Department of Information and Computer Science

Learning Mental States from Biosignals

Melih Kandemir

Aalto University

DOCTORAL DISSERTATIONS

Aalto University publication series
DOCTORAL DISSERTATIONS 61/2013

Learning Mental States from Biosignals

Melih Kandemir

Doctoral dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the School of Science for public examination and debate in Auditorium D of the main building at the Aalto University School of Science (Espoo, Finland) on the 4th of May 2013 at noon.

Aalto University
School of Science
Department of Information and Computer Science

Supervising professors
Prof. Samuel Kaski

Thesis advisors
Dr. Arto Klami

Preliminary examiners
Dr. Päivi Majaranta, University of Tampere, Finland
Dr. David Roi Hardoon, SAS, Singapore

Opponent
Dr. Cristina Conati, University of British Columbia, Canada

Aalto University publication series
DOCTORAL DISSERTATIONS 61/2013

ISBN 978-952-60-5116-1 (printed)
ISBN 978-952-60-5117-8 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)
http://urn.fi/URN:ISBN:978-952-60-5117-8

Unigrafia Oy
Helsinki 2013

Finland

441 697
Printed matter

Aalto University

Abstract

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

Author
Melih Kandemir

Name of the doctoral dissertation
Learning Mental States from Biosignals

Publisher School of Science

Unit Department of Information and Computer Science

Series Aalto University publication series DOCTORAL DISSERTATIONS 61/2013

Field of research Computer and Information Science

Manuscript submitted 20 November 2012 **Date of the defence** 4 May 2013

Permission to publish granted (date) 14 March 2013 **Language** English

☐ **Monograph** ☒ **Article dissertation (summary + original articles)**

Abstract

As computing technology evolves, users perform more complex tasks with computers. Hence, users expect from user interfaces to be more proactive than reactive. A proactive interface should anticipate the user's intentions and take the right action without requiring a user command. The crucial first step for such an interface is to infer the user's mental state, which gives important cues about user intentions. This thesis consists of several case studies on inferring mental states of computer users.

Biosensing technology provides a variety of hardware tools for measuring several aspects of human physiology, which is correlated with emotions and mental processes. However, signals gathered with biosensors are notoriously noisy. The mainstream approach to overcome this noise is either to increase the signal precision by expensive and stationary sensors or to control the experiment setups more heavily. Both of these solutions undermine the usability of the developed methods in real-life user interfaces.

In this thesis, machine learning is used as an alternative strategy for handling the biosignal noise in mental state inference. Computer users have been monitored under loosely controlled experiment setups by cheap and inaccurate biosensors, and novel machine learning models that infer mental states such as affective state, mental workload, relevance of a real-world object, and auditory attention are built.

The methodological contributions of the thesis are mainly on multi-view learning and multitask learning. Multi-view learning is used for integrating signals of multiple biosensors and the stimuli. Multitask learning is used for inferring multiple mental states at once, and for exploiting the inter-subject similarities for higher prediction accuracy. A novel multitask learning algorithm that transfers knowledge across multi-view learning tasks is introduced. Another novelty is a Bayesian factor analyzer with a time-dependent latent space that captures the dynamic nature of biosignals better than methods that assume independent samples. The overall outcome of the thesis is that it is feasible to predict mental states from unobtrusive biosensors with reasonable accuracy using state-of-the-art machine learning models.

Keywords Multitask Learning, Multiple Kernel Learning, Probabilistic Modeling, Affective Computing, Intelligent User Interfaces

ISBN (printed) 978-952-60-5116-1 **ISBN (pdf)** 978-952-60-5117-8

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

Location of publisher Espoo **Location of printing** Helsinki **Year** 2013

Pages 184 **urn** http://urn.fi/URN:ISBN:978-952-60-5117-8

Preface

This thesis comprises my studies as a doctoral student in the Statistical Machine Learning and Bioinformatics (MI) group in the Department of Information and Computer Science (ICS), Aalto University School of Science (former Adaptive Informatics Research Centre (AIRC) of the Laboratory of Computer and Information Science). I am also affiliated with Helsinki Institute for Information Technology HIIT under Finnish Center of Excellence in Computational Inference (COIN). My research was funded by Nokia Research Center, Multidisciplinary Institute of Digitalisation and Energy (MIDE) research program of Aalto University, and the Pattern Analysis, Statistical Modeling and Computational Learning Network of Excellence (PASCAL2 EU Network of Excellence). I thank Helsinki Doctoral Programme in Computer Science (Hecse) and Finnish Doctoral Programme in Computational Sciences (FICS) for supporting my participation to scientific conferences, giving me the chance of improving my presentation skills in summer schools and getting very valuable feedback about my research from a large community of researchers during my doctoral study.

I am very grateful to my supervisor Prof. Samuel Kaski for giving me the chance to do research in his research group, which is full of very talented scientists, and providing both the freedom to show my creativity and the guidance to learn how to do world-class science. I wish to thank my instructor, Dr. Arto Klami, for his effective leadership and his patience in helping me even in very practical details. I would like to thank to my thesis pre-examiners Dr. Päivi Majaranta and Dr. David Hardoon for their very useful comments. I express my gratitudes to my M.Sc. thesis supervisor Çiğdem Gündüz Demir for admitting me to the science community, in which I dream about staying lifelong.

I am deeply beholden to all current and former members of the MI

group for their warm friendship and all the inspiring interactions we had. I am indebted the Nokia Research Center for supporting me financially, and to Akos Vetek and Dr. Jari Kangas for their productive collaboration. I thank Veli-Matti Saarinen for his collaboration in designing and conducting experiments, to Professor Pekka Orponen for providing us a wonderful research environment, to our secretaries Leila Koivisto, Minna Kauppila, and Tarja Pihamaa, and also to Dr. Miki Sirola for their support whenever required. I am thankful to my office mates and co-authors Antti Ajanki and Dr. Mehmet Gönen for all the warm chats we had, all their help in practical issues, and their very valuable collaboration.

I feel very lucky to have met Tommi Suvitaival, Ali Faisal, Alp Karakoç, Aydın Karaer, Utku Öztürk, Suleiman Ali Khan, Caner Demirpolat, Ömer Furkan Tercan, Dr. Sohen Seth, Hande Topa, and Dr. Onur Dikmen, and to have spent an amazing time together. I also send my special thanks to Tommi Suvitaival and Dr. Sohan Seth for the exciting tennis games we had, and to Dr. José Caldas for his close company.

I send my very special thanks to Süleyman Doğankaya for taking the beautiful Bosphorus Bridge picture and letting me to put it on the thesis cover.

Finally, I would like to express my deepest gratitudes to my grandmother Hayriye Turgut, father Ali Kandemir, mother Fatma Kandemir, sister Melike Kandemir, and my beloved wife Fatma Gül Özen Kandemir who made this thesis possible with their love and endless moral support.

Heidelberg, March 26, 2013,

Melih Kandemir

Contents

List of Publications

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

I Melih Kandemir, Veli-Matti Saarinen, Samuel Kaski. Inferring Object Relevance from Gaze in Dynamic Scenes. In *Eye Tracking Research and Applications*, Austin TX, USA, pages 105–108, 2010.

II Antti Ajanki, Mark Billinghurst, Hannes Gamper, Toni Järvenpää, Melih Kandemir, Samuel Kaski, Markus Koskela, Mikko Kurimo, Jorma Laaksonen, Kai Puolamaki, Teemu Ruokolainen, Timo Tossavainen. An augmented reality interface to contextual information. *Virtual Reality*, 15(2):161–173, 2011.

III Mehmet Gönen, Melih Kandemir, Samuel Kaski. Multitask Learning Using Regularized Multiple Kernel Learning. In *International Conference on Neural Information Processing*, Shanghai, China, pages 500–509, 2011.

IV Melih Kandemir, Samuel Kaski. Learning Relevance from Natural Eye Movements in Pervasive Interfaces. In *International Conference on Multimodal Interaction*, Santa Monica, CA, USA, pages 85–92, 2012.

V Melih Kandemir, Arto Klami, Akos Vetek, Samuel Kaski. Unsupervised inference of auditory attention from biosensors. In *European Conference on Machine Learning and Practice of Knowledge Discovery in Databases, Bristol, UK, 2012, Lecture Notes in Computer Science,*

7524:403–418, 2012.

VI Melih Kandemir, Akos Vetek, Mehmet Gönen, Arto Klami, Samuel Kaski. Multi-task and multi-view learning of user state. *Submitted to a journal*, 24 pages, 2012.

Author's Contribution

Publication I: "Inferring Object Relevance from Gaze in Dynamic Scenes"

The experiment design was the joint work of all authors. The author constructed the experiment setup, made the experiments in collaboration with the second author, analyzed the data, and co-wrote the manuscript.

Publication II: "An augmented reality interface to contextual information"

The author implemented the object relevance predictor, wrote the related parts in the manuscript, helped in conducting the experiments and revising the text in general.

Publication III: "Multitask Learning Using Regularized Multiple Kernel Learning"

The author collaborated with the first author in the design and implementation of the model, co-wrote the manuscript, and collected one of the data sets (CogState) used in experiments.

Publication IV: "Learning Relevance from Natural Eye Movements in Pervasive Interfaces"

The experiment was designed by both authors collaboratively. The author constructed the experiment setup, made the data analysis, and co-wrote the manuscript.

Publication V: "Unsupervised inference of auditory attention from biosensors"

The experiment was designed in collaboration with all authors. The author constructed the experiment setup. First and second authors made the data analysis. The manuscript is the joint work of all authors.

Publication VI: "Multi-task and multi-view learning of user state"

The experiment was designed in collaboration with all authors. The author constructed the experiment setup, made the data analysis, and co-wrote the manuscript.

List of Abbreviations and Symbols

AUC	Area Under Curve
ARD	Automatic Relevance Determination
AVA	All-Versus-All
CDF	Cumulative Distribution Function
CCA	Canonical Correlation Analysis
ECG	Electrocardiography
EEG	Electroencephalography
EMG	Electromyography
EOG	Electrooculography
fMRI	functional Magnetic Resonance Imaging
FN	False Negative
FP	False Positive
GP	Gaussian Process
GPLVM	Gaussian Process Latent Variable Models
GFA	Group Factor Analysis
GSR	Galvanic Skin Response
HRV	Heart Rate Variability
IRLS	Iterative Reweighted Least Qquares
IVM	Information Vector Machine
KKT	Karush-Kuhn-Tucker
LOO-CV	Leave-One-Out Cross Validation
LTW	Linear Time Warping
MAP	Mean Average Precision
MCMC	Markov Chain Monte Carlo
MEG	Magneto-encephalography
MKL	Multiple Kernel Learning
ML-II	Type II maximum likelihood
MLE	Maximum Likelihood Estimation

MSE	Mean Squared Error
OVA	One-versus-all
PRBEP	Precision-Recall Breakeven Point
RBF	Radial Basis Function
RKHS	Reproducing Kernel Hilbert Space
RMSE	Root Mean Squared Error
ROC	Receiver Operating Characteristic
RVM	Relevance Vector Machine
SMO	Sequential Minimal Optimization
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
x, y, z	Scalar variables
$\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$	Vectoral variables
$\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$	Matrix variables
$\mathcal{N}(\mu, \sigma^2)$	Gaussian (normal) distribution with mean μ and variance σ^2
$p(x)$	Probability density function of x
$p(\boldsymbol{x})$	Joint probability density function of $\boldsymbol{x} = [x_1, x_2, \cdots, x_D]$
$p(x\|y)$	Conditional probability density function of x given y
$\exp(.)$	Exponential function
$\mathbf{I}$	Identity matrix
$\log(.)$	Logarithmic function
$\otimes$	Element-wise matrix multiplication
δ_{xy}	Kronecker delta function that returns 1 if $x = y$, and 0 otherwise
$\|\mathbf{X}\|_F$	Frobenius norm of matrix $\mathbf{X}$
$\mathbb{E}(x)$	Expected value of random variable x
$\mathbf{X}^T$	Transpose of matrix $\mathbf{X}$
$\mathbf{X}^{-1}$	Inverse of matrix $\mathbf{X}$
$\mathcal{D}$	A collection of observations

1. Introduction

1.1 Motivation

The interaction of humans with computers is becoming more complex as the information technology evolves. Both the amount of data available to users and the complexity of everyday tasks performed by users are growing, which calls for more sophisticated interaction methods than the standard methods of item selection or text inputting via keyboard and mouse. Today's computer systems are expected to participate actively in the process of interaction rather than behaving like finite-state automata. The user asks more abstract questions to the computer such as to find the most relevant information source for the current task, rather than to locate a document by its precise name. During moments of heavy multitasking (e.g. video conferencing while reading documents related to the meeting, or writing a report while following high-priority e-mails), the user demands more empathy from computers, such as not being disturbed by an e-mail alert at a very crucial moment of a video conference.

Implementation of this sort of intelligent behaviour presupposes that the computer is aware of the state of the user's mind. The main question of this thesis is whether such an awareness can be possible. The thesis also seeks answers to a couple of sub-questions that arise from this main question:

- Which aspects of the user's mind can be revealed from indicators that can be measured in an unobtrusive way?

- How can we infer these aspects of the user's mind from the available indicators?

In this thesis, the state of any mental process of the user (emotions, interests, intentions, mental workload, etc.) at a certain time is named as a *mental state*. The mental states investigated in this thesis are listed below:

- **Object relevance:** This is a measure of how much the user is interested in an object she is interacting. In Publication I, the ranking of real-world objects with respect to their relevance is inferred. And in Publication II and Publication IV, the actual relevances of real-world objects are estimated from eye movement patterns.

- **Affective state:** This is a quantitative measure of the emotional state. In this thesis, the valence-arousal [107] scale has been adopted to quantize the emotional state. *Valence* refers to the intrinsic attractiveness (positive valence) or aversiveness (negative valence) of an event, object, or situation [36]. And *arousal* is the degree of reactiveness to stimuli. In Publication III and Publication VI, valence and arousal are inferred from a set of biosignals.

- **Mental workload (Cognitive load):** This denotes how busy the working memory is with a cognitive process. In Publication III and Publication VI, mental workload is represented in a discrete scale and inferred from a set of biosignals.

- **Liking:** In Publication VI, liking is used as a discrete measure of how much the user liked a presented video. In this work, liking is inferred from a comprehensive set of biosignals.

- **Auditory Attention:** This is the measure of how much attention the user pays to an audio content. In Publication V, auditory attention is quantized in a discrete scale from 1 to 4, indicating increasing level of attention. In this work, auditory attention is inferred from three biosignals.

Among these mental states, object relevance, liking, and auditory attention are meant for being useful especially in content selection. The interest and attention of the user to the previously presented contents can guide educated guesses on which items among the available set would be

the most desirable for the user. This can be considered as an information retrieval problem in a broader sense. On the other hand, affective state and mental workload are chosen to be useful in especially in deciding how and when the selected content should be presented. For instance, the content can be filtered when the user is in negative valence or under high mental workload, and can be made more salient when the user's arousal is low.

This thesis consists of case studies on inferring the abovementioned mental states of users who are monitored by biosensors on novel experimental setups. The thesis also introduces novel machine learning models tailored for these inference tasks.

1.2 Proactive user interfaces

User interfaces can be classified into two categories based on how they communicate with the user: i) command-based interfaces, ii) proactive interfaces. Command-based interfaces work in a finite-state-automaton fashion; given an explicit command, they take the associated action. On the other hand, proactive interfaces [116] constantly monitor the user, anticipate the user's interests and automatically execute the compatible actions. This type of user interfaces are also referred to as *non-command interfaces* [89]. *Attentive interfaces* [122] are also a special type of proactive interfaces with a focus narrowed on a single mental state category: attention.

As opposed to command-based interfaces, the interaction in proactive interfaces is implicit, in which the computer participates actively. The primary difficulty in developing such interfaces lies in inferring the user interests. This thesis introduces machine learning techniques that take a step towards solving this problem by extracting cues from the user's mental processes and physiology.

1.3 Biosensing

Biosensing technology enables measuring many aspects of human physiology, such as neuronal activity, eye movements, pupil diameter, heart rate, body temperature, skin conductance etc. Advances in recent years brought about unobtrusive biosensors that do not hinder subjects from

performing real-world tasks, making them feasible for experimental studies in naturalistic setups. In this thesis, users are measured under various novel experimental scenarios by biosensors such as electroencephalograph (EEG), heart rate sensor, accelerometer, and eye tracker. The biosensing technologies used in the thesis are detailed in Chapter 6.

1.4 Machine learning

Machine learning is a field of science that deals with capturing complicated properties of noisy data based on mathematical models. It is used in solving problems such as predicting outputs from given inputs, classifying patterns, forming groups of similar samples, representing the data in a lower dimensional space, extracting the relationships between co-occuring data sets, etc. The discriminative property of machine learning algorithms is that they take into account uncertainty in a principled way. This field has produced standard tools widely used in the scientific community, such as support-vector machines, linear discriminant analysis, and Gaussian processes. As an introductory text to the field, see, for example, [15].

Machine learning lies at the heart of the methodology of this thesis. The thesis introduces novel machine learning models for mental state inference. These models effectively handle the notoriously large amount of noise in the biosensor data. They also reveal interesting properties of human nature, such as how synchronized human body is with the data the user is interacting (Publication V), and how similar the emotional reactions of subjects are to certain conditions (Publication VI).

1.5 Affective computing

Application of machine learning to biosensor data for inferring the emotional state has been studied under the name of *affective computing* [94]. This discipline studies methods to recognize and interpret human affects from signals such as speech, facial expression, skin conductance, heart rate, brain activity, etc. Publication III and Publication VI can be subscribed to this field of science.

1.6 Contribution of the thesis

In this thesis, inference of a set of mental states is investigated in the following novel experimental setups:

- **Inferring the relevance ranking of objects in real-world video scenes from gaze patterns**: The user watches a video of real-world scenes, where objects are augmented with textual information. Eye movements of the user are recorded in the meantime. After the experiment, the user ranks the objects on each frame with respect to their relevance at that time. The analysis task is to infer the rankings from eye movement patterns and visual properties of the objects (such as their size and relative distance). See Publication I for details.

- **Inferring the relevance of real-world objects from gaze patterns when the user is mobile**: The user explores an experimental art gallery, wearing a helmet with an attached eye tracker. She marks the paintings she likes most. Her eye movements are recorded in the meantime. The goal is to infer the marked paintings from eye movements. See Publication IV for details. Publication II introduces a pilot system for proactive contextual information access, where gaze is used for the first time to retrieve information about the objects in the scene that is relevant to the inferred context. This system is then used as an infrastructure in Publication IV.

- **Inferring the affective state, mental workload, and liking of a desktop computer user from biosignals**: The user performs a bunch of realistic tasks on a desktop computer such as exploring images, filling in surveys, and solving logical puzzles. The user is measured by four biosensors (EEG, ECG, motion sensor, and pupil dilation) in the meantime. After the experiment, the user annotates her affective state and mental workload at several stages of the experiment. The task is to infer these annotations from the measured biosignals. See Publication III and Publication VI for details.

- **Inferring auditory attention from the correlation between biosignals of the listener and the listened audio content**: The user listens to an audio while simultaneously performing another visual task.

Single-channel EEG, 3D body motion vector, and pupil dilation are measured from her body in the meantime. Her ground-truth level of attention to the audio is controlled by playing with the difficulty of the visual task. The task is to infer this ground-truth level of attention from biosignals. See Publication V for details.

To infer the mental states under the abovementioned experimental setups, the following three novel machine learning models have been tailored:

- A multitask learning model that shares information across learning tasks by inducing similar tasks to combine multiple data views in similar ways (Publication III and Publication VI). Inferring each mental state and inferring mental state for each user are treated as related tasks, and each sensor is treated as a view. This model demonstrates higher prediction accuracy and demands less computational time than its predecessors on several benchmark data sets. See Publication III for details.

