}_m)^T + \sum_{m=1}^{M} \boldsymbol{\mu}\boldsymbol{\mu}^T$$

and thus

$$\boldsymbol{\Sigma} = \frac{1}{\tau + d + 2 + M}\left[\pi\boldsymbol{\mu}\boldsymbol{\mu}^T + \boldsymbol{\Sigma}_0^{-1} + \sum_{m=1}^{M} \boldsymbol{\Sigma}_m + \sum_{m=1}^{M} (\boldsymbol{\mu}_m - \boldsymbol{\mu})(\boldsymbol{\mu}_m - \boldsymbol{\mu})^T\right]$$

which is the biased estimator of the variance and where $\boldsymbol{\mu}$ is the new mean found in the M-step. To obtain an unbiased estimator we consider

$$\boldsymbol{\Sigma} = \frac{1}{\tau + M}\left[\pi\boldsymbol{\mu}\boldsymbol{\mu}^T + \boldsymbol{\Sigma}_0^{-1} + \sum_{m=1}^{M} \boldsymbol{\Sigma}_m + \sum_{m=1}^{M} (\boldsymbol{\mu}_m - \boldsymbol{\mu})(\boldsymbol{\mu}_m - \boldsymbol{\mu})^T\right]$$

which for $\boldsymbol{\Sigma}_0 = \boldsymbol{I}$ gives Equation (3.11). The update for the mean is obtained in a similar way and leads to Equation (3.10).

Note that considering the maximum-likelihood estimate, without the penalization term, i.e., maximizing $Q(\Lambda^{(t+1)}, \Lambda^{(t)})$, has the nice interpretation of the negative Kullback-Leibler divergence (up to again irrelevant constants independent of $\Lambda^{(t+1)}$) between a single Gaussian $p(\boldsymbol{\alpha}|\Lambda^{(t+1)})$ and a mixture of Gaussians, where each of the Gaussians in the mixture corresponds to the posterior of a subject given the previous setting of prior mean and variance. The maximum of this function is then found by moment matching: we have to match the moments of the single Gaussian to the moments of the mixture of Gaussians. □

Chapter 4

Efficient Preference Learning

This chapter presents a framework for optimizing the preference learning process. In many real-world applications in which preference learning is involved the available training data is scarce and obtaining labeled training data is expensive. Fortunately in many of the preference learning situations data is available from multiple subjects. We use the multi-task formalism to enhance the individual training data by making use of the preference information learned from other subjects. Furthermore, since obtaining labels is expensive, we optimally choose which data to ask a subject for labelling to obtain the most of information about her/his preferences. This paradigm — called active learning— has hardly been studied in a multi-task formalism. We propose an alternative for the standard criteria in active learning which actively chooses queries by making use of the available preference data from other subjects. The advantage of this alternative is the reduced computation costs and reduced time subjects are involved. We validate empirically our approach on three real-world data sets involving the preferences of people.[1]

[1] This chapter is based on the work: A. Birlutiu, P. Groot and T. Heskes "Efficiently Learning the Preferences of People." *(under review)*. It is based on work originally presented as: A. Birlutiu and T. Heskes " Optimal Experimental Design in a Hierarchical Setting for Probabilistic Choice Models." *Cost-Sensitive Learning workshop at Neural Information Processing Systems conference, 2008.*

4.1 Introduction

There has been an increasing interest recently in learning the preferences of people within artificial intelligence research (Doyle, 2004). Preference learning provides the means for modeling and predicting people's desires and this makes it a crucial aspect in modern applications such as decision support systems (Chajewska et al., 2000), recommender systems (Blythe, 2002; Blei et al., 2003), and personalized devices (Clyde et al., 1993; Heskes and de Vries, 2005).

A prototypical example for an application of preference learning that we will use in this paper is fitting hearing-aids, i.e., tuning of hearing-aid parameters so as to maximize user satisfaction. This is a complex task, due to three reasons: 1) high dimensionality of the parameter space, 2) the determinants of hearing-impaired user satisfaction are unknown, and 3) the evaluation of this satisfaction through listening tests is costly (in terms of patient burden and clinical time investment) and unreliable (due to inconsistent responses). The last point illustrates an important issue that preference learning has to address, which is the limited availability of labeled data used for model training. Obtaining appropriate training data in preference learning applications requires time and effort from the modeled user. This shortcoming can be addressed by taking advantage of two characteristics of the settings in which preference learning is usually applied. First, the training data is mostly acquired through interactions with the modeled user; and, second, preferences are modeled for multiple users, as a result multiple training data sets are available. In order for the preference learning methods to be implemented in real-world systems, they must be capable of exploiting all possible sources of information and in the most efficient way.

In most of the situations in which preference learning is involved data is available from multiple subjects. Thus, even though individual data is scarce and difficult to obtain, we can optimize the learning of preferences of a new subject by making use of the available data from other subjects. Learning in this setting is well-known as multi-task or hierarchical learning and has been studied extensively in recent years in machine learning. By using the multi-task formalism, the preference data collected for other subjects can be gathered and used as prior information when learning the preferences of a new subject. Furthermore, to deal with the fact that obtaining labeled data is expensive we can speed up learning by optimally choosing the examples to be queried. At each learning step we can decide which example gives the most information about the subject's preferences. This paradigm, called active learning in the machine learning literature and related to sequential experimental design in statistics, has been studied extensively, but hardly in the multi-task setting.

The aim of this work is to present an efficient framework for optimizing the preference learning process. This framework considers the combination between active learning and multi-task learning in the preference learning context. The contribution of this work is a criterion for active learning designed for the multi-task setting. The advantages of this criterion are in its interpretation and the ease in computability.

The structure of this paper is as follows. We finish this section by presenting related works on preference learning, multi-task learning and active learning.

In Section 4.2 we describe the learning framework. We consider learning from qualitative preference observations in which the subject makes a choice for one of the presented alternatives. This can be modeled using the probabilistic choice models introduced in Section 4.2.1. Learning a utility function representing the preferences of a subject from this type of preference observations is described in Section 4.2.2. Learning the utility function in a multi-task setting by making use of the data available from other subjects is considered in Section 4.2.3.

In Section 4.3 we present several criteria for selecting the most informative experiments with respect to a subject's preferences. After reviewing some of the standard criteria from experimental design, we propose an alternative criterion which makes use of the preference observations collected already from a community of subjects. We show that this alternative criterion is connected to the standard criteria from experimental design.

In Section 4.4 we demonstrate experimentally the usefulness of our framework on three data sets, a subset of the Letor data set, an audiological data set and a data set about people's preferences for art images.

In Section 4.5 we conclude and discuss several directions for future research.

4.1.1 Background and Related Work

In this section we review some studies from preference learning, multi-task learning, and active learning related to the work presented in this paper.

Preference Learning

Preference learning is the creation of a model from collected data which can be used to model and predict people's desires. A very recent and complete overview about preference learning is given in (Fürnkranz and Hüllermeier, 2010). There are different approaches to preference learning which can be categorized according to the learning task, the learning technique and the application area. We will briefly enumerate them and state in which category our current work can be included (for more details we refer the interested reader to (Fürnkranz and Hüllermeier, 2010)).

1. Based on the application area, preference learning approaches can be divided into the following main groups: *i)* applied to the field of information retrieval, e.g., learning to rank search results of a query or a search engine, *ii)* applied to recommender systems, e.g., used by online stores to recommend products to their customers, or for personalized devices, and *iii)* bipartite ranking and label ranking, which find applications in disciplines such as medicine and biology. The application scenarios that we consider in the experimental evaluation in Section 4.4

belong to information retrieval (the Letor data set) and recommender systems (the audiological and art data sets).

2. The learning technique divides the preference learning approaches into four categories: *i)* learn a binary preference relation that compares pairs of alternatives, *ii)* model-based approach that aims at identifying the preference relation by making sufficiently restrictive model assumptions, *iii)* local estimation techniques which lead to aggregating preferences, and *iv)* learning utility functions by using regression to map instances to target valuations for direct ranking. We focus on the latter approach and use a utility function in order to model the subject's preferences. The utility function is learned in a Bayesian framework.

3. The learning task includes label, instance, and object ranking. Label ranking can be seen as a generalization of classification where a complete ranking of labels is associated with an instance instead of only a class label. Instance ranking can be seen as a generalization of ordinal classification where an instance belongs to one among a finite set of classes and the classes have an order. The setting of object ranking has the particularity of having no supervision in the sense that no class label is associated with an object. Instead, a finite set of pairwise preferences or other ordering between objects is given. The setting that we consider in this work belongs to the last category.

In many preference learning settings it is important to take into account the context, i.e., context-aware preference learning (Adomavicius et al., 2005). The motivation for context aware preference learning is that the same subject/user/consumer may use different decision-making strategies and prefer different products under different contexts. In one of the application scenarios that we use in this work, hearing-aid fitting, this means that a user would prefer a certain setting of the hearing-aid parameters if he is listening to a concert and another setting if he is in a discussion. In general, for context aware preferences, bigger data set are needed, as for all contextual situations preferences would have to be learned. The approach that we present in this paper can be applicable in this setting as well. While this is an interesting, related topic, it is beyond the scope of the current work.

Multi-Task Learning

The idea behind multi-task learning is to utilize labeled data from other "similar" learning tasks in order to improve the performance on a target task. It is inspired by the research on transfer of learning in psychology, more specifically on the dependency of human learning on prior experience. For example, the abilities acquired while learning to walk presumably apply when one learns to run, and knowledge gained while learning to recognize cars could apply when recognizing trucks. The initial foundations for multi-task learning were laid by (Thrun, 1995; Caruana, 1997). The psychological theory of transfer of learning implies the similarity between tasks. In a related way, the multi-task learning assumes similarity between models of different tasks. For example, (Evgeniou et al., 2005; Argyriou et al., 2009) exploit similarity between the deterministic parts of the models by means of regularization, with the effect of improvement in performance. In this work we implement multi-task learning using a Bayesian approach. The Bayesian approach to multi-task learning assumes the parameters of individual models to be drawn from the same prior distribution. Examples of the Bayesian approach to multi-task learning are (Bakker and Heskes, 2003) where a mixture of Gaussians is used for the top of the hierarchy. This leads to clustering the tasks, one cluster for each Gaussian in the mixture. In (Yu et al., 2005; Birlutiu et al., 2010) a hierarchical Gaussian Process is derived with a normal-inverse Wishart distribution used at the top of the hierarchy.

Active Learning

Active learning, also known in the statistics literature as sequential experimental design, is suitable for situations in which labeling points is difficult, time-consuming, and expensive. The idea behind active learning is that by optimal selection of the training points a better performance can be achieved then by random selection. The scenarios in which active learning can be applied belong to one of the following three categories: *i)* generating *de novo* points for labeling; *ii)* stream-based active learning where the learner decides whether to request the label of a given instance or not; *iii)* pool-based active learning where queries are selected from a large pool of unlabeled data. In this work we consider the pool-based active learning setting.

Methods for active learning can be roughly divided into two categories: those with and without an explicitly defined objective function. Uncertainty sampling (Lewis and Gale, 1994), Query-by-Committee (Seung et al., 1992; Freund et al., 1997) and variants thereof belong to the latter category. They are based on the idea of selecting the most uncertain data given the previously trained models. The methods with an explicit objective function are often motivated by the theory of experimental design (Fedorov, 1972; Chaloner and Verdinelli, 1995; Schein and Ungar, 2007; Lewi et al., 2009; Dror and Steinberg, 2008). The objective function then quantifies the expected gain of labeling a particular input, for example in terms of the expected reduction in the entropy of the model parameters (MacKay, 1992; Cohn et al., 1996). With respect to the performance of the two categories of methods, Schein and Ungar (2007) show that the methods from the second approach perform better but are computationally more expensive due to retraining the models for each candidate point. A trend is to improve the performance of the active learning methods by combining them with heuristics designed either for the context in which they are applied or by the models they use, e.g., making use of the unlabeled data available (McCallum and Nigam, 1998; Yu et al., 2006), exploiting the clusters in the data (Dasgupta and Hsu, 2008), diversifying the set of hypotheses (Melville and Mooney, 2004), or adapting the active learning to Gaussian processes (Chu and Ghahramani, 2005b; Brochu et al., 2008).

Preference learning can benefit from the active learning paradigm. In most of the preference learning settings labels are given by people in an explicit way. This means that for acquiring training preference data, the subjects have to interact with the system, and they need to express their preferences explicitly. These situations appear when it is impossible or insufficient to implicitly collect training preference data. For example, when learning preferences in a live system where subjects choose electronically their favorite movie, labelling is done automatically by the selection, but, for other scenarios, like for example, fitting hearing-aids, this implicit way of collecting training data cannot be applied. In these situations, it makes sense to use active learning in order collect the most informative data. There are severals studies in the literature that use active learning in a preference learning setting. Brinker (2004) presents some extensions of pool-based active learning to label ranking problems; Xu et al. (2010) address the problem of preference learning using relational models between items; Guo and Sanner (2010) investigate active preference learning for real-time systems; Brochu et al. (2008) propose a criterion for active learning that maximizes the expected improvement at each query without accurately modelling the entire valuation surface. Furthermore, there are several studies which investigate active preference learning for practical applications such as, collaborative filtering (Jin and Si, 2004; Harpale and Yang, 2008; Boutilier et al., 2003), personalized calendar scheduling (Gervasio et al., 2005), or for optimizing search results for biomedical documents (Arens, 2008). The difference between our work and the other studies for active preference learning mentioned above is that we consider multiple learning tasks, i.e., active preference learning in a multi-task setting. We further propose an alternative to the standard active learning criteria which makes use of the preference observations collected already from a community of subjects. This criterion, which we call the Committee criterion, is thus particularly designed for the multi-task setting that we consider in this work. The idea behind the Committee criterion is related to the Query-by-Committee method from active learning which selects those queries that have maximum disagreement amongst an ensemble of hypotheses. The difference in our case is that the group of subjects, for which the preferences were already learned, plays the role of the ensemble of hypotheses instead of an ensemble of models learned on the same task.

4.1.2 Notation

Boldface notation is used for vectors and matrices and normal fonts for their components. Upperscripts are used to distinguish between different vectors or matrices and lowerscripts to address their components. The notation $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is used for a multivariate Gaussian with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The transpose of a matrix M is denoted by M^T. Capital letters are used for constants and small letters for indices, e.g., $i = 1, \ldots, I$.

4.2 Learning Framework

The idea of using the preference observations from other subjects in order to optimize the process of learning the preferences of a new subject can be basically applied in any preference learning context. In this work, we consider the case of qualitative preference observations which can be modeled using the probabilistic choice models described in this section.

4.2.1 Probabilistic Choice Models

In many real-world applications preferences are learned from experiments in which the subject makes a choice for one of the presented alternatives. The motivation for this is that people are very good at making comparisons between alternatives and expressing a preference for one of them, i.e., qualitative preference observations. This is in contrast to quantitative preference observations were the people have to assign an absolute rating to each alternative independently. Let $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_I\}$ be a set of inputs. Let D be a set of J observed preference comparisons over instances in X corresponding to a subject,

$$D = \{(a_j, c_j) | 1 \leq j \leq J, c_j \in \{1, \ldots, A\}\} \tag{4.1}$$

with $a_j = (\boldsymbol{x}_{i_1(j)}, \ldots, \boldsymbol{x}_{i_A(j)})$ the alternatives presented and c_j the choice made, $i_1, \ldots, i_A : \{1, \ldots, J\} \rightarrow \{1, \ldots, I\}$ index functions such that $i_1(j)$ represents the input presented first in the jth preference comparison and $c_j = c$ means that $\boldsymbol{x}_{i_c(j)}$ is chosen from the A alternatives presented in the jth comparison. For $A = 2$ this setup reduces to pairwise comparisons between two alternatives.

The main idea behind probabilistic choice models is to assume a latent utility function value $U(\boldsymbol{x}_i)$ associated with each input $\boldsymbol{x}_i$ which captures the individual preference of a subject for $\boldsymbol{x}_i$ (the utility function will be formally defined in the next section). In the ideal case the latent function values are consistent with the preference observations. This means that alternative c is preferred over the other alternatives c' in the jth comparison whenever the utility for c exceeds the utilities for the other alternatives c', i.e., $U(\boldsymbol{x}_{i_c(j)}) > U(\boldsymbol{x}_{i_{c'}(j)})$. In practice, however, subjects are often inconsistent in their responses. A very inconsistent subject will have a high uncertainty associated with the utility function; this uncertainty is directly taken into account in the probabilistic framework. We define this probabilistic framework using the Bradley-Terry model (Bradley and Terry, 1952; Kanninen, 2002; Glickman and Jensen, 2005) by making a standard modeling assumption that the probability that the cth alternative is chosen by the subject in the jth comparison follows a multinomial logistic model, which is defined as

$$p(c_j = c | a_j, U) = \frac{\exp\left[U(\boldsymbol{x}_{i_c(j)})\right]}{\sum_{c'=1}^{A} \exp\left[U(\boldsymbol{x}_{i_{c'}(j)})\right]}\,, \tag{4.2}$$

where "exp" is the exponential function and the other terms are as defined before. Efficiently learning preferences reduces to learning the unknown utility function U as accurately and with as few comparisons as possible.

One important drawback of the Bradley-Terry model is that it assumes very strong transitivity conditions of preference relations, while some psychological experiments have shown that human preference judgments can violate transitivity (Anand, 1993; Tversky, 1998). In most situations transitivity violations can be considered as noise. When this is not applicable, specific probabilistic models for human preference judgements which preserve intransitive reciprocal relation have to be designed. This was recently investigated in (Pahikkala et al., 2009) which introduced a new kernel function in the framework of regularized least squares which is capable of inferring intransitive reciprocal relations.

4.2.2 The Utility Function

The utility function U is a real-valued function, $U : X \to \mathbb{R}$, which associates with every input $\boldsymbol{x} \in X$ a real number $U(\boldsymbol{x})$. Each input $\boldsymbol{x} \in X$ is characterized by a set of features, $\boldsymbol{\phi}(\boldsymbol{x}) \in \mathbb{R}^D$. One possible choice for the utility function is to express it as a linear combination of the features,

$$U(\boldsymbol{x}) = \sum_{i=1}^{D} \alpha_i \phi_i(\boldsymbol{x}) \,, \tag{4.3}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_D)$ is a vector of weights which captures the importance of each feature of $\boldsymbol{x}$ when evaluating the utility U for a specific subject, $\phi_i(\boldsymbol{x})$ are the components of the vector $\boldsymbol{\phi}(\boldsymbol{x})$. The preferences of a subject are thus encoded in the vector $\boldsymbol{\alpha}$ and learning the utility function reduces to learning $\boldsymbol{\alpha}$.

In order to make the definition of the utility function more flexible, we can use a semiparametric model in which the utility function is defined as a linear combination of basis functions. The basis functions are defined by a kernel function κ centered on the data points,

$$U(\boldsymbol{x}) = \sum_{i=1}^{I} \alpha_i \kappa(\boldsymbol{x}, \boldsymbol{x}_i) \,, \tag{4.4}$$

where the vector $\boldsymbol{\alpha}$ with dimension I—the number of data points (the size of the set of inputs X)—captures the preferences of the subject. A non-linear utility function can be obtained by using, for example, a Gaussian kernel,

$$\kappa_{\text{Gauss}}(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\ell}{2} \sum_{j=1}^{D} (\phi_j(\boldsymbol{x}) - \phi_j(\boldsymbol{x}'))^2 \right) \,, \tag{4.5}$$

where ℓ is a length-scale parameter. The two definitions of the utility function from

Equations (4.3) and (4.4) are similar in the sense that they are both linear in the parameter. Equation (4.4) is suited when the number of features is larger than the number of data points, i.e., $D > I$ and for introducing non-linearity in the utility model.

In order to learn the utility function, we use a Bayesian framework in which we treat the vector of parameters $\boldsymbol{\alpha}$ as a random variable. We consider a Gaussian prior distribution over $\boldsymbol{\alpha}$, $p(\boldsymbol{\alpha}) = \mathcal{N}(\boldsymbol{\mu}, \Sigma)$, which is updated based on the observations from the preference comparisons using Bayes' rule,

$$p(\boldsymbol{\alpha}|D, \boldsymbol{\mu}, \Sigma) \propto p(\boldsymbol{\alpha}) \prod_{j=1}^{J} p(c_j|a_j, \boldsymbol{\alpha}) , \tag{4.6}$$

with the likelihood terms of the form given in Equation (4.2). The choice of the prior will be discussed in the next section. The posterior distribution obtained is approximated to a Gaussian. The Gaussian approximation of the posterior is a good approximation because with few data points the posterior is close to the prior which is a Gaussian, and with many data points the posterior approaches again a Gaussian as a consequence of the central limit theorem (Bishop, 2006). To perform the approximation of the posterior a good choice is to use deterministic methods (e.g., Laplace's method (Mackay, 2002), Expectation Propagation (Minka, 2001)) since they are computationally cheaper than the non-deterministic ones (sampling) and because they are known to be accurate for these types of models (Glickman and Jensen, 2005).

4.2.3 Multi-task Preference Learning

One property which distinguishes preference learning from other learning settings is that in most of the cases in which preference learning is involved, observations are available from multiple subjects. For example, in the case of product recommendation, preferences are learned from multiple consumers. Furthermore, in situations where the training preference data is collected in an explicit way, by interactions with the subject, the individual training data is usually small. As a consequence, it is reasonable to make use of all the data available. In this direction, we make the assumption that each subject is not an isolated case, but belongs to a group of people sharing a common underlying rationale in their way of making preference decisions. This property combined with the Bayesian framework allows the transfer of information about preferences between different subjects. Basically, we use the preference data previously seen from some other subjects to learn an informed prior which will be used as the starting prior when learning the preferences of a new subject. For learning this informed prior, we use Bayesian hierarchical modeling which assumes that the parameters for individual models are drawn from the same hierarchical prior distribution. Let us assume that we already have preference data available from a group of M subjects. We make the common assumption of a Gaussian prior distribution, $p(\boldsymbol{\alpha}^m) = \mathcal{N}(\bar{\boldsymbol{\mu}}, \bar{\Sigma})$, $m = 1, \ldots, M$ with the same $\bar{\boldsymbol{\mu}}$ and $\bar{\Sigma}$ for the preference models of all subjects. This prior is updated using Bayes' rule based on the observations from each subject, resulting in a posterior distribution for each individual subject. The common prior over all task parameters controls the general part of the model. This common prior is learned from the data belonging to a group of tasks, other than the current (new) task for which the learning is performed. Starting from this general-model given by the common prior, the model is updated using the observation (data) seen in the current task. These task-specific observations control the task-specific part of the model. The hierarchical prior is obtained by maximizing the penalized log-likelihood of all data in a so-called type-II maximum likelihood approach. This optimization is performed by applying the EM algorithm (Gelman et al., 2003), which reduces to the iteration (until convergence) of the following steps:

E-step: Estimate the sufficient statistics (mean $\boldsymbol{\mu}^m$ and covariance matrix $\boldsymbol{\Sigma}^m$) of the posterior distribution corresponding to each subject m, given the current estimates at step t ($\bar{\boldsymbol{\mu}}^{(t)}$ and $\bar{\boldsymbol{\Sigma}}^{(t)}$) of the hierarchical prior.

M-step: Re-estimate the parameters of the hierarchical prior:

$$\bar{\boldsymbol{\mu}}^{(t+1)} = \frac{1}{M}\sum_{m=1}^{M} \boldsymbol{\mu}^m \,, \tag{4.7}$$

$$\bar{\boldsymbol{\Sigma}}^{(t+1)} = \frac{1}{M}\sum_{m=1}^{M} (\boldsymbol{\mu}^m - \bar{\boldsymbol{\mu}}^{(t+1)})(\boldsymbol{\mu}^m - \bar{\boldsymbol{\mu}}^{(t+1)})^T + \frac{1}{M}\sum_{m=1}^{M} \boldsymbol{\Sigma}^m \,. \tag{4.8}$$

The details of the derivations of Equations (4.7) and (4.8) can be found in Appendix C of our paper (Birlutiu et al., 2010). The hierarchical prior is set to $p(\boldsymbol{\alpha}) = \mathcal{N}(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\Sigma}})$ where $\bar{\boldsymbol{\mu}} = \bar{\boldsymbol{\mu}}^T$ and $\bar{\boldsymbol{\Sigma}} = \bar{\boldsymbol{\Sigma}}^T$, at step T when iterations were stopped (at convergence). Once we have learned the hierarchical prior we can use it as an informative prior for the preference model of a new subject in Equation (4.6).

4.3 Active Preference Learning

In this section we discuss methods for active preference learning. We start from Query-by-Committee (QBC) (Seung et al., 1992) method for active learning and based on it we propose some variants of QBC adapted to the setting of preference learning for multiple subjects (Section 4.3.1). Furthermore, we show how these variants of QBC can be naturally linked to the hierarchical Bayesian modeling for reducing the computations (Section 4.3.2). Finally, we show connections between the variants of QBC proposed and other active learning criteria (Section 4.3.4).

4.3.1 QBC for Preference Learning

In this section we will discuss how to adapt QBC to our preference learning setting.

The Committee Members

For the QBC approach to be effective it is important that the committee is made of consistent and representative models. The main idea in this work is to exploit the preference learning setting with multiple subjects and use the learned models of other subjects $\mathcal{M}_1, \ldots, \mathcal{M}_M$ as committee members when learning the preferences of a new subject.

After choosing the committee we still have to decide upon a suitable criterion for selecting the next examples. Some measures of disagreement among the committee members appear to be most obvious, and in the following we will consider two alternatives.

Vote Criterion

A simple and straightforward way is to consider the labels assigned by the other subjects, e.g., through the Vote criterion defined as

$$\text{Vote}(a) = \max_c \sum_{m=1}^{M} \delta(a, c; m) , \tag{4.9}$$

where $\delta(a, c; m) = 1$ if $(a, c) \in D_m$, and $\delta(a, c; m) = 0$ otherwise. The score Vote(a) is minimal when the labels assigned by the committee members are equally distributed (total disagreement) and maximal when all members fully agree. There are two problems with this criterion. First, a comparison a may not be labeled by a subject m. This can be overcome if we consider the predictions computed based on the learned model of subject m and allow each committee member to 'vote' for its winning class. This same idea is implemented in the so-called vote entropy method (Dagan and Engelson, 1995). The entropy is measured over the final classes assigned to an example by possible models, and not over class probabilities given by possible models. Second, in practical applications just scoring votes turns out to be suboptimal. The reason, as also suggested in (McCallum and Nigam, 1998), is that the Vote criterion does not take into account the confidences of the committee members' predictions.

Committee Criterion

We will use the following notation for the predictive probability corresponding to a subject $m = 1, \ldots, M$,

$$p_m(c|a) \equiv p(c|a, \mathcal{M}_m) .$$

The predictive probability can be computed either by taking into account the entire distribution $\mathcal{M}_m = \mathcal{N}(\boldsymbol{\mu}^m, \boldsymbol{\Sigma}^m)$

$$p_m(c|a) = \int p(c|a, \boldsymbol{\alpha}) \mathcal{N}(\boldsymbol{\alpha}|\boldsymbol{\mu}^m, \boldsymbol{\Sigma}^m) d\boldsymbol{\alpha} ,$$

or, for computational reasons, we can consider only a point estimate for $\mathcal{M}_m$, for example, the mean of the Gaussian distribution, and use it to compute the predictive probabilities using Equation (4.2)

$$p_m(c|a) \approx p(c|a, \boldsymbol{\mu}^m) \,. \tag{4.10}$$

Inspired by (McCallum and Nigam, 1998), we propose to measure disagreement by taking the average prediction of the entire committee and computing the average Kullback-Leibler (KL) divergence of the individual predictions from the average:

$$\text{Committee}(a) = \sum_{m=1}^{M} \frac{1}{M} \text{KL}[\bar{p}(\cdot|a)||p_m(\cdot|a)] \,, \tag{4.11}$$

with $\bar{p}(\cdot|a)$ the average predictive probability of the entire committee, which will be more precisely defined in Section 4.3.2.

