Learning for the Internet: Kernel Embeddings and Optimisation

Novi Quadrianto

August 2011

A thesis submitted for the degree of Doctor of Philosophy
of the Australian National University

To my parents,
Yusdhi Haris and Maria Haris,
for their unwavering love.

Declaration

The work in this thesis is my own except where otherwise stated.

Novi Quadrianto

Wien, Österreich

August, 2011

Acknowledgements

I first would like to thank the members of my committee: Tibério S. Caetano, Wray L. Buntine, Dale Schuurmans, Christoph Lampert, and my advisor Alex J. Smola. I always find myself at peace after talking to Tiberio. Be research problems or life problems, without fail, Tiberio finds the time to listen and give advices to me. I remember my very first conference presentation at ICML in Helsinki: Tiberio with his smile at the first row at the noon of my presentation to bring my heart pumping rate back at normal. I thank Wray for agreeing the troubles of chairing my supervisory committee, for always being eager to share with me his current research affairs (though we are presently in different research interests), and for almost giving me cooking lessons. I thank Dale for teaching me optimisation methods, and for allowing me to experience a real winter in Edmonton. I thank Christoph for supervising the last 10 months of my studies in a home away from home, for patiently listening to and constructively criticising my half-baked research ideas, and for telling me and everyone else that I give him pains and headaches.

I would like to express my utmost gratitude to my advisor Alex Smola. I admire his brilliance, determination, and always hungry of knowledge. Though I only managed to be closely supervised by him in Canberra, Australia for 9 months, his influence throughout my thesis is undoubtful. The next level of supervision I get from Alex is so called an *online* supervision via Skype. I am thankful to him for always being available for me. This phenomenon is best explained by Alex himself: 'Even my wife can do 100% prediction on who is calling in the middle of the night'. Alex taught me some life lessons as well. I recall his dinner treat at NIPS in Vancouver and he asked me to remember how happy I was getting a treat from a supervisor and that I should do the same in the future. I feel deeply indebted to Alex and a huge portion of the knowledge and confidence that I have today is due to him.

I would like to thank my close collaborator and dear mentor, Kristian Kersting. I first met Kristian when he visited NICTA in September 2008. From then on, I enjoy every single research conversations we have at a *weekly* Skype call. I thank him for two memorable research visits in Germany in February 2009 and August 2010. I learn from Kristian on how to do an efficient research, to sell research ideas, and not least importantly how to play fussball and to live in the feeling like Texas Chain Saw Massacre house. My conference experience with Kristian and his group at ECML in Barcelona has been the most enjoyable one.

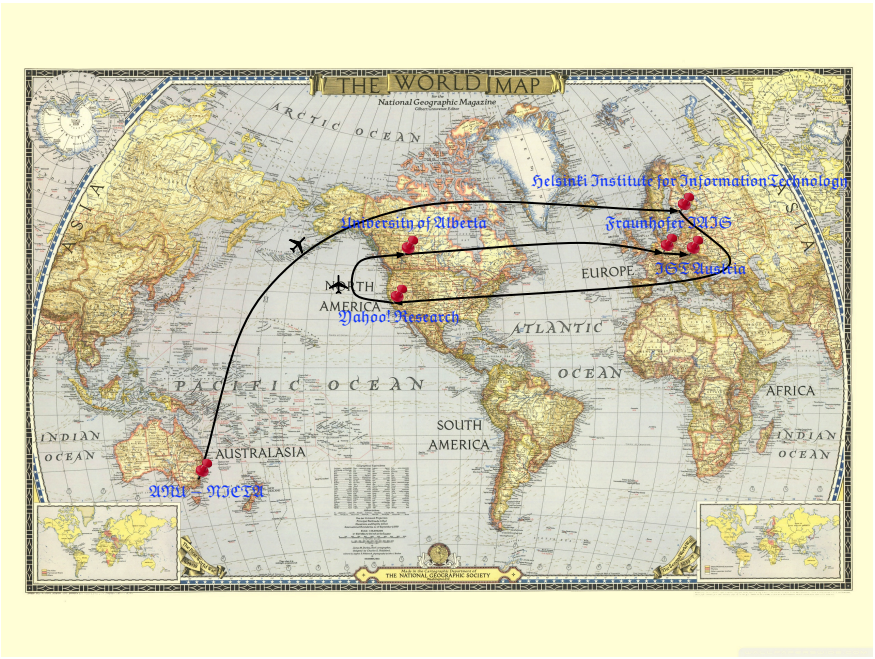


Figure 1: PhD Life Journey

During my PhD studies in Canberra, Australia, I am quite fortunate to have several overseas research experiences, and to be able to interact with great researchers all over the world. I would like to thank my mentors: Petri Myllymäki and Patrik Floréen at HIIT, Helsinki, Finland (a 1-month visit in February 2009), Alex Smola and Kostas Tsioutsiouliklis at Yahoo! Research, Santa Clara, US (a 4-month visit from August 2009), Dale Schuurmans at University of Alberta, Edmonton, Canada (a 3-month visit from January 2010), Kristian Kersting at Fraunhofer IAIS, Sankt Augustin, Germany (a 2-month visit from August 2010), and Christoph Lampert at IST Austria, Vienna, Austria (a 10-month visit from October 2010). I thank my mentors and their institutions for fully funding my research visits.

My friends and collaborators at ANU-NICTA, HIIT, University of Alberta, Yahoo! Research, IST Austria, and Fraunhofer IAIS: Babak Ahmadi, Akshay Asthana, Marconi Barbosa, Debdeep Banerjee, Christian Bauckhage, Sourav Bhattacharya, Edwin Bonilla, Chao Chen, Li Cheng, Lan Du, Kishor Gawande, Fabian Hadiji, Morten Bojsen-Hansen, Marcus Hutter, Cong Phuoc Huynh, Ahmed Jawad, Dmitry Kamenetsky, Filip Korč, Gilbert Leung, Jason Li, John Lim, Quoc Viet Le, Paul Malcolm, Julian McAuley, Sriraam Natarajan, Marion Neumann, Petteri Nurmi, James Petterson, Mark Reid, Scott Sanner, Peter Sunehag, Viktoriia Sharmanska, Le Song, Choonhui Teo, Christian Thurau, Tatiana Tommasi, Antti Tuominen, Tinne Tuytelaars, Owen Thomas, Taneli Vähakangas, Mirwaes Wahabzada, Markus Weimer, Chris Webers, Zhao Xu, Jin Yu, Xinhua Zhang. I am sorry if I neglected to mention anyone.

I would like to thank the ANU Research School of Computer Science (formely RSISE) and the ANU College of Engineering and Computer Science administration staffs, especially Michelle Moravec, Di Kossatz, Suzanne Van Haeften, Debbie Pioch, Marie Katselas, and Jonathan Peters for helping me focus on hedonistic research pleasure, and for being so patient in handling all administrative disorder due to my research visits.

I would like to thank my mother Maria, my father Yusdhi, my brother Raymond, and my sister Silvia. My dear family helped shape me into who I am today. I am thankful for their support and confidence in me.

This work was made possible through scholarship by ANU and NICTA, fellowship from Microsoft Research, travel grants from the NIPS foundation, ICML conference, ANU Vice-Chancellor and PASCAL Network of Excellence.

Abstract

In this thesis, we develop principled machine learning methods suited for complex real-world Internet challenges. The Internet has supplied an unprecedented amount of data; the challenge now is to transform this massive amount of data into information that supports knowledge creation. Machine learning techniques have become prevalent for modelling, prediction, and decision making from massive scale data. This thesis makes contributions in addressing data to knowledge transformation in the context of machine learning; we introduce non-standard machine learning problems and devise solutions for those, as well we present scalable solutions for several existing machine learning problems.

The present work focuses on addressing Internet complexity on output label dimensions. The first part of this thesis deals with formulation and solution of *non-standard* machine learning problems. Traditionally, supervised machine learning settings draw inference and make prediction from a set of input objects; each of which is supervised by a desired output value. Internet poses challenges of weak label supervision and label inconsistency. We focus on the following three new settings:

1. Learning from only label proportions: A learning setting where instead of each input is supervised with an output, we are given groups of unlabelled inputs. Each group is endowed with information on class label proportions. The number of group is at least as many as number of labels (Chapter 3).

2. Learning input-output correspondences: A learning setting where a set of inputs and a set of outputs are given however they are not paired (Chapter 4).

3. Learning from several related tasks with distinct label sets: A learning setting where several related tasks are given however each task has potentially distinct label sets and label correspondences are not readily available (Chapter 5).

The second part addresses refinements of existing machine learning models and algorithms to *scale* to large data. The contributions of this thesis include a streaming algorithm for the following two problems:

1. Transductive learning: We present a scalable algorithm for learning with labelled and unlabelled data simultaneously by matching the output distributions on labelled and unlabelled data (Chapter 6).

2. Storage and indexing management: We present a scalable algorithm for indexing in the context of webpage tiering. The goal is to allocate pages to caches such that the most frequently accessed pages reside in the caches with the smallest latency whereas the least frequently retrieved pages are stored in the backtiers of the caching system. This indexing and storage problem is related to a larger class of parametric flow problem (Chapter 7).

The solutions presented in this thesis are centred around two main mathematical ingredients. First, Hilbert Space embeddings of distributions via averages are used. This allows distance computation between data distributions in terms of distances between averages, which, in turn, yield elegant ways to deal with distributions without the need of estimating them as an intermediate step. Second, recent advances in field of optimisation are exploited to address the sheer size and the non-convex nature of mostly Internet problems.

Keywords Weak Label Supervision, Label Inconsistency, Transductive Learning, Parametric Maximum Flow, Streaming Algorithms, Kernel Embeddings, Optimisation;

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The amount of Internet data corresponds to stack of books stretching from Earth to Pluto 10 *times.*
Richard Wray

Approximately 1.5 billion people use the Internet. People write news articles, blogs, and reviews; people upload videos, audios, and photos. People become web content creators. This directly translates to the availability of half a *Zettabyte* of data. Synergistically with rapid progress in *machine learning* models and algorithms, as well as rapid rises in computing power and storage, the challenge of the 21st century consists of finding ways to transform this complex massive yet noisy and sparse Internet data coming from a variety of sources into insights in support of knowledge creation. This thesis makes contributions in addressing data to knowledge transformation in the context of machine learning; we introduce non-standard machine learning problems and devise solutions for those, as well we present scalable solutions for several existing machine learning problems.

Machine learning techniques have become prevalent for drawing inference and making prediction from massive scale data. Given input-output data pairs, the goal of learning is to infer a latent function that maps inputs to outputs. This function will then be used to predict an output for a given unseen input. In classification, the most frequently applied setting of machine learning and the one that is considered in this thesis, the output is the set of possible labels. Other machine learning settings include regression, structured estimation and novelty detection and variants of classification such as online and batch learning, transductive learning, co-training, active learning, among others. Consider as an illustrative example, a task of categorising web videos (user-generated videos from video sharing websites). Here the inputs are the web videos and the outputs

are the categories such as entertainment, music, news and politics, science and technology, among others.

The complex nature of Internet data manifests itself along both the input (feature) and output (label) dimensions. On the input dimensions, we deal not only with potentially millions of features but also the features might come from multiple modalities or data sources. Web videos admit the conventional representation of audio-visual features, the associated text (the filenames, titles, or descriptions) and even the intricate social network representation (the relationship among videos through the users, links, or recommendations). On the label dimensions, the information is sparse. For instance, there might be millions of web videos but only a few of them are labelled by a particular user or labelled with a particular tag. Adding to the sparsity challenge, Internet data tend to have multiple sets of different labels. In the case of our illustrative example, the categorisation of web videos from several different video sharing services depends heavily on the editors of each web service. Different editors have very different perception of video categories, thus the label categories are often 'inconsistent'. The focus of this thesis is on Internet applications for the following two reasons:

- The Internet offers new challenges that do not naturally fit into existing machine learning methods;
- The Internet requires large-scale solution to problems.

1.1 Thesis Contribution

This thesis aims to address research challenges for Internet applications in the context of machine learning. Challenging web applications lead to development and investigation of non-standard machine learning settings. The present work focuses on addressing Internet complexity on output label dimensions. Specifically, we introduce problems and present models and algorithms for the following two *non-standard* learning frameworks:

- Weak label supervision
  Traditional classification setting infers a statistical model based on observed input-output data pairs. We introduce a new problem where instead of each input is supervised with an output, we are given groups of unlabelled inputs (Chapter 3). Each group is endowed with information on class label proportions. The number of groups is at least as many as number of class

labels. This seemingly contrived setting has plethora of applications in areas like politics, spam filtering, e-commerce, and improper content detection. We also introduce a learning framework where a set of inputs and a set of outputs are given however they are not paired (Chapter 4). The goal of learning is now to infer a correspondence or a permutation that maps each input to its output. This has applications in data visualisation, image search browsing, photo album summarisation, cross-domain matching, to name a few.

- Label inconsistency
  In machine learning it is folk knowledge that if several prediction tasks are related, then learning them simultaneously can improve performance. For instance, a web videos categoriser trained with data from several different video sharing sites is likely to be more accurate than one that is trained with data from a single video site. We introduce a new setting of jointly learning several related tasks where each task has potentially distinct label sets and label correspondences are not readily available (Chapter 5). This is in contrast with existing settings which either assume that the label sets shared by different tasks are the same or that there exists a label mapping oracle.

The above contributions on new machine learning settings constitute the first part of this thesis. The second part deals with refinements of existing machine learning models and algorithms to *scale* to large data. The contributions of this thesis include a scalable algorithm for the following two problems:

- Transductive learning
  Internet data, while very large, is very sparse on its label, i.e. only a minute amount of them are human annotated. In the transductive setting, this unlabelled data is harnessed to improve the performance of classifier simply trained on annotated data. We present a transductive algorithm exploiting a simple fact that the distributions over the outputs on annotated and unannotated data should match (Chapter 6). As our solution is amenable to an online optimisation method, it can process received data one at a time and then discard it in an excess data stream.

- Storage and indexing management
  Finding a needle in a haystack best describes a process of locating relevant data points in monstrous Internet space. Thus, each data point needs to

have a label index before it is stored for a later efficient retrieval. We propose an algorithm for webpage tiering for search engine indices that can process billion of webpages in seconds (Chapter 7). Our presented algorithm solves an integer linear program in an online fashion. This indexing and storage problem is related to a larger class of parametric maximum flow problem and therefore our algorithm has potential applications also in those problems.

The set of publications related to this thesis are listed below:

1. N. Quadrianto, A. J. Smola, T. S. Caetano, Q. V. Le. Estimating Labels from Label Proportions. *International Conference on Machine Learning* ***ICML***, 2008 (Quadrianto et al., 2008);

2. N. Quadrianto, L. Song, A. J. Smola. Kernelized Sorting. *Advances in Neural Information Processing Systems* ***NIPS*** *21*, 2008 (Quadrianto et al., 2009b);

3. N. Quadrianto, J. Petterson, A. J. Smola. Distribution Matching for Transduction. *Advances in Neural Information Processing Systems* ***NIPS*** *22*, 2009 (Quadrianto et al., 2009e);

4. N. Quadrianto, A. J. Smola, T. S. Caetano, Q. V. Le. Estimating Labels from Label Proportions. *Journal of Machine Learning Research* ***JMLR***, vol. 10, 2009 (Quadrianto et al., 2009a);

5. N. Quadrianto, A. J. Smola, L. Song, T. Tuytelaars. Kernelized Sorting. *IEEE Trans. on Pattern Analysis and Machine Intelligence* ***PAMI***, vol. 32, 2010 (Quadrianto et al., 2010c);

6. N. Quadrianto, A. J. Smola, T. S. Caetano, S.V.N. Vishwanathan, J. Petterson. Multitask Learning without Label Correspondences. *Advances in Neural Information Processing Systems* ***NIPS*** *23*, 2010 (Quadrianto et al., 2010d);

7. G. Leung, N. Quadrianto, A. J. Smola, K. Tsioutsiouliklis. Optimal Web-scale Tiering as a Flow Problem. *Advances in Neural Information Processing Systems* ***NIPS*** *23*, 2010 (Leung et al., 2010);

8. N. Quadrianto, K. Kersting, T. Tuytelaars, W. L. Buntine. Beyond 2D-Grids: A Dependence Maximization View on Image Browsing. *ACM International Conference on Multimedia Information Retrieval* ***MIR***, 2010 (Quadrianto et al., 2010a);

9. N. Quadrianto and C.H. Lampert. Learning Multi-View Neighborhood Preserving Projections, *International Conference on Machine Learning* ***ICML***, 2011 (Quadrianto & Lampert, 2011a).

Besides the above contributions, the author has also carried out research in other areas such as Gaussian processes regression and state estimation. For the reason of consistency, these contents are not included in this thesis. The research contributions for Gaussian processes regression include:

1. N. Quadrianto, K. Kersting, M. D. Reid, T. S. Caetano, W. L. Buntine. Kernel Conditional Quantile Estimation via Reduction Revisited. *IEEE International Conference on Data Mining* ***ICDM***, 2009 (Quadrianto et al., 2009d);

2. A. Asthana, R. Goecke, N. Quadrianto, T. Gedeon. Learning based Automatic Face Annotation for Arbitrary Poses and Expressions from Frontal Images Only. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* ***CVPR***, 2009 (Asthana et al., 2009).

The research contributions for state estimation include:

1. W. P. Malcolm, N. Quadrianto, L. Aggoun. State Estimation Schemes for Independent Component Coupled Hidden Markov Models. *Journal of Stochastic Analysis and Applications*, vol. 28, 2010 (Malcolm et al., 2010);

2. N. Quadrianto, T. S. Caetano, J. Lim, D. Schuurmans. Convex Relaxation of Mixture Regression with Efficient Algorithms. *Advances in Neural Information Processing Systems* ***NIPS*** *22*, 2010 (Quadrianto et al., 2009c).

The author has also contributed book chapters covering widely used machine learning techniques. Those manuscripts include:

1. N. Quadrianto and C. H. Lampert. Kernel-based Learning. *Encyclopedia of Systems Biology*, Springer, 2011 (Quadrianto & Lampert, 2011b);

2. N. Quadrianto, K. Kersting, Z. Xu. Gaussian Processes. *Encyclopedia of Machine Learning*, Springer, 2010 (Quadrianto et al., 2010b);

3. N. Quadrianto and W. L. Buntine. Regression. *Encyclopedia of Machine Learning*, Springer , 2010 (Quadrianto & Buntine, 2010c);

4. N. Quadrianto and W. L. Buntine. Linear Regression. *Encyclopedia of Machine Learning*, Springer, 2010 (Quadrianto & Buntine, 2010b);

5. N. Quadrianto and W. L. Buntine. Linear Discriminant. *Encyclopedia of Machine Learning*, Springer, 2010 (Quadrianto & Buntine, 2010a).

1.2 Thesis Structure

The rest of this thesis will be organized into seven chapters. The main contents of each chapter are summarised below:

Chapter 2 Background In this chapter, we will cover background knowledge needed for our later theory and algorithm development. This includes a brief tour to kernel methods. Subsequently, we list essential properties of kernels. Particularly, we put emphasis on the reproducing kernel Hilbert spaces and universal kernels. These two aspects of kernels will play a key role in the subsequent kernel embeddings approach for distribution analysis. Optimisation lies at almost every heart of machine learning problems. In the last three sections, we pile up necessary optimisation background including convex optimisation, non-convex optimisation (convex-concave procedure) and stochastic optimisation (stochastic gradient descent). The dependencies of subsequent chapters and the sections in this background chapter are depicted in Figure 1.1.

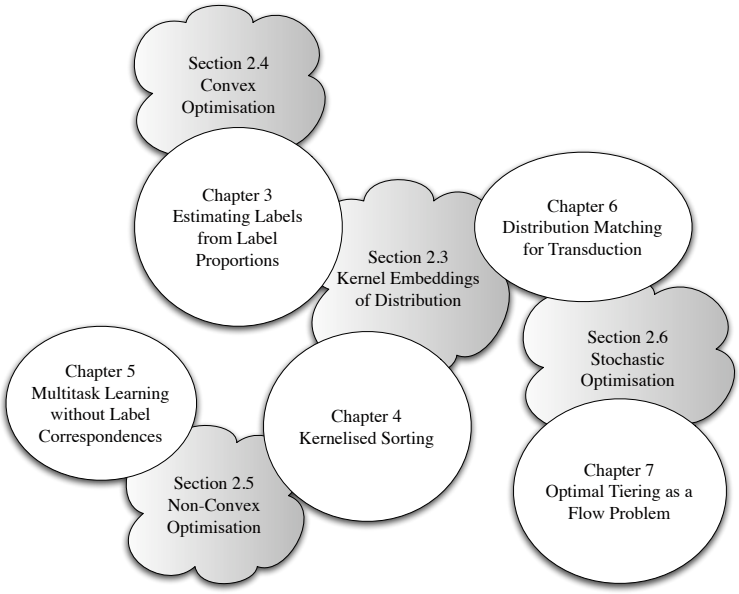


Figure 1.1: Dependency Diagram of Chapters and Background Sections. Overlaps denote dependencies.

Part I: Formulation and Solution for Non-Standard Machine Learning Problems

Chapter 3 Estimating Labels from Label Proportions In this chapter, we introduce a learning setting where we are given sets of unlabelled observations, each set with known label proportions; the goal is to predict the labels of another set of observations, possibly with known label proportions. This setting occurs in areas like e-commerce, politics, spam filtering and improper content detection. We present consistent estimators which can reconstruct the correct labels with high probability in a uniform convergence sense. Our method works by modelling a conditional exponential likelihood and approximating the unknown mean of sufficient statistics. The approximation is done by exploiting the convergence properties of a sample mean operator to its population counterpart and by solving a linear system of equations formed by the known proportions. Experiments on benchmark datasets show that our method works well in practice.

Chapter 4 Kernelised Sorting In this chapter, we introduce a learning framework where a set of data inputs and a set of data outputs are given however they are not paired; the goal of learning is to infer the latent input-output correspondences. This is achieved by maximising the dependency between input-output pairs of observations. We use a kernel embeddings based dependency measure called the Hilbert Schmidt Independence Criterion. This problem can be cast as one of maximising a quadratic assignment objective with special structure and we present a simple algorithm for finding a locally optimal solution to it. We show applications of this setting in data visualisation, photo album summarisation, image search engine browser, estimation and cross-domain matching.

Chapter 5 Multitask Learning without Label Correspondences In this chapter, we introduce a setting of jointly learning several related tasks where each task has potentially distinct label sets, and label correspondences are not readily available. Our method directly maximises the mutual information among the labels, and we show that the resulting objective function can be decomposed into a difference of convex functions and thus is amenable to an efficient optimisation via the convex-concave procedure. Our proposed approach has a direct application for data integration with different label spaces, and we show one such application in integrating Yahoo! and DMOZ web directories.

Part II: Scalable Solution for Existing Machine Learning Problems

Chapter 6 Distribution Matching for Transduction In this chapter, we present a scalable algorithm for transductive learning, learning with labelled and unlabelled data simultaneously, based on distribution matching. We developed an algorithm that exploits the fact that for a good classifier, the distributions over the outputs on labelled data and unlabelled data should match. We cast the goal as a two-sample problem which can be solved efficiently by using a distance measure in Hilbert Space. One key advantage of our approach is that it is 'plug and play', i.e. it is applicable to all estimation problems ranging from classification and regression to structured estimation by just defining appropriate loss functions. Further, by formulating our solution as an online optimisation method, our approach scales well and this was demonstrated in our experiments by solving a multi-category problem with millions of observations.

Chapter 7 Optimal Tiering as a Flow Problem In this chapter, we present a scalable algorithm for performing storage and indexing management. Our algorithm solves an integer linear program in an online fashion. It exploits total unimodularity of the constraint matrix and a Lagrangian relaxation to solve the problem as a convex online game. The algorithm generates approximate solutions of maximum flow problems by performing stochastic gradient descent on a set of flows. We apply the algorithm to optimise tier arrangement of over 84 million web pages on a layered set of caches to serve an incoming query stream optimally.

Chapter 8 Conclusion and Future Directions In this chapter, we summarise the main results in this thesis. We also discuss some future directions in developing machine learning models and algorithms for complex real-world Internet challenges. These include addressing Internet complexity on input feature dimensions.

Chapter 2

Background

In this chapter, we will cover essential background knowledge needed for our later models and algorithms development for addressing Internet challenges. We will start with the concept of kernels as measures of similarity, discuss the theoretical properties of kernels, and list a number of kernel examples. We then outline several estimation and learning methods on classification, regression and dimensionality reduction problems utilizing these kernels. We put emphasis on the application of kernels as an embedding approach for distribution analysis. The last three sections are dedicated to the background on mathematical programming including convex, non-convex, and stochastic optimisation.

2.1 Kernels

Kernels (Aronszajn, 1950) are non-linear measures of pairwise similarity between arbitrary data objects that generalise the Euclidean inner product, which is a linear measures of similarity between two vectors. By use of the so-called *kernel trick* we can obtain a non-linear version of many linear methods for data analysis (vide Section 2.2): first, we formulate the method in a way that refers to the data only in term of inner product between data points. Subsequently, we replace all occurrences of the inner products by evaluations of the kernel function.

From a practical point of view, the most favourable properties of kernel-based learning methods are their *modularity*, i.e. the complete detachment between the choice of similarity function encoded by kernels and the design of the algorithm itself, and *flexibility*, i.e. the possibility to compute similarity between discrete objects such as strings, trees, and graphs.

Assume that we have a set of m objects $\mathcal{D} = \{x_1, \ldots, x_m\}$ at hand, for example

the web videos. We denote the set of all possible objects $\mathcal{X}$, such that $x_i \in \mathcal{X}$ for $i = 1, \ldots, m$. Our dataset is drawn independently and identically distributed (i.i.d.) from an unknown probability distribution P on $\mathcal{X}$ with a density $p(x)$. Prior to performing any analysis on this data, the first issue that needs to be tackled is how to represent this set of data $\mathcal{D}$, that is, define a representation function $\phi : \mathcal{X} \to \mathcal{F}$ for each possible object $x \in \mathcal{X}$. Further data analysis is subsequently performed on this set of representations $\{\phi(x_1), \ldots, \phi(x_m)\}$. Most classical techniques work only if the feature space $\mathcal{F}$ is a vector space of finite dimension, i.e. $\mathbb{R}^d$. This is a significant restriction, as it means, for example, that we cannot represented a web video by the *variable length* sequence of time series information, but we have to resort to feature representation or similar procedures that come with a loss of information.

2.1.1 From Data Representation to Similarity Matrix

Kernel-based learning overcomes the limitation that data must be represented as finite dimensional vectors. It no longer probes the objects based on their individual data representation but through a set of similarity measures between pairs of objects (refer to Figure 2.1). That is, a set of m data points is translated into a symmetric $m \times m$ similarity matrix $K \in \mathbb{R}^{m \times m}$ with $K_{i,j} = k(x_i, x_j)$ and $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is the positive definite kernel function. All kernel methods rely exclusively on such square similarity matrices. This is particularly appealing in many applications in which comparing two objects is often an easier task than representing the object by a finite sized vector.

Any similarity measure can be used as part of a kernel-based learning method if it fulfils the properties of a positive definite kernel function.

Definition 1 (Positive Definite Kernels) *Let $\mathcal{X}$ be a nonempty set. A symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a positive definite kernel on $\mathcal{X}$ if*

$$\sum_{i,j=1}^{m} c_i \, c_j \, k(x_i, x_j) \geq 0, \tag{2.1}$$

for all $m \in \mathbb{N}$, $x_1, \ldots, x_m \in \mathcal{X}$ and $c_1, \ldots, c_m \in \mathbb{R}$.

We will hereafter denote positive definite kernels simply as kernels. Note that the positive definiteness of a kernel function translates into the positive definiteness of the similarity matrix K, which is also called a Gram matrix. This positive definiteness has an important implication not only on the theoretical side but also on the effectiveness of kernel methods in many practical applications.

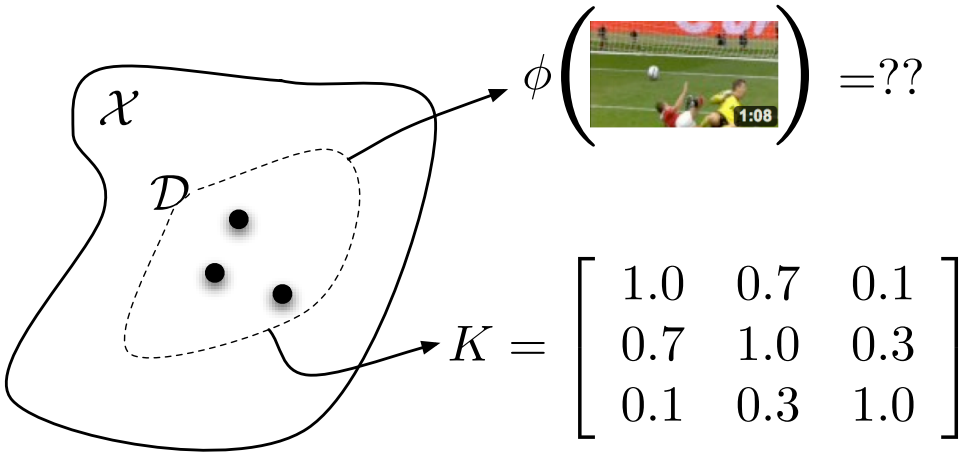


Figure 2.1: Two different ways of representing the same data set. $\mathcal{X}$ is the set of user-generated videos from YouTube$^{\text{TM}}$ and $\mathcal{D}$ is a data set of three particular videos. Traditionally, a representation $\phi(x)$ for each element of $x \in \mathcal{X}$ first needs to be defined prior to any data analysis which sometimes might not be trivial (`upper right`). Kernel methods represent the same data set in term of a matrix of pairwise similarity between elements, irrespective of any complexity associated with each element (`lower right`).

2.1.2 Examples of Kernels

In this thesis, we will mainly deal with vectorial data. Table 2.1 gives some examples of a kernel function defined on vectors. All the functions listed in the table are symmetric and positive definite, thus constitute a kernel function. Besides vectors, kernels have also been defined on discrete objects such as strings, trees, and graphs. For more comprehensive examples on kernels, we refer interested readers to (Schölkopf & Smola, 2002; Shawe-Taylor & Cristianini, 2004; Schölkopf et al., 2004; Hofmann et al., 2008).

2.1.3 Reproducing Kernel Hilbert Spaces

We will now establish a theoretical foundation of kernel as an *implicit* way to construct a representation function ϕ. Calculating the similarity value between two objects $x_1, x_2 \in \mathcal{X}$ by means of a kernel function is equivalent to forming vector representation $\phi(x_1), \phi(x_2) \in \mathcal{F}$ and evaluating their inner product in the feature space $\mathcal{F}$. However, by using a kernel, the vector representation is only present implicitly, and we do never have to compute it explicitly. This is a

Name	$k(x, x')$	θ
linear kernel	$\langle x, x' \rangle$	NA
polynomial kernel	$(\langle x, x' \rangle + c)^d$, $c > 0$, $d \in \mathbb{N}$	$\{c, d\}$
Gaussian kernel	$\exp\left(-\frac{1}{\sigma^2} \|x - x'\|^2\right)$	$\{\sigma\}$
Laplace kernel	$\exp\left(-\frac{1}{\sigma^2} \|x - x'\|\right)$	$\{\sigma\}$
delta kernel	$\mathbf{1}_{(x=x')} = \begin{cases} 1 & \text{if } x = x' \\ 0 & \text{otherwise} \end{cases}$	NA

Table 2.1: Examples of Kernel Functions on $\mathbb{R}^n$. θ denotes the set of hyper-parameters.

major advantage to the methods that require traditional feature representations, as there are cases where the feature space associated with a kernel is very high dimensional, or even infinite-dimensional. Representations in such feature spaces would be difficult or even impossible to compute explicitly, whereas computing the kernel matrix remains possible. Note that, the feature map ϕ associated with a kernel k is usually not unique. For example, the degree 2 (homogeneous) polynomial kernel, $k(x, x') = \langle x, x' \rangle^2$, can have $\phi(x) = [x(1)^2, x(2)^2, \sqrt{2}x(1)x(2)]$, $\phi(x) = [x(1)^2, x(2)^2, x(1)x(2), x(2)x(1)]$ and many other variants as the feature space. However, there is a special feature space which is unique for a given kernel. We state the result of this special feature space construction *assuming only* the positive definiteness of the kernel function in the form of a theorem (see for example (Schölkopf & Smola, 2002; Shawe-Taylor & Cristianini, 2004)).