- A Gaussian process classifier that classifies multivariate time series. In Publication IV, eye movements inside a target object are modeled as a time series with an attached binary label indicating whether that object is relevant. Once the classifier has been trained, it predicts the relevance of the newly seen objects from eye movements better than dwell-time thresholding, which is the only earlier method that solves the same problem.

- A Bayesian formulation of canonical correlation analysis (CCA) tailored for time-dependent data, such as biosignals. An existing Bayesian CCA model is extended with a Markov chain driven latent representation. This model predicts the user's auditory attention without requiring any training labels (Publication V) from the correlation between the audio content and the biosignals of the user listening to it. The model performs better than time-independent variants of CCA in predicting attention in large time periods.

1.7 Organization of the thesis

In this manuscript, an overview of the computational methods used in the published thesis work is given, and the main contributions are highlighted. In Chapter 2, a methodological background is developed starting from basic concepts of machine learning. In Chapter 3, short descriptions of supervised learning models referred in the publications are given. These models are used either as components of the proposed models or as baselines. In Chapter 4 the *multi-view learning* concept is advertised for incorporating data coming from different sensors. In Chapter 5, the *multitask learning* idea is suggested for learning related tasks together by a single model. In Chapter 6, a flavour of the biosensing technology is provided, focusing on the sensors used in the experiments. In Chapter 7, the experiment setups constructed in the publications and the inferred mental states are explained. Finally in Chapter 8, ideas on possible future research directions motivated by this thesis are given.

2. Machine Learning Basics

2.1 Learning a model from data

Machine learning models serve for two main goals:

- predicting the outcome of a non-deterministic process from available input factors.

- revealing intrinsic properties of data (such as, the relationship between factors, or the low-dimensional manifold the data lie on), which is particularly useful when our prior knowledge on the underlying process is too limited to devise a model.

In this chapter, machine learning problems encountered throughout the thesis are explored. An introduction is also provided to methods of assessing model performance. The chapter classifies machine learning models as *supervised* and *unsupervised* models based on whether they use ground-truth output data, and as *parametric* and *non-parametric* models, based on modeling principles. These dichotomies are simplified for structuring the methods used in this thesis.

2.2 Probabilistic analysis

Statisticians have two competing views on what a *probability* is: the frequentist view, and the Bayesian view. For frequentists, probability is a characteristic of an event that can be calculated by repeating an experiment infinitely many times and taking the fraction of the occurrence of the event. For Bayesians, it is a measure of the belief in whether the event

can take place under the given conditions. Frequentists say that probability is an objective measure, while Bayesians define it as a subjective value [27].

Suppose we have a probabilistic model $p(\mathrm{x}|\boldsymbol{\theta})$ of a set of joint variables $\boldsymbol{x} = [x_1, x_2, \cdots, x_D]$ parameterized by $\boldsymbol{\theta} = [\theta_1, \theta_2, \cdots, \theta_K]$. The goal is to estimate $\boldsymbol{\theta}$ from a set of observations $\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N]$. The probability of the occurrence of the observations $p(\mathbf{X}|\boldsymbol{\theta})$ is called the *likelihood*. From the frequentist point of view, the model parameters are scalar values that converge to their true value as $N \to \infty$. In applications, we are restricted to a finite set of observations. Hence, the parameter values $\boldsymbol{\theta}_{MLE}$ that give the highest likelihood are the best possible guess we can make. In frequentist inference, model parameters are estimated by $\boldsymbol{\theta}_{MLE}$, which is called *maximum likelihood estimation (MLE)*. The probabilistic model in Publication I is inferred using maximum likelihood estimation.

The Bayesian approach sees model parameters as probability distributions, as everything else in the model. The inference problem is estimating $p(\boldsymbol{\theta}|\mathbf{X})$, which is called the *posterior*. By *Bayes' rule* [11], this distribution can be decomposed as

$$p(\boldsymbol{\theta}|\mathbf{X}) = \frac{p(\mathbf{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{X})}.$$

The additional distribution on the parameters $p(\boldsymbol{\theta})$ is called the *prior*, since it is observation-independent. Bayesian analysis allows prior beliefs to be incorporated into the model via the chosen prior. The source of these beliefs could be expert knowledge or the outcomes of earlier analyses. In Bayesian formulation, an event that never occurs in the observations does not necessarily have non-zero probability. The denominator $p(\mathbf{X})$ is called the *evidence*. It gives an overall idea of how well the model fits to the data.

Inference of a Bayesian model is about finding the posterior distribution that best explains the unseen samples. However, the posterior distributions of many models cannot be expressed in closed form as a function of parameters. In these cases, the posterior is approximated using techniques such as Markov Chain Monte Carlo (MCMC) sampling [105], and variational inference [57]. In MCMC, the posterior is approximated by a large collection of samples randomly drawn from the posterior distribution. In variational Bayes, it is approximated by a combination of simpler tractable distributions. MCMC often gives higher accuracy, while learning with variational Bayes is usually faster. Variational inference is used for the Bayesian model in Publication V. A detailed investigation of Bayesian modeling can be found in [42].

2.3 Unsupervised learning

Unsupervised learning is the task of finding structure in the data where the samples are associated with output labels. An unsupervised model separates the data into an assumed structure and pure noise. Some examples of unsupervised learning problems are:

- *clustering*: Forming groups of similar samples. A simple example is the k-means clustering algorithm [80]. Infinite mixture models with Dirichlet process priors [101] and latent dirichlet allocation [16] are more advanced variants.

- *density estimation*: Fitting a probability density to data. A simple choice is the normal distribution. The learning task is to infer the unknown mean and variance.

- *dimensionality reduction*: Finding a low-dimensional manifold that best explains data for compression, noise removal, or visualisation. A typical example is Principal Component Analysis (PCA) [93]. Gaussian Process Latent Variable Models (GPLVM) [74] and Informative Discriminant Analysis [63] are more recent variants.

- *dependency modeling*: Modeling dependencies between co-occurring data sets. A typical example is Canonical Correlation Analysis (CCA) [53, 125].

In Publication V, an unsupervised model that predicts auditory attention from the dependencies between the user biosignals and audio is introduced. See Section 7.3 for details.

2.4 Supervised learning

In supervised learning, each sample in the data has an assigned output. The task is to learn a mapping from the input samples to the output values. This mapping is meant for predicting the output of new samples as successfully as possible.

The performance of a supervised learning algorithm is evaluated by splitting the data set at hand into two parts. The first part is shown to the

model with the known outputs. This is called the *training* (or *learning*) phase. Then, the outputs of the second part are predicted by the learned model, and the predictions are compared to the true output values. This is called the *test* (or *generalization*, or *evaluation*) phase. The former data split is called the *training data set* and the latter *test data set*.

Supervised learning problems can be classified into three main categories based on the output structure: regression, classification, and ordinal regression. In the rest of this section, these three types of supervised learning algorithms are briefly described and examples to these algorithms are demonstrated. Other types not used in this thesis, such as structured prediction [8], are ignored for simplicity.

2.4.1 Regression

In regression, the goal is to predict real valued outputs $y \in \mathbb{R}$ from input samples $\boldsymbol{x}$. A simple regression method is linear regression

$$p(y|\boldsymbol{w}, \beta) = \mathcal{N}(\boldsymbol{w}^T\boldsymbol{x}, \sigma^2)$$

where we assume that the output is a weighted linear combination of the input variables with an additive residual white noise with variance σ^2. The maximum-likelihood estimate of this model has the following analytical solution

$$\hat{\boldsymbol{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{y}.$$

Geometrically, linear regression corresponds to fitting a hyperplane on the data space that reduces the expected prediction error. Figure 2.1 illustrates the idea on one-dimensional data.

More advanced regression methods are used in Publication V to infer the level of user attention from biosignals. These methods are detailed in Chapter 3.

2.4.2 Classification

In classification, the task is to assign an input sample $\boldsymbol{x}$ to one of the possible categories $\mathcal{C} = \{c_1, ..., c_K\}$. An example is handwritten digit recognition, where we take the pixel values of the image of a handwritten digit as the input, and predict the digit as the output.

A simple classification method is *logistic regression*. We explain this model for binary classification $\{0, 1\}$; its extension to the multiclass case is

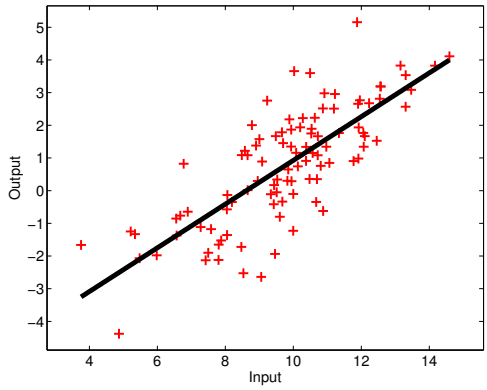


Figure 2.1. Linear regression finds a hyperplane (a line for one-dimensional input) that best maps the inputs to the outputs. A synthetic data set of 100 points generated from a bivariate normal distribution with mean $[10; 1]$ and covariance $[21; 0.51.5]$ is shown as red pluses; the first variate being the one-dimensional input data (shown on the x-axis), and the second the output values (shown on the y-axis). The black line is the predictor learned by linear regression.

straightforward. Logistic regression directly models the class conditionals in a discriminative fashion by a linear regressor on the *log odds*

$$\log\left[\frac{p(y=1|\boldsymbol{x},\boldsymbol{w})}{p(y=0|\boldsymbol{x},\boldsymbol{w})}\right] = \boldsymbol{w}^T\boldsymbol{x}.$$

By rearranging the terms, this equals to squeezing the output of a linear regressor by the logistic function

$$p(y=1|\boldsymbol{x},\boldsymbol{w}) = \frac{1}{1+\exp(\boldsymbol{w}^T\boldsymbol{x})}.$$

Squeezing the output of a regression method by a sigmoid function to model class-conditional densities is a common trick also used in state-of-the-art classifiers such as Gaussian processes, as will be seen in Chapter 3.

In logistic regression, the likelihood of a data set $\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N]$ and the corresponding vector $\boldsymbol{y} = [y_1, \cdots, y_N]$ of binary labels $\{0, 1\}$ is

$$p(\boldsymbol{y}|\mathbf{X},\boldsymbol{w}) = \prod_{i=1}^{N} p(y_i|\boldsymbol{x}_i,\boldsymbol{w})^{y_i}(1-p(y_i|\boldsymbol{x}_i,\boldsymbol{w}))^{1-y_i}.$$

The negative log-likelihood then takes the form [51]

$$J(\boldsymbol{w}) = -\ln p(\boldsymbol{y}|\mathbf{X},\boldsymbol{w}) = -\sum_{i=1}^{N}\{\ln p(y_i|\boldsymbol{x}_i,\boldsymbol{w}) + (1-y_i)\ln(1-p(y_i|\boldsymbol{x}_i,\boldsymbol{w}))\} \tag{2.1}$$

$$= -\sum_{i=1}^{N}\left\{y_i\boldsymbol{w}^T\boldsymbol{x}_i - \ln(1+\exp(\boldsymbol{w}^T\boldsymbol{x}_i))\right\}.$$

The maximum-likelihood solution to this model is not available in closed-form due to the non-linear sigmoid function. However, the concavity of the negative log-likelihood enables to reach the global maximum using iterative methods. A standard choice is the *Newton-Raphson* method which suggests

$$\boldsymbol{w}^{new} = \boldsymbol{w}^{old} - \mathbf{H}^{-1}\nabla J(\boldsymbol{w})$$

as the update rule, where $\mathbf{H}$ is the Hessian matrix with respect to $\boldsymbol{w}$. Using this method, the update rule for logistic regression becomes

$$\boldsymbol{w}^{new} = \boldsymbol{w}^{old}(\mathbf{X}^T\mathbf{R}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{R}\boldsymbol{z}$$

where $\mathbf{R}$ is an $N \times N$ diagonal matrix with $R_{ii} = p(y_i|\boldsymbol{w})(1 - p(y_i|\boldsymbol{w}))$ and

$$\boldsymbol{z} = \mathbf{X}\boldsymbol{w}^{old} + \mathbf{R}^{-1}(\boldsymbol{y} - \boldsymbol{p})$$

with $\boldsymbol{p} = [p(y_1|\boldsymbol{w}^{old}, \cdots, p(y_N|\boldsymbol{w}^{old})]$. This method is also known as *iterative reweighted least squares (IRLS)* [106] because at each iteration it solves a least-squares problem weighted by $\mathbf{R}$ [51].

Mental states such as affective state, mental workload, and object relevance are predicted from biosignals in Publication III, Publication VI, and Publication IV using state-of-the-art classification methods.

2.4.3 Ordinal regression

In ordinal regression, the output is discrete as in classification, but there is an ordering relationship between the possible output values. In other words, this is a regression problem with discrete output values. See [73] for a review of applications of ordinal regression to medical data analysis.

A simple ordinal regression model can be obtained by modifying the logistic regression model so that instead of direct class conditionals, the log odds are taken with respect to the cumulative class conditional distributions

$$\log\left[\frac{p(y <= k|\boldsymbol{x}, \boldsymbol{w})}{1 - p(y <= k|\boldsymbol{x}, \boldsymbol{w})}\right] = {\boldsymbol{w}_k}^T\boldsymbol{x}$$

for each category k. The CDF of the class conditional then becomes

$$p(y <= k|\boldsymbol{x}, \boldsymbol{w}) = \frac{\exp({\boldsymbol{w}_k}^T\boldsymbol{x})}{1 + \exp({\boldsymbol{w}_k}^T\boldsymbol{x})}.$$

The inference of this model is done by Newton-Raphson updates very similarly to standard logistic regression.

This model is used in Publication I to predict relevance rankings of real-world objects from eye movements in video scenes.

2.5 Learning parametric and non-parametric models

Most machine learning methods can be fully described by a finite set of parameters $\boldsymbol{\theta} = [\theta_1, \cdots, \theta_P]$, where the number of parameters $P < \infty$ is predetermined by the model assumptions. For example, the linear regression has $D+1$ parameters ($w_1, \cdots, w_D$, and b) for D dimensional data. This kind of models are known as *parametric models*. In probabilistic parametric models, new samples are predicted based on the posterior $p(\boldsymbol{\theta}|\mathcal{D})$ by

$$p(\boldsymbol{x}^*|\mathcal{D}) = \int p(\boldsymbol{x}^*|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta}.$$

Parametric models can be non-probabilistic as well. Let $M(\boldsymbol{\theta})$ be a model parameterized by $\boldsymbol{\theta}$. The learning phase is then about searching for a parameter set $\hat{\boldsymbol{\theta}}$ that maximizes a criterion of the expected generalization performance. The trained model $M(\hat{\boldsymbol{\theta}})$ then makes its predictions based on this estimate.

There is another group of models, called *non-parametric models* that cannot be described by a finite set of parameters. In such models, the information in the training data is not summarized by a predetermined number of parameters. Hence, learning and prediction stages cannot be entirely separated. All the training data are directly used in predicting new samples. Non-parametric models usually make less assumptions on the data distribution than parametric models. However, they demand more data for reliable performance.

In this thesis, parametric models are used in Publication I, Publication III, Publication V, and Publication VI, and non-parametric models in Publication IV and Publication V.

2.6 Tuning model hyperparameters

Many machine learning models have some parameters that are not designed solely to be learned from data. They can also be used to induce our prior beliefs and assumptions about the data into models. This type of parameters are called *hyperparameters*. If sufficient domain knowledge is available, hyperparameters can be manually tuned. Otherwise, they are tuned in a data-driven manner. Below, two popular methods are explained for data-driven hyperparameter tuning.

Type II maximum likelihood (ML-II)
This method is applicable to probabilistic models, where hyperparameters are the parameters of prior distributions. Given a data set $\mathcal{D}$, a vector of model parameters $\boldsymbol{\theta}$, and a vector of hyperparameters $\boldsymbol{\gamma}$, the type II maximum likelihood method suggests learning the hyperparameters by maximizing the marginal likelihood

$$p(\boldsymbol{y}|\mathcal{D}, \boldsymbol{\gamma}) = \int p(\boldsymbol{y}|\mathcal{D}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\boldsymbol{\gamma}) d\boldsymbol{\theta}$$

with respect to $\boldsymbol{\gamma}$. In the proper Bayesian treatment, the hyperparameters should be assigned a hyperprior $p(\boldsymbol{\gamma})$ if they will be learned from data. However in many situations, this makes the computations complicated, such as in RVMs. Hence, we are satisfied with a more biased but computationally more efficient estimate of the hyperparameters.

Cross-Validation
Validation refers to measuring the goodness-of-fit of a model. Cross-validation is a special validation method. When it is used for evaluating a model hyperparameter, the training data are split into a number of partitions at random. For each possible hyperparameter value in the set, the model is tested on each partition after being trained on the other partitions. The model performance for that hyperparameter value is measured by averaging over all partitions. Finally, the value with the highest performance is chosen. If the data are split into K partitions, the method is referred to as *K-fold cross validation*. In the extreme case, there can be N partitions. Then at each iteration, only one sample is held out, which is known as *leave-one-out (LOO) cross validation*.

Cross-validation can be used for evaluating generalization performance as well. In that case, instead of the training data, the whole data are split into partitions, the held out partition is used for testing, and the rest for training.

2.7 Measuring model performance

In this section, the performance measures for regression and classification used in the publications are listed.

2.7.1 Measuring regression performance

In regression problems, the most intuitive measure of performance is the expectation of the divergence of predictions from the true values, which is referred to as the *mean-squared error (MSE)* and usually estimated by the sample mean

$$MSE(\hat{y}) = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

where y_i and $\hat{y}_i$ are the true and predicted outputs for data point i, respectively. It is often preferred to use the square-root of MSE, called the *root-mean-squared error (RMSE)*, since it has the same unit as the output values. In Publication III, regression performance is measured using RMSE.

2.7.2 Measuring classification performance

Suppose we have N samples, and N binary predictions. The predicted labels compare with the true labels in four possible ways, as shown in Table 2.1. Each entry in this table, also known as the *confusion matrix*, shows the count of samples for which the predicted and true labels compare as the entry denotes. Many measures revealing different aspects of

		Prediction	
		0	**1**
Ground Truth	**0**	True Negative (TN)	False Positive (FP)
	1	False Negative (FN)	True Positive (TP)

Table 2.1. Confusion matrix showing each possible relationship between the predicted label and the true label. Each entry in the matrix shows the count of samples whose predicted and true labels have the relationship identified by the entry.

the model performance can be calculated from the entries of this table, such as:

- **Accuracy :** The proportion of the correctly classified samples ($(TP + TN)/N$). It is the most intuitive way of measuring classification performance, but it is sensitive to class imbalance. For a data set having 98 samples labeled as 0, and 2 samples as 1, always predicting 0 gives 98% accuracy.

- **Precision:** The proportion of correctly classified positive samples to all

samples classified as positive ($TP/(TP+FP)$).

- **Recall :** The proportion of correctly classified positive samples to all positive samples in the data ($TP/(TP+FN)$).

- F_1**-score :** It is the harmonic mean of precision and recall

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

It is a preferred performance measure for imbalanced data sets. If the cost of misdetecting each class label is equal, using *Macro* F_1 *score*, which is the mean of the F_1-scores with respect to each label versus all others, is convenient. This variant is used in Publication IV and Publication VI. For cases where recall and precision have unequal importance, this measure is extended to

$$F_\beta = (1+\beta^2) \cdot \frac{TP}{(1+\beta^2) \cdot TP + \beta^2 \cdot FN + FP}$$

where the ratio of the importance of recall to precision is tuned by β [104].