The KL divergence for discrete probabilities is defined as

$$\text{KL}[p_1(\cdot|a)||p_2(\cdot|a)] = \sum_c p_1(c|a) \log\left(\frac{p_1(c|a)}{p_2(c|a)}\right) \,.$$

The KL divergence can be seen as a distance between probabilities, where we abused the notion of distance, since the KL-divergence is not symmetric, i.e., $\text{KL}[p_1||p_2] \neq \text{KL}[p_2||p_1]$. This drawback of the KL-divergence can be overcome by considering a symmetric measure, for example, $\text{KL}[p_1||p_2] + \text{KL}[p_2||p_1]$. In (McCallum and Nigam, 1998), the disagreement is computed between committee members constructed based on the current model, i.e., the committee changes with every update and the criterion has to be recomputed with every update. A committee of models learned on different tasks is fixed and thus selecting examples solely based on it leads to a fixed instead of an active design: all examples can be ranked beforehand (the same applies to the Vote criterion defined above).

To arrive at an active design and take into account the current model, we propose a small modification based on the following intuition. Querying examples on which the committee members disagree makes sense, because it will force the current model to make a choice between options that, according to the committee members, are reasonably plausible. However, when the current model on a particular example already “made up its mind”, i.e., deviates substantially from the average prediction of the committee based on what it learned from other input/output pairs, it makes no sense to still query that example, even though the committee members might disagree. Taking into account this consideration, we propose the Committee criterion which assigns a score to a candidate query comparison a through

$$\text{Committee}(a) = \frac{1}{M}\sum_{m=1}^{M} \text{KL}[\bar{p}(\cdot|a)||p_m(\cdot|a)] - \gamma\text{KL}[\bar{p}(\cdot|a)||p(\cdot|a)] \, , \tag{4.12}$$

with $p(\cdot|a)$ the current model's predictive probability based on the data seen so far and γ a parameter that accounts for the degree of similarity between subjects. According to the Committee criterion, the most interesting experiments are those on which the other models disagree (the first term on the righthand side of Equation (4.12)), with the current model (still) undecided (the second term on the righthand side of Equation (4.12)).

An advantage of the Committee criterion is its computational efficiency: the first term on the righthand side of Equation (4.12) as well as the average predictive probability can be computed beforehand. The Committee criterion does require computation of the predictive probabilities corresponding to the current model, but this is the least one could expect from an active design. This is to be compared with the QBC criterion (any of the two variants considered), which requires constructing new committee members with each update, and D-optimal experimental design, which calls for keeping track of variances.

Note that we have not made any restriction so far with respect to the probabilistic models used in the active learning design. In the following we will consider only the log-linear models introduced in Section 4.2. They have some nice properties, which simplify the computation of the Committee criterion (Section 4.3.2), and provide a natural link to hierarchical Bayesian modeling (Section 4.2.3). The general idea, of using the already learned models from the other tasks as the committee members in a QBC-like approach, is of course also applicable to other models.

4.3.2 Average Probability

In this section we discuss how to efficiently compute the average probability used for computing the committee criterion in Equation (4.12) in the case of log-linear models (Christensen, 1997). For linear utility functions the likelihood function defined in Equation (4.2) is a log-linear model. The log-odds of the model are linear in the parameter.

Let $p_m(c|a)$ be the predictive probability defined in Equation (4.10). We define the average predictive probability of the committee, $\bar{p}(c|a)$, as the prediction probability that is closest to the prediction probabilities of the members:

$$\bar{p}(c|a) \equiv \operatorname*{argmin}_{p(c|a)} \sum_{m=1}^{M} \frac{1}{M}\text{KL}[p(c|a)||p_m(c|a)] \, . \tag{4.13}$$

The solution of the optimization from above is the so-called logarithmic opinion pool (Bordley, 1982)

$$\bar{p}(c|a) = \frac{1}{Z(a)} \prod_{m=1}^{M} [p_m(c|a)]^{\frac{1}{M}} = \frac{1}{Z(a)} \exp\left(\frac{1}{M} \sum_{m=1}^{M} \log p_m(c|a) \right) , \tag{4.14}$$

with $Z(a)$ a normalization constant

$$Z(a) = \sum_{c} \prod_{m=1}^{M} [p_m(c|a)]^{\frac{1}{M}} .$$

For log-linear models, the logarithmic opinion pool boils down to a simple averaging of model parameters:

$$\bar{p}(c|a) = p(c|a, \bar{\boldsymbol{\mu}}) \text{ with } \bar{\boldsymbol{\mu}} = \frac{1}{M} \sum_{m=1}^{M} \boldsymbol{\mu}^m . \tag{4.15}$$

This natural combination between log-linear models and logarithmic opinion pools is the advantage of using the logarithmic opinion pool instead of the linear opinion pool used in (McCallum and Nigam, 1998).

As can be seen from the EM updates in Equation (4.7), the average $\bar{\boldsymbol{\mu}}$ in the logarithmic opinion pool is then precisely the mean of the learned hierarchical prior. Summarizing, once we have learned a hierarchical prior from the data available for subjects 1 through M using the EM algorithm, we can start off the new model $M + 1$ from this prior (as is normally done in hierarchical Bayesian learning). On top of this, the same EM algorithm gives us the information we need to compute the Committee criterion that can be used subsequently to select new inputs to label.

4.3.3 Standard Criteria for Active Learning

In this section we discuss how several strategies for active learning can be implemented in the learning framework considered here. All the strategies are being concerned with evaluating the informativeness of the unlabeled points. Let the new model obtained after incorporating an observation (a, c) be $\mathcal{M}_{(a,c)} \approx \mathcal{N}(\boldsymbol{\mu}_{(a,c)}, \Sigma_{(a,c)})$.

1. **Uncertainty Sampling** (Lewis and Gale, 1994). In this strategy an active learner chooses for labeling the example for which the model's predictions are most uncertain. The uncertainty of the predictions can be measured, for example, using Shannon entropy

$$\text{Uncertainty}(a) = -\sum_{c} p(c|a, \mathcal{M}) \log p(c|a, \mathcal{M}) . \tag{4.16}$$

For a binary classifier this strategy reduces to querying points whose prediction probabilities are close to 0.5. Intuitively this strategy aims at finding as fast as possible the decision boundary since this is indicated by the regions where the model is most uncertain.

2. **Variance Reduction** (MacKay, 1992). This strategy, also known in experimental design as D-optimality (Fedorov, 1972; Chaloner and Verdinelli, 1995; Berger, 1994; Ford and Silvey, 1980), chooses as the most informative experiments the ones that give the most reduction in the model's uncertainty. The motivation behind this strategy is a result of (Geman et al., 1992) which shows that the generalization error can be decomposed into three components: *i)* noise (which is independent of the model or training data); *ii)* bias (due to the model); *iii)* model's uncertainty. Since the model cannot influence the noise and the bias components, the future generalization error can only be influenced via the model's variance. Formally, this criterion can be written as

$$\text{Variance}(a) = \sum_c p(c|a, \mathcal{M})\text{variance}[\mathcal{M}_{(a,c)}] - \text{variance}[\mathcal{M}] \,. \qquad (4.17)$$

In the setting considered in this work the variance of the model is expressed in the covariance of the Gaussian distribution. In order to use Equation (4.17) we need to choose a measure for the variance. We can consider, for example, the log-determinant of the covariance matrix

$$\text{Variance-logdet}(a) = \sum_c p(c|a, \mathcal{M}) \log\det(\mathbf{\Sigma}_{(a,c)}) - \log\det(\mathbf{\Sigma}) \,, \qquad (4.18)$$

which is actually minimizing the entropy of the Gaussian random variable representing the current model, or the trace of the covariance matrix

$$\text{Variance-trace}(a) = \sum_c p(c|a, \mathcal{M}) \operatorname{Tr}(\mathbf{\Sigma}_{(a,c)}) - \operatorname{Tr}(\mathbf{\Sigma}) \,. \qquad (4.19)$$

3. **Expected Model Change** (Cohn et al., 1996). This strategy chooses as the most informative query the one which when added to the training set would yield the greatest model change. Quantifying the model change depends on the learning framework. For gradient-based optimization the change can be measured via the training gradient, i.e., the vector used to re-estimate parameter values (Settles and Craven, 2008). In the Bayesian framework, the model change can be quantified via a distance measure between the current distribution and the posterior distribution obtained after incorporating the candidate point

$$\text{Change}(a) = \sum_c p(c|a, \mathcal{M})\text{distance}\left[\mathcal{M}, \mathcal{M}_{(a,c)}\right] \,.$$

A suitable distance for our setting is the Kullback-Leibler divergence between distributions, which for two Gaussians has a closed form solution and can be written as follows

$$\begin{aligned}\text{Change-KL}(a) =& \sum_c p(c|a,\mathcal{M}) KL\left[\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})||\mathcal{N}(\boldsymbol{\mu}_{(a,c)},\boldsymbol{\Sigma}_{(a,c)})\right] \\ =& \sum_c p(c|a,\boldsymbol{\mu},\boldsymbol{\Sigma}) \Bigg[\log\left(\frac{\det \boldsymbol{\Sigma}_{(a,c)}}{\det \boldsymbol{\Sigma}}\right) + \text{Tr}\left(\boldsymbol{\Sigma}^{-1}_{(a,c)}\boldsymbol{\Sigma}\right) + \\ & + \left(\boldsymbol{\mu}_{(a,c)} - \boldsymbol{\mu}\right)^T \boldsymbol{\Sigma}^{-1}_{(a,c)}(\boldsymbol{\mu}_{(a,c)} - \boldsymbol{\mu}) - n \Bigg] . \qquad (4.20)\end{aligned}$$

The KL divergence between Gaussians is used by Seeger (2008) to design an efficient sequential experimental design in a setting similar to the one used in this work.

Uncertainty sampling, and QBC and its variants are attractive due to their applicability in various machine learning settings. Variance reduction and expected model change are robust and in many situations they have proved to be the best one can do (Schein and Ungar, 2007). Although more robust, the variance reduction and expected model change strategies are computationally more demanding since for each candidate comparison query and each possible label the posterior distribution induced has to be computed. The posterior distribution cannot be computed analytically and approximations are needed; these approximations are usually costly. The variants of QBC proposed in this paper can address this drawback related to computational efficiency.

4.3.4 Similarities between Criteria

In this section we consider the following active learning criteria: Variance-logdet, Committee, Variance-trace and Change-KL. We investigate how similar the active learning criteria are and how they can be related. We analyze the modifications induced to the model by the criteria after updating the probability model to incorporate the information from new training points. A single update induces a small change in the posterior distribution, and this allows for Taylor expansions, keeping only the lowest non-zero contribution. In the following we present the main results of the approximations while some of the details can be found in the appendix.

Assuming that the updates of the posterior distribution for each alternative a and choice c lead to small changes in the model $\mathcal{M}$, we can approximate the active criteria to the form

$$\sum_c p(c|a,\boldsymbol{\alpha})\boldsymbol{g}(c|a,\boldsymbol{\alpha})^T \boldsymbol{Q}\boldsymbol{g}(c|a,\boldsymbol{\alpha}) , \qquad (4.21)$$

for some vector $\boldsymbol{\alpha}$ and matrix $\boldsymbol{Q}$ and with $\boldsymbol{g}$ the gradient of the log-probabilities

$$\boldsymbol{g}(c|a,\boldsymbol{\alpha}) \equiv \frac{\partial \log p(c|a,\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}} .$$

The following lemma approximates the Variance-logdet criterion to the form from Equation (4.21).

Lemma 4.3.1 *In a first order approximation, assuming that $\boldsymbol{\Sigma}_{(a,c)}$ is close to $\boldsymbol{\Sigma}$, we can simplify*

$$\textit{Variance-logdet}(a) \approx \sum_c p(c|a,\boldsymbol{\mu})\boldsymbol{g}(c|a,\boldsymbol{\mu})^T\,\boldsymbol{\Sigma}\,\boldsymbol{g}(c|a,\boldsymbol{\mu})\,. \tag{4.22}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ represent the mean and covariance of the Gaussian posterior distribution.

Proof.

In a first order approximation we have

$$\boldsymbol{\Sigma}_{(a,c)}^{-1} \approx \boldsymbol{\Sigma}^{-1} - \left.\frac{\partial^2 \log p(c|a,\boldsymbol{\alpha})}{\partial\boldsymbol{\alpha}\partial\boldsymbol{\alpha}^T}\right|_{\boldsymbol{\alpha}=\boldsymbol{\mu}} \tag{4.23}$$

where we ignored the change from the old $\boldsymbol{\alpha}$ to a new MAP solution depending on c and a. We will use the notation

$$\boldsymbol{H}(c|a,\boldsymbol{\alpha}) \equiv \frac{\partial^2 \log p(c|a,\boldsymbol{\alpha})}{\partial\boldsymbol{\alpha}\partial\boldsymbol{\alpha}^T}\,.$$

For a matrix $\boldsymbol{A}$ and ϵ small compared to $\boldsymbol{A}$, the following holds (see, for example, Boyd and Vandenberghe (2004), page 642)

$$\log\det(\boldsymbol{A}+\epsilon\boldsymbol{I}) \approx \log\det(\boldsymbol{A}) + \mathrm{Tr}[\boldsymbol{A}^{-1}\epsilon]\,. \tag{4.24}$$

Assuming $\boldsymbol{\Sigma}_{(a,c)}^{-1}$ is close to $\boldsymbol{\Sigma}^{-1}$ which makes $\boldsymbol{H}(c|a,\boldsymbol{\mu})$ small, we can use Equation (4.23) in Equation (4.24) with the following substitutions $\boldsymbol{A} = \boldsymbol{\Sigma}^{-1}$ and $\boldsymbol{H}(c|a,\boldsymbol{\mu}) = \epsilon\boldsymbol{I}$ to obtain

$$\log\det\boldsymbol{\Sigma}_{(a,c)}^{-1} \approx \log\det\boldsymbol{\Sigma}^{-1} - \mathrm{Tr}[\boldsymbol{\Sigma}\,\boldsymbol{H}(c|a,\boldsymbol{\mu})]\,. \tag{4.25}$$

The probability that the subject gives the response c when presented the alternatives a follows by integrating $p(c|a,\boldsymbol{\alpha})$ over the current posterior. We make a second order Taylor expansion of $p(c|a,\boldsymbol{\alpha})$ around the point $\boldsymbol{\mu}$:

$$\begin{aligned} p(c|a) &= \int \mathrm{d}\boldsymbol{\alpha} p(c|a,\boldsymbol{\alpha})\mathcal{N}(\boldsymbol{\alpha}|\boldsymbol{\mu},\boldsymbol{\Sigma}) \\ &\approx \int \mathrm{d}\boldsymbol{\alpha} p(c|a,\boldsymbol{\alpha}) + (\boldsymbol{\alpha}-\boldsymbol{\mu})^T \frac{\partial p(c|a,\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}}\Big|_{\boldsymbol{\alpha}=\boldsymbol{\mu}} \mathcal{N}(\boldsymbol{\alpha}|\boldsymbol{\mu},\boldsymbol{\Sigma}) \\ &\qquad + \frac{1}{2}(\boldsymbol{\alpha}-\boldsymbol{\mu})^T \frac{\partial^2 p(c|a,\boldsymbol{\alpha})}{\partial\boldsymbol{\alpha}\partial\boldsymbol{\alpha}^T}\Big|_{\boldsymbol{\alpha}=\boldsymbol{\mu}} (\boldsymbol{\alpha}-\boldsymbol{\mu})\mathcal{N}(\boldsymbol{\alpha}|\boldsymbol{\mu},\boldsymbol{\Sigma}) \\ &= p(c|a,\boldsymbol{\mu}) + \frac{1}{2}\mathrm{Tr}\left[\boldsymbol{\Sigma}\frac{\partial^2 p(c|a,\boldsymbol{\alpha})}{\partial\alpha\partial\alpha^T}\Big|_{\boldsymbol{\alpha}=\boldsymbol{\mu}}\right] . \end{aligned}$$

The first order term cancels since the gradient is zero at the maximum solution $\boldsymbol{\alpha} = \boldsymbol{\mu}$. In a lowest order approximation we can ignore the correction upon $p(c|a, \boldsymbol{\alpha})$ to obtain

$$\begin{aligned} \text{Variance-logdet}(a) &= -\sum_c p(c|a,\boldsymbol{\mu}) \log\det \boldsymbol{\Sigma}_{(a,c)} + \log\det\boldsymbol{\Sigma} \\ &= -\sum_c p(c|a,\boldsymbol{\mu})[-\log\det(\boldsymbol{\Sigma}_{(a,c)}^{-1}) + \log\det(\boldsymbol{\Sigma}^{-1})] \\ &\approx -\sum_c p(c|a,\boldsymbol{\mu})\mathrm{Tr}[\boldsymbol{\Sigma}\boldsymbol{H}(c|a,\boldsymbol{\mu})] , \end{aligned}$$

where for the last approximation we used the approximation from Equation (4.25). To obtain the proof of this lemma we use Lemma 4.6.1 in the appendix at the end of the paper which states a relationship between Hessian and Fisher matrices. □

Using the same type of approximation, the Committee criterion can be approximated to the same form given in Equation (4.21).

Lemma 4.3.2 *In a lowest order approximation the Committee criterion can be written as*

$$\mathit{Committee}(a) \approx \frac{1}{2}\sum_c p(c|a,\bar{\boldsymbol{\mu}})\boldsymbol{g}(c|a,\bar{\boldsymbol{\mu}})^T\tilde{\boldsymbol{\Sigma}}\,\boldsymbol{g}(c|a,\bar{\boldsymbol{\mu}}) ,$$

where $\bar{\boldsymbol{\mu}}$ is the mean of the hierarchical prior learned from the other subjects and

$$\tilde{\boldsymbol{\Sigma}} \equiv \frac{1}{M}\sum_{m=1}^{M}(\boldsymbol{\mu}^m - \bar{\boldsymbol{\mu}})(\boldsymbol{\mu}^m - \bar{\boldsymbol{\mu}})^T - (\boldsymbol{\mu}-\bar{\boldsymbol{\mu}})(\boldsymbol{\mu}-\bar{\boldsymbol{\mu}})^T .$$

We make a second order Taylor expansion of the KL divergences from the definition of the Committee criterion in Equation (4.12)

$$\mathrm{KL}[\bar{p}(\cdot|a)||p(\cdot|a)] = \sum_c p(c|a,\bar{\boldsymbol{\mu}})\log\left[\frac{p(c|a,\bar{\boldsymbol{\mu}})}{p(c|a,\boldsymbol{\mu})}\right] ,$$

around the point $\bar{\boldsymbol{\mu}}$.

The first order term of the Taylor expansion is:

$$-\sum_c p(c|a,\bar{\boldsymbol{\mu}})\frac{\partial \log p(c|a,\bar{\boldsymbol{\mu}})}{\partial \boldsymbol{\mu}}\Big|_{\boldsymbol{\mu}=\bar{\boldsymbol{\mu}}}(\boldsymbol{\mu}-\bar{\boldsymbol{\mu}})^T = \sum_c p(c|a,\bar{\boldsymbol{\mu}})\boldsymbol{g}(c|a,\bar{\boldsymbol{\mu}})(\boldsymbol{\mu}-\bar{\boldsymbol{\mu}})^T$$

which cancels since based on Equation (4.28) from the appendix $\sum_c p(c|a,\bar{\boldsymbol{\mu}})\boldsymbol{g}(c|a,\bar{\boldsymbol{\mu}}$ is the vector with every component 0.

The second order term can be rewritten using Lemma 4.6.1 as:

$$\begin{aligned}
-\frac{1}{2}\sum_c & p(c|a,\bar{\boldsymbol{\mu}})(\boldsymbol{\mu}-\bar{\boldsymbol{\mu}})^T\boldsymbol{H}(c|a,\bar{\boldsymbol{\mu}})(\boldsymbol{\mu}-\bar{\boldsymbol{\mu}}) \\
&= \frac{1}{2}(\boldsymbol{\mu}-\bar{\boldsymbol{\mu}})^T\sum_c p(c|a,\bar{\boldsymbol{\mu}})\boldsymbol{g}(c|a,\bar{\boldsymbol{\mu}})\boldsymbol{g}(c|a,\bar{\boldsymbol{\mu}})^T(\boldsymbol{\mu}-\bar{\boldsymbol{\mu}}) \\
&= \frac{1}{2}\sum_c p(c|a,\bar{\boldsymbol{\mu}})\boldsymbol{g}(c|a,\bar{\boldsymbol{\mu}})^T(\boldsymbol{\mu}-\bar{\boldsymbol{\mu}})(\boldsymbol{\mu}-\bar{\boldsymbol{\mu}})^T\boldsymbol{g}(c|a,\bar{\boldsymbol{\mu}})\,.
\end{aligned}$$

Since the other terms cancel, we obtain that the KL-divergence between the predictive probabilities can be approximated as

$$\mathrm{KL}[\bar{p}(\cdot|a)||p(\cdot|a)] = \frac{1}{2}\sum_c p(c|a,\bar{\boldsymbol{\mu}})\boldsymbol{g}(c|a,\bar{\boldsymbol{\mu}})^T(\boldsymbol{\mu}-\bar{\boldsymbol{\mu}})(\boldsymbol{\mu}-\bar{\boldsymbol{\mu}})^T\boldsymbol{g}(c|a,\bar{\boldsymbol{\mu}})\,.$$

Making this approximation for all the KL-divergences in the definition of the Committee criterion from Equation (4.12) and computing the sum we obtain the result stated in this lemma. □

Furthermore, it can also be showed that Variance-trace and Change-KL can be approximated to the same form given in Equation (4.21), namely

$$\text{Variance-trace}(a) \approx \sum_c p(c|a,\boldsymbol{\mu})\boldsymbol{g}(c|a,\boldsymbol{\mu})^T\boldsymbol{\Sigma}^2\boldsymbol{g}(c|a,\boldsymbol{\mu})\,, \tag{4.26}$$

and

$$\text{Change-KL}(a) \approx \text{Variance-logdet}(a)\,. \tag{4.27}$$

For the derivations of these approximations see Lemmas 4.6.2 and 4.6.3 in the appendix.

We will focus on the differences between the Variance-logdet criterion (considered as the reference) and the Committee criterion. The differences between their approximations are as follows.

1. The gradients $\boldsymbol{g}(c|a,\cdot)$ are evaluated at different points: the prior hierarchical mean $\bar{\boldsymbol{\mu}}$ and the current posterior mean $\boldsymbol{\mu}$. This effect is small since $\boldsymbol{\mu}$ is still close

enough to $\bar{\mu}$ for a sufficiently accurate approximation of the gradients, in particular at the start of the learning when selecting the right points to label is more important.

2. The current posterior variance Σ is replaced by $\tilde{\Sigma}$. The effect of the precise weighting of the gradients is not so important, and again, at the beginning of learning $\tilde{\Sigma}$ is close to Σ.

The way in which experiments are selected is more important at the beginning of the learning process, when μ is still close to the prior mean $\bar{\mu}$, and $\tilde{\Sigma}$ to Σ.

4.4 Experimental Evaluation

This section presents the experimental evaluation of the framework proposed in this paper for pairwise comparisons data.

4.4.1 Data Sets

The following data sets related to the preferences of people were used in the experimental evaluation.

Letor

We used the OHSUMED data set from Letor 3.0 (Qin et al., 2010). This data set consists of relevance levels assigned to documents with respect to a given textual query. The relevances are assessed by human experts, using three rank scales: definitely relevant, partially relevant, and not relevant. We used the subset of this data set related to Query 1 which contains 138 references with the following labels: 24 definitely relevant, 26 partially relevant, and 88 not relevant. Each of the samples is characterized by a 45-dimensional vector consisting of text features extracted from the titles and abstracts of the documents. The features were normalized. Based on this data set we constructed pairwise preferences belonging to 50 subjects in a way that we describe below. We followed a procedure similar to (Xu et al., 2010) to turn the relevance levels into pairwise preference comparisons. Since such coarse relevance judgements are considered unrealistic in many real-world applications, Xu et al. (2010) proposed to add uniform noise in the range $[-0.5, 0.5]$ to the true relevance levels. This addition preserves the relative order between definitely relevant (resp. partially relevant) documents and partially relevant (resp. not relevant) ones, but randomly breaks ties within each relevance level. To introduce a hierarchical component, we replaced the random tie-breaking of Xu et al. (2010) by a subject-specific one. We do this by changing the uniform noise by a subject (and feature) dependent term as follows. For subject m, a weight vector $\boldsymbol{\alpha}_m$ is drawn from a zero mean fully factorized Gaussian with unit variance, $\boldsymbol{\alpha}_m \sim \mathcal{N}(0, \boldsymbol{I})$. Given features $\boldsymbol{x}_i$, noise terms are then the inner products $\boldsymbol{\alpha}_m^T \boldsymbol{x}_i$, linearly scaled back to the interval $[-0.5, 0.5]$ (not to destroy the relative order of the true relevance levels), and the relevance levels are taken to be the true relevance levels plus these noise terms.

Audio

The second data set is related to people's preferences for sound quality and it consists of evaluations of sound quality from 32 subjects. Each subject performed 576 pairwise comparison listening experiments of the form (a, c), where the alternatives are $a = (\boldsymbol{x}_1, \boldsymbol{x}_2)$ with $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ representing one sound sample processed with two different settings of the hearing-aid parameters, and the choice $c = \{1, 2\}$ denotes which of the two alternatives was preferred by the user. More details about this data can be found in (Arehart et al., 2007).

Art

The third data set is related to people's preferences for art images. The preferences were collected from a web-based survey in which 190 subjects participated, and in which 642 images were available for rating. Each subject was presented a number of images and asked to rate each of them: like/dislike. Each subject rated, on average, around 90 images. We considered the 32 subjects who rated more than 120 images. Each image is described by a 275-dimensional feature vector, with features which characterize the image such as, color, shape, texture, etc. For computational efficiency reasons we used a subset of the 10 most informative features where the informativeness of the features was measured by averaging the correlations between features and observations. More details about this data can be found in (Yu et al., 2003). Note that this data set does not contain pairwise comparisons like the other two data sets. With each instance, a binary label is associated: like or dislike, which makes the learning task on this data set to be a binary classification task. The combination of multi-task and active learning that we propose in this work can be still applied in this case in the same framework which was introduced in Section 4.2 by using the logistic regression model for the classifier.

4.4.2 Protocol

Our experiments use a leave-one-out scheme in which each subject was considered once as the current/test subject for which the preferences need to be learned. For each test subject the learning started with the hierarchical prior learned from the data of the other remaining subjects. The data for the test subject was split into 5 folds, 1 fold was used for training and the rest was used for testing. The training data was used as a pool out of which points were selected for labeling either randomly or actively using one of the active learning criteria. The hierarchical prior was updated based on these data points. After every update predictions were made on the test set using the current model. We used accuracy (percentage of correct predictions among all the predictions) as a measure of performance. The accuracy of the predictions on the test data measures how much we learned about the subject preferences. The results were averaged over the 5 splits and over the subjects.

4.4.3 Performance

The framework that we propose in this work for optimizing preference learning consists of combining the multi-task formalism together with active learning. The multi-task ideas in preference learing are especially useful when the training preference data from a subject is very small. In this situation it makes sense to use the preference data from other subjects as additional information.

Letor

The pairwise comparisons from Letor data set were generated by adding noise in the interval $[-0.5, 0.5]$ such that the relative order between the three relevance levels is preserved, but ties within each relevance level are broken. As a result, different subjects do agree on comparisons between different relevance levels. Thus, the data was constructed to have an underlying common structure in the preference of different subjects. Because of this reason we expect that the multi-task learning would improve the performance. In order to validate this hypothesis, we checked whether the preferences of a new subject can be learned more accurately by using the available preference data from other subjects. We compared the hierarchical model with the method of Chu and Ghahramani (2005a) Gaussian processes for preference learning which assumes no prior information. The hierarchical/community prior was obtained by applying the EM algorithm described in Section 4.2.3 in combination with the semiparametric utility function from Equation (4.4); the hierarchical prior was learned from 20 samples from each of the other subjects. The method of Chu and Ghahramani (2005a) was applied with a Gaussian kernel.

Figure 4.1, left panel, compares the accuracy obtained using the hierarchical model to the Gaussian process method for preference learning of (Chu and Ghahramani, 2005a) in a non-active setting, i.e., for both models the updates are done with training points randomly selected. The prediction accuracy is shown as a function of the number of data points included in the training set for the test subject. The plots show that the improvement obtained with the hierarchical model depends on the size of the training data. This is in accordance with the expectation that the multi-task formalism is suited for situations in which the available training data is small. Figure 4.1, right panel, compares the accuracy obtained with random and active selection and starting from a hierarchical prior. Please note the change in scaling of the y-axis. The active selection was implemented using the Committee criterion. These plots show that the combination between multi-task and active learning indeed improves the performance.