Theorem 2 (Kernels as Feature Representations) *For any kernel k on a space $\mathcal{X}$, there exist a Hilbert space[1] $\mathcal{F}$ and a mapping $\phi : \mathcal{X} \to \mathcal{F}$ such that*

$$k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{F}}, \text{ for any } x, x' \in \mathcal{X}, \tag{2.2}$$

where $\langle \cdot, \cdot \rangle_{\mathcal{F}}$ is the inner product in the space $\mathcal{F}$.

Proof Given a kernel function k, define a map from $\mathcal{X}$ into the space of functions mapping $\mathcal{X}$ into $\mathbb{R}$ (denoted as $\mathcal{F}' := \{f : \mathcal{X} \to \mathbb{R}\}$) as $\phi(x) = k(x, \cdot)$[2]. That is, $\phi(x) : \mathcal{X} \to \mathbb{R}$ is a function which has a value of $k(x, x')$ at $x' \in \mathcal{X}$. We can then

[1] A Hilbert space is a linear space endowed with an inner product that is complete in the norm corresponding to that inner product.

[2] We use a $\cdot$ to indicate the argument of the function.

construct the following vector space:

$$\mathcal{F}' := \text{span}\{\phi(x) : x \in \mathcal{X}\} = \left\{ \sum_{i=1}^{n} \alpha_i k(x_i, \cdot) : i \in \mathbb{N}, \alpha_i \in \mathbb{R}, x_i \in \mathcal{X} \right\}. \tag{2.3}$$

For $f, g \in \mathcal{F}'$ be given by $f_\alpha(\cdot) = \sum_{i=1}^{n} \alpha_i k(x_i, \cdot)$ and $g_\beta(\cdot) = \sum_{j=1}^{n'} \beta_j k(x_j, \cdot)$, we can define

$$\langle f_\alpha, g_\beta \rangle = \sum_{i=1}^{n} \sum_{j=1}^{n'} \alpha_i \beta_j k(x_i, x_j). \tag{2.4}$$

We need to show that $\langle f_\alpha, g_\beta \rangle$ is in fact an inner product by checking the following 3 criteria:

- Symmetry:

$$\langle f_\alpha, g_\beta \rangle = \sum_{i=1}^{n} \sum_{j=1}^{n'} \alpha_i \beta_j k(x_i, x_j) = \sum_{j=1}^{n'} \sum_{i=1}^{n} \beta_j \alpha_i k(x_j, x_i) = \langle g_\beta, f_\alpha \rangle ;$$

- Bilinearity:

$$\langle f_\alpha, g_\beta \rangle = \sum_{i=1}^{n} \sum_{j=1}^{n'} \alpha_i \beta_j k(x_i, x_j) = \sum_{i=1}^{n} \alpha_i g_\beta(x_i) = \sum_{j=1}^{n'} \beta_j f_\alpha(x_j).$$

  We still need to show that (2.4) is well-defined, i.e. it is independent of the representation of f and g. We do so by re-defining the functions as $f_\alpha(\cdot) = \sum_{i=1}^{\hat{n}} \hat{\alpha}_i k(\hat{x}_i, \cdot) = f_{\hat{\alpha}}(\cdot)$ and $g_\beta(\cdot) = \sum_{j=1}^{\hat{n}'} \hat{\beta}_j k(\hat{x}_j, \cdot) = g_{\hat{\beta}}(\cdot)$ and evaluating:

$$\begin{aligned} \left\langle f_{\hat{\alpha}}, g_{\hat{\beta}} \right\rangle &= \sum_{i=1}^{\hat{n}} \sum_{j=1}^{\hat{n}'} \hat{\alpha}_i \hat{\beta}_j k(\hat{x}_i, \hat{x}_j) \\ &= \sum_{i=1}^{\hat{n}} \hat{\alpha}_i g_\beta(\hat{x}_i) = \sum_{i=1}^{\hat{n}} \sum_{j=1}^{n'} \hat{\alpha}_i \beta_j k(\hat{x}_i, x_j) \\ &= \sum_{j=1}^{n'} \beta_j f_\alpha(x_j) = \sum_{i=1}^{n} \sum_{j=1}^{n'} \alpha_i \beta_j k(x_i, x_j) \\ &= \langle f_\alpha, g_\beta \rangle ; \end{aligned} \tag{2.5}$$

- Positive definiteness: $\langle f, f \rangle = \sum_{i,j=1}^{n} \alpha_i \alpha_j k(x_i, x_j) \geq 0$ follows directly from the positive definiteness of k.

We have just defined an inner product space $\mathcal{F}'$. By including all the limit functions in $\mathcal{F}'$, we can obtain the Hilbert space $\mathcal{F}$. ■

Remark 3 (Reproducing Property) *We have the following property:*

$$\langle f, k(x,\cdot)\rangle = \sum_{i=1}^{n} \alpha_i k(x_i, x) = f(x) \quad \text{for any } f \in \mathcal{F}'. \tag{2.6}$$

Further, this property implies

$$k(x,x') = \langle k(x,\cdot), k(x',\cdot)\rangle\,. \tag{2.7}$$

Proof Plugging in $k(x,\cdot)$ as $g(\cdot)$ to (2.4) shows the first property. While, replacing f in (2.6) with $k(x',\cdot)$ shows the second property. ■

Due to this property, the Hilbert space defined in the theorem above is also called the reproducing kernel Hilbert Space (RKHS), associated with the reproducing kernel k. An alternate way to define a RKHS is by defining continuous evaluational operators and subsequently invoking the Riesz representer theorem to construct the reproducing kernel (Schaback, 2007). The Moore-Aronszajn theorem (Aronszajn, 1950) guarantees the existence of a unique RKHS for every positive definite kernel and vice versa.

2.1.4 Universal Kernels

We describe properties of RKHS that will be useful for the later development of kernel-based embedding of distributions in Section 2.3. We focus on compact $\mathcal{X} \subset \mathbb{R}^d$. For construction of universal kernels on compact metric space $\mathcal{X} \not\subset \mathbb{R}^d$ such as for trees, graphs and histograms, refer to (Christmann & Steinwart, 2010).

Definition 4 (Continuous Kernel (Steinwart, 2001)) *Let k be a kernel on a compact space $\mathcal{X}$, and $\phi : \mathcal{X} \to \mathcal{F}$ be the feature map to the RKHS of k, that is, $\phi(x) = k(x,\cdot)$. Then k is called a continuous kernel if and only if ϕ is continuous.*

Definition 5 (Universal Kernel (Steinwart, 2001)) *Let $C(\mathcal{X})$ be a space of continuous functions on a compact domain $\mathcal{X}$. A continuous kernel k on $\mathcal{X}$ is called universal if the RKHS $\mathcal{F}$ induced by k is dense in $C(\mathcal{X})$ in L_∞ sense, that is, for every function $f \in C(\mathcal{X})$ and every $\epsilon > 0$, there exists a function $g \in \mathcal{F}$ such that $\|f - g\|_\infty < \epsilon$.*

From the list of kernels in Table 2.1, Gaussian and Laplace are examples of a universal kernel. Linear and polynomial kernels while continuous are not universal. Delta kernel is not continuous, thus not universal. Several criteria of universality are available (Steinwart, 2001), and can be used to check whether a kernel is universal.

Two most important characteristics of universal kernels are (Steinwart, 2001):

1. Every universal kernel separates[1] all compact subsets;

2. Every feature map of a universal kernel is injective.

The first characteristic implies that for any finite subset $X = \{x_1, \ldots, x_m\} \subseteq \mathcal{X}$, the classes corresponding to every possible label assignment can always be correctly separated by a function in the RKHS. While the second characteristic means a unique observation will be represented as a unique element in the RKHS. This second characteristic plays a crucial role in the kernel embeddings.

2.1.5 Regularised Risk Functionals

Recall that, in the supervised machine learning setting, we are given m input-output pairs $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^m$ (training set), where $x_i \in \mathcal{X}$ (the set of inputs) and $y_i \in \mathcal{Y}$ (the set of outputs). The output set $\mathcal{Y}$ is a discrete set of possible labels for classification problems. These data pairs are drawn independently and identically distributed (i.i.d.) from an unknown probability distribution P on $\mathcal{X} \times \mathcal{Y}$ with a density $p(x, y)$. Earlier we have seen how to implicitly define representation of the data via kernel similarity matrix; we can now draw our attention to the learning task of inferring a function, $f \in \mathcal{F}$ for some function spaces $\mathcal{F}$, that maps inputs to outputs, $f : \mathcal{X} \to \mathcal{Y}$. This function f will then be used to *predict* an output $y \in \mathcal{Y}$ for a given unseen input $x \in \mathcal{X}$. To assess the quality of this inferred function, we need a notion of *loss function*, $l : \mathcal{X} \times \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$. The loss function measures the loss or penalty incurred for predicting the example x_i to have a value of $f(x_i)$ when the true output value is y_i. For example, in binary classification, the 0/1-loss $l(x, y, f(x)) = \frac{1}{2}(1 - \text{sgn}(yf(x)))$ where the signum function is defined as $\text{sgn}(x) = \begin{cases} -1 & \text{if } x \leq 0 \\ 1 & \text{otherwise} \end{cases}$ can be used. It assigns a unit

[1] We say k separates A and B means that there exists a pair $(w, b) \in \mathcal{F} \times \mathbb{R}$ such that $\langle \phi(x), w \rangle + b \geq 0$ for all $x \in A$ and $\langle \phi(y), w \rangle + b < 0$ for all $y \in B$.

penalty for a misprediction and no penalty for a correct prediction. Finding the best function f can then be casted as a risk minimisation problem:

$$\min_{f \in \mathcal{F}} R(f), \tag{2.8}$$

where the risk $R(f)$ is defined as

$$R(f) := \int_{\mathcal{X} \times \mathcal{Y}} l(x, y, f(x)) p(x, y) dx dy = \mathbf{E}_{p(x,y)}[l(x, y, f(x))]. \tag{2.9}$$

We are faced by a problem that the risk in (2.9) can not be minimised directly since the underlying probability distribution that generates the observed data is unknown. A step forward is to replace the density $p(x, y)$ by its empirical counterpart based on the available training data. We are now trying to infer a function by minimising the empirical risk

$$R_{\text{emp}}(f) := \frac{1}{m} \sum_{i=1}^{m} l(x_i, y_i, f(x_i)). \tag{2.10}$$

Though (2.10) might appear to be the solution, for too rich function space $\mathcal{F}$, the deviation between the empirical risk and the expected risk might be substantial; this is so called an *overfitting* problem (Tikhonov, 1943, 1963; Vapnik, 1995). A way to restrict the richness of the function space is to introduce a regularisation term $\Omega(f)$ to the empirical risk objective function. We now find a function from the following class of regularised risk functionals (Tikhonov, 1943, 1963; Vapnik, 1995):

$$R_{\text{reg}}(f) = R_{\text{emp}}(f) + \lambda \Omega(f). \tag{2.11}$$

The regularisation term $\Omega(f)$ controls the richness of the function space such that the chosen function is able to generalise well to unseen data points. The regularisation parameter $\lambda > 0$ balances the relative influence of loss and regularisation terms.

Kernel-based methods select a function from a possibly infinite dimensional RKHS; this appears to be a hard optimisation problem. However, by virtue of the following theorem (Kimeldorf & Wahba, 1971; Schölkopf & Smola, 2002), the search in possibly an infinite-dimensionals space can be conveniently transformed to Euclidean space.

Theorem 6 (Representer Theorem) *Let $\mathcal{X}$ be a set of input data points endowed with a kernel k and $\mathcal{F}$ its corresponding RKHS and $\mathcal{Y}$ be a set of output data points. Let $\{(x_1, y_1), \ldots, (x_m, y_m)\} \subset \mathcal{X} \times \mathcal{Y}$ be a finite set of input-output pairs, $\Omega : [0, \infty) \to \mathbb{R}$ be a strictly monotonic increasing function, and*

$\tilde{l} : (\mathcal{X} \times \mathcal{Y} \times \mathcal{Y})^m \to \mathbb{R} \cup \{\infty\}$ be an arbitrary loss function. Then each minimiser $f \in \mathcal{F}$ of the regularised risk functional

$$\tilde{l}((x_1, y_1, f(x_1)), \ldots, (x_m, y_m, f(x_m))) + \Omega(\|f\|_{\mathcal{F}}) \tag{2.12}$$

admits a representation of the form

$$f(x) = \sum_{i=1}^{m} \alpha_i k(x_i, x). \tag{2.13}$$

Proof From Theorem 2, we know that the kernel k satisfies

$$k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{F}}, \text{ for any } x, x' \in \mathcal{X}, \tag{2.14}$$

where $\langle \cdot, \cdot \rangle_{\mathcal{F}}$ is the inner product in the RKHS $\mathcal{F}$. We decompose $f \in \mathcal{F}$ into a part $f^{||} \in \mathcal{F}$ which lives inside the span of $\phi(x_1), \ldots, \phi(x_m)$ and its orthogonal complement $f^{\perp} \in \mathcal{F}$. We have

$$f = \sum_{i=1}^{m} \alpha_i \phi(x_i) + f^{\perp} \tag{2.15}$$

with $\alpha_i \in \mathbb{R}$ and $f^{\perp}$ satisfying, for all $i \in \{1, \ldots, m\}$, $\left\langle f^{\perp}, \phi(x_i) \right\rangle = 0$. By the reproducing property of $\mathcal{F}$, for an arbitrary input data point x_j, we have

$$f(x_j) = \left\langle \sum_{i=1}^{m} \alpha_i \phi(x_i) + f^{\perp}, \phi(x_j) \right\rangle \tag{2.16}$$

$$= \sum_{i=1}^{m} \alpha_i \left\langle \phi(x_i), \phi(x_j) \right\rangle. \tag{2.17}$$

Hence, we see that the term $f^{\perp}$ is irrelevant for the loss term in (2.12). We now check the effect of $f^{\perp}$ on the regularisation term. We have

$$\Omega(\|f\|) = \Omega\left(\left\| \sum_{i=1}^{m} \alpha_i \phi(x_i) + f^{\perp} \right\| \right) \tag{2.18}$$

$$= \Omega\left(\sqrt{\left\| \sum_{i=1}^{m} \alpha_i \phi(x_i) \right\|^2 + \|f^{\perp}\|^2} \right) \tag{2.19}$$

$$\geq \Omega\left(\left\| \sum_{i=1}^{m} \alpha_i \phi(x_i) \right\| \right), \tag{2.20}$$

where the middle equality is due to the orthogonality of $f^{\perp}$ to $\sum_{i=1}^{m} \alpha_i \phi(x_i)$ and the last inequality is due to the strict monotonicity of Ω. We see that taking $f^{\perp}$

to be 0 strictly reduces the regularisation term while has no effect on the loss term, thus any minimiser of (2.12) must have $f^{\perp} = 0$. ■

The above representer theorem shows that the solution of an optimisation problem in possibly an infinite-dimensional RKHS $\mathcal{F}$ lies in the span of m particular kernels centred on the given data points. On the practical side, this is computationally attractive as it reduces a potentially infinite dimensional optimisation problem to m-dimensional problem where m is the number of given data points. Monotonicity of the regulariser Ω is necessary for the representer theorem to hold, however, it does not prevent the problem in (2.12) to have many solutions. For ensuring a unique minimiser, we would need the convexity of the loss term and the reguralization term (for information on convexity, refer to Section 2.4). If the regulariser does not have a strict monotonicity property, then potentially there will be minimisers of (2.12) which do not admit the representation. However, it still follows that there exists at least another minimiser that does admit the representation form.

2.2 Kernel-Based Learning Methods

Once we have defined a suitable similarity measure (kernel) for the data we want to analyse, several techniques are readily available. Particularly successful are technique that were originally developed for linear analysis of vector-valued data, and that are made applicable in a general, non-linear setting by the kernel trick.

2.2.1 Kernel Classification

The most widely used kernel-based classification method is *Support Vector Classification* (SVC) (Boser et al., 1992; Cortes & Vapnik, 1995). Given input-output pairs $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^m$ where $x_i \in \mathcal{X}$ and $y_i \in \{1, -1\}$, SVC requires solving the optimisation problem

$$\max_{\alpha_1,\ldots,\alpha_m \in \mathbb{R}} -\frac{1}{2}\sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j k(x_i, x_j) + \sum_{i=1}^{m} \alpha_i. \tag{2.21}$$

The above optimisation problem is the so-called dual problem of (2.12) with a loss function $\tilde{l}((x_1, y_1, f(x_1)), \ldots, (x_m, y_m, f(x_m)))$ expressed as the average of a hinge loss term, $l(x, y, f(x)) = \max(0, 1 - yf(x))$, and a quadratic regulariser,

$\Omega(\|f\|_{\mathcal{F}}) = \|f\|^2_{\mathcal{F}}$. From the solution we obtain a prediction $f(x) = \sum_i \alpha_i k(x_i, x)$ from which we obtain the label assignment as $F(x) = \text{sgn}(f(x))$.

2.2.2 Kernel Regression

Another common task in data analysis is the prediction of a function with real-valued output, that is a regression task. A straight-forward kernel-based regression method is *Kernel Ridge Regression* (KRR) (Saunders et al., 1998) that generalises linear least-squares regression. Given a data set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^m$ where $x_i \in \mathcal{X}$ and $y_i \in \mathbb{R}$, KRR predicts the function

$$f(x) = \sum_{i=1}^{m} \alpha_i k(x_i, x) = \sum_{i=1}^{m} \underbrace{\sum_{j=1}^{m} y_j [(\lambda I_{[m\times m]} + K)^{-1}]_{ij}}_{\alpha_i} k(x_i, x), \tag{2.22}$$

where $I_{[m\times m]}$ is the $m \times m$ identity matrix, K is the $m \times m$ kernel matrix, and λ is a small ridge parameter that avoids overfitting. The form of the predictive function is a result of solving (2.12) with a loss function that is decomposable into a sum of squared loss terms, $l(x, y, f(x)) = (y - f(x))^2$, and a quadratic regulariser. Another successful kernel method for regression is *Support Vector Regression* (SVR) (Vapnik, 1995). SVR has been shown to be particularly robust against outliers in the data thanks to the ingenious ϵ-insensitive loss function, $l(x, y, f(x)) = \max(0, |y - f(x)| - \epsilon)$.

2.2.3 Kernel-Based Dimensionality Reduction

Kernel-based learning also allows the computation of a low-dimensional vector representation for potentially high-dimensional non vectorial data. The best known kernel method for this tasks is *Kernel Principal Component Analysis* (Schölkopf et al., 1996), which generalises Principal Component Analysis (Jolliffe, 1986). Kernel Principal Component Analysis is of particular interest when performing interactive data analysis, as it provides a low-dimensional representation suitable, e.g., for visualisation, even in cases where it is difficult to manually define a finite-dimensional vector representation, such as for strings and graphs. For a given set of objects $\mathcal{D} = \{x_i, \ldots, x_m\}$ where $x_i \in \mathbb{R}^d$, we first compute the kernel matrix $K \in \mathbb{R}^{m\times m}$. From this, we compute the so-called *centred* kernel matrix $\tilde{K} = K - \frac{1}{m}\mathbf{1}_{[m\times m]}K - \frac{1}{m}K\mathbf{1}_{[m\times m]} + \frac{1}{m^2}\mathbf{1}_{[m\times m]}K\mathbf{1}_{[m\times m]}$ where $\mathbf{1}_{[m\times m]}$ is the $m \times m$-dimensional matrix in which all entries are 1. Let $\lambda_1, \ldots, \lambda_m$ and

$v_1, \ldots, v_m$ denote the eigenvalues and eigenvectors of $\tilde{K}$, sorted in decreasing order of the eigenvalues. An n-dimensional representation $\hat{x}_1, \ldots, \hat{x}_m \in \mathbb{R}^n$ of $\mathcal{D}$ is then given by

$$[\hat{x}_i]_j = \lambda_j [v_j]_i \tag{2.23}$$

for $j = 1, \ldots, n$ where $[\cdot]_j$ denotes the j-th component of a vector.

2.2.4 Other Kernel-Based Learning Methods

Besides for the problems of classification, regression and dimensionality reduction, there are also kernel-based learning methods for multiple other data analysis problems such as novelty detection, clustering, correlation analysis, to name a few. References to those methods can be found in many textbooks and article on kernel-based learning, such as (Schölkopf & Smola, 2002; Shawe-Taylor & Cristianini, 2004; Schölkopf et al., 2004; Hofmann et al., 2008).

2.3 Kernel Embeddings of Distribution

The previous section describes applications of the kernel trick in deriving a non-linear version of many linear methods that was shown to excel in a wide range of learning tasks. In contrast, this section is devoted to the recent interest in considering basic linear statistics in RKHS instead in the usual Euclidean space (Smola et al., 2007a). For example, the usual notion of mean in Euclidean corresponds to the notion of mean element in RKHS. As we will see later, this RKHS element characterises a probability distribution and is useful, for instance, for comparing distributions. In comparison to the long standing measure of distance between distributions such as Kullback-Leibler divergence, the distance measure via mean element is directly estimated from samples; it avoids the need of estimating the distributions as an intermediate step. Below we give the existence and definition of the mean element:

Theorem 7 *Let P_X be a probability distribution on $\mathcal{X}$ and k be a kernel on $\mathcal{X}$ with its corresponding RKHS $\mathcal{F}$. If $\mathbf{E}_{(x)\sim P_X}[\sqrt{k(x,x)}] < \infty$, there exists an element $\mathbf{E}_{(x)\sim P_X}[k(x,\cdot)] \in \mathcal{F}$ such that for all $f \in \mathcal{F}$ we have $\mathbf{E}_{x\sim P_X}[f(x)] = \mathbf{E}_{(x)\sim P_X}[\langle f, k(x,\cdot)\rangle] = \langle f, \mu[P_X]\rangle$.*

Proof We note that the mapping $f \mapsto \mathbf{E}_{(x)\sim P_X}[f(x)]$ is a bounded linear functional on $\mathcal{F}$ if the condition $\mathbf{E}_{(x)\sim P_X}[\sqrt{k(x,x)}] < \infty$ is satisfied due to the follow-

ing chain of (in)-equalities:

$$\begin{aligned} |\mathbf{E}_{(x)\sim P_X}[f(x)]| &\leq \mathbf{E}_{(x)\sim P_X}[|\,\langle f, k(x,\cdot)\rangle\,|] \\ &\leq \|f\|_{\mathcal{F}}\,\mathbf{E}_{(x)\sim P_X}[\|k(x,\cdot)\|_{\mathcal{F}}] \\ &= \|f\|_{\mathcal{F}}\,\mathbf{E}_{(x)\sim P_X}[\sqrt{k(x,x)}]. \end{aligned} \tag{2.24}$$

The condition is easily satisfied for a bounded kernel k. The existence of the element $\mathbf{E}_{(x)\sim P_X}[k(x,\cdot)] \in \mathcal{F}$ is then due to the Riesz representation theorem. ■

Definition 8 (Mean Element) *We call the element* $\mu[P_X] := \mathbf{E}_{(x)\sim P_X}[k(x,\cdot)]$ *as a mean element.*

Expectations of function f with respect to P_X are computed simply by taking its inner products with the mean element $\mu[P_X]$ in the RKHS.

Typically, we will have access to independent and identically distributed (i.i.d.) samples $X = \{(x_i)\}_{i=1}^m$ from P_X, thus we can only compute an empirical estimate of the mean element, i.e.

$$\mu[X] := \frac{1}{m}\sum_{i=1}^{m} k(x_i, .), \tag{2.25}$$

which is obviously in $\mathcal{F}$. The empirical means of f with respect to the data samples are $\frac{1}{m}\sum_{i=1}^m f(x_i) = \frac{1}{m}\sum_{i=1}^m \langle f, k(x_i,.)\rangle = \langle f, \mu[X]\rangle$.

The attractiveness of embedding distribution to Hilbert space lies in the following two reasons, which we will state in the form of theorems:

Theorem 9 (Injectivity of Mean Mapping (Smola et al., 2007a)) *Let k be a universal kernel, then the mapping $P_X \mapsto \mu[P_X]$ is injective.*

Theorem 10 (Concentration of Measure (Smola et al., 2007a)) *The deviation between the empirical quantity of the mean element $\mu[X]$ estimated from m samples and its expectation value $\mu[P_X]$ is in the order of $O(m^{-\frac{1}{2}})$.*

Theorem 9 says that the mean element $\mu[P_X]$ is a characterisation of distribution P_X, that is, different distributions have different mean elements in $\mathcal{F}$. Theorem 10 ensures that $\mu[X]$, the empirical quantity that can be easily computed from samples, can be seen as a good proxy for $\mu[P_X]$. Consequently, the first theorem allows us to use mean elements to define a distance between distributions and the second theorem ensures that the defined distance measure can be computed from observed samples.

2.3.1 Distance between Distributions

The mean element $\mu[P_X]$ can be used as a distance measure between distributions P_X and P_Y (Smola et al., 2007a):

$$\begin{aligned} D(P_X, P_Y) &:= \|\mu[P_X] - \mu[P_Y]\|^2 \\ &= \|\mu[P_X]\|^2 + \|\mu[P_Y]\|^2 - 2\langle \mu[P_X], \mu[P_Y] \rangle \\ &= \mathop{\mathbf{E}}_{(x)\sim P_X (x')\sim P_X} [k(x, x')] + \mathop{\mathbf{E}}_{(y)\sim P_Y (y')\sim P_Y} [k(y, y)] - 2 \mathop{\mathbf{E}}_{(x)\sim P_X (y)\sim P_Y} [k(x, y)], \end{aligned} \tag{2.26}$$

where x' is a random variable independent of x drawn from distribution P_X, likewise, y' and y are independent random variables drawn according to P_Y.

Empirical Estimate Suppose we have at our disposition m i.i.d. samples $\{x_1, \ldots, x_m\}$ of P_X and n i.i.d. samples $\{y_1, \ldots, y_n\}$ of P_Y, then the sample estimate of $D(P_X, P_Y)$ can be easily computed by substituting the empirical estimates of the mean elements $\mu[X] := \frac{1}{m}\sum_{i=1}^m k(x_i, .)$ and $\mu[Y] := \frac{1}{n}\sum_{i=1}^n k(y_i, .)$:

$$D(X, Y) = \frac{1}{m^2}\sum_{i,j=1}^{m} k(x_i, x_j) + \frac{1}{n^2}\sum_{i,j=1}^{n} k(y_i, y_j) - \frac{2}{mn}\sum_{i,j=1}^{m,n} k(x_i, y_j). \tag{2.27}$$

The above sample estimate is a consistent but biased estimator. An unbiased empirical estimator of $D(P_X, P_Y)$ is a U-statistic (Serfling, 1980), and for the explicit form of the estimator, refer to Smola et al. (2007a). However, this unbiased estimator imposes a restriction that the same number of samples are drawn from both P_X and P_Y, that is $(m = n)$.

2.3.2 Hilbert-Schmidt Independence Criterion (HSIC)

The above embedding of distribution approach can also be used to measure the dependence between two random variables x and y on domains $\mathcal{X}$ and $\mathcal{Y}$ respectively (Smola et al., 2007a). Let k be a kernel on $\mathcal{X}$ with RKHS $\mathcal{F}$ and $\mathcal{G}$ be the RKHS on $\mathcal{Y}$ with kernel l. Denote a joint space, $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ and h be a kernel on this product space $\mathcal{Z}$ with its corresponding RKHS $\mathcal{H}$. Let the joint distribution on $\mathcal{Z}$ be P_{XY} and the marginal distributions be P_X and P_Y. We note that X and Y are independent if and only if the joint distribution factorizes as a product of its marginals, $P_{XY} = P_X P_Y$. A distance between P_{XY} and $P_X P_Y$ in term of the mean elements $\mu[P_{XY}] := \mathbf{E}_{(x,y)\sim P_{XY}}[h((x, y), \cdot)]$ and $\mu[P_X \times P_Y] := \mathbf{E}_{(x)\sim P_X}\mathbf{E}_{(y)\sim P_Y}[h((x, y), \cdot)]$ can then be used as an independence

measure. Assuming that the RKHS $\mathcal{H}$ is a direct product $\mathcal{F} \otimes \mathcal{G}$ of the RKHSs on $\mathcal{X}$ and $\mathcal{Y}$ which leads to a factorized kernel $h((x,y),(x',y')) = k(x,x')l(y,y')$, the distance-based measure of dependence is now

$$\begin{aligned} D(P_{XY}, P_X P_Y) &= \|\mu[P_{XY}] - \mu[P_X \times P_Y]\|^2 \\ &= \left\|\mathbf{E}_{(x,y)\sim P_{XY}}[k(x,\cdot)l(y,\cdot)] - \mathbf{E}_{(x)\sim P_X}[k(x,\cdot)]\mathbf{E}_{(y)\sim P_Y}[l(y,\cdot)]\right\|^2 \\ &= \mathop{\mathbf{E}}_{(x,y)\sim P_{XY}(x',y')\sim P_{XY}}[k(x,x')l(y,y')] \\ &+ \mathop{\mathbf{E}}_{x\sim P_X x'\sim P_X}[k(x,x')] \mathop{\mathbf{E}}_{(y)\sim P_Y (y')\sim P_Y}[l(y,y')] \\ &- 2 \mathop{\mathbf{E}}_{(x,y)\sim P_{XY}}[\mathop{\mathbf{E}}_{x'\sim P_X}[k(x,x')] \mathop{\mathbf{E}}_{(y')\sim P_Y}[l(y,y')]]. \end{aligned} \tag{2.28}$$

The above measure is what Gretton et al. (2005) show to be the Hilbert-Schmidt norm of the covariance operator between RKHSs, thus the name Hilbert-Schmidt Independence Criterion (HSIC). For universal kernels, this measure is zero if and only if x and y are independent.

Empirical Estimate Given a sample $Z = \{(x_1,y_1),\ldots,(x_m,y_m)\}$ of size m drawn from P_{XY} an empirical estimate of HSIC can be derived by estimating the terms in (2.28) using the kernel matrices $K \in \mathbb{R}^{m\times m}$ and $L \in \mathbb{R}^{m\times m}$ for the set $X = \{x_1,\ldots,x_m\}$ and the set $Y = \{y_1,\ldots,y_m\}$ respectively, i.e. $K_{ij} = k(x_i,x_j)$ and $L_{ij} = l(y_i,y_j)$. The sample estimate for each term is as follow:

$$\begin{aligned} \mathop{\mathbf{E}}_{(x,y)\sim P_{XY}(x',y')\sim P_{XY}}[k(x,x')l(y,y')] &\underset{\text{sample}}{\rightarrow} \frac{1}{m^2}\operatorname{tr}(KL) \\ \mathop{\mathbf{E}}_{(x,y)\sim P_{XY}}[\mathop{\mathbf{E}}_{x'\sim P_X}[k(x,x')] \mathop{\mathbf{E}}_{(y')\sim P_Y}[l(y,y')]] &\underset{\text{sample}}{\rightarrow} \frac{1}{m^3}\mathbf{1}_{[m\times m]}^\top KL\mathbf{1}_{[m\times m]} \\ \mathop{\mathbf{E}}_{x\sim P_X x'\sim P_X}[k(x,x')] \mathop{\mathbf{E}}_{(y)\sim P_Y (y')\sim P_Y}[l(y,y')] &\underset{\text{sample}}{\rightarrow} \frac{1}{m^4}\mathbf{1}_{[m\times m]}^\top K\mathbf{1}_{[m\times m]}\mathbf{1}_{[m\times m]}^\top L\mathbf{1}_{[m\times m]}. \end{aligned}$$

Combining the above sample estimate terms, the empirical estimate of HSIC is given by

$$D(Z) = m^{-2}\operatorname{tr} HKHL = m^{-2}\operatorname{tr}\bar{K}\bar{L}. \tag{2.29}$$

The term $H_{ij} = \delta_{ij} - m^{-1}$ centres the observations of set X and set Y in feature space. Finally, $\bar{K} := HKH$ and $\bar{L} := HLH$ denote the centred versions of K and L respectively. Note that (2.29) is a consistent but biased estimate where the expectations with respect to x, x', y, y' have all been replaced by empirical averages over the set of observations (for further properties of this empirical estimator and the unbiased sample estimate refer to (Smola et al., 2007a) and references therein).

2.4 Convex Optimisation

Optimisation lies at almost every heart of machine learning problems (Bennett et al., 2006). Consider the regularised risk functionals in Section 2.1.5. After an appropriate choice of a loss function and a regulariser, finding the best function f in (2.11) constitutes solving an optimisation problem with $R_{\text{reg}}(f)$ as the objective function. In general, minimising an arbitrary objective function is difficult, however for a certain class of functions, called convex functions, the optimisation efforts become considerably easier. Minimisation of a convex function over a convex feasible set admits a property that every local optimum is in fact a global optimum. It is then not surprising that many machine learning algorithms are (re)-formulated in terms of convex optimisation problems. For an introduction to the field of convex optimisation refer to (Boyd & Vandenberghe, 2004); we briefly touch upon definitions of crucial ingredients of convex optimisation problem, these are convex functions and convex sets.