2.7.3 Receiver operating characteristic

Receiver operating characteristic (ROC) curve plots the change of recall as a function of false-positive rate ($FP/(FP+TN)$) as the decision threshold varies. ROC curves are used for comparing classifiers and for choosing the optimum decision threshold which gives the best trade-off between false positives and false negatives for a classifier. In Figure 2.2, ROC curves for two classifiers are given in blue circles and red triangles. The diagonal line shown in dashed black denotes the random chance level. The classifier shown as blue circles is better than the one shown as red triangles since its recall is higher for all false-positive rates. Two ROC curves can be quantitatively compared by the area between the curves and the false-positive rate axis. This measure, referred to as the *area under the ROC curve (AUC)*, is used as a standandard goodness measure for comparing models especially when the data set is class-imbalanced [70]. This measure is used for comparing models in Publication IV.

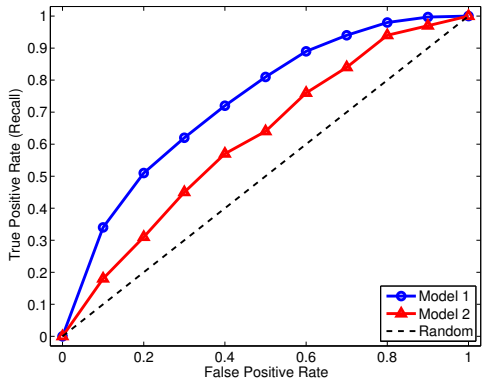


Figure 2.2. ROC curves of two models, and the random chance level, are given in blue circles, red triangles, and black dashed line, respectively. Both models are better than random chance and Model 1 is better than Model 2.

2.7.4 Precision-recall curve

Precision-recall curve is the plot of change in precision as a function of recall as the decision threshold varies. It is frequently used as a goodness measure by the information retrieval community [7] for comparing models where output labels are binary (relevant and irrelevant) and there is a grand imbalance between classes. In Figure 2.3, the precision-recall curves of two imaginary models are given as blue circles and red triangles. The curve of the model shown as blue circles is above that of the model shown as red triangles, which means that for any relevance level, the blue model has higher precision. Hence, the blue model is said to have better retrieval performance than the red one. The inverse proportion between precision and recall in both models is due to the well-known *precision-recall trade-off* [7]. Typical measures to summarize precision-recall curves include area under the precision-recall curve (applied in the same way as the ROC curve), and the *precision-recall breakeven point (PRBEP)* (the point on the curve where precision and recall are equal), and *mean average precision (MAP)* (mean of the average of the precision scores over a set of queries at threshold values where a true positive is obtained for the newly added data point).

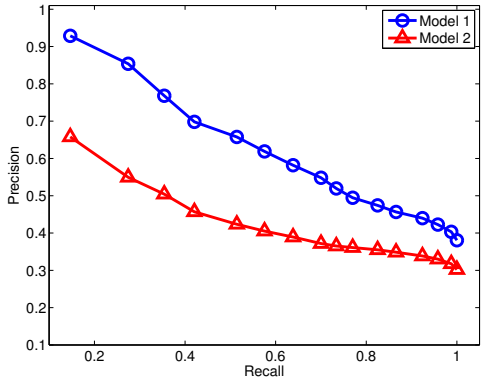


Figure 2.3. Precision-recall curves of two models are given in blue circles and red triangles. Model 1 is better than Model 2, since for each recall level, its precision is higher.

3. Supervised Learning by Kernels

Kernelizing a learning algorithm refers to changing its input space using a mapping function. Any algorithm whose formulation has the input data always in dot-product form can be kernelized by replacing these dot-product terms by a function, called a *kernel*, that satisfies certain mathematical properties. This technique is called the *kernel trick*. Kernelizing an algorithm brings in benefits, such as:

1. obtaining a richer feature representation, which yields better class seperability,

2. integrating data coming from multiple modalities into a single learner,

3. operating on non-numeric or structured input data, such as string sequences and time series.

Kernel methods are used in Publication V due to 1, in Publication III and Publication VI due to 1 and 2, and in Publication IV due to 1 and 3. In this section, a brief mathematical background is given on kernels, and the kernel types and the kernel-based models used in the publications are described.

3.1 Kernels

3.1.1 Mathematical Background

We look for a mapping from the original feature space to a higher dimensional vector space: $\Phi : \mathcal{X} \rightarrow \mathcal{H}$. We prefer this new space to be associated with a dot product, and to have the possibility of assigning coordinates to

each point on the space. A Hilbert space holds both of these properties.

Definition 1 (Hilbert Space) *A **Hilbert space** $\mathcal{H}$ is an inner product space endowed with a dot product $\langle .,. \rangle$ that is separable*[1] *and complete*[2].

Designing the mapping function Φ explicitly is both tedious and practically impossible for a high-dimensional target space. Alternatively, by exploiting the dot product defined on $\mathcal{H}$, we can design Φ indirectly by a function that maps a pair of points ($\boldsymbol{x}$,$\boldsymbol{x'}$) to their dot product on $\mathcal{H}$. Such a function is called a *kernel*.

Definition 2 (Kernel) *Given an arbitrary set $\mathcal{X}$, a Hilbert space $\mathcal{H}$ and a map $\Phi : \mathcal{X} \to \mathcal{H}$, a **kernel** k is a function that satistifies $k(\boldsymbol{x}, \boldsymbol{x'}) = \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{x'}) \rangle$.*

A direct consequence of this definition is that each kernel has to be symmetric (i.e. $k(\boldsymbol{x}, \boldsymbol{x'}) = k(\boldsymbol{x'}, \boldsymbol{x})$), due to the symmetry of the dot product. Definition 2 allows even infinite-dimensional feature spaces, as will be exemplified in Section 3.1.2.

When kernelizing a model, a central question is how to assure whether the chosen kernel $k : \mathcal{X} \times \mathcal{X} \to \mathcal{R}$ corresponds to a dot product on $\mathcal{H}$, in other words, whether a kernel is *valid*. Two concepts are essential for assuring the validity of a kernel: Gram matrix, and positive semi-definiteness.

Definition 3 (Gram Matrix) *Given a set $\mathcal{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}$ and a kernel k, the corresponding **Gram matrix** is an $N \times N$ matrix with entries*

$$\mathbf{G}_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

for each $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}$.

Definition 4 (Positive Semi-definite Matrix) *An $N \times N$ matrix $\mathbf{X}$ satisfying $\boldsymbol{v}\mathbf{X}\boldsymbol{v'} \geq 0$ for any $N \times 1$ vector $\boldsymbol{v}$ is referred to as a **positive semi-definite matrix**.*

Combining these two concepts, we can perform the validity check using the Mercer's theorem.

Theorem 1 (Mercer's Theorem) *A function $k(\boldsymbol{x}, \boldsymbol{x'}) : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ is a valid kernel if the Gram matrix $\mathbf{G}$ it produces for any finite set of points $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}$ with $\boldsymbol{x}_i \in \mathbb{R}^D$ is symmetric and positive semi-definite [85].*

[1] A vector space $\mathcal{H}$ is separable if and only if it has a countable orthonormal basis.
[2] A vector space $\mathcal{H}$ is complete if every Cauchy sequence of elements in $\mathcal{H}$ converges to an element of $\mathcal{H}$.

A vector space associated with a valid kernel corresponds to a special type of Hilbert space with additional properties, called a Reproducing Kernel Hilbert Space.

Definition 5 (Reproducing Kernel Hilbert Space) *A Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}_k$ of kernel k is a Hilbert space of functions $f : \mathcal{X} \to \mathbb{R}$ with a dot product $\langle ., . \rangle$ and kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ satisfying the following properties:*

1. *k has the reproducing property $\langle f, k(x, .) \rangle = f(x)$ for any $f \in \mathcal{H}_k$, and thus $\langle k(x, .), k(x', .) \rangle = k(x, x')$.*

2. *k spans $\mathcal{H}$ (i.e. $\mathcal{H} = \left\{ f \mid f(.) = \sum_{i=1}^{N} \alpha_i k(x_i, .), \alpha_i \in \mathbb{R} \right\}$) [38, 109].*

According to the reproducing property, any positive definite kernel is represented as a dot product of two functions on the RKHS spanned by the kernel. Hence, a kernel can be treated as a similarity measure for pairs of data points. See [109] for a more thorough discussion.

3.1.2 Example Kernels

Below, the kernels used in this thesis are explained.

Radial Basis Function (RBF) Kernel
The RBF kernel is defined by

$$k_{RBF}(\boldsymbol{x}, \boldsymbol{x'}) = \exp\left(\frac{\|\boldsymbol{x} - \boldsymbol{x'}\|^2}{2\sigma^2}\right)$$

where σ is a hyperparameter referred to as the *length scale*. It determines the smoothness of the decision boundary imposed by the kernel. The larger σ is, the smoother the decision surface is. This kernel is also known as the *Gaussian kernel* since its formula is proportional to the PDF of the normal distribution. It is one of the most frequently used kernels due to its simplicity, interpretability, and ability to capture non-linear boundaries.

Linear Time Warping Kernel
This kernel is used for data sets whose samples are multivariate time series. It uses linear time warping for aligning time series having possibly different lengths. *Alignment* is an element-wise matching between two time series. Matched elements are then passed through an arbitrary

kernel and summed. Formally, a linear time warping kernel is [111]

$$k_{LTW}(\mathbf{X}, \mathbf{V}) = \frac{1}{L} \sum_{k=1}^{L} k(\boldsymbol{x}_{\psi(k)}, \boldsymbol{v}_{\theta(k)}),$$

where $\psi(k) = \lfloor (|\mathbf{X}|/L)k \rfloor$ and $\theta(k) = \lfloor (|\mathbf{V}|/L)k \rfloor$ are the linear time warping functions and L is an arbitrary integer. Any valid kernel can be chosen for $k(.,.)$. In Publication IV, this time-series alignment is preferred rather than more advanced choices, such as [29], in order to keep computations fast enough for real-time use in pervasive setups.

3.2 Support Vector Machines

The support vector machine is a kernelizeable pattern classification algorithm. Since it was introduced [120], it has been used in quite many applications, and shown to be a robust and computationally efficient model that generalizes to unseen data quite well [45, 87].

Let $\mathbf{X} = \{(\boldsymbol{x}_i, y_i) |\ i = 1, \cdots, N\}$ be a labelled data set with N data points, where $\boldsymbol{x}_i$ is a data point and $y_i \in \{-1, +1\}$ is the corresponding label. The support vector machine searches for a hyperplane

$$f(\boldsymbol{x}) = \boldsymbol{w}^T \phi(\boldsymbol{x}) + b,$$

also called the *decision boundary* that separates the two classes. If the classes are linearly separable, SVM chooses the hyperplane that maximizes the distance of the closest sample to the decision boundary given by $y_i f(\boldsymbol{x}_i)/\|\boldsymbol{w}\|$, called the *margin*. The idea is visually illustrated in Figure 3.1. This corresponds to the following optimization problem

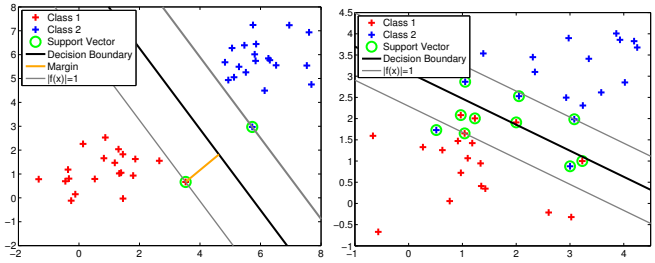
$$\arg\max_{\boldsymbol{w},b} \{ \frac{1}{\|\boldsymbol{w}\|} \min [y_i(\boldsymbol{w}^T \phi(\boldsymbol{x}_i) + b)] \}.$$

Since the distance of a sample to the decision boundary is scale-invariant, we can set $y_i(\boldsymbol{w}^T \phi(\boldsymbol{x}_i) + b) = 1$ in order to convert the above optimization problem to the following quadratic program

$$\begin{aligned} \underset{\boldsymbol{w},b}{\text{minimize}} \ & \{ \frac{1}{2} \|\boldsymbol{w}\|^2 \} \\ \text{s.t.} \ & y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) \geq 1, \ \ \forall i = 1, \cdots, N \end{aligned}$$

which can be solved by off-the-shelf packages.

In order to handle overlapping class distributions, and also to avoid sensitivity to outliers, we modify the above formulation to allow misclassification by introducing a penalty term ζ_i, called a *slack variable*, that



(a) Hard Margin SVM (b) Soft Margin SVM

Figure 3.1. Illustration of the *margin maximization* idea in SVMs. **Left:** Hard margin SVM assumes linearly separable classes. It maximizes the distance of the closest samples (*support vectors*) to the decision boundary. This distance is called the *margin*. Note that the area within the margin boundaries (between the two parallel grey lines) is empty. **Right:** In order to handle linearly unseparable class distributions, soft margin SVM relaxes the above problem to allow data points to be misclassified. For this, a penalty term ζ_i is assigned to each data point $\boldsymbol{x}_i$. This term takes a positive value if $\boldsymbol{x}_i$ lies beyond the margin boundary ($f(\boldsymbol{x}_i) = y_i$).

takes a positive value for every data point $\boldsymbol{x}_i$ lying beyond the margin boundary ($f(\boldsymbol{x}_i) = y_i$). The resulting optimization problem then becomes

$$\begin{aligned} \underset{\boldsymbol{w},b,\boldsymbol{\zeta}}{\text{minimize}} \;\; & \{\frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\zeta_i\} \\ \text{s.t.} \;\; & y_i(\boldsymbol{w}\cdot\boldsymbol{x}_i + b) \geq 1 - \zeta_i, \;\; \forall i = 1, \cdots, n \\ & \zeta_i \geq 0, \;\; \forall i. \end{aligned}$$

Here, C is a hyperparameter that determines a trade-off between the training error and margin size. This corresponds to the bias/variance trade-off between model fit and generalization. The C is set either by cross validation or using a heuristic [17, 22].

The Lagrangian of the constrained optimization problem is

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}\zeta_i - \sum_{i=1}^{N}\alpha_i[y_i f(\boldsymbol{x}_i) - 1 + \zeta_i] - \sum_{i=1}^{N}\beta_i\zeta_i$$

where $\boldsymbol{\alpha} = [\alpha_i, \cdots, \alpha_N]$ and $\boldsymbol{\beta} = [\beta_i, \cdots, \beta_N]$ are Lagrange multipliers. This formulation of the problem is not kernelizeable, since the input data $\boldsymbol{x}_i$ does not appear in the dot-product form. If we convert the Lagrangian into its dual form by setting its gradient to zero and expressing all vari-

ables in terms of the Lagrange multipliers $\boldsymbol{\alpha}$, we have

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\text{maximize}} \quad & \left\{ \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i}^{N} \sum_{j}^{N} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j \right\} \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq C, \ \ \forall i = 1, \cdots, N \\ & \sum_{i=1}^{N} \alpha_i y_i = 0 \end{aligned} \tag{3.1}$$

where the data points appear in dot products ($\boldsymbol{x}_i^T \boldsymbol{x}_j$). This enables us to replace the dot product with a kernel $k(\boldsymbol{x}, \boldsymbol{x'})$. This dual form comes with the following set of conditions:

$$\begin{aligned} \alpha_i &\geq 0, \\ y_i f(\boldsymbol{x}_i) - 1 + \zeta_i &\geq 0, \\ \alpha_i (y_i f(\boldsymbol{x}_i) - 1 + \zeta_i) &= 0, \\ \beta_i &\geq 0, \\ \zeta_i &\geq 0, \\ \beta_i \zeta_i &= 0, \end{aligned}$$

which are known as the Karush-Kuhn-Tucker (KKT) conditions. The solution of the quadratic problem in Equation 3.1 can be speeded up using the *sequential minimal optimization* (SMO) [96] algorithm, which analytically solves a pair of Lagrange multipliers at a time, iterating over different pairs.

Once the model is trained, the output of a new sample can be predicted by

$$f(\boldsymbol{x}^*) = \sum_{i=1}^{N} \alpha_i y_i k(\boldsymbol{x}_i, \boldsymbol{x}^*) + b.$$

From the first three KKT conditions, we see that $\alpha_i = 0$ for samples lying in the correct margin. Hence, they do not contribute to prediction. Thus, it is sufficient to store only those training data points that are outside the correct margin. These data points are called *support vectors*. This sparsity property allows fast prediction in SVMs.

The SVM is essentially formulated for binary classification. Its extension to multiclass classification can be done in multiple ways. Amongst the most common ways are:

- **OVA (One-versus-all):** For a K class classification problem, K SVMs are trained. For each class, a separate SVM discriminates that class

from all others. In prediction, the new sample is assigned to the class for which the value of the decision function $f_k(\boldsymbol{x}^*)$ is the largest (winner-take-all).

- **AVA (All-versus-all):** A separate SVM is trained to discriminate each pair of classes. Hence, $K(K-1)/2$ SVMs are trained in total. In prediction, the choice of each classifier is counted as one vote, and the sample assigned to the class having the highest votes.

- **Unified optimization:** The optimization problem is reformulated so that K one-versus-all classifiers are jointly trained [129]. This strategy has been shown to result in poor computational performance due to the increased complexity of the optimization problem.

In Publication III and Publication VI the OVA approach is adopted for multiclass classification with SVMs due to its low computational demand.

3.3 Relevance Vector Machines

While being a robust and effective method, SVM has some disadvantages, such as the difficulties in fixing the hyperparameter C, not providing a probabilistic interpretation of predictions, and requiring a positive semi-definite kernel. The Relevance Vector Machine (RVM) is a probabilistic model, inspired from SVM, which is introduced to eliminate these shortcomings [117].

RVM was originally introduced as a Bayesian linear regression model with the likelihood function

$$p(y|\mathbf{X}, \boldsymbol{w}, \sigma^2) = \mathcal{N}(y|f(\boldsymbol{x}), \sigma^2) = \mathcal{N}(y|\sum_{i=1}^{N} w_i k(\boldsymbol{x}_i, \boldsymbol{x}) + b, \sigma^2).$$

where $k(.,.)$ is a kernel. This likelihood function essentially corresponds to the noisy version of the SVM prediction function. In order to impose sparsity to data points similarly as SVM, an *automatic relevance determination* (ARD) prior is placed on the weights

$$p(\boldsymbol{w}|\boldsymbol{\alpha}) = \prod_{i=1}^{N} \mathcal{N}(w_i|0, {\alpha_i}^{-1})$$

where $\boldsymbol{\alpha} = [\alpha_i, \cdots, \alpha_N]$ denote the vector of hyperparameters. The poste-

rior of $\boldsymbol{w}$ then becomes

$$p(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{\alpha}, \sigma^2) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

with $\boldsymbol{\Sigma} = (\sigma^{-2}\boldsymbol{\Phi}^T\boldsymbol{\Phi}+\mathbf{A})^{-1}$ and $\boldsymbol{\mu} = \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{\Phi}^T\boldsymbol{y}$ where $\mathbf{A} = \text{diag}(\alpha_0, \alpha_1, \cdots, \alpha_N)$. Here, $\boldsymbol{\Phi}$ is an $N \times (N+1)$ matrix with $\boldsymbol{\Phi}_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_{j-1})$ and $\boldsymbol{\Phi}_{i1}$=1.