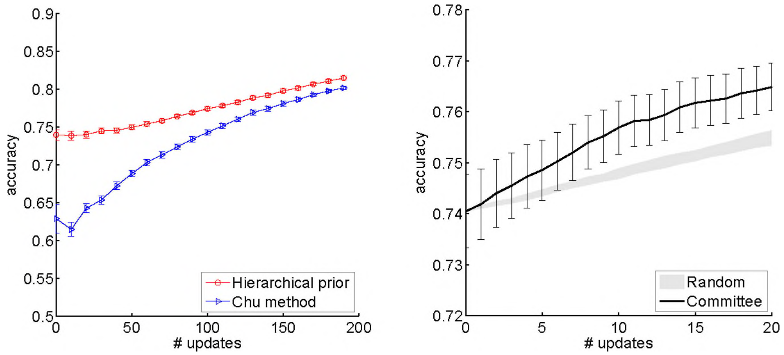

Figure 4.1: Left: Comparison between the hierarchical model that we discussed in Section 4.2.3 and the Gaussian processes for preference learning model (Chu and Ghahramani, 2005a). The setting is non-active, the updates are done using training points randomly selected. Right: Random vs active selection of training points. The performance is evaluated as a function of the number of data points included in the training set. The active selection was implemented using the Committee criterion. The shaded region shows the range of 10 random strategies. The error bars give the standard deviation of the mean accuracy, averaged over the subjects. Please note the change in scaling of the y-axis between the left and right plots.

Audio

Figure 4.2 shows the performance of the Committee criterion on the left and Variance-logdet criterion on the right versus random selection on the audio data set. The plots show the prediction accuracy (on the y-axis) as a function of the number of updates from the hierarchical prior (on the x-axis). The shaded region indicates the accuracy of 10 random selection runs. The error bars give the standard deviation of the mean accuracy, averaged over the 32 subjects. We used the Committee criterion with $\gamma = 0$ since the subjects in the committee are quite similar between each other, which is also suggested by the small error bars. The informative prior improves the predictions at the beginning when no preference observations have been observed for the new subject. The hierarchical prior already gives an accuracy of almost 0.7 for the audio data at the beginning of learning. The hierarchical prior was learned from 20 randomly selected data points per subject. Committee and Variance-logdet strongly overlap and are considerably better than a random strategy. The audio data set contains a few very informative data points and some which are not informative. In some cases the difference between the two sound samples presented in an experiment is so small that the subject cannot hear any difference. Such experiments are not informative because a subject's answer is close to random and does not provide any information with respect to the subject's preferences. The active learning criteria avoid selecting these type of experiments and obtain better performance than random selection. The performance of the other active learning strategies (not shown) is comparable to the active learning strategies shown in Figure 4.2, except for the Vote criterion which does not seem to perform better than random. We refer to Section 4.4.5 for an empirical evaluation of the similarities between the active learning criteria considered in Section 4.3.

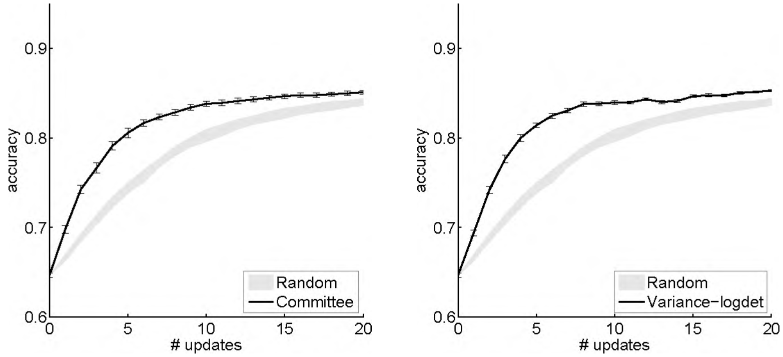

Figure 4.2: Performance of the Committee criterion on the left and Variance-logdet criterion on the right versus random selection for the audio data set. The plots show the prediction accuracy (on the y-axis) as a function of the number of updates from the hierarchical prior (on the x-axis). The error bars give the standard deviation of the mean accuracy, averaged over the 32 subjects. The shaded region shows the range of 10 random strategies.

Art

Figure 4.3 shows the performance of the Committee criterion on the left and Variance-logdet criterion on the right versus random selection on the art data set. The plots show the prediction accuracy (on the y-axis) as a function of the number of updates from the hierarchical prior (on the x-axis). The shaded region indicates the accuracy of 10 random selection runs. The error bars give the standard deviation of the mean accuracy, averaged over the subjects. For the art data which has a higher variability between subjects the Committee criterion with $\gamma = 1$ performs slightly better than the Committee criterion with $\gamma = 0$. The preferences of people for art images are more difficult to predict, since preferences do not depend on some low-level characteristics of the image, like texture, color, etc. This is why the accuracy obtained on the art data is less than the accuracy obtained, for example, on the audio data. The Variance-logdet criterion appears to perform slightly better than the Committee criterion. Furthermore, the benefit of active learning over random selection is much smaller. Like in the case of audio data set, the performance of the other active learning strategies (not shown) is comparable to the active learning strategies shown in Figure 4.3.

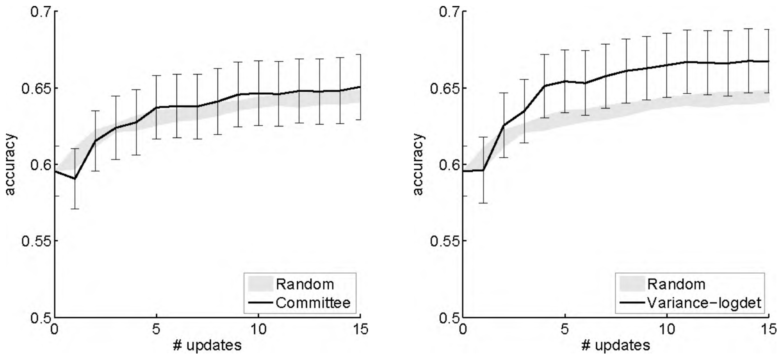

Figure 4.3: Performance of the Committee criterion on the left and Variance-logdet criterion on the right versus random selection for the art data set. The plots show the prediction accuracy (on the y-axis) as a function of the number of updates from the hierarchical prior (on the x-axis). The error bars give the standard deviation of the mean accuracy, averaged over the subjects. The shaded region shows the range of 10 random strategies.

4.4.4 Computational Complexity

One of the main advantages of the Committee criterion is its computational simplicity in comparison with the standard criteria used in experimental design and exemplified here by the Variance-logdet criterion. For every candidate data point to be included in the training set, the Variance-logdet criterion needs to infer the posterior distribution induced. A standard method for performing this approximation is Laplace's method which has a cubic complexity in the dimension of the data features because it involves inversions of the covariance matrices. For more details about other inference methods suited to pairwise comparison data we refer to (Birlutiu and Heskes, 2007). All the algorithms presented here are linear in the number of data points which makes them scalable to large data sets. Table 4.1 shows a comparison between the execution times for Variance-logdet and Committee criterion as a function of feature dimension. Since these execution times for a fixed number of updates (one in our case) do not depend on the actual nature of the data, we randomly generated data in order to be able to change the number of input dimensions. The data was randomly generated to have dimension 10, 50, 100 and 200. The time was evaluated for 100 candidate data points and for 1 update step. The Committee criterion was computed from data belonging to 20 users. The simulations were performed using Matlab on an Intel Xeon processor with 16Gb of memory which runs Fedora release 9 with Linux kernel 2.6.27. In the case of the

Committee criterion, only KL divergences between predictive probabilities are needed to be computed. The Committee criterion is clearly much faster than the Variance-logdet, furthermore, contrary to the Variance-logdet criterion, the computational complexity of the Committee criterion is independent of the dimension of the features.

Table 4.1: Execution time (in seconds) for Variance-logdet and Committee criterion a function of feature dimension.

Feature dimension	Variance-logdet (sec)	Committee (sec)
10	2.894	0.014
50	13.543	0.010
100	37.926	0.009
200	172.661	0.010

4.4.5 Similarities between Criteria

In order to test empirically the approximations and similarities from Section 4.3.4, we computed the Spearman rank correlations (Hollander and Wolfe, 1999) between the scores assigned by the criteria when evaluating the informativeness of the data points. We did this for both the audio and the art data sets. The correlations were computed for both the audio and art data sets. For each subject the learning started with the hierarchical prior learned from the data of the other subjects. This prior was updated by taking into account the information from 20 randomly selected data points for both data sets. After these updates, we computed the scores assigned by each of the active learning criteria to 50 randomly chosen data points. Figure 4.4 shows these Spearman rank correlation coefficients for each pair of criteria; the darker the color the closer to 1 the correlations are and the stronger the similarity between the two criteria. There are several observations to be made from this figure: *i)* One can notice a darker square on the left-down part of the figures, both for the audio and art data set. This square involves the Variance-logdet, Change-KL, Variance-trace, and Committee criteria. The correlations between each pair of them are very close to 1 which suggests that these criteria perform in practice very similar. This is also what the theory from Section 4.3.4 suggests by approximating these criteria to a similar form. *ii)* The Variance-logdet and Change-KL criterion have the Spearman rank correlation extremely close to 1. Their approximations are proven to be equivalent in Lemma 4.6.3 in the Appendix. These two observations also suggest that the approximations of the Variance-logdet and Change-KL are very accurate. *iii)* The Vote criterion performs in some situations randomly since when the number of subjects is much smaller than the number of data points considered, the scores assigned by the Vote criterion are the same to most of the experiments. *iv)* The Uncertainty criterion is most different from the others.

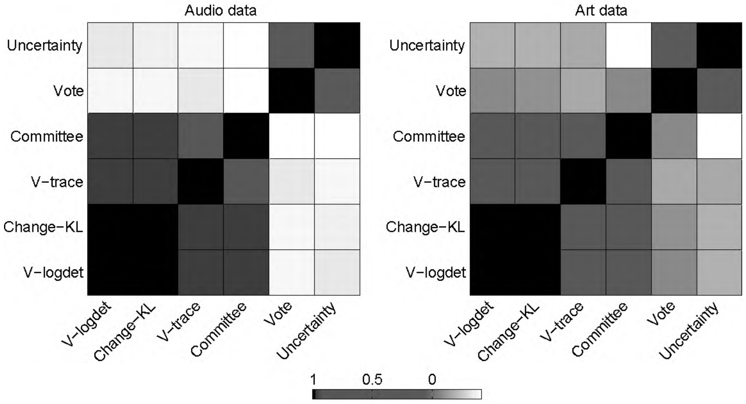

Figure 4.4: Spearman rank correlation coefficients for the scores assigned by the active learning criteria on the audio data (left) and on the art data (right). The darker the color, the higher the correlations, suggesting that the criteria are very similar. The lighter the color, the lower the correlations, suggesting that the criteria are not similar. The correlations between the Variance-logdet and Committee criterion are about 0.95 for the audio data set and around 0.85 for the art data set. The Variance-logdet and Variance-trace criteria are referred in the plots as V-logdet and V-tracce.

4.5 Conclusions and Discussions

This work studied how to exploit models learned on other scenarios to actively learn a model for a new scenario in an efficient way. Our approach to active learning in a multi-scenario setting combines a hierarchical Bayesian prior (to learn from related scenarios) with active learning (to learn efficiently by selecting informative examples). Our new Committee criterion inspired by the Query-by-Committee method is very similar to the standard criteria from experimental design, in particular in the early stages of active learning, but computationally more efficient. Aside from the computational advantage, the Committee criterion introduces the idea to have the data, available from other users, collaborate in order to select the most informative experiments to perform with a new user. The same idea is already implicit in the Query-by-Committee algorithm. We show, theoretically and through experiments, that this conceptual idea also works with a committee of people. This can be interpreted as another way of using people as the elements of a machine learning algorithm, which is a very promising research area, as suggested also by (Sanborn and Griffiths, 2008).

4.5.1 Future Work

There are several directions for extending our work. *i)* A direction worth investigating is a non-myopic design, similar to the one proposed by Boutilier (2002). A non-myopic design "looks" more than just one step ahead when evaluating the informativeness of a data point. It is theoretically closer to the best possible design but computationally much more expensive. Due to the computational complexity involving a non-myopic design, we discussed all the active learning criteria from a myopic perspective, however, a non-myopic perspective can be applied to all of them. *ii)* The end goal of learning the preferences of a person is to make recommendations about an item he/she would like. In this paper we focused on accurately learning the utility function. The criteria discussed in this work could be adapted to the setting in which we focus on finding the item which maximizes the utility function, similar to the criteria for finding the maximum of the utility function proposed in (Groot et al., 2010). *iii)* In this work we used log-linear models and Gaussian distributions to model the preference data. The same idea, of using models learned on data from different subjects (or scenarios) to actively select examples for a new subject, can be applied to other models and starting from different priors as well, although the mathematics will be a bit more involved and less intuitive. In particular, considering a mixture of Gaussians as the prior may still be feasible and may lead to an active learning strategy that tries to find those examples that can best discriminate to which mixture component the current model belongs.

Acknowledgments

We would like to thank Kai Yu for providing the art data set and Wei Chu for making available his code on preference learning with Gaussian Processes.

4.6 Appendix

In this appendix we prove the equivalences between the active learning criteria stated in Section 4.3.4. We show that these criteria can be approximated to the same form, namely

$$\sum_c p(c|a,\boldsymbol{\alpha})\boldsymbol{g}(c|a,\boldsymbol{\alpha})^T\boldsymbol{Q}\boldsymbol{g}(c|a,\boldsymbol{\alpha})\,,$$

for some vector $\boldsymbol{\alpha}$ and matrix $\boldsymbol{Q}$. The difference between the approximations for different criteria is the point $\boldsymbol{\alpha}$ in which the gradients and the probabilities are evaluated and the weighting matrix of the gradients $\boldsymbol{Q}$.

We consider probabilistic choice models of the form given in Equation (4.2), which by using the definition of the utility function from Equations (4.3) and (4.4) can be rewritten as

$$p(c|a,\boldsymbol{\alpha}) = \frac{\exp\left[\sum_{i=1}^{D}\phi_i(\boldsymbol{x}_c)\alpha_i\right]}{Z(\boldsymbol{\alpha},a)} \quad \text{with } Z(\boldsymbol{\alpha},a) \equiv \sum_{c'} \exp\left[\sum_i \phi_i(\boldsymbol{x}_{c'})\alpha_i\right] \; .$$

We define the derivatives of the log probabilities

$$\boldsymbol{g}(c|a,\boldsymbol{\alpha}) \equiv \frac{\partial \log p(c|a,\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}} \, , \qquad \boldsymbol{H}(c|a,\boldsymbol{\alpha}) \equiv \frac{\partial^2 \log p(c|a,\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha} \partial \boldsymbol{\alpha}^T} \; .$$

We first prove a lemma which states a relationship between the Hessian and the Fisher matrices which will be used in further proofs.

Lemma 4.6.1 *For any input a and vector $\boldsymbol{\alpha}$ we have the following relationship between the Hessian and the Fisher matrices:*

$$\sum_c p(c|a,\boldsymbol{\alpha})\boldsymbol{H}(c|a,\boldsymbol{\alpha}) = -\sum_c p(c|a,\boldsymbol{\alpha})\boldsymbol{g}(c|a,\boldsymbol{\alpha})\boldsymbol{g}(c|a,\boldsymbol{\alpha})^T \; ,$$

Proof. We use shorthand notation $p_c = p(c|a,\boldsymbol{\alpha})$, $g_{cj} = g_j(c|a,\boldsymbol{\alpha})$, $\phi_{ci} = \phi_i(\boldsymbol{x}_c)$, omitting the dependencies on a and $\boldsymbol{\alpha}$.

From $\log p_c = \sum_j \phi_{cj}\alpha_j - \log Z$, it is easy to see that

$$g_{cj} = \phi_{cj} - \frac{\partial \log Z}{\partial \alpha_j} , \qquad H_{c,ij} = -\frac{\partial^2 \log Z}{\partial \alpha_i \partial \alpha_j} \; .$$

Furthermore,

$$\begin{aligned}
\frac{\partial \log Z}{\partial \alpha_j} &= \frac{1}{Z}\frac{\partial Z}{\partial \alpha_j} = \frac{1}{Z}\sum_c \exp\left[\sum_{j'} \phi_{cj'}\alpha_{j'}\right] \phi_{cj} = \sum_c p_c \phi_{cj} \; , \\
\frac{\partial^2 \log Z}{\partial \alpha_i \partial \alpha_j} &= \sum_c \phi_{cj} \left[\frac{\exp(\sum_{j'} \phi_{cj'}\alpha_{j'})\phi_{ci} Z - \frac{\partial Z}{\partial \alpha_i}\exp(\sum_{j'} \phi_{cj'}\alpha_{j'})}{Z^2} \right] \\
&= \sum_c \phi_{cj}[\phi_{ci}p_c - \sum_{c'}(p_{c'}\phi_{c'i})p_c] = \sum_c p_c\phi_{cj}\phi_{ci} - \sum_c p_c \phi_{cj} \sum_{c'} p_{c'}\phi_{c'i} ,
\end{aligned}$$

and thus

$$g_{cj} = \phi_{cj} - \sum_{c'} p_{c'}\phi_{c'j} \; , \tag{4.28}$$

$$H_{c,ij} = -\sum_{c'} p_{c'}\phi_{c'j}\phi_{c'i} + \sum_{c'} p_{c'}\phi_{c'j} \sum_{c''} p_{c''}\phi_{c''i} = H_{ij} \; .$$

Note that the second derivative is in fact independent of c. We then have

$$\sum_c p_c H_{c,ij} = \sum_c p_c H_{ij} = H_{ij} = -\sum_c p_c \phi_{ci}\phi_{cj} + \sum_c p_c \phi_{ci} \sum_{c'} p_{c'} \phi_{c'j}$$
$$= -\sum_c p_c \left(\phi_{ci} - \sum_{c'} p_{c'}\phi_{c'i}\right)\left(\phi_{cj} - \sum_{c'} p_{c'}\phi_{c'j}\right) = -\sum_c p_c g_{ci} g_{cj} \, .$$

□

The following lemma proves the approximation of the Variance-trace criterion from Equation (4.26).

Lemma 4.6.2 *In a first order approximation, the Variance-trace criterion boils down to*

$$\textit{Variance-trace}(a) = \sum_c p(c|a, \boldsymbol{\mu}) \boldsymbol{g}(c|a, \boldsymbol{\mu})^T \boldsymbol{\Sigma}^2 \boldsymbol{g}(c|a, \boldsymbol{\mu}) \, .$$

Proof. We have

$$\boldsymbol{\Sigma}_{(a,c)} = \left(\boldsymbol{\Sigma}_{(a,c)}^{-1}\right)^{-1} = \left(\boldsymbol{\Sigma}^{-1} + \boldsymbol{\Sigma}_{(a,c)}^{-1} - \boldsymbol{\Sigma}^{-1}\right)^{-1} \approx -\boldsymbol{\Sigma}\left[\boldsymbol{\Sigma}_{(a,c)}^{-1} - \boldsymbol{\Sigma}^{-1}\right]\boldsymbol{\Sigma} \, .$$

Use of Equation (4.23) and Lemma 4.6.1 gives the result. □

The following lemma proves the approximation of the Change-KL criterion from Equation (4.27).

Lemma 4.6.3 *In a first order approximation, assuming that $\boldsymbol{\Sigma}_{(a,c)}$ is close to $\boldsymbol{\Sigma}$, we have*

$$\textit{Change-KL}(a) \approx \textit{Variance-logdet}(a) \, ,$$

i.e., the two criteria are indistinguishable.

Proof. We evaluate the terms of the Change-KL criterion one by one,

$$\text{Change-KL}(a) = \sum_c p(c|a) \left[\log\left(\frac{\det \boldsymbol{\Sigma}_{(a,c)}}{\det \boldsymbol{\Sigma}}\right) + \mathrm{Tr}\left(\boldsymbol{\Sigma}_{(a,c)}^{-1}\boldsymbol{\Sigma}\right)\right]$$
$$+ \sum_c p(c|a)\left[(\boldsymbol{\mu}_{(a,c)} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}_{(a,c)}^{-1} (\boldsymbol{\mu}_{(a,c)} - \boldsymbol{\mu}) - n\right] \, .$$

The first term gives

$$\sum_c p(c|a) \log \left(\frac{\det \boldsymbol{\Sigma}_{(a,c)}}{\det \boldsymbol{\Sigma}} \right) = -\text{Variance-logdet}(a) \, .$$

The second term

$$\begin{aligned}
\sum_c p(c|a) \operatorname{Tr} \left[\boldsymbol{\Sigma}^{-1}_{(a,c)} \boldsymbol{\Sigma} \right] &= \sum_c p(c|a) \operatorname{Tr}[(\boldsymbol{\Sigma}^{-1} - \boldsymbol{H}(c|a, \boldsymbol{\mu}))\boldsymbol{\Sigma}] \\
&= n - \sum_c p(c|a) \operatorname{Tr}[H(c|a, \boldsymbol{\mu})\boldsymbol{\Sigma}] = n - \sum_c p(c|a, \boldsymbol{\mu}) \operatorname{Tr}[\boldsymbol{\Sigma}\, \boldsymbol{H}(c|a, \boldsymbol{\mu})] \\
&= n + \text{Variance-logdet}(a)
\end{aligned}$$

In the same lowest order, we obtain for the third term

$$\begin{aligned}
&\sum_c p(c|a) \left(\boldsymbol{\mu}_{(a,c)} - \boldsymbol{\mu} \right)^T \boldsymbol{\Sigma}^{-1}_{(a,c)} (\boldsymbol{\mu}_{(a,c)} - \boldsymbol{\mu}) \approx \\
&\qquad \approx \sum_c p(c|a, \boldsymbol{\mu}) \boldsymbol{g}(c|a, \boldsymbol{\mu})^T \boldsymbol{\Sigma}\, \boldsymbol{g}(c|a, \boldsymbol{\mu}) = \text{Variance-logdet}(a) \, .
\end{aligned}$$

Collection of all the terms then gives the result. □

Part II

Supervised Network Inference

Chapter 5

Bayesian Framework for Protein-Protein Interaction Prediction

The reconstruction of protein-protein interaction networks is nowadays an important challenge in systems biology. Computational approaches can address this problem by complementing high-throughput technologies and by helping and guiding biologists in designing new laboratory experiments. The proteins and the interactions between them form a network, which has been shown to possess several topological properties. In addition to information about proteins and interactions between them, knowledge about the topological properties of these networks can be used to learn accurate models for predicting unknown protein-protein interactions. This paper presents a principled way, based on Bayesian inference, for combining network topology information jointly with information about proteins and interactions between them. The goal of this combination is to build accurate models for predicting protein-protein interactions. We define a random graph model for generating networks with topology similar to the ones observed in protein-protein interaction networks. We define a probability model for protein features given the absence/presence of an interaction, and combine these with the random graph model by using Bayes' rule, to finally arrive at a model incorporating both topological and feature information.[1]

[1] This chapter is based on the work: A. Birlutiu, F. d'Alché-Buc and T. Heskes "Combining protein and network topology information for predicting protein-protein interactions." *(under review).*

5.1 Introduction

Knowledge about protein-protein interactions (PPIs) is essential to the understanding of the cellular functions and biological processes inside a living cell. Deciphering the entire network of PPIs of an organism is a very complex task since these interactions can only be established by costly and tedious laboratory experiments. Computational techniques for predicting PPIs have become standard tools to address this problem, complementing their experimental counterparts. Accurately predicting which proteins might interact can help in designing and guiding future laboratory experiments. Therefore, developing computational methods that can accurately predict PPIs is currently an active research area.

A recent trend in computational approaches for predicting PPIs is to frame this problem in a supervised learning setting. That is, information about proteins and labels for protein pairs as interacting or not, supervise the estimation of a function that can predict whether an interaction exists or not between two proteins. PPI prediction can thus be seen as a pattern recognition problem, i.e., find patterns in the interacting protein pairs that do not exist in the non-interacting pairs. This can be further framed as a binary classification problem which takes as input a set of features for a protein pair and gives as output a label: interact or non-interact. Binary classification has been studied extensively in machine learning community, and many algorithms designed to solve it have been also applied for predicting PPIs, including Bayesian networks (Jansen et al., 2003), kernel-based methods (Ben-Hur and Noble, 2005; Yamanishi et al., 2004), logistic regression (Lin et al., 2004; Sprinzak et al., 2006), decision trees and random forest based methods (Zhang et al., 2004; Qi et al., 2005; Chen and Liu, 2005), metric or kernel learning (Yamanishi et al., 2004) and (Geurts et al., 2006, 2007b,a).

In addition to information about proteins and interactions between them, PPI networks are characterized by several topological properties (Jeong et al., 2001; Maslov and Sneppen, 2002; Friedel and Zimmer, 2006; Przulj et al., 2004; Tanaka et al., 2005). Network topology can uncover important biological information that is independent of other available biological information. One of the most important topological properties is the existence of a few nodes in the networks, called hubs, which have many links with the other nodes, while most of the nodes have just a few links. This characteristic is present in PPI networks and also in other real-world networks, such as the internet and citation networks. Topology only has been shown to be able to predict protein functions (Milenkovic and Przulj, 2008) and PPIs (Kuchaiev et al., 2009) and to complement sequence information in various biological tasks, like for example, homology detection (Memisevic et al., 2010). Summarizing, we can distinguish two types of information that can be used for predicting PPIs: first, information about proteins and labels for protein pairs as interacting or not, and second, information about topological properties of PPI networks. These two sources of information can complement each other and are both valuable for constructing performant models which can accurately predict interactions between proteins.

In this contribution, we present a principled way of combining topology and feature information for constructing models for predicting PPIs. We combine models that have been previously used for modeling each type of information separately. We use a random graph generator for addressing the topology information and a naive Bayes model for addressing the feature information. We show that by making a few simplifying assumptions, both topological and protein information can be incorporated and we show experimentally that this improves the prediction accuracy in two PPI networks.

5.2 Models and Methods

The approach that we use to combine topology and feature information is graphically summarized in Figure 5.1. It consists of a random graph generator model and a naive Bayes model which are combined using Bayes' rule to finally arrive to a logistic regression model (we will ignore for the moment the details of this figure but come back to it throughout the section). The random graph generator gives rise to networks which based on topology can all be plausible hypotheses for the PPI network that we want to reconstruct. Incorporating the actual data will reduce this set of plausible hypotheses to just a few, out of which we can pick the one which has the highest likelihood. We implement this in a Bayesian framework by treating our random graph model as a prior and define a probability model for the features given the absence/presence of an edge and combine these two using Bayes' rule, to finally arrive at a model incorporating both topological and feature information. The way in which each of these models is constructed and then combined is detailed in the rest of this section.

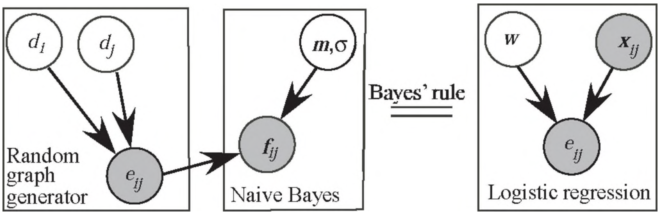

Figure 5.1: Graphical representation of the model which combines topology and feature information. Left box: random graph generator model. Center box: naive Bayes model. Right box: the result of applying Bayes' rule, the model which combines topology and feature information.

5.2.1 Topological Properties of PPI Networks

We will focus on one essential topological characteristics of PPI networks: the node degree distribution. The degree of a node represents the number of connections the node has with the other nodes in the network. The probability distribution of these degrees over the whole network, $p(k)$, is defined as the fraction of nodes in the network with degree k,

$$p(k) = \frac{N_k}{N},$$

where N is the total number of nodes in the network and N_k is the number of nodes with degree k. The majority of real-world networks have a node degree distribution that is highly right-skewed, which means that most of the nodes have low degrees, while a small number of nodes, known as "hubs", have high degrees. The degree of hubs is typically several order of magnitudes larger than the average degree of a node in the network. This property is a distinctive characteristic of PPI networks as well (Jeong et al., 2001), although the reason why some proteins interact with a multitude of proteins and others interact with only a few is not well understood. It has been shown that the connectivity of a protein is related to its function (Ekman et al., 2006), high connectivity is often associated with proteins involved in information storage and processing (transcription in particular) and cellular processes and signaling. Among the non-hubs, there are many proteins that participate in metabolism, while proteins with poorly characterized functions frequently have few or no interactors.