Definition 11 (Convex Set) *A set $C \subset \mathbb{R}^d$ is said to be convex if for any x_1, $x_2 \in C$ and any λ with $0 \leq \lambda \leq 1$, we have*

$$\lambda x_1 + (1 - \lambda)x_2 \in C. \tag{2.30}$$

Geometrically, the above means any line segment between any two points x_1 and x_2 from the set C must lie in C.

Definition 12 (Convex Function) *A function f defined on a set $\mathcal{X} \subset \mathbb{R}^d$ is convex if for any x_1, $x_2 \in \mathcal{X}$ and any $0 \leq \lambda \leq 1$ such that $\lambda x_1 + (1 - \lambda)x_2 \in \mathcal{X}$, we have*

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2). \tag{2.31}$$

Geometrically, the above inequality means a line segment between $(x_1, f(x_1))$ and $(x_2, f(x_2))$ must lie above the graph of the function f.

For a differentiable function $f : \mathcal{X} \to \mathbb{R}$, we can test whether the function is convex by checking the following condition:

$$f(x_2) \geq f(x_1) + \langle x_2 - x_1, \nabla f(x)|_{x_1} \rangle \quad \forall\, x_1, x_2 \in \mathcal{X}. \tag{2.32}$$

The condition means that its first order Taylor approximation is always a lower bound for a convex function. Further, the function f is called strictly convex if the inequalities in (2.31) or (2.32) are strict whenever $x_1 \neq x_2$. For a twice

differentiable function, the convexity condition is even simpler; the Hessian of the function must be positive semi-definite, that is

$$\nabla^2 f(x) \succeq 0. \tag{2.33}$$

For a positive definite Hessian, $\nabla^2 f(x) \succ 0$, the function is strictly convex.

The following lemma describes a relation between convex functions and convex sets:

Lemma 13 (Relationship between Convex Sets and Convex Functions) *For a convex function $f : \mathcal{X} \to \mathbb{R}$ and for every $c \in \mathbb{R}$, the following set, called c-sublevel set,*

$$\mathcal{X}_c := \{x \in \mathcal{X} | f(x) \leq c\} \tag{2.34}$$

is convex.

Proof For $x_1, x_2 \in \mathcal{X}_c$, we have $f(x_1), f(x_2) \leq c$. By convexity of the function f, we also have

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2) \leq c \quad \forall\, 0 \leq \lambda \leq 1, \tag{2.35}$$

and thus $\lambda x_1 + (1-\lambda)x_2 \in \mathcal{X}_c$. ■

Note that, the converse is not true, that is, a function can have all its sublevel sets convex, but not be a convex function. We are now ready to state a theorem that shows a minimisation of a strictly convex function over a convex feasible set admits exactly one global optimum. We use the term *convex minimisation* to refer to a minimisation problem with a convex feasible set and a convex objective function.

Theorem 14 (Global Optimum of Convex Minimisation) *Let $f : \mathcal{X} \to \mathbb{R}$ be a convex function on a convex set $\mathcal{X} \subset \mathbb{R}^d$. If the function f attains its minimum, then the set where the minimum value is attained is convex. Further, for a strictly convex function this set is a singleton.*

Proof Suppose c is the minimal value of f on $\mathcal{X}$. Then the c-sublevel set $\mathcal{X}_c := \{x \in \mathcal{X} | f(x) \leq c\}$ is convex by Lemma 13. If f is strictly convex, then for $x_1, x_2 \in \mathcal{X}_c$ with $x_1 \neq x_2$ and any $0 \leq \lambda \leq 1$, we have

$$f(\lambda x_1 + (1-\lambda)x_2) < \lambda f(x_1) + (1-\lambda) f(x_2) = \lambda c + (1-\lambda)c = c. \tag{2.36}$$

Where the inequality follows from the definition of strictly convex. This is a contradiction since we start off with the assumption that f attains its minimum on $\mathcal{X}_c$. Hence $\mathcal{X}_c$ can not contain x_1, x_2 with $x_1 \neq x_2$ and must therefore be a singleton. ■

Lemma 15 (Optimality Criterion for a Differentiable Convex Function) *Let $f : \mathcal{X} \to \mathbb{R}$ be a differentiable convex function. Then x is a minimiser of f if and only if*

$$\langle y - x, \nabla f(x) \rangle \geq 0 \quad \forall\, y \in \mathcal{X}. \tag{2.37}$$

Proof We want to show the forward implication; suppose x is the optimum but (2.37) does not hold, that is, for some $y \in \mathcal{X}$, we have

$$\langle y - x, \nabla f(x) \rangle < 0. \tag{2.38}$$

Take a line segment $z(\lambda) = (1 - \lambda)x + \lambda y$ with $0 \leq \lambda \leq 1$. Since the set $\mathcal{X}$ is convex, $z(\lambda)$ lies in $\mathcal{X}$. However, by

$$\frac{d}{d\lambda} f(z(\lambda))|_{\lambda=0} = \langle y - x, \nabla f(x) \rangle < 0, \tag{2.39}$$

so there exist a small positive $\lambda \in [0, 1]$ such that $f(z(\lambda)) < f(x)$. This is a contradiction to x being optimal. To show the reverse implication, we note that $f(y) \geq f(x)$ by (2.32) whenever (2.37) holds. ■

For an (unconstrained) convex minimisation problem, the condition for x to be optimal in (2.37) reduces to the gradient vanishing sufficient and necessary condition, $\nabla f(x) = 0$. To show this, consider the set $\mathcal{X}$ to be an open set. Take $y = x - \epsilon \nabla f(x)$ for $\epsilon \in \mathbb{R}$. For a positive and small ϵ, the condition in (2.37) becomes $\langle y - x, \nabla f(x) \rangle = \langle -\epsilon \nabla f(x), \nabla f(x) \rangle = -\epsilon \left\| \nabla f(x) \right\|^2 \geq 0$; we conclude $\nabla f(x) = 0$. Convex optimisation appears in many machine learning problems such as the regularised risk functionals in Section 2.1.5 provided the objective function is decomposed into convex loss functions coupled with convex regularisers.

2.5 Stochastic Optimisation

Recall that in the regularised risk functional framework (Section 2.1.5), the computation of the best function f involves minimising the following term:

$$R_{\text{reg}}(f) = \underbrace{\frac{1}{m}\sum_{i=1}^{m} l(x_i, y_i, f(x_i))}_{R_{\text{emp}}(f)} + \lambda\Omega(f). \tag{2.40}$$

The empirical risk term $R_{\text{emp}}(f)$ is expressed as the average over loss terms $l(x_i, y_i, f(x_i))$ for the whole training set comprising m data samples $\{(x_i, y_i)\}_{i=1}^{m}$. We consider the function space $\mathcal{F}$ to be the set of functions linearly parameterised by $w \in \mathcal{W}$, that is $f(x) := \langle w, \phi(x)\rangle$, where $\mathcal{W} = \mathbb{R}^d$ for linear classifiers or $\mathcal{W}$ is a RKHS for kernel methods. Since each $f \in \mathcal{F}$ is identified by a corresponding parameter w, we rewrite the regulariser $\Omega(f)$, the regularised risk $R_{\text{reg}}(f)$, the empirical risk $R_{\text{emp}}(f)$, and the loss term $l(x_i, y_i, f(x_i))$ as $\Omega(w)$, $R_{\text{reg}}(w)$, $R_{\text{emp}}(w)$ and $l(x_i, y_i, w)$, respectively. Traditional gradient based optimisation methods update the parameter w based on the gradient information accumulated over the whole m training samples. That is, for example in the context of regularised risk functionals, the following iterative scheme is adopted:

$$\begin{aligned} w_{t+1} &\leftarrow w_t - \eta \left\{ \frac{1}{m}\sum_{i=1}^{m} \nabla l(x_i, y_i, w)|_{w_t} + \lambda \nabla\Omega(w)|_{w_t} \right\} \\ t &\leftarrow t+1, \end{aligned} \tag{2.41}$$

for appropriately chosen learning rate or stepsize η. This type of optimisation technique is known as a batch algorithm. For general objective functions, the gradient descent methods converge to a local optimum or saddle point. For convex loss functions and regularisers, the above iterative procedure does not depend on the starting point of the iterates, $w_{t=0}$. This is due to the property that at the fixed point of the iterates w_*, the gradient $\nabla R_{\text{reg}}(w)$ vanishes. By the convexity property, this is a sufficient and necessary condition for w_* to be a (globally) optimal solution. In Internet setting, datasets can however involve large numbers of data samples which might make the batch techniques computationally prohibitive. Batch gradient based methods are also not suited for the situation where data arrives in a continuous stream and a model needs to be built as data arrives. In contrast, stochastic gradient based methods (Bottou, 1998) process small sub-samples (mini-batches) of training samples to calculate the gradient information, and can be more suited to handle large datasets.

2.5.1 Stochastic Gradient Descent

We describe the simplest stochastic optimisation algorithm, called Stochastic Gradient Descent (SGD). Stochastic gradient based methods substitute the regularised risk $R_{\text{reg}}(w)$ by an instantaneous estimate R_t which is computed from a mini-batch of size k comprising a subset of samples drawn from the dataset, $\{(x_i^t, y_i^t)\}_{i=1}^k$:

$$R_t(w) = \frac{1}{k}\sum_{i=1}^{k} l(x_i^t, y_i^t, w) + \lambda\Omega(w). \tag{2.42}$$

If we take $k = 1$, we obtain an algorithm which processes data one at a time as it arrives.

Algorithm 1 Stochastic Gradient Descent

Input maximum iterations T, batch size k, and parameter τ
Initialize $t = 0$ and $w_0 = 0$
while $t < T$ **do**
  Choose a k mini-batch data points $\{(x_i^t, y_i^t)\}_{i=1}^k$
  Calculate gradient $\nabla R_t(w)|_{w_t}$
  Calculate stepsize $\eta_t = \sqrt{\frac{\tau}{\tau+t}}$
  Update $w_{t+1} \leftarrow w_t - \eta_t \nabla R_t(w)|_{w_t}$
  Set $t \leftarrow t + 1$
end while
Return w_T.

The parameter update of SGD then takes the following form:

$$w_{t+1} \leftarrow w_t - \eta_t \nabla R_t(w)|_{w_t}, \tag{2.43}$$

where η_t denotes the stepsize at time t. For convex loss and regulariser and for the stepsize decays as $O(1/\sqrt{t})$, Zinkevich (2003) shows that SGD asymptotically converges to the true minimiser of $R(w)$. One example is a stepsize of the form $\eta_t = \sqrt{\frac{\tau}{\tau+t}}$ with a tuning parameter $\tau > 0$. Refer to Algorithm 1 for pseudo-codes.

2.6 Non-Convex Optimisation

Many machine learning problems, prominently in the real-world Internet application areas, involve non-convex objective functions. Non-convex optimisation

problems are significantly harder than the convex counterpart as they can have many local optima and not all of them are global optima. As gradient based optimisation methods such as in (2.41) converge to a local optimum, the initial guess or the starting point of the optimisation effort has a significant effect on the quality of the solution. We describe one algorithm which is applicable whenever the objective function can be decomposed *explicitly* as the difference of two convex functions.

2.6.1 Convex-ConCave Procedure

The Convex-ConCave Procedure (CCCP) (Yuille & Rangarajan, 2003) works as follow: for a function $f(w)$ that can be decomposed into the difference of two convex functions, that is

$$f(w) = g(w) - h(w), \tag{2.44}$$

where g and h are convex functions; an upper bound can be found by (linearising) replacing h with its first order Taylor expansion at w'

$$f(w) \leq g(w) - h(w') - \langle \nabla h(w)|_{w'}, w - w' \rangle \,. \tag{2.45}$$

The CCCP then performs the following iterative scheme on the upper bound (refer to Algorithm 2 for pseudo-codes):

$$\begin{aligned} w_{t+1} &\leftarrow \arg\min_{w} \{g(w) - h(w_t) - \langle w - w_t, \nabla h(w)|_{w_t} \rangle\} \\ t &\leftarrow t + 1, \end{aligned} \tag{2.46}$$

This upper bound is convex and thus at each iteration of CCCP, a convex optimisation problem is solved (provided the feasible set is a convex set).

Algorithm 2 Convex-ConCave Procedure

Input maximum iterations T, initial parameter w_0, convex functions g, h
Initialize $t = 0$
while $t < T$ **do**
  Update $w_{t+1} \leftarrow \arg\min_{w} \{g(w) - h(w_t) - \langle w - w_t, \nabla h(w)|_{w_t} \rangle\}$
  Set $t \leftarrow t + 1$
end while
Return w_T.

Lemma 16 (Monotonically Decreasing Objective Values) *Let f be a function which can be decomposed into the difference of two convex functions g and h. The sequence $f(w_t), f(w_{t+1}), \ldots$ generated by the iterative scheme in (2.46) is monotonically decreasing.*

Proof We have

$$f(w_{t+1}) \leq g(w_{t+1}) - h(w_t) - \langle \nabla h(w)|_{w_t}, w_{t+1} - w_t \rangle \leq g(w_t) - h(w_t) = f(w_t). \tag{2.47}$$

The first inequality follows from (2.45), while the second inequality follows from (2.46). The decomposable property of the function f gives the last equality. ■

We define a stationary point of a function f as a point where the gradient vanishes, i.e. $\nabla f(w) = 0$. For non-convex functions, this stationary point could be a local minimum, a local maximum or a saddle point. The following lemma shows a relationship between the fixed point of the iterates generated by the iterative scheme in (2.46) and the stationary point of the original objective function f.

Lemma 17 (Fixed Point of CCCP) *A fixed point of the iterates generated by (2.46) is a stationary point of f.*

Proof At each iteration t, the resulting optimisation problem is convex thus the sufficient condition for w_{t+1} to be an optimal solution is

$$\nabla \left(g(w) - h(w_t) - \langle w - w_t, \nabla h(w)|_{w_t} \rangle \right)|_{w_{t+1}} = 0. \tag{2.48}$$

The above means $\nabla g(w)|_{w_{t+1}} = \nabla h(w)|_{w_t}$. Let w_* be a fixed point of the iterates. Then $\nabla g(w)|_{w_*} = \nabla h(w)|_{w_*}$ which implies $\nabla f(w_*) = 0$ and therefore w_* be the stationary point of the original objective function f. ■

We could instead directly perform the gradient based technique in (2.41) for non-convex objective function, however, the CCCP method is arguably more efficient since it converges to a stationary point without the need to play around with the learning rate η.

In the DC (difference of convex) programming literature, Dinh & An (1988) propose a general purpose solver for solving a difference of convex functions where

g and h are lower semi-continuous convex functions. The class of lower semi-continuous functions forms a larger class of functions than the class of differentiable functions. Whenever h is differentiable, the algorithm of Dinh & An (1988) reduces to CCCP.

The following remark implies that the above CCCP is applicable to large class of optimisation problems (Yuille & Rangarajan, 2003, Theorem 1).

Remark 18 (Difference of Convex Functions) *Any function with a bounded Hessian can always be decomposed into the difference of two convex functions.*

Part I

Formulation and Solution for Non-Standard Machine Learning Problems

Chapter 3

Estimating Labels from Label Proportions

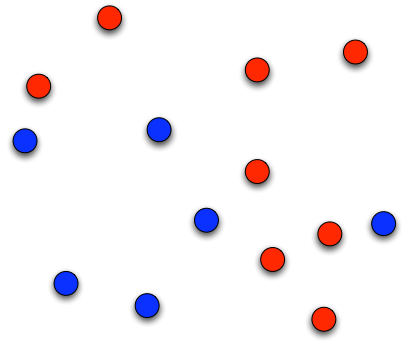
Traditionally, different types of learning problems assume different problem settings. In *supervised* learning, we are given sets of labelled instances. Another learning type called *unsupervised* learning focuses on the setting where unlabelled instances are given. Recently, it has been realised that unlabelled instances when used in conjunction with a small amount of labelled instances can deliver considerable learning performance improvement in comparison to using labelled instances alone. This leads to a *semi-supervised* or *transduction* learning setting.

In this chapter, we introduce a learning setting where groups of unlabelled instances are given. The number of group is at least as many as number of classes. Each group is endowed with information on class label *proportions*. We called this informative group as aggregate (see Figure 3 for an illustration). The goal of learning is to predict the labels of another set of observations, possibly with known label proportions.

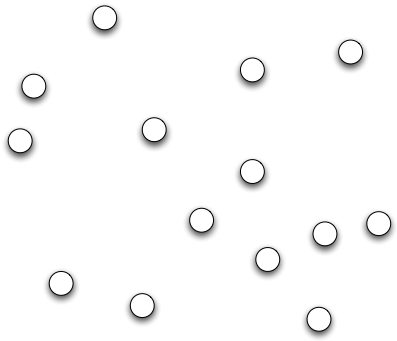
This type of learning problem appears in areas like e-commerce, politics, spam filtering and improper content detection, as we illustrate below.

3.1 Motivating Examples

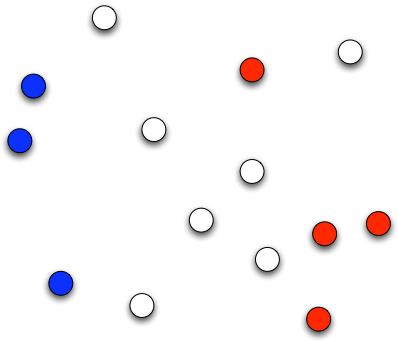
Assume that an Internet services company wants to increase its profit in sales. Obviously sending out discount coupons will increase sales, but sending coupons to customers who would have purchased the goods anyway decreases the margins. Alternatively, failing to send coupons to customers who would only buy in case of a discount reduces overall sales. We would like to identify the class of would-be



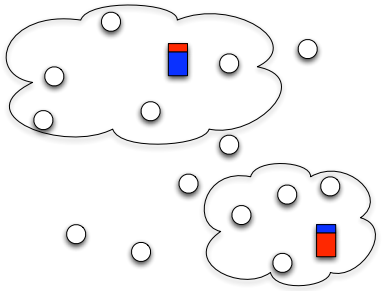
(a) Supervised Learning



(b) Unsupervised Learning



(c) Semi - supervised Learning



(d) Learning from Proportions

Figure 3.1: Different types of learning problems (colours encode class labels). 3.1(a) - `supervised learning`: only labelled instances are given; 3.1(b) - `unsupervised learning`: only unlabelled instances are given; 3.1(c) - `semi-supervised learning`: both labelled and unlabelled instances are given; 3.1(d): `learning from proportions`: at least as many data aggregates (groups of data with their associated class label proportions) as there are number of classes are given.

customers who are most likely to change their purchase decision when receiving a coupon. The problem is that there is no direct access to a sample of would-be customers. Typically only a sample of people who buy regardless of coupons (those who bought when there was no discount) and a mixed sample (those who bought when there was discount) are available. The mixing *proportions* can be reliably estimated using random assignment to control and treatment groups. How can we use this information to determine the would-be customers?

Politicians face the same problem. They can rely on a set of always-favourable voters who will favour them regardless, plus a set of swing voters who will make their decision dependent on what the candidates offer. Since the candidate's resources (finance, ability to make election promises, campaign time) are limited, it is desirable for them to focus their attention on that part of the demographic where they can achieve the largest gains. Previous elections can directly reveal the profile of those who favour regardless, that is those who voted in favour where low campaign resources were committed. Those who voted in favour where substantial resources were committed can be either swing voters or always-favourable. So in a typical scenario there is no separate sample of swing voters.

Likewise, consider the problem of spam filtering. Datasets of spam are likely to contain almost pure spam (this is achieved e.g. by listing e-mails as spam bait), while user's inboxes typically contain a mix of spam and non-spam. We would like to use the inbox data to improve estimation of spam. In many cases it is possible to estimate the *proportions* of spam and non-spam in a user's inbox much more cheaply than the actual labels. We would like to use this information to categorise e-mails into spam and non-spam.

Similarly, consider the problem of filtering images with "improper content". Datasets of such images are readily accessible thanks to user feedback, and it is reasonable to assume that this labelling is highly reliable. However the rest of images on the Internet (those not labelled) is a far larger dataset, albeit without labels (after all, this is what we would like to estimate the labels for). That said, it is considerably cheaper to obtain a good estimate of the *proportions* of proper and improper content in addition to having one dataset of images being of likely improper content. We would like to obtain a classifier based on this information.

3.2 Problem Definition

In this chapter, we present a method that makes use of the knowledge of label proportions *directly*. As motivated by the above examples, our method would be practically useful in many domains such as identifying potential customers, potential voters, spam e-mails and improper images. We also prove bounds indicating that the estimates obtained are close to those from a fully labelled scenario.

Before defining the problem, we emphasise that the formal setting is more general than the above examples might suggest. More specifically, we may not require *any* label to be known, only their proportions within each of the involved datasets. Also the general problem is not restricted to the binary case but instead can deal with large numbers of classes. Finally, it is possible to apply our method to problems where the *test label proportions* are unknown, too. This simple modification allows us to use this technique whenever covariate shift via label bias is present.

Formally, in a learning from proportions setting, we are given n sets of observations $X_i = \left\{x_1^i, \ldots, x_{m_i}^i\right\}$ of respective sample sizes m_i (calibration set) $i = 1, \ldots, n$ as well as a set $X = \{x_1, \ldots, x_m\}$ (test set). Moreover, we are given the fractions π_{iy} of labels $y \in \mathcal{Y}$ ($|\mathcal{Y}| \leq n$) contained in each set X_i. These fractions form a full (column) rank mixing matrix, $\pi \in \mathbb{R}^{n \times |\mathcal{Y}|}$ with the constraint that each row sums up to 1 and all entries are nonnegative. The marginal probability $p(y)$ of the test set X may or may not be known. Note that the label dictionaries $\mathcal{Y}_i$ do not need to be the same across all sets i (define $\mathcal{Y} := \cup_i \mathcal{Y}_i$) and we also allow for $\pi_{iy} = 0$ if needed. It is our goal to design algorithms which are able to obtain conditional class probability estimates $p(y|x)$ solely based on this information.

As an illustration, take the spam filtering example. We have X_1 = "mail in spam box" (only spam) and X_2 = "mail in inbox" (spam mixed with non-spam). Also suppose that we may know the proportion of spam vs non-spam in our inbox is 1 : 9. That means, we know: $\pi_{1,\text{spam}} = 1.0, \pi_{1,\text{non-spam}} = 0, \pi_{2,\text{spam}} = 0.1$ and $\pi_{2,\text{non-spam}} = 0.9$. The test set X then may be X_2 itself, for example. Thus, the marginal probability of the test set will simply be: $p(y = \text{spam}) = 0.1, p(y = \text{non} - \text{spam}) = 0.9$. The goal is to find $p(\text{spam}|\text{mail})$ in X. Note that, in general, our setting is different and more difficult than that of transduction. The latter requires at least some labelled instances of *all classes* are given. In the spam filtering example, we have no pure non-spam instances.

Key to our proposed solution is a conditional independence assumption, $x \perp\!\!\!\perp$

$i \,|\, y$. In other words, we assume that the *conditional* distribution of x is independent of the index i, as long as we know the label y. This is a crucial assumption: after all, we want the distributions within each class to be independent of which aggregate they can be found in. If this were not the case it would be impossible to infer about the distribution on the test set from the (biased) distributions over the aggregates.

3.3 The Model

Our idea relies on uniform convergence properties of the expectation operator and of corresponding risk functionals (Altun & Smola, 2006; Dudík & Schapire, 2006). In doing so, we are able to design estimators with the same performance guarantees in terms of uniform convergence as those with full access to the label information.

At the heart of our reasoning lies the fact that many estimators rely on data by solving a convex optimisation problem. We begin our exposition by discussing how this strategy can be employed in the context of exponential families. Subsequently we state convergence guarantees and we discuss how our method can be extended to other estimates such as Csiszar and Bregman divergences and other function spaces.

3.3.1 Exponential Families

Denote by $\mathcal{X}$ the space of observations and let $\mathcal{Y}$ be the space of labels. Moreover, let $\phi(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathcal{H}$ be a feature map into a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{F}$ with kernel $k((x, y), (x', y'))$. In this case we may state conditional exponential models via

$$p(y|x, \theta) = \exp\left(\langle \phi(x, y), \theta \rangle - g(\theta|x)\right) \text{ with } g(\theta|x) = \log \sum_{y \in \mathcal{Y}} \exp \langle \phi(x, y), \theta \rangle , \tag{3.1}$$

where the normalisation g is called the log-partition function, often referred to as the cumulant generating function. Note that while in general there is no need for $\mathcal{Y}$ to be discrete, we make this simplifying assumption in order to be able to reconstruct the class probabilities efficiently. For $\{(x_i, y_i)\}$ drawn i.i.d. from a

distribution $p(x, y)$ on $\mathcal{X} \times \mathcal{Y}$ the conditional log-likelihood is given by

$$\log p(Y|X, \theta) = \sum_{i=1}^{m} [\langle \phi(x_i, y_i), \theta \rangle - g(\theta|x_i)] = m \langle \mu_{XY}, \theta \rangle - \sum_{i=1}^{m} g(\theta|x_i) \tag{3.2}$$

where the empirical mean in feature space μ_{XY} is defined as in Table 3.1. In order to avoid overfitting one commonly maximises the log-likelihood penalised by a prior $p(\theta)$. This means that we need to solve the following optimisation problem

$$\theta^* := \operatorname*{argmin}_{\theta} [-\log\{p(Y|X, \theta)p(\theta)\}]. \tag{3.3}$$

For instance, for a Gaussian prior on θ, i.e. for

$$-\log p(\theta) = \lambda \|\theta\|^2 + \text{const.} \tag{3.4}$$

we have

$$\theta^* = \operatorname*{argmin}_{\theta} \left[\sum_{i=1}^{m} g(\theta|x_i) - m \langle \mu_{XY}, \theta \rangle + \lambda \|\theta\|^2 \right]. \tag{3.5}$$

The problem is that in our setting we do not know the labels y_i, so the sufficient statistics μ_{XY} cannot be computed exactly. Note, though that the only place where the labels enter the estimation process is via the mean μ_{XY}. Our strategy is to exploit the fact that this quantity, however, is statistically well behaved and converges under relatively mild technical conditions at rate $O(m^{-\frac{1}{2}})$ to its expected value

$$\mu_{xy} := \mathbf{E}_{(x,y)\sim p(x,y)}[\phi(x, y)], \tag{3.6}$$

as will be shown in Theorem 21. Our goal therefore will be to estimate μ_{xy} and use it as a proxy for μ_{XY}, and only then solve (3.5) with the estimated $\hat{\mu}_{XY}$ instead of μ_{XY}. We will discuss explicit convergence guarantees in Section 3.5 after describing how to compute the mean operator in detail.

3.3.2 Estimating the Mean Element

In order to obtain θ^* we would need μ_{XY}, which is impossible to compute exactly, since we do not have the labels Y. However, we know that μ_{XY} converges to μ_{xy}. Hence, if we are able to approximate μ_{xy} then this, in turn, will be a good estimate for μ_{XY}.

Table 3.1: Major quantities of interest

Numbers on the left represent the order in which the corresponding quantity is computed in the algorithm (letters denote the variant of the algorithm: 'a' for general feature map $\phi(x, y)$ and 'b' for factorising feature map $\phi(x, y) = \psi(x) \otimes \varphi(y)$). Lowercase subscripts refer to model expectations, uppercase subscripts are sample averages.

Expectations with respect to the model:

$$\begin{aligned} \mu_{xy} &:= \mathbf{E}_{(x,y)\sim p(x,y)}[\phi(x,y)] \\ \mu_x^{\text{class}}[y,y'] &:= \mathbf{E}_{(x)\sim p(x|y)}[\phi(x,y')] \\ \mu_x^{\text{set}}[i,y'] &:= \mathbf{E}_{(x)\sim p(x|i)}[\phi(x,y')] \\ \mu_x^{\text{class}}[y] &:= \mathbf{E}_{(x)\sim p(x|y)}[\psi(x)] \\ \mu_x^{\text{set}}[i] &:= \mathbf{E}_{(x)\sim p(x|i)}[\psi(x)] \end{aligned}$$

Expectations with respect to data:

$$\begin{aligned} \mu_{XY} &:= \tfrac{1}{m}\textstyle\sum_{i=1}^{m} \phi(x_i, y_i) \\ \text{(1a)}\quad \mu_X^{\text{set}}[i,y'] &:= \tfrac{1}{m_i}\textstyle\sum_{x\in X_i} \phi(x,y') \ \textbf{(known)} \\ \text{(1b)}\quad \mu_X^{\text{set}}[i] &:= \tfrac{1}{m_i}\textstyle\sum_{x\in X_i} \psi(x) \ \textbf{(known)} \end{aligned}$$

Estimates:

$$\begin{aligned} \text{(2)}\quad \hat{\mu}_x^{\text{class}} &= (\pi^\top\pi)^{-1}\pi^\top \mu_X^{\text{set}} \\ \text{(3a)}\quad \hat{\mu}_{XY} &= \textstyle\sum_{y\in\mathcal{Y}} p(y)\hat{\mu}_x^{\text{class}}[y,y] \\ \text{(3b)}\quad \hat{\mu}_{XY} &= \textstyle\sum_{y\in\mathcal{Y}} p(y)\varphi(y)\otimes\hat{\mu}_x^{\text{class}}[y] \\ \text{(4)}\quad \hat{\theta}^* &\text{ solution of (3.5) for } \mu_{XY} = \hat{\mu}_{XY}. \end{aligned}$$

Our quest is therefore as follows: express μ_{xy} as a linear combination over expectations with respect to the distributions on the datasets $X_1, \ldots, X_n$ (where $n \geq |\mathcal{Y}|$). Secondly, show that the expectations of the distributions having generated the sets X_i ($\mu_x^{\text{set}}[i, y']$, see Table 3.1), can be approximated by empirical means ($\mu_X^{\text{set}}[i, y']$, see Table 3.1). Finally, we need to combine both steps to provide guarantees for μ_{XY}.

It will turn out that in certain cases some of the algebra can be sidestepped, in particular whenever we may be able to identify several sets with each other (e.g. the test set X is one of the calibration datasets X_i) or whenever $\phi(x, y)$ factorizes into $\psi(x) \otimes \varphi(y)$. We will discuss these simplifications in Section 3.4.

Mean Element Since μ_{xy} is a linear operator mapping $p(x, y)$ into a Hilbert Space we may expand μ_{xy} via

$$\mu_{xy} = \mathbf{E}_{(x,y)\sim p(x,y)}[\phi(x, y)] = \sum_{y\in\mathcal{Y}} p(y)\mathbf{E}_{x\sim p(x|y)}[\phi(x, y)] = \sum_{y\in\mathcal{Y}} p(y)\mu_x^{\text{class}}[y, y] \quad (3.7)$$

where the shorthand $\mu_x^{\text{class}}[y, y]$ is defined in Table 3.1. This means that if we were able to compute $\mu_x^{\text{class}}[y, y]$ we would be able to "reassemble" μ_{xy} from its individual components. We now show that $\mu_x^{\text{class}}[y, y]$ can be estimated directly.

Our conditional independence assumption, $p(x|y, i) = p(x|y)$, yields the following:

$$p(x|i) = \sum_y p(x|y, i)p(y|i) = \sum_y p(x|y)\pi_{iy}. \quad (3.8)$$

In the above equation, we form a mixing matrix π with the element $\pi_{iy} = p(y|i)$. This allows us to define the following means

$$\mu_x^{\text{set}}[i, y'] := \mathbf{E}_{x\sim p(x|i)}[\phi(x, y')] \overset{(3.8)}{=} \sum_y \pi_{iy}\mu_x^{\text{class}}[y, y'].$$

Note that in order to compute $\mu_x^{\text{set}}[i, y']$ we do *not* need any label information with respect to $p(x|i)$. It is simply the expectation of $\phi(\cdot, y')$ on the distribution of bag i. However, since we have at least $|\mathcal{Y}|$ of those equations and we assumed that π has full column rank, they allow us to solve a linear system of equations and compute $\mu_x^{\text{class}}[y, y]$ from $\mu_x^{\text{set}}[i, y']$ for all i. In shorthand we may use

$$\mu_x^{\text{set}} = \pi\mu_x^{\text{class}} \text{ and hence } \mu_x^{\text{class}} = (\pi^\top\pi)^{-1}\pi^\top\mu_x^{\text{set}} \quad (3.9)$$

to compute $\mu_x^{\text{class}}[y, y]$ for all $y \in \mathcal{Y}$. With some slight abuse of notation we have μ_x^{class} and μ_x^{set} represent the *matrices* of terms $\mu_x^{\text{class}}[y, y']$ and $\mu_x^{\text{set}}[i, y']$ respectively. There will be as many matrices as the dimensions of $\phi(x, y)$, thus (3.9) has to be solved separately for each dimension of $\phi(x, y)$.