Since incorporating hyperpriors over $\boldsymbol{\alpha}$ and σ^2 would make computations complicated, hyperparameters are typically learned by type II maximum likelihood [79], as described in Section 2.6. When the weights $\boldsymbol{w}$ are integrated out, the resulting marginal likelihood is

$$p(\boldsymbol{y}|\boldsymbol{\alpha}, \sigma^2) = (2\pi)^{-N/2}|\sigma^{\mathbf{2}}\mathbf{I}_N + \boldsymbol{\Phi}^T\boldsymbol{A}^{-1}\boldsymbol{\Phi}|^{-1/2}\exp\{-\frac{1}{2}\boldsymbol{y}^T(\sigma^{\mathbf{2}}\mathbf{I}_N + \boldsymbol{\Phi}\boldsymbol{A}^{-1}\boldsymbol{\Phi}^T)^{-1}\boldsymbol{y}\}.$$

If we set the derivative of the marginal likelihood to zero, we get

$$\alpha_i{}^{new} = \frac{\gamma_i}{\mu_i{}^2},$$

$$(\sigma^2)^{new} = \|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\mu}\|^2/(N - \sum_i^N \gamma_i)$$

with $\gamma_i = 1 - \alpha_i\Sigma_{ii}$. The RVM is trained iteratively; in each iteration, the posterior of the weights is calculated based on the newest values of the hyperparameters, then the hyperparameters are re-calculated based on the new posterior. This procedure is repeated until convergence. At the end of the iterations, most of the weights are forced towards an infinite peak at zero by very large α_i values. Hence, the corresponding kernels are pruned from the model. The remaining active data points having nonzero weights are called *relevance vectors*, analogously to the support vectors in SVM.

RVM usually finds sparser solutions than SVM [117], enabling faster prediction. It also learns all model hyperparameters in a single run. Meanwhile, RVM and SVM are comparable in terms of generalization error. RVM can be used for classification as well by passing the regression output through a sigmoid function. Its application to multiclass case is straightforward, unlike SVM. RVM is used as a baseline model in Publication V for predicting auditory attention from biosignals.

3.4 Gaussian Processes

Gaussian processes (GPs) are stochastic processes such that any finite set of samples are distributed as multivariate normal. A Gaussian process is specified by a mean function $m(\boldsymbol{x})$ and a covariance function $k(\boldsymbol{x}, \boldsymbol{x'})$

[102],

$$f(\boldsymbol{x}) = \mathcal{GP}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x'})).$$

The Gaussian process serves as a prior over functions mapping a set of variates to a real-valued output. Given any two input points $\boldsymbol{x}$, $\boldsymbol{x'}$, and assuming that the data are centred (i.e. $m(\boldsymbol{x}) = 0$), the GP prior on the corresponding outputs f, f' is the normal distribution

$$\mathcal{N}\left(\mathbf{0}, \begin{bmatrix} k(\boldsymbol{x}, \boldsymbol{x}) & k(\boldsymbol{x}, \boldsymbol{x'}) \\ k(\boldsymbol{x'}, \boldsymbol{x}) & k(\boldsymbol{x'}, \boldsymbol{x'}) \end{bmatrix}\right).$$

A main structural difference of the Gaussian process from the models introduced so far is that it is a non-parametric model. One way of deriving the Gaussian process is to integrate out the model parameters (weights) of Bayesian linear regression. Hence, the Gaussian process can be thought of as a distribution over functions that are not restricted to a finite parametric set.

3.4.1 Regression

The Gaussian process is applied to regression along with a noise model (also called the likelihood) that takes into account the noise in observations. A desirable choice is a Gaussian noise model which ensures closed-form calculation of the predictive distribution: $p(y|f) = \mathcal{N}(f, \sigma_n^2)$. The joint distribution of the output of the training data and that of a new sample $\boldsymbol{x}^*$ then becomes

$$\begin{bmatrix} \boldsymbol{y} \\ f^* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} k(\mathbf{X}, \mathbf{X}) + \sigma^2 \boldsymbol{I} & k(\mathbf{X}, \boldsymbol{x}^*) \\ k(\boldsymbol{x}^*, \mathbf{X}) & k(\boldsymbol{x}^*, \boldsymbol{x}^*) \end{bmatrix}\right).$$

The predictive distribution can be analytically computed based on standard properties of the normal distribution

$$p(y^*|\mathbf{X}, \boldsymbol{y}, \boldsymbol{x}^*) = \mathcal{N}(\boldsymbol{\mu_p}, \boldsymbol{\Sigma}_p) \tag{3.2}$$

with

$$\begin{aligned} \boldsymbol{\mu_p} &= k(\boldsymbol{x}^*, \mathbf{X})[k(\mathbf{X}, \mathbf{X}) + {\sigma_n}^2]^{-1}\boldsymbol{y}, \\ \boldsymbol{\Sigma}_p &= k(\boldsymbol{x}^*, \mathbf{X})[k(\mathbf{X}, \mathbf{X}) + {\sigma_n}^2]^{-1}k(\mathbf{X}, \boldsymbol{x}^*). \end{aligned}$$

The kernel parameters (if there are any) have a significant effect on model performance. They can be tuned using the type II maximum likelihood method. Let $\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \cdots, \gamma_N]$ denote the vector of kernel parameters, the log marginal likelihood of the model is

$$\log p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{\gamma}) = -\frac{1}{2}\boldsymbol{y}^T\mathbf{K}\boldsymbol{y} - \frac{1}{2}\log|\mathbf{K}| - \frac{N}{2}\log 2\pi.$$

The partial derivative of the marginal likelihood with respect to each hyperparameter is given by [102]

$$\frac{\partial \log p(\boldsymbol{y}|\mathbf{X},\boldsymbol{\gamma})}{\partial \gamma_i} = \frac{1}{2}\boldsymbol{y}^T\mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \gamma_i}\mathbf{K}^{-1}\boldsymbol{y} - \frac{1}{2}\mathrm{tr}\left(\mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \gamma_i}\right).$$

Based on these equations, the hyperparameter values that maximize the marginal likelihood can efficiently be found by a gradient-based optimizer.

Gaussian process regression is used in Publication V as a supervised predictor of the level of user's auditory attention from biosignals.

3.4.2 Classification

The GP can be applied to classification problems by passing its prediction output through a sigmoid function. We assume a latent decision function $\boldsymbol{f} \in \mathbb{R}$, and place a GP prior on it. For binary class labels $\{-1, +1\}$, the sign of f gives the predicted class and its magnitude gives the confidence of our prediction, similarly to $y(\boldsymbol{x})$ in SVMs. Given a test sample $\boldsymbol{x}^*$, its class y^* is predicted by

$$p(y^* = +1|\mathbf{X}, \boldsymbol{y}, \boldsymbol{x}^*) = \int p(y^* = +1|f^*)\; p(f^*|\mathbf{X}, \boldsymbol{y}, \boldsymbol{x}^*)df^*$$

where

$$p(f^*|\mathbf{X}, \boldsymbol{y}, \boldsymbol{x}^*) = \int p(f^*|\mathbf{X}, \boldsymbol{x}^*, \boldsymbol{f})\; p(\boldsymbol{f}|\mathbf{X}, \boldsymbol{y})d\boldsymbol{f}. \tag{3.3}$$

The predictive distribution of the latent function $p(f^*|\mathbf{X}, \boldsymbol{x}^*, \boldsymbol{f})$ is identical to GP prediction (Equation 3.2). By Bayes' theorem, the posterior of the latent function is

$$p(\boldsymbol{f}|\mathbf{X}, \boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{f})p(\boldsymbol{f}|\boldsymbol{X})}{p(\boldsymbol{y}|\mathbf{X})}.$$

Here, $\boldsymbol{f}|\boldsymbol{X} \sim \mathcal{GP}(\mathbf{0}, \mathbf{K})$ is the Gaussian process prior. The entries of the covariance matrix $\mathbf{K}$ are calculated by applying the kernel $k(\boldsymbol{x}, \boldsymbol{x'})$ on each pair of samples. The value of the latent function is converted to the posterior class probability by a sigmoid likelihood function $p(y|f) = \sigma(f)$. Some possible sigmoid functions are the logistic function $\sigma(f) = 1/(1+\exp(-f))$ and the probit function $\Phi(z) = \int_{-\infty}^{z} \mathcal{N}(\tau|0,1)d\tau$. It is not necessary to calculate the marginal likelihood $p(\boldsymbol{y}|\mathbf{X})$ explicitly in model inference. Since it does not depend on $\boldsymbol{f}$, it appears as a constant during inference. However, it is useful in learning hyperparameters, as for GP regression.

GP classification is used in Publication IV to predict the relevance of real-world objects from eye movement patterns. It is preferred over SVM for the probabilistic interpretation of its outputs. The probability

of an object being relevant $p(y = +1|\boldsymbol{x})$ is taken as a scalar measure of relevance.

GP classification does not have an analytical solution since the non-linear likelihood function makes the integral in Equation 3.3 intractable. Hence, approximation methods are used for inference, such as Laplace approximation [130], expectation propagation [86], and MCMC sampling [118]. In Publication IV, the Laplace method is applied which approximates $p(\boldsymbol{f}|\mathbf{X}, \boldsymbol{y})$ by a normal distribution.

4. Multi-view Learning

In many applications, multiple data sources coming from the same process are available. An example is a data base of images with textual annotations. For each image, we have both pixel values and annotations explaining the same scene in the image in different ways. Learning models that relate the co-occurring samples of multiple data sources is referred to as *multi-view learning*. Another application of multi-view learning is what is focused on in this thesis: integrating multiple biosignals gathered from a computer user for inferring her affective state.

In this thesis, two multi-view learning approaches are used. In the first approach, the data sources are integrated into a single model by designing a similarity measure for each data source and combining these measures. Finding the best way of combining the measures is the central machine learning problem. In the second approach, dependencies between data sources are analyzed. To this end, data sources are projected to a new space, which is chosen to be the one that best reveals their dependencies on each other.

4.1 Multiple kernel learning

Mercer's Theorem [85] assures that combining valid kernels using elementary operations, such as addition, and element-wise multiplication produces a valid kernel. This property gives us the opportunity to design complex kernels from simple ones. Combining kernels brings the following benefits:

- More complex kernels can capture more complex properties of data, which results in improved model performance.

- We can perform multi-view learning using any kernel machine by assigning a kernel to each available data source. This both enables integrating data sources with incompatible representations, and provides the model the flexibility of capturing distinct properties of each data source.

The multiple kernel learning (MKL) technique can be applied to any kernelizeable model. The machine learning model on which MKL is performed is called the *base learner* [45].

Kernel combination strategies can be put roughly into three categories [45]:

1. **Combination by fixed rules:** Pre-determined fixed rules are applied for combination, such as $\mathbf{K} = \mathbf{K}_1 + \mathbf{K}_2$ or $\mathbf{K} = \mathbf{K}_1 \otimes \mathbf{K}_2$ [110]. This approach is used for designing *pairwise kernels*, where the idea is to define rules of similarity between two pairs of samples $(\boldsymbol{x}_i^a, \boldsymbol{x}_j^a)$, $(\boldsymbol{x}_i^b, \boldsymbol{x}_j^b)$ coming from views a and b. An intuitive rule would be to sum the cross-similarities of the pairs

$$k^P(\{\boldsymbol{x}_i^a, \boldsymbol{x}_j^a\}, \{\boldsymbol{x}_i^b, \boldsymbol{x}_j^b\}) = k(\boldsymbol{x}_i^a, \boldsymbol{x}_i^b)k(\boldsymbol{x}_j^a, \boldsymbol{x}_j^b) + k(\boldsymbol{x}_i^a, \boldsymbol{x}_j^b)k(\boldsymbol{x}_i^a, \boldsymbol{x}_j^b)$$

which is called the *genomic kernel*. This kernel has been shown to be quite useful in bioinformatics applications [12].

2. **Heuristic combination:** The combined kernel is calculated based on heuristics. An example is a linear combination of P kernels [115]

$$\boldsymbol{K}^P = \sum_{p=1}^{P} w_p \mathbf{K}_p$$

where $\boldsymbol{w} = [w_1, \cdots, w_P]$ is the vector of kernel weights, which are determined by the following heuristic

$$w_p = \frac{\pi_p - \delta}{\sum_{i=1}^{P}(\pi_i - \delta)}.$$

Here, π_p is the accuracy obtained by training the base learner on kernel $\mathbf{K}_p$ only, and δ is a hyperparameter in the range $[0, \min\{\pi_1, \cdots, \pi_P\}]$ serving as a threshold.

3. **Combination by optimization:** The optimal combination is learned from data by optimizing the cost function of the base learner with respect to the combination parameters. If we assume a weighted linear

sum of the views, we can rewrite the SVM primal optimization problem as [100]

$$
\begin{aligned}
J(\boldsymbol{q}) = \underset{\boldsymbol{w}_p, b, \boldsymbol{\zeta}}{\text{minimize}} \quad & \left\{ \frac{1}{2} \sum_{p=1}^{P} \frac{1}{q_p} \|\boldsymbol{w}_p\|^2 + C \sum_{i=1}^{n} \zeta_i \right\} \\
\text{s.t.} \quad & y_i \sum_{p=1}^{P} (\boldsymbol{w}_p^T \cdot \Phi(\boldsymbol{x}_i)) + b \geq 1 - \zeta_i, \quad \forall i = 1, \cdots, n \\
& \zeta_i \geq 0, \quad \forall i \\
& \sum_{p=1}^{P} q_p = 1.
\end{aligned}
$$

The dual of this problem is

$$
\begin{aligned}
J(\boldsymbol{q}) = \underset{\boldsymbol{\alpha}}{\text{maximize}} \quad & \{ \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i}^{n} \sum_{j}^{n} \alpha_i \alpha_j y_i y_j \left(\sum_{p=1}^{P} q_p k_p(\boldsymbol{x}_i^p, \boldsymbol{x}_j^p) \right) \} \\
\text{s.t.} \quad & 0 \leq \alpha_i \leq C, \quad \forall i = 1, \cdots, n \\
& \sum_{i=1}^{n} \alpha_i y_i = 0.
\end{aligned}
$$

All of the parameters, including the kernel weights $\boldsymbol{q} = [q_1, q_2, \cdots, q_P]$, can be efficiently optimized in an iterative procedure consisting of two steps. First, the kernel weights are kept constant, and the model parameters $\boldsymbol{\alpha}$ are learned in the same way as standard SVM. Then, the model parameters are kept constant and the kernel weights are learned by gradient descent using the gradient of the SVM cost function with respect to $\boldsymbol{q}$

$$
\frac{\partial J(\boldsymbol{q})}{\partial q_p} = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j k_p(\boldsymbol{x}_i^p, \boldsymbol{x}_j^p), \quad \forall p = 1, \cdots, P.
$$

This procedure is repeated until convergence. In Publication III and Publication VI, this strategy is extended to a multi-task learning method, which will be detailed in Chapter 5.

The MKL technique is used in Publication III and Publication VI for incorporating signals coming from different biosensors, motivated by the assumption that each biosensor has different signal characteristics that can be better captured by an individual kernel. Furthermore, a multi-task learning method is introduced that enables information transfer across tasks by enforcing correlated tasks to have similar kernel weights. See Chapter 5 for further details.

4.2 Modeling correlations between views

4.2.1 Canonical correlation analysis

Canonical correlation analysis (CCA) is a factor analysis method for analyzing dependencies between co-occurring data sets. CCA searches for linear projections $\boldsymbol{u}$ and $\boldsymbol{v}$ that maximize the correlation between data sets $\mathbf{X} = [\boldsymbol{x}_1; \boldsymbol{x}_2; \cdots; \boldsymbol{x}_N]$ and $\mathbf{Y} = [\boldsymbol{y}_1; \boldsymbol{y}_2; \cdots; \boldsymbol{y}_N]$ [53]

$$\rho = \underset{\boldsymbol{u},\boldsymbol{v}}{\text{maximize}} \ \text{cor}(\boldsymbol{u}^T\mathbf{X}, \boldsymbol{v}^T\mathbf{Y}). \tag{4.1}$$

A common measure of correlation is Pearson's correlation [40, 92], which is defined as the covariance of two random variables normalized by the product of their standard deviations

$$\rho_{xy} = \text{cor}(x, y)\frac{\mathbb{E}[(x - \mathbb{E}[x])(y - \mathbb{E}[y])]}{\sqrt{\mathbb{E}[(x - \mathbb{E}[x])^2]}\sqrt{\mathbb{E}[(y - \mathbb{E}[y])^2]}}. \tag{4.2}$$

This normalization makes the magnitude of the covariance interpretable. Correlation can take values within $[-1, 1]$ due to the Cauchy-Schwarz inequality. The sign is negative when the two variables have an inverse linear relationship, and positive when the relationship is direct linear. The magnitude tells how strong the relationship is, in other words, how close it is to a perfect line.

Plugging this measure of correlation into Equation 4.1 gives

$$\rho = \underset{\boldsymbol{u},\boldsymbol{v}}{\text{maximize}} \ \frac{\boldsymbol{u}^T\Sigma_{12}\boldsymbol{v}}{\sqrt{\boldsymbol{u}^T\Sigma_{11}\boldsymbol{u}}\sqrt{\boldsymbol{v}^T\Sigma_{22}\boldsymbol{v}}}$$

where $\Sigma_{11} = \text{cov}(\mathbf{X})$, $\Sigma_{22} = \text{cov}(\mathbf{Y})$, and $\Sigma_{12} = \text{cov}(\mathbf{X}, \mathbf{Y})$. The analytical solutions for $\boldsymbol{u}$ and $\boldsymbol{v}$ are given by the principal eigenvectors of

$$\Sigma_{11}{}^{-1/2}\Sigma_{12}\Sigma_{22}{}^{-1/2}\Sigma_{21} \tag{4.3}$$

and

$$\Sigma_{22}{}^{-1/2}\Sigma_{21}\Sigma_{11}{}^{-1}\Sigma_{12}, \tag{4.4}$$

respectively. The corresponding eigenvalue ρ, which is common in both eigendecomposition problems, is the maximized correlation of the univariate projections of the two data sets. Once either of the two vectors is known, the other can be calculated using

$$\boldsymbol{v} = \frac{\Sigma_{22}{}^{-1/2}\Sigma_{21}}{\rho}\boldsymbol{u},$$
$$\boldsymbol{u} = \frac{\Sigma_{11}{}^{-1}\Sigma_{12}}{\rho}\boldsymbol{v}.$$

The data sets $\mathbf{X}$ and $\mathbf{Y}$ can be projected to a latent space with $R = \min(D1, D2)$ dimensions by a projection matrix formed by the eigenvectors of matrices (4.3) and (4.4) corresponding to the highest R eigenvalues. The projections are orthogonal since (4.3) and (4.4) are symmetric matrices, resulting in projected variates $\mathbf{U} = [\boldsymbol{U}_1, \cdots, \boldsymbol{U}_R]$ and $\mathbf{V} = [\boldsymbol{V}_1, \cdots, \boldsymbol{V}_R]$ that are uncorrelated with each other both within and between data sets [60]

$$\begin{aligned}
\text{cor}(\boldsymbol{U}_k, \boldsymbol{U}_l) &= 0, k \neq l, \\
\text{cor}(\boldsymbol{V}_k, \boldsymbol{V}_l) &= 0, k \neq l, \\
\text{cor}(\boldsymbol{U}_k, \boldsymbol{V}_l) &= 0, k \neq l.
\end{aligned}$$

CCA can be used for measuring the overall dependence between data sets. One possible measure is the highest correlation between the univariate projections, which corresponds to the largest eigenvalue of (4.3) or (4.4). A more robust measure that takes into account multiple projections is the *mutual information* [20]

$$I(\mathbf{X}, \mathbf{Y}) = -\frac{1}{2} \sum_{i=1}^{R} \log(1 - \rho_i^2).$$

Note that this is a measure of mutual information for views that are jointly distributed as multivariate normal.

Classical CCA is fast, easy-to-implement, and provides the global maximum. One disadvantage of the classical CCA is that it overfits badly for high dimensions if the number of samples is close to the dimensionality [65]. Regularization techniques such as adding a value to the diagonals of the covariance matrices [124]

$$\text{cov}_{reg}(\boldsymbol{X}_i) = \text{cov}(\boldsymbol{X}_i) + \lambda_i \mathbf{I}$$

in a ridge regression fashion partly solve this problem. An alternative solution is adopted in this thesis that brings benefits additional to preventing overfitting. The problem is formulated within the Bayesian framework, as will be discuss in the next section. This way, a probabilistic interpretation of the outcome is obtained, which enables extensions to more complex models with little effort, as was done in [2, 39, 55, 123].