5.2.2 Random Graph Generator

The first step of our approach is to define a model for generating networks with the node degree distribution similar to the one of PPI networks (the left-hand side box of Figure 5.1). The random graph generator that we define here is inspired by the general random graph method (Chung and Lu, 2002). The general random graph method assigns each node with its expected degree and edges are inserted probabilistically according to a probability proportional to the product of the degrees of the two endpoints, i.e., the probability of an edge between two nodes i and j is proportional to the product of the expected degrees of the nodes i and j. We introduce a latent variable, d_i, related to the degree of node i, i.e., d_i is roughly proportional to the degree of node i. Let e_{ij} be a random variable with two possible values: $e_{ij} = 1$ if a link is present between nodes i and j, and $e_{ij} = -1$ if there is no link. In Figure 5.1, the random variables d_i and d_j are represented by white color circles because they are unobserved while e_{ij} is represented by a gray color circle because it is observed. Our model generates links in the network as follows,

$$p(e_{ij} = 1|d_i, d_j) = \frac{d_i d_j}{1 + d_i d_j},$$

$$p(e_{ij} = -1|d_i, d_j) = \frac{1}{1 + d_i d_j},$$

which can be rewritten as

$$p(e_{ij}|d_i, d_j) \propto \exp\left[e_{ij}\frac{1}{2}(\log d_i + \log d_j)\right] . \tag{5.1}$$

The random graph generator can generate networks with a desired topology, more specifically with a desired node degree distribution, by assuming a well-chosen distribution for the latent variable associated with the node degree, i.e, d_i. The first choice for the distribution over d_i would be a power-law distribution which is in general used for modeling the degree distribution of PPI networks (Jeong et al., 2001). Networks with a power law distribution for node degrees are referred to as scale-free networks (Barabási and Albert, 1999) and include among others the world wide web (Broder et al., 2000), metabolic networks (Jeong et al., 2000), citation networks (Redner, 1998). An exponential distribution for d_i and d_j in Equation (5.1) gives rise to a scale-free network (Chung and Lu, 2002). A log-normal distribution is another option for modeling the node degree distribution of scale-free networks (Pennock et al., 2002). Power-law and log-normal distributions are intrinsically connected in the sense that similar generative models can lead to either power law or log-normal distributions (Mitzenmacher, 2003). For computational reasons which will become clear later, we consider a log-normal distribution for d_i, this means that $\log d_i$ is normally distributed,

$$p(\log d_i) = \mathcal{N}(\log d_i; m_0, \sigma_0^2) , \tag{5.2}$$

where m_0 is a scaling parameter, and the parameter σ_0 controls the shape of the distribution. These parameters can be fit such that the networks randomly generated with the model from Equation (5.1) have the desired topology. We have defined d_i to be roughly proportional to the degree of node i, thus a log-normal distribution for d_i results in a distribution for the degree of node i which is approximately log-normal, which is similar to what is observed in practice.

In summary, the random graph generator for a given topology performs the following steps. 1) Choose m_0 and σ_0 the parameters of the log-normal distribution for d_i. 2) Draw from this distribution a random sample $(d_1, \ldots, d_N)$ of size N the number of nodes in the network. 3) Based on this sample construct the network by inserting edges with probability given in Equation (5.1).

Figure 5.2 shows the node degree distributions of three networks randomly generated with the method described above and starting from three different log-normal distributions for d_i, left: $m_0 = -3$, $\sigma_0 = 1$, center: $m_0 = 0$, $\sigma_0 = 1$ and right: $m_0 = 3$, $\sigma_0 = 1$. The first network is very sparse, with a connectivity of 5% (the percentage of actual links from the total number of possible links); the node degree distribution has, in this case, an exponential decay, thus we can consider this network as having a scale-free architecture. The second and third networks have connectivities of 50% and 95%, respectively, and are quite far from the topology of PPI networks. For $m_0 < -3$ the network becomes more sparse and for $m_0 > 3$ the network becomes more

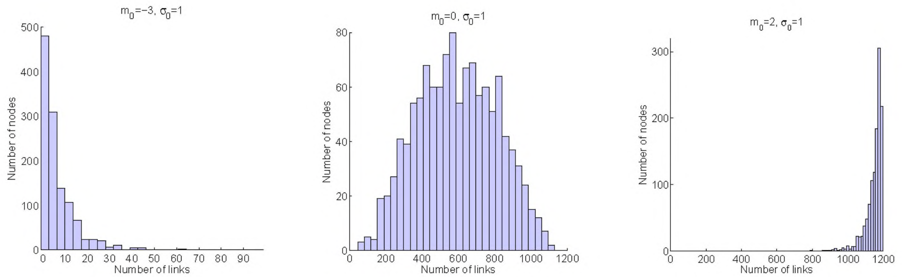

Figure 5.2: Node degree distributions of three networks randomly generated from three different log-normal distributions for d_i; left: $m_0 = -3$, $\sigma_0 = 1$, center: $m_0 = 0$, $\sigma_0 = 1$ and right: $m_0 = 3$, $\sigma_0 = 1$.

connected. The parameter σ controls the width of the distribution.

The histograms in Figure 5.3 compare the node degree distribution in two types of networks: 1) PPI networks observed in two species: yeast and human (the histograms on the left-hand side) and 2) networks randomly generated from the random graph generator defined above (the histograms on the right-hand side). The description of the PPI networks for yeast and human is given in Section 5.3.1. The parameters of the log-normal distribution for d_i were set such that the histograms of the random networks are similar to the histograms of the PPI networks, for the random network from the first row: $m_0 = -3.7$ and $\sigma_0 = 1$ and for the random network from the second row: $m_0 = -4.2$ and $\sigma_0 = 1.11$. These histograms show that the random graph generator that we defined indeed yields networks with node degree distribution very similar to those observed in practice. Based on Figures 5.2 and 5.3 an appropriate choice for the parameters of the log-normal distribution that we will use in the rest of this work is $m_0 = -3$ and $\sigma_0 = 1$.

5.2.3 Bayesian Framework for Combining Topology and Feature Information

In order to combine the topology and feature information, we treat the random graph model as a prior and define a probability model for the protein pairs features given the absence/presence of an interaction. We make use of a a naive Bayes model to express the likelihood of a protein pairs feature given the absence/presence of an interaction. The likelihood is thus computed as a product of 1-dimensional Gaussian distributions, each Gaussian distribution expressing the probability of a feature component f_{ij}^k given the

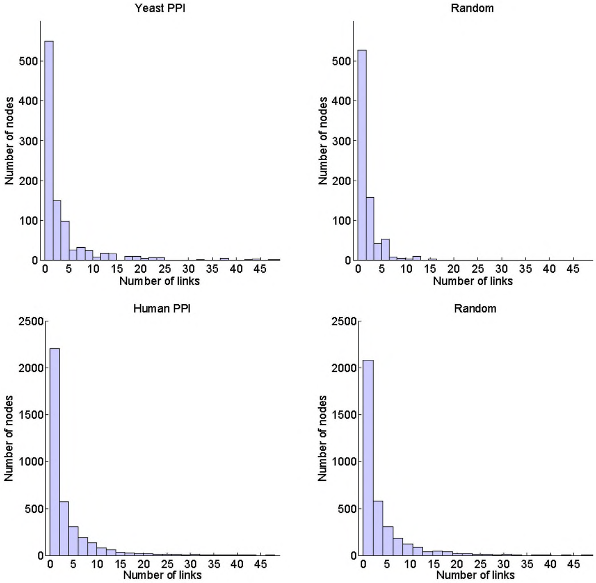

Figure 5.3: Left: histograms of node degrees of yeast PPI network (top row) and human PPI network (bottom row); the description of these networks is given in Section 5.3.1. Right: histograms of node degrees of two random networks generated with the model from Section 5.2.2 and with the parameters of the log-normal distribution for the latent variables d_i: $m_0 = -3.7$ and $\sigma_0 = 1$ (top row) and $m_0 = -4.2$ and $\sigma_0 = 1.11$ (bottom row).

edge variable e_{ij} and the parameters mean m_k and variance σ,

$$p(\boldsymbol{f}_{ij}|e_{ij}, \boldsymbol{m}, \sigma) = \prod_{k=1}^{D} \mathcal{N}(f_{ij}^k; m_k e_{ij}, \sigma)$$

$$\propto \prod_{k=1}^{D} \exp\left(-\frac{(f_{ij}^k - e_{ij}m_k)^2}{2\sigma^2}\right) . \tag{5.3}$$

We refer to the center box of Figure 5.1 for a graphical representation of this model. The naive Bayes model defined above treats the features as independent, which might not be the case in practice. Despite this simplifying assumption, the naive Bayes model is known to be a competitive classification method, with similar performance as the closely related logistic regression algorithm.

The posterior distribution for e_{ij} which combines topology and feature information is computed using Bayes' rule as the product between the prior defined in Equation (5.1) and the likelihood terms defined in Equation (5.3), i.e.,

$$p(e_{ij}|\boldsymbol{f}_{ij}, d_i, d_j) \propto p(e_{ij}|d_i, d_j)p(\boldsymbol{f}_{ij}|e_{ij}, d_i, d_j)$$

$$\propto \exp\left(e_{ij}\frac{1}{2}(\log d_i + \log d_j) - \frac{\sum_k (f_{ij}^k - e_{ij}m_k)^2}{2\sigma^2}\right) \tag{5.4}$$

$$\propto \exp\left(e_{ij}\frac{1}{2}(\log d_i + \log d_j) + \frac{e_{ij}\sum_k f_{ij}^k m_k}{\sigma^2}\right) \tag{5.5}$$

$$\propto \exp\left(e_{ij}(\sum_{k=1}^{D} \frac{f_{ij}^k m_k}{\sigma^2} + \frac{1}{2}\log d_i + \frac{1}{2}\log d_j)\right) \tag{5.6}$$

where when going from (5.4) to (5.5) we discarded the square terms. In the above, we can ignore any term that does not depend on e_{ij}, since it will only affect the normalization. This includes the term $e_{ij}^2 m_k^2/\sigma^2$, since $e_{ij} \in \{-1, 1\}$. The normalization term does play a role and, when incorporated, leads to Equation (5.8) below. The unknown quantities of our model are $\frac{m_k}{\sigma^2}$, $k = \{1, \ldots, D\}$ and $\log d_i$, $i = \{1, \ldots, N\}$, and these will be estimated based on the available training data in a learning procedure that we describe below.

The first step is to adjoin the unknown quantities in a single random variable, that is

$$\boldsymbol{w} = [\frac{m_1}{\sigma^2}, \ldots, \frac{m_D}{\sigma^2}, \frac{1}{2}\log d_1, \ldots, \frac{1}{2}\log d_N] , \tag{5.7}$$

and the same for the information available, that is protein features and topological information

$$\boldsymbol{x}_{ij} = [\boldsymbol{f}_{ij}, \boldsymbol{t}_{ij}] ,$$

where $\boldsymbol{t}_{ij}$ is the position vector having 1 on positions i and j and 0 everywhere else. Then, the normalized probability that there is an interaction between the proteins i and j from Equation (5.6) can be rewritten as

$$p(e_{ij}|\boldsymbol{x}_{ij}, \boldsymbol{w}) = \frac{1}{1 + \exp(-2e_{ij}\boldsymbol{w}^T\boldsymbol{x}_{ij})} . \tag{5.8}$$

Note that in the sum

$$\boldsymbol{w}^T\boldsymbol{x}_{ij} = \sum_{k=1}^{D} w^k f_{ij}^k + \sum_{k=1}^{N} w^{D+k} t_{ij}^k , \tag{5.9}$$

the first term on the right-hand side originates from the protein features information and the second term from the topological information.

The unknown parameter $\boldsymbol{w}$ is learned in a Bayesian framework which consists in setting a prior distribution for it, and updating this prior based on observations. The update is performed using Bayes' rule given below

$$p(\boldsymbol{w}|\text{observations}) \propto \prod_{o=1}^{n_{\text{obs}}} p(e_{ij}^o|\boldsymbol{x}_{ij}^o, \boldsymbol{w})p(\boldsymbol{w}) . \tag{5.10}$$

where n_{obs} is the size of the training data, i.e., the number of known interacting/non-interacting protein pairs, and $p(e_{ij}^o|\boldsymbol{x}_{ij}^o, \boldsymbol{w})$ is given in Equation (5.8). $p(\boldsymbol{w})$ is the prior and we choose it to be a Gaussian distribution

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) .$$

The hyperparameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ of the prior are chosen such that the topological information is included in the model. This is implemented by making the correspondence with the prior for the latent variables d_i. Recall from Equation (5.7) that $w_{i+D} = \frac{1}{2}\log d_i, i = 1, \ldots, N$ and from Equation (5.2) that $\log d_i$ is normally distributed, consequently w_{i+D} will also be normally distributed, i.e.,

$$w_{i+D} \sim \mathcal{N}\left(\frac{m_0}{2}, \frac{\sigma_0^2}{4}\right) , i = 1, \ldots, N .$$

A good choice for the hyperparameters m_0 and σ_0 was discussed in relation to Figures 5.2 and 5.3. Thus, we set $\boldsymbol{\mu}_{D+1:N} = \frac{m_0}{2} = -1.5$ which corresponds to a network with a node degree distribution of the form shown in the histogram from the left-side of Figure 5.2. The hyperparameters μ_i, $i = 1, \ldots, D$ that correspond to the feature information were set to 0, and the covariance matrix $\boldsymbol{\Sigma}$ was chosen to be the identity matrix (this choice for the parameters of the prior corresponding to protein features makes sense when the features are normalized like we do in the experimental evaluation). We will also see in the experimental evaluation, Section 5.3.5, that the choice of the prior's parameters has a big influence on the performance.

The vectors $\boldsymbol{x}_{ij}$ are sparse because their components $\boldsymbol{t}_{ij}$ of dimension N contain only two non-zero elements on positions i and j. This sparsity property can be exploited for making the computations more efficient. Predictions can be done for an unknown interaction between a pair of proteins i', j' characterized by the feature vector $\boldsymbol{x}_{i'j'}$. These predictions can be done either averaging the posterior over $\boldsymbol{w}$ in Equation (5.8) or by using a point estimate of this posterior, let $\boldsymbol{w}^*$ be the mean of $p(\boldsymbol{w}|\text{observations})$, and computing $p(e_{i'j'}|\boldsymbol{x}_{i'j'}, \boldsymbol{w}^*)$ using Equation (5.8).

We refer back to the graphical sketch of our model in Figure 5.1 at the beginning of this section. The box on the left-hand side, corresponds to the random graph generator model. The observation e_{ij}, which expresses the presence or absence of an edge between nodes i and j, depends on the latent variables d_i and d_j which are related to the degrees of nodes i and j. The random graph generator model incorporates feature information through the naive Bayes model with unknown parameters $\boldsymbol{m}$ and σ, represented in the center box. The combination of the two models is obtained using Bayes' rule. The result is shown in the right-hand side box. The unknown quantities d_i, d_j, and $\boldsymbol{m}$, σ are combined in the node $\boldsymbol{w}$ which is unobserved, and $\boldsymbol{f}_{ij}$ together with $\boldsymbol{t}_{ij}$ which is implicitly expressed by indices i and j form the observed quantity $\boldsymbol{x}_{ij}$.

Summarizing, in order to incorporate topological information for PPI prediction we propose a relatively simple method: logistic regression on an extended feature space. The extended feature space is obtained by adding to the feature vector $\boldsymbol{f}_{ij}$ for a pair of proteins i and j, a vector of dimension N with 1 on positions i and j and 0 everywhere else. The regression weights are treated as latent variables and those weights corresponding to the additional topology features are in a one-to-one correspondence with the latent variables d_i of the random graph generator. The scale-free like architectures of the random graph generator follow from a log-normal prior distribution on these d_is.

In the experimental evaluation from Section 5.3 we will compare four models. All the models are based on Equation (5.10) with a Gaussian prior and likelihood terms of the form given in Equation (5.8) and they vary in the way of computing the dot product from Equation (5.9) and on the parameters of the Gaussian prior.

1. Model 1 (Features+Topology): is the model we propose in this work. It makes use of the following dot product

$$\boldsymbol{w}^T \boldsymbol{x}_{ij} = \sum_{k=1}^{D} w^k f_{ij}^k + \sum_{k=1}^{N} w^{D+k} t_{ij}^k\,, \tag{5.11}$$

   and a Gaussian prior with mean $\boldsymbol{\mu}_{1:D} = 0$, $\boldsymbol{\mu}_{D+1:N} = \frac{-3}{2} = -1.5$ and covariance matrix equal to the identity matrix.

2. Model 2 (Features only): uses only information about proteins, and the dot product

is computed as

$$\boldsymbol{w}^T \boldsymbol{x}_{ij} = \sum_{k=1}^{D} w^k f_{ij}^k + w^{D+1} . \tag{5.12}$$

The second term on the right-hand side of Equation (5.12) is a bias term to address the unbalancedness of the data. This bias term also corresponds to the second term on the right-hand side of Equation (5.11); for an edge e_{ij} the contributions in Equation (5.11) are $w_{D+i} + w_{D+j}$ while in Equation (5.12) we constraint $w_{D+i} = \frac{1}{2} w_{D+1}$, $\forall i = 1, \ldots, N$. This observation also motivates the choice of the prior for this model: mean $\boldsymbol{\mu}_{1:D} = 0$ and $\mu_{D+1} = -3$ and covariance equal to the identity matrix.

3. Model 3 (Topology only): uses only topology information and the dot product is computed as

$$\boldsymbol{w}^T \boldsymbol{x}_{ij} = \sum_{k=1}^{N} w^k t_{ij}^k .$$

The Gaussian prior is of dimension N with mean equal to the vector $\boldsymbol{\mu}_{1:N} = -1.5$ and covariance matrix equal to the identity matrix. The choice for $\boldsymbol{\mu}_{1:N} = -1.5$ corresponds to the log-normal distribution with $m_0 = -3$, thus to a network with a node degree distribution of the form of the left-side plot from Figure 5.2.

4. Model 4 (Topology-enriched features): uses the information about proteins and about topology in the following form

$$\boldsymbol{w}^T \boldsymbol{x}_{ij} = \sum_{k=1}^{D} w^k f_{ij}^k + w^{D+1} \log(\hat{d}_i + 1) + w^{D+2} \log(\hat{d}_j + 1) ,$$

where $\hat{d}_i$ and $\hat{d}_j$ are the estimated degrees of nodes i and j computed on the training data. Basically, the features $\boldsymbol{f}_{ij}$ for a pair of proteins i and j are being extended by adding two new columns corresponding to the degrees of nodes i and j computed on the training set. For computational reasons we considered the logarithms of node degrees to which we added 1. The idea behind this model is similar to the one used in (Tastan et al., 2009; Qi et al., 2010), i.e., the topological features are added to protein features resulting in an enriched set of features. The features are being standardized and the parameters of the Gaussian prior are set to $\boldsymbol{\mu}_{1:D+2} = 0$ and covariance equal to the identity matrix.

5.3 Results

The four models previously described were empirically evaluated and the results are presented in this section.

5.3.1 Data Sets

We used two data sets. Details for each of them are given below.

Yeast Data

This data set was borrowed from (Geurts et al., 2007a) and it consists of the high confidence physical interactions between proteins highlighted in (von Mering et al., 2002). The PPI network has 984 nodes (proteins) connected by 2438 links (interactions). We consider all the protein pairs not present in the 2438 interactions as non-interacting. The yeast PPI graph is very sparse, as a result the data is highly unbalanced, with less than 1% from the total examples belonging to the positive class. Each protein has associated a vector of dimension 157 representing gene expression values in various experiments. We constructed the features for protein pairs by summing the individual protein features. The degree distribution of the nodes is shown in the top-left plot on Figure 5.3.

Human Data

This data set was created and made available by (Qi et al., 2007) and consists of protein pairs with an associated label: interact or non-interact. Each pair of proteins is characterized by a 27-dimensional feature vector. The features were constructed based on Gene Ontology (GO) cell component (1), GO molecular function (1), GO biological process (1), co-occurrence in tissue (1), gene expression (16), sequence similarity (1), homology based (5) and domain interaction (1), where the numbers in brackets correspond to the number of elements contributed by the feature type to the feature vector. Unlike positive interactions, non-interacting pairs are not experimentally reported. Thus, a common strategy is to consider as non-interacting pairs a randomly drawn fraction from the total set of potential protein pairs excluding the pairs known to interact. The resulting data set has 14,608 interacting pairs and 432,197 non-interacting pairs. The PPI graph consists of $24,380$ nodes connected by $14,608$ edges. As in the case of the yeast data set, the PPI graph of the human data is very sparse, the interacting pairs represent less then 1% from the possible links in the graph.

Both data sets are highly unbalanced, with 1% and 5% positive pairs for yeast data and human data, respectively.

5.3.2 Experimental Setup

The experimental setup considered a part of the data for training and the rest for testing. The training data was used to learn the models and the testing data was used to evaluate how good these models can predict PPIs. We randomly sampled a training set containing 1%, 5%, 10% and 20% protein pairs from the yeast and human data set. The training features were standardized to have mean zero and standard deviation of one. This data sample was used to train the classification model (i.e., learn the weight parameter of the logistic regression). The remaining protein pairs were used for testing the performance. These steps were repeated 10 times and average results are reported (mean ± standard deviation).

Evaluation Measure

Area under the receiver operating characteristic curve (AUC) was used as a measure for evaluating the performance. The receiver operator characteristic (ROC) curve plots the true positive rate against the false positive rate for different thresholds. The AUC statistic can be interpreted as the probability that a randomly chosen missing edge (a true positive) is given a higher score by the method than a randomly chosen pair of proteins without an interaction (a true negative).

5.3.3 Performance

Table 5.1 shows the comparison of the performance of the four models discussed in Section 5.2. Model 1 represents the Bayesian framework for combining feature and topology information, Model 2 uses only protein information, Model 3 uses only topology information and Model 4 uses protein features which are enriched with node degrees. The comparison was performed for the yeast data (the four upper rows in Table 5.1) and human data sets (the four lower rows in Table 5.1). The protocol described in Section 5.3.2 was used and the averaged AUC scores with their standard deviations are reported. The statistical significance between Model 1 and Model 2 was assessed by using a Mann-Whitney U-test (Hollander and Wolfe, 1999) on the AUC values obtained from the two models for 10 random splits of the data into training and testing. A 5% significance level has been considered. The * indicates that the results obtained for Model 1 are significantly better than the results obtained for Model 2.

The results show that the combination of the two sources of information, protein features and topology, gives a better performance than using only one type of information. In particular Model 1 (Features+Topology) performs significantly better than Model 2 (Features only) in most of the cases. Model 1 and Model 4 have a similar performance for human data, and Model 1 performs better than Model 4 for yeast data. An explanation for this is related to how the protein features were constructed in the two cases; for yeast data the features for a protein pair resulted from summing the feature vectors corresponding to the two proteins, while for human data the protein features are more

Table 5.1: AUC values (mean ± standard deviation) for the four models presented in Section 5.2. The * indicates that the results obtained for Model 1 are significantly better than the results obtained for Model 2. The four upper rows correspond to the yeast data set while the four lower rows correspond to the human data set.

% Train data	Model 1 Features+ Topology	Model 2 Features only	Model 3 Topology only	Model 4 Topology features	Link Propagation
1%	0.639 ± 0.014	0.639 ± 0.018	0.577 ± 0.016	0.582 ± 0.022	0.592 ± 0.016
5%	0.708 ± 0.006	0.697 ± 0.009	0.688 ± 0.010	0.689 ± 0.009	0.625 ± 0.009
10%	0.731 ± 0.005*	0.712 ± 0.005	0.720 ± 0.006	0.717 ± 0.007	0.650 ± 0.009
20%	0.746 ± 0.009*	0.719 ± 0.006	0.742 ± 0.009	0.737 ± 0.010	0.690 ± 0.008
1%	0.863 ± 0.006*	0.851 ± 0.006	0.608 ± 0.014	0.822 ± 0.012	
5%	0.909 ± 0.002*	0.859 ± 0.001	0.793 ± 0.007	0.899 ± 0.003	
10%	0.931 ± 0.002*	0.861 ± 0.001	0.864 ± 0.005	0.931 ± 0.002	
20%	0.952 ± 0.002*	0.862 ± 0.001	0.917 ± 0.003	0.954 ± 0.002	

related to the protein pair than to individual proteins. The results vary also as a function of the size of the training data. For a small training set the network topology is not well defined, and we can see that in this case the improvement is smaller, but, as we increase the training set, meaning that the knowledge about the network topology increases, the performance obtained by adding the topology information improves more.

The framework based on the logistic regression classifier that we use in this work gives similar performance to other methods for PPI prediction. For the comparison, we show in the last column of Table 5.1 the performance of link propagation method for PPI prediction recently proposed by (Kashima et al., 2009). The results are shown for the yeast data set and using protein features as information. Link propagation could not be applied to the human data since in this case protein features based on which to compute the similarity matrix between proteins are not available. We also tested other classifiers like SVMs and random forests but they did not perform comparable to the logistic regression, probably because the unbalancedness of the data is quite difficult to handle in their case.

5.3.4 Topology Learning

The combination between the protein features and topology information from Model 1 has the best performance in comparison with the other models since it is able to learn faster and more accurate the topology of the network. In order to show this we analyzed how accurate the degrees of the nodes are estimated in the cases of Model 1 (Features+Topology) and Model 2 (Features only). The comparison was performed on the yeast data and using the protocol described in Section 5.3.2. Both models were learned on the training data. The predicted node degrees were computed on the test data by summing the predictive probabilities from Equation (5.8) for edges e_{ij} in which one of the indices i or j corresponds to the node of interest. The estimation of the node degrees for the two models is shown in Figure 5.4 with the x-axis showing the percentage of data used for training, and the y-axis showing the error of the estimates measured using the the root mean square of the difference between the predicted and actual node degrees (left plot) and the root mean square of the difference between the logarithms of the predicted and actual node degrees (right plot). Where the root mean square of the node degrees themselves measures the absolute error in estimated node degrees, which is quite sensitive to correctly estimating the hub nodes, the root mean square of the logarithms measures the relative error. It can be seen that indeed Model 1 (Features+Topology) gives a much better estimate to the actual node degrees in comparison to Model 2 (Features) and this explains also why the performance of Model 1 is better than that of Model 2 as shown also in Table 5.1.

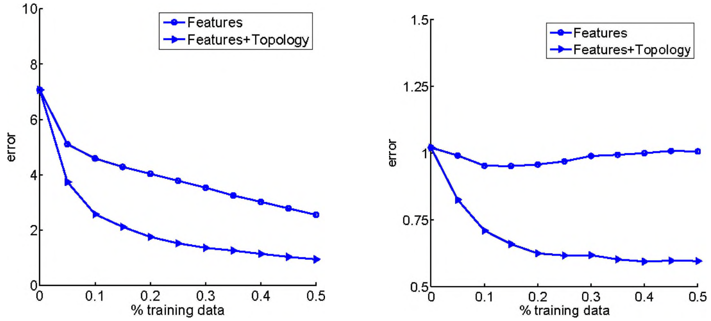

Figure 5.4: The root mean square of the difference between the predicted and actual node degrees (left plot) and between the logarithms of the predicted and actual node degrees (right plot) for the two models: Model 2 (features only) and Model 1 (features+topology) as a function of the percentage of data considered in the training set.

5.3.5 The Influence of the Prior

The prior defined by the random graph generator has a bigger effect especially when the number of observations is small. Furthermore, the hyperparameters of the prior, i.e., the m_0 and σ_0 have also an important influence on the performance. Table 5.2 shows the performance obtained for predicting PPIs using Model 1 for a size of the training data of 1% from the total data set and with three parameter settings for the prior. These correspond to the three parameter settings for the log-normal distribution based on which the histograms from Figure 5.2 were generated. The performance is best for $m_0 = -3$, $\sigma_0 = 1$ which corresponds to the left-hand side histogram in Figure 5.2 and which is also a valid assumption for the topology of PPI networks.

Table 5.2: AUC scores (mean ± standard deviation) for predicting PPIs using Model 1 for three parameter settings for the log-normal distribution (1st column). The results are shown for the yeast data and human data. The size of the training data is 1% from the total data set.