Obviously we cannot compute $\mu_x^{\text{set}}[i, y']$ explicitly, since we only have *samples* from $p(x|i)$. However the same convergence results governing the convergence of μ_{XY} to μ_{xy} also hold for the convergence of $\mu_X^{\text{set}}[i, y']$ to $\mu_x^{\text{set}}[i, y']$. Hence we may use the empirical average $\mu_X^{\text{set}}[i, y']$ as the estimate for $\mu_x^{\text{set}}[i, y']$ and from that find an estimate for μ_{XY}.

Big Picture Overall, our strategy is as follows: use empirical means on the bags X_i to approximate expectations with respect to the bag distribution. Use

Algorithm 3 Learning from Label Proportions

Input datasets X, $\{X_i\}$, probabilities π_{iy} and $p(y)$
for $i = 1$ **to** n **and** $y' \in \mathcal{Y}$ **do**
  Compute empirical means $\mu_X^{\text{set}}[i, y']$
end for
Compute $\hat{\mu}_x^{\text{class}} = (\pi^\top \pi)^{-1} \pi^\top \mu_X^{\text{set}}$
Compute $\hat{\mu}_{XY} = \sum_{y \in \mathcal{Y}} p(y) \hat{\mu}_x^{\text{class}}[y, y]$
Solve the minimisation problem

$$\hat{\theta}^* = \operatorname*{argmin}_\theta \left[\sum_{i=1}^{m} g(\theta | x_i) - m \left\langle \hat{\mu}_{XY}, \theta \right\rangle + \lambda \|\theta\|^2 \right]$$

Return $\hat{\theta}^*$.

the latter to compute expectations with respect to a given label, and finally, use the means conditional on the label distribution to obtain μ_{xy} which is a good proxy for μ_{XY} (see Algorithm 3).

$$\mu_X^{\text{set}}[i, y'] \longrightarrow \mu_x^{\text{set}}[i, y'] \longrightarrow \mu_x^{\text{class}}[y, y'] \longrightarrow \mu_{xy} \longrightarrow \mu_{XY}$$

For the first and last step in the chain we can invoke uniform convergence results. The remaining two steps in the chain follow from linear algebra. As we shall see, whenever there are considerably more bags than classes we can exploit the overdetermined system to our advantage to reduce the overall estimation error and use a rescaled version of (3.9).

3.4 Special Cases

In some cases the calculations described in Algorithm 3 can be carried out more efficiently. They arise whenever the matrix π has special structure or whenever the test set and one of the training sets coincide. Moreover, we may encounter situations where the fractions of observations in the test set are unknown and we would like, nonetheless, to find a good proxy for μ_{XY}.

3.4.1 Minimal number of sets

Assuming that $|\mathcal{Y}| = n$ and that π has full rank it follows that $(\pi^\top \pi)^{-1} \pi^\top = \pi^{-1}$. Hence we can obtain the proxy for μ_{XY} more directly via $\mu_x^{\text{class}} = \pi^{-1} \mu_x^{\text{set}}$.

3.4.2 Testing on one of the calibration sets

Note that there is no need for requiring that the test set X be different from one of the calibration sets (vide example in Problem Definition). In particular, when $X = X_i$ the uncertainty in the estimate of μ_{XY} can be greatly reduced provided that the estimate of μ_{XY} as given in (3.9) contains a large fraction of the mean of at least one of the classes. We will discuss this situation in more detail when it comes to binary classification since there the advantages will be most obvious.

3.4.3 Special feature map

Whenever the feature map $\phi(x, y)$ factorises into $\psi(x) \otimes \varphi(y)$ we can simplify calculation of the means considerably. More specifically, instead of estimating $O(|\mathcal{Y}| \cdot n)$ parameters we only require calculation of $O(n)$ terms. The reason for this is that we may pull the dependency on y out of the expectations. Defining $\mu_x^{\text{class}}[y], \mu_x^{\text{set}}[i]$, and $\mu_X^{\text{set}}[i]$ as in Table 3.1 allows us to simplify

$$\hat{\mu}_{XY} = \sum_{y \in \mathcal{Y}} p(y)\varphi(y) \otimes \hat{\mu}_x^{\text{class}}[y] \text{ where } \hat{\mu}_x^{\text{class}} = (\pi^\top \pi)^{-1} \pi^\top \mu_X^{\text{set}}. \tag{3.10}$$

Here the last equation is understood to apply to the vector of means $\mu_x := (\mu[1], \ldots, \mu[n])$ and μ_X accordingly. A significant advantage of (3.10) is that we only need to perform $O(n)$ averaging operations rather than $O(n \cdot |\mathcal{Y}|)$. Obviously the cost of computing $(\pi^\top \pi)^{-1} \pi^\top$ remains unchanged but the latter is negligible compared to the operations in Hilbert Space. Note that $\psi(x) \in \mathbb{R}^D$ denotes an arbitrary feature representation of the inputs, which in many cases can be defined implicitly via a kernel function. As the joint feature map $\phi(x, y)$ factorises into $\psi(x) \otimes \varphi(y)$, we can write the inner product in the joint representation as $\langle \phi(x, y), \phi(x', y') \rangle = \langle \psi(x), \psi(x') \rangle \langle \varphi(y), \varphi(y') \rangle = k(x, x')k(y, y')$. In general, the kernel function on inputs and labels can be different. Specifically, for a label diagonal kernel $k(y, y') = \delta(y, y')$, the standard winner-takes-all multiclass classification is recovered (Tsochantaridis et al., 2005). With this setting, the input feature $\psi(x)$ can be defined implicitly via a kernel function by invoking the Representer Theorem (Theorem 6).

3.4.4 Binary classification

One may show (Hofmann et al., 2008) that the feature map $\phi(x, y)$ takes on a particularly appealing form of $\phi(x, y) = y\psi(x)$ where $y \in \{\pm 1\}$. This follows

since we can always re-calibrate $\langle \phi(x,y), \theta \rangle$ by an offset independent of y such that $\phi(x,1) + \phi(x,-1) = 0$.

If we moreover assume that X_1 only contains class 1 and $X_2 = X$ contains a mixture of classes with labels 1 and -1 with proportions $p(1) =: \rho$ and $p(-1) = 1 - \rho$ respectively, we obtain the mixing matrix

$$\pi = \begin{bmatrix} 1 & 0 \\ \rho & 1-\rho \end{bmatrix} \Rightarrow \pi^{-1} = \begin{bmatrix} 1 & 0 \\ \frac{-\rho}{1-\rho} & \frac{1}{1-\rho} \end{bmatrix}$$

Plugging this into (3.10) yields

$$\begin{aligned} \hat{\mu}_{XY} &= \rho \mu_X^{\text{set}}[1] - (1-\rho)\left[\tfrac{-\rho}{1-\rho}\mu_X^{\text{set}}[1] + \tfrac{1}{1-\rho}\mu_X^{\text{set}}[2]\right] \\ &= 2\rho\mu_X^{\text{set}}[1] - \mu_X^{\text{set}}[2]. \end{aligned} \tag{3.11}$$

Consequently, taking a simple weighted difference between the averages on two sets, e.g. one set containing spam whereas the other one containing an unlabelled mix of spam and non-spam, allows one to obtain the sufficient statistics needed for estimation.

3.4.5 Overdetermined Systems

Assume that we have significantly more bags n than class labels $|\mathcal{Y}|$, possibly with varying numbers of observations m_i per bag. In this case it would make sense to find a weighting of the bags such that those which are largest and most relevant for the test set are given the highest degree of importance. Instead of stating the problem as one of solving a linear system we now restate it as one of solving an approximation problem. To simplify notation we assume that the feature map factorises, i.e. that $\phi(x,y) = \psi(x) \otimes \varphi(y)$. A weighted linear combination of the squared discrepancy between the class means and the set means is given by

$$\underset{\mu_x^{\text{class}}}{\text{minimise}} \sum_{i=1}^{n} w_i \left\| \mu_X^{\text{set}}[i] - \sum_{y \in \mathcal{Y}} \pi_{iy} \mu_x^{\text{class}}[y] \right\|^2 \tag{3.12}$$

where w_i are some previously chosen weights which reflect the importance of each bag. Typically we might choose $w_i = O(m_i^{-\frac{1}{2}})$ to reflect the fact that convergence between empirical means and expectations scales with $O(m^{-\frac{1}{2}})$. Before we discuss specific methods for choosing a weighting, let us review the statistical properties of the estimator.

Remark 19 (Underdetermined Systems) *Similarly, when we have less bags n than class labels $|\mathcal{Y}|$, we can state the problem as one of solving a regularised least squares problem as follows*

$$\underset{\mu_x^{\text{class}}}{\text{minimise}} \sum_{i=1}^{n} \left\| \mu_X^{\text{set}}[i] - \sum_{y\in\mathcal{Y}} \pi_{iy} \mu_x^{\text{class}}[y] \right\|^2 + \lambda\Omega(\mu_x^{\text{class}}[y] \,\forall\, y \in \mathcal{Y}) \tag{3.13}$$

For example, we can let $\Omega(\mu_x^{\text{class}}[y] \,\forall\, y \in \mathcal{Y}) = \sum_{y\in\mathcal{Y}} \left\|\mu_x^{\text{class}}[y] - \mu_x^{\text{class}}[y+1]\right\|^2$. This makes sense whenever different labels have related means $\mu_x^{\text{class}}[y]$.

3.5 Convergence Bounds

The obvious question is how well $\hat{\mu}_{XY}$ manages to approximate μ_{XY} and secondly, how badly any error in estimating μ_{XY} would affect the overall quality of the solution. We approach this problem as follows: first we state the uniform convergence properties of μ_{XY} and similar empirical operators relative to μ_{xy}. Secondly, we apply those bounds to the cases discussed above, and thirdly, we show that the approximate minimiser of the log-posterior has a bounded deviation from what we would have obtained by knowing μ_{XY} exactly. Much of the reasoning follows the ideas of Altun & Smola (2006).

3.5.1 Uniform Convergence for Mean Elements

An important tool in studying uniform convergence properties of random variables are Rademacher averages (Ledoux & Talagrand, 1991; Mendelson, 2002). They are needed to state the key results in our context.

Definition 20 (Rademacher Averages) *Let $\mathcal{X}$ be a domain and p a distribution on $\mathcal{X}$ and assume that $X := \{x_1, \ldots, x_m\}$ is drawn iid from p. Moreover, let $\mathcal{F}$ be a class of functions $\mathcal{X} \to \mathbb{R}$. Furthermore denote by σ_i Rademacher random variables, i.e. $\{\pm 1\}$ valued with zero mean. The Rademacher average is*

$$R_m(\mathcal{F}, p) := \mathbf{E}_X \mathbf{E}_\sigma \left[\sup_{f\in\mathcal{F}} \left| \frac{1}{m} \sum_{i=1}^{m} \sigma_i f(x_i) \right| \right]. \tag{3.14}$$

This quantity measures the flexibility of the function class $\mathcal{F}$ — in our case linear functions in $\phi(x, y)$. In other words, the Rademacher average measure the ability of the function class to predict random labels. Thus, choosing a function class with low Rademacher average lowers the chance of detecting spurious pattern. Altun & Smola (2006) state the following result:

Theorem 21 (Convergence of Empirical Means) *Denote by $\phi : \mathcal{X} \to \mathcal{B}$ a map into a Banach space $\mathcal{B}$, denote by $\mathcal{B}^*$ its dual space and let $\mathcal{F}$ the class of linear functions on $\mathcal{B}$ with bounded $\mathcal{B}^*$ norm by 1. Let $R > 0$ such that for all $f \in \mathcal{F}$ we have $|f(x)| \leq R$. Moreover, assume that X is an m-sample drawn from p on $\mathcal{X}$. For $\bar{\epsilon} > 0$ we have that with probability at least $1 - \exp(-\bar{\epsilon}^2 m/R^2)$ the following holds:*

$$\|\mu_X - \mu_x\|_{\mathcal{B}} \leq 2R_m(\mathcal{F}, p) + \bar{\epsilon} \tag{3.15}$$

Theorem 22 (Bartlett & Mendelson (2002)) *Whenever $\mathcal{B}$ is a Reproducing Kernel Hilbert Space with kernel $k(x, x')$ the Rademacher average can be bounded from above by $R_m(\mathcal{F}) \leq m^{-\frac{1}{2}} \left[\mathbf{E}_x[k(x, x)]\right]^{\frac{1}{2}}$*

Our approximation error can be bounded as follows. From the triangle inequality we have:

$$\|\hat{\mu}_{XY} - \mu_{XY}\| \leq \|\hat{\mu}_{XY} - \mu_{xy}\| + \|\mu_{xy} - \mu_{XY}\|.$$

For the second term we may employ Theorem 21 directly. To bound the first term note that by linearity

$$\epsilon := \hat{\mu}_{XY} - \mu_{xy} = \sum_y p(y) \left[(\pi^\top \pi)^{-1} \pi^\top \hat{\epsilon}\right]_{y,y} \tag{3.16}$$

where we define the "matrix" of coefficients

$$\hat{\epsilon}[i, y'] := \mu_x^{\text{set}}[i, y'] - \mu_X^{\text{set}}[i, y']. \tag{3.17}$$

In the more general case of overdetermined systems we have

$$\epsilon = \sum_y p(y) \left[(\pi^\top W \pi)^{-1} \pi^\top W \hat{\epsilon}\right]_{y,y} \tag{3.18}$$

Now note that all $\hat{\epsilon}[i, y']$ also satisfy the conditions of Theorem 21 since the sets X_i are drawn iid from the distributions $p(x|i)$ respectively. We may bound each term individually in this fashion and subsequently apply the union bound to ensure that all $n \cdot |\mathcal{Y}|$ components satisfy the constraints. Hence each of the terms needs to satisfy the constraint with probability $1 - \delta/(n|\mathcal{Y}|)$ to obtain an overall bound with probability $1 - \delta$. To obtain bounds we would need to bound the linear operator mapping $\hat{\epsilon}$ into ϵ.

Note that this statement can be improved since all errors $\hat{\epsilon}[i, y']$ and $\hat{\epsilon}[j, y']$ for $i \neq j$ are independent of each other simply by the fact that each bag X_i was sampled independently from the other. We will discuss this in the context of choosing a practically useful value of W below.

3.5.2 Special Cases

A closed form solution in the general case is not particularly useful since it depends heavily on the kernel k, the mixing proportions π and the class probabilities on the test set. However, for a number of special cases it is possible to provide more detailed explicit analysis: firstly the situation where $\phi(x,y) = \psi(x) \otimes \varphi(y)$ and secondly, the binary classification setting where $\phi(x,y) = y\psi(x)$ and $X_2 = X$, where much tighter bounds are available.

Special feature map with full rank

Here we only need to deal with n rather than with $n \times |\mathcal{Y}|$ empirical estimates, i.e. $\mu_X^{\text{set}}[i]$ vs. $\mu_X^{\text{set}}[i,y']$. Hence (3.16) and (3.17) specialise to

$$\epsilon = \sum_y p(y) \sum_{i=1}^{n} \varphi(y) \otimes \left[(\pi^\top \pi)^{-1} \pi^\top\right]_{yi} \hat{\epsilon}[i] \tag{3.19}$$

$$\hat{\epsilon}[i] := \mu_x^{\text{set}}[i] - \mu_X^{\text{set}}[i]. \tag{3.20}$$

Assume that with high probability each $\hat{\epsilon}[i]$ satisfies $\|\hat{\epsilon}[i]\| \leq c_i$ (we will deal with the explicit constants c_i later). Moreover, assume for simplicity that $|\mathcal{Y}| = n$ and that π has full rank (otherwise we need to follow through on our expansion using $(\pi^\top\pi)^{-1}\pi^\top$ instead of π^{-1}). This implies that

$$\begin{aligned} \|\epsilon\|^2 &= \sum_{i,j} \langle \hat{\epsilon}[i], \hat{\epsilon}[j] \rangle \times \sum_{y,y'} p(y)p(y')k(y,y') \left[\pi^{-1}\right]_{yi} \left[\pi^{-1}\right]_{y'j} \\ &\leq \sum_{i,j} c_i c_j \left| \left[\pi^{-1}\right]^\top K^{y,p} \pi^{-1} \right|_{ij} \end{aligned} \tag{3.21}$$

where $K^{y,p}_{y,y'} = k(y,y')p(y)p(y')$. Combining several bounds we have the following theorem:

Theorem 23 *Assume that we have n sets of observations X_i of size m_i, each of which drawn from distributions with probabilities π_{iy} of observing data with label y. Moreover, assume that $k((x,y),(x',y')) = k(x,x')k(y,y') \geq 0$ where $k(x,x) \leq 1$ and $k(y,y) \leq 1$. Finally, assume that $m = |X|$. In this case the mean element μ_{XY} can be estimated by $\hat{\mu}_{XY}$ with probability at least $1-\delta$ with precision*

$$\|\mu_{XY} - \hat{\mu}_{XY}\| \leq \left[2 + \sqrt{\log((n+1)/\delta)}\right] \times \left[m^{-\frac{1}{2}} + \left[\sum_{i,j} m_i^{-\frac{1}{2}} m_j^{-\frac{1}{2}} \left| \left[\pi^{-1}\right]^\top K^{y,p} \pi^{-1} \right|_{ij}\right]^{\frac{1}{2}}\right]$$

Proof We begin our argument by noting that both for $\phi(x,y)$ and for $\psi(x)$ the corresponding Rademacher averages R_m for functions of RKHS norm bounded by 1 is bounded by $m^{-\frac{1}{2}}$. This is a consequence of all kernels being bounded by 1 in Theorem 22 and $k \geq 0$.

Next note that in Theorem 21 we may set $R = 1$, since for $\|f\| \leq 1$ and $k((x,y),(x,y)) \leq 1$ and $k(x,x) \leq 1$ it follows from the Cauchy Schwartz inequality that $|f(x)| \leq 1$. Solving $\delta \leq \exp -m\epsilon^2$ for ϵ yields $\epsilon \leq m^{-\frac{1}{2}} \left[2 + \sqrt{\log(1/\delta)}\right]$.

Finally, note that we have $n+1$ deviations which we need to bound: one between μ_{XY} and μ_{xy}, and n for each of the $\epsilon[i]$ respectively. Dividing the failure probability δ into $n+1$ cases yields bounds of the form $m^{-\frac{1}{2}} \left[2 + \sqrt{\log((n+1)/\delta)}\right]$ and $m_i^{-\frac{1}{2}} \left[2 + \sqrt{\log((n+1)/\delta)}\right]$ respectively. Plugging all error terms into (3.21) and summing over terms yields the claim and substituting this back into the triangle inequality proves the claim. ■

Binary Classification

Next we consider the special case of binary classification where $X_2 = X$. Using (3.11) we see that the corresponding estimator is given by

$$\hat{\mu}_{XY} = 2\rho\mu_X^{\text{set}}[1] - \mu_X^{\text{set}}[2]. \tag{3.22}$$

Since $\hat{\mu}_{XY}$ shares a significant fraction of terms with μ_{XY} we are able to obtain tighter bounds as follows:

Theorem 24 *With probability $1-\delta$ (for $1 > \delta > 0$) the following bound holds:*

$$\|\hat{\mu}_{XY} - \mu_{XY}\| \leq 2\rho \left[2 + \sqrt{\log(2/\delta)}\right] \left[m_1^{-\frac{1}{2}} + m_+^{-\frac{1}{2}}\right]$$

m_+ is the number of observations with $y = 1$ in X_2.

Proof Denote by $\mu[X_+]$ and $\mu[X_-]$ the averages over the subsets of X_2 with positive and negative labels respectively. By construction we have that

$$\mu_{XY} = \rho\mu[X_+] - (1-\rho)\mu[X_-]; \quad \hat{\mu}_{XY} = 2\rho\mu_X^{\text{set}}[1] - \rho\mu[X_+] - (1-\rho)\mu[X_-]$$

Taking the difference yields $2\rho\left[\mu_X^{\text{set}}[1] - \mu[X_+]\right]$. To prove the claim note that we may use Theorem 21 both for $\left\|\mu_X^{\text{set}}[1] - \mathbf{E}_{x\sim p(x|y=1)}[\psi(x)]\right\|$ and for $\left\|\mu[X_+] - \mathbf{E}_{x\sim p(x|y=1)}[\psi(x)]\right\|$. Taking the union bound and summing over terms proves the claim. ■

The bounds we provided show that $\hat{\mu}_{XY}$ converges at the same rate to μ_{xy} as μ_{XY} does, assuming that the sizes of the sets X_i increase at the same rate as X.

Overdetermined Systems

Given the optimal value of weighting W, the class mean can be reconstructed as a solution of a weighted least square problem in (3.12) and this minimiser is given by

$$\hat{\mu}_x^{\text{class}} = (\pi^\top W \pi)^{-1} \pi^\top W \mu_X^{\text{set}} \text{ where } W = \text{diag}(w_1, \ldots, w_n) \text{ and } w_i > 0. \quad (3.23)$$

It is easy to see that whenever $n = |\mathcal{Y}|$ and π has full rank there is only one possible solution regardless of the choice of W. For overdetermined systems the choice of W may greatly affect the quality of the solution and it is therefore desirable to choose a weighting which minimises the error in estimating μ_{XY}.

In choosing a weighting, we may take advantage of the fact that the errors $\hat{\epsilon}[i]$ are independent for all i. This follows from the fact that all bags are drawn independently of each other. Moreover, we know that $\mathbf{E}[\hat{\epsilon}[i]] = 0$ for all i. Finally we make the assumption that $k(y, y') = \delta(y, y')$, that is, that the kernel in the labels is diagonal. In this situation our analysis is greatly simplified and we have:

$$\epsilon = \sum_y \varphi(y) \otimes p(y)(\pi^\top W \pi)^{-1} \pi W \hat{\epsilon} \quad (3.24)$$

$$\text{and hence } \mathbf{E}\left[\|\epsilon\|^2\right] = \sum_{i=1}^n \sum_y \mathbf{E}\left[\|\hat{\epsilon}[i]\|^2\right] W_{ii}^2 \left[\pi_i^\top (\pi^\top W \pi)^{-1}\right]_y^2 p^2(y) \quad (3.25)$$

Using the assumption that $\mathbf{E}\left[\|\hat{\epsilon}[i]\|^2\right] = O(m_i^{-1})$ we may find a suitable scale of the weight vectors by minimising

$$\sum_{i=1}^n \sum_y \frac{W_{ii}^2}{m_i} \left[\pi_i^\top (\pi^\top W \pi)^{-1}\right]_y^2 p^2(y) \quad (3.26)$$

with respect to the diagonal matrix W. Note that the optimal value of W depends *both* on the mixtures of the bags π_i *and* on the propensity of each class $p(y)$. That is, being able to well estimate a class which hardly occurs at all is of limited value.

3.5.3 Stability Bounds

To complete our reasoning we need to show that our bounds translate into guarantees in terms of the minimiser of the log-posterior. In other words, estimates using the correct mean μ_{XY} vs. its estimate $\hat{\mu}_{XY}$ do not differ by a significant amount. For this purpose we make use of (Altun & Smola, 2006, Lemma 17).

Lemma 25 *Denote by f a convex function on $\mathcal{H}$ and let $\mu, \hat{\mu} \in \mathcal{H}$. Moreover let $\lambda > 0$. Finally denote by $\theta^*, \in \mathcal{H}$ the minimiser of*

$$L(\theta, \mu) := f(\theta) - \langle \mu, \theta \rangle + \lambda \|\theta\|^2 \tag{3.27}$$

with respect to θ and $\hat{\theta}^*$ the minimiser of $L(\hat{\theta}, \hat{\mu})$ respectively. In this case the following inequality holds:

$$\left\|\theta^* - \hat{\theta}^*\right\| \leq \lambda^{-1} \|\mu - \hat{\mu}\| . \tag{3.28}$$

This means that a good estimate for μ immediately translates into a good estimate for the minimiser of the approximate log-posterior. This leads to the following bound on the risk minimiser.

Corollary 26 *The deviation between θ^*, as defined in (3.3) and $\hat{\theta}^*$, the minimiser of the approximate log-posterior using $\hat{\mu}_{XY}$ rather than μ_{XY}, is bounded by $O(m^{-\frac{1}{2}} + \sum_i m_i^{-\frac{1}{2}})$.*

Finally, we may use (Altun & Smola, 2006, Theorem 16) to obtain bounds on the quality of $\hat{\theta}^*$ when considering how well it minimises the *true* negative log-posterior. Using the bound

$$L(\hat{\theta}^*, \mu) - L(\theta^*, \mu) \leq \left\|\hat{\theta}^* - \theta^*\right\| \|\hat{\mu} - \mu\| \tag{3.29}$$

yields the following bound for the log-posterior:

Corollary 27 *The minimiser $\hat{\theta}^*$ of the approximate log-posterior using $\hat{\mu}_{XY}$ rather than μ_{XY} incurs a penalty of at most $\lambda^{-1} \|\hat{\mu}_{XY} - \mu_{XY}\|^2$.*

3.5.4 Stability Bounds under Perturbation

Denote $1 \in \{1\}^{|\mathcal{Y}|}$ as the vector of all ones and $0 \in \{0\}^{|\mathcal{Y}|}$ as the vector of all zeros. Let Δ be the perturbation matrix such that the perturbed mixing matrix $\tilde{\pi}$ is related to the original mixing matrix π by $\tilde{\pi} = \pi + \Delta$. Note that the perturbed mixing matrix $\tilde{\pi}$ still needs to have non-negative entries and each row sums up to 1, $\tilde{\pi}1 = 1$. The stochasticity constraint on the perturbed mixing matrix imposes special structure on the perturbation matrix, i.e. each row of perturbation matrix must sum up to 0, $\Delta 1 = 0$. Let $\hat{\theta}^*$ be the minimiser of (3.5) with mean $\hat{\mu}_{XY}$ approximated via mixing matrix π. Similarly, define $\tilde{\theta}^*$ for $\tilde{\mu}_{XY}$ with mixing matrix $\tilde{\pi}$. We would like to bound the distance $\left\|\hat{\theta}^* - \tilde{\theta}^*\right\|$ between the minimisers. Our perturbation bound relies on Lemma 25 and on the fact that we can bound the errors made in computing an (pseudo-) inverse of a matrix:

Lemma 28 (Stability of Inverses) *For any matrix norm $\|.\|$ and full rank matrices π and $\pi + \Delta$, the error between the inverses of π and $\pi + \Delta$ is bounded by*

$$\left\|\pi^{-1} - (\pi + \Delta)^{-1}\right\| \leq \left\|\pi^{-1}\right\| \left\|(\pi + \Delta)^{-1}\right\| \|\Delta\| . \tag{3.30}$$

Proof We use the following identity $\pi^{-1} - (\pi + \Delta)^{-1} = (\pi + \Delta)^{-1}\Delta\pi^{-1}$. The identity can be shown by left multiplying both sides of equation with $(\pi + \Delta)$. Finally, by submultiplicative property of a matrix norm, the inequality $\|\pi^{-1}\Delta(\pi + \Delta)^{-1}\| \leq \|\pi^{-1}\| \|\Delta\| \|(\pi + \Delta)^{-1}\|$ follows. ■

Theorem 29 (Stability of Pseudo-Inverses: Wedin (1973)) *For any unitarily invariant matrix norm $\|.\|$ and full column rank matrices π and $\pi + \Delta$, the error between the pseudo-inverses of π and $\pi + \Delta$ is bounded by*

$$\left\|\pi^{\dagger} - (\pi + \Delta)^{\dagger}\right\| \leq \mu \left\|\pi^{\dagger}\right\|_{\sigma\infty} \left\|(\pi + \Delta)^{\dagger}\right\|_{\sigma\infty} \|\Delta\| , \tag{3.31}$$

where μ denotes a scalar constant depending on the matrix norm, $\|.\|_{\sigma\infty}$ denotes the spectral norm of a matrix, and the pseudo-inverse $\pi^{\dagger}$ defined as $\pi^{\dagger} := (\pi^{\top}\pi)^{-1}\pi^{\top}$.

Proof See (Wedin, 1973, Theorem 4.1) for a proof. ■

Remark 30 *For full rank matrices, the constant term μ in Theorem 29 is equal to unity regardless of the matrix norm considered (Wedin, 1973).*

First, we would like to bound the difference between $\hat{\mu}_{XY}$ and $\tilde{\mu}_{XY}$, i.e. $\epsilon_p := \hat{\mu}_{XY} - \tilde{\mu}_{XY}$. For the special feature map with full rank, this translates to

$$\epsilon_p = \sum_y p(y) \sum_{i=1}^{n} \varphi(y) \otimes \left[\pi^{-1} - \tilde{\pi}^{-1}\right]_{yi} \mu_X^{\text{set}}[i] \tag{3.32}$$

$$\|\epsilon_p\|^2 = \sum_{i,j} \left\langle \mu_X^{\text{set}}[i], \mu_X^{\text{set}}[j] \right\rangle \times \left[(\pi^{-1} - \tilde{\pi}^{-1})^{\top} K^{y,p} (\pi^{-1} - \tilde{\pi}^{-1})\right]_{ij} . \tag{3.33}$$

Lemma 31 *Define $K^{y,p} := V_{y,p}^{\top} V_{y,p}$. With the spectral norm $\|.\|_{\sigma\infty}$ and a full rank mixing matrix π, the following bound holds:*

$$\|\hat{\mu}_{XY} - \tilde{\mu}_{XY}\|_{\sigma\infty} \leq \|V_{y,p}\|_{\sigma\infty} \left\|\pi^{-1}\right\|_{\sigma\infty} \|\Delta\|_{\sigma\infty} \left\|(\pi + \Delta)^{-1}\right\|_{\sigma\infty} \left[\sum_{i,j} \left\langle \mu_X^{set}[i], \mu_X^{set}[j] \right\rangle\right]^{\frac{1}{2}} . \tag{3.34}$$

Proof We first upper bound $\left[(\pi^{-1}-\tilde{\pi}^{-1})^\top K^{y,p}(\pi^{-1}-\tilde{\pi}^{-1})\right]_{ij}$ by $\left\|(\pi^{-1}-\tilde{\pi}^{-1})^\top K^{y,p}(\pi^{-1}-\tilde{\pi}^{-1})\right\|_{\sigma\infty}$. We factorize $K^{y,p}$ as $V_{y,p}^\top V_{y,p}$ since $K^{y,p}$ is a positive (semi-) definite matrix. The element $K^{y,p}_{y,y'} = k(y,y')p(y)p(y')$ is obtained by multiplying a kernel $k(y,y')$ with a rank-one kernel $k'(y,y') = p(y)p(y')$ where p is a positive function. This conformal transformation preserves the positive (semi-) definiteness of $K^{y,p}$ (Schölkopf & Smola, 2002). Thus, $\left\|(\pi^{-1}-\tilde{\pi}^{-1})^\top K^{y,p}(\pi^{-1}-\tilde{\pi}^{-1})\right\|_{\sigma\infty} \leq \|V_{y,p}(\pi^{-1}-\tilde{\pi}^{-1})\|^2_{\sigma\infty} \leq \left[\|V_{y,p}\|_{\sigma\infty}\|(\pi^{-1}-\tilde{\pi}^{-1})\|_{\sigma\infty}\right]^2 \leq \left[\|V_{y,p}\|_{\sigma\infty}\|\pi^{-1}\|_{\sigma\infty}\|\Delta\|_{\sigma\infty}\|(\pi+\Delta)^{-1}\|_{\sigma\infty}\right]^2$. The last inequality follows directly from Lemma 28. ■

Corollary 32 *Define $K^{y,p} := V_{y,p}^\top V_{y,p}$. With the spectral norm $\|.\|_{\sigma\infty}$ and a full column rank mixing matrix π, the following bound holds:*

$$\|\hat{\mu}_{XY}-\tilde{\mu}_{XY}\|_{\sigma\infty} \leq \sqrt{2}\,\|V_{y,p}\|_{\sigma\infty}\left\|\pi^\dagger\right\|_{\sigma\infty}\|\Delta\|_{\sigma\infty}\left\|(\pi+\Delta)^\dagger\right\|_{\sigma\infty}\left[\sum_{i,j}\left\langle\mu_X^{set}[i],\mu_X^{set}[j]\right\rangle\right]^{\frac{1}{2}}. \tag{3.35}$$

Proof Similar to Lemma 31 with the constant factor μ in Theorem 29 equals to $\sqrt{2}$ for a spectral norm. ■

Combining Lemma 31 for the full rank mixing matrix case (or Corollary 32 for the full column rank mixing matrix case) with Lemma 25, we are ready to state the stability bound under perturbation:

Lemma 33 (Stability Bound under Perturbation) *The distance ϵ_s between the two minimisers, $\hat{\theta}^*$ and $\tilde{\theta}^*$, is bounded by*

$$\epsilon_s \leq \lambda^{-1}\left\|\hat{\mu}_{XY}-\tilde{\mu}_{XY}\right\|. \tag{3.36}$$

It is clear from (3.34) and (3.35) that the stability of our algorithm under perturbation will depend on the size of the perturbation and on the behaviour of the (pseudo-) inverse of the perturbed mixing matrix. Note that by the triangle inequality, the distance in (3.28) can be decomposed as $\left\|\theta^*-\hat{\theta}^*\right\| \leq \left\|\theta^*-\tilde{\theta}^*\right\| + \left\|\tilde{\theta}^*-\hat{\theta}^*\right\|$ and the second term in RHS vanishes whenever the size of perturbation Δ is zero.