CCA has been shown to be useful in a wide range of applications, such as detecting mental task switches from the correlation of signals in different parts of the brain [131], clustering samples coming from multiple data sources [23], and combining histological images and additional

information into a unified space where biochemical recurrence of cancer patients are more predictable [44]. Classical CCA is used in Publication V to predict the user's level of auditory attention to an audio stimulus. The prediction is performed in an unsupervised manner from the correlation between biosignals and audio.

4.2.2 Bayesian canonical correlation analysis

CCA can also be formulated as minimization of the distance between the views projected into a new orthogonal space [49]

$$\underset{\boldsymbol{u},\boldsymbol{v}}{\text{minimize}} \ \|\boldsymbol{u}^T\mathbf{X} - \boldsymbol{v}^T\mathbf{Y}\|_F.$$

This property inspires its probabilistic interpretation to have the structure that the samples come from a common latent space consisting of uncorrelated variates. The latent samples are then projected to separate observation spaces. Gaussian additive noise is assumed in all spaces, retaining the Gaussianity assumption of the classical CCA formulation. The generative process of probabilistic CCA is [6]

$$\begin{aligned} z &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \\ \boldsymbol{x} &\sim \mathcal{N}(\mathbf{W}_1\boldsymbol{z}, \boldsymbol{\Psi}_1), \\ \boldsymbol{y} &\sim \mathcal{N}(\mathbf{W}_2\boldsymbol{z}, \boldsymbol{\Psi}_2) \end{aligned} \tag{4.5}$$

where $\boldsymbol{z} \in \mathbb{R}^{R\times 1}$ is the R-dimensional latent representation of the sample, $\mathbf{W}_1 \in \mathbb{R}^{D_1\times R}$ and $\mathbf{W}_2 \in \mathbb{R}^{D_2\times R}$ are projection matrices from the latent space to the observation space. The view-specific variation is incorporated by the noise covariance matrices $\boldsymbol{\Psi}_1$ and $\boldsymbol{\Psi}_2$. Assuming a full covariance sharply increases the number of parameters in the model, which results in a high risk of overfitting and makes inference very hard especially when the data are high-dimensional.

We can solve this problem by incorporating priors that induce sparse projection vectors [125], leading to a Bayesian CCA (BCCA) formulation. By row-wise concatenating the views, we can formulate the model as Bayesian factor analysis [14] with a groupwise sparsity prior on the projection weights

$$\begin{aligned} \boldsymbol{z} &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \\ [\mathbf{x}_1; \mathbf{x}_2] &\sim \mathcal{N}(\mathbf{W}\mathbf{z}, \boldsymbol{\Sigma}) \end{aligned}$$

where

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \sigma_2^2\mathbf{I} \end{bmatrix}$$

is a block diagonal matrix, and the concatenated projection matrix

$$p(\mathbf{W}) = \prod_{r=1}^{R} \left[\mathcal{N}(\mathbf{W}_1(r)|\mathbf{0}, \beta_{1r}^{-1}\mathbf{I})\, \mathcal{N}(\mathbf{W}_2(r)|\mathbf{0}, \beta_{2r}^{-1}\mathbf{I})\right]$$

is assigned the priors $\boldsymbol{\beta}_1 = [\beta_{1r}, \cdots, \beta_{1R}]$ and $\boldsymbol{\beta}_2 = [\beta_{2r}, \cdots, \beta_{2R}]$ with

$$\beta_{1r} \sim \mathcal{G}(\alpha_0, \beta_0),$$
$$\beta_{2r} \sim \mathcal{G}(\alpha_0, \beta_0).$$

Setting the hyperparameters to very small values ($\alpha_0 = \beta_0 = 10^{-14}$), we obtain flat Gamma distributions imposing $\mathbf{W}$ to have the block structure

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_1 & \mathbf{V}_1 & \mathbf{0} \\ \mathbf{W}_2 & \mathbf{0} & \mathbf{V}_2 \end{bmatrix}$$

where the columns of $[\mathbf{W_1}; \mathbf{W_2}]$ denote the latent components shared by both views, and the columns of $\mathbf{V}_1$ and $\mathbf{V}_2$ show the ones specific to a view. This is the well-known *automatic relevance determination (ARD)* technique applied separately to the projection weights of each view, in a similar fashion to Group-lasso [132]. With such a formulation, analyzing the shared and source-specific variance becomes very easy, unlike previous Bayesian formulations of CCA, such as [2, 65, 98, 126]. Marginalizing the source-specific latent components out, we get a model equivalent to (4.5) with the low-rank noise covariances

$$\boldsymbol{\Psi}_1 = \mathbf{V}_1\mathbf{V}_1^T + \sigma_1^2\mathbf{I},$$
$$\boldsymbol{\Psi}_2 = \mathbf{V}_2\mathbf{V}_2^T + \sigma_2^2\mathbf{I}$$

for the two views. This way, the ARD technique learns the rank of the noise covariances automatically from data, instead of requiring them to be explicitly specified. This Bayesian CCA formulation is called *Group Factor Analysis (GFA)* [125]. The inference of this model is shown to be reasonably fast using a mean-field variational approximation procedure. See [125] for the details.

In GFA, given a set of paired samples from two views $\mathbf{X}^*$ and $\mathbf{Y}^*$, the correlation between the views on the learned latent space can be estimated by $\text{cor}(\mathbb{E}[\boldsymbol{z}|\mathbf{X}^*], \mathbb{E}[\boldsymbol{z}|\mathbf{Y}^*])$. This estimate is adopted for calculating the correlation between biosignals and audio in Publication V.

4.2.3 Time-dependent Bayesian canonical correlation analysis

Classical CCA and Bayesian CCA both assume that the observed data are independent and identically distributed (iid). However, in many cases, the data are time-dependent (i.e. a sample at any time point is dependent on samples at previous time points). In Publication V, a new variant of Bayesian CCA is introduced that captures the time dependence in the data. In particular, CCA is extended to a state-space model by setting the prior of the latent sample representation to

$$\begin{aligned}\mathbf{z}_0 &\sim N(\mathbf{0}, \mathbf{I}),\\ \mathbf{z}_t &\sim N(\mathbf{T}\mathbf{z}_{t-1}, \sigma_0^2\mathbf{I})\end{aligned}$$

where $\mathbf{T}$ is the transition matrix that governs the trend in the state-space and σ_0^2 incorporates an amount of additive noise to this space. This model is named as *time-dependent Bayesian CCA (T-BCCA*).

The inference of T-BCCA is very similar to BCCA. All variational update equations of BCCA are applicable here except the one for $\mathbf{Z} = [\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_N]$. The time-dependent latent representations can be updated in a forward-backward fashion, as described in Algorithm 1, which is taken from [10]. In the forward pass, $\mathbf{Z}$ is estimated by a Kalman filter learned on the current estimates of $\mathbf{W}_1$, $\mathbf{W}_2$, σ_1^2, and σ_1^2. Then these estimates are corrected by the Rauch-Tung-Striebel smoother [46] in the backward pass. The update rule given in [10] for the transition matrix $\mathbf{T}$ is also reusable here

$$\langle \mathbf{T} \rangle = \left[\sum_{t=2}^{N} \left[\mathbf{A}_t \langle \mathbf{C}_t \rangle + \langle \boldsymbol{z}_{t-1} \rangle \langle \boldsymbol{z}_t^T \rangle \right] \right] \left[\sum_{t=2}^{N} \left[\langle \mathbf{C}_t \rangle + \langle \boldsymbol{z}_t \rangle \langle \boldsymbol{z}_t^T \rangle \right] \right]^{-1}.$$

The advantage of T-BCCA is illustrated on simulated data by comparing it with BCCA. The $\mathbf{T}$ is restricted to be diagonal, imposing independence across latent components, as in BCCA. The σ_0^2 is set to 1 for simplicity. The simulated data are generated from three latent components, two of which are heavily time-dependent and one is white noise with large variation. The true latent components, and their estimates by BCCA and T-BCCA are given in Figure 4.1. As seen in the figure, T-BCCA estimates the time-dependent components more accurately.

Algorithm 1 The two-pass variational update rule for the latent representations $\boldsymbol{z}_i$, as suggested in [10]. Here, $\langle \mathbf{B} \rangle = [\langle \mathbf{W}_1 \rangle; \langle \mathbf{W}_2 \rangle; \mathbf{I}; \mathbf{U}_b]$ where $\mathbf{U}_b^T \mathbf{U}_b = \sum_{i=1,2} [1/\langle \sigma_i^2 \rangle \ \langle \mathbf{W}_i^T \mathbf{W}_i \rangle - 1/\langle \sigma_i^2 \rangle \ \langle \mathbf{W}_i^T \rangle \langle \mathbf{W}_i \rangle]$. The forward pass calculates the new estimates of the latent representations $\{\langle \boldsymbol{z}_1 \rangle, \langle \boldsymbol{z}_2 \rangle, \cdots, \langle \boldsymbol{z}_N \rangle\}$, their individual covariances $\{\langle \boldsymbol{C}_1 \rangle, \langle \boldsymbol{C}_2 \rangle, \cdots, \langle \boldsymbol{C}_N \rangle\}$, and a set of temporary matrices $\{\boldsymbol{Q}_1, \boldsymbol{Q}_2, \cdots, \boldsymbol{Q}_N\}$. The backward pass takes these values as input, applies the Rauch-Tung-Striebel smoother [46], and outputs the corrected $\langle \boldsymbol{z}_i \rangle$'s and $\langle \boldsymbol{C}_i \rangle$'s which serve as the variational estimates for the current iteration.

procedure FORWARD
  $\mathbf{L} \leftarrow \mathbf{I}$, $\mathbf{Q}_1 \leftarrow \mathbf{I}$, $\boldsymbol{m}_1 \leftarrow \mathbf{0}$
  $\mathbf{K} \leftarrow \mathbf{L}\langle \mathbf{B}^T \rangle (\langle \mathbf{B} \rangle \mathbf{L} \langle \mathbf{B}^T \rangle + \langle \mathbf{\Sigma} \rangle)^{-1}$
  $\langle \mathbf{C}_1 \rangle \leftarrow (\mathbf{I} - \mathbf{K}\langle \mathbf{B} \rangle)\mathbf{L}$
  $\langle \boldsymbol{z}_1 \rangle \leftarrow \boldsymbol{m}_1 + \mathbf{K}([\boldsymbol{x}_t; \boldsymbol{y}_t] - \langle \mathbf{B} \rangle \boldsymbol{m}_1)$
  for $t \leftarrow 2, N$ **do**
    $\mathbf{Q}_t \leftarrow \langle \mathbf{T} \rangle \langle \mathbf{C}_{t-1} \rangle \langle \mathbf{T}^T \rangle + \mathbf{I}$
    $\mathbf{L} \leftarrow \mathbf{Q}_t$
    $\boldsymbol{m}_t \leftarrow \langle \mathbf{T} \rangle \langle \boldsymbol{z}_{t-1} \rangle$
    $\mathbf{K} \leftarrow \mathbf{L}\langle \mathbf{B} \rangle^T (\langle \mathbf{B} \rangle \mathbf{L} \langle \mathbf{B} \rangle^T + \langle \mathbf{\Sigma} \rangle)^{-1}$
    $\langle \mathbf{C}_t \rangle \leftarrow (\mathbf{I} - \mathbf{K}\langle \mathbf{B} \rangle)\mathbf{L}$
    $\langle \boldsymbol{z}_t \rangle \leftarrow \boldsymbol{m}_t + \mathbf{K}([\boldsymbol{x}_t; \boldsymbol{y}_t] - \langle \mathbf{B} \rangle \boldsymbol{m}_t)$
  end for
end procedure
procedure BACKWARD
  for $t \leftarrow N - 1, 1$ **do**
    $\mathbf{A}_t \leftarrow \langle \mathbf{C}_t \rangle \mathbf{A}^T (\mathbf{Q}_{t+1})^{-1}$
    $\langle \mathbf{C}_t \rangle \leftarrow \langle \mathbf{C}_t \rangle + \mathbf{A}_t (\langle \mathbf{C}_{t+1} \rangle - \mathbf{Q}_{t+1}) \mathbf{A}_t^T$
    $\langle \boldsymbol{z}_t \rangle \leftarrow \langle \boldsymbol{z}_t \rangle + \mathbf{A}_t (\langle \boldsymbol{z}_{t+1} \rangle - \langle \mathbf{T} \rangle \langle \boldsymbol{z}_t \rangle)$
  end for
end procedure

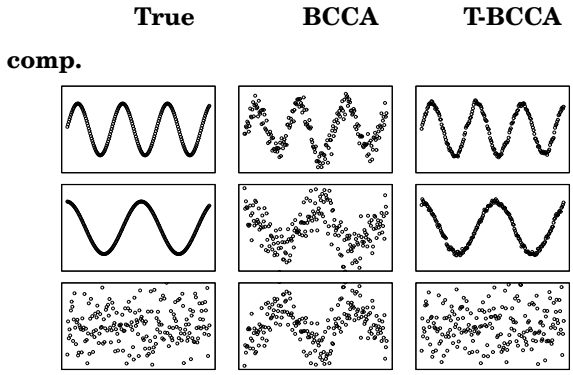


Figure 4.1. Comparison of time-dependent latent space CCA and iid latent space CCA on simulated data. The left column shows the components of the true simulated data, two of which are heavily time-dependent and the other is white noise. The middle column shows the latent components estimated by iid latent space CCA (BCCA), and the right column shows the estimation of time-dependent CCA (T-BCCA). T-BCCA captures the time-dependent components more accurately than BCCA. T-BCCA also captures the white noise component shown at the bottom row, while BCCA misses it.

5. Multitask Learning

5.1 Introduction

Many real-world supervised learning tasks are closely related. For instance, handwritten character recognition is a similar problem to digit recognition. *Multitask learning* suggests that learning these tasks together could bring better performance than considering them as independent problems [21].

The goal of handling multiple machine learning problems together is to transfer knowledge across problems to improve performance, which is called *transfer learning*. There are many types of transfer learning, which are explained below adopting the definitions and the dichotomy given in [112]. A machine learning problem can be defined by a domain $\mathcal{D} = \{\mathcal{X}, p(\mathbf{X})\}$ and a task $\mathcal{T} = \{\mathcal{Y}, f(.)\}$, where $\mathcal{X}$ is a feature space, $p(\mathbf{X})$ is the marginal distribution of data $\mathbf{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N\}$, $\mathcal{Y}$ is the output label space, and $f(.)$ is the predictive function we are trying to learn. Let us suppose we have a *source* problem $\{\mathcal{D}_S, \mathcal{T}_S\}$, and a *target* problem $\{\mathcal{D}_T, \mathcal{T}_T\}$. In transfer learning, the goal is to improve the solution of the target problem by transfering knowledge from the source problem.

Transfer learning problems can be cast into three categories based on the relationship of source and target problems and availability of output labels:

- *Inductive transfer learning*: Source and target tasks are different ($\mathcal{T}_S \neq \mathcal{T}_T$), while the domains can be either the same or different. Each individual task is solved by *inductive learning*, meaning that a general prediction rule is learned from training data, without using the test data whatsoever [41]. *Multitask learning* refers to a specific kind of induc-

tive transfer, where both source and target labels are available. The goal is to solve all problems simultaneously by transferring knowledge mutually. The problems are not in distinct groups as source and target. Instead, every problem is both the source and the target. *Multi-output (multilabel) learning* is a special case of multitask learning, which assumes a single domain and different label spaces for tasks. In Publication III and Publication VI, some of the learning settings are of this sort. Another setting where transfer learning has been shown to be very useful is when gathering labels for the target problem is cumbersome, while we have abundant *unlabeled* data for a different but closely related problem. This setting is called *self-taught learning* [99].

- *Transductive transfer learning*: Each individual task is solved by *transductive learning* [41], which means that the test input data are used in learning together with the training data for better estimation of the data distribution. In this setting, training and prediction are no longer exclusive processes. For each different test set, a new model has to be learned, which brings a considerable computational burden. This also entails the strict assumption that source and target tasks are the same ($\mathcal{T}_S = \mathcal{T}_T$) and domains are different ($\mathcal{D}_S \neq \mathcal{D}_T$). This setting has been introduced in [4].

- *Unsupervised transfer learning*: Multiple unsupervised learning tasks are learned together, such as clustering [30] or dimensionality reduction [127]. Hence, neither source nor target labels are available. Either the domain or the task or both are different between source and target problems.

There is no guarantee that knowledge transferred across tasks will always be useful. When the tasks are uncorrelated, the transfer might decrease the performance, causing what is called *negative transfer*. A proper multitask learning method should avoid negative transfer by allowing knowledge transfer only when the tasks are correlated.

5.2 Examples of multitask learning

In this section, examples of multitask learning are given from previous research. The main focus is put on extensions of SVMs, Gaussian processes,

and generative models to multitask setups, following the main theme of the thesis.

Suppose we have T supervised learning tasks on the labeled data sets $(\mathbf{X}_t, \boldsymbol{y}_t)$ where $\mathbf{X}_t = [\boldsymbol{x}_1^t, \boldsymbol{x}_2^t, \cdots, \boldsymbol{x}_N^t]$ is the set of input samples and $\boldsymbol{y}_t$ is the vector of corresponding labels y_i^t for task t.

The pioneer work that extends SVMs to multitask learning [35] transfers knowledge across tasks by binding the parameters of the tasks. It assumes that the hyperplane parameters are decomposed into two additive components,

$$\boldsymbol{w} = \boldsymbol{w}_0 + \boldsymbol{v}_t$$

where $\boldsymbol{w}_0$ represents the shared structure, and $\boldsymbol{v}_t$ the task-specific structure. The resulting optimization problem for T tasks indexed by t is

$$\begin{aligned} \underset{\boldsymbol{w}_t, b, \boldsymbol{\zeta}}{\text{minimize}} \quad & \left\{ \frac{1}{2} \sum_{t=1}^{T} \frac{\lambda_1}{T} \|\boldsymbol{v}_t\|^2 + \lambda_2 \|\boldsymbol{w}_0\|^2 + \sum_{t=1}^{T} \sum_{i=1}^{N} \zeta_i^t \right\} \\ \text{s.t.} \quad & y_i(\boldsymbol{w}_0 + \boldsymbol{v}_t) \cdot \Phi(\boldsymbol{x}_i^t) + b \geq 1 - \zeta_i^t, \quad \forall i, t \\ & \zeta_i^t \geq 0, \quad \forall i, t. \end{aligned} \tag{5.1}$$

Multitask learning on Gaussian processes was first studied by [75] as an extension the *informative vector machine* (IVM) [76]. Knowledge transfer between tasks is made possible by introducing cross-covariances between samples of different tasks. In a later study [19], Bonilla et al. followed this line by stacking the data of all tasks together and feeding into a single Gaussian process with a modified kernel

$$k_{MT}(\boldsymbol{x}_i^l, \boldsymbol{x}_j^m | \boldsymbol{\theta}, \mathbf{T}) = \mathbf{T}_{lm}\, k(\boldsymbol{x}_i^l, \boldsymbol{x}_j^m | \boldsymbol{\theta})$$

where $\boldsymbol{x}_i^l$ is the ith data point of task l, $\boldsymbol{x}_j^m$ is the jth data point of task m, $k(.,.)$ is a kernel parameterized by $\boldsymbol{\theta}$, and $\mathbf{T}$ is a matrix that stores the correlations between pairs of tasks in its entries. The matrix $\mathbf{T}$ is treated as a kernel parameter and learned together with $\boldsymbol{\theta}$ using type II maximum likelihood. An interesting property of this model is that knowledge transfer occurs only when the observations have additive noise.