Prior parameter settings	Yeast data	Human data
$m_0 = -3, \sigma_0 = 1$	0.639 ± 0.014	0.863 ± 0.006
$m_0 = 0, \sigma_0 = 1$	0.595 ± 0.015	0.808 ± 0.014
$m_0 = 3, \sigma_0 = 1$	0.566 ± 0.015	0.742 ± 0.029

5.4 Discussions

In addition to the node degree distribution, networks in general, and PPI networks in particular, can be characterized by other global topological properties, including the clustering coefficient, the network diameter, and the average shortest path. (Maslov and Sneppen, 2002) showed the existence in PPI networks of highly inter-connected regions which are correlated with biological functions and large multi-protein complexes. PPI networks are shown to adhere also to the small world phenomenon, i.e., most pairs of proteins are connected to each other by a short chain of links involving several intermediate proteins. In addition to these global topological properties, PPI networks are also characterized by the so-called network motifs. A network motif is a small subgraph which appears in the network significantly more frequently than in a randomized network. Different types of real-world networks have been shown to have different motifs (Milo et al., 2002). We believe that frameworks similar to the one introduced here for incorporating information about the node degree distribution, can be derived for including other types of topological information. The so-called node signature (Milenkovic and Przulj, 2008), which represents the topology in the neighborhood of a node, might be useful in this direction. In the same direction, other random graph generators have to be investigated, like for example, exponential random graph models (Robins et al., 2007)

Treating PPI prediction as a supervised inference problem is not straightforward since data is typically associated to individual proteins while the labels correspond to pairs of proteins. This issue is addressed by constructing features for pairs of proteins from individual protein features. In many cases the problem of identifying high-quality features can in itself be quite difficult. This has led to the development of powerful techniques known as kernel methods. Kernel methods allow the user to specify a particular kind of pairwise function between data objects, known as a kernel function, which is used by the algorithm instead of explicit features. Kernels have been successfully used for constructing features for PPI prediction (Ben-Hur and Noble, 2005).

The logistic regression classifier is a natural choice in our framework since latent variables in the graph generator can be translated to weights in the logistic regressor. Similar ideas may also work in connection to other classifiers and may be used for the reconstruction of other biological networks, such as, metabolic, gene regulatory or signaling networks.

5.5 Appendix: Bayesian Inference

The approach that we use for incorporating topology and feature information brings us to computing the posterior distribution

$$p(\boldsymbol{w}|\text{observations}) \propto \prod_{o=1}^{n_{\text{obs}}} p(e_{ij}^{o}|\boldsymbol{x}_{ij}^{o}, \boldsymbol{w})p(\boldsymbol{w}) \,. \tag{5.13}$$

where the likelihood terms are of the form given in Equation (5.14)

$$p(e_{ij}|\boldsymbol{x}_{ij}, \boldsymbol{w}) = \frac{1}{1 + \exp(-2e_{ij}\boldsymbol{w}^{T}\boldsymbol{x}_{ij})} \,. \tag{5.14}$$

and the prior over $\boldsymbol{w}$ is a Gaussian distribution

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \,,$$

where the hyper-parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ that correspond to $\boldsymbol{t}_{ij}$ are derived from the parameters of the log-normal distribution for d_i, specifically $\boldsymbol{\mu}_{D+1:N} = -3$, $\boldsymbol{\mu}_{1:D} = 0$ and the covariance matrix $\boldsymbol{\Sigma}$ is the identity matrix.

$$\boldsymbol{w}^{*} \equiv \underset{\boldsymbol{w}}{\operatorname{argmin}} -L(\boldsymbol{w})$$

where

$$\begin{aligned} -L(\boldsymbol{w}) &= -\sum_{o=1}^{n_{\text{obs}}} \log p(e_{ij}^{o}|\boldsymbol{x}_{ij}^{o}, \boldsymbol{w}) - \log p(\boldsymbol{w}) \\ &= \sum_{o=1}^{n_{\text{obs}}} \log(1 + \exp(-2e_{ij}^{o}\boldsymbol{w}^{T}\boldsymbol{x}_{ij}^{o})) + \frac{1}{2}(\boldsymbol{w} - \boldsymbol{\mu})^{T}\boldsymbol{\Sigma}^{-1}(\boldsymbol{w} - \boldsymbol{\mu}) + c \,, \end{aligned}$$

where c is a constant term independent of $\boldsymbol{w}$. The optimization is solved by using conjugate gradient for which we compute the first derivatives of $L(\boldsymbol{w})$ with respect to $\boldsymbol{w}$.

$$\begin{aligned}\frac{\partial L(\boldsymbol{w})}{\partial \boldsymbol{w}^T} &= \sum_{o=1}^{n_{\text{obs}}} -\frac{\partial}{\partial \boldsymbol{w}^T} \log(1+\exp(-2e_{ij}^o \boldsymbol{w}^T \boldsymbol{x}_{ij}^o)) + (\boldsymbol{w}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} \\ &= \sum_{o=1}^{n_{\text{obs}}} \frac{-2e_{ij}^o \boldsymbol{x}_{ij}^o}{1+\exp(2e_{ij}^o \boldsymbol{w}^T \boldsymbol{x}_{ij}^o)} + (\boldsymbol{w}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} \,.\end{aligned}$$

These computations can be simplified by taking into account the specific form of the vectors $\boldsymbol{x}_{ij}$, i.e., the sparse form of the part corresponding to the topology information.

Chapter 6

Link Transfer for Protein-Protein Interaction Prediction using Multiple Species

Protein-protein interaction network inference has attracted interest of machine learning researchers as a typical problem of structured data mining. Mining the protein interactions graph of a species can give useful information to biologists about which proteins might interact. Deciphering biological networks is crucial for the understanding of the cellular functions and biological processes inside a living cell. As a consequence, the reconstruction of these networks is currently a major challenge with important applications in medicine and biology. In this study, we investigate how to incorporate the information available from several reference species, i.e., transfer learning, in order to improve the performance of protein-protein network inference for a target organism. We propose a method based on a so-called converter function from the reference species to the target species. The underlying idea of the converter is to increase the training set of the target species by converting the output space of the reference species to the output space of the target species.[1]

[1] This chapter is based on the work: F. d'Alché-Buc, A. Birlutiu, C. Brouard, T. Heskes and M. Szafranski "Regularized Output Kernel Regression for protein-protein interaction prediction: application to link transfer and transduction." *Machine Learning in Computational Biology workshop at Neural Information Processing Systems conference, 2010.*

6.1 Introduction

There has been a recent interest in devising new techniques and improving existing ones for mining various structured data types such as the ones arising, for example, in biology. Mining the protein interactions graph of a species, also known as protein-protein interactions (PPI) network inference (Vert, 2008), can give useful information to biologists about which proteins might interact. The basic idea of PPI network inference is to use the known edges of the network to supervise the estimation of a classifier whose input is a pair of nodes and outputs a binary value that codes for the presence of a physical interaction between two proteins. The training data used for this task are usually input feature vectors that represent information about the proteins and a given adjacency matrix that codes for the known interactions. This settings allows to automatically learn which features of the data are the most informative to predict the presence of an interaction between two proteins. This task is also known as link prediction in social networks. Among supervised link prediction approaches, let us cite pairwise SVM based on tensor kernel (Ben-Hur and Noble, 2005), metric or kernel learning (Yamanishi et al., 2004) and (Geurts et al., 2006, 2007b,a), and local approaches developed in (Bleakley et al., 2007).

In parallel, bioinformatics researchers have defined other strategies that consist in mapping known interactions between a reference organism onto a target organism and this for the orthologous genes: this is called the protein-protein interologs approach (Michaut et al., 2008). Predicting using interologs is based on the theory that proteins interacting in one organism co-evolve such that their respective orthologs maintain the ability to interact in another organism. Recent works which study the prediction of PPIs based on ortholog information with other species are (Lee et al., 2008; Lehner and Fraser, 2004; Persico et al., 2005; Wiles et al., 2010). As far as PPI networks as well as the homology between protein sequences are available for potential reference organisms, this strategy sounds relevant if data are not too noisy. In this work, we define a new task of link prediction, we call it “link transfer”, that resembles the interolog approach while remaining in the supervised learning framework. The underlying idea of link transfer is to use PPI networks of other species, we call these reference species, to constrain the training of a supervised predictor of PPI in a target species. This paradigm thus differs from the classical transfer learning or multi-task learning settings (Evgeniou et al., 2005; Bakker and Heskes, 2003) but corresponds to a realistic setting of PPI network inference.

We formulate this new task in the framework of output kernel learning (Weston et al., 2003; Cortes et al., 2005; Geurts et al., 2006, 2007b,a). The basic idea of this framework is to learn a mapping from inputs to a feature space associated with the outputs. The key aspect here is that the existing structure in the outputs can be exploited in the learning. We investigate how to incorporate the information available from the reference species in order to improve the performance of the output kernel regressor. We propose to use output kernel regression twice, first to convert output feature vectors from a reference species to the target species and then to learn the target network. The underlying idea of the converter is to increase the training set of the target species by converting the output space of the reference species to the output space of the target species.

This chapter is organized as follows. In Section 6.2 we describe the general framework of output kernel regression for PPI network inference. Its extension to link transfer is presented in Section 6.3. In Section 6.4 we evaluate it empirically using yeast as the target species. We conclude in Section 6.5 with some discussions of related approaches and directions for future research.

6.2 Regularized Output Kernel Regression

In this section we introduce the general framework of Output Kernel Regression for protein-protein network inference. We consider a single target species. Let $\mathcal{P}$ be the set of proteins in the target species. During the training phase, $\mathcal{T}$ a subset of n proteins, and the adjacency matrix for the interactions between the corresponding n proteins are given. These available data are encoded into:

- A Gram matrix $\boldsymbol{K}^{(X)}$ whose coefficients are defined from some positive definite kernel function: $\boldsymbol{K}^{(X)}(i,j) = \kappa^{(X)}(p_i,p_j)$, $p_i, p_j \in \mathcal{P}$.
- A Gram matrix $\boldsymbol{K}^{(Y)}$ that codes for the proximity of proteins as nodes in the interaction graph only known for the proteins of $\mathcal{T}$. We use here the diffusion kernel (Kondor and Lafferty, 2002) matrix $\boldsymbol{K}^{(Y)} = \exp(-\beta \boldsymbol{L}^{(Y)})$ where $\boldsymbol{L}^{(Y)} = \boldsymbol{D} - \boldsymbol{J}$ with $\boldsymbol{J}$ the adjacency matrix given for the n proteins and $\boldsymbol{D}$ the corresponding degree matrix.

We assume that there is an (unknown) positive definite kernel function $\kappa_Y : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$, that when given two proteins as input, reproduces the corresponding element of the kernel matrix, i.e., $\forall p_i, p_j \in \mathcal{P}, \kappa_Y(p_i,p_j) = \boldsymbol{K}^{(Y)}(i,j)$. Let $\mathcal{Y}$ be the associated (possibly infinitely dimensional) feature space endowed with the kernel $\kappa^{(Y)}$, i.e., there is some feature map $\boldsymbol{y}(\cdot) : \mathcal{P} \to \mathcal{Y}$ such that

$$\forall p, p',\ \kappa^{(Y)}(p,p') = \boldsymbol{y}(p)^T \boldsymbol{y}(p') .$$

Throughout the derivations below, we will act as if the mapping $\boldsymbol{y}(\cdot)$ is known, i.e., as if we can always compute any feature vector $\boldsymbol{y}(p)$. Later we will realize that all we actually need are inner products between different feature vectors. Similarly, we will use the mapping $\boldsymbol{x}(\cdot) : \mathcal{P} \to \mathcal{X}$ corresponding to the kernel function $\kappa^{(X)}$, and as for the mapping $\boldsymbol{y}(\cdot)$ all we need are actually inner products that can be represented by kernels. Let us define $f : \mathcal{P} \times \mathcal{P} \to \{0,1\}$ a classifier whose input is a pair of protein features and which outputs a binary value that indicates the presence or absence of an interaction between those proteins. Knowing $\kappa^{(Y)}$ we can define the classifier f by thresholding the kernel:

$$f_\theta(p,p') = \mathrm{sgn}(\kappa^{(Y)}(p,p') - \theta) .$$

However, we do not know $\kappa^{(Y)}$ but only the corresponding Gram matrix $\boldsymbol{K}^{(Y)}$, defined for the proteins of the training set. In the framework of output kernel regression, we propose to approximate $\kappa^{(Y)}$ by using a dot product between images of a hypothesis function $\boldsymbol{h} : \mathcal{P} \rightarrow \mathcal{Y}$,

$$\kappa^{(Y)}(p, p') \approx \boldsymbol{h}(p)^T \boldsymbol{h}(p') \,. \tag{6.1}$$

Learning the classifier reduces thus to learning $\boldsymbol{h}$, a function that uses the kernel trick in the output space. This new learning task has been referred to as Output Kernel Regression in previous works (Geurts et al., 2006, 2007b,a) and was tackled by extending regression trees to output kernel feature space. In this work we focus on Regularized Output Kernel Regression, a recently proposed model (Brouard et al., 2011) that shares the same form as SVMs and the Maximum Margin Robot (Szedmak et al., 2005).

Suppose we are given a training set of size n, consisting of input features of dimension m, denoted by the m−by−n matrix $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$, where $\boldsymbol{x}_i = \boldsymbol{x}(p_i)$, and corresponding output features of dimension q denoted by the q−by−n matrix $\boldsymbol{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n\}$, where $\boldsymbol{y}_i = \boldsymbol{y}(p_i)$. Following the approach of (Cortes et al., 2007), we assume a linear model from input to output, i.e.,

$$\boldsymbol{h}(p) = \boldsymbol{A}\boldsymbol{x}(p) \,. \tag{6.2}$$

We find the weights $\boldsymbol{A}$ through a regularized least-squares regression, i.e., by minimizing a regularized least square loss:

$$\sum_{i=1}^{n} \| \boldsymbol{A}\boldsymbol{x}_i - \boldsymbol{y}_i \|^2 + \lambda \| \boldsymbol{A} \|_F^2 \,, \tag{6.3}$$

which is equivalent to

$$\| \boldsymbol{A}\boldsymbol{X} - \boldsymbol{Y} \|_F^2 + \lambda \| \boldsymbol{A} \|_F^2 \;.$$

where $\| \cdot \|_F$ stands for the Frobenius norm. The optimal solution is found by computing the gradient, setting it to zero and solving for $\boldsymbol{A}$ (for the complete derivation, see (Cortes et al., 2007)):

$$\hat{\boldsymbol{A}} = \boldsymbol{Y}\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{X}^T + \lambda \boldsymbol{I}_n)^{-1} = \boldsymbol{Y}(\boldsymbol{K}^{(X)} + \lambda \boldsymbol{I}_n)^{-1}\boldsymbol{X}^T \,. \tag{6.4}$$

where $\boldsymbol{I}_n$ is the identity matrix of dimension n. So, we find that the linear model $\boldsymbol{h}(p)$ is of the form $\boldsymbol{h}(p) = \boldsymbol{Y}\boldsymbol{C}\boldsymbol{X}^T\boldsymbol{x}(p)$ with $\boldsymbol{C} = (\boldsymbol{K}^{(X)} + \lambda \boldsymbol{I}_n)^{-1}$. If we would start from this form and then solve for $\boldsymbol{C}$, we would find the exact same solution for $\boldsymbol{C}$. A simplified version of this model was considered in (d'Alché Buc et al., 2010), where $\boldsymbol{C}$

is now constrained to be a diagonal matrix:

$$\boldsymbol{h}(p) = \sum_{i=1}^{n} a_i \boldsymbol{y}_i \kappa_X(p_i, p) = \boldsymbol{Y}\text{diag}(\boldsymbol{a})\boldsymbol{X}^T \boldsymbol{x}(p), \ , \tag{6.5}$$

where diag$(\boldsymbol{a})$ is a diagonal matrix with the parameter vector $\boldsymbol{a}$ on the diagonal. This parameter vector is again fitted through a regularized least squares regression, where the regression term only depends on $\boldsymbol{a}$ (see the appendix at the end of this chapter for details). We will refer to the model (6.5) as the diagonal model and to the model (6.2) as the full model. Last but not least, it is easy to see that (6.1) indeed can be expressed in terms of inner products between (sets of) input features and between (sets of) output features. These inner products can be represented by kernels.

6.3 Link Transfer

Besides the target species, we have additional, so-called reference species. For the moment we consider only one reference species, and later we will discuss how to consider multiple reference species. Some of the target proteins are orthologs of the proteins of this reference species. For the reference species, we do not have input features, but there is available an adjacency matrix based on which we can compute a kernel matrix $\boldsymbol{K}^{(W)}$. We further assume that we have a corresponding kernel function $\kappa^{(W)}(\cdot, \cdot)$ with corresponding feature space $\mathcal{W}$ and mapping $\boldsymbol{w}(\cdot)$. As before, for now we assume that we can always compute any feature vector $\boldsymbol{w}(p)$. The link transfer task consists in adding the information contained in the PPI network of the reference species/species to help the prediction task for the target species.

6.3.1 Converter

The transfer learning is based on a converter function from the reference species to the target species. The idea is to increase the training information on which the mapping $\boldsymbol{h}$ is learned by incorporating the data from the reference species.

The set of orthologs is a subset of all proteins, $\mathcal{O} \subseteq \mathcal{P}$. Furthermore the set of orthologs $\mathcal{O}$ can be divided into two subsets $\mathcal{O}_1 \subseteq \mathcal{T}$ and $\mathcal{O}_2 \not\subseteq \mathcal{T}$. We notice that the two adjacency matrices of the target species and the reference species define two different Hilbert spaces: the Hilbert space $\mathcal{Y}$ spanned by the images of $\boldsymbol{y}(p_i), p_i \in \mathcal{T}$ and the Hilbert space $\mathcal{W}$ spanned by the images of $\boldsymbol{w}(p_i), p_i \in \mathcal{O}$. In order to cope with these two different spaces, we use an output kernel regressor $\boldsymbol{u}$ that converts for a given protein p, $\boldsymbol{w}(p)$ into $\boldsymbol{y}(p)$. As a first step, we aim to compute a linear mapping $\boldsymbol{u} : \mathcal{W} \rightarrow \mathcal{Y}$, from the reference feature space $\mathcal{W}$ to the target feature space $\mathcal{Y}$, with $\boldsymbol{u}(\boldsymbol{w}) = \boldsymbol{B}\boldsymbol{y}$. We learn this mapping using pairs of reference proteins and corresponding ortholog target proteins. We refer to the corresponding input feature matrix as $\boldsymbol{W}_{\mathcal{O}_1}$ and to the corresponding output feature matrix as $\boldsymbol{Y}_{\mathcal{O}_1}$. That is, $\boldsymbol{W}_{\mathcal{O}_1}$ and $\boldsymbol{Y}_{\mathcal{O}_1}$ represent a particular subset of the features corresponding to the reference and target species, respectively. Using the approach presented in the previous section, the optimal $\hat{\boldsymbol{B}}$ which minimizes

$$\| \boldsymbol{B}\boldsymbol{W}_{\mathcal{O}_1} - \boldsymbol{Y}_{\mathcal{O}_1} \|_F^2 + \lambda_c \| \boldsymbol{B} \|_F^2 \ .$$

is found to obey

$$\hat{\boldsymbol{B}} = \boldsymbol{Y}_{\mathcal{O}_1}(\boldsymbol{W}_{\mathcal{O}_1}^T \boldsymbol{W}_{\mathcal{O}_1} + \lambda_c \boldsymbol{I}_{|\mathcal{O}_1|})^{-1} \boldsymbol{W}_{\mathcal{O}_1}^T \ , \tag{6.6}$$

where $\boldsymbol{I}_{|\mathcal{O}_1|}$ is the identity matrix of dimension $|\mathcal{O}_1|$, i.e., the number of proteins in the set $\mathcal{O}_1$.

Next we would like to use the learned mapping to improve link prediction for the target species. We stick to the same linear model $\boldsymbol{h}(p) = \boldsymbol{A}\boldsymbol{x}(p)$. We not only aim to fit this model to the target features themselves, but also to the target features estimated from the ortholog features. To this end, we use the two subsets of target proteins with indicators $\mathcal{T}$ and $\mathcal{O}_2$. For the subset $\mathcal{T}$, we aim to fit $\boldsymbol{h}$ to $\boldsymbol{y}$, whereas for subset $\mathcal{O}_2$, we aim to fit $\boldsymbol{h}$ to $\hat{\boldsymbol{u}}(\boldsymbol{w}) = \hat{\boldsymbol{B}}\boldsymbol{w}$. Let $\boldsymbol{X}_{\mathcal{T}}$, $\boldsymbol{Y}_{\mathcal{T}}$, $\boldsymbol{X}_{\mathcal{O}_2}$, $\hat{\mathrm{U}}_{\mathcal{O}_2}$, and $\boldsymbol{W}_{\mathcal{O}_2}$ refer to the various input and feature subsets, with $\hat{\mathrm{U}}_{\mathcal{O}_2} = \hat{\boldsymbol{B}}\boldsymbol{W}_{\mathcal{O}_2}$. The goal is then to find the weight matrix $\hat{\boldsymbol{A}}$ that minimizes the criterion

$$J(\boldsymbol{A}) = \| \boldsymbol{A}\boldsymbol{X}_{\mathcal{T}} - \boldsymbol{Y}_{\mathcal{T}} \|_F^2 + \gamma \| \boldsymbol{A}\boldsymbol{X}_{\mathcal{O}_2} - \hat{\mathrm{U}}_{\mathcal{O}_2} \|_F^2 + \lambda \| \boldsymbol{A} \|_F^2 \ , \tag{6.7}$$

where λ and γ are two (regularization) constants. If we simply concatenate the (weighted) inputs and outputs, e.g., by defining

$$\boldsymbol{X}_{\mathcal{D}} \equiv (\boldsymbol{X}_{\mathcal{T}}, \sqrt{\gamma}\boldsymbol{X}_{\mathcal{O}_2}) \ \text{ and } \ \boldsymbol{Y}_{\mathcal{D}} \equiv (\boldsymbol{Y}_{\mathcal{T}}, \sqrt{\gamma}\hat{\mathrm{U}}_{\mathcal{O}_1}) \ ,$$

we obtain the standard form

$$J(\boldsymbol{A}) = \| \boldsymbol{A}\boldsymbol{X}_{\mathcal{D}} - \boldsymbol{Y}_{\mathcal{D}} \|_F^2 + \lambda \| \boldsymbol{A} \|_F^2 \ ,$$

with corresponding optimal solution

$$\hat{\boldsymbol{A}} = \boldsymbol{Y}_{\mathcal{D}}(\boldsymbol{X}_{\mathcal{D}}^T \boldsymbol{X}_{\mathcal{D}} + \lambda \boldsymbol{I}_{|\mathcal{D}|})^{-1} \boldsymbol{X}_{\mathcal{D}}^T \ .$$

In the above, we acted as if the features $\boldsymbol{y}$, $\boldsymbol{w}$, and $\boldsymbol{x}$ are actually known. Typically, they are not, and we can only compute inner products. Luckily, this happens to be all we need. In particular we need to be able to approximate the kernel function between two proteins o and o'. We have

$$\begin{aligned} \kappa^{(Y)}(p, p') &= \boldsymbol{y}(p)^T \boldsymbol{y}(o') \approx \boldsymbol{h}(p)^T \boldsymbol{h}(o') = \boldsymbol{x}(p)^T \hat{\boldsymbol{A}}^T \hat{\boldsymbol{A}} \boldsymbol{x}(p) \\ &= \boldsymbol{x}(p)^T \boldsymbol{X}_{\mathcal{D}}(\boldsymbol{X}_{\mathcal{D}}^T \boldsymbol{X}_{\mathcal{D}} + \lambda \boldsymbol{I}_{|\mathcal{D}|})^{-1} \boldsymbol{Y}_{\mathcal{D}}^T \boldsymbol{Y}_{\mathcal{D}} (\boldsymbol{X}_{\mathcal{D}}^T \boldsymbol{X}_{\mathcal{D}} + \lambda \boldsymbol{I}_{|\mathcal{D}|})^{-1} \boldsymbol{X}_{\mathcal{D}}^T \boldsymbol{x}(p) \ . \end{aligned}$$

Furthermore

$$\boldsymbol{X}_{\mathcal{D}}^T\boldsymbol{X}_{\mathcal{D}} = \begin{pmatrix} \boldsymbol{X}_{\mathcal{T}}^T\boldsymbol{X}_{\mathcal{T}} & \sqrt{\gamma}\boldsymbol{X}_{\mathcal{T}}^T\boldsymbol{X}_{\mathcal{O}_2} \\ \sqrt{\gamma}\boldsymbol{X}_{\mathcal{O}_2}^T\boldsymbol{X}_{\mathcal{T}} & \gamma\boldsymbol{X}_{\mathcal{O}_2}^T\boldsymbol{X}_{\mathcal{O}_2} \end{pmatrix}$$

and

$$\boldsymbol{Y}_{\mathcal{D}}^T\boldsymbol{Y}_{\mathcal{D}} = \begin{pmatrix} \boldsymbol{Y}_{\mathcal{T}}^T\boldsymbol{Y}_{\mathcal{T}} & \sqrt{\gamma}\boldsymbol{Y}_{\mathcal{T}}^T\hat{\mathbf{U}}_{\mathcal{O}_1} \\ \sqrt{\gamma}\hat{\mathbf{U}}_{\mathcal{O}_1}^T\boldsymbol{Y}_{\mathcal{T}} & \gamma\hat{\mathbf{U}}_{\mathcal{O}_1}^T\hat{\mathbf{U}}_{\mathcal{O}_1} \end{pmatrix}$$

where

$$\begin{aligned} \boldsymbol{Y}_{\mathcal{T}}^T\hat{\mathbf{U}}_{\mathcal{O}_1} &= \boldsymbol{Y}_{\mathcal{T}}^T\boldsymbol{Y}_{\mathcal{O}_1}(\boldsymbol{W}_{\mathcal{O}_1}^T\boldsymbol{W}_{\mathcal{O}_1} + \lambda_c\boldsymbol{I}_{|\mathcal{O}_1|})^{-1}\boldsymbol{W}_{\mathcal{O}_1}^T\boldsymbol{W}_{\mathcal{O}_2} \\ \hat{\mathbf{U}}_{\mathcal{O}_1}^T\hat{\mathbf{U}}_{\mathcal{O}_1} &= \boldsymbol{W}_c^T\boldsymbol{W}_{\mathcal{O}_1}(\boldsymbol{W}_{\mathcal{O}_1}^T\boldsymbol{W}_{\mathcal{O}_1} + \lambda_c\boldsymbol{I}_{|\mathcal{O}_1|})^{-1}\boldsymbol{Y}_{\mathcal{O}_1}^T\boldsymbol{Y}_{\mathcal{O}_1} \\ &\quad \cdot (\boldsymbol{W}_{\mathcal{O}_1}^T\boldsymbol{W}_{\mathcal{O}_1} + \lambda_c\boldsymbol{I}_{|\mathcal{O}_1|})^{-1}\boldsymbol{W}_{\mathcal{O}_1}^T\boldsymbol{W}_{\mathcal{O}_2} \,. \end{aligned}$$

This idea can be extended to include the information from multiple reference species by adding extra terms in the optimization from Equation (6.7), each extra term corresponding to one reference species.

The derivation of the link transfer using the diagonal model defined in Equation (6.5) is given in the appendix from the end of this chapter.

6.3.2 Link Propagation

A simpler idea than the converter function is inspired from the link propagation used in (Kashima et al., 2009). The intuition is that on ortholog proteins the value of the diffusion kernel for the reference species on p_i and p_j weights how similar $\boldsymbol{h}(p_i)$ and $\boldsymbol{h}(p_j)$ are to each other.

$$J(\boldsymbol{A}) = \parallel \boldsymbol{A}\boldsymbol{X}_{\mathcal{T}} - \boldsymbol{Y}_{\mathcal{T}} \parallel_F^2 + \lambda \parallel \boldsymbol{A} \parallel_F^2 + \gamma \sum_{i,j\in\mathcal{O}_2} \kappa^{(W)}(p_i, p_j)||\boldsymbol{A}\boldsymbol{X}_i - \boldsymbol{A}\boldsymbol{X}_j||^2 \,. \tag{6.8}$$

6.4 Empirical Evaluation

In this section we evaluate empirically the transfer learning approaches described in the previous section.