3.6 Extensions

3.6.1 Function Spaces

Note that our analysis so far focused on a specific setting, namely maximum-a-posteriori analysis in exponential families. While this is a common and popular setting, the derivations are by no means restricted to this. We have the entire class of (conditional) models described by Altun & Smola (2006); Dudík & Schapire (2006) at our disposition. They are characterised via

$$\underset{p}{\text{minimise}} -H(p) \text{ subject to } \|\mathbf{E}_{z\sim p}\left[\phi(z)\right] - \mu\| \leq \epsilon$$

Here p is a distribution, H is an entropy-like quantity defined on the space of distributions, and $\phi(z)$ is some evaluation map into a Banach space. This means that the optimisation problem can be viewed as an approximate maximum entropy estimation problem, where we do not enforce exact moment matching of μ but rather allow ϵ slack. In both Altun & Smola (2006) and Dudík & Schapire (2006) the emphasis lay on *unconditional* density models: the dual of the above optimisation problem. In particular, it follows that for H being the Shannon-Boltzmann entropy, the dual optimisation problem is the maximum a posteriori estimation problem, which is what we are solving here.

In the conditional case, p denotes the collection of probabilities $p(y|x_i)$ and the operator $\mathbf{E}_{z\sim p}\left[\phi(z)\right] = \frac{1}{m}\sum_{i=1}^{m}\mathbf{E}_{y|p(y|x_i)}\left[\phi(x_i,y)\right]$ is the conditional expectation operator on the set of observations. Finally, $\mu = \frac{1}{m}\sum_{i=1}^{m}\phi(x_i,y_i)$, that is, it describes the empirical observations. We have two design parameters:

Function Space

Depending on which Banach Space norm we may choose to measure the deviation between μ and its expectation with respect to p in terms of e.g. the ℓ_2 norm, the ℓ_1 norm or the ℓ_∞ norm. The latter leads to sparse coding and convex combinations. This means that instead of solving an optimisation problem of the form of (3.5) we would minimise expression of the form

$$\sum_{i=1}^{m} g(\theta|x_i) - m\left\langle \mu_{XY}, \theta\right\rangle + \lambda\left\|\theta\right\|_{\mathcal{B}^*}^{p} \tag{3.37}$$

where $p \geq 1$ and $\mathcal{B}^*$ is the Banach space of the natural parameter θ which is dual to the space $\mathcal{B}$ associated with the evaluation functionals $\phi(x,y)$. The most popular choice for $\mathcal{B}^*$ is ℓ_1 which leads to sparse coding (Candes & Tao, 2005; Chen et al., 1995).

Entropy and Regularity

Depending on the choice of entropy and divergence functionals we obtain a range of diverse estimators. For instance, if we were to choose the *unnormalized* entropy instead of the entropy, we would obtain algorithms more akin to boosting. We may also use Csiszar and Bregmann divergences. The key point is that our reasoning of estimating μ_{XY} based on an aggregate of samples with unknown labels but known label proportions is still applicable.

3.6.2 Unknown test label proportions

In many practical applications we may not actually know the label proportions on the test set. For instance, when deploying the algorithm to assess the spam in a user's mailbox we will *not* know what the fraction would be. Nor is it likely that the user would be willing or able or trustworthy enough to provide a reliable estimate. This means that we need to estimate those proportions in addition to the class means μ_x^{class}.

We may use a fairly straightforward simplification of the covariate shift correction procedure of Huang et al. (2007) in this context. The basic idea is to exploit the fact that there the map $p(x) \to \mu[p(x)] = \mathbf{E}_x[\psi(x)]$ is injective for universal kernels (Section 2.1.4). This means that as long as the conditional distributions $p(x|y)$ are different for different choices of y we will be able to recover the test label proportions by the simple procedure of minimising the distance between $\mu[p]$ and $\sum_y \alpha_y \mu[p(x|y)]$. While we may not have access to the true expectations we are still able to estimate $\mu_x^{\text{class}}[y]$ for all $y \in \mathcal{Y}$. This leads to the optimisation problem

$$\underset{\alpha}{\text{minimise}} \quad \left\| \frac{1}{m} \sum_{i=1}^{m} \psi(x_i) - \sum_{y \in \mathcal{Y}} \alpha_y \mu_X^{\text{class}}[y] \right\|^2 \tag{3.38}$$

$$\text{subject to } \alpha_y \geq 0 \text{ and } \sum_{y \in \mathcal{Y}} \alpha_y = 1. \tag{3.39}$$

Here the sum is taken over the elements of the test set, that is $x_j \in X$. Very similar bounds to those by Huang et al. (2007) can be obtained and they are omitted for the sake of brevity as the reasoning is essentially identical.

Note that obviously (3.38) may be used *separately* from the previous discussion, that is, when the training proportions are known but the test proportions are not. However, we believe that the most significant benefit is obtained in

using both methods in conjunction since many practical situations exhibit both problems simultaneously.

3.7 Related Work and Alternatives

While being highly relevant in practice, the problem has not seen as much attention by researchers as one would expect. Some of the few works which cover a related subject are those by Chen et al. (2006); Musicant et al. (2007), and by Kück & de Freitas (2005). We hope that our work will stimulate research in this area (for example, Rüping (2010)) as relevant problems are fairly widespread.

Transduction

In transduction one attempts to solve a related problem: the patterns x_i on the test set are known, usually also some label proportions on the test set are known but obviously the actual labels on the test set are *not* known. One way of tackling this problem is to perform transduction by enforcing a proportionality constraint on the unlabelled data, e.g. via a Gaussian Process model (Gärtner et al., 2006; Mann & McCallum, 2007).

At first glance these methods might seem applicable for our problem but they do require that we have at least some labelled instances of *all classes* at our disposition which need to be drawn in an unbiased fashion. This is clearly not the case in our setting. That said, it is well possible to use our setting in the context of transduction, that is, to replace the unknown mean μ_{XY}^{test} on the test set by the empirical estimate on the training set. Such strategies lead to satisfactory performance on par with (albeit not exceeding) existing transduction approaches.

Self consistent proportions

Kück & de Freitas (2005) introduced a more informative variant of the binary multiple-instance learning, in which groups of instances are given along with estimates of the fraction of positively-labelled instances per group. The authors build a fully generative model of the process which determines the assignment of observations to individual bags. Such a procedure is likely to perform well when a large number of bags is present.

In order to deal with the estimation of the missing variables a MCMC sampling procedure is used. While Kück & de Freitas (2005) describe the approach only for a binary problem, it could be extended easily to multiclass settings.

In a similar vein, Chen et al. (2006) and Musicant et al. (2007) also use a self-consistent approach where the conditional class estimates need to match the observed ones. Consequently it shares the same similar drawbacks, since we typically only have as many sets as classes.

Conditional Probabilities

A seemingly valid alternative approach is to try building a classifier for $p(i|x)$ and subsequently recalibrating the probabilities to obtain $p(y|x)$, e.g. via $p(y|i)$. At first sight this may appear promising since this method is easily implemented by most discriminative methods. The idea would be to reconstruct $p(y|x)$ by

$$p(y|x) = \sum_i \pi_{iy} p(i|x). \tag{3.40}$$

However, this is not a useful estimator in our setting for a simple reason: it assumes the conditional independence $y \perp\!\!\!\perp x \mid i$, which obviously does not hold. Instead, we have the property that $i \perp\!\!\!\perp x| \, y$, that is, the distribution over x for a given class label does not depend on the bag. This mismatch in the probabilistic model can lead to disastrous estimates as the following simple example illustrates:

Example 1 *Assume that* $\mathcal{X}, \mathcal{Y} = \{1, 2\}$ *and that* $p(y = 1|x = 1) = p(y = 2|x = 2) = 1$. *In other words, the estimation problem is solvable since the classes are well separated. Moreover, assume that* π *is given by*

$$\pi = \begin{bmatrix} 0.5 - \epsilon & 0.5 + \epsilon \\ 0.5 & 0.5 \end{bmatrix} \text{ for } 0 < \epsilon \ll 1.$$

Here, $p(i|x)$ *is useless for estimating* $p(y|x)$, *since we will only exceed random guessing by at most* ϵ. *On the other hand, it is easily possible to obtain a good estimate for* μ_{XY} *by our proposed procedure.*

The reason for this failure can be found in the following expansion

$$p(y|x) = \sum_i p(y|x, i)p(i|x) \neq \sum_i p(y|i)p(i|x) \text{ since } p(y|x, i) \neq p(y|i). \tag{3.41}$$

The problem with (3.41) is that the estimator does not really attempt to compute the probability $p(y|x)$, which we are interested in but instead, it attempts to discern which mixture distribution p_i the observation x most likely originated from. For this to work we would need good probability estimates as the *basis* of reweighting. Our approach tackles the problem at the source by recalibrating the sufficient statistics directly.

Reduction to Binary

For binary classification and real-valued classification scores we may resort to a rather straightforward heuristic: build a classifier which is able to distinguish between the sets X_1 and X_2 and subsequently threshold labels such that the appropriate fraction of observations in X_1 and X_2 matches the proper labels. The intuition is that since the bags X_1 and X_2 do contain some information about how the two classes differ, we should be able to use this information to distinguish between different class labels.

It is likely that one might be able to obtain a proper reduction bound in this context. However, extensions to multi-class are highly nontrivial. It also turns out that even in the binary case this method, while overall fairly competitive, is inferior to our approach.

Density Estimation

One way of obtaining $p(x|i)$ is to carry out density estimation. While, in principle, this approach is flawed because of the incorrect conditional independence assumptions, it can still lead to acceptable results whenever each of the bags contains one majority class. This allows us to obtain

$$p(x|y) = \sum_i \left[\pi^{-1}\right]_{yi} p(x|i) \tag{3.42}$$

To re-calibrate the probability estimates Bayes' theorem is invoked to compute posterior probabilities. Since this approach involves density estimation it tends to fail fairly catastrophically for high-dimensional data due to the curse of dimensionality. These problems are also manifest in the experiments.

3.8 Experiments

Datasets: We use binary and three-class classification datasets from the UCI repository[1] and the LibSVM site.[2] If separate training and test sets are available, we merge them before performing nested 10-fold cross-validation. Since we need to generate as many splits as classes, we limit ourselves to three classes.

For the binary datasets we use half of the data for X_1 and the rest for X_2. We also remove all instances of class 2 from X_1. That is, the conditional class prob-

[1] http://archive.ics.uci.edu/ml/

[2] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/

abilities in X_2 match those from the repository, whereas in X_1 their counterparts are deleted.

For three-class datasets we investigate two different partitions. In scenario A we use class 1 exclusively in X_1, class 2 exclusively in X_2, and a mix of all three classes weighted by $(0.5 \cdot p(1), 0.6 \cdot p(2), 0.7 \cdot p(3))$ to generate X_3. In scenario B we use the following splits

$$\begin{bmatrix} c_1 \cdot 0.4 \cdot p(1) & c_1 \cdot 0.2 \cdot p(2) & c_1 \cdot 0.2 \cdot p(3) \\ c_2 \cdot 0.1 \cdot p(1) & c_2 \cdot 0.2 \cdot p(2) & c_2 \cdot 0.1 \cdot p(3) \\ c_3 \cdot 0.5 \cdot p(1) & c_3 \cdot 0.6 \cdot p(2) & c_3 \cdot 0.7 \cdot p(3) \end{bmatrix}$$

Here the constants c_1, c_2 and c_3 are chosen such that the probabilities are properly normalized. As before, X_3 contains half of the data.

Model Selection: As stated, we carry out a *nested* 10-fold cross-validation procedure: 10-fold cross-validation to assess the performance of the estimators; within each fold, 10-fold cross-validation is performed to find a suitable value for the parameters.

For supervised classification, i.e. discriminative sorting, such a procedure is quite straightforward because we can directly optimize for classification error. For kernel density estimation (KDE), we use the log-likelihood as our criterion.

Due to the high number of hyper-parameters (at least 8) in MCMC, it is difficult to perform *nested* 10-fold cross-validation. Instead, we choose the *best* parameters from a simple 10-fold crossvalidation run. In other words, we are giving the MCMC method an unfair advantage over our approach by reporting the best performance during the model selection procedure.

Finally, for the re-calibrated sufficient statistics $\hat{\mu}_{XY}$ we use the estimate of the log-likelihood on the validation set as the criterion for cross-validation, since no other quantity, such as classification errors is readily available for estimation.

Algorithms: For discriminative sorting we use an SVM with a Gaussian RBF kernel whose width is set to the median distance between observations (Schölkopf, 1997); the regularisation parameter is chosen by cross-validation. The same strategy applies for our algorithm. For KDE, we use Gaussian kernels. Cross-validation is performed over the kernel width. For MCMC, 10 000 samples are generated after a burn-in period of 10 000 steps (Kück & de Freitas (2005)).

Optimisation: Bundle methods (Smola et al., 2007b; Teo et al., 2007) are used to solve the optimisation problem in Algorithm 3. For our regularised log-likelihood, the solver converges to ϵ precision in $O(\log(1/\epsilon))$ steps.

Results: The experimental results are summarised in Table 3. Our method

outperforms KDE and discriminative sorting. In terms of computation, our approach is somewhat more efficient, since it only needs to deal with a smaller sample size (only X rather than the union of all X_i). The training time for our method is less than 2 minutes for all cases, whereas MCMC on average takes 15 minutes and maybe even much longer when the number of active kernels and/or observations are high. Note that KDE fails on two datasets due to numerical problems (high dimensional data).

Our method also performs well on multiclass datasets. As described in Section 3.5.2, the quality of our minimiser of the negative log-posterior depends on the mixing matrix and this is noticeable in the reduction of performance for the dense mixing matrix (scenario B) in comparison to the better conditioned sparse mixing matrix (scenario A). In other words, for ill conditioned π even our method has its limits, simply due to numerical considerations of effective sample size.

Unknown test label proportions: In this experiment, we use binary and three-class classification datasets with the same split procedure as in the previous experiment but we select testing examples by a biased procedure to introduce unknown test label proportions. To describe our biased procedure, consider a random variable ξ_i for each point in the pool of possible testing samples where $\xi_i = 1$ means the i-th sample is being included and $\xi_i = 0$ means the sample is discarded. In our case, the biased procedure only depends on the label y, i.e. $P(\xi = 1|y = 1) = 0.5$ and $P(\xi = 1|y = -1) = 1.0$ for binary problems and $P(\xi = 1|y = 1) = 0.6$, $P(\xi = 1|y = 2) = 0.3$, and $P(\xi = 1|y = 3) = 0.1$ for three-class problems. We then estimate the test proportion by solving the quadratic program in (3.38) with interior point methods (or any other successive optimisation procedure). Since we are interested particularly to assess the effectiveness of our test proportion estimation method, in solving (3.38) we assume that we can compute $\mu_X^{\text{class}}[y]$ directly, i.e. the instances are labelled. The mean square error rates of test proportions for several binary and three-class datasets are presented in Table 4. The results show that our proportion estimation method works reasonably well.

Overdetermined systems: Here we are interested to assess the performance of our estimator with optimized weights when we have more datasets n than class labels $|\mathcal{Y}|$ with varying number of observations m_i per dataset. We simulate the

Table 3. Classification error on UCI/LibSVM datasets

Errors are reported in mean ± standard error. The best result and those not significantly worse than it, are highlighted in boldface. We use a one-sided paired t-test with 95% confidence.

`MM`: Mean Map (our method); `KDE`: Kernel Density Estimation; `DS`: Discriminative Sorting (only applicable for binary classification); `MCMC`: the sampling method; `BA`: Baseline, obtained by predicting the major class. †: Program fails (too high dimensional data - only `KDE`). ‡: Program fails (large datasets - only `MCMC`).

Data	**MM**	**KDE**	**DS**	**MCMC**	**BA**
ionosphere	18.4±3.2	**17.5±3.2**	**12.2±2.6**	18.0±2.1	35.8
iris	**10.0±3.6**	**16.8±3.4**	**15.4±1.1**	**21.1±3.6**	29.9
optdigits	1.8±0.5	**0.7±0.4**	9.8±1.2	2.0±0.4	49.1
pageblock	**3.8±2.3**	7.1±2.8	18.5±5.6	**5.4±2.8**	43.9
pima	**27.5±3.0**	34.8±0.6	34.4±1.7	**23.8±1.8**	34.8
tic	31.0±1.5	34.6±0.5	**26.1±1.5**	31.3±2.5	34.6
yeast	**9.3±1.5**	**6.5±1.3**	25.6±3.6	10.4±1.9	39.9
wine	**7.4±3.0**	**12.1±4.4**	18.8±6.4	**8.7±2.9**	40.3
wdbc	**7.8±1.3**	**5.9±1.2**	10.1±2.1	15.5±1.3	37.2
sonar	**24.2±3.5**	35.2±3.5	31.4±4.0	39.8±2.8	44.5
heart	**30.0±4.0**	38.1±3.8	**28.4±2.8**	**33.7±4.7**	44.9
breastcancer	**5.3±0.8**	14.2±1.6	**3.5±1.3**	**4.8±2.0**	34.5
australian	**17.0±1.7**	33.8±2.5	**15.8±2.9**	30.8±1.8	44.4
svmguide3	**20.4±0.9**	27.2±1.3	25.5±1.5	24.2±0.8	23.7
adult	**18.9±1.2**	24.5±1.3	22.1±1.4	**18.7±1.2**	24.6
cleveland	**19.1±3.6**	35.9±4.5	**23.4±2.9**	**24.3±3.1**	22.7
derm	**4.9±1.4**	27.4±2.6	**4.7±1.9**	14.2±2.8	30.5
musk	**25.1±2.3**	28.7±2.6	**22.2±1.8**	**19.6±2.8**	43.5
german	**32.4±1.8**	41.6±2.9	37.6±1.9	**32.0±0.6**	32.0
covertype	37.1±2.5	41.9±1.7	**32.4±1.8**	41.1±2.2	45.9
splice	**25.2±2.0**	35.5±1.5	**26.6±1.7**	28.8±1.6	48.4
gisette	**10.3±0.9**	†	**12.2±0.8**	50.0±0.0	50.0
madelon	**44.1±1.5**	†	**46.0±2.0**	49.6±0.2	50.0
cmc	**37.5±1.4**	43.8±0.7	45.1±2.3	46.9±2.6	49.9
bupa	**48.5±2.9**	50.8±5.1	**40.3±4.9**	50.4±0.8	49.7
protein A	**43.3±0.4**	48.9±0.9	N/A	65.5±1.7	60.6
protein B	**46.9±0.3**	55.2±1.5	N/A	66.1±2.1	60.6
dna A	**14.8±1.2**	28.1±0.6	N/A	39.8±2.6	41.6
dna B	**31.3±1.3**	**30.4±0.7**	N/A	41.5±0.1	41.6
senseit A	**19.8±0.1**	44.2±0.0	N/A	‡	44.2
senseit B	**21.1±0.1**	44.2±0.0	N/A	‡	44.2

problem in binary settings with the following split ($n = 8$)

$$\begin{bmatrix} c_1 \cdot 0.25 \cdot p(1) & c_1 \cdot 0.10 \cdot p(2) \\ c_2 \cdot 0.15 \cdot p(1) & c_2 \cdot 0.10 \cdot p(2) \\ c_3 \cdot 0.05 \cdot p(1) & c_3 \cdot 0.20 \cdot p(2) \\ c_4 \cdot 0.05 \cdot p(1) & c_4 \cdot 0.10 \cdot p(2) \\ c_5 \cdot 0.05 \cdot p(1) & c_5 \cdot 0.00 \cdot p(2) \\ c_6 \cdot 0.05 \cdot p(1) & c_6 \cdot 0.05 \cdot p(2) \\ c_7 \cdot 0.05 \cdot p(1) & c_7 \cdot 0.15 \cdot p(2) \\ c_8 \cdot 0.35 \cdot p(1) & c_8 \cdot 0.30 \cdot p(2) \end{bmatrix}$$

and the split ($n = 6$) in three-class settings is as follows

$$\begin{bmatrix} c_1 \cdot 0.30 \cdot p(1) & c_1 \cdot 0.10 \cdot p(2) & c_1 \cdot 0.00 \cdot p(3) \\ c_2 \cdot 0.10 \cdot p(1) & c_2 \cdot 0.10 \cdot p(2) & c_2 \cdot 0.20 \cdot p(3) \\ c_3 \cdot 0.05 \cdot p(1) & c_3 \cdot 0.00 \cdot p(2) & c_3 \cdot 0.05 \cdot p(3) \\ c_4 \cdot 0.05 \cdot p(1) & c_4 \cdot 0.20 \cdot p(2) & c_4 \cdot 0.05 \cdot p(3) \\ c_5 \cdot 0.00 \cdot p(1) & c_5 \cdot 0.05 \cdot p(2) & c_5 \cdot 0.10 \cdot p(3) \\ c_6 \cdot 0.50 \cdot p(1) & c_6 \cdot 0.55 \cdot p(2) & c_6 \cdot 0.60 \cdot p(3) \end{bmatrix}$$

We use BFGS to obtain the optimal weights of the minimization problem in (3.26). We perform 10-fold cross validation with respect to the log-likelihood. The error rates are presented in Table 5. For all cases except one, the estimator with optimised weights improves error rates compared with the unweighted one.

Table 4. Unknown test label proportion case

Square errors of estimating the test proportions on UCI/LibSVM datasets. The 10-run errors are reported in mean ± standard error.

Binary datasets

Data	**MSE**
australian	0.00804±0.00275
breastcancer	0.00137±0.00063
adult	0.00610±0.00267
derm	0.00398±0.00175
gisette	0.00331±0.00108
wdbc	0.00319±0.00103

Three-class datasets

Data	**MSE**
protein	0.00290±0.00066
dna	0.00339±0.00075
senseit	0.00072±0.00031

Stability of Mixing Matrices: Lastly, we are interested to assess the performance of our proposed method when the given mixing matrix π are perturbed

Table 5. Overdetermined systems

Errors of weighted/unweighted estimators for overdetermined systems on UCI/LibSVM datasets. The 10-fold cross validation errors are reported in mean ± standard error. The numbers in boldface are significant with 95% confidence (one-sided paired t-test).

Binary datasets

Data	**unweighted**	**weighted**
wdbc	23.29±2.68	**14.22±1.79**
australian	34.44±4.03	29.58±3.71
svmguide3	24.28±2.20	**18.50±1.73**
gisette	8.77±1.05	7.69±0.51
splice	33.43±1.65	**21.12±2.59**

Three-class datasets

Data	**unweighted**	**weighted**
protein	57.46±0.02	57.46±0.02
senseit	28.25±2.60	23.51±0.78
dna	20.01±1.26	**16.80±1.19**

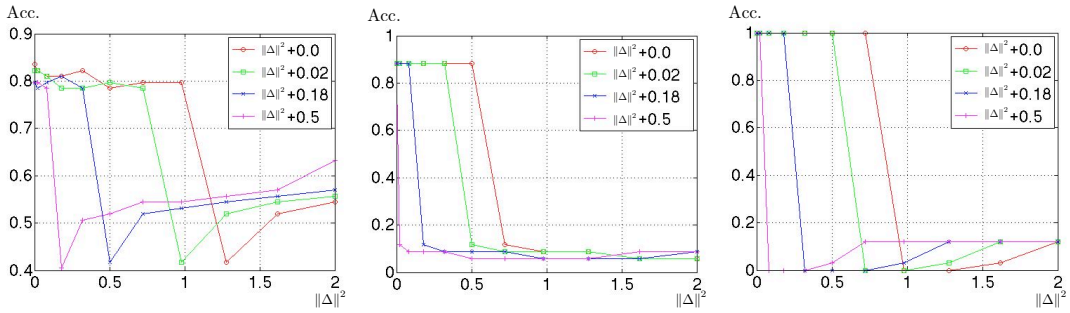
so that they do not exactly match how the data is generated. We used binary classification datasets and defined the perturbed mixing matrix as

$$\tilde{\pi} = \pi + \Delta = \begin{bmatrix} 1 & 0 \\ \rho & 1-\rho \end{bmatrix} + \begin{bmatrix} -\epsilon_1 & \epsilon_1 \\ \epsilon_2 & -\epsilon_2 \end{bmatrix}.$$

We varied $\epsilon_1 \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ and $\epsilon_2 \in \{0.0, 0.1, 0.3, 0.5\}$ and measured the performance as a function of the size of the perturbation, $\eta = \|\Delta\|^2 = tr(\Delta^\top \Delta)$. Note that unperturbed mixing matrix refer to the case of $\{\epsilon_1, \epsilon_2\} = \{0, 0\}$. The experiments are summarised in Figure 3.8. The results suggest that for a reasonable size of perturbations, our method is stable.

3.9 Conclusion

In this chapter we obtained a rather surprising result, namely that it is possible to consistently reconstruct the labels of a dataset if we can only obtain information about the proportions of occurrence of each class (in at least as many data



(a) Adult (b) Australian (c) Breastcancer

Figure 3.2: Performance accuracy of binary classification datasets ($n = |\mathcal{Y}| = 2$) as a function of the amount of perturbation applied to the mixing matrix, $\|\Delta\|^2 = tr(\Delta^\top \Delta)$ with $\Delta = \tilde{\pi} - \pi$. 3.2(a): Adult, 3.2(b): Australian and 3.2(c): Breastcancer datasets. x-axis denotes $\|\Delta\|^2$ as a function of $\epsilon_1 \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. Color coded plots denote $\|\Delta\|^2$ as a function of $\epsilon_2 \in \{0.0, 0.1, 0.3, 0.5\}$, for example red colored plot refers to performance when only label proportions of the first set are perturbed.

aggregates as there are classes). In particular, we proved that up to constants, our algorithm enjoys the same rates of convergence afforded to methods which have full access to all label information.

This finding has significant implications with regard to the amount of privacy afforded by summary statistics. In particular, it implies that whenever accurate summary statistics exist and whenever the available individual statistics are highly dependent on the summarised random variable we will be able to perform inference on the summarised variable with a high degree of confidence. In other words, some techniques used to anonymise observations, e.g. demographic data, may not be really safe (at least when it is possible to estimate the missing information, provided enough data).

Chiaia et al. (2007) applied a summarisation technique to infer drug use based on the concentration of metabolites in the sewage of cities, suburbs or at an even more finely grained resolution. While this only provides aggregate information about the proportions of drug users, such data, in combination with detailed demographic information might be used to perform more detailed inference with regard to the propensity of individuals to use controlled substances. It is in these types of problem where our method could be applied straightforwardly.

Chapter 4

Kernelised Sorting

In this chapter, we introduce a learning setting where we are given a set of data inputs and a set of data outputs, however they are not paired. The goal of learning is to infer the input-output correspondences. This type of learning has applications in areas like data visualisation, photo album summarisation, hybrid keyword-based and content-based search engine, estimation and cross-domain matching, as shown in our experiments.

4.1 Motivating Examples

Matching pairs of objects is a fundamental operation of unsupervised learning. For instance, we might want to match a photo with a textual description of a person, a map with a satellite image, or a music score with a music performance. In those cases it is desirable to have a compatibility function which determines how one set may be translated into the other. For many such instances we may be able to *design* a compatibility score based on prior knowledge or to observe one based on the co-occurrence of such objects. This has led to good progress in areas such as graph matching (Gold & Rangarajan, 1996; Caetano et al., 2007; Cour et al., 2006).

In some cases, however, such a match may not exist or it may not be given to us beforehand. That is, while we may have a good understanding of two sources of observations, say $\mathcal{X}$ and $\mathcal{Y}$, we may not understand the mapping between the two spaces. For instance, we might have two collections of documents purportedly covering the same content, written in two different languages. Here it should be our goal to determine the correspondence between both sets and to identify a mapping between the two domains (Jebara, 2004). In yet other cases, matching

by minimization of a distance function is a popular strategy for point assignment (Caetano et al., 2006; Walder et al., 2006; Steinke et al., 2007).

4.2 Problem Definition

We present a method which is able to perform the above matchings *without* the need of a cross-domain similarity measure and we shall show that if such measures exist it generalises existing approaches.

The basic idea underlying our algorithm is simple. Denote by $X = \{x_1, \ldots, x_m\} \subseteq \mathcal{X}$ the set of inputs and $Y = \{y_1, \ldots, y_m\} \subseteq \mathcal{Y}$ the set of outputs between which we would like to find a correspondence. That is, we would like to find some element π of the permutation group Π_m on m elements

$$\Pi_m := \left\{\pi | \pi \in \{0,1\}^{m \times m} \text{ where } \pi 1_m = 1_m, \pi^\top 1_m = 1_m\right\}$$

such that the set of pairs $Z(\pi) := \left\{(x_i, y_{\pi(i)}) \text{ for } 1 \leq i \leq m\right\}$ corresponds to maximally dependent random variables. We use $\pi(i)$ to denote permutation mapping of i-th element and π to denote permutation matrix whose entries are all 0 except that in row i, the entry $\pi(i)$ equals 1. Here $1_m \in \mathbb{R}^m$ is the vector of all ones. We seek a permutation π such that the mapping $x_i \rightarrow y_{\pi(i)}$ and its converse mapping from y to x are simple.

4.3 The Model

Our method relies on the fact that one may estimate the *dependence* between sets of random variables even without knowing the cross-domain mapping. Various dependence criteria are available. We choose the Hilbert Schmidt Independence Criterion (Section 2.3.2) between two sets and we maximise over the permutation group to find a good match. As a side-effect we obtain an explicit representation of the covariance. We show that our method generalises sorting. When using a different measure of dependence, namely an approximation of the mutual information, our method is related to an algorithm proposed by Jebara (2004).

Formally, for a given measure $D(Z(\pi))$ of the dependence between x and y we define nonparametric sorting of X and Y as follows:

$$\pi^* := \operatorname*{argmax}_{\pi \in \Pi_m} D(Z(\pi)). \tag{4.1}$$

This chapter is concerned with measures of D and approximate algorithms for (4.1). In particular we will investigate the Hilbert Schmidt Independence Criterion (HSIC).

4.3.1 Kernelised Sorting

We use HSIC to *construct* a mapping between X and Y by permuting Y to maximise dependence. There are several advantages in using HSIC as a dependence criterion. First, HSIC satisfies concentration of measure conditions. That is, for random draws of observation from $\Pr_{xy}$, HSIC provides values which are very similar. This is desirable, as we want our mapping to be robust to small changes. Second, HSIC is easy to compute, since only the kernel matrices are required and no density estimation is needed. The freedom of choosing a kernel allows us to incorporate prior knowledge into the dependence estimation process. The consequence is that we are able to generate a family of methods by simply choosing appropriate kernels for X and Y.

Lemma 34 *With $D(Z(\pi))$ as in equation (2.29), the nonparametric sorting problem, called Kernelised Sorting, is given by*

$$\pi^* = \operatorname{argmax}_{\pi \in \Pi_m} \operatorname{tr} \bar{K} \pi^\top \bar{L} \pi. \tag{4.2}$$

Proof We only need to establish that $H\pi^\top = \pi^\top H$ since the rest follows immediately from the definition of (2.29). Since $\pi 1_m = 1_m$ and $\pi^\top 1_m = 1_m$, then $H\pi = (I_m - \frac{1}{m} 1_m 1_m^\top)\pi = (\pi - \frac{1}{m} 1_m 1_m^\top \pi) = (\pi - \frac{1}{m} 1_m 1_m^\top) = (\pi - \frac{1}{m} \pi 1_m 1_m^\top) = \pi(I_m - \frac{1}{m} 1_m 1_m^\top) = \pi H$. Hence H and π matrices commute. ■

Note that the optimisation problem (4.2) is in the form of Koopmans-Beckmann equation (Finke et al., 1987) and is in general NP hard as it is an instance of a quadratic assignment problem (Garey & Johnson, 1979). Nonetheless the objective function is indeed reasonable. We demonstrate this by proving that sorting is a special case of the optimisation problem set out in (4.2). For this we need the following inequality due to Polya, Littlewood, Hardy, and Blackwell (Sherman, 1951):

Lemma 35 *Let $a, b \in \mathbb{R}^m$ where a is sorted ascendingly. If* argsort b *denotes the vector of ranks of ascendingly sorted entries of vector b, then $a^\top \pi b$ is maximised for $\pi =$* argsort b.