Learning multiple tasks together is also studied within the Bayesian framework. In an early study, Bakker and Heskes [9] formulate a two-layer neural network for supervised learning tasks assuming that the input-to-hidden weights encode the shared knowledge and the hidden-to-output weights encode the task-specific knowledge. In a later study, Rai and Daumé III [32] suggest learning the latent hierachical relationships

of multiple tasks by using *Kingman's coalescent* as a prior over task parameters. Recently Archambeau et al. [3] approach multi-output regression by introducing a sparse matrix-variate Gaussian prior on the weight matrix. Titsias and Lázaro-Gredilla [119] propose transferring knowledge across several multiple kernel learning tasks via the kernel weights. In Publication III, an extension to SVM-based multiple kernel learning is introduced that follows the same knowledge transfer strategy, as will be detailed below.

5.3 Multitask multiple kernel learning for SVMs

The multitask extension of SVM in Equation 5.1 can be shown to be equivalent to the standard SVM with kernel [35]

$$\widehat{k}(\boldsymbol{x}_i^l, \boldsymbol{x}_j^m) = (1/\nu + \delta_{lm})k(\boldsymbol{x}_i^l, \boldsymbol{x}_j^m)$$

where ν denotes the similarity between tasks and δ_{lm} is the delta function. This model can be extended to the multiple kernel case simply by replacing $k(\boldsymbol{x}_i^l, \boldsymbol{x}_j^m)$ with a combined kernel. However, this brings severe drawbacks. For instance, all tasks are forced to share the same feature and label spaces, which makes the model incompatible to many applications. In addition, stacking the data of all tasks together heavily increases the time and space complexity.

Motivated by these shortcomings, in Publication III, a novel method called *multitask multiple kernel learning (MT-MKL)* is introduced. This method transfers knowledge across tasks via feature representations, instead of parameters. The method assumes that the tasks are multiple kernel learning problems, and transfers knowledge by regularizing the kernel combination parameters of similar tasks towards each other. In this model, the model parameters $\boldsymbol{\alpha}$ and the kernel combination parameters $\boldsymbol{\eta}$ of T tasks are learned jointly in a single min-max problem

$$\underset{\{\boldsymbol{\eta}^r\}_{r=1}^T}{\text{minimize}} \;\; \mathcal{O}_\eta = \left\{ \underset{\{\boldsymbol{\alpha}^r\}_{r=1}^T}{\text{maximize}} \;\; \Omega(\{\boldsymbol{\eta}^r\}_{r=1}^T) + \sum_{r=1}^T J^r(\boldsymbol{\alpha}^r, \boldsymbol{\eta}^r) \right\}$$

where $J^r(\boldsymbol{\alpha}^r, \boldsymbol{\eta}^r)$ denotes the optimization function of learner r, which is given by

$$J^r(\boldsymbol{\alpha}^r, \boldsymbol{\eta}) = \sum_{i=1}^{N^r} \alpha_i^r - \frac{1}{2} \sum_{i=1}^{N^r} \sum_{j=1}^{N^r} \alpha_i^r \alpha_j^r y_i^r y_j^r \left(k_\eta^r(\boldsymbol{x}_i^r, \boldsymbol{x}_j^r; \boldsymbol{\eta}) + \frac{\delta_i^j}{2C} \right)$$

with the constraint

$$\sum_{i=1}^{N^r} \alpha_i^r y_i^r = 0,\ \alpha_i^r \in \mathbb{R} \qquad \forall i.$$

Here, C is the regularization parameter as described in Section 3.2. Linear combinations of kernels are considered

$$k_\eta^r(\boldsymbol{x}_i^r, \boldsymbol{x}_j^r; \boldsymbol{\eta}^r) = \sum_{m=1}^{P} \eta_m^r k_m^r(\boldsymbol{x}_i^r, \boldsymbol{x}_j^r)$$

for simplicity and the parameter space is restricted to convex combinations

$$\sum_{m=1}^{P} \eta_m = 1,\ \eta_m \geq 0, \qquad \forall m$$

for interpretability. The regularization term

$$\Omega(\{\boldsymbol{\eta}^r\}_{r=1}^T) = -\nu \sum_{r=1}^{T} \sum_{s=1}^{T} \langle \boldsymbol{\eta}^r, \boldsymbol{\eta}^s \rangle \tag{5.2}$$

is applied to push the kernel combination parameters of similar tasks towards each other. Here, ν is a hyperparameter for tuning the scale of regularization. Tasks merge into a single task for very large values, and they become independent for very small values. This hyperparameter is learned by cross-validation.

An iterative algorithm is applied to solve this optimization problem, consisting of three steps. In the first step, the combined kernel matrix $\boldsymbol{K}_{\boldsymbol{\eta}}^r$ is computed for each task r with the current value of $\boldsymbol{\eta}$. In the second step, the model parameters $\boldsymbol{\alpha}_r$ are updated for each task by training a standard SVM on the precomputed kernel matrix $\boldsymbol{K}_{\boldsymbol{\eta}}^r$. In the third step, both the kernel matrix and the model parameters are kept fixed, and the kernel combination parameters are updated applying projected gradient-descent. In particular, a step is taken towards the opposite direction to the gradient of the cost function with respect to $\boldsymbol{\eta}_r$, satisfying the convexity constraint. The gradient of the cost function is

$$\frac{\partial \mathcal{O}_\eta}{\partial \eta_m^r} = -2\nu \sum_{s=1}^{T} \eta_m^s - \frac{1}{2} \sum_{i=1}^{N^r} \sum_{j=1}^{N^r} \alpha_i^r \alpha_j^r y_i^r y_j^r k_m^r(\boldsymbol{x}_i^r, \boldsymbol{x}_j^r)$$

for binary classification, and

$$\frac{\partial \mathcal{O}_\eta}{\partial \eta_m^r} = -2\nu \sum_{s=1}^{T} \eta_m^s - \frac{1}{2} \sum_{i=1}^{N^r} \sum_{j=1}^{N^r} \alpha_i^r \alpha_j^r k_m^r(\boldsymbol{x}_i^r, \boldsymbol{x}_j^r)$$

for regression.

MT-MKL allows both feature and label representations of tasks to be different, since the only coupling of tasks is via kernel combination parameters. It also tackles the computational infeasibility problem of [35] since in the optimization problem, the learners of tasks are additively combined. Although the regularization term in Equation 5.2 makes the cost function concave, the optimization problem still converges quickly due to that the kernel weights are bounded to feasible sets.

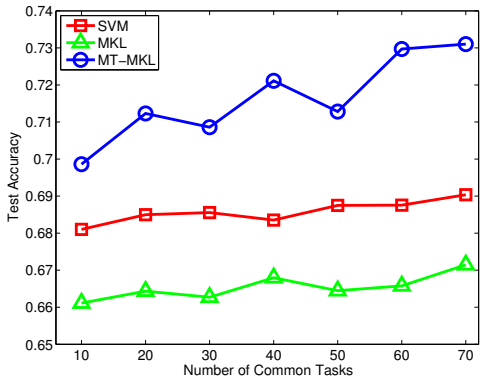


Figure 5.1. Test accuracies of the multitask multiple kernel learning model (MT-MKL), standard multiple kernel learning (MKL), and standard SVM (SVM) on synthetic data as a function of the number of common tasks T are given. The increase in the accuracy of MT-MKL as a function of T demonstrates that MT-MKL effectively transfers knowledge across similar tasks.

In Figure 5.1, the knowledge transfer in MT-MKL is demonstrated on synthetic data. A number of binary classification tasks are generated that share the same data distribution. Each task contains 12 samples, 6 per each output label. The sample size is set to such a small value to simulate a scarce data regime, where multitask learning is the most useful. The samples are drawn from two overlapping normal distributions of three dimensions. Two uncommon tasks are also included to test whether the model avoids negative transfer. The data are generated for each of these two tasks similarly as above, except that the samples are drawn from different normal distributions. MT-MKL is compared with single-task MKL and the standard SVM. In MKL and MT-MKL, a Gaussian kernel with unit length scale is assigned to each of the three input features. For SVM, a Gaussian kernel with a length scale of $\sqrt{3} = 1.73$ is used. In all methods, C is picked from $\{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$, and in MT-MKL also ν from $\{0.01, 0.1, 1, 10, 100\}$ by cross-validation. Test accuracies of the three models in comparison are given in Figure 5.1 as a function of the number of common tasks T. MT-MKL always outperforms the single-task models,

and its accuracy follows an increasing trend proportional to the number of common tasks.

In Publication III and Publication VI, MT-MKL is applied to learning the affective state and mental workload from biosignals coming from multiple sensors. In this setting, the learning tasks are defined in two ways: (i) learning a predictor for each mental state, and (ii) given a mental state, learning a predictor for each user. A multiple kernel learning problem is set up for each task by assigning one kernel to each sensor.

6. Biosensing Technology

In this chapter, an overview of the biosensing technology is given, focusing on the sensors used in the experiments in the thesis. The working mechanisms of these sensors are explained in brief and a list of most frequently used features of their signals is given.

6.1 Eye tracking

Eye tracker is a device that measures the eye movements. Eye tracking techniques can be classified into two based on what they monitor [33]:

- **the position of the eye relative to the head:** This is commonly monitored by the technique called *electrooculography (EOG)*. EOG tracks the rotation of the eye from the change of the electrostatic field on the skin caused by the rotation. EOG signal has been used in Publication VI for recognizing affective states such as valence, arousal, and liking.

- **the orientation of the eye in space:** Here, the goal is to locate where in the visual field the subject is looking at, in other words, where the gaze target is. Hence, this type of eye tracking is sometimes referred to as *gaze tracking* [47]. There are two major gaze tracking techniques [43]:

  – **Contact lens-based tracking**: Eye movements are tracked by contact lenses that have special properties. Tiny planar mirrors are placed into the lens, and the eye direction is calculated from reflections of the light [72]. This technique delivers extremely high resolution. However, the hardware is intrusive.

– **Optics-based tracking**: A camera tracks the eye movements from the light reflecting from the outer surface of the cornea (the first Purkinje image) and the center of the pupil. Accuracy could be increased by using an infrared camera, which is not affected from ambient light changes. Reflections from both the outer cornea and the back of the lens (the fourth Purkinje image) could be used for even higher accuracy [28]. Optical eye trackers can be used in both stationary and mobile setups by attaching the infrared camera to a suitable place in the setup. The eye trackers with cameras placed away from the user, such as on the monitor frame, are called *remote eye trackers*, while the ones whose cameras are attached to the user's head are called *head-mounted trackers*. Optical tracking is the most widespread technology in commercial eye trackers, since it provides reasonable accuracy using cheap and unobtrusive hardware. All the eye movement data used in this thesis have been collected by this technology.

Human visual system operates based on neural adaptation; while abrupt changes in the visual stimuli cause strong responses, stable stimuli fade out [82]. The physiology of the eye is also compatible to this neural infrastructure. Within a layer of light-sensitive cells in the eye, called *retina*, only a tiny spot with a diameter of about 1.0 mm gives high visual acuity, due to the high concentration of the photoreceptors (cones) it has. This spot is called *fovea*. Different images having high resolution at the centre and blurred off-the-centre are gathered by fast eye movements, and integrated in the brain. When we desire to get a more detailed view of a certain location, we restrict our eye movements into a small area. This is called a *fixation*. However, the eye keeps on tiny movements, called *microsaccades* [54], within that region involutarily. In many applications, microsaccades are neglected for being too fine-grained. Given an eye trajectory, the fixations are detected by merging closely located consecutive targets into a single target point. What remains is then the rapid jumps between fixations, called *saccades* [33].

Some useful eye movement features for analyzing user behaviour are:

- **Fixation duration:** Fixation duration is a very good indicator of user interest, as the eye fixates on a point as long as more information is needed [1, 66, 103]. Mean and standard deviation of fixation duration within the object of interest (used in Publication IV), and mean distance

of fixations to the object center (used in Publication I) are among useful features for mental state inference.

- **Saccade length:** Mean and standard deviation of the saccade length are useful in discriminating differences in the user's intentions [69]. In Publication IV, these features are used in inferring whether the painting the user looks at is relevant.

- **Pupil diameter:** Pupil size is correlated with mental activity [52]. This feature is used in detecting mental workload in Publication VI, and as an auxiliary indicator of user interest and attention in Publication IV and Publication V, respectively.

- **Electrooculogram:** The raw EOG signal is the electrostatic field of the eye measured from the skin near the eye. It is possible to detect eye blinks from EOG signals [68]. Eye blink rate is a strong indicator of arousal; high arousal significantly increases the eye blink rate [62]. Features of this signal such as energy, mean, and variance are discriminative in inferring the affect [67].

Eye gaze is correlated with visual attention [61]. This encouraged the human-computer interaction community to take the user input directly from eye movements. Bolt introduced this idea [18] and illustrated it on an eye movement-based interface for selecting and zooming video streams simultaneously playing on a computer screen. Later on, eye gaze has been used as a side modality to speed up hands-free gaze-based typing [128], pan-and-zooming [113], and scrolling [71]. As machine learning and pattern recognition methods matured, interaction schemes based on more abstract notions of the user (e.g. interests, preferences, etc.) have been developed for applications such as text [48] and content-based image retrieval [69]. The approach of this thesis to eye movement analysis falls into this last category.

In Publication I, Publication V, Publication III, and Publication VI, Tobii 1750 remote eye tracker with 50Hz sampling rate and an accuracy of 0.5 degrees of visual angle has been used (see the monitor in Publication VI). This eye tracker has an infra-red camera and light attached to the monitor frame. In Publication II, a head-mounted near-to-eye display with an integrated eye tracker collected the eye movements with a sam-

pling rate of 25 Hz and an accuracy of 1 degree of visual angle (see the image on the left in Figure 4 of Publication II). The device is produced by Nokia Research Center as a research prototype [59]. And in Publication IV, gaze data has been collected by the SMI iView X HED head-mounted eye tracker which has a sampling rate of 50 Hz and an accuracy of 0.5 to 1 degrees of visual angle.

6.2 Electroencephalography

Brain activity causes ionic flow within the brain neurons, which elicits voltage changes on the scalp [88]. Electroencephalography (EEG) measures the electric potential on the scalp surface by electrodes placed on locations over the brain regions of interest. The resulting signal is typically decomposed into the following frequency bands:

- **Delta:** 1 to 3 Hz,
- **Theta:** 4 to 7 Hz,
- **Alpha1:** 8 to 9 Hz,
- **Alpha2:** 10 to12 Hz,
- **Beta1:** 13 to 17 Hz,
- **Beta2:** 18 to 30 Hz,
- **Gamma1:** 31 to 40 Hz,
- **Gamma2:** 41 to 50 Hz

In Publication III, Publication V, and Publication VI, spectral powers of these bands are used as features in analysis.

EEG has good and bad properties compared to other brain signaling techniques such as *functional magnetic resonance imaging (fMRI)* and *magnetoencephalography (MEG)*. Its greatest advantage is that its equipment is much cheaper and much less intrusive compared to the other two. It also gives better time resolution than fMRI. Its downsides are its poor

spatial resolution and low accuracy at inner brain regions. A readily available data set of eight EEG channels is used from the previously published DEAP (A Database for Emotion Analysis [67]) data set in Publication VI. In Publication V and Publication VI, two new data sets are collected by NeuroSky single-channel EEG sensor with a sampling rate of 512 Hz. The EEG sensor is placed on a sensor arm connected to headphones. The arm is placed to the FP1 location of the International 10-20 system (see Figure 2 in Publication VI).

6.3 Motion sensing

Monitoring the mental state of a person by the naked eye from body motion has been studied for centuries [31], and has been exploited in recent research for developing emotion-aware interaction schemes [13, 114]. Body motion is measured by a device called *accelerometer*, which senses the *g-force* (acceleration relative to free-fall) exerted on a location from the displacement of a damped mass along a spring. This mechanism can be extended to three axes, and the resulting three dimensional signal can be treated as the 3D acceleration vector. In Publication III, Publication V, and Publication VI, the 3D acceleration vector was measured by a research prototype accelerometer from the nape of the user at 15 Hz (see Figure 2 in Publication VI). The following features are extracted from each of the three dimensions of the acceleration signal:

- mean and variance of the signal,
- mean of the derivative of the signal,
- mean, median, and maximum peak-to-peak interval.

6.4 Heart rate monitoring

In an earlier study, it has been shown that high arousal increases heart rate, and that this increase is higher for low valence (anger, fear, and sadness), compared to high valence (happiness) and neutral valence (surprise) [34]. Heart signal is an essential element in affective state recogni-

tion [94]. This signal is also used in Publication III and Publication VI in the same context.

There are two major techniques for measuring the heart rate:

- **Electrocardiography (ECG):** Heart activity is monitored from the electrical changes on the skin by electrodes placed on the chest, similarly to EEG, EOG, and EMG. Typically, electrodes are attached on a strap which is tied on the chest during data collection. ECG data have been collected in Publication III and Publication VI using a Suunto heart belt which records RR-intervals (the time between two consecutive R waves in the electrocardiogram (ECG)) at 2 Hz.

- **Plethysmography:** The heart beat changes the blood flux in the vessels. This allows monitoring the heart activity from the volume of an organ, such as the thumb. Although this signal is less precise than ECG, it is easier and cheaper to measure. Plethysmograph data taken from [67] has been used in Publication VI.

Features useful for analyzing heart signals include [67]:

- **Raw heart signal:** The energy ratio between (0.04-0.15) Hz and (0.15-0.5) Hz bands of the raw heart signal.

- **Interbeat (R-R) interval:** The time interval between two heart beats. It can be calculated from the interval between two R waves, which are sharp peaks in the beat signal pattern.

- **Heart Rate Variability (HRV):** Variation in the interbeat interval. It can be calculated simply by taking the derivative of the interbeat interval by finite-difference approximation. Spectral powers of (0.1-0.2)Hz, (0.2-0.3)Hz, (0.3-0.4)Hz, (0.01-0.08)Hz (0.08-0.15)Hz, (0.15-0.5)Hz bands of the HRV signal are known to be informative of the affective state.

6.5 Other useful biosensors

The DEAP data set [67] used in Publication VI contains a number of other sensors than the ones above. Here, we introduce these sensors briefly:

- **Electromyography (EMG)**: EMG measures the electrical activity on muscles. The electrodes can be placed to any muscle depending on the application. As an example, in [78], the muscular tension at the trapezius is measured, and its correlation with stress is studied. In the DEAP data set, EMG data from the right trapezius and zygomaticus major are available. Energy, mean, and variance has been used as the features of this signal in Publication VI.

- **Galvanic Skin Response (GSR)**: GSR measures the electrical conductance of the skin. It is a good indicator of arousal, since the sympathetic nervous system controls the sweat glands [83]. The DEAP data set has GSR data collected from the middle finger and the ring finger. Some useful features of this signal are its mean, mean of the derivative, mean of the positive derivatives, proportion of negative samples in the derivative, number of local minima, and 10 spectral powers in the (0-2.4)Hz frequency interval [67].

- **Respiration sensor**: This sensor measures the moisture level on the skin. It is usually attached on a belt that is tied on the chest. Respiration is highly correlated with emotions [50]. Some useful features of the respiration signal are: band energy ratio, average respiration signal, mean of the derivative, standard deviation, range of the greatest breath, 10 spectral powers between (0-2.4)Hz, average and median peak-to-peak time.

- **Skin Temperature**: This signal is highly correlated with emotions [84]. The DEAP data set includes skin temperature data measured from the pinky finger. This signal has been summarized with its mean, mean of the derivative, spectral power in (0-0.1)Hz and (0.1-0.2)Hz.

6.6 Biosensor importance in mental state inference

In this section, relative contributions of the sensors to prediction of mental states are given for the case studies that involve multiple-sensors measurement setups, and the implications of these contributions are briefly discussed. It is worthwhile to mention that the sensor importance results given below are an automatic by-product of the machine learning model that has been introduced.

As seen in Figure 2 of Publication III and Figure 4 of Publication VI, 3D body motion made the highest contribution, and pupil dilation the second highest. This ranking is supports other studies on inferring affective states from these sensors with reasonable success [26, 108]. This is a very promising result for proactive interaction by biosensors, considering the low obtrusiveness of especially the accelerometer.