6.4.1 Data

We considered the baker's yeast (*Saccharomyces cerevisiae*) as the target organism. We used the yeast PPI network data of high-confidence physical protein-protein interactions also used in (Yamanishi et al., 2004; Bleakley et al., 2007; Geurts et al., 2007a). It consists of 2438 interactions that link 984 proteins. Each protein was associated with its gene expression, its location and its phylogenetic profile which was used to construct the input kernel. The following species were considered as reference species: *Schizosaccharomyces pombe* –fission yeast, *Mus musculus* –house mouse, *Arabidopsis thaliana* –plant. The PPI networks of the reference species were extracted from the String.db database (*http://string-db.org/*). This database has 7 types of interactions between proteins (neighborhood, fusion, occurrence, coexpression, experiments, database, textmining) from which we considered only the interactions which were validated by laboratory experiments. The set of orthologs between the target species and each of the reference species was obtained from the Inparanoid database (*http://inparanoid.sbc.su.se/*). The fission yeast has 271 orthologs with the target species, the mouse has 147 orthologs and the plant has 120 orthologs. In addition to the PPI networks of these three reference species, we also considered an artificially constructed PPI network. The nodes of the artificial PPI network have a one-to-one mapping with the orthologs that the target species has with fission yeast, thus 271 proteins. The absence/presence of links in the artificial PPI network corresponds to the presence/absence of links in the target PPI network.

6.4.2 Protocol

We conducted experiments on the data set described above to determine whether the extra term (or terms for multiple reference species) in the optimization from Equation (6.7) improves the performance. The performance was evaluated as a function of the parameter γ. We fixed the other parameters of the model except γ to its optimal values determined in the no-transfer case, i.e., the parameter of the diffusion kernel for computing $\boldsymbol{K}^{(Y)}$, $\beta = 3$, the parameter of the Gaussian kernel for computing $\boldsymbol{K}^{(X)}$, $\sigma = 4$, the regularization parameter $\lambda = 0.9$ and the regularization parameter for the converter in Equation (6.6), i.e. $\lambda_b = 0.1$. Further, the data set was randomly split 10 times into training and testing with different percentage for the size of the training data 10%, 15% and 20%. The model was learned on the training set for $\gamma \in 0 : 0.1 : 1$ and the performance was measured using area under the ROC curve (AUC) computed on the testing set.

6.4.3 Results

Figure 6.1 shows the performance for the full model as a function of the parameter γ and for different sizes of the training set: 10%, 15% and 20%. For each training set, we evaluated the improvement obtained by the reference species: fission yeast, mouse and plant and the artificial reference species. The error bars give the standard deviation to the mean of the difference in performance between no transfer and transfer settings for the 10 runs with randomly selected training and testing sets. The optimal value $\gamma > 0$ suggests that the information from the reference species improves the performance. The improvement is larger for the reference species which are closer to the target species, i.e., fission yeast and the artificial reference species. Adding more reference species did not

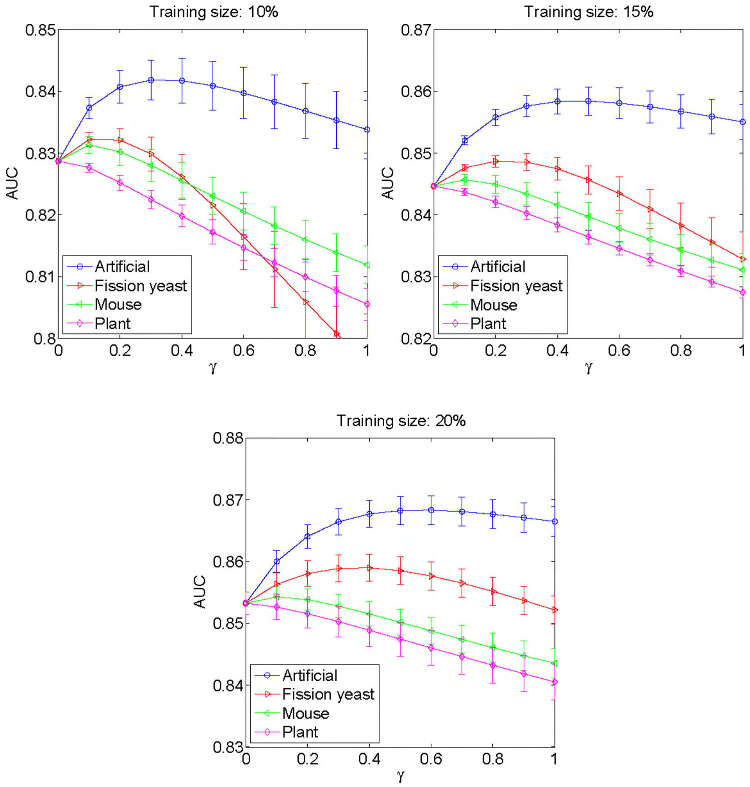

Figure 6.1: Plots of the AUC values as a function of the parameter γ for the full model. The three graphs correspond to three sizes of the training set: 10%, 15% and 20%. The error bars give the standard deviation of the mean of the difference in performance between no transfer and transfer settings for the 10 runs with randomly selected training and testing sets.

significantly improve the performance for the full models. Apparently, the full model is already on this data set quite accurate, which makes it hard to further improve. The diagonal models happens to perform a bit worse on this data set, therefore, we checked whether the addition of more reference species does help in this case.

Figure 6.2 plots the AUC values as a function of the parameter γ for the diagonal model from Equation (6.5). The three plots on the left-side figure correspond to three sizes of the training data, 10%, 15% and 20% and one reference species, the fission yeast. The error bars give the standard deviation of the mean of the difference in performance between no transfer and transfer settings for the 10 runs with randomly selected training and testing sets. Again, the optimal value $\gamma > 0$ suggests that the information from the reference species improves the performance. The improvement is bigger for a small size of the training set and decreases as the training set gets bigger, which is a behavior observed in most of the multi-task learning situations. The plots on the right side are an extension of the three plots from the left-hand side to multiple reference species: results for one reference species (fission yeast) are plotted with solid lines, results for two reference species (fission yeast and plant) are plotted with dashed lines, and results for three reference species (fission yeast, plant and house mouse) are plotted with dotted lines. The plots suggests that including multiple reference species as multiple sources of information increases the performance.

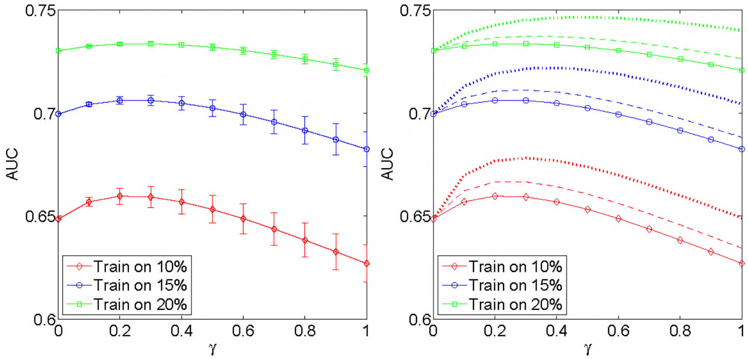

Figure 6.2: Plots of the AUC values as a function of the parameter γ for the diagonal model. Left: The three plots correspond to three sizes of the training data, 10%, 15% and 20%, the error bars give the standard deviation of the mean of the difference in performance between no transfer and transfer settings for the 10 runs with randomly selected training and testing sets. Right: The plots are an extension of the three plots from the left-hand side to multiple reference species: the solid lines are the results obtained one reference species (fission yeast), the dashed lines are the results obtained with two reference species (fission yeast and plant), and the dotted lines are the results for three reference species (fission yeast, plant and house mouse).

Figure 6.3 shows the performance as a function of the parameter γ for the diagonal model and the transfer implemented using the link propagation method from Equation (6.8). The three plots correspond to three sizes of the training data, 10%, 15% and 20%. Using this method, the transfer did not improve the performance compared to the non-transfer setting.

6.5 Conclusions

Transfer learning has been recently considered for predicting PPIs. The approach of (Kashima et al., 2009) is directed to simultaneously learning PPI networks of multiple

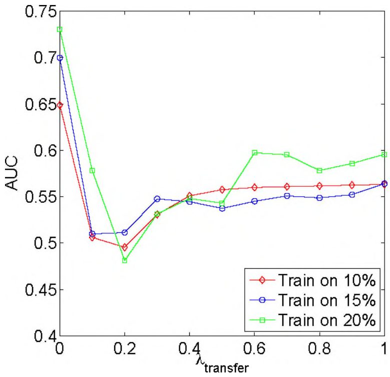

Figure 6.3: Plots of the AUC values as a function of the parameter γ for the diagonal model and the transfer implemented using the link propagation method from Equation (6.8). The three plots correspond to three sizes of the training data, 10%, 15% and 20%.

species in a setting different from ours, i.e., genomic data and PPIs are available for all species, while we consider that genomic data is available only for the target species. Similar work is also (Kato et al., 2010) which applies transfer learning for biological network inference by integrating multiple types of data and use the multi-task approach of Evgeniou et al. (2005) with each learning task belonging to one data assay.

We described a method for transfer learning which increases the training set of a target species using a converter from the output space of the reference species to the output space of the target species. We conducted experiments using baker yeast as the target species. The experiments show that the transfer learning improves the performance, particularly when the reference species is close to the target species and when the training set is of smaller size.

6.6 Appendix

This appendix derives the analytical solution for the diagonal model from Equation (6.5) corresponding to the setting from Section 6.3.1. It is the analog of the optimization problem from Equation (6.7) but for the diagonal model. The expression that we need to optimize is composed of a loss term, a regularizer for the parameter vector $\boldsymbol{a}$ and a transfer term. The loss term and the regularization for $\boldsymbol{a}$ correspond to the non-transfer case. The transfer term corresponds to the extra information used for transfer learning.

$$J(\boldsymbol{a}) = \underbrace{\sum_i ||\boldsymbol{y}(p_i) - \boldsymbol{h}(p_i)||^2}_{\text{Loss}} + \lambda||\boldsymbol{a}||^2 + \gamma \underbrace{\sum_i ||\boldsymbol{u}(\boldsymbol{w}(p_i)) - \boldsymbol{h}(p_i)||^2}_{\text{Transfer}}\,, \qquad (6.9)$$

To shorten the notation a bit, we will work in terms of kernels $\boldsymbol{K}$ instead of inner products $\boldsymbol{X}^T\boldsymbol{X}$.

$$
\begin{aligned}
\text{Loss} &= \sum_i ||\boldsymbol{y}(p_i) - \boldsymbol{h}(p_i)||^2 \\
&= \sum_i \left[K^{(Y)}_{ii} - 2\sum_{i'} a_{i'} K^{(Y)}_{i'i} K^{(X)}_{ii'} + \sum_{i',j'} a_{i'} a_{j'} K^{(Y)}_{i'j'} K^{(X)}_{ii'} K^{(X)}_{ij'} \right] \\
&= \sum_i \left(K^{(Y)}_{ii} - 2\boldsymbol{a}^T (\boldsymbol{K}^{(X)}_{\cdot i} \boldsymbol{K}^{(Y)}_{\cdot i}) + \boldsymbol{a}^T \boldsymbol{K}^{(Y)} * (\boldsymbol{K}^{(X)}_{\cdot i} \boldsymbol{K}^{(X)}_{i\cdot})\boldsymbol{a} \right)
\end{aligned}
$$

where $\boldsymbol{M} * \boldsymbol{M}'$ denotes the element-wise product between matrices $\boldsymbol{M}$ and $\boldsymbol{M}'$, $\boldsymbol{M}_{\cdot i}$ denotes the column-vector corresponding to column i from matrix $\boldsymbol{M}$ and $\boldsymbol{M}_{i\cdot}$ denotes the row-vector corresponding to row i from matrix $\boldsymbol{M}$.

Let $\hat{\boldsymbol{b}}$ be the solution of the conversion function, then

$$
\begin{aligned}
\text{Transfer} &= \sum_i ||\boldsymbol{h}(p_i) - \boldsymbol{u}(\boldsymbol{w}(p_i))||^2 \\
&= \sum_i \sum_{i'j'} a_{i'} a_{j'} K^{(Y)}_{i'j'} K^{(X)}_{ii'} K^{(X)}_{ij'} - 2\sum_i \sum_j \hat{b}_j K^{(W)}_{ij} \sum_{i'} a_{i'} K^{(Y)}_{i'j} K^{(X)}_{i'i} \\
&\quad + \sum_i \sum_{j',j'} \hat{b}_{i'} \hat{b}_{j'} K^{(Y)}_{i'j'} K^{(W)}_{ii'} K^{(W)}_{ij'} \\
&= \sum_i \boldsymbol{a}^T \boldsymbol{K}^{(Y)} * (\boldsymbol{K}^{(X)}_{\cdot i} \boldsymbol{K}^{(X)}_{i\cdot})\boldsymbol{a} - 2\sum_i \boldsymbol{a}^T \boldsymbol{K}^{(Y)} * (\boldsymbol{K}^{(W)}_{\cdot i})\boldsymbol{K}^{(X)}_{i\cdot})\hat{\boldsymbol{b}} \\
&\quad + \sum_i \hat{\boldsymbol{b}}^T \boldsymbol{K}^{(Y)} * (\boldsymbol{K}^{(W)}_{\cdot i} \boldsymbol{K}^{(W)}_{i\cdot})\hat{\boldsymbol{b}}\,.
\end{aligned}
$$

In order to minimize the expression $J(\boldsymbol{a})$, we solve the equation:

$$
\frac{\partial \text{Loss}}{\partial \boldsymbol{a}^T} + 2\lambda \boldsymbol{a} + \gamma \frac{\partial \text{Transfer}}{\partial \boldsymbol{a}^T} = 0\,.
$$

This is equivalent to

$$
\begin{aligned}
&\sum_i \left(-2\boldsymbol{K}^{(Y)}_{\cdot i} \boldsymbol{K}^{(X)}_{i\cdot} + 2\boldsymbol{K}^{(Y)} * (\boldsymbol{K}^{(X)}_{\cdot i} \boldsymbol{K}^{(X)}_{i\cdot})\boldsymbol{a} \right) + 2\lambda \boldsymbol{I}\boldsymbol{a} \\
&\gamma \sum_i \left(2\boldsymbol{K}^{(Y)} * (\boldsymbol{K}^{(X)}_{\cdot i} \boldsymbol{K}^{(X)}_{i\cdot})\boldsymbol{a} - 2\boldsymbol{K}^{(Y)} * (\boldsymbol{K}^{(W)}_{\cdot i} \boldsymbol{K}^{(X)}_{i\cdot})\hat{\boldsymbol{b}} \right) = 0
\end{aligned}
$$

$$
\begin{aligned}
&\left(\boldsymbol{K}^{(Y)} * (\boldsymbol{K}^{(X)} \boldsymbol{K}^{(X)}) + \lambda \boldsymbol{I} + \gamma \boldsymbol{K}^{(Y)} * (\boldsymbol{K}^{(X)} \boldsymbol{K}^{(X)}) \right) \boldsymbol{a} \\
&= \text{diag}(\boldsymbol{K}^{(Y)} \boldsymbol{K}^{(X)}) + \gamma \boldsymbol{K}^{(Y)} * (\boldsymbol{K}^{(W)} \boldsymbol{K}^{(X)})\hat{\boldsymbol{b}}
\end{aligned}
$$

It can be easily shown that the term in front of the vector $\boldsymbol{a}$ is non-singular, so that the solution for $\boldsymbol{a}$ can be found through inversion.

Bibliography

A. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. Inf. Syst*, 23(1):103–145, 2005.

F. Aiolli and A. Sperduti. Learning preferences for multiclass problems. In *Advances in Neural Information Processing Systems 17*, pages 17–24. MIT Press, 2004.

S. Alagumalai, D. Curtis, and N. Hungi. *Applied Rasch Measurement: A Book of Examples*. Springer, 2005. ISBN ISBN 978-1-4020-3076-5.

A. Anand. The philosophy of intransitive preferences. *The Economic Journal*, pages 337–346, 1993.

K.H. Arehart, J.M. Kates, C.A. Anderson, and L.O. Harvey Jr. Effects of noise and distortion on speech quality judgments in normal-hearing and hearing-impaired listeners. *Journal of the Acoustical Society of America*, 122(2):1150–1164, August 2007.

R. Arens. Learning SVM ranking function from user feedback using document metadata and active learning in the biomedical domain. In *Proceedings of the ECML/PKDD Workshop on Preference Learning*, 2008.

A. Argyriou, C.A. Micchelli, and M. Pontil. When is there a representer theorem? vector versus matrix regularizers. *Journal of Machine Learning Research*, 10:2507–2529, 2009.

B. Bakker and T. Heskes. Task clustering and gating for Bayesian multitask learning. *Journal of Machine Learning Research*, 4:83–99, 2003.

A.L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286 (5439):509–512, 1999.

A.L. Barabási, N. Gulbahce, and J. Loscalzo. Network medicine: a network-based approach to human disease. *Nature reviews. Genetics*, 12(1):56–68, 2011. doi: 10.1038/nrg2918.

D. Barber and C.M. Bishop. Ensemble learning in Bayesian neural networks. *Neural Networks and Machine Learning*, pages 215–237, 1998.

A. Ben-Hur and W.S. Noble. Kernel methods for predicting protein–protein interactions. *Bioinformatics*, 21(1):38–46, 2005. ISSN 1367-4803.

M.P.F. Berger. D-optimal sequential sampling designs for item response theory models. *Journal of Educational and Behavioral Statistics*, 19:43–56, 1994.

A. Birlutiu and T. Heskes. Expectation propagation for rating players in sports competitions. In *Knowledge Discovery in Databases: PKDD 2007, 11th European Conference on Principles and Practice of Knowledge Discovery in Databases*, volume 4702 of *Lecture Notes in Computer Science*, pages 374–381. Springer, 2007. ISBN 978-3-540-74975-2.

A. Birlutiu, P. Groot, and T. Heskes. Multi-task preference learning with an application to hearing aid personalization. In *Neurocomputing*, volume 73, pages 1177–1185, 2010. doi: 10.1016/j.neucom.2009.11.025.

C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. ISBN 978-0-387-31073-2.

K. Bleakley, G. Biau, and J.-P. Vert. Supervised reconstruction of biological networks with local models. *Bioinformatics*, 23(13):i57–i65, 2007.

D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent Dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003. ISSN 1532-4435. doi: 0.1162/jmlr.2003.3.4-5.993.

B.J.N. Blight and L. Ott. A Bayesian approach to model inadequacy for polynomial regression. *Biometrika*, 1:79–88, 1975.

J. Blythe. Visual exploration and incremental utility elicitation. In *Eighteenth national conference on Artificial intelligence*, pages 526–532, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence. ISBN 0-262-51129-0.

E. Bonilla, K.M. Chai, and C. Williams. Multi-task Gaussian process prediction. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, pages 153–160. MIT Press, Cambridge, MA, 2008.

R.F. Bordley. A multiplicative formula for aggregating probability assessments. *Management Science*, 28(10):1137–1148, 1982. ISSN 00251909.

C. Boutilier. A POMDP formulation of preference elicitation problems. In *International Joint Conference on Artificial Intelligence*, pages 239–246, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence. ISBN 0-262-51129-0.

C. Boutilier, R.S. Zemel, and B. Marlin. Active collaborative filtering. In *In Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence*, pages 98–106, 2003.

S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

R.A. Bradley and M.E. Terry. Rank analysis of incomplete block designs, I. The method of paired comparisons. *Biometrika*, 39:324–345, 1952.

Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 1950.

K. Brinker. Active learning of label ranking functions. In *The 27th International Conference on Machine Learning*, pages 129–136, 2004.

E. Brochu, N. de Freitas, and A. Ghosh. Active preference learning with discrete choice data. In J.C. Platt, Y. Koller, D. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 409–416. MIT Press, Cambridge, MA, 2008.

A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. *Comput. Netw.*, 33(1-6):309–320, 2000. ISSN 1389-1286. doi: http://dx.doi.org/10.1016/S1389-1286(00)00083-9.

C. Brouard, F. d'Alché Buc, and M. Szafranski. Semi-supervised protein-protein interaction network inference with regularized output kernel regression. In *sumitted*, 2011.

R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

R. Caruana, S. Baluja, and T. Mitchell. Using the future to sort out the present: Rankprop and multitask learning for medical risk evaluation. In *Advances in Neural Information Processing Systems 8*, pages 959–965, 1996.

U. Chajewska, D. Koller, and R. Parr. Making rational decisions using adaptive utility elicitation. In *In Proceedings of the Seventeenth National Conference on Artificial Intelligence*, pages 363–369, 2000.

K. Chaloner and I. Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10:273–304, 1995.

O. Chapelle and Z. Harchaoui. A machine learning approach to conjoint analysis. In Lawrence K.S., Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 257–264. MIT Press, Cambridge, MA, 2005.

X.W. Chen and M. Liu. Prediction of protein-protein interactions using random decision forest framework. *Bioinformatics*, 21(24):4394–4400, 2005.

R. Christensen. *Log-Linear Models and Logistic Regression.* Springer-Verlag, second edition, 1997.

W. Chu and Z. Ghahramani. Preference learning with Gaussian processes. In *Proceedings of the 22nd International Conference on Machine Learning*, volume 119 of *ACM International Conference Proceeding Series*, pages 137–144, Bonn, Germany, 2005a.

W. Chu and Z. Ghahramani. Extensions of Gaussian processes for ranking: semi-supervised and active learning. In *NIPS 2005 Workshop on Learning to Rank*, Whistler, BC, 2005b.

F. Chung and L. Lu. Connected components in random graphs with given expected degree sequences. *Annals of Combinatorics*, 6(2):125–145, 2002. doi: 10.1007/PL00012580.

M. Clyde, P. Müller, and G. Parmigiani. Optimal designs for heart defibrillators. volume 105 of *Lecture Notes in Statistics*, pages 278–292, New York, 1993. Springer.

D.A. Cohn, Z. Ghahramani, and M.I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.

C. Cortes, M. Mohri, and J. Weston. A general regression technique for learning transductions. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 153–160, New York, NY, USA, 2005. ACM. ISBN 1-59593-180-5.

C. Cortes, M. Mohri, and J. Weston. A general regression framework for learning string-to-string mappings. In *Predicting Structured Data.* MIT Press, 2007.

R.T. Cox. Probability, frequency and reasonable expectation. *American Journal of Physics*, 14:1–13, 1946.

K. Crammer and Y. Singer. Pranking with ranking. In *Advances in Neural Information Processing Systems 14*, pages 641–647. MIT Press, 2001.

I. Dagan and S. P. Engelson. Committee-based sampling for training probabilistic classifiers. In *In Proceedings of the Twelfth International Conference on Machine Learning*, pages 150–157. Morgan Kaufmann, 1995.

F. d'Alché Buc, A. Birlutiu, C. Brouard, T. Heskes, and M. Szafranski. Regularized output kernel regression for protein-protein interaction prediction: application to link transfer and transduction. *Machine Learning in Computational Biology workshop*, 2010.

S. Dasgupta and D. Hsu. Hierarchical sampling for active learning. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 208–215, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4. doi: 10.1145/1390156.1390183.

J. Doyle. Prospects for preferences. *Computational Intelligence*, 20(2):111–136, 2004.

H.A. Dror and D.M. Steinberg. Sequential experimental designs for generalized linear models. *Journal of the American Statistical Association*, (103):288–298, 2008.

D. Ekman, S. Light, A.K. Bjorklund, and A. Elofsson. What properties characterize the hub proteins of the protein-protein interaction network of Saccharomyces cerevisiae? *Genome Biology*, 7(R45), 2006. doi: 10.1186/gb-2006-7-6-r45.

T. Evgeniou, C.A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005. ISSN 1533-7928.

V.V. Fedorov. *Theory of Optimal Experiments.* Academic Press, 1972.

I. Ford and S.D. Silvey. A sequentially constructed design for estimating a nonlinear parametric function. *Biometrika*, 67:381–388, 1980.

Y. Freund, E. Shamir, and N. Tishby. Selective sampling using the Query by Committee algorithm. In *Machine Learning*, pages 133–168, 1997.

B.J. Frey, A. Kannan, and N. Jojic. Product analysis: Learning to model observations as products of hidden variables. In *Neural Information Processing Systems*, pages 729–735, 2001.

C. Friedel and R. Zimmer. Inferring topology from clustering coefficients in protein-protein interaction networks. *BMC Bioinformatics*, 7:519, 2006. doi: 10.1186/1471-2105-7-519.

J. Fürnkranz and E. Hüllermeier. Pairwise preference learning and ranking. In *Proceedings of the 14th European Conference on Machine Learning*, pages 145–156. Springer-Verlag, 2003.

J. Fürnkranz and E. Hüllermeier. Preference learning. *Künstliche Intelligenz*, 19(1): 60–61, 2005.

J. Fürnkranz and E. Hüllermeier. *Preference Learning.* Springer, 2010.

A. Gelman, J.B. Carlin, H.S. Stern, and D.B. Rubin. *Bayesian Data Analysis, Second Edition.* Chapman & Hall/CRC, July 2003. ISBN 158488388X.

S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992. ISSN 0899-7667. doi: 10.1162/neco.1992.4.1.1.

M.T. Gervasio, M.D. Moffitt, and M.E. Pollack. Active preference learning for personalized calendar scheduling assistance. In *10th International Conference on Intelligent User Interfaces*, pages 90–97. ACM Press, 2005.

T. van Gestel, J.A.K. Suykens, G. Lanckriet, A. Lambrechts, B. de Moor, and J. Vandewalle. Bayesian framework for least-squares support vector machine classifiers, Gaussian processes, and kernel Fisher discriminant analysis. *Neural Computation*, 14: 1115–1147, 2002.

P. Geurts, L. Wehenkel, and d'Alché Buc F. Kernelizing the output of tree-based methods. In *Proceedings of the 23th International Conference on Machine Learning*, pages 345–352, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2. doi: http://doi.acm.org/10.1145/1143844.1143888.

P. Geurts, N. Touleimat, M. Dutreix, and F. d'Alché-Buc. Inferring biological networks with output kernel trees. *BMC Bioinformatics (PMSB06 special issue)*, 8(Suppl 2):S4, 2007a.

P. Geurts, L. Wehenkel, and F. d'Alché-Buc. Gradient boosting for kernelized output spaces. In *ACM International Conference Proceeding Series (Proceedings of the 24th International Conference on Machine Learning)*, volume 227, pages 289–296. ACM, 2007b.

M. Glickman. *Paired Comparison Models with Time Varying Parameters*. PhD thesis, Harvard University, 1993.

M. Glickman and S. Jensen. Adaptive paired comparison design. *Journal of Statistical Planning and Inference*, 127:279–293, 2005.

J.P. Gosling. *Elicitation: A Nonparametric View*. PhD thesis, Department of Probability and Statistics, School of Mathematics and Statistics, 2005.

J.P. Gosling, J.E. Oakley, and A. O'Hagan. Nonparametric elicitation for heavy-tailed prior distributions. *Bayesian Analysis*, 2:693–718, 2007.

P.C. Groot, A. Birlutiu, and T. Heskes. Bayesian Monte Carlo for the global optimization of expensive functions. In *The 19th European Conference on Artificial Intelligence*, pages 249–254, 2010.

S. Guo and S. Sanner. Real-time multiattribute bayesian preference elicitation with pairwise comparison queries. In *Proceedings of the Thirteen International Conference on Artificial Intelligence and Statistics*, pages 289–296, 2010.

A.S. Harpale and Y. Yang. Personalized active learning for collaborative filtering. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 91–98, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-164-4.

R. Herbrich, T. Graepel, P. Bollmann-Sdorra, and K. Obermayer. Learning preference relations for information retrieval. In *Proceedings Workshop Text Categorization and*

Machine Learning, International Conference on Machine Learning, pages 80–84, 1998.

R. Herbrich, T. Minka, and T. Graepel. TrueSkill: A Bayesian skill rating system. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 569–576. MIT Press, Cambridge, MA, 2007.

T. Heskes and B. de Vries. Incremental utility elicitation for adaptive personalization. In K. Verbeeck, K. Tuyls, A. Nowé, B. Manderick, and B. Kuijpers, editors, *Proceedings of the Seventeenth Belgium-Netherlands Conference on Artificial Intelligence*, pages 127–134, Brussels, 2005.

M. Hollander and D.A. Wolfe. *Nonparametric Statistical Methods*. John Wiley & Sons, 1999.

T.K. Huang, C.J Lin, and R.C. Weng. A generalized Bradley-Terry model: From group competition to individual skill. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 601–608. MIT Press, Cambridge, MA, 2005.