Consider the case of scalar random variables and a linear kernel:

Lemma 36 *Let $\mathcal{X} = \mathcal{Y} = \mathbb{R}$ and let $k(x, x') = xx'$ and $l(y, y') = yy'$. Moreover, assume that x is sorted ascendingly. In this case (2.29) is maximised by either $\pi = \text{argsort } y$ or by $\pi = \text{argsort } -y$.*

Proof Under the assumptions we have that $\bar{K} = Hxx^\top H$ and $\bar{L} = Hyy^\top H$. Hence we may rewrite the objective as $\left[(Hx)^\top \pi(Hy)\right]^2$. This is maximised by sorting Hy ascendingly. Since the centring matrix H only changes the offset but not the order this is equivalent to sorting y. We have two alternatives, since the objective function is insensitive to sign reversal of y. ■

This means that sorting is a special case of Kernelised Sorting, hence the name. The ambiguity in the solution of the optimisation problem arises from the fact that instead of having direct access to the entries x_i we only access them by means of the kernel $k(x_i, x_j)$. In this context changes of all observations via $x \leftarrow -x$ leave the kernel unchanged, hence they cannot be detected in the sorting operation. When solving the general problem, it turns out that a projection onto the principal eigenvectors of $\bar{K}$ and $\bar{L}$ is a good initialisation of an optimisation procedure.

4.3.2 Diagonal Dominance

In some cases the biased estimate of HSIC as given by (2.29) leads to rather undesirable results, in particular in the case of document analysis. This arises from the fact that kernel matrices on texts tend to be diagonally dominant: a document tends to be *much* more similar to itself than to others, hence the values of the diagonal entries K_{ii} considerably exceed those of the off-diagonal terms. In this case the $O(1/m)$ bias of (2.29) is significant. After all, it is due to the terms in $\operatorname{tr} HKHL$ which contain matching index pairs $\{ii\}$ with respect to K and L that are responsible for the bias. While their number is only $O(m)$ (the total number of terms is $O(m^2)$), they can still cause considerable damage on finite amounts of data.

Unfortunately, the minimum variance unbiased estimator (Smola et al., 2007a) does not have a computationally appealing form. This can be addressed as follows at the expense of a slightly less efficient estimator with a considerably reduced bias: we replace the expectations (2.28) by sums where no pairwise summation

indices are identical. This leads to the objective function

$$\frac{1}{m(m-1)} \sum_{i \neq j} K_{ij} L_{ij} + \frac{1}{m^2(m-1)^2} \sum_{i \neq j, u \neq v} K_{ij} L_{uv} - \frac{2}{m(m-1)^2} \sum_{i,j \neq i, v \neq i} K_{ij} L_{iv}.$$

This estimator still has a small degree of bias, albeit significantly reduced since it only arises from the product of expectations over (potentially) independent random variables. Using the shorthand $\tilde{K}_{ij} = K_{ij}(1 - \delta_{ij})$ and $\tilde{L}_{ij} = L_{ij}(1 - \delta_{ij})$ for kernel matrices where the main diagonal terms have been removed we arrive at the expression $(m-1)^{-2} \operatorname{tr} H \tilde{K} H \tilde{L}$. The advantage of this term is that it can be used as a drop-in replacement in Lemma 34 without any need for changing the optimisation algorithm.

4.3.3 Stability Analysis

Before discussing practical issues of optimisation let us briefly study the statistical properties of the objective function. First note that the solution $\operatorname{argmax}_\pi \operatorname{tr} \bar{K} \pi^\top \bar{L} \pi$ is *not* stable under sampling in general. A simple example may illustrate this. Assume that $X = \{1, 2, 3\}$ and that $Y = \{1, 2, 2 + \epsilon\}$. In this case the identity permutation $[(1)(2)(3)]$ is sufficient for maximal alignment between X and Y. Now replace the third element in Y, that is $2 + \epsilon$ by $2 - \epsilon$. In this case the permutation $[(1)(2, 3)]$ which swaps the elements 2 and 3 is optimal. Nonetheless, by a suitable choice of ϵ we can make the change in the objective function arbitrarily small.

What we can prove, however, is that changes in the minimum value of the *objective* function are well controlled under the optimisation procedure. This relies on McDiarmid's concentration inequality (McDiarmid, 1989) and on the fact that the minima of close functions are close:

Lemma 37 *Denote by f and g functions on a domain $\mathcal{X}$ with $|f(x) - g(x)| < \epsilon$ for all $x \in \mathcal{X}$. In this case $|\min_{x \in \mathcal{X}} f(x) - \min_{x \in \mathcal{X}} g(x)| < \epsilon$.*

Proof Consider $x^* = \arg\min_{x \in \mathcal{X}} g(x)$, then $|f(x^*) - g(x^*)| < \epsilon$. Since $f(x^*) \geq \min_{x \in \mathcal{X}} f(x)$, then $|\min_{x \in \mathcal{X}} f(x) - g(x^*)| \leq |f(x^*) - g(x^*)| < \epsilon$. ■

Lemma 38 (Concentration Inequality McDiarmid (1989)) *Denote by $f : \mathcal{X}^m \to \mathbb{R}$ a function satisfying*

$$|f(\ldots, x_{i-1}, x, x_{i+1}, \ldots) - f(\ldots, x_{i-1}, x', x_{i+1}, \ldots)| \leq c/m$$

for all $x, x', x_i \in \mathcal{X}$. *Moreover, let* Pr *be a distribution on* $\mathcal{X}$. *In this case for* $X = \{x_1, \ldots, x_m\}$ *drawn from* $\mathcal{P}^m$ *we have that with probability exceeding* $1 - 2\exp\left(-m\epsilon^2/c^2\right)$ *the following bound holds:*

$$|f(X) - \mathbf{E}_{X \sim \Pr^m}[f(X)]| \leq \epsilon. \tag{4.3}$$

Lemma 39 (Stability of optimal alignment) *Denote by* $A(X, Y) := m^{-2} \operatorname{argmin}_{\pi \in \Pi_m} \operatorname{tr} \pi^\top \bar{K} \pi \bar{L}$ *the minimum of the alignment objective function for the sets* X *and* Y. *Moreover, assume that the kernels* k *and* l *are bounded by* $|k(x, x')|, |l(y, y')| \leq R$. *In this case* $|A(X, Y) - \mathbf{E}_{X,Y}[A(X, Y)]| \leq \epsilon$ *holds with probability at least* $1 - 4\exp\left(-m\epsilon^2/8R^2\right)$.

Proof The first step in the proof is to check that if we replace any x_i by some x_i' or alternatively some y_j by y_j' the value of $A(X, Y)$ only changes by $2R/m$. This can be seen by using the fact that HSIC can be seen as the difference between the joint and the marginal expectation of the feature map $k(x, \cdot)l(y, \cdot)$.

Secondly, to deal with the fact that we have expectations over X and Y we apply the concentration inequality twice and chain the arguments. To guarantee a total deviation of at most ϵ we apply a bound of $\epsilon/2$ to the deviation between the empirical average and the expectation $\mathbf{E}_X$, and one more between the expectation $\mathbf{E}_X$ and $\mathbf{E}_{X,Y}$. Applying the union bound for the corresponding probabilities of failure prove the claim. ■

The consequence of this analysis is that while the optimal assignment itself is not stable, at least the objective function has this desirable property, i.e. for random draws of observations from joint distribution, the objective function provides values which are very similar. This means that in practice also most assignments are rather stable when it comes to subsampling. This is evident in the experiments of Section 4.7.2.

4.4 Optimisation

Quadratic assignment problems (Finke et al., 1987) are notoriously hard and have attracted a rather diverse set of algorithms from simulated annealing, tabu search and genetic algorithms to ant colony optimisation. Below we present a rather simple method which is guaranteed to obtain a locally optimal solution by exploiting convexity in the optimisation problem. It is very simple to implement provided that a linear assignment solver is available.

4.4.1 Convex-ConCave Procedure

To find a local maximum of the matching problem we may take recourse to a well-known algorithm, namely the Convex-ConCave Procedure (CCCP) (Section 2.6.1). For the problem in Lemma 34, $h(w) = 0$ and thus CCCP corresponds to a successive maximisation of linear lower bounds.

Lemma 40 *Define π as a doubly stochastic matrix (4.4). The function $\operatorname{tr} \bar{K}\pi^\top \bar{L}\pi$ is convex in π.*

Proof Since $\bar{K}, \bar{L} \succeq 0$ we may factorise them as $\bar{K} = U^\top U$ and $\bar{L} = V^\top V$. Hence by the circularity of the trace we may rewrite the objective function as $\left\|V\pi U^\top\right\|^2$ or as $\|(U \otimes V)\mathrm{vec}(\pi)\|^2$ with vec(.) denotes stacking column vectors of a matrix. This is clearly a convex quadratic function in π. ■

Note that the set of feasible permutations π is constrained in a unimodular fashion, that is, the set

$$P_m := \left\{ \begin{array}{l} M \in \mathbb{R}^{m \times m} \text{ where } M_{ij} \geq 0 \text{ and} \\ \sum_i M_{ij} = 1 \text{ and } \sum_j M_{ij} = 1 \end{array} \right\} \tag{4.4}$$

has only integral vertices, namely admissible permutation matrices. This means that the following procedure will generate a succession of permutation matrices which will yield a local maximum for the assignment problem:

$$\pi_{i+1} \leftarrow (1 - \lambda)\pi_i + \lambda \operatorname*{argmax}_{\pi \in P_m} \left[\operatorname{tr} \bar{K}\pi^\top \bar{L}\pi_i\right] \tag{4.5}$$

Here choosing $\lambda = 1$ in the last step will ensure integrality. The optimisation subproblem is well known as a Linear Assignment Problem and effective solvers are freely available (Jonker & Volgenant, 1987).

Lemma 41 *The algorithm described in (4.5) for $\lambda = 1$ terminates in a finite number of steps.*

Proof We know that the objective function may only increase for each step of (4.5). Moreover, the solution set of the linear assignment problem is finite. Hence the algorithm does not cycle. ■

We refer to Algorithm 4 for a summarisation of the Kernelised Sorting algorithm.

Algorithm 4 Kernelised Sorting

Input Two sets of objects $X = \{x_1, \ldots, x_m\}$ and $Y = \{y_1, \ldots, y_m\}$

Compute kernel similarity matrix K on set X

Compute kernel similarity matrix L on set Y

Center the kernel matrices: $\bar{K} := HKH$ and $\bar{L} := HLH$ with $H_{ij} = \delta_{ij} - m^{-1}$

while not converge **do**

Solve linear assignment problem

$$\pi_{i+1} \leftarrow \operatorname*{argmax}_{\pi \in P_m} \left[\operatorname{tr} \bar{K} \pi^\top \bar{L} \pi_i\right]$$

$$\text{with } P_m := \left\{ \begin{array}{l} \pi \in [0,1]^{m \times m} \text{ where } \pi_{ij} \geq 0 \text{ and} \\ \pi 1_m = 1_m, \pi^\top 1_m = 1_m \end{array} \right\}$$

end while

Return Locally optimum permutation matrix π^*

Non-convex Objective Function

When using the bias corrected version of the objective function the problem is no longer guaranteed to be convex. In this case we need to add a line-search procedure along $\lambda \in [0, 1]$ which maximises

$$\operatorname{tr} H\tilde{K}H[(1-\lambda)\pi_i + \lambda\hat{\pi}_i]^\top H\tilde{L}H[(1-\lambda)\pi_i + \lambda\hat{\pi}_i], \tag{4.6}$$

with $\hat{\pi}_i = \operatorname{argmax}_{\pi \in P_m} \left[\operatorname{tr} \tilde{K} \pi^\top \tilde{L} \pi_i\right]$. Since the function is quadratic in λ we only need to check whether the search direction remains convex in λ; otherwise we may maximise the term by solving a simple linear equation.

Initialisation

Since quadratic assignment problems are in general NP hard we may obviously not hope to achieve an optimal solution. That said, a good initialisation is critical for good estimation performance. This can be achieved by using Lemma 36. That is, if $\bar{K}$ and $\bar{L}$ only had rank 1, the problem could be solved by sorting X and Y in matching fashion. Instead, we use the projections onto the first principal vectors as initialisation in our experiments.

4.4.2 Relaxation to a constrained eigenvalue problem

Yet another alternative is to find an approximate solution of the problem in Lemma 34 by solving

$$\underset{\eta}{\text{maximise}} \ \eta^\top M \eta \text{ subject to } A\eta = b \tag{4.7}$$

Here the matrix $M = \bar{K} \otimes \bar{L} \in \mathbb{R}^{m^2 \times m^2}$ is given by the outer product of the constituting kernel matrices, $\eta \in \mathbb{R}^{m^2}$ is a vectorized version of the permutation matrix π, and the constraints imposed by A and b amount to the polytope constraints imposed by Π_m. This approach has been proposed by Cour et al. (2006) in the context of balanced graph matching.

Note that the optimisation algorithm for (4.7) as proposed by Cour et al. (2006) is suboptimal. Instead, it is preferable to use the exact procedure described in Gander et al. (1989) which is also computationally somewhat more efficient. Nonetheless the problem with the relaxation (4.7) is that it does not scale well to large estimation problems as the size of the optimisation problem scales $O(m^4)$. Moreover, the integrality of the solution cannot be guaranteed: while the constraints are totally unimodular, the objective function is not linear. This problem can be addressed by subsequent projection heuristics. Given the difficulty of the implementation and the fact that it does not even guarantee an improvement over solution at the starting point we did not pursue this approach in our experiments.

4.5 Extensions

4.5.1 Multivariate Dependence Measures

A natural extension is to align several sets of observations. For this purpose we need to introduce a multivariate version of the Hilbert Schmidt Independence Criterion. One way of achieving this goal is to compute the Hilbert Space norm of the difference between the expectation operator for the joint distribution and the expectation operator for the product of the marginal distributions, since this difference only vanishes whenever the joint distribution and the product of the marginals are identical.

Multivariate Mean Operator

Formally, let there be T random variables $x_i \in \mathcal{X}_i$ which are jointly drawn from some distribution $p(x_1, \ldots, x_m)$. Moreover, denote by $k_i : \mathcal{X}_i \times \mathcal{X}_i \to \mathbb{R}$ the corresponding kernels. In this case we can define a kernel on $\mathcal{X}_1 \otimes \ldots \otimes \mathcal{X}_T$ by $k_1 \cdot \ldots k_T$. The expectation operator with respect to the joint distribution and with respect to the product of the marginals is given by Smola et al. (2007a). For instance, the joint expectation operator can be written as follows:

$$\begin{aligned} f(x_1, \ldots, x_T) \to \mathbf{E}_{x_1,\ldots,x_T}\left[f\left(x_1, \ldots, x_T\right)\right] \\ = \mathbf{E}_{x_1,\ldots,x_T}\left[\left\langle f, \prod_{i=1}^{T} k_i(x_i, \cdot)\right\rangle\right] \\ = \left\langle f, \mathbf{E}_{x_1,\ldots,x_T}\left[\prod_{i=1}^{T} k_i(x_i, \cdot)\right]\right\rangle \end{aligned} \tag{4.8}$$

Hence we can express the joint expectation operator and the product of the marginal expectation operators in Hilbert space via

$$\mathbf{E}_{x_1,\ldots,x_T}\left[\prod_{i=1}^{T} k_i(x_i, \cdot)\right] \text{ and } \prod_{i=1}^{T} \mathbf{E}_{x_i}\left[k_i(x_i, \cdot)\right] \tag{4.9}$$

respectively. Straightforward algebra shows that the squared norm of the difference between both terms is given by

$$\begin{aligned} \mathbf{E}_{x_{i=1}^T, x'^T_{i=1}}\left[\prod_{i=1}^{T} k_i(x_i, x_i')\right] + \prod_{i=1}^{T} \mathbf{E}_{x_i, x_i'}[k_i(x_i, x_i')] \\ -2\mathbf{E}_{x_{i=1}^T}\left[\prod_{i=1}^{T} \mathbf{E}_{x_i'}[k(x_i, x_i')]\right]. \end{aligned} \tag{4.10}$$

which we refer to as multiway HSIC. A biased empirical estimate of the above is obtained by replacing sums by empirical averages. Denote by K_i the kernel matrix obtained from the kernel k_i on the set of observations $X_i := \{x_{i1}, \ldots, x_{im}\}$. In this case the empirical estimate of (4.10) is given by

$$\begin{aligned} &\mathrm{HSIC}[X_1, \ldots, X_T] \\ =&1_m^\top \left[\bigodot_{i=1}^{T} K_i\right] 1_m + \prod_{i=1}^{T} 1_m^\top K_i 1_m - 2 \cdot 1_m^\top \left[\bigodot_{i=1}^{T} K_i 1_m\right] \end{aligned} \tag{4.11}$$

where $\odot_{t=1}^{T} *$ denotes elementwise product of its arguments (the '.*' notation of Matlab).

Optimisation

To apply this new criterion to sorting we only need to define T permutation matrices $\pi_i \in \Pi_m$ and replace the kernel matrices K_i by $\pi_i^\top K_i \pi_i$.

Without loss of generality we may set $\pi_1 = \mathbf{1}$, since we always have the freedom to fix the order of one of the T sets with respect to which the other sets are to be ordered. In terms of optimisation the same considerations as presented in Section 4.4 apply. That is, the objective function is convex in the permutation matrices π_i and we may apply the CCCP to find a locally optimal solution.

4.5.2 Semi-Supervised Kernelised Sorting

Kernelised Sorting aims to find a correspondence between two sets of objects from different domains which only requires a similarity measure within each of the two domains. Other than the within-domain similarities, no other information is provided to guide the correspondence. In other words, Kernelised Sorting can be viewed as an unsupervised technique. However, for some applications such as search engines, it might be beneficial to introduce a small amount of supervision to guide or adjust the correspondence.

Assume now that we wish to enforce a specific preference preference on a subset of objects, $\mathcal{P} = \{(x_i, y_j) \text{ for } (i,j) \subset \{1,\dots,m\} \times \{1,\dots,m\}\}$. To solve this, additional constraints associated with the preference are added to the original optimisation problem in (4.2) as follows:

$$\begin{aligned} \pi^* &= \operatorname{argmax}_\pi \operatorname{tr} \bar{K} \pi^\top \bar{L} \pi \\ \text{s.t.} \quad \pi_{ij} &= 1 \quad \forall (i,j) \in \mathcal{P} \end{aligned} \tag{4.12}$$

However, the above method is sub-optimal whenever $|\mathcal{P}| << m$ in the sense that neighbouring objects of the constraints are not mapped in the proximal locations of the constraints. As it is considerably cheaper to simply satisfy the constraints independently and to enforce smoothness on the rest of the objects as if without the constraints. This sub-optimality can be addressed by re-weighting the preference constraint with the within-domain similarities, as follows:

$$\begin{aligned} \pi^* &= \operatorname{argmax}_\pi \operatorname{tr} \bar{K} \pi^\top \bar{L} \pi \\ \text{s.t.} \quad \sum_{k,l} \pi_{k,l} K_{k,i} W_{i,j} L_{j,l} &\leq \sum_{i,j} K_{m,i} W_{i,j} L_{j,n} + C \quad \forall (m,n) \in \mathcal{P}, \end{aligned} \tag{4.13}$$

for appropriately chosen constant C and $W \in \{0,1\}^{m \times m}$ with $w_{ij} = 1 \; \forall (i,j) \in \mathcal{P}$, otherwise $w_{ij} = 0$. The partial Lagrangian formulation will look like

$$\pi^* = \operatorname{argmin}_\pi - \operatorname{tr} \bar{K} \pi^\top \bar{L} \pi - \sum_{z=(m,n) \in \mathcal{P}} \alpha_z \left(\sum_{k,l} \pi_{k,l} K_{k,i} W_{i,j} L_{j,l} - \sum_{i,j} K_{m,i} W_{i,j} L_{j,n} - C \right) \tag{4.14}$$

Here $\alpha_z \geq 0$ are nonnegative constants which act as Lagrange multipliers to ensure all re-weighted preference constraints are met. Exploiting the fact that each of our preference constraints is differ only by a constant, we can turn these multiple in-equality constraints into a single constraint as follows:

$$\pi^* = \operatorname{argmin}_\pi - \operatorname{tr} \bar{K} \pi^\top \bar{L} \pi - \alpha \left(\operatorname{tr} \pi^T K W L \right) \tag{4.15}$$

where $\alpha \geq 0$ is the Lagrange multiplier. The above problem corresponds to adding a linear mixing matrix to the original Kernelised Sorting objective function in (4.2) thus the modified objective function is still convex in π. This means that the optimisation problem in (4.15) is still amenable to the CCCP with succession of linear assignment solvers.

4.6 Related Work

Matching and layout are clearly problems that have attracted a large degree of prior work. We now discuss a number of algorithms which are related to or special cases of what we proposed by means of Kernelised Sorting.

4.6.1 Mutual Information

Probably the most closely related work is that of Jebara (2004), who aligns bags of observations by sorting via minimum volume PCA. Here, we show that when using mutual information, our scheme leads to a criterion very similar to the one proposed by Jebara (2004). Mutual information, defined as $I(X,Y) = h(X) + h(Y) - h(X,Y)$, is a natural means of studying the dependence between random variables x_i and $y_{\pi(i)}$. In general, this is difficult, since it requires density estimation. However, this can be circumvented via an effective approximation, where instead of maximizing the mutual information directly, we maximise a lower bound to the mutual information. First, we note that only the last term matters since the first two are independent of π. Maximizing a lower bound

on the mutual information then corresponds to minimizing an upper bound on the joint entropy $h(X, Y)$. An upperbound for the entropy of any distribution with variance Σ is given by the differential entropy of a normal distribution with covariance Σ, which can be computed as

$$h(p) = \frac{1}{2} \log |\Sigma| + \text{constant}. \tag{4.16}$$

Hence the problem reduces to minimizing the joint entropy $J(\pi) := h(X, Y)$, where x and y are assumed jointly normal in the Reproducing Kernel Hilbert Spaces spanned by the kernels k, l and $k \cdot l$. By defining a joint kernel on $\mathcal{X} \times \mathcal{Y}$ via $k((x, y), (x', y')) = k(x, x')l(y, y')$ we arrive at the optimisation problem

$$\operatorname*{argmin}_{\pi \in \Pi_m} \log |HJ(\pi)H| \text{ where } J_{ij} = K_{ij} L_{\pi(i),\pi(j)}. \tag{4.17}$$

Note that this is *related* to the optimisation criterion proposed by Jebara (2004) in the context of sorting via minimum volume PCA. What we have obtained here is an alternative derivation of Jebara (2004)'s criterion based on information theoretic considerations.

The main difference with our work is that Jebara (2004) uses the setting to align a large number of bags of observations by optimizing $\log |HJ(\pi)H|$ with respect to re-ordering within each of the bags. Obviously (4.17) can be extended to multiple random variables, simply by taking the pointwise product of a sequence of kernel matrices. In terms of computation (4.17) is considerably more expensive to optimise than (4.2) since it requires computation of inverses of matrices even for gradient computations.

4.6.2 Object Layout

A more direct connection exists between object layout algorithms and Kernelised Sorting. Assume that we would like to position m objects on the vertices of a graph, such as a layout grid for photographs with the desire to ensure that related objects can be found in close proximity. We will now show that this is equivalent to Kernelised Sorting between a kernel on objects and the normalised graph Laplacian induced by the graph.

To establish our claim we need some additional notation. Denote by $G(V, E)$ an undirected graph with a set of vertices V and edges E. With some abuse of notation we will denote by G also the symmetric edge adjacency matrix. That is, $G_{ij} = 1$ if there is an edge between vertex i and j and $G_{ij} = 0$ if no edge is

present. This definition naturally extends to weighted graphs simply by allowing that $G_{ij} \geq 0$ rather than $G_{ij} \in \{0, 1\}$. Moreover, we denote by $d_i := \sum_j G_{ij}$ the degree of vertex i in the graph and we let $D := \operatorname{diag}(d)$ be a diagonal matrix containing the degrees. Finally we denote by

$$L := D - G \tag{4.18}$$

the graph Laplacian L.

It is well known, see e.g. (Fiedler, 1973; Chung-Graham, 1997), that local smoothness functionals on graphs can be expressed in terms of L. More specifically we have

$$\sum_{i,j} G_{ij} \left\|\phi(x_i) - \phi(x_j)\right\|^2 = \operatorname{tr} KL \tag{4.19}$$

where $\phi(x_i)$ can be treated as the vertex value and $K_{ij} = k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$. Basically, expression (4.19) sums over the squared differences between the values of adjacent vertices. The smaller the number $\operatorname{tr} KL$, the smoother the vertex values vary across the graph. By construction, (4.19) is translation invariant, that is, changes from $\phi(x_i) \leftarrow \phi(x_i) - \mu$ leave the functional unchanged. Hence we have $\operatorname{tr} KL = \operatorname{tr} HKHL$.

If we were to layout objects such that similar objects are assigned to adjacent vertices in G, we can maximise the smoothness by minimizing $\operatorname{tr} HKH\pi^\top L\pi$. Here the main difference to (4.2) is that we are *minimizing* a convex form rather than maximizing it.

Such difference can be removed by a simple substitution of L by $\|L\| I - L$. Indeed, note that the eigenvalues of L range between 0 and $\|L\|$. The transformation $\|L\| I - L$ shifts the eigenvalues into positive territory while changing the objective function only by a constant independent of π, thus leading to a Kernelised Sorting problem for the "kernel" $L' = \|L\| I - L$. This is also consistent with the definition of a kernel which is the *inverse* of a regularisation operator (Girosi, 1997; Smola et al., 1998). That is, while in a regularisation operator large eigenvalues correspond to properties of a function which are undesirable, the converse is true in a kernel, where large eigenvalues correspond to simple functions (Schölkopf et al., 1998).

A consequence of these considerations is that for object layout there exists an alternative strategy for optimisation: first relax the set of permutation matrices Π_m into the set of doubly stochastic matrices P_m and solve the relaxed problem $\min_{\pi \in P_m} \operatorname{tr} HKH\pi^\top L\pi$ exactly; and then employ the CCCP procedure described

in section 4.4.1 to find a locally optimal integral solution. While theoretically appealing, this approach nonetheless suffers from a range of problems: the number of variables required to deal with in the quadratic program is $O(m^2)$ which makes an efficient implementation a challenge even for modest amounts of data, unless the special structure of the quadratic form in π is exploited.

4.6.3 Morphing

In object morphing one may use a compatibility function defined on local similarity between source and destination matches. Assume that $X, Y \in \mathbb{R}$ are sets of scalars (e.g. intensity values in an image). In this context Walder et al. (2006); Steinke et al. (2007) use scoring functions of the form

$$\frac{1}{2}\sum_{i=1}^{m}\left(x_i - y_{\pi(i)}\right)^2 = \frac{1}{2}\sum_i x_i^2 + y_{\pi(i)}^2 - \sum_i x_i y_{\pi(i)}. \tag{4.20}$$

Whenever π is a bijection[1] the first two terms are independent of π and the problem of matching becomes one of maximizing $\sum_i x_i y_{\pi(i)}$, ie. $X^\top \pi Y$. By the same argument as in the proof of Lemma 36 this can be rewritten in the form of

$$\operatorname*{argmax}_{\pi \in \Pi_m} \operatorname{tr} X X^\top \pi Y Y^\top \pi^\top \tag{4.21}$$

simply by squaring the objective function of $X^\top \pi Y$. The only ambiguity left is that of an arbitrary sign, i.e. we might end up *minimizing* the match between X and Y rather than maximizing it. That said, our argument shows that morphing and Kernelised Sorting have closely related objective functions.

4.6.4 Smooth Collages

The generation of collages is a popular application in the processing of composite images. In this process one uses a template image Y (often a company logo or a face of a person) and a collection $X = \{x_1, \ldots, x_m\}$ of reference images to generate a collage where the individual "pixels" of the collage are taken from the set of reference images such that the collage best resembles the template. This problem is easily solved by a linear assignment algorithm as follows:

Denote by $d(x, y)$ a distance function between an image x and a pixel y in the template. Moreover, denote by y_i a pixel in Y. In this case the optimal

[1] Note that this is *not* required by Walder et al. (2006); Steinke et al. (2007). In fact, their objective function is not even symmetric between source and destination images.

assignment of reference images to Y is achieved by finding the permutation π which minimises

$$\sum_i d(x_i, y_{\pi(i)}) = \operatorname{tr} \pi^\top D \text{ where } D_{ij} := d(x_i, y_j). \tag{4.22}$$

In other words, one attempts to find an overall allocation of reference images to the template such that the sum of distances is minimised. While this is desirable in itself, it would also be best if there were some spatial coherence between images. This is achieved by mixing the objective function of (4.22) with the Kernelised Sorting objective. Since this constitutes only a linear offset of the optimisation problem of (4.2) it can be solved in an identical way to what is required in Kernelised Sorting, namely by a CCCP procedure.

4.7 Applications

To investigate the performance of our algorithm (it is a fairly nonstandard unsupervised method) we applied it to a variety of different problems ranging from visualisation to matching and estimation.

In all our experiments, the maximum number of iterations used in the updates of π is 100 and we terminate early if progress is less than 0.001% of the objective function.

4.7.1 Data Visualisation

In many cases we may want to visualise data according to the metric structure inherent in it. In particular, we may want to align it according to a given template, such as a grid, a torus, or any other *fixed* structure. Such problems occur when presenting images or documents to a user.

While there is a large number of algorithms for low dimensional object layout (Maximum Variance Unfolding (MVU) (Weinberger & Saul, 2006), Local-Linear Embedding (LLE) (Roweis & Saul, 2000), ...), most of them suffer from the problem that the low dimensional presentation is nonuniform. This has the advantage of revealing cluster structure but given limited screen size the presentation is undesirable.

Alternatively, one can use the Self-Organizing Map (SOM) (Kohonen, 1982) or the Generative Topographic Mapping (GTM) (Bishop et al., 1998) to layout images according to a pre-defined grid structure. These methods, however, often

map several images into a single grid element, and hence some grid elements may have no data associated with them. Such grouping creates blank spaces in the layout and still under-utilises the screen space.

Instead, we may use Kernelised Sorting to layout objects. Here the kernel matrix K is given by the similarity measure between the objects x_i that are to be laid out. The kernel L, on the other hand, denotes the similarity between the locations of grid elements where objects are to be aligned to.

Image Layout on a Uniform Grid

For the first visualisation experiment, we want to layout images on a 2D rectangular grid. We have obtained 320 images from Flickr[1] which are resized and downsampled to 40×40 pixels. We convert the images from RGB into Lab space, yielding $40 \times 40 \times 3$ dimensional objects. The grid, corresponding to Y is a 16×20 mesh on which the images are to be laid out. We use a Gaussian RBF kernel between the objects to be laid out and also between the positions of the grid, i.e. $k(x, x') = \exp(-\gamma \|x - x'\|^2)$. The kernel width γ is adjusted to the inverse median of $\|x - x'\|^2$ such that the argument of the exponential is $O(1)$ (refer to Yamada & Sugiyama (2011) for a recent work on how to adjust the kernel width based on a cross validation principle). After sorting we display the images according to their matching coordinates. The result is shown in Figure 4.1(a). Clearly, images with similar colour composition are found at proximal locations.

For comparison, we apply an SOM[2] and a GTM[3] to the same data set. The results are shown in Figure 4.2(a) and 4.2(b). If a grid element (corresponding to a neuron and a latent variable) has been assigned multiple images, only one of the assigned images is displayed. The detail of all other overlapping images can be found in Figure 4.7.1 and Figure 4.4.

Image Layout on an Irregular Grid

To reinforce the point that matching can occur between arbitrary pairs of objects we demonstrate that images can be aligned with the letters 'PAMI 2009' displayed as a pixelated grid on which the images are to be laid out. The same colour

[1] http://www.flickr.com

[2] http://www.cis.hut.fi/projects/somtoolbox/. A Gaussian neighbourhood and inverse learning rate functions are used.