Among the larger and more accurate sensor set used in the public DEAP data set, the 32-channel EEG sensor clearly dominates the other sensors in contribution to prediction, while GSR sensor comes the second (see Figure 1 in Publication VI). The domination of the EEG sensor is a sensible result considering that it is significantly data-richer than the other sensors in the setup. This outcome also reveals the trade-off between unobtrusiveness and high accuracy of sensors. Better prediction of the mental state is more likely as more accurate sensors are used. On the other hand, higher accuracy in sensing often comes at the expense of higher obtrusiveness. In the extreme case, we can think about functional Magnetic Resonance Imaging (fMRI) as an extremely space-accurate technique of monitoring the brain activity, which is impossible to be used in real-life human-computer interaction scenarios today and in the near future.

7. Inferring Mental State

A proactive interface requires a circular information flow between the user and the system. The system monitors the user, extracts cues about the user interests, and changes the interface accordingly. The user behaviour changes, hopefully improves, in this new interface. Meanwhile, the system keeps on monitoring the user, extracts new cues, and this interchange of information goes on in a virtuous circle, as illustrated in Figure 7.1. The success of a proactive interface depends on how accurately the system is able to infer the user interests from the available cues. The main contribution of this thesis is a constellation of novel machine learning models for inferring valuable information about user interests. The models extract cues from biosensors attached to users who are monitored under various novel experimental setups. This thesis is a feasibility study consisting of models covering the first half of the information flow shown in Figure 7.1 as thick black arrows.

In each of the experiment setups in the thesis, a different aspect of the user's mental state is analyzed. Below, these setups are described, positioned in the existing literature, their novelties are highlighted, and hints are given about how they can be useful in the future human-computer interaction systems.

7.1 Inferring the relevance of real-world objects

Relevance has been extensively studied in the context of information retrieval for text documents [7] and images [81, 77]. This thesis extends this term to real-world objects by treating them as information channels whose relevances to the users are to be predicted. This information is valuable especially for pervasive information access systems [56, 90, 91]. Since traditional input media, such as keyboard and mouse, are not avail-

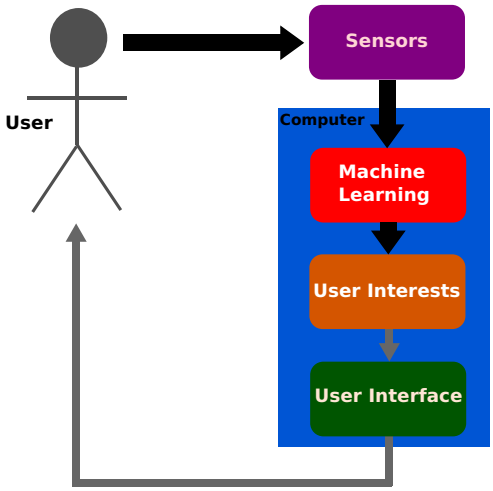


Figure 7.1. The circular information flow in a proactive interface is depicted as a block diagram. The user is monitored by sensors, machine learning algorithms infer the user interests from the monitored signals, and then the system changes the user interface accordingly. The user's reactions to the updated interface are then monitored and handled in the next cycle. In this thesis, the first half of this flow shown in thick black arrows is investigated as an attempt to construct a basis for proactive interaction via biosensors.

able in these setups, hands-free solutions are very desirable.

In Publication II we built a pervasive contextual information access system consisting of goggles with an attached eye tracker and a near-to-eye display fed by a forward-pointing camera. Real-world objects (faces and augmented-reality markers) are recognized from the video image of the field of view gathered by this camera. An information box is then displayed near each object the user looks at. The textual information shown in the box is retrieved from a database based on the context. The context is inferred from the relevance of the previously shown items, which is estimated by the proportion of the time they are looked at within a fixed-length time window (*gaze intensity* [97]). The survey made on the users reveals that the system successfully infers the context and retrieves useful information about the objects in the scene (Question 3 in Figure 5 of Publication II).

As a feasibility study for more advanced relevance estimators, in the next step object relevance is investigated in real-world video scenes, as an approximation to pervasive scenarios. In Publication I, the users were monitored by a remote eye tracker attached on a desktop computer while they were watching a video of a real-world scene where some objects were augmented with textual information. After the experiment, the users

were shown snapshots of the video and asked to rank the objects in each snapshot by relevance. It has been observed that ordinal logistic regression from a combination of gaze pattern features and visual features to the relevance rankings of objects predicts the true object relevance rankings with up to 85% accuracy. This accuracy is approximately 10% higher than ranking based on visual saliency (Figure 2 of Publication I).

Motivated by the results of Publication I, object relevance inference problem has been investigated on a pervasive setup in Publication IV. Users explored an experimental painting gallery holding a button in their hand and clicked that button when they were viewing a picture they found interesting. The machine learning question has been to predict the clicks from gaze patterns. The Gaussian process classifier with a time-series kernel predicted the relevant objects with an area under ROC curve (AUC) of up to 76%, which is 15% above the accuracy of dwell-time thresholding.

The overall outcome of these studies is that gaze patterns in pervasive scenes contain a significant amount of information about the user's interests, and this information can be extracted by machine learning techniques. The ideal case where the user's intentions are predicted with perfect accuracy would be a zero-effort solution to the well-known *Midas touch*[1] problem [58] of gaze-based user interfaces. Considering the accuracies reported in the studies above, it is not yet possible to claim that zero-effort commanding is possible. However, the accuracies still suggest that the amount of relevance feedback is already high enough for building gaze-based pervasive recommender systems.

7.2 Inferring affective state and mental workload

Affective computing is a field of research, the goal of which is to predict the affective (or emotional) state of subjects from user actions [95], facial expressions [133], or biosignals [5, 25, 64, 94]. The outcome of this research is valuable for developing *emotionally intelligent* machines [24, 94]. A machine aware of the user's affective state is beneficial especially when the user is exposed to heavy multitasking. For instance, when the user is in

[1] In gaze-based user interfaces, a mechanism is required to distinguish whether the user intends to click the object at which she is looking. The absence of such a mechanism results in a click on wherever the user looks, clearing away all the charm of gaze-based commanding. The problem is similar to that of the Greek mythological character Midas who turns every object he touches into gold.

deep thought, she would not want to be disturbed by e-mail alerts.

In an experiment, desktop users were measured by four sensors (EEG, body motion, ECG, and pupil dilation) while they were performing naturalistic tasks. These tasks include filling in a personal survey, comparing pictures, and solving logical puzzles (see Figure 3 of Publication VI for details). Subjects annotated the ground-truth levels of their valence, arousal, and mental workload during each step of the experiment.

In Publication III, a novel multitask learning multiple kernel learning algorithm (MT-MKL) is introduced. The model assumes that related learning tasks are all multiple kernel learning tasks on the same set of kernels. The model transfers knowledge across tasks by forcing similar tasks to have a similar combination of kernels (see Section 5.3 for further details). MT-MKL is observed to perform better than its counterparts on three benchmark data sets. The first one is a multitask regression problem, where MT-MKL gives an RMSE of 23, while its existing counterpart [35] gives 38. The second one is a handwritten recognition data set, where each task is binary discrimination of visually similar letters such as f and t. MTMKL improves over single-task learning by 0.5% of accuracy. Its existing counterpart is not applicable to this learning setup. The third data set is the one collected by the experiment described above. MT-MKL improves over single-task learning by 5% here. Its existing counterpart is not applicable to this setup also. MT-MKL is also observed to demand significantly less computational time than [35].

In Publication VI, the benefits of MT-MKL in mental state inference are more thoroughly investigated. The model is applied to two data sets. The first is the publicly available DEAP [67] data set, which consists of measurements of seven sensors while subjects were watching video clips. After the experiment, the ground-truth labels (valence, arousal, and liking of the subject during each video) are taken from the users by showing them snippets of the videos. When each subject is taken as a learning task, MT-MKL gives 65% prediction accuracy, which is 4% higher than the Naive Bayes classifier of [67]. The second data set is the one collected by the experiment described above. This experimental setup is less controlled and contains a smaller set of sensors than the DEAP data set. Hence, it can be considered as a step taken towards the real-life scenarios. When each subject is taken as a task, MT-MKL predicts the affective state and mental workload with 71% accuracy, which is 2% higher than the single-task SVM. When each output label (mental state) is taken as a

task, hence in the multi-output prediction case, MT-MKL gives 64% accuracy, while single-task SVM is 4% less accurate. It is also worthwhile to note that MT-MKL also outputs the contributions of sensors to prediction as a by-product.

The overall outcome of this line of research has been that it is possible to predict affective state and mental workload clearly over the chance level. Furthermore, the prediction gets more accurate as more advanced models are employed, which motivates further research in machine learning methods development for this application field. Nevertheless, the accuracies are not yet at the sufficient level for practical applications. There are two major sources that induce prediction errors. The first is the low signal-to-noise ratio of sensors, which can be partially overcome as the instrumentation improves. The second is the ground-truth label noise, which stems from the fact that subjects cannot remember, nor can they evaluate, their own mental states perfectly. This problem can be solved by more extensive experimentation on setups where ground-truth is imposed by the setup itself.

7.3 Inferring auditory attention

We manage the excessive information continuously provided by our senses by focusing our attention on a subset of it. For example, we fixate our eyes at a location that is visually interesting for us. This is called *visual attention*. A very strong indicator of visual attention is the point of regard [61]. Today's technology allows monitoring the point of regard, hence visual attention to a large extent, with reasonable accuracy. Visual attention has been observed to be a very useful information for proactive interaction in previous studies [121].

Attention can be directed to auditory stimuli as well, which is then called *auditory attention* [37]. There is no directly observable indicator of auditory attention, hence it cannot be monitored as easily as visual attention. Auditory attention can be used in developing very interesting applications, such as:

- When the user's attention is detected to be low, media players can put a bookmark to the played audio book. Then the user can fast-rewind to those moments later.

- Parts of the music that the user most enjoyed, hence paid highest attention, could be used as a query for retrieving similar songs from a data base.

- Moments of high attention to a dynamic audio content being recorded by a microphone could be used for meeting summarization.

Despite these potential benefits, recognition of auditory attention has not so far attracted much interest. Previous work has targeted the low-level physiology of auditory attention (see [37] for a survey) observed by heavy hardware such as fMRI on highly controlled setups.

In Publication V, inferring auditory attention is studied for the first time from a data modeling perspective. Less controlled stimuli, naturalistic user tasks, and low-quality unobtrusive biosensors have been used to make the experimental setup as compatible as possible to real-life scenarios. Desktop computer users were measured by single-channel EEG sensor, accelerometer, and eye tracker while they were listening to naturalistic audio content (scientific podcast, music, and radio drama). As a second simultaneous task, the users solved a visual search puzzle (given a grid of objects, identifying the odd one in shape and colour). Ground-truth attention levels were imposed to periods of audio stimuli by varying the difficulty level of the visual task.

Given a labeled data set as above, auditory attention can be inferred by any supervised learning algorithm. However, gathering labeled ground-truth data from end-users would not be as feasible and reliable in a realistic end-user scenario as it is in laboratory conditions. To overcome this problem, a novel machine learning model has been built that does not require labels in training, but can still predict labels of new instances. This model calculates the correlation between the audio stimulus and the biosignals, and predicts the level of attention based on the hypothesis that the correlation is proportional to the level of attention paid to the stimulus. The model calculates the correlations using a novel variant of Bayesian CCA which assumes time-dependence in the latent space, compatible to the time-series spirit of biosignals. The prediction accuracy of this unsupervised model has been 44% in a four-class classification problem (four attention levels), while the best other CCA variant reached 42%, and the best supervised model reached 47% (see Table 2 of Publication V for details). The accuracy of the time-dependent CCA, which is very close

to supervised models, is not yet high enough for end-user applications. However, its being significantly above chance level (25%) implies that the research direction is promising. Better accuracies are very likely with a more accurate sensor setting, where the same prediction model will still be applicable.

8. Conclusions

Proactive user interfaces anticipate the user's interests and automatically take actions desirable for the user. The user's mental state gives strong cues about the user's interests. In this thesis, inferring the user's mental state from signals such as EEG, heart rate, body motion, and eye movements is studied. The users have been measured by biosensors in various naturalistic experimental setups, and their various mental states have been inferred in these setups by novel machine learning models. The investigated mental states are:

- the affective state,
- mental workload,
- liking,
- real-world object relevance,
- auditory attention.

Biosensing technology allows monitoring many biosignals that correlate with emotions and mental processes. However, these signals are very noisy and their correlation to valuable information is not easily observable. Previous studies followed two strategies to reduce the biosignal noise:

- to incorporate large sets of sensors including ones that are very expensive, obtrusive, and unportable,

- to simplify experimental setups for better controlling the residual factors.

It is clear that these two strategies are not generalizeable to end-user applications. In this thesis, machine learning is approached as an alternative strategy for improved mental state inference. Instead of reducing the sources of noise in the setup by extensively controlled experiments and extracting the optimal feature sets from the signals, more advanced machine learning models are built. Relying on the assumption that high noise levels are inevitable in real-world interaction setups, in the experiments, unobtrusive and cheaper sensors have been used, contrasting to stationary and very expensive technologies such as MEG and fMRI.

Table 8.1 gives a list of contributions of this thesis, classified into two as methodological and application oriented. The former refers to the machine learning algorithms, and the latter to the experimental setups.

8.1 Discussion and Future Directions

This thesis has shown in several case studies that various mental states of can be inferred at reasonable accuracies. Inferrable mental states include the user's emotions, interests, and mental workload, which are very fruitful pieces of information for building more intelligent proactive interfaces. The fact that all reported prediction accuracies are clearly over random indicates that biosignals do contain significant information about mental states. And the fact that the highest accuracies are reached by the proposed machine learning methods indicate that developing advanced learning models for suboptimal biosensor sets is a very promising strategy for proactive inferface development research.

The experimental setups reported in the thesis are designed to be naturalistic, and relatively loosely controlled. The prediction models also do not contain any features closely tied to the studied setups. Hence, it is sensible to expect the generalization of the outcome to other domains to be at a large extent. Nonetheless, empirical investigation of the generalization issue should still be addressed in future work.

Another future direction is to improve the prediction accuracy by more customized features and models. The intuitive and simple feature sets used in the thesis could be replaced by optimal ones that can be constructed by dedicated perceptual science experiments. All the models pre-

Publ.	Forum	Methodological	Application
I	ETRA	–	Inferring object relevance in video scenes.
II	Virtual Reality	–	Experimental study of a wearable system that uses object relevance as a contextual cue for information retrieval.
III	ICONIP	Multitask MKL	–
IV	ICMI	GP classification using a LTW kernel	Inferring relevance of real-world objects
V	ECML	Time-dependent CCA	Inferring auditory attention
IV	Submitted	–	• Improving affective state inference by multitask MKL. • Learning biosensor importance by MKL.

Table 8.1. The methodological and application-oriented contributions of the publications in the thesis.

sented in this thesis are trained by batches of samples. Online learning variants of these models could adapt to trends better, and could be more suitable for proactive interaction due to reduced training times. On the way to proactive interfaces, the final challenge is to accomodate the interaction environment to the inferred mental state of the user. This problem is also left to future work.

Bibliography

[1] A. Ajanki, D.R. Hardoon, S. Kaski, K. Puolamäki, and J. Shawe-Taylor. Can eyes reveal interest? Implicit queries from gaze patterns. *User Modeling and User-Adapted Interaction*, 19(4):307–339, 2009.

[2] C. Archambeau and F.R. Bach. Sparse probabilistic projections. In Y. Bengio L. Bottou D. Koller, D. Schuurmans, editor, *Proc. of the Advances in Neural Information Processing Systems*, pages 73–80, Cambridge, MA, 2008. MIT Press.

[3] C. Archambeau, S. Guo, and O. Zoeter. Sparse Bayesian multi-task learning. In J. Shawe-Taylor, R.S. Zemel, P.L. Bartlett, and K.Q. Weinberger F. Pereira, editors, *Proc. of the Advances in Neural Information Processing Systems*, pages 1755–1763, Cambridge, MA, 2011. MIT Press.

[4] A. Arnold, R. Nallapati, and W.W. Cohen. A comparative study of methods for transductive transfer learning. In *Proc. of the IEEE Int'l Conf. on Data Mining Workshops*, pages 77–82, Washington, DC, 2007. IEEE Computer Society.

[5] I. Arroyo, D.G. Cooper, W. Burleson, B.P. Woolf, K. Muldner, and R. Christopherson. Emotion sensors go to school. In *Proc. of the Conf. on Artificial Intelligence in Education*, pages 17–24, Amsterdam, The Netherlands, 2009. IOS Press.

[6] F.R. Bach and M.I. Jordan. A probabilistic interpretation of canonical correlation analysis. *Technical Report 688, Department of Statistics, University of California, Berkeley*, 2005.

[7] R.A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[8] G.H. Bakir, T. Hofmann, B. Schölkopf, A.J. Smola, B. Taskar, and S.V.N. Vishwanathan. *Predicting structured data*. Neural Information Processing Series. The MIT Press, 2007.

[9] B. Bakker and T. Heskes. Task clustering and hating for Bayesian multitask learning. *J. Mach. Learn. Res.*, 4:83–99, 2003.

[10] D. Barber and S. Chiappa. Unified inference for variational Bayesian linear Gaussian state-space models. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Proc. of the Advances in Neural Information Processing Systems*, pages 81–88, Cambridge, MA, 2006. MIT Press.

[11] T. Bayes. An essay towards solving a problem in the doctrine of chances. *Phil. Trans.*, 53:370 – 418, 1763.

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

[13] D. Bernhardt. Emotion inference from human body motion. Technical Report UCAM-CL-TR-787, University of Cambridge, Computer Laboratory, 2010.

[14] C.M. Bishop. Variational principal components. In *Proc. of the Int'l Conf. on Artificial Neural Networks*, pages 509–514, Heidelberg, 1999. Springer-Verlag.

[15] C.M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, NY, 2006.

[16] D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[17] M. Boardman and T. Trappenberg. A heuristic for free parameter optimization with support vector machines. In *Proc. of the Int'l Joint Conf. on Neural Networks*, pages 610–617, New York, NY, 2006. IEEE.

[18] R.A. Bolt. Gaze-orchestrated dynamic windows. *SIGGRAPH Computer Graphics*, 15(3):109–119, 1981.

[19] E. Bonilla, K.M. Chai, and C.I. Williams. Multi-task Gaussian process prediction. In D. Koller, D. Schuurmans, Y. Y. Bengio, and L. Bottou, editors, *Proc. of the Advances in Neural Information Processing Systems*, pages 153–160, Cambridge, MA, 2008. MIT Press.

[20] M. Borga. Canonical Correlation: A Tutorial. 2001. `http://www.imt.liu.se/~magnus/cca/`.

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

[22] O. Chapelle and V. Vapnik. Model selection for support vector machines. In S.A. Solla, T.K. Leen, and K.R. Müller, editors, *Proc. of the Advances in Neural Information Processing Systems*, pages 230–236, Cambridge, MA, 1999. MIT Press.

[23] K. Chaudhuri, S.M. Kakade, K. Livescu, and K. Sridharan. Multi-view clustering via canonical correlation analysis. In A.P. Danyluk, L. Bottou, and M.L. Littman, editors, *Proc. of the Int'l Conf. on Machine Learning*, pages 129–136, New York, NY, 2009. ACM.

[24] D. Chen and R. Vertegaal. Using mental load for managing interruptions in physiologically attentive user interfaces. In E. Dykstra-Erickson and M. Tscheligi, editors, *Extended Abstracts on Human Factors in Computing Systems*, pages 1513–1516, New York, NY, 2004. ACM.