R. Jansen, H. Yu, D. Greenbaum, Y. Kluger, N.J. Krogan, S. Chung, A. Emili, M. Snyder, J.F. Greenblatt, and M. Gerstein. A Bayesian networks approach for predicting protein-protein interactions from genomic data. *Science*, 302(5644):449–453, 2003.

H. Jeong, B. Tombor, R. Albert, Z.N. Oltval, and A.L. Barabasi. The large-scale organization of metabolic networks. *Nature*, 407:651–654, October 2000.

H. Jeong, S.P. Mason, A.-L. Barabási, and Z.N. Oltvai. Lethality and centrality in protein networks. *Nature*, 411(6833):41–42, 2001.

R. Jin and L. Si. A Bayesian approach toward active learning for collaborative filtering. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 278–285, Arlington, Virginia, United States, 2004. AUAI Press. ISBN 0-9749039-0-6.

B. Kanninen. Optimal design for multinomial choice experiments. *Journal of Marketing Research*, 39:307–317, 2002.

H. Kashima, Y. Yamanishi, T. Kato, M. Sugiyama, and K. Tsuda. Simultaneous inference of biological networks of multiple species from genome-wide data and evolutionary information. *Bioinformatics*, 25(22):2962–2968, 2009. ISSN 1367-4803.

T. Kato, K. Tsuda, and Kiyoshi A. Selective integration of multiple biological data for supervised network inference. *International Journal of Knowledge Discovery in Bioinformatics*, 2010.

G.S. Kimmeldorf and G. Wahba. A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics*, 41:495–502, 1970.

R.I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete structures. In *Proceedings of the 19th International Conference on Machine Learning*, pages 315–322, 2002.

J. Kropko and G. Rabinowitz. Choosing between multinomial logit and multinomial probit models for analysis of unordered choice data. Paper presented at the annual meeting of the MPSA Annual National Conference, Palmer House Hotel, Hilton, Chicago, 2008.

O. Kuchaiev, M. Rasajski, D.J. Higham, and N. Przulj. Geometric de-noising of protein-protein interaction networks. *PLOS Computational Biology*, 5(8), 2009.

S. Kullback and R.A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

P.R. Kumar, Y. Yu, R. Sternglanz, S.A. Johnston, and L. Joshua-Tor. NADP regulates the yeast gal induction system. *Science*, 5866(319):1090–1092, 2008. doi: 10.1126/science.1151903.

S.A. Lee, C.C. Chan, C.H. Tsai, J.M. Lai, F.S. Wang, C.Y. Kao, and C.Y. Huang. Ortholog-based protein-protein interaction prediction and its application to inter-species interactions. *BMC Bioinformatics*, 9(Suppl 12):S11, 2008.

B. Lehner and AG. Fraser. A first-draft human protein-interaction map. *Genome Biol.*, 5(9):R63, 2004.

J. Lewi, R. Butera, and L. Paninski. Efficient active learning with generalized linear models. *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, 2007.

J. Lewi, R. Butera, and L. Paninski. Sequential optimal design of neurophysiology experiments. *Neural Computation*, 21(3):619–687, 2009. ISSN 0899-7667.

D. Lewis and W. Gale. A sequential algorithm for training text classifiers. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12, New York, NY, USA, 1994. Springer-Verlag New York, Inc. ISBN 0-387-19889-X.

N. Lin, B. Wu, R. Jansen, M. Gerstein, and H. Zhao. Information assessment on predicting protein-protein interactions. *BMC Bioinformatics*, 5(154), 2004.

D.J.C. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4:590–604, 1992.

D.J.C. Mackay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, New York, NY, USA, 2002. ISBN 0521642981.

B. Marlin. Modeling user rating profiles for collaborative filtering. In *Proceedings of the 17th Annual Conference on Neural Information Processing Systems (NIPS03)*. MIT Press, 2003.

S. Maslov and K. Sneppen. Specificity and stability in topology of protein networks. *Science*, 296:910–913, 2002. doi: 10.1126/science.1065103.

A. McCallum and K. Nigam. Employing EM and pool-based active learning for text classification. In *International Conference on Machine Learning*, pages 350–358, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 1-55860-556-8.

P. Melville and R. Mooney. Diverse ensembles for active learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 74, New York, NY, USA, 2004. ACM. ISBN 1-58113-828-5. doi: 10.1145/1015330.1015385.

V. Memisevic, T. Milenkovic, and N. Przulj. Complementarity of network and sequence information in homologous proteins. *Journal of Integrative Bioinformatics*, 7(3):135, 2010.

C.A. Micchelli and M. Pontil. Kernels for multi–task learning. *Advances in Neural Information Processing Systems*, pages 921–928, 2005.

M. Michaut, S. Kerrien, L. Montecchi-Palazzi, F. Chauvat, C. Cassier-Chauvat, J.C. Aude, P. Legrain, and H. Hermjakob. Interoporc: automated inference of highly conserved protein interaction networks. *Bioinformatics*, 24(14):1625–1631, 2008.

T. Milenkovic and N. Przulj. Uncovering biological network function via graphlet degree signatures. *Cancer Informatics*, 6:257–273, 2008.

R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: Simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002. doi: 10.1126/science.298.5594.824.

T.P. Minka. *A Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, M.I.T., 2001.

T.P. Minka and J.D. Lafferty. Expectation-propogation for the generative aspect model. In *Proceedings of the 18th Conference in Uncertainty in Artificial Intelligence*, pages 352–359, 2002.

M. Mitzenmacher. A brief history of generative models for power law and lognormal distributions. *Internet Mathematics*, 1(2), 2003.

F.A. Moala and A. O'Hagan. Elicitation of Multivariate Prior Distributions: A nonparametric Bayesian approach. *Journal of Statistical Planning and Inference*, 2009.

M. Opper and O. Winther. Tractable approximations for probabilistic models: The adaptive Thouless-Anderson-Palmer mean field approach. *Physical Review Letters*, 2001.

T. Pahikkala, W. Waegeman, E. Tsivtsivadze, B. De Baets, and T. Salakoski. From ranking to intransitive preference learning: Rock-paper-scissors and beyond. In Eyke Hllermeier and Johannes Frnkranz, editors, *Proceedings of the ECML/PKDD-Workshop on Preference Learning (PL-09)*, pages 84–100, 2009.

D. Pennock, G.W. Flake, S. Lawrence, E.J. Glover, and C.L. Giles. Winners don't take all: Characterizing the competition for links on the web. In *Proceedings of the National Academy of Sciences*, pages 5207–5211, 2002.

M. Persico, A. Ceol, C. Gavrila, R. Hoffmann, A. Florio, and G. Cesareni. Homomint: an inferred human network based on orthology mapping of protein interactions discovered in model organisms. *BMC Bioinformatics*, 2005.

J. Platt, C. Burges, S. Swenson, C. Weare, and A. Zheng. Learning a Gaussian process prior for automatically generating music playlists. In *In Advances in Neural Information Processing Systems*, pages 1425–1432. MIT Press, 2002.

W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1992.

N. Przulj, D.G. Corneil, and I. Jurisica. Modeling interactome: scale-free or geometric?. *Bioinformatics*, 20(18):3508–3515, 2004.

Y. Qi, J. Klein-Seetharaman, and Z. Bar-Joseph. Random forest similarity for protein-protein interaction prediction from multiple sources. In Russ B. Altman, Tiffany A. Jung, Teri E. Klein, A. Keith Dunker, and Lawrence Hunter, editors, *Pacific Symposium on Biocomputing*. World Scientific, 2005.

Y. Qi, J. Klein-Seetharaman, and Z. Bar-Joseph. A mixture of feature experts approach for protein-protein interaction prediction. *BMC Bioinformatics*, 8(Suppl 10):S6, 2007.

Y. Qi, O. Tastan, J.G. Carbonell, J. Klein-Seetharaman, and J. Weston. Semi-supervised multi-task learning for predicting interactions between HIV-1 and human proteins. *Bioinformatics*, 26(18):i645–i652, 2010.

T. Qin, T.-Y. Liu, J. Xu, and H. Li. Letor: A benchmark collection for research on learning to rank for information retrieval. *Inf. Retr.*, 13(4):346–374, 2010.

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

S. Redner. How popular is your paper? an empirical study of the citation distribution. *European Physical Journal B*, 4:131–134, 1998.

G. Robins, P. Pattison, Y. Kalish, and D. Lusher. An introduction to exponential random graph models for social networks. *Social Networks*, 29(2):173–191, 2007.

S.L. Salzberg. On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery*, 1(3):317–328, 1997.

A.N. Sanborn and T.L. Griffiths. Markov chain Monte Carlo with people. *Neural Information Processing Systems*, 2008.

A. Schein and L. Ungar. Active learning for logistic regression: an evaluation. *Machine Learning*, 68(3):235–265, 2007. ISSN 0885-6125. doi: 10.1007/s10994-007-5019-5.

B. Schölkopf, R. Herbrich, and A.J. Smola. A generalized representer theorem. In *Proceedings of the 14th Annual Conference on Computational Learning Theory and 5th European Conference on Computational Learning Theory*, pages 416–426, London, UK, 2001. Springer-Verlag. ISBN 3-540-42343-5.

A. Schwaighofer, V. Tresp, and K. Yu. Learning Gaussian process kernels via hierarchical bayes. In L.K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1209–1216, Cambridge, MA, 2005. MIT Press.

M. Seeger. Notes on minka's expectation propagation for gaussian process classification. Technical report, University of Edinburgh, 2002.

M.W. Seeger. Bayesian inference and optimal design for the sparse linear model. *Journal of Machine Learning Research*, 9:759–813, 2008. ISSN 1533-7928.

B. Settles and M. Craven. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1070–1079, Morristown, NJ, USA, 2008. Association for Computational Linguistics.

H.S. Seung, M. Opper, and H. Sompolinsky. Query by Committee. In *COLT*, pages 287–294, New York, NY, USA, 1992. ACM. ISBN 0-89791-497-X.

E. Sprinzak, Y. Altuvia, and H. Margalit. Characterization and prediction of protein-protein interactions within and between complexes. *PNAS*, 103(40):1471814723, 2006.

S. Szedmak, J. Shawe-Taylor, and E. Parado-Hernandez. Learning via linear operators: Maximum margin regression. Technical report, University of Southampton, UK, 2005.

R. Tanaka, T.M. Yi, and J. Doyle. Some protein interaction data do not exhibit power law statistics. *FEBS Letters*, 579:5140–5144, 2005.

O. Tastan, Y. Qi, J.G. Carbonell, and J. Klein-Seetharaman. Prediction of interactions between HIV-1 and human proteins by information integration. *Proceedings of the Pacific Symposium on Biocomputing*, 14:516–527, 2009.

S. Thrun. Is learning the n-th thing any easier than learning the first? In *NIPS*, pages 640–646, 1995.

M. Tipping. Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.

A. Tversky. *Preference, Belief, and Similarity*. MIT Press, 1998.

J.-P. Vert. Reconstruction of biological networks by supervised machine learning approaches. *arXiv:0806.0215v2*, 2008.

C. von Mering, R. Krause, B. Snel, M. Cornell, S.G. Oliver, S. Fields, and P. Bork. Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, 417(6887):399–403, 2002.

J. Weston, O. Chapelle, A. Elisseeff, B. Schlkopf, and V. Vapnik. Kernel dependency estimation. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing*, volume 15, pages 873–880, Cambridge, MA, USA, 2003. MIT Press.

A.M. Wiles, M. Doderer, J. Ruan, T.T. Gu, D. Ravi, B. Blackman, and A. Bishop. Building and analyzing protein interactome networks by cross-species comparisons. *BMC Systems Biology*, 2010.

Z. Xu, K. Kersting, and T. Joachims. Fast active exploration for link-based preference learning using Gaussian processes. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 499–514, 2010.

Y. Xue, X. Liao, L. Carin, and B. Krishnapuram. Multi-task learning for classification with Dirichlet process priors. *Journal of Machine Learning Research*, 8:35–63, 2007. ISSN 1533-7928.

Y. Yamanishi, J.-P. Vert, and M. Kanehisa. Protein network inference from multiple genomic data: a supervised approach. *Bioinformatics*, 20(1):363–370, 2004. ISSN 1367-4803. doi: http://dx.doi.org/10.1093/bioinformatics/bth910.

K. Yu, A. Schwaighofer, V. Tresp, W.Y. Ma, and H.J. Zhang. Collaborative ensemble learning: Combining collaborative and content-based information filtering. In *In Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence*, pages 616–623, 2003.

K. Yu, V. Tresp, and A. Schwaighofer. Learning Gaussian processes from multiple tasks. In *Proceedings of the 22nd International Conference on Machine Learning*, 2005.

K. Yu, J. Bi, and V. Tresp. Active learning via transductive experimental design. In *International Conference on Machine Learning*, pages 1081–1088, New York, 2006. ISBN 1-59593-383-2.

L.V. Zhang, S.L. Wong, O.D. King, and F.P. Roth. Predicting co-complexed protein pairs using genomic and proteomic data integration. *BMC Bioinformatics*, 5:38, 2004.

Samenvatting

Machinaal leren is het onderdeel binnen de kunstmatige intelligentie waarin het gaat om het ontwerpen en ontwikkelen van methoden die machines in staat stellen te leren. Op basis van observaties en ervaring kunnen machines leren accurate voorspellingen te doen en zinvolle beslissingen te nemen. De meeste machinaal-leren taken zijn gesuperviseerd, wat wil zeggen dat het leerproces bestaat uit het afleiden van een functie op basis van trainingsdata. De trainingsdata bestaat uit een verzameling van voorbeelden, waarbij ieder voorbeeld is opgebouwd uit een invoer object en een gewenste uitkomst. Bij spamfilters wordt een geautomatiseerde classificatie geleerd met behulp van oude emails waaraan door een menselijke gebruiker een label "spam" of "geen spam" is toegekend. In geautomatiseerde classificatie van mammogrammen, wordt het model geleerd op basis van medische foto's waaraan door radiologen labels als goedaardig en kwaadaardig zijn toegevoegd. Menselijke spraak kan door machines herkend worden door te leren van spraakfragmenten die van tevoren zijn geannoteerd in woorden en zinnen. In al deze gevallen en in vele andere, is het erg lastig trainingsdata te verkrijgen voor het trainen van de algoritmen. Dit is dan ook een belangrijke uitdaging binnen het vakgebied van machinaal leren. Transfer/multi-task leren en actief leren, zijn vakgebieden binnen machinaal leren die werken aan oplossingen voor deze uitdagingen. De gedachte achter multi-taskleren is prestatieverbetering van de doeltaak door het inzetten van gelabelde data van gelijksoortige taken. Multi-task-leren is dus toepasbaar als er voor een specifiek scenario erg weinig data beschikbaar is, en er wel data beschikbaar is voor soortgelijke scenarios. Actief leren kan worden toegepast wanneer het algoritme interactief kan vragen om de labels van tot dan toe ongelabelde voorbeelden. De motivatie achter actief leren is dat een algoritme in staat is met veel minder trainingsdata een goede classifier te leren als deze actief om bepaalde informatieve voorbeelden vraagt, in tegenstelling tot willekeurige voorbeelden. Het werk in dit proefschrift is een verkenning van transfer/multi-task-leren en actief leren voor twee specifieke richtingen: preferentieleren, het leren van voorkeuren van gebruikers, en gesuperviseerde netwerkinferentie.

Dit proefschrift begint in hoofdstuk 2 met een evaluatie van methoden geschikt voor data die is opgebouwd uit paren. Dit type data wordt in de hierop volgende hoofdstukken gebruikt voor het leren van voorkeuren en voor het voorspellen van proteïne-proteïne interactie (PPIs). De analyse van de dataparen wordt gedaan in een Bayesiaans raamwerk waarin exacte inferentie onhaalbaar is. In de literatuur worden verscheidene technieken voorgesteld voor het benaderen van deze inferentie. De meest populaire is expectation propagation (EP), het verwachtingswaarde-aanpassings algoritme. We stellen aanpassingen voor omdat enerzijds EP computationeel erg duur kan zijn en anderzijds het geschikt gemaakt moet worden voor dataparen. De vraag die we hiermee in dit hoofdstuk willen beantwoorden is: hoe goed presteren de verschillende varianten van EP voor analyse van dataparen in een Bayesiaanse context? Deze varianten worden geëvalueerd door het bepalen van de sterkte van spelers in sportcompetities, in dit geval tenniscompetities.

In de hoofdstukken 3 en 4 worden machinaal leren toepassingen voor preferentieleren onderzocht. Preferentieleren houdt zich bezig met het bestuderen van methoden voor het voorspellen en modelleren van de voorkeuren van een gebruiker. Het gedeelte over preferentieleren volgt twee richtingen. De eerste richting neemt bij het leren van de voorkeur van een nieuwe gebruiker de beschikbare informatie van eerdere gebruikers mee. De tweede richting richt zich op het kiezen van experimenten die aan een gebruiker worden aangeboden om diens voorkeuren te leren. Hoofdstuk 3 onderzoekt multi-task-leren voor het leren van voorkeuren met Gaussische processen. We gebruiken het multi-task-formalisme om individuele trainingsdata van een gebruiker te verbeteren door gebruik te maken van eerder geleerde voorkeursinformatie van andere gebruikers. De bijdrage in dit hoofdstuk is de combinatie van multi-task-leren met een andersoortig leeralgoritme, namelijk Gaussische processen (GPs), voor het leren van voorkeuren. GPs worden gecombineerd met het multi-task-formalisme met behulp van een semi-parametrisch model. Door gebruik te maken van een semi-parametrische representatie kan multi-task-leren eenvoudig worden gemplementeerd door de theorie van hiërarchisch modelleren van parametrische modellen te gebruiken terwijl de voordelen van GPs worden behouden. We laten hiervan het nut zien door het model toe te passen op audiologische data. Het verzamelen van voorkeuren een lastig proces is, is het belangrijk dit proces zo efficiënt mogelijk te maken om zo de kosten en benodigde tijd terug te dringen. In hoofdstuk 4 wordt een raamwerk voorgesteld voor het optimaliseren van preferentieleren. Dit raamwerk richt zich op de combinatie actief leren en multi-task-leren. Actief leren is tot nu toe nog nauwelijks onderzocht in een multi-task-formalisme. In dit hoofdstuk bieden we een alternatief voor de standaardcriteria bij actief leren, door actief experimenten te selecteren door gebruik te maken van voorkeursdata van andere gebruikers. De voordelen van dit alternatief criterium zijn zowel de gereduceerde computationele kosten als de verkorte tijd die van de gebruiker wordt gevraagd. De bijdrage van dit hoofdstuk is een criterium voor actief leren dat is ontwikkeld voor de multi-task-setting; we laten zowel in theorie als in de praktijk zien dat dit nieuwe criterium op een soortgelijke manier presteert als de standaard criteria bij het ontwikkelen van "optimal experimental design", de tak van soort die zich bezig houdt met het ontwerpen van een optimaal experiment. Het voordeel van dit criterium is de interpretatie en het computationele gemak. We valideren onze aanpak empirisch door deze toe te passen op drie datasets met daarin praktijkdata met voorkeuren van gebruikers.

De volgende hoofdstukken, hoofdstuk 5 en hoofdstuk 6 onderzoeken toepassingen in machinaal leren voor gesuperviseerde netwerkinferentie in PPI netwerken. Het gedeelte dat zich richt op gesuperviseerd leren is toegespitst op twee themas. Ten eerste wordt een aanpak onderzocht voor het combineren van informatie uit zowel de topologische structuur van biologische netwerken als informatie over elk proteïne om een accuraat model van de PPI voorspelling te verkrijgen. Ten tweede worden methoden voor gesuperviseerde netwerkinferentie in een multi-task configuratie, het omzetten van kennis over PPIs van referentie soorten naar de doelsoort, gebruikmakend van orthologische informatie, onderzocht. Hoofdstuk 5 presenteert een op Bayesiaanse inferentie gebaseerde methode, voor het combineren van netwerktopologie informatie en observaties over proteïne paren als zijnde interacterend of niet. Het doel van deze combinatie is het verbeteren van de voorspelling voor PPIs. We definiëren een model voor het genereren van willekeurige grafen die qua topologie lijken op de topologie van PPI netwerken. In dit model incorporeren we de daadwerkelijke informatie van een netwerk door het willekeurige-graafmodel als een prior te behandelen en een waarschijnlijkheidsmodel voor eigenschappen van proteïnes te definiëren gegeven de aanwezigheid of afwezigheid van bepaalde interacties. Door dit te combineren en tegelijkertijd gebruik te maken van de regel van Bayes, komen we tot een model dat zowel topologische informatie als informatie over eigenschappen van proteïnes omvat. We laten met behulp van experimenten zien dat het resulterende model de nauwkeurigheid van de voorspellingen verbetert in PPI netwerken van zowel gist als mensen. Hoofdstuk 6 onderzoekt zogenaamde "link transfer" in het raamwerk van uitkomst-kernel-regressie, toegepast op PPI voorspellingen. Onderzoekers in de bioinformatica hebben strategieën gedefinieerd die bestaan uit het afbeelden van bekende interacties tussen referentieorganismen naar een doelorganisme voor de orthologische genen; dit wordt de proteïne-proteïne interologie aanpak genoemd. Voorspellen met behulp van interologieën is gebaseerd op de theorie dat proteïnen die in organismen interacteren zich op een dusdanige manier co-ontwikkelen, dat hun respectievelijke interologieën in staat blijven te interacteren in andere organismen. De contributie van dit hoofdstuk is een raamwerk voor het voorspellen van dit soort verbindingen, wat we "link transfer" zullen noemen. Link transfer is gebaseerd op uitvoer-kernel-regressie, maar dan door dit twee maal toe te passen. De eerste keer om de uitvoereigenschapsvectoren van de referentiesoort naar de doelsoort te converteren en de tweede keer om het doelnetwerk te leren. Het achterliggende idee van deze omzetting is het vermeerderen van de verzameling trainingsdata van de doelsoort door het converteren van de uitkomstruimte van de referentiesoort naar de uitkomstruimte van de doelsoort.

Acknowledgements

I would like to thank everyone who one way or another contributed to this thesis. First of all, I would like to thank my supervisor Tom Heskes. His guidance and support were really valuable and were there whenever were needed. I am thankful to the people with whom I collaborated, Bert, Perry, Rolph, Tjeerd and all the other members of the HearClip project. I want to thank my former group-mates and in particular to Ali, Arjen, Botond, Daan, Daniel, Evgeni, Fabio, Henriette, Ildiko, Janos, Max, Marcel, Pavol, Rasa, Saiden, Tom C., for making a pleasant working atmosphere and for the enjoyable times we spent outside of work, I am glad that many of you have become my friends. I am grateful to Florence d'Alché and the members of her group for the time spent during my internship in France. I feel lucky to have met while working for this thesis many nice people that became dearest friends, Alice, Alina, Amalia, Andreea, Angelica, Delia, Georgeta, Gleb, Irene, Iziah, Lucian, Luminita, Marcela, Maria, Olha, Peer, Rosalien, and from more far away, Ionela, Ramona, Sorina, who were all very supportive in all these years. I am grateful to my god-parents for their guidance. I want to thank Jasper for his support and encouragement. Last, but not least, I owe a lot to my parents, Gavrilă and Silvia, without whose support nothing would have been possible. *Mulţumesc!*

September, 2011 Adriana

SIKS Dissertation Series

1998:

1998-1 Johan van den Akker (CWI)
DEGAS - An Active, Temporal Database of Autonomous Objects
1998-2 Floris Wiesman (UM)
Information Retrieval by Graphically Browsing Meta-Information
1998-3 Ans Steuten (TUD)
A Contribution to the Linguistic Analysis of Business Conversations
within the Language/Action Perspective
1998-4 Dennis Breuker (UM)
Memory versus Search in Games
1998-5 E.W.Oskamp (RUL)
Computerondersteuning bij Straftoemeting

1999:

1999-1 Mark Sloof (VU)
Physiology of Quality Change Modeling; Automated modeling of Quality Change
of Agricultural Products
1999-2 Rob Potharst (EUR)
Classification using decision trees and neural nets
1999-3 Don Beal (UM)
The Nature of Minimax Search
1999-4 Jacques Penders (UM)
The practical Art of Moving Physical Objects
1999-5 Aldo de Moor (KUB)
Empowering Communities: A Method for the Legitimate User-Drive Specification
of Network Information Systems
1999-6 Niek J.E. Wijngaards (VU)
Re-design of compositional systems
1999-7 David Spelt (UT)
Verification support for object database design
1999-8 Jacques H.J. Lenting (UM)
Informed Gambling: Conception and Analysis of a Multi-Agent Mechanism for Discrete Reallocation.