[3] http://www.ncrg.aston.ac.uk/GTM/, again forcing the images into a 2D grid. The principal components are used for the initialisation and the mode projection is used to map data into the (2D grid) latent space.

features and the same Gaussian RBF kernels as in the previous experiment are used. The result is presented in Figure 4.1(b). As expected, the layout achieves a dual goal: it fully utilises the elements on the irregular grid while at the same time preserving the colour grading.

Image Layout on Hierarchical Structures

It is quite straightforward to extend image layouting on a 2D grid to a hierarchy of 2D grids. Here one additional axis can be used to specify the hierarchy level. Instead of (x, y) position, now a point is identified by its (x, y, z) coordinates in a three dimensional coordinate system. The similarity measure on the structure will then be either the similarity measure between points within the same hierarchy level or between points across different hierarchy levels. The z axis plays an important role on how spatial coherence in one hierarchy level is propagated to subsequent hierarchy levels. The higher its value, the more *independently* the organization of images on one level is done with respect to other levels. Equivalent to a 2D grid, the inverse of the exponentiated straight line distance is used to measure the similarity between two points on the hierarchy where the distance is now defined on the three coordinate axis. This concept is also easily extended to a hierarchy of 2D spheres where the straight line distance is now replaced with the great circle distance (a special case of the geodesic distance on a sphere manifold). The results are shown in Figure 4.6 and Figure 4.7 for a hierarchy of 2D grids and 2D spheres, respectively.

Visualisation of Semantic Structure

While colour based image layout gives visually pleasing results, one might desire to layout images based on their semantic content and explore the high dimensional semantic space inherent in images by providing a two dimensional layout. To this end, we represent images as bag-of-visual-words (Csurka et al., 2004), i.e. histograms of vector quantized local image descriptors. This representation has been shown successful in the context of visual object recognition. Here we use a combination of densely sampled, overlapping patches with the SIFT descriptor (Lowe, 2004). Then the inverse of the exponentiated χ^2 distance, denoted as $\exp(-\gamma \left\|x - x'\right\|_\chi^2)$, is used to measure the similarity between the images. Gaussian RBF kernel is still used to measure similarity between the positions of the grid. We apply this scheme to 570 images from the MSRC2 database.[1] The result

[1] http://research.microsoft.com/vision/cambridge/recognition/

(a) Layout of 320 images into a 2D grid of size 16 by 20 using Kernelised Sorting

(b) Layout of 280 images into a 'PAMI 2009' letter grid using Kernelised Sorting

Figure 4.1: Image layouting on a 2D grid and letter grid with Kernelised Sorting. One can see that images are laid out in the grids according to their colour grading.

(a) Layout of 320 images into a 2D grid of size 16 by 20 using SOM

(b) Layout of 320 images into a 2D grid of size 16 by 20 using GTM

Figure 4.2: Comparison with SOM and GTM for image layout on a 2D grid and a compressed representation of images. Note that both algorithms do not guarantee unique assignments of images to nodes.

Figure 4.3: Image matching as obtained by Kernelised Sorting. The images are cut vertically into two equal halves and Kernelised Sorting is used to pair up image halves that originate from the same images.

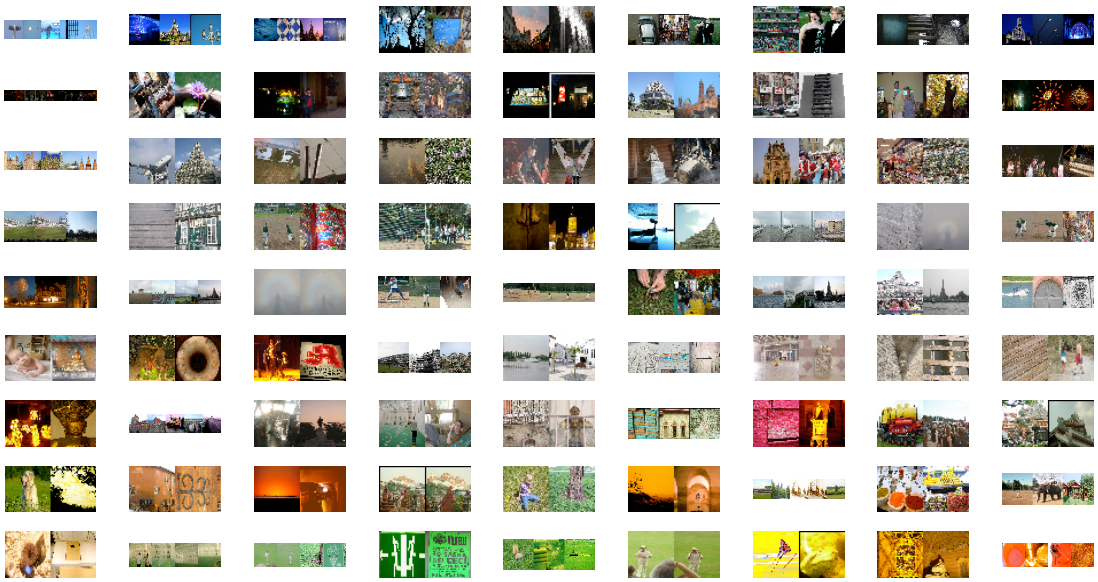


Figure 4.4: 81 out of the 320 neurons in SOM are assigned more than one image, effectively clustering the images into 81 groups. We show the cluster membership of each group.

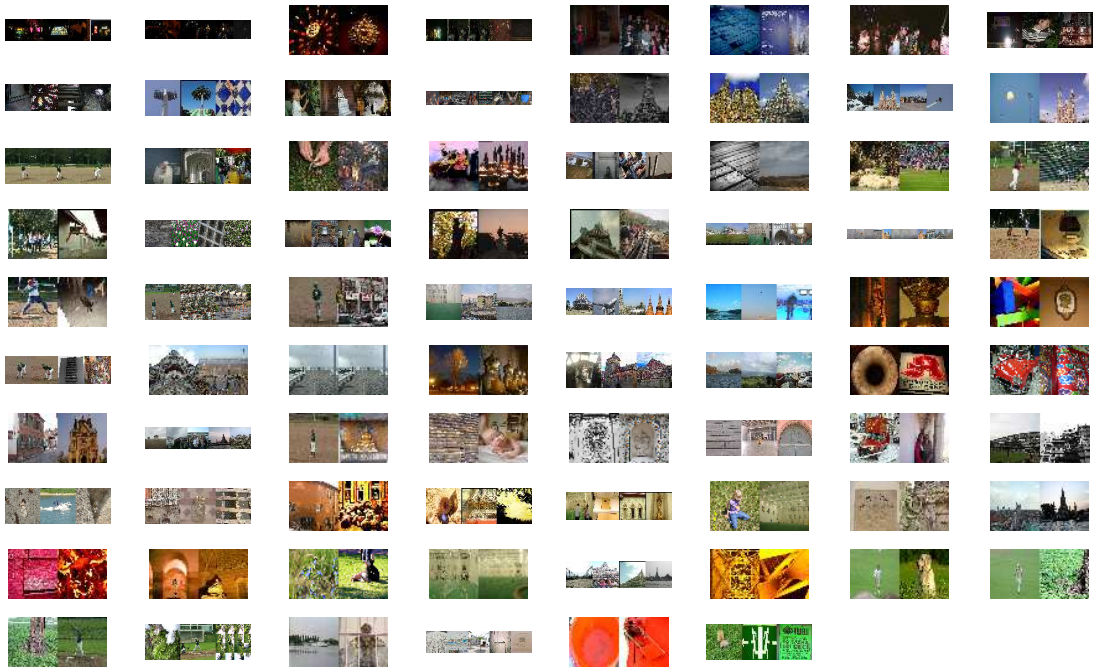


Figure 4.5: 78 out of the 320 latent variables in GTM are assigned more than one image. This effectively cluster the images into 78 groups. This figure shows the cluster membership of each group.

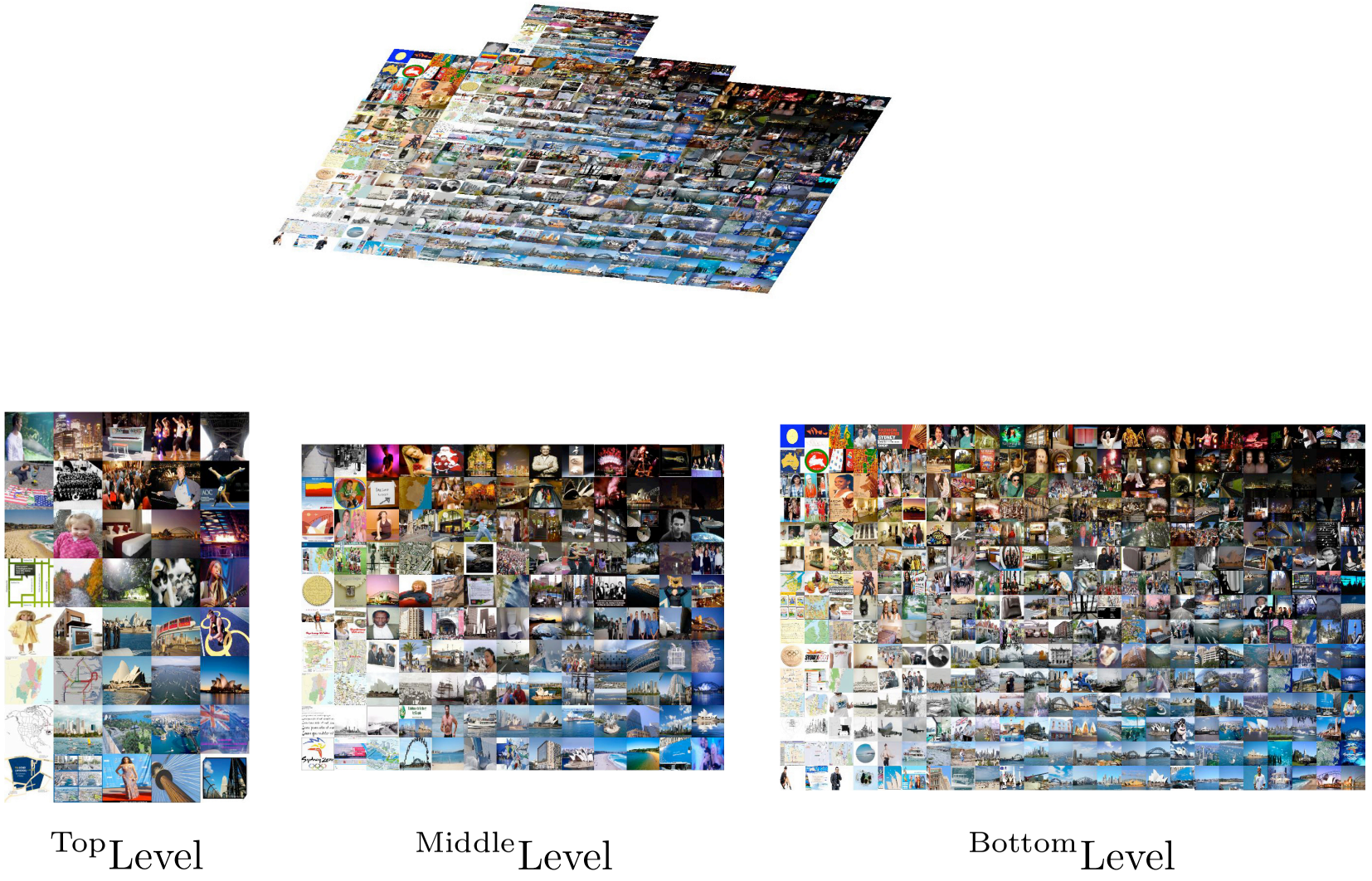


Figure 4.6: Image Layouting on a Hierarchical structure of 2D grids.

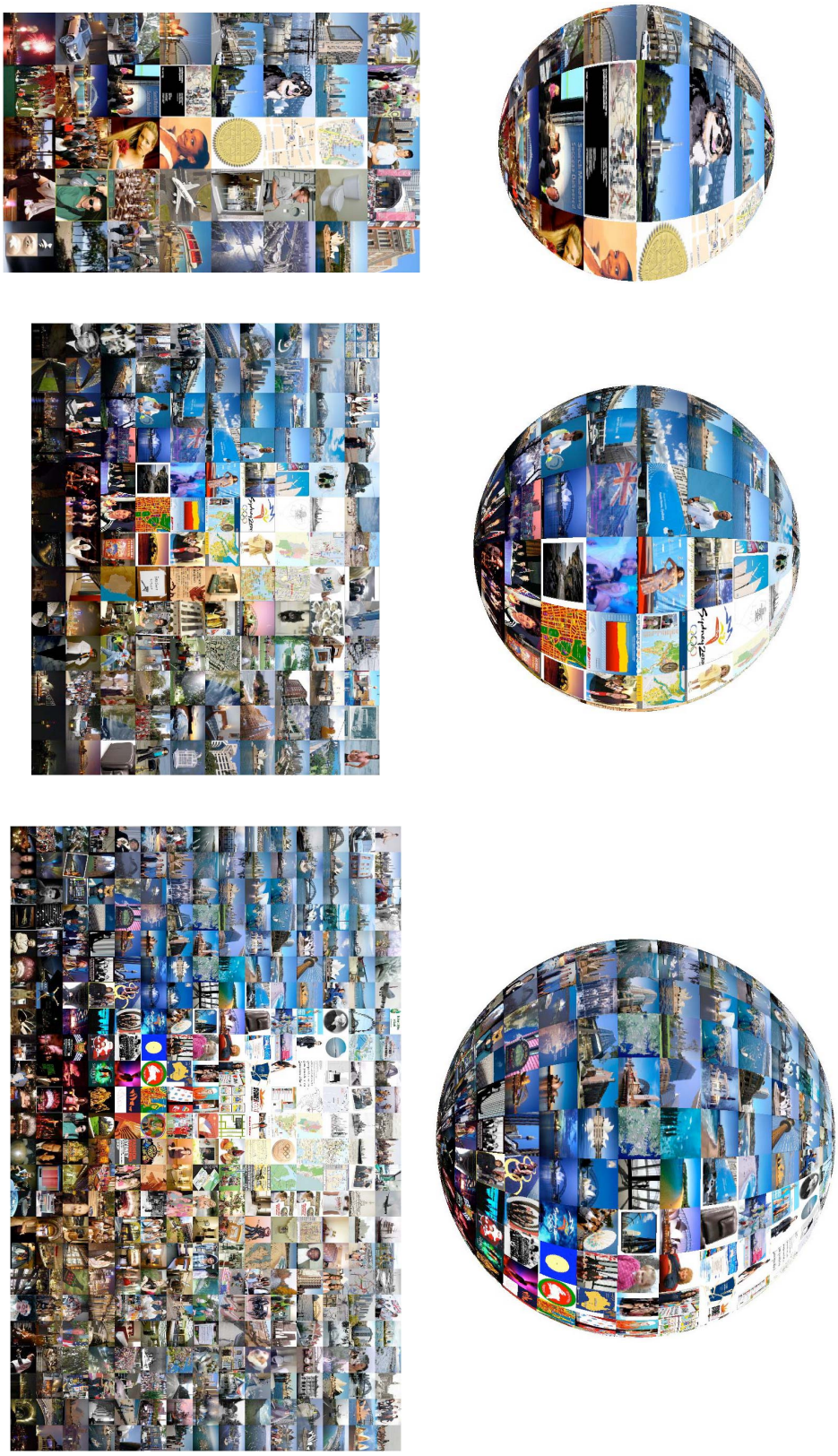


Figure 4.7: Image Layouting on a Hierarchical structure of 2D spheres.

is presented in Figure 4.8.

First, one can observe that objects are grouped according to their categories. For example, books, cars, planes, and people have all or most of their instances visualised in proximal locations. Second, beyond categories, another ordering based on the overall composition of the images is also visible. Images near the lower left corner consist mostly of rectangular shaped objects; along the antidiagonal direction of the layout, the shapes of the objects become more and more irregular. This reveals structure of the metric space which has not been explicitly designed.

Another example of visualisation of semantic similarities by Kernelised Sorting is a recent work of (Torralba et al., 2010, Figure 6) which uses Kernelised Sorting on a set of 12201 images.

Photo Album Summarisation

An immediately useful application of Kernelised Sorting is a tool for presenting a summary of personal photo collections. This is particularly challenging when photos are taken by different persons, with different scenery, with different cameras or over a large time period.

Depending on the way a viewer wants to explore the photo album, the photos can be summarised either based on colour information or on a bag-of-visual-words based image representation. Figure 4.9 shows the corresponding summaries for a collection of holiday photos of my supervisor, Alex Smola[1]. Comparing the two summaries, we can see that the latter presents a much clearer separation between natural scenery and human subjects.

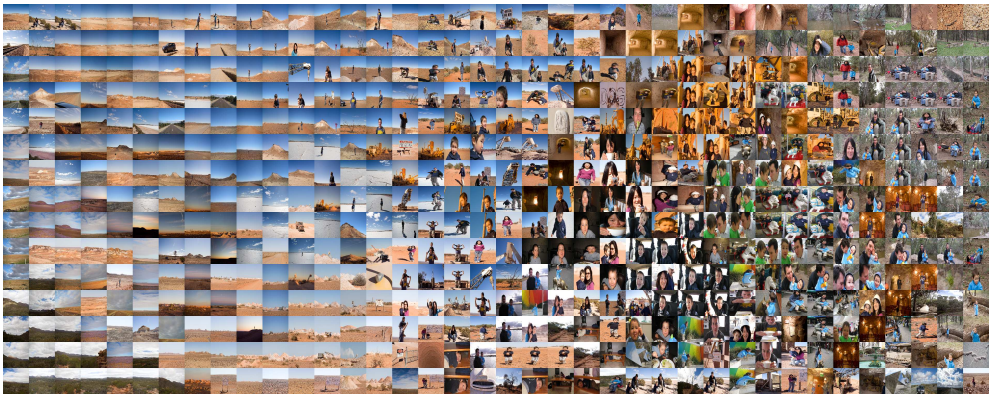
Hybrid Keyword-Based and Content-Based Search Engine

Another application of Kernelised Sorting is to build an image search engine that combines the advantages of keyword-based and content-based search engines. All *commercially* available image search engines (such as Google, Yahoo!, and Bing, among others) are keyword-based. Although this has proven its value in common web search engines, it is often perceived as being a limitation for presenting image search results. It is a typical situation that these keyword-based search engines possibly return search results that are not even related to the query. This situation is caused by the usage of keywords – the filename of the image

[1] Alex has kindly agreed to share his holiday photos with me and the rest of the world for the purpose of this chapter.

Figure 4.8: Layout of 570 images into a 2D grid of size 15 by 38 using bag-of-visual-words based Kernelised Sorting. Several object categories, like books, cars, planes, and people are grouped into proximal locations.

(a) Photos summarisation by colour based Kernelised Sorting.



(b) Photos summarisation by bag-of-visual-words based Kernelised Sorting.

Figure 4.9: Application of Kernelised Sorting as a photo collection summarisation tool.

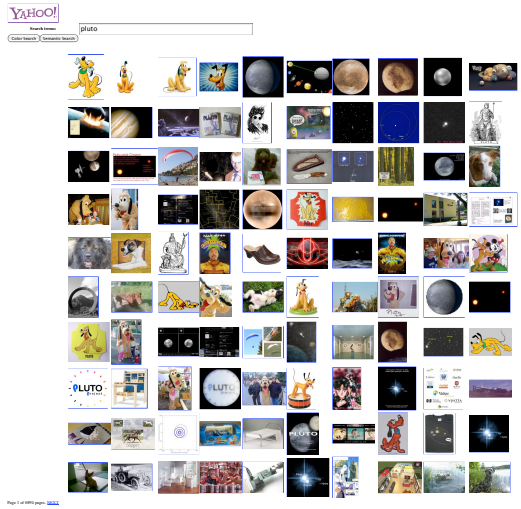
and descriptions associated with the image – to represent visual characteristics of an image, rather than the actual image content. It can happen that visually different images have the same keywords while visually similar images have totally different keywords. In contrast, a content-based search engine (such as Cortina[1]) analyses the actual contents of the image such as colours, shapes, textures, or any other information that can be derived from the image itself to determine the returned results. However, the performance of content-based search engines is still far from being ready to be deployed as real-world commercial image retrieval engines. One of the identified problems with the current content-based approaches is the reliance on visual similarity for judging semantic similarity, which may be problematic due to the semantic gap between low-level content and higher-level concepts.

We propose an image search interface, called Globby (available at http://globby.iais.fraunhofer.de), which bridges the gap between keyword-based and content-based search by returning *keyword*-based query relevant objects in a set of pages, where each page contains several objects with similar *content* objects are placed at proximal locations. This content similarity visualisation is achieved via Kernelised Sorting. Further, in Globby, the highly keyword-based query relevant image is placed at a specific location, for example at the top-left corner. This can be achieved via a semi-supervised extension of Kernelised Sorting (Section 4.5.2) where the preference constraint is now a ranking constraint. Figure 4.10 shows the comparison between a keyword-based search engine (Yahoo!) and Globby when colour and SIFT descriptors are used to represent the content of the images.

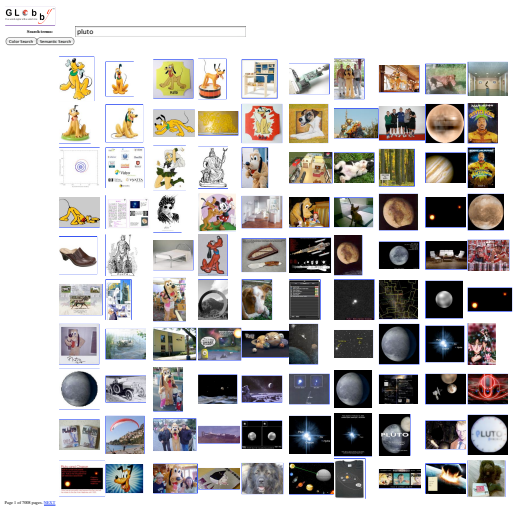
4.7.2 Matching

Apart from visualisation, Kernelised Sorting can also be used to align or match two related data sets even without cross data set comparison. In the following set of experiments, we will use data sets with known ground truth of matching. This allows us to quantitatively evaluate Kernelised Sorting. To create such data sets, we either split an image or a vector of data attributes into two halves, or use multilingual documents that are translations of each other. A recent work of Tripathi et al. (2011) uses the concept of matching without cross-similarities in Bioinformatics application such as to perform a joint analysis of mRNA and protein concentrations without the mapping between genes and proteins.

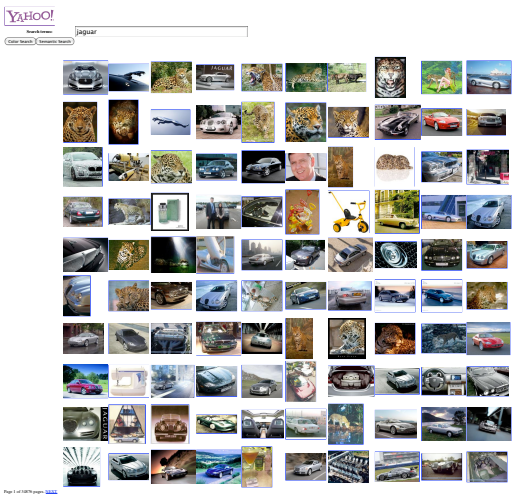
[1] http://vision.ece.ucsb.edu/multimedia/cortina.shtml



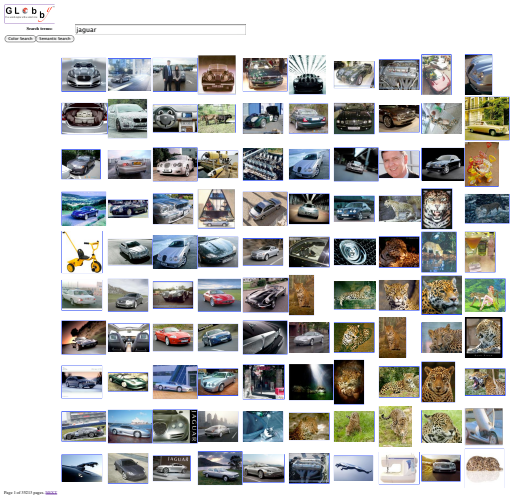
(a) Keyword-based layout (query: `pluto`)



(b) Globby colour layout with the highest rank at top left using the same set of images as in Figure 4.10(a)



(c) Keyword-based layout (query: `jaguar`)



(d) Globby semantic layout with the highest rank at top left using the same set of images as in Figure 4.10(c)

Figure 4.10: Comparisons between Yahoo! search engine (without Kernelised Sorting) and Globby search engine (with Kernelised Sorting).

Image Matching

Our first experiment is to match image halves. For this purpose we use the same set of Flickr images as in section 4.7.1 but split each image (40×40 pixels) into two equal halves (20×40 pixels). The aim is to match the image halves using Kernelised Sorting. More specifically, given x_i being the left half of an image and y_i being the right half of the same image, we want to find a permutation π which lines up x_i and $y_{\pi(i)}$ by maximizing the dependence.

Of course, this would be relatively easy if we were allowed to compare the two image halves x_i and $y_{\pi(i)}$ directly. While such comparison is clearly feasible for images where we *know* the compatibility function, it may not be possible for generic objects. Figure 4.3 shows the image matching result. For a total of 320 images we correctly match 140 pairs. This is quite respectable given that the chance level would be only 1 correct pair (a random permutation matrix has on expectation one nonzero diagonal entry).

Multilingual Document Matching

To illustrate that Kernelised Sorting is able to recover nontrivial similarity relations we apply our algorithm to the matching of multilingual documents in this second experiment. For this purpose we use the Europarl Parallel Corpus.[1] It is a collection of the proceedings of the European Parliament, dating back to 1996 (Koehn, 2005). We select the 300 longest documents of Danish (Da), Dutch (Nl), English (En), French (Fr), German (De), Italian (It), Portuguese (Pt), Spanish (Es), and Swedish (Sv). The purpose is to match the non-English documents (source languages) to its English translations (target language). Note that our algorithm does *not* require a cross-language dictionary. In fact, one could use Kernelised Sorting to generate a dictionary after an initial matching has been created.

We use standard TF-IDF (term frequency - inverse document frequency) features of a-bag-of-words kernel. As preprocessing we remove stopwords (via NLTK[2]) and perform stemming using Snowball.[3] Finally, the feature vectors are normalised to unit length in term of ℓ_2 norm. Since these kernel matrices on documents are notoriously diagonally dominant we use the bias-corrected version of our optimisation problem.

[1] http://www.statmt.org/europarl/
[2] http://nltk.sf.net/
[3] http://snowball.tartarus.org

As a reference we use a fairly straightforward means of document matching via its length. That is, longer documents in one language will be most probably translated into longer documents in the other language. This observation has also been used in the widely adopted sentence alignment method (Gale & Church, 1991). Alternatively, we can use a dictionary-based method as an upperbound for what can be achieved by matching. We translate the documents in the source languages into the target language word by word using Google Translate[1]. This effectively allows us to directly compare documents written in different languages. Now for each source language and the target language we can compute a kernel matrix based on a-bag-of-words kernel; and the ij-th entry of this kernel matrix is the similarity between document i in the source language and document j in the target language. Then we can use this kernel matrix and a linear assignment to find the matches between documents across languages.

The experimental results are summarised in Table 4.1. Here we use two versions of our algorithm: one with a fixed set of λs and the other with automatic tuning of λ (as in section 4.4). In practice we find that trying out different λ from a fixed set ($\lambda \in \{0.1, 0.2, \ldots, 1.0\}$) and then choosing the best λ in terms of the objective function works better than automatic tuning. Low matching performance for the document length-based method might be due to small variance in the document length after we choose the 300 longest documents. The dictionary-based method gives near perfect matching. Our method produces results consistent with the dictionary-based method, for instance the notably low performance for matching German documents to its English translations. We suspect that the difficulty of German-English document matching is inherent to this data set as it was also observed in Koehn (2005). Arguably the matching produced by Kernelised Sorting is quite encouraging as our method uses only a within language similarity measure while still matching more than 2/3 of what a dictionary-based method is capable of in most cases.

Data Attribute Matching

In our last experiment, we aim to match attributes of vectorial data. In our setup we use benchmark data sets for supervised learning from the UCI repository[2] and LibSVM site.[3] We split the attributes (or dimensions) of each data point into

[1] http://translate.google.com Note that we did not perform stemming on the words and thus the dictionary is highly customized to the problem at hand.

[2] http://archive.ics.uci.edu/ml

[3] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools

two halves, and we want to match them back. Here we use the estimation error to quantify the quality of the match. That is, assumed that y_i is associated with the observation x_i. In this case, we compare y_i and $y_{\pi(i)}$ using homogeneous misclassification loss for binary and multiclass problems and squared loss for regression problem. Note that this measure of goodness is different from the ones we used in image matching and document matching. This is because for data attribute matching we may not be able to match back the two halves of an individual data point exactly, but we can restore the overall characteristic of the data such as class separability.

To ensure good dependence between the splitted attributes, we choose a split which ensures correlation. This is achieved as follows: first we compute the correlation matrix of the data; then among the pairs of attributes which achieves the largest correlation we pick the dimension with the smallest index as the reference; next we choose the dimensions that have at least 0.5 correlation with the reference and split them equally into two sets, set A and set B (we also put the reference dimension into set A); last we divide the remaining dimensions (with less than 0.5 correlation with the reference) into two equal halves, and allocate them into set A and B respectively. This scheme ensures that at least one dimension in set B is strongly correlated with at least one dimension in set A.

As before, we use a Gaussian RBF kernel with median adjustment for the kernel width for both x and y. To obtain statistically meaningful results, we subsample 80% of the data 10 times and compute the error of the match on the subset (this is done in lieu of cross-validation since the latter is meaningless for matching). As a reference we compute the expected performance of random permutations which can be done exactly.[1] As a lower bound for the estimation error, we use the original data set and perform classification/regression using 10-fold cross-validation. The results are summarised in Table 4.2. Basically, the closer the results obtained by Kernelised Sorting to the lower bound the better. In many cases, Kernelised Sorting is able to restore significant information related to the class separability in the classification problems and the functional relationship in the regression problems.

[1]For classification: $1 - \sum_{i=1}^{|\mathcal{Y}|} p_i^2$ and for regression: $2\left(\mathbf{E}_y[y^2] - \mathbf{E}_y^2[y]\right)$. Here y denotes the class label and p_i denotes the proportion of class i in the data set.

Table 4.1: The number of correct matches from documents written in various source languages to those in English.
We compare Kernelised Sorting (KS) to a reference procedure which simply matches the lengths of documents (RE : Reference) and a dictionary-based approach (UB : Upper Bound). We also include results of line search or automatic tuning of λ (KS - LS). Reported are the numbers of correct matches (out of 300) for various source languages.

Language	PT	ES	FR	SV	DA	IT	NL	DE
KS	252	218	246	150	230	237	223	95
KS - LS	241	216	193	99	83	236	211	70
RE	9	12	8	6	6	11	7	4
UB	298	298	298	296	297	300	298	284

4.7.3 Multivariate Extension

In this experiment, we align 5 USPS digits of 0's using multiway HSIC. In this case, each non-zero pixel in an image is a data point and each image has 100 non-zero pixels. On each set of digits, we use a Gaussian RBF kernel with median adjustment of the kernel width. Furthermore we use the first digit as the target set (i.e. $\pi_1 = I$) and the other digits as the sources. The sorting performance is visualised by computing linear interpolations between the matching pixels. If meaningful matching is obtained, such interpolation will result in meaningful intermediate images (Jebara, 2004).

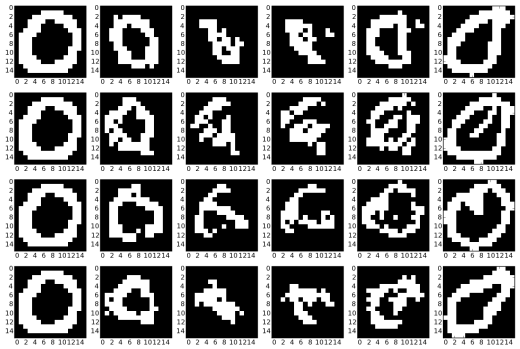
For comparison we also perform the same task using the method proposed by Jebara (2004). Briefly, Jebara (2004) proposes a method to sort many sets (or bags) of objects by maximizing likelihood under a Gaussian model to minimise the volume data occupies in Hibert space. An iterative likelihood maximisation procedure is devised by interleaving update of Gaussian's moments and adjustment of permutation configuration of each set of objects. We implemented our own version as we were unable to obtain their code for reasons beyond the control of the authors of Jebara (2004). We only experimented with the simpler version using the mean estimator (locking covariance matrix as a constant multiplication of an identity matrix) and LAP as it was observed that this simpler version performs as well as his more sophisticated counterpart based on a covariance estimator (allowing covariance matrix as an arbitrary positive semi-definite matrix) (Jebara, 2004). Here we also use a Gaussian RBF kernel with median trick as the base kernel. Although we are only interested in sorting 5 digits of 0's,

Table 4.2: Estimation error for data attribute matching
We compare estimation errors between the original data set (LB : Lower Bound), data set after Kernelised Sorting (KS), and data set after random permutation (RE : Reference).