[25] C. Conati and H. Maclaren. Modeling user affect from causes and effects. In M. Zancanaro and F. Pianesi, editors, *Proc. of User Modeling, Adaptation, and Personalization*, pages 4–15, Berlin, Heidelberg, 2009. Springer-Verlag.

[26] Cristina Conati and Christina Merten. Eye-tracking for user modeling in exploratory learning environments: An empirical evaluation. *Knowledge-Based Systems*, 20(6):557–574, 2007.

[27] R.T. Cox. Probability, frequency and reasonable expectation. *American Journal of Physics*, 17:1 – 13, 1946.

[28] H. D. Crane and C. M. Steele. Generation V dual Purkinje-image eyetracker. *Applied Optics*, 24(4):527–537, 1985.

[29] M. Cuturi. Fast global alignment kernels. In L. Getoor and T. Scheffer, editors, *Proc. of the Int'l Conf. on Machine Learning*, ICML '11, pages 929–936, New York, NY, 2011. ACM.

[30] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu. Self-taught clustering. In Z. Ghahramani, editor, *Proc. of the Int'l Conf. on Machine Learning*, pages 200–207, New York, NY, 2008. ACM.

[31] C. Darwin. *The expression of the emotions in man and animals*. Harper Perennial, 1872/2009.

[32] H. Daumé, III. Bayesian multitask learning with latent hierarchies. In Bilmes J and A.Y. Ng, editors, *Proc. of Uncertainty in Artificial Intelligence*, pages 135–142, Arlington, VA, 2009. AUAI Press.

[33] Andrew T. Duchowski. *Eye Tracking Methodology: Theory and Practice*. Springer-Verlag Inc., New York, NY, USA, 2007.

[34] P. Ekman, R.W. Levenson, and W.V. Friesen. Autonomic nervous system activity distinguishes among emotions. *Science*, 221(4616):1208–1210, 1983.

[35] T. Evgeniou and M. Pontil. Regularized multi-task learning. In W. Kim, R. Kohavi, J. Gehrke, and W. DuMouchel, editors, *Proc. of the Int'l Conf. on Knowledge Discovery and Data Mining*, pages 109–117, New York, NY, 2004. ACM.

[36] N.H. Frijda. *The Emotions*. Cambridge University Press, Cambridge, UK, 2002.

[37] J.B. Fritz, M. Elhilali, S.V. David, and S.A. Shamma. Auditory attention–focusing the searchlight on sound. *Current Opinions in Neurobiology*, 17(4):437–455, 2007.

[38] H. Fröhlich. Kernel methods in chemo- and bioinformatics. *Ph.D Thesis, University of Tübingen*, 2006.

[39] Y. Fujiwara, Y. Miyawaki, and Y. Kamitani. Estimating image bases for visual image reconstruction from human brain activity. In Y. Bengio, D. Schuurmans, J.D. Lafferty, C.I. Williams, and A. Culotta, editors, *Proc. of the Advances in Neural Information Processing Systems*, pages 576–584, Cambridge, MA, 2009. MIT Press.

[40] F. Galton. Regression towards mediocrity in hereditary stature. *The Journal of the Anthropological Institute of Great Britain and Ireland*, 15:246–263, 1886.

[41] A. Gammerman, V. Vovk, and V. Vapnik. Learning by transduction. In G.F. Cooper and S. Moral, editors, *Proc. of Uncertainty in Artificial Intelligence*, pages 148–155, Burlington, MA, 1998. Morgan Kaufmann.

[42] A. Gelman, J.B. Carlin, H.S. Stern, and D.B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, 2003.

[43] A.J. Glenstrup and T. Engell-Nielsen. Eye controlled media: Present and future state, 1995.

[44] A. Golugula, G. Lee, S.R. Master, M.D. Feldman, J.E. Tomaszewski, and A. Madabhushi. Supervised regularized canonical correlation analysis: Integrating histologic and proteomic data for predicting biochemical failures. *BMC Bioinformatics*, 12(3):483–495, 2011.

[45] M. Gönen and E. Alpaydın. Multiple kernel learning algorithms. *J. Mach. Learn. Res.*, pages 2211–2268, 2011.

[46] M.S. Grewal and A.P. Andrews. *Kalman Filtering : Theory and Practice Using MATLAB*. John Wiley and Sons, Inc., 2001.

[47] Dan Witzner Hansen and Qiang Ji. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(3):478–500, 2010.

[48] D. Hardoon, J. Shawe-Taylor, A. Ajanki, K. Puolamäki, and S. Kaski. Information retrieval by inferring implicit queries from eye movements. In *Proc. of the Int'l Conf. on Artificial Intelligence and Statistics*, 2007.

[49] D.R. Hardoon, S.R. Szedmak, and J. Shawe-taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, 2004.

[50] Y. Haruki, I. Homma, A. Umezawa, and Y. Masaoka. *Respiration and Emotion*. Springer, 2001.

[51] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, 2003.

[52] E.H. Hess and J.M. Polt. Pupil size in relation to mental activity during simple problem-solving. *Science*, 143(3611):1190–1192, 1964.

[53] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

[54] D.H. Hubel. *Eye, Brain, and Vision*. Scientific American Library, New York, NY, USA, 1988.

[55] I. Huopaniemi, T. Suvitaival, J. Nikkilä, M. Orešič, and S. Kaski. Multivariate multi-way analysis of multi-source data. *Bioinformatics*, 26(12):i391–i398, 2010.

[56] Yoshio Ishiguro, Adiyan Mujibiya, Takashi Miyaki, and Jun Rekimoto. Aided eyes: eye activity sensing for daily life. In *Proceedings of the Augmented Human International Conference (AH)*, pages 1–7, New York, NY, USA, 2010. ACM.

[57] T.S. Jaakkola. Tutorial on variational approximation methods. In *In Advanced Mean Field Methods: Theory and Practice*, pages 129–159, Cambridge, MA, 2000. MIT Press.

[58] R.J.K. Jacob. The use of eye movements in human-computer interaction techniques: what you look at is what you get. *ACM Transactions on Information Systems*, 9(2):152–169, 1991.

[59] T. Jarvenpaa and V. Aaltonen. Compact near-to-eye display with integrated gaze tracker. In *Proceedings of SPIE, vol 7001, SPIE, Bellingham, WA*, page 700106–1–700106–8, 2008.

[60] R.A. Johnson and D.W. Wichern. *Applied Multivariate Statistical Analysis*. Prentice Hall, 2007.

[61] M.A. Just and P.A. Carpenter. Eye fixations and cognitive processes. *Cognitive Psychology*, 8:441–480, 1976.

[62] F.H. Kanfer. Verbal rate, eyeblink, and content in structured psychiatric interviews. *Abnormal Social Psychology*, 61(11):341 – 347, 1960.

[63] S. Kaski and J. Peltonen. Informative discriminant analysis. In T. Fawcett and N. Mishra, editors, *In: Proc. of the Int'l Conf. on Machine Learning.*, pages 329–336, Menlo Park, CA, 2003. AAAI Press.

[64] J. Kim and E. André. Emotion recognition based on physiological changes in music listening. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(12):2067–2083, 2008.

[65] A. Klami and S. Kaski. Local dependent components. In *Proc. of the Int'l Conf. on Machine Learning*, pages 425–432, New York, NY, USA, 2007. ACM.

[66] A. Klami, C. Saunders, T.E. de Campos, and S. Kaski. Can relevance of images be inferred from eye movements? In *Proc. of the Int'l Conf. on Multimedia Information Retrieval*, pages 134–140, New York, NY, 2008. ACM.

[67] S. Koelstra, C. Mühl, M. Soleymani, A. Yazdani, J.S. Lee, T. Ebrahimi, T. Pun, A. Nijholt, and I. Patras. DEAP: A Database for emotion analysis using physiological signals. *IEEE Trans. Affective Computing*, 2011.

[68] X. Kong and G.F. Wilson. A new eog-based eyeblink detection algorithm. *Behavior Research Methods, Instruments, and Computers*, 30(4):713–719, 1998.

[69] L. Kozma, A. Klami, and S. Kaski. GaZIR: Gaze-based zooming interface for image retrieval. In J. Crowley, Y. Ivanov, and C. Wren, editors, *Proc. of the Int'l Conf. on Multimodal Interfaces*, pages 305–312, New York, NY, 2009. ACM.

[70] M. Kubat and S. Matwin. Addressing the curse of imbalanced training sets: One-sided selection. In D.H. Fisher, editor, *Proc. of the Int'l Conf. on Machine Learning*, pages 179–186, Burlington, MA, 1997. Morgan Kaufmann.

[71] M. Kumar and T. Winograd. Gaze-enhanced scrolling techniques. In *Proc. of the Symposium on User Interface Software and Technology*, pages 213–216, New York, NY, 2007. ACM.

[72] Matin L. and Pearce D.P. Three dimensional recording of rotational eye movements by a new contact-lens technique. *Biomed. Sei. Insirum.*, 2:79–95, 1964.

[73] R. Lall, M. J. Campbell, S. J. Walters, and K. Morgan. A review of ordinal regression models applied on health-related quality of life assessments. *Statistical methods in medical research*, 11(1):49–67, 2002.

[74] N. Lawrence. Gaussian process latent variable models for visualization of high dimensional data. In S. Thrun, S. Lawrence, and B. Schölkopf, editors, *Proc. of the Advances in Neural Information Processing Systems*, volume 15. MIT Press, 2003.

[75] N.D. Lawrence and J.C. Platt. Learning to learn with the informative vector machine. In E. Brodley C, editor, *Proc. of the Int'l Conf. on Machine Learning*, pages 65–72, New York, NY, 2004. ACM.

[76] N.D. Lawrence, M. Seeger, and R. Herbrich. Fast sparse Gaussian process methods: The informative vector machine. In S. Thrun, L.K. Saul, and B. Schölkopf, editors, *Proc. of the Advances in Neural Information Processing Systems*, pages 609–616, Cambridge, MA, 2003. MIT Press.

[77] Ying Liu, Dengsheng Zhang, Guojun Lu, and Wei-Ying Ma. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40(1):262 – 282, 2007.

[78] U. Lundberg, R. Kadefors, B. Melin, G. Palmerud, P. Hassmen, M. Engstrom, and I.E. Dohns. Psychophysiological stress and EMG activity of the trapezius muscle. *Int. J. Behav. Med.*, 1(4):354–370, 1994.

[79] D.J.C. MacKay. Bayesian interpolation. *Neural Computation*, 4:415–447, 1991.

[80] J.B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, Berkeley, CA, 1967. University of California Press.

[81] Oge Marques and Borivoje Furht. *Content-Based Image and Video Retrieval*. Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[82] S. Martinez-Conde, S.L. Macknik, and D.H. Hubel. The role of fixational eye movements in visual perception. *Nature Reviews Neuroscience*, 5(3):229–240, 2004.

[83] F.H. Martini and E.F. Bartholomew. *Essentials of Anatomy & Physiology*. Benjamin Cummings Inc., 2009.

[84] R.A. McFarland. Relationship of skin temperature changes to the emotions accompanying music. *Applied Psychophysiology and Biofeedback*, 10:255–267, 1985.

[85] J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philos. Trans. Royal Soc. (A)*, 83(559):69–70, 1909.

[86] T.P. Minka. Expectation propagation for approximate Bayesian inference. In J. Breese and D. Koller, editors, *Proc. of Uncertainty in Artificial Intelligence*, pages 362–369, San Francisco, CA, 2001. Morgan Kaufmann.

[87] K.-R. Müller, S. Mika, G. Ratsch, K. Tsuda, and B. Schölkopf. An Introduction to Kernel-based Learning Algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–201, 2001.

[88] E. Niedermeyer and F.L. da Silva. *Electroencephalography: Basic principles, clinical applications, and related fields*. Lippincott Williams & Wilkins, 2004.

[89] J. Nielsen. Noncommand user interfaces. *Communications of the ACM*, 36(4):83 – 99, 1993.

[90] S. Nilsson, T. Gustafsson, and P. Carleberg. Hands free interaction with virtual information in a real environment. In *Proceedings of The Conference on Communications by Gaze Interaction (COGAIN)*, pages 53–57, 2007.

[91] Hyung Min Park, Seok Han Lee, and Jong Soo Choi. Wearable augmented reality system using gaze interaction. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 175–176, Washington, DC, USA, 2008. IEEE.

[92] K. Pearson. Mathematical contributions to the theory of evolution. III. Regression, heredity, and panmixia. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 187:253–318, 1896.

[93] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572, 1901.

[94] R.W. Picard, E. Vyzas, and J. Healey. Toward machine emotional intelligence: Analysis of affective physiological state. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(10):1175–1191, 2001.

[95] A. Piolat, T. Olive, J. Roussey, O. Thunin, and J.C. Ziegler. SCRIPTKELL: A tool for measuring cognitive effort and time processing in writing and other complex cognitive activities. *Behavior Research Methods*, 31(1):113–121, 1999.

[96] J.C. Platt. Advances in Kernel Methods. chapter Fast training of support vector machines using sequential minimal optimization, pages 185–208. MIT Press, Cambridge, MA, 1999.

[97] P. Qvarfordt and S. Zhai. Conversing with the user based on eye-gaze patterns. In G. van der Veer, editor, *Proc. of Human Factors in Computing Systems*, pages 221–230, New York, NY, 2005. ACM.

[98] P. Rai and H. Daumé III. Multi-label prediction via Sparse Infinite CCA. In D. Koller, Y. Bengio, L. Bottou, and A. Culotta, editors, *Proc. of the Advances in Neural Information Processing Systems*, Cambridge, MA, 2009. MIT Press.

[99] R. Raina, A. Battle, H. Lee, B. Packer, and A.Y. Ng. Self-taught learning: Transfer learning from unlabeled data. In Z. Ghahramani, editor, *Proc. of the Int'l Conf. on Machine Learning*, pages 759–766, New York, NY, 2007. ACM.

[100] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. More efficiency in multiple kernel learning. In Z. Ghahramani, editor, *Proc. of the Int'l Conf. on Machine Learning*, pages 775–782, New York, NY, 2007. ACM.

[101] C.E. Rasmussen. The infinite Gaussian mixture model. In T.K. Leen, T.G. Dietterich, and V. Tresp, editors, *Proc. of the Advances in Neural Information Processing Systems*, pages 554–560, Cambridge, MA, 2000. MIT Press.

[102] C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[103] K. Rayner. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin*, 124(3):372–422, 1998.

[104] C. J. Van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, 1979.

[105] C.P. Robert and G. Casella. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[106] D.B. Rubin. Iteratively reweighted least squares. *Encyclopedia of Statistical Sciences*, 4:272–275, 1983.

[107] J. A. Russell. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6):1161–1178, 1980.

[108] N Savva and N Bianchi-Berthouze. Automatic recognition of affective body movement in a video game scenario. In *Int'l Conf. on Intelligent Technologies for Interactive Entertainment*, pages 149 – 158, 2011.

[109] B. Scholkopf and A.J. Smola. *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT Press, 2001.

[110] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[111] H. Shimodaira, K. Noma, M. Nakai, and S. Sagayama. Dynamic time-alignment kernel in support vector machine. In T.G. Dietterich, Suzanna Becker, and Zoubin Ghahramani, editors, *Proc. of the Advances in Neural Information Processing Systems*, pages 921–928, Cambridge, MA, 2001. MIT Press.

[112] J.P. Sinno and Y. Qiang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[113] S. Stellmach and R. Dachselt. Investigating gaze-supported multimodal pan and zoom. In C.H. Morimoto and H. Istance, editors, *Proc. of the Symposium on Eye Tracking Research and Applications*, pages 357–360, New York, NY, 2012. ACM.

[114] P. Sundström, A. Ståhl, and K. Höök. eMoto: Affectively involving both body and mind. In G. van der Veer, editor, *Extended Abstracts on Human Factors in Computing Systems*, pages 2005–2008, New York, NY, 2005. ACM.

[115] H. Tanabe, T.B. Ho, C.H. Nguyen, and S. Kawasaki. Simple but effective methods for combining kernels in computational biology. In *Proc. of the Int'l Conf. on Research, Innovation and Vision for the Future*, pages 71–78, New York, NY, 2008. IEEE.

[116] D. Tennenhouse. Proactive computing. *Communications of ACM*, 43(5):43–50, 2000.

[117] M.E. Tipping. Sparse Bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.*, 1:211–244, 2001.

[118] M. Titsias, N.D. Lawrence, and M. Rattray. Efficient sampling for Gaussian process inference using control variables. In D. Koller, Y. Bengio D. Schuurmans, and L. Bottou, editors, *Proc. of the Advances in Neural Information Processing Systems*, pages 1681–1688, Cambridge, MA, 2008. MIT Press.

[119] M. Titsias and M. Lázaro-Gredilla. Spike and slab variational inference for multi-Task and multiple kernel learning. In J. Shawe-Taylor, R.S. Zemel, P.L. Bartlett, and K.Q. Weinberger F. Pereira, editors, *Proc. of the Advances in Neural Information Processing Systems*, pages 2339–2347, Cambridge, MA, 2011. MIT Press.

[120] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., 1995.

[121] R. Vertegaal and J.S. Shell. Attentive user interfaces: the surveillance and sousveillance of gaze-aware objects. *Social Science Information*, 47(3):275–298, 2008.

[122] Roel Vertegaal. Designing attentive interfaces. In *Proc. of the Symposium on Eye Tracking Research and Applications*, pages 23–30, New York, NY, USA, 2002. ACM.

[123] J. Viinikanoja, A. Klami, and S. Kaski. Variational Bayesian mixture of robust CCA models. In J. Balcázar, F. Bonchi, A. Gionis, and M. Sebag, editors, *Proc. of the European Conf. on Machine Learning and Knowledge Discovery in Databases*, pages 370–385, Berlin, Heidelberg, 2010. Springer-Verlag.

[124] H. Vinod. Canonical ridge and econometrics of joint production. *Journal of Econometrics*, 4(2):147 – 166, 1976.

[125] S. Virtanen, A. Klami, and S. Kaski. Bayesian CCA via group sparsity. In L. Getoor and T. Scheffer, editors, *Proc. of the Int'l Conf. on Machine Learning*, pages 457–464, New York, NY, 2011. ACM.

[126] C. Wang. Variational Bayesian approach to canonical correlation analysis. *IEEE Transactions on Neural Networks*, 18:905 – 910, 2007.

[127] Z. Wang, Y. Song, and C. Zhang. Transferred dimensionality reduction. In W. Daelemans, B. Goethals, and K. Morik, editors, *Proc. of the European Conf. on Machine Learning and Knowledge Discovery in Databases*, pages 550–565, Berlin, Heidelberg, 2008. Springer-Verlag.

[128] D. J. Ward and D. J. C. Mackay. Fast Hands-free writing by gaze direction. *Nature*, 418(6900), 2002.

[129] J. Weston and C. Watkins. Multi-class support vector machines. In *Proc. of European Symposium on Artificial Neural Networks*, 1999.

[130] C.K.I. Williams and D. Barber. Bayesian classification with Gaussian processes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(12):1342–1351, 1998.

[131] F. Yger, M. Berar, G. Gasso, and A. Rakotomamonjy. Adaptive canonical correlation analysis based On matrix manifolds. In A. McCallum, J. Langford, and J. Pineau, editors, *Proc. of the Int'l Conf. on Machine Learning*, New York, NY, 2012. ACM.

[132] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68:49–67, 2006.

[133] Z. Zeng, M. Pantic, G.I. Roisman, and T. S. Huang. A survey of affect recognition methods: Audio, visual and spontaneous expressions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(1):39–58, 2009.

Aalto-DD 61/2013

ISBN 978-952-60-5116-1
ISBN 978-952-60-5117-8 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934
ISSN 1799-4942 (pdf)

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

BUSINESS + ECONOMY

ART + DESIGN + ARCHITECTURE

SCIENCE + TECHNOLOGY

CROSSOVER

DOCTORAL DISSERTATIONS