2000:

2000-1 Frank Niessink (VU)
Perspectives on Improving Software Maintenance
2000-2 Koen Holtman (TUE)
Prototyping of CMS Storage Management
2000-3 Carolien M.T. Metselaar (UVA)
Sociaal-organisatorische gevolgen van kennistechnologie; een procesbenadering en actorperspectief.
2000-4 Geert de Haan (VU)
ETAG, A Formal Model of Competence Knowledge for User Interface Design
2000-5 Ruud van der Pol (UM)
Knowledge-based Query Formulation in Information Retrieval.
2000-6 Rogier van Eijk (UU)
Programming Languages for Agent Communication
2000-7 Niels Peek (UU)
Decision-theoretic Planning of Clinical Patient Management
2000-8 Veerle Coup (EUR)
Sensitivity Analyis of Decision-Theoretic Networks
2000-9 Florian Waas (CWI)
Principles of Probabilistic Query Optimization
2000-10 Niels Nes (CWI)
Image Database Management System Design Considerations, Algorithms and Architecture
2000-11 Jonas Karlsson (CWI)
Scalable Distributed Data Structures for Database Management

2001:

2001-1 Silja Renooij (UU)
Qualitative Approaches to Quantifying Probabilistic Networks
2001-2 Koen Hindriks (UU)
Agent Programming Languages: Programming with Mental Models
2001-3 Maarten van Someren (UvA)
Learning as problem solving
2001-4 Evgueni Smirnov (UM)
Conjunctive and Disjunctive Version Spaces with Instance-Based Boundary Sets
2001-5 Jacco van Ossenbruggen (VU)
Processing Structured Hypermedia: A Matter of Style
2001-6 Martijn van Welie (VU)
Task-based User Interface Design
2001-7 Bastiaan Schonhage (VU)
Diva: Architectural Perspectives on Information Visualization
2001-8 Pascal van Eck (VU)
A Compositional Semantic Structure for Multi-Agent Systems Dynamics.
2001-9 Pieter Jan 't Hoen (RUL)
Towards Distributed Development of Large Object-Oriented Models, Views of Packages as Classes
2001-10 Maarten Sierhuis (UvA)
Modeling and Simulating Work Practice BRAHMS: a multiagent modeling
and simulation language for work practice analysis and design
2001-11 Tom M. van Engers (VUA)
Knowledge Management: The Role of Mental Models in Business Systems Design

2002:

2002-01 Nico Lassing (VU)
Architecture-Level Modifiability Analysis
2002-02 Roelof van Zwol (UT)
Modelling and searching web-based document collections
2002-03 Henk Ernst Blok (UT)
Database Optimization Aspects for Information Retrieval
2002-04 Juan Roberto Castelo Valdueza (UU)
The Discrete Acyclic Digraph Markov Model in Data Mining
2002-05 Radu Serban (VU)
The Private Cyberspace Modeling Electronic Environments inhabited by Privacy-concerned Agents
2002-06 Laurens Mommers (UL)
Applied legal epistemology; Building a knowledge-based ontology of the legal domain
2002-07 Peter Boncz (CWI)
Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications
2002-08 Jaap Gordijn (VU)
Value Based Requirements Engineering: Exploring Innovative E-Commerce Ideas
2002-09 Willem-Jan van den Heuvel(KUB)
Integrating Modern Business Applications with Objectified Legacy Systems
2002-10 Brian Sheppard (UM)
Towards Perfect Play of Scrabble
2002-11 Wouter C.A. Wijngaards (VU)
Agent Based Modelling of Dynamics: Biological and Organisational Applications
2002-12 Albrecht Schmidt (Uva)
Processing XML in Database Systems
2002-13 Hongjing Wu (TUE)
A Reference Architecture for Adaptive Hypermedia Applications
2002-14 Wieke de Vries (UU)
Agent Interaction: Abstract Approaches to Modelling, Programming and
Verifying Multi-Agent Systems
2002-15 Rik Eshuis (UT)
Semantics and Verification of UML Activity Diagrams for Workflow Modelling
2002-16 Pieter van Langen (VU)
The Anatomy of Design: Foundations, Models and Applications

2002-17 Stefan Manegold (UVA)
Understanding, Modeling, and Improving Main-Memory Database Performance

2003:

2003-01 Heiner Stuckenschmidt (VU)
Ontology-Based Information Sharing in Weakly Structured Environments
2003-02 Jan Broersen (VU)
Modal Action Logics for Reasoning About Reactive Systems
2003-03 Martijn Schuemie (TUD)
Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy
2003-04 Milan Petkovic (UT)
Content-Based Video Retrieval Supported by Database Technology
2003-05 Jos Lehmann (UVA)
Causation in Artificial Intelligence and Law - A modelling approach
2003-06 Boris van Schooten (UT)
Development and specification of virtual environments
2003-07 Machiel Jansen (UvA)
Formal Explorations of Knowledge Intensive Tasks
2003-08 Yongping Ran (UM)
Repair Based Scheduling
2003-09 Rens Kortmann (UM)
The resolution of visually guided behaviour
2003-10 Andreas Lincke (UvT)
Electronic Business Negotiation: Some experimental studies on the interaction
between medium, innovation context and culture
2003-11 Simon Keizer (UT)
Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks
2003-12 Roeland Ordelman (UT)
Dutch speech recognition in multimedia information retrieval
2003-13 Jeroen Donkers (UM)
Nosce Hostem - Searching with Opponent Models
2003-14 Stijn Hoppenbrouwers (KUN)
Freezing Language: Conceptualisation Processes across ICT-Supported Organisations
2003-15 Mathijs de Weerdt (TUD)
Plan Merging in Multi-Agent Systems
2003-16 Menzo Windhouwer (CWI)
Feature Grammar Systems - Incremental Maintenance of Indexes to Digital Media Warehouses
2003-17 David Jansen (UT)
Extensions of Statecharts with Probability, Time, and Stochastic Timing
2003-18 Levente Kocsis (UM)
Learning Search Decisions

2004:

2004-01 Virginia Dignum (UU)
A Model for Organizational Interaction: Based on Agents, Founded in Logic
2004-02 Lai Xu (UvT)
Monitoring Multi-party Contracts for E-business
2004-03 Perry Groot (VU)
A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving
2004-04 Chris van Aart (UVA)
Organizational Principles for Multi-Agent Architectures
2004-05 Viara Popova (EUR)
Knowledge discovery and monotonicity
2004-06 Bart-Jan Hommes (TUD)
The Evaluation of Business Process Modeling Techniques
2004-07 Elise Boltjes (UM)
Voorbeeldig onderwijs; voorbeeldgestuurd onderwijs, een opstap naar abstract denken,
vooral voor meisjes
2004-08 Joop Verbeek(UM)
Politie en de Nieuwe Internationale Informatiemarkt, Grensregionale

politiële gegevensuitwisseling en digitale expertise
2004-09 Martin Caminada (VU)
For the Sake of the Argument; explorations into argument-based reasoning
2004-10 Suzanne Kabel (UVA)
Knowledge-rich indexing of learning-objects
2004-11 Michel Klein (VU)
Change Management for Distributed Ontologies
2004-12 The Duy Bui (UT)
Creating emotions and facial expressions for embodied agents
2004-13 Wojciech Jamroga (UT)
Using Multiple Models of Reality: On Agents who Know how to Play
2004-14 Paul Harrenstein (UU)
Logic in Conflict. Logical Explorations in Strategic Equilibrium
2004-15 Arno Knobbe (UU)
Multi-Relational Data Mining
2004-16 Federico Divina (VU)
Hybrid Genetic Relational Search for Inductive Learning
2004-17 Mark Winands (UM)
Informed Search in Complex Games
2004-18 Vania Bessa Machado (UvA)
Supporting the Construction of Qualitative Knowledge Models
2004-19 Thijs Westerveld (UT)
Using generative probabilistic models for multimedia retrieval
2004-20 Madelon Evers (Nyenrode)
Learning from Design: facilitating multidisciplinary design teams

2005:

2005-01 Floor Verdenius (UVA)
Methodological Aspects of Designing Induction-Based Applications
2005-02 Erik van der Werf (UM))
AI techniques for the game of Go
2005-03 Franc Grootjen (RUN)
A Pragmatic Approach to the Conceptualisation of Language
2005-04 Nirvana Meratnia (UT)
Towards Database Support for Moving Object data
2005-05 Gabriel Infante-Lopez (UVA)
Two-Level Probabilistic Grammars for Natural Language Parsing
2005-06 Pieter Spronck (UM)
Adaptive Game AI
2005-07 Flavius Frasincar (TUE)
Hypermedia Presentation Generation for Semantic Web Information Systems
2005-08 Richard Vdovjak (TUE)
A Model-driven Approach for Building Distributed Ontology-based Web Applications
2005-09 Jeen Broekstra (VU)
Storage, Querying and Inferencing for Semantic Web Languages
2005-10 Anders Bouwer (UVA)
Explaining Behaviour: Using Qualitative Simulation in Interactive Learning Environments
2005-11 Elth Ogston (VU)
Agent Based Matchmaking and Clustering - A Decentralized Approach to Search
2005-12 Csaba Boer (EUR)
Distributed Simulation in Industry
2005-13 Fred Hamburg (UL)
Een Computermodel voor het Ondersteunen van Euthanasiebeslissingen
2005-14 Borys Omelayenko (VU)
Web-Service configuration on the Semantic Web; Exploring how semantics meets pragmatics
2005-15 Tibor Bosse (VU)
Analysis of the Dynamics of Cognitive Processes
2005-16 Joris Graaumans (UU)
Usability of XML Query Languages
2005-17 Boris Shishkov (TUD)
Software Specification Based on Re-usable Business Components

2005-18 Danielle Sent (UU)
Test-selection strategies for probabilistic networks
2005-19 Michel van Dartel (UM)
Situated Representation
2005-20 Cristina Coteanu (UL)
Cyber Consumer Law, State of the Art and Perspectives
2005-21 Wijnand Derks (UT)
Improving Concurrency and Recovery in Database Systems by Exploiting Application Semantics

2006:

2006-01 Samuil Angelov (TUE)
Foundations of B2B Electronic Contracting
2006-02 Cristina Chisalita (VU)
Contextual issues in the design and use of information technology in organizations
2006-03 Noor Christoph (UVA)
The role of metacognitive skills in learning to solve problems
2006-04 Marta Sabou (VU)
Building Web Service Ontologies
2006-05 Cees Pierik (UU)
Validation Techniques for Object-Oriented Proof Outlines
2006-06 Ziv Baida (VU)
Software-aided Service Bundling - Intelligent Methods & Tools for Graphical Service Modeling
2006-07 Marko Smiljanic (UT)
XML schema matching – balancing efficiency and effectiveness by means of clustering
2006-08 Eelco Herder (UT)
Forward, Back and Home Again - Analyzing User Behavior on the Web
2006-09 Mohamed Wahdan (UM)
Automatic Formulation of the Auditor's Opinion
2006-10 Ronny Siebes (VU)
Semantic Routing in Peer-to-Peer Systems
2006-11 Joeri van Ruth (UT)
Flattening Queries over Nested Data Types
2006-12 Bert Bongers (VU)
Interactivation - Towards an e-cology of people, our technological environment, and the arts
2006-13 Henk-Jan Lebbink (UU)
Dialogue and Decision Games for Information Exchanging Agents
2006-14 Johan Hoorn (VU)
Software Requirements: Update, Upgrade, Redesign - towards a Theory of Requirements Change
2006-15 Rainer Malik (UU)
CONAN: Text Mining in the Biomedical Domain
2006-16 Carsten Riggelsen (UU)
Approximation Methods for Efficient Learning of Bayesian Networks
2006-17 Stacey Nagata (UU)
User Assistance for Multitasking with Interruptions on a Mobile Device
2006-18 Valentin Zhizhkun (UVA)
Graph transformation for Natural Language Processing
2006-19 Birna van Riemsdijk (UU)
Cognitive Agent Programming: A Semantic Approach
2006-20 Marina Velikova (UvT)
Monotone models for prediction in data mining
2006-21 Bas van Gils (RUN)
Aptness on the Web
2006-22 Paul de Vrieze (RUN)
Fundaments of Adaptive Personalisation
2006-23 Ion Juvina (UU)
Development of Cognitive Model for Navigating on the Web
2006-24 Laura Hollink (VU)
Semantic Annotation for Retrieval of Visual Resources
2006-25 Madalina Drugan (UU)
Conditional log-likelihood MDL and Evolutionary MCMC
2006-26 Vojkan Mihajlovic (UT)

Score Region Algebra: A Flexible Framework for Structured Information Retrieval
2006-27 Stefano Bocconi (CWI)
Vox Populi: generating video documentaries from semantically annotated media repositories
2006-28 Borkur Sigurbjornsson (UVA)
Focused Information Access using XML Element Retrieval

2007:

2007-01 Kees Leune (UvT)
Access Control and Service-Oriented Architectures
2007-02 Wouter Teepe (RUG)
Reconciling Information Exchange and Confidentiality: A Formal Approach
2007-03 Peter Mika (VU)
Social Networks and the Semantic Web
2007-04 Jurriaan van Diggelen (UU)
Achieving Semantic Interoperability in Multi-agent Systems: a dialogue-based approach
2007-05 Bart Schermer (UL)
Software Agents, Surveillance, and the Right to Privacy:
a Legislative Framework for Agent-enabled Surveillance
2007-06 Gilad Mishne (UVA)
Applied Text Analytics for Blogs
2007-07 Natasa Jovanovic' (UT)
To Whom It May Concern - Addressee Identification in Face-to-Face Meetings
2007-08 Mark Hoogendoorn (VU)
Modeling of Change in Multi-Agent Organizations
2007-09 David Mobach (VU)
Agent-Based Mediated Service Negotiation
2007-10 Huib Aldewereld (UU)
Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols
2007-11 Natalia Stash (TUE)
Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System
2007-12 Marcel van Gerven (RUN)
Bayesian Networks for Clinical Decision Support:
A Rational Approach to Dynamic Decision-Making under Uncertainty
2007-13 Rutger Rienks (UT)
Meetings in Smart Environments; Implications of Progressing Technology
2007-14 Niek Bergboer (UM)
Context-Based Image Analysis
2007-15 Joyca Lacroix (UM)
NIM: a Situated Computational Memory Model
2007-16 Davide Grossi (UU)
Designing Invisible Handcuffs.
Formal investigations in Institutions and Organizations for Multi-agent Systems
2007-17 Theodore Charitos (UU)
Reasoning with Dynamic Networks in Practice
2007-18 Bart Orriens (UvT)
On the development an management of adaptive business collaborations
2007-19 David Levy (UM)
Intimate relationships with artificial partners
2007-20 Slinger Jansen (UU)
Customer Configuration Updating in a Software Supply Network
2007-21 Karianne Vermaas (UU)
Fast diffusion and broadening use: A research on residential adoption and usage of
broadband internet in the Netherlands between 2001 and 2005
2007-22 Zlatko Zlatev (UT)
Goal-oriented design of value and process models from patterns
2007-23 Peter Barna (TUE)
Specification of Application Logic in Web Information Systems
2007-24 Georgina Ramrez Camps (CWI)
Structural Features in XML Retrieval
2007-25 Joost Schalken (VU)
Empirical Investigations in Software Process Improvement

2008:

2008-01 Katalin Boer-Sorban (EUR)
Agent-Based Simulation of Financial Markets: A modular,continuous-time approach
2008-02 Alexei Sharpanskykh (VU)
On Computer-Aided Methods for Modeling and Analysis of Organizations
2008-03 Vera Hollink (UVA)
Optimizing hierarchical menus: a usage-based approach
2008-04 Ander de Keijzer (UT)
Management of Uncertain Data - towards unattended integration
2008-05 Bela Mutschler (UT)
Modeling and simulating causal dependencies on process-aware information systems
from a cost perspective
2008-06 Arjen Hommersom (RUN)
On the Application of Formal Methods to Clinical Guidelines,
an Artificial Intelligence Perspective
2008-07 Peter van Rosmalen (OU)
Supporting the tutor in the design and support of adaptive e-learning
2008-08 Janneke Bolt (UU)
Bayesian Networks: Aspects of Approximate Inference
2008-09 Christof van Nimwegen (UU)
The paradox of the guided user: assistance can be counter-effective
2008-10 Wauter Bosma (UT)
Discourse oriented summarization
2008-11 Vera Kartseva (VU)
Designing Controls for Network Organizations: A Value-Based Approach
2008-12 Jozsef Farkas (RUN)
A Semiotically Oriented Cognitive Model of Knowledge Representation
2008-13 Caterina Carraciolo (UVA)
Topic Driven Access to Scientific Handbooks
2008-14 Arthur van Bunningen (UT)
Context-Aware Querying; Better Answers with Less Effort
2008-15 Martijn van Otterlo (UT)
The Logic of Adaptive Behavior: Knowledge Representation and Algorithms for
the Markov Decision Process Framework in First-Order Domains.
2008-16 Henriette van Vugt (VU)
Embodied agents from a user's perspective
2008-17 Martin Op 't Land (TUD)
Applying Architecture and Ontology to the Splitting and Allying of Enterprises
2008-18 Guido de Croon (UM)
Adaptive Active Vision
2008-19 Henning Rode (UT)
From Document to Entity Retrieval: Improving Precision and Performance of Focused Text Search
2008-20 Rex Arendsen (UVA)
Geen bericht, goed bericht. Een onderzoek naar de effecten van de introductie van elektronisch
berichtenverkeer met de overheid op de administratieve lasten van bedrijven
2008-21 Krisztian Balog (UVA)
People Search in the Enterprise
2008-22 Henk Koning (UU)
Communication of IT-Architecture
2008-23 Stefan Visscher (UU)
Bayesian network models for the management of ventilator-associated pneumonia
2008-24 Zharko Aleksovski (VU)
Using background knowledge in ontology matching
2008-25 Geert Jonker (UU)
Efficient and Equitable Exchange in Air Traffic Management Plan Repair
using Spender-signed Currency
2008-26 Marijn Huijbregts (UT)
Segmentation, Diarization and Speech Transcription: Surprise Data Unraveled
2008-27 Hubert Vogten (OU)
Design and Implementation Strategies for IMS Learning Design

2008-28 Ildiko Flesch (RUN)
On the Use of Independence Relations in Bayesian Networks
2008-29 Dennis Reidsma (UT)
Annotations and Subjective Machines - Of Annotators, Embodied Agents, Users,
and Other Humans
2008-30 Wouter van Atteveldt (VU)
Semantic Network Analysis: Techniques for Extracting, Representing and Querying Media Content
2008-31 Loes Braun (UM)
Pro-Active Medical Information Retrieval
2008-32 Trung H. Bui (UT)
Toward Affective Dialogue Management using Partially Observable Markov Decision Processes
2008-33 Frank Terpstra (UVA)
Scientific Workflow Design; theoretical and practical issues
2008-34 Jeroen de Knijf (UU)
Studies in Frequent Tree Mining
2008-35 Ben Torben Nielsen (UvT)
Dendritic morphologies: function shapes structure

2009:

2009-01 Rasa Jurgelenaite (RUN)
Symmetric Causal Independence Models
2009-02 Willem Robert van Hage (VU)
Evaluating Ontology-Alignment Techniques
2009-03 Hans Stol (UvT)
A Framework for Evidence-based Policy Making Using IT
2009-04 Josephine Nabukenya (RUN)
Improving the Quality of Organisational Policy Making using Collaboration Engineering
2009-05 Sietse Overbeek (RUN)
Bridging Supply and Demand for Knowledge Intensive Tasks -
Based on Knowledge, Cognition, and Quality
2009-06 Muhammad Subianto (UU)
Understanding Classification
2009-07 Ronald Poppe (UT)
Discriminative Vision-Based Recovery and Recognition of Human Motion
2009-08 Volker Nannen (VU)
Evolutionary Agent-Based Policy Analysis in Dynamic Environments
2009-09 Benjamin Kanagwa (RUN)
Design, Discovery and Construction of Service-oriented Systems
2009-10 Jan Wielemaker (UVA)
Logic programming for knowledge-intensive interactive applications
2009-11 Alexander Boer (UVA)
Legal Theory, Sources of Law & the Semantic Web
2009-12 Peter Massuthe (TUE, Humboldt-Universitaet zu Berlin)
Operating Guidelines for Services
2009-13 Steven de Jong (UM)
Fairness in Multi-Agent Systems
2009-14 Maksym Korotkiy (VU)
From ontology-enabled services to service-enabled ontologies
(making ontologies work in e-science with ONTO-SOA)
2009-15 Rinke Hoekstra (UVA)
Ontology Representation - Design Patterns and Ontologies that Make Sense
2009-16 Fritz Reul (UvT)
New Architectures in Computer Chess
2009-17 Laurens van der Maaten (UvT)
Feature Extraction from Visual Data
2009-18 Fabian Groffen (CWI)
Armada, An Evolving Database System
2009-19 Valentin Robu (CWI)
Modeling Preferences, Strategic Reasoning and Collaboration in
Agent-Mediated Electronic Markets
2009-20 Bob van der Vecht (UU)

Adjustable Autonomy: Controling Influences on Decision Making
2009-21 Stijn Vanderlooy(UM)
Ranking and Reliable Classification
2009-22 Pavel Serdyukov (UT)
Search For Expertise: Going beyond direct evidence
2009-23 Peter Hofgesang (VU)
Modelling Web Usage in a Changing Environment
2009-24 Annerieke Heuvelink (VU)
Cognitive Models for Training Simulations
2009-25 Alex van Ballegooij (CWI)
RAM: Array Database Management through Relational Mapping
2009-26 Fernando Koch (UU)
An Agent-Based Model for the Development of Intelligent Mobile Services
2009-27 Christian Glahn (OU)
Contextual Support of social Engagement and Reflection on the Web
2009-28 Sander Evers (UT)
Sensor Data Management with Probabilistic Models
2009-29 Stanislav Pokraev (UT)
Model-Driven Semantic Integration of Service-Oriented Applications
2009-30 Marcin Zukowski (CWI)
Balancing vectorized query execution with bandwidth-optimized storage
2009-31 Sofiya Katrenko (UVA)
A Closer Look at Learning Relations from Text
2009-32 Rik Farenhorst (VU) and Remco de Boer (VU)
Architectural Knowledge Management: Supporting Architects and Auditors
2009-33 Khiet Truong (UT)
How Does Real Affect Affect Affect Recognition In Speech?
2009-34 Inge van de Weerd (UU)
Advancing in Software Product Management: An Incremental Method Engineering Approach
2009-35 Wouter Koelewijn (UL)
Privacy en Politiegegevens; Over geautomatiseerde normatieve informatie-uitwisseling
2009-36 Marco Kalz (OU)
Placement Support for Learners in Learning Networks
2009-37 Hendrik Drachsler (OU)
Navigation Support for Learners in Informal Learning Networks
2009-38 Riina Vuorikari (OU)
Tags and self-organisation: a metadata ecology for learning resources in a multilingual context
2009-39 Christian Stahl (TUE, Humboldt-Universitaet zu Berlin)
Service Substitution – A Behavioral Approach Based on Petri Nets
2009-40 Stephan Raaijmakers (UvT)
Multinomial Language Learning: Investigations into the Geometry of Language
2009-41 Igor Berezhnyy (UvT)
Digital Analysis of Paintings
2009-42 Toine Bogers (UvT)
Recommender Systems for Social Bookmarking
2009-43 Virginia Nunes Leal Franqueira (UT)
Finding Multi-step Attacks in Computer Networks using Heuristic Search and Mobile Ambients
2009-44 Roberto Santana Tapia (UT)
Assessing Business-IT Alignment in Networked Organizations
2009-45 Jilles Vreeken (UU)
Making Pattern Mining Useful
2009-46 Loredana Afanasiev (UvA)
Querying XML: Benchmarks and Recursion

2010:

2010-01 Matthijs van Leeuwen (UU)
Patterns that Matter
2010-02 Ingo Wassink (UT)
Work flows in Life Science
2010-03 Joost Geurts (CWI)
A Document Engineering Model and Processing Framework for Multimedia documents

2010-04 Olga Kulyk (UT)
Do You Know What I Know? Situational Awareness of Co-located Teams
in Multi-display Environments
2010-05 Claudia Hauff (UT)
Predicting the Effectiveness of Queries and Retrieval Systems
2010-06 Sander Bakkes (UvT)
Rapid Adaptation of Video Game AI
2010-07 Wim Fikkert (UT)
Gesture interaction at a Distance
2010-08 Krzysztof Siewicz (UL)
Towards an Improved Regulatory Framework of Free Software.
Protecting user freedoms in a world of software communities and eGovernments
2010-09 Hugo Kielman (UL)
Politile gegevensverwerking en Privacy, Naar een effectieve waarborging
2010-10 Rebecca Ong (UL)
Mobile Communication and Protection of Children
2010-11 Adriaan Ter Mors (TUD)
The world according to MARP: Multi-Agent Route Planning
2010-12 Susan van den Braak (UU)
Sensemaking software for crime analysis
2010-13 Gianluigi Folino (RUN)
High Performance Data Mining using Bio-inspired techniques
2010-14 Sander van Splunter (VU)
Automated Web Service Reconfiguration
2010-15 Lianne Bodenstaff (UT)
Managing Dependency Relations in Inter-Organizational Models
2010-16 Sicco Verwer (TUD)
Efficient Identification of Timed Automata, theory and practice
2010-17 Spyros Kotoulas (VU)
Scalable Discovery of Networked Resources: Algorithms, Infrastructure, Applications
2010-18 Charlotte Gerritsen (VU)
Caught in the Act: Investigating Crime by Agent-Based Simulation
2010-19 Henriette Cramer (UvA)
People's Responses to Autonomous and Adaptive Systems
2010-20 Ivo Swartjes (UT)
Whose Story Is It Anyway? How Improv Informs Agency and Authorship of Emergent Narrative
2010-21 Harold van Heerde (UT)
Privacy-aware data management by means of data degradation
2010-22 Michiel Hildebrand (CWI)
End-user Support for Access to Heterogeneous Linked Data
2010-23 Bas Steunebrink (UU)
The Logical Structure of Emotions
2010-24 Dmytro Tykhonov
Designing Generic and Efficient Negotiation Strategies
2010-25 Zulfiqar Ali Memon (VU)
Modelling Human-Awareness for Ambient Agents: A Human Mind-reading Perspective
2010-26 Ying Zhang (CWI)
XRPC: Efficient Distributed Query Processing on Heterogeneous XQuery Engines
2010-27 Marten Voulon (UL)
Automatisch contracteren
2010-28 Arne Koopman (UU)
Characteristic Relational Patterns
2010-29 Stratos Idreos (CWI)
Database Cracking: Towards Auto-tuning Database Kernels
2010-30 Marieke van Erp (UvT)
Accessing Natural History - Discoveries in data cleaning, structuring, and retrieval
2010-31 Victor de Boer (UVA)
Ontology Enrichment from Heterogeneous Sources on the Web
2010-32 Marcel Hiel (UvT)
An Adaptive Service Oriented Architecture: Automatically solving Interoperability Problems
2010-33 Robin Aly (UT)
Modeling Representation Uncertainty in Concept-Based Multimedia Retrieval

2010-34 Teduh Dirgahayu (UT)
Interaction Design in Service Compositions
2010-35 Dolf Trieschnigg (UT)
Proof of Concept: Concept-based Biomedical Information Retrieval
2010-36 Jose Janssen (OU)
Paving the Way for Lifelong Learning;
Facilitating competence development through a learning path specification
2010-37 Niels Lohmann (TUE)
Correctness of services and their composition
2010-38 Dirk Fahland (TUE)
From Scenarios to components
2010-39 Ghazanfar Farooq Siddiqui (VU)
Integrative modeling of emotions in virtual agents
2010-40 Mark van Assem (VU)
Converting and Integrating Vocabularies for the Semantic Web
2010-41 Guillaume Chaslot (UM)
Monte-Carlo Tree Search
2010-42 Sybren de Kinderen (VU)
Needs-driven service bundling in a multi-supplier setting
- the computational e3-service approach
2010-43 Peter van Kranenburg (UU)
A Computational Approach to Content-Based Retrieval of Folk Song Melodies
2010-44 Pieter Bellekens (TUE)
An Approach towards Context-sensitive and User-adapted Access to
Heterogeneous Data Sources, Illustrated in the Television Domain
2010-45 Vasilios Andrikopoulos (UvT)
A theory and model for the evolution of software services
2010-46 Vincent Pijpers (VU)
e3alignment: Exploring Inter-Organizational Business-ICT Alignment
2010-47 Chen Li (UT)
Mining Process Model Variants: Challenges, Techniques, Examples
2010-48 Withdrawn
2010-49 Jahn-Takeshi Saito (UM)
Solving Difficult Game Positions
2010-50 Bouke Huurnink (UVA)
Search in Audiovisual Broadcast Archives
2010-51 Alia Khairia Amin (CWI)
Understanding and Supporting Information Seeking Tasks in Multiple Sources
2010-52 Peter-Paul van Maanen (VU)
Adaptive Support for Human-Computer Teams:
Exploring the Use of Cognitive Models of Trust and Attention
2010-53 Edgar Meij (UVA)
Combining Concepts and Language Models for Information Access

2011:

2011-01 Botond Cseke (RUN)
Variational Algorithms for Bayesian Inference in Latent Gaussian Models
2011-02 Nick Tinnemeier(UU)
Organizing Agent Organizations. Syntax and Operational Semantics of an
Organization-Oriented Programming Language
2011-03 Jan Martijn van der Werf (TUE)
Compositional Design and Verification of Component-Based Information Systems
2011-04 Hado Philip van Hasselt (UU)
Insights in Reinforcement Learning; Formal analysis and empirical evaluation
of temporal-difference learning algorithms
2011-05 Bas van de Raadt (VU)
Enterprise Architecture Coming of Age - Increasing the Performance of an Emerging Discipline
2011-06 Yiwen Wang(TUE)
Semantically-Enhanced Recommendations in Cultural Heritage
2011-07 Yujia Cao (UT)
Multimodal Information Presentation for High Load Human Computer Interaction

2011-08 Nieske Vergunst (UU)
BDI-based Generation of Robust Task-Oriented Dialogues
2011-09 Tim de Jong (OU)
Contextualised Mobile Media for Learning
2011-10 Bart Bogaert (UvT)
Cloud Content Contention
2011-11 Dhaval Vyas (UT)
Designing for Awareness: An Experience-focused HCI Perspective
2011-12 Carmen Bratosin (TUE)
Grid Architecture for Distributed Process Mining
2011-13 Xiaoyu Mao (UvT)
Airport under Control; Multiagent Scheduling for Airport Ground Handling
2011-14 Milan Lovric(EUR)
Behavioral Finance and Agent-Based Artificial Markets
2011-15 Marijn Koolen (UVA)
The Meaning of Structure: the Value of Link Evidence for Information Retrieval
2011-16 Maarten Schadd (UM)
Selective Search in Games of Different Complexity
2011-17 Jiyin He (UVA)
Exploring Topic Structure: Coherence, Diversity and Relatedness
2011-18 Mark Ponsen (UM)
Strategic Decision-Making in Complex Games
2011-19 Ellen Rusman (OU)
The Mind ' s Eye on Personal Profiles
2011-20 Qing Gu (VU)
Guiding service-oriented software engineering - A view-based approach
2011-21 Linda Terlouw (TUD)
Modularization and Specification of Service-Oriented Systems
2011-22 Junte Zhang (UVA)
System Evaluation of Archival Description and Access
2011-23 Wouter Weerkamp (UVA)
Finding People and their Utterances in Social Media
2011-24 Herwin van Welbergen (UT)
Behavior Generation for Interpersonal Coordination with Virtual Humans On Specifying,
Scheduling and Realizing Multimodal Virtual Human Behavior
2011-25 Syed Waqar ul Qounain Jaffry (VU)
Analysis and Validation of Models for Trust Dynamics
2011-26 Matthijs Aart Pontier (VU)
Virtual Agents for Human Communication - Emotion Regulation and
Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots
2011-27 Aniel Bhulai (VU)
Dynamic website optimization through autonomous management of design patterns
2011-28 Rianne Kaptein (UVA)
Effective Focused Retrieval by Exploiting Query Context and Document Structure
2011-29 Faisal Kamiran (TUE)
Discrimination-aware Classification
2011-30 Egon van den Broek (UT)
Affective Signal Processing (ASP): Unraveling the mystery of emotions