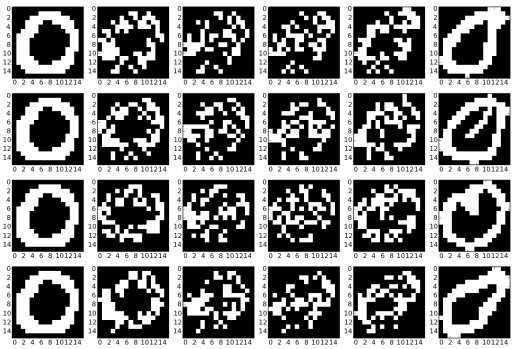
Type	Data set	m	KS	RE	LB
Binary	australian	690	0.29±0.02	0.49	0.21±0.04
	breastcancer	683	0.06±0.01	0.46	0.06±0.03
	derm	358	0.08±0.01	0.43	0.00±0.00
	optdigits	765	0.01±0.00	0.49	0.01±0.00
	wdbc	569	0.11±0.04	0.47	0.05±0.02
Multiclass	satimage	620	0.20±0.01	0.80	0.13±0.04
	segment	693	0.58±0.02	0.86	0.05±0.02
	vehicle	423	0.58±0.08	0.75	0.24±0.07
Regression	abalone	417	13.9±1.70	18.7	6.44±3.14
	bodyfat	252	4.5±0.37	7.20	3.80±0.76

the method of Jebara (2004) requires more digits (200 in our experiments) to get a decent ML estimate of the feature space mean. As such, the usage of Jebara (2004)'s method in finding a correspondence with just two sets of observations (as in Section 4.7.1, 4.7.2, 4.7.2, and 4.7.2), i.e. this translates to get an ML mean estimate of the Gaussian likelihood with just two samples, is not obvious.

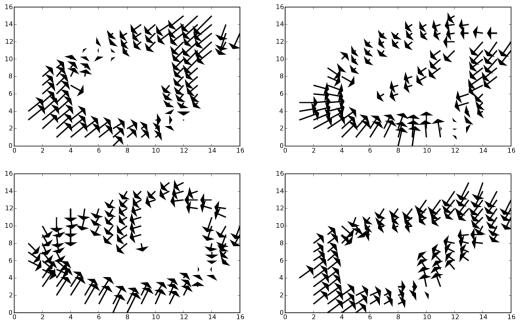
The interpolation results are shown in Figure 4.11(a) and 4.11(b). Due to the symmetric structure of the 0's digit, some of the correspondences are reversed (the top is matched to the bottom and the bottom is matched to the top) which is apparent from Figure 4.11(a). Nevertheless, the interpolations obtained with HSIC seem to produce a better local consistency than those obtained with entropy. This is clear from the flows of arrows in the velocity plots (arrows are pointing away from a matching pixel in the source digits) shown in Figure 4.11(c) and 4.11(d) for each digit pair in Figure 4.11(a) and 4.11(b). For example, in the upper right plot of Figure 4.11(c), all the arrows 'inside' the 0 are pointing downwards. However, in Figure 4.11(d) some arrows are pointing downwards but some upwards. This local flow consistency implies that in the matching neighbouring pixels in one digit will be mapped to the neighbouring locations in the other digit as well.



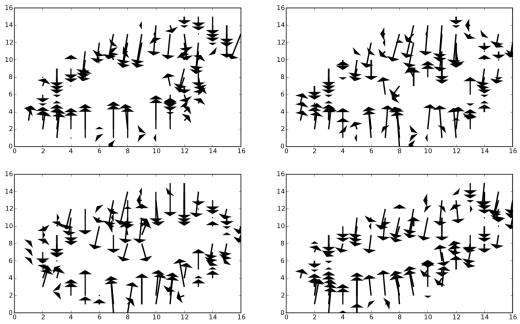
(a) Linear interpolation using multiway HSIC



(b) Linear interpolation using Entropy



(c) Arrows showing the matching of strokes of digit pairs sorted using multiway HSIC.



(d) Arrows showing the matching of strokes of digit pairs sorted using Entropy

Figure 4.11: Linear interpolation of 4 pairs of the digit 0 after sorting using multiway HSIC and Entropy Jebara (2004).

4.8 Conclusion

In this chapter, we generalised sorting by maximizing the dependency between matched pairs of observations by means of the Hilbert Schmidt Independence Criterion. This way we are able to perform matching *without* the need of a cross-domain similarity measure and we managed to put sorting and assignment operations onto an information theoretic footing. The proposed sorting algorithm is efficient and it can be applied to a variety of different problems ranging from data visualisation to image and multilingual document matching. Moreover, we showed that our approach is closely related to matching and object layout algorithms and that by changing the dependence measure we are able to recover previous work on sorting in Hilbert Spaces.

Chapter 5

Multitask Learning without Label Correspondences

In this chapter, we introduce a learning setting of jointly learning several related tasks where each task has potentially distinct label sets, and label correspondences are not readily available. It is widely known in machine learning that if several tasks are related, then learning them simultaneously can improve performance (Caruana, 1997; Argyriou et al., 2008; Yu et al., 2005; Ando & Zhang, 2005). For instance, a personalized spam classifier trained with data from several different users is likely to be more accurate than one that is trained with data from a single user. Traditionally, multitask learning assumes that the set of labels for all the tasks are the same, or that we have access to an oracle that gives correspondences between the label sets. However, as we argue below, in many natural settings these assumptions are not satisfied.

5.1 Motivating Examples

Our motivating example is the problem of learning to automatically categorise objects on the Internet into an ontology or directory. It is well established that many web-related objects such as web directories and RSS directories admit a (hierarchical) categorisation, and web directories aim to do this in a semi-automated fashion. For instance, it is desirable, when building a categoriser for the Yahoo! directory[1], to take into account other web directories such as DMOZ[2]. Although the tasks are clearly related, their label sets are not identical. For instance, some

[1] http://dir.yahoo.com/
[2] http://www.dmoz.org/

section heading and sub-headings may be named differently in the two directories. Furthermore, different editors may have made different decisions about the ontology depth and structure, leading to incompatibilities. To make matters worse, these ontologies evolve with time and certain topic labels may die naturally due to lack of interest or expertise while other new topic labels may be added to the directory. Given the large label space, it is unrealistic to expect that a label mapping function is readily available. However, the two tasks are clearly related and learning them simultaneously is likely to improve performance.

5.2 Problem Definition

We present a method to learn classifiers from a collection of related tasks or data sets, in which each task has its own label dictionary, without constructing an explicit label mapping among them. Formally, if one views learning as the task of inferring a function f from the input space $\mathcal{X}$ to the output space $\mathcal{Y}$, then multitask learning is the problem of inferring several functions $f_i : \mathcal{X}_i \mapsto \mathcal{Y}_i$ simultaneously. In the standard multitask learning, one either assumes that the set of labels $\mathcal{Y}_i$ for all the tasks are the same (that is, $\mathcal{Y}_i = \mathcal{Y}$ for all i), or that we have access to an oracle mapping function $g_{i,j} : \mathcal{Y}_i \mapsto \mathcal{Y}_j$. It is our goal to design algorithms which can learn jointly several functions merely based on the information that the tasks are related while sidestepping the label mapping construction.

5.3 The Model

Our solution relies on the duality principle of approximate maximum entropy and maximum a posteriori estimation (Altun & Smola, 2006; Dudík & Schapire, 2006), and an assumption that, for correlated label sets, the joint label distribution should exhibit high mutual information. We are then able to formulate the problem as that of maximising mutual information among the labels sets.

5.3.1 Maximum Entropy Duality for Conditional Distributions

Here we briefly summarise the well known duality relation between approximate conditional maximum entropy estimation and maximum a posteriori estima-

tion (MAP) (Altun & Smola, 2006; Dudík & Schapire, 2006). We will exploit this in Section 5.4. Recall the definition of the Shannon entropy, $H(y|x) := -\sum_y p(y|x) \log p(y|x)$, where $p(y|x)$ is a conditional distribution on the space of labels $\mathcal{Y}$. Let $x \in \mathcal{X}$ and assume the existence of $\phi(x, y) : \mathcal{X} \times \mathcal{Y} \mapsto \mathcal{H}$, a feature map into a Hilbert space $\mathcal{H}$. Given a data set $(X, Y) := \{(x_1, y_1), \ldots, (x_m, y_m)\}$, where $X := \{x_1, \ldots, x_m\}$, define

$$\mathbf{E}_{y\sim p(y|X)}\left[\phi(X, y)\right] := \frac{1}{m}\sum_{i=1}^{m} \mathbf{E}_{y\sim p(y|x_i)}\left[\phi(x_i, y)\right], \text{ and } \mu = \frac{1}{m}\sum_{i=1}^{m} \phi(x_i, y_i). \quad (5.1)$$

Lemma 42 (Altun & Smola (2006), Lemma 6) *With the above notation we have*

$$\min_{p(y|x)} \sum_{i=1}^{m} -H(y|x_i) \;\; s.t. \;\; \left\|\mathbf{E}_{y\sim p(y|X)}\left[\phi(X, y)\right] - \mu\right\|_{\mathcal{H}} \leq \epsilon \;\; and \;\; \sum_{y\in\mathcal{Y}} p(y|x_i) = 1 \quad (5.2a)$$

$$= \max_{\theta} \langle\theta, \mu\rangle_{\mathcal{H}} - \sum_{i=1}^{m} \log \sum_{y} \exp(\langle\theta, \phi(x_i, y)\rangle) - \epsilon \left\|\theta\right\|_{\mathcal{H}}. \quad (5.2b)$$

Although we presented a version of the above theorem using Hilbert spaces, it can also be extended to Banach spaces. Choosing different Banach space norms recovers well known algorithms such as ℓ_1 or ℓ_2 regularized logistic regression. Also note that by enforcing the moment matching constraint exactly, that is, setting $\epsilon = 0$, we recover the well-known duality between maximum (Shannon) entropy and maximum likelihood (ML) estimation.

5.3.2 Multitask Learning via Mutual Information

For the purpose of explaining our basic idea, we focus on the case when we want to integrate two data sources such as Yahoo! directory and DMOZ. Associated with each data source are labels $Y = \{y_1, \ldots, y_c\} \subseteq \mathcal{Y}$ and observations $X = \{x_1, \ldots, x_m\} \subseteq \mathcal{X}$ (resp. $Y' = \{y'_1, \ldots, y'_{c'}\} \subseteq \mathcal{Y}'$ and $X' = \{x'_1, \ldots, x'_{m'}\} \subseteq \mathcal{X}'$). The observations are *disjoint* but we assume that they are drawn from the same domain, i.e., $\mathcal{X} = \mathcal{X}'$ (in our running example they are webpages).

If we are interested to solve each of the categorisation tasks independently, a maximum entropy estimator described in Section 5.3.1 can be readily employed (Ghamrawi & McCallum, 2005). Here we would like to learn the two tasks simultaneously in order to improve classification accuracy. Assuming that

the labels are different yet correlated we should assume that the joint distribution $p(y, y')$ displays high mutual information between y and y'. Recall that the mutual information between random variables y and y' is defined as $I(y, y') = H(y) + H(y') - H(y, y')$, and that this quantity is high when the two variables are mutually dependent. To illustrate this, consider in our running example of integrating Yahoo! and DMOZ web directories, we would expect there is a high mutual dependency between section heading 'Computer & Internet' at Yahoo! directory and 'Computers' at DMOZ directory although they are named somewhat slightly different. Since the marginal distributions over the labels, $p(y)$ and $p(y')$ are fixed, maximising mutual information can then be viewed as minimising the joint entropy

$$H(y, y') = -\sum_{y,y'} p(y, y') \log p(y, y'). \tag{5.3}$$

This reasoning leads us to adding the joint entropy as an additional term for the objective function of the multitask problem. If we define

$$\mu = \frac{1}{m}\sum_{i=1}^{m} \phi(x_i, y_i) \text{ and } \mu' = \frac{1}{m'}\sum_{i=1}^{m'} \phi(x'_i, y'_i), \tag{5.4}$$

then we have the following objective function

$$\underset{p(y|x)}{\text{maximise}} \quad \sum_{i=1}^{m} H(y|x_i) + \sum_{i=1}^{m'} H(y'|x'_i) - \lambda H(y, y') \quad \text{for some } \lambda > 0 \tag{5.5a}$$

$$\text{s.t.} \quad \left\| \mathbf{E}_{y\sim p(y|X)} \left[\phi(X, y)\right] - \mu \right\| \leq \epsilon \text{ and } \sum_{y\in\mathcal{Y}} p(y|x_i) = 1 \tag{5.5b}$$

$$\left\| \mathbf{E}_{y'\sim p(y'|X')} \left[\phi'(X', y')\right] - \mu' \right\| \leq \epsilon' \text{ and } \sum_{y'\in\mathcal{Y}'} p(y'|x'_i) = 1. \tag{5.5c}$$

Intuitively, the above objective function tries to find a 'simple' distribution p which is consistent with the observed samples via moment matching constraints while also taking into account task relatedness. We can recover the single task maximum entropy estimator by removing the joint entropy term (by setting $\lambda = 0$), since the optimisation problem (the objective functions as well as the constraints) in (5.5) will be decoupled in terms of $p(y|x)$ and $p(y'|x')$. There are two main challenges in solving (5.5):

- The joint entropy term $H(y, y')$ is concave, hence the above objective of the optimisation problem is not concave in general (it is the difference of two concave functions). We therefore propose to solve this non-concave problem using the Convex-ConCave Procedure (CCCP) (Section 2.6.1).

- The joint distribution between labels $p(y, y')$ is unknown. We will estimate this quantity (therefore the joint entropy quantity) from the observations x and x'. Further, we assume that y and y' are conditionally independent given an arbitrary input $x \in \mathcal{X}$, that is $p(y, y'|x) = p(y|x)p(y'|x)$. For instance, in our example, annotations made by an editor at Yahoo! and an editor at DMOZ on the set of webpages are assumed conditionally independent given the set of webpages. This assumption essentially means that the labelling process depends entirely on the set of webpages, i.e., any other latent factors that might connect the two editors are ignored.

In the following section we discuss in further detail how to address these two challenges, as well as the resulting optimisation problem obtained, which can be solved efficiently by existing convex solvers.

5.4 Optimisation

To find a local maximum of the matching problem we may take recourse to a well-known algorithm, namely the Convex-ConCave Procedure (CCCP). Therefore, one potential approach to solve the optimisation problem in (5.5) is to use successive linear lower bounds on $H(y, y')$ and to solve the resulting decoupled problems in $p(y|x)$ and $p(y'|x')$ separately. We estimate the joint entropy term $H(y, y')$ by its empirical quantity on x and x' with the conditional independence assumption (in the sequel, we make the dependency of $p(y|x)$ on a parameter θ explicit and similarly for the dependency of $p(y'|x')$ on θ'), that is

$$H(y, y'|X) = -\sum_{y,y'} \left[\frac{1}{m}\sum_{i=1}^{m} p(y|x_i, \theta)p(y'|x_i, \theta')\right] \log \left[\frac{1}{m}\sum_{j=1}^{m} p(y|x_j, \theta)p(y'|x_j, \theta')\right], \tag{5.6}$$

and similarly for $H(y, y'|X')$. Each iteration of CCCP approximates the convex part (negative joint entropy) by its tangent, that is $\langle \nabla h(w)|_{w'}, w\rangle$ in (2.45). Therefore, taking derivatives of the joint entropy with respect to $p(y|x_i)$ and evaluating at parameters at iteration $t-1$, denoted as θ_{t-1} and θ'_{t-1}, yields

$$g_y(x_i) := -\partial_{p(y|x_i)} H(y, y'|X) \tag{5.7}$$

$$= \frac{1}{m}\sum_{y'} \left[1 + \log \frac{1}{m}\sum_{j=1}^{m} p(y|x_j, \theta_{t-1})p(y'|x_j, \theta'_{t-1})\right] p(y'|x_i, \theta'_{t-1}). \tag{5.8}$$

Define similarly $g_y(x'_i)$, $g_{y'}(x_i)$, and $g_{y'}(x'_i)$ for the derivative with respect to $p(y|x'_i)$, $p(y'|x_i)$ and $p(y'|x'_i)$, respectively. This leads, by optimizing the lower bound in (2.45), to the following decoupled optimisation problems in $p(y|x_i)$ and an analogous problem in $p(y'|x'_i)$:

$$\min_{p(y|x)} \sum_{i=1}^{m} \left[-H(y|x_i) + \lambda \sum_y g_y(x_i) p(y|x_i) \right] + \sum_{i=1}^{m'} \left[-H(y|x'_i) + \lambda' \sum_y g_y(x'_i) p(y|x'_i) \right] \tag{5.9a}$$

$$\text{subject to} \quad \left\| \mathbf{E}_{y \sim p(y|X)}[\phi(X, y)] - \mu \right\| \leq \epsilon. \tag{5.9b}$$

The above objective function is still in the form of maximum entropy estimation, with the linearisation of the joint entropy quantities acting like additional evidence terms. Furthermore, we also impose an additional maximum entropy requirement on the 'off-set' observations $p(y|x'_i)$, as after all we also want the 'simplicity' requirement of the distribution p on the input x'_i. We can of course weigh the requirement on 'off-set' observations differently.

While we succeed in reducing the non-concave objective function in (5.5) to a decoupled concave objective function in (5.9), it might be desirable to solve the problem in the dual space due to difficulty in handling the constraint in (5.9b). The following lemma shows the duality of the objective function in (5.9).

Lemma 43 *The corresponding Fenchel's dual of (5.9) is*

$$\min_{\theta} \sum_{i=1}^{m} \log \sum_y \exp(\langle \theta, \phi(x_i, y) \rangle - \lambda g_y(x_i)) + \sum_{i=1}^{m'} \log \sum_y \exp(\langle \theta, \phi(x'_i, y) \rangle - \lambda' g_y(x'_i)) - \frac{1}{m} \sum_{i=1}^{m} \langle \theta, \phi(x_i, y_i) \rangle + \epsilon \left\| \theta \right\|_{\ell_2} \tag{5.10}$$

Proof Denote by $\mathcal{B}$ a Banach space and let $\mathcal{B}^*$ be its dual. Denote space of conditional distributions $\mathcal{P} = \{p_{y|x} \mid p(y|x) \geq 0, \sum_{y \in \mathcal{Y}} p(y|x) = 1, \forall x \in \mathcal{X}, y \in \mathcal{Y}\}$. Let A be the conditional expectation operator of the feature map $\phi(x, y)$ with respect to conditional distribution $p(y|x)$, that is $A p_{y|x} = \mathbf{E}_{y \sim p(y|x)}[\phi(x, y)]$. Fenchel's Duality (Borwein & Zhu, 2005, Theorem 4.4.3) states

$$\inf_{p_{y|x} \in \mathcal{P}} \{ f(p_{y|x}) + g(A p_{y|x}) \} = \sup_{\theta \in \mathcal{B}^*} \{ -f^*(A^* \theta) - g^*(-\theta) \}. \tag{5.11}$$

First, note that the adjoint of the linear operator A is $\left\langle A p_{y|x}, \theta \right\rangle = \left\langle A^* \theta, p_{y|x} \right\rangle$, then we have $\left\langle \sum_{y \in \mathcal{Y}} p_{y|x} \phi(x, y), \theta \right\rangle = \sum_{y \in \mathcal{Y}} p_{y|x} \left\langle \phi(x, y), \theta \right\rangle$, thus $A^* \theta = \langle \phi(x, y), \theta \rangle$.

Algorithm 5 Multitask Mutual Information

Input: Datasets (X, Y) and (X', Y') with $\mathcal{Y} \neq \mathcal{Y}'$, number of iterations N
Output: θ, θ'
Initialize $p(y) = 1/|\mathcal{Y}|$ and $p(y') = 1/|\mathcal{Y}'|$
for $t = 1$ to N **do**
  Solve the dual problem in (5.10) w.r.t. $p(y|x, \theta)$ and obtain θ_t
  Solve the dual problem in (5.10) w.r.t. $p(y'|x', \theta')$ and obtain θ'_t
end for
return $\theta \leftarrow \theta_N$, $\theta' \leftarrow \theta'_N$

Define $f(p_{y|x}) = p_{y|x} \log p_{y|x} + c \cdot p_{y|x} + \Lambda_{p_{y|x}}(\sum_{y \in \mathcal{Y}} p_{y|x} - 1)$ where c is the constant part w.r.t. $p_{y|x}$ (i.e. the gradient of the joint entropy), we the have $f^*(p^*_{y|x}) = \Lambda_{p^*_{y|x}} + \exp(p^*_{y|x} - 1 - c - \Lambda_{p^*_{y|x}})$ as its dual. Hence the dual of $\sum_{x \in \mathcal{X}} \left[-H(p_{y|x}) + \lambda \sum_{y \in \mathcal{Y}} g_y(x) p_{y|x} \right]$ is

$$\sum_{i=1}^{m} \left[\sum_{y} \exp(\langle \theta, \phi(x_i, y) \rangle - 1 - \lambda g_y(x_i) - \Lambda_{p^*_{y|x}}) + \Lambda_{p^*_{y|x}} \right] \tag{5.12}$$

Solving for optimality in $\Lambda_{p^*_{y|x}}$ gives $\sum_{i=1}^{m} \log \sum_y \exp(\langle \theta, \phi(x_i, y) \rangle - \lambda g_y(x_i))$. Similarly for $x' \in \mathcal{X}'$. The dual of the approximate moment matching constraint follows directly from (Altun & Smola, 2006, Lemma 6). ■

The above dual problem still has the form of logistic regression with the additional evidence terms from task relatedness appearing in the log-partition function. Several existing convex solvers can be used to solve the optimisation problem in (5.10) efficiently. Refer to Algorithm 5 for a pseudocode of our proposed method.

Initialisation For each iteration of CCCP, the linearisation part of the joint entropy function requires the value of θ and θ' at the previous iteration (refer to (5.8)). At the beginning of the iteration, we can start the algorithm with a uniform prior, i.e. set $p(y) = 1/|\mathcal{Y}|$ and $p(y') = 1/|\mathcal{Y}'|$.

5.5 Related Work

As described earlier, our work is closely related to the research efforts on multitask learning, where the problem of simultaneously learning multiple related

tasks is addressed. Several papers have empirically and theoretically highlighted the benefits of multitask learning over single-task learning when the tasks are related. There are several approaches to define task relatedness. The works of Argyriou et al. (2008); Obozinski et al. (2007); Flamary et al. (2009) consider the setting when the tasks to be learned jointly share a common subset of features. This can be achieved by adding a mixed-norm regularisation term that favours a common sparsity profile in features shared by all tasks. Task relatedness can also be modelled as learning functions that are close to each other in some sense (Yu et al., 2005; Evgeniou et al., 2005). Crammer et al. (2007) consider the setting where, in addition to multiple sources of data, estimates of the dissimilarities between these sources are also available. There is also work on data integration via multitask learning where each data source has the same binary label space, whereas the attributes of the inputs can admit different orderings as well as be linearly transformed (Ben-David et al., 2002). Developed independently of our work, Parameswaran & Weinberger (2010) addressed the same problem of multitask learning without label correspondences by learning a Mahalanobis metric that is shared amongst all the tasks and another Mahalanobis metric specific to each task in a framework of large margin nearest neighbour (Weinberger & Saul, 2009).

5.6 Experiments

To assess the performance of our proposed multitask algorithm, we perform binary n-task ($n \in \{3, 5, 7, 10\}$) experiments on MNIST digit dataset and a multiclass 2-task experiment on the Reuters1-v2 dataset plus an application on integrating Yahoo! and DMOZ web directory. We detail those experiments in turn in the following sections.

5.6.1 MNIST

Datasets MNIST data set[1] consists of 28×28-size images of hand-written digits from 0 through 9. We use a small sample of the available training set to simulate the situation when we only have limited number of labeled examples and test the performance on the entire available test set. In this experiment, we look at a binary n-task ($n \in \{3, 5, 7, 10\}$) problem. We consider digits $\{8, 9, 0\}$, $\{6, 7, 8, 9, 0\}$, $\{4, 5, 6, 7, 8, 9, 0\}$ and $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 0\}$ for the 3-task, 5-task,

[1] http://yann.lecun.com/exdb/mnist

7-task and 10-task, respectively. To simulate the problem that we have distinct label dictionaries for each task, we consider the following setting: in the 3-task problem, the first task has binary labels $\{+1, -1\}$, where label $+1$ means digit 8 and label -1 means digit 9 and 0; in the second task, label $+1$ means digit 9 and label -1 means digit 8 and 0; lastly in the third task, label $+1$ means digit 0 and label -1 means digit 8 and 9. Similar one-against-rest grouping is also used for 5-task, 7-task and 10-task problems. Each of the tasks has its *own* input x.

Algorithms We couldn't find in the literature of multitask learning methods addressing the same problem as the one we study: learn multiple tasks when there is no correspondence between the output spaces. Therefore we compared the performance of our multitask method against the baseline given by the maximum entropy estimator applied to each of the tasks independently. Note that we focus on the setting in which data sources have disjoint sets of covariate observations (vide Section 5.3.2) and thus a simple strategy of multilabel prediction with union of label sets corresponds to our baseline. For both ours and the baseline method, we use a Gaussian kernel to define the implicit feature map on the inputs. The width of the kernel was set to the median between pairs of observations, as suggested in Schölkopf (1997). The regularisation parameter was tuned for the single task estimator and the same value was used for the multitask. The weight on the joint entropy term was set to be equal to 1.

Pairwise Label Correlation Section 5.3.2 describes the multitask objective function for the case of the 2-task problem. For the case when the number of tasks to be learned jointly is greater than 2, we experiment in two different ways: in one approach we can define the joint entropy term on the full joint distribution, that is when we want to learn jointly 3 different tasks having label y, y' and y'', we can then define the joint entropy as $H(y, y', y'') = -\sum_{y,y',y''} p(y, y', y'') \log p(y, y', y'')$. As more computationally efficient way, we can consider the joint entropy on the pairwise distribution instead. We found that the performance of our method is quite similar for the two cases and we report results only on the pairwise case.

Results The experiments are repeated for 10 times and the results are summarised in Table 5.1. We find that, on average, jointly learning the multiple related tasks *always* improves the classification accuracy. When assessing the performance on each of the tasks, we notice that the advantage of learning jointly is particularly significant for those tasks with smaller number of observations.

Table 5.1: Performance assessment, Accuracy ± STD. $m(m')$ denotes the number of training data points (number of test points). `STL:` single task learning; `MTL:` multi task learning and `Upper Bound:` multi class learning. **Boldface** indicates a significance difference between STL and MTL (one-sided paired Welch t-test with 99.95% confidence level).

Tasks	m (m')	STL	MTL	Upper Bound
8 \-8	15 (2963)	77.39±5.23	**80.03±4.83**	93.42±0.87
9 \-9	15 (2963)	91.12±5.94	91.96±5.42	95.99±0.75
0 \-0	120 (2963)	98.66±0.67	98.21±0.92	98.79±0.25
Average		89.06	90.07	96.07
6 \-6	25 (4949)	81.79±10.18	**83.86±9.51**	96.37±1.06
7 \-7	25 (4949)	70.73±16.58	**72.84±15.77**	91.99±2.23
8 \-8	25 (4949)	62.52±10.15	**66.77±9.43**	92.05±1.76
9 \-9	25 (4949)	63.80±13.70	**67.26±12.65**	92.53±1.65
0 \-0	150 (4949)	**97.35±1.33**	96.60±1.64	97.59±0.62
Average		75.84	77.47	94.10
4 \-4	70 (6823)	71.69±6.83	**73.49±6.77**	91.20±1.55
5 \-5	70 (6823)	67.55±4.70	**70.10±4.61**	89.30±0.34
6 \-6	70 (6823)	86.31±2.93	**87.21±2.77**	94.03±0.95
7 \-7	70 (6823)	83.34±3.54	84.02±3.69	91.94±0.90
8 \-8	70 (6823)	75.61±6.00	76.97±5.12	87.46±1.69
9 \-9	70 (6823)	63.69±11.42	65.74±10.15	86.89±1.79
0 \-0	210 (6823)	**97.20±1.49**	96.56±1.67	97.24±0.73
Average		77.91	79.16	91.15
1 \-1	100 (10000)	96.59±2.11	96.80±1.91	96.89±0.59
2 \-2	100 (10000)	67.77±3.49	**69.95±2.68**	88.74±1.94
3 \-3	100 (10000)	72.59±5.90	**74.18±5.54**	87.59±2.95
4 \-4	100 (10000)	69.91±5.82	71.76±5.47	92.87±0.94
5 \-5	100 (10000)	53.78±2.78	**57.26±2.72**	85.71±1.38
6 \-6	100 (10000)	79.22±5.21	80.54±4.53	92.93±0.98
7 \-7	100 (10000)	76.57±10.2	77.18±9.43	89.83±1.24
8 \-8	100 (10000)	63.57±2.65	**65.85±2.50**	83.51±0.63
9 \-9	100 (10000)	63.28±6.69	**65.38±6.09**	84.94±1.45
0 \-0	300 (10000)	**98.43±0.84**	97.81±1.01	98.49±0.40
Average		74.17	75.67	90.82

5.6.2 Ontology

News Ontologies In this experiment, we consider multiclass learning in a 2-task problem. We use the Reuters1-v2 news article dataset (Lewis et al., 2004) which has been pre-processed[1]. In the pre-processing stage, the label hierarchy is reorganised by mapping the data set to the second level of topic hierarchy. The documents that only have labels of the third or fourth levels are mapped to their parent category of the second level. The documents that only have labels of the first level are not mapped onto any category. Lastly any multi-labelled instances are removed. The second level hierarchy consists of 53 categories and we perform experiments on the top 10 categories. TF-IDF features are used, and the dictionary size (feature dimension) is 47236. For this experiment, we use 12500 news articles to form one set of data and another 12500 news article to form the second set of data. In the first set, we group the news articles having the label $\{1, 2\}$, $\{3, 4\}$, $\{5, 6\}$, $\{7, 8\}$ and $\{9, 10\}$ and re-label it as $\{1, 2, 3, 4, 5\}$. For the second set of data, it also has 5 labels but this time the labels are generated by $\{1, 6\}$, $\{2, 7\}$, $\{3, 8\}$, $\{4, 9\}$ and $\{5, 10\}$ grouping. We split equally the news articles on each set to form training and test sets. We run a maximum entropy estimator independently, $p(y|x, \theta)$ and $p(y'|x', \theta')$, on the two sets achieving accuracy of 92.59% for the first set and 91.53% for the second set. We then learn the two sets of the news articles jointly and in the first test set, we achieve accuracy of 93.81%. For the second test set, we achieve an accuracy of 93.31%. This experiment further emphasises that it is possible to learn several related tasks simultaneously even though they have different label sets and it is beneficial to do so.

Web Ontologies We also perform an experiment on the data integration of Yahoo! and DMOZ web directories. We consider the top level of the Yahoo!'s topic tree and sample web links listed in the directory. Similarly we also consider the top level of the DMOZ topic tree and retrieve sampled web links. We consider the content of the first page of each web link as our input data. It is possible that the first page that is being linked from the web directory contain mostly images (for the purpose of attracting visitors), thus we only consider those webpages that have enough texts to be a valid input. This gives us 19186 webpages for Yahoo! and 35270 for DMOZ. For the sake of getting enough texts associated with each link, we can actually crawl many more pages associated with the link. However, we find that it is quite damaging to do so because as we crawl deeper the topic of

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html

Table 5.2: Yahoo! Top Level Categorisation Results. `STL:` single task learning accuracy; `MTL:` multi task learning accuracy; `% Imp.:` relative performance improvement. The highest relative improvement at Yahoo! is for the topic of *'Computer & Internet'*, i.e. there is an increase in accuracy from 48.12% to 52.57%. Interestingly, DMOZ has a similar topic but was called *'Computers'* and it achieves accuracy of 75.72%.

Topic	MTL/STL	(% Imp.)	Topic	MTL/STL	(% Imp.)
Arts	56.27/55.11	(2.10)	News & Media	15.23/14.83	(1.03)
Business & Economy	66.52/66.88	(-0.53)	Recreation	68.81/67.00	(2.70)
Computer & Internet	52.57/48.12	(9.25)	Reference	26.65/24.81	(7.42)
Education	62.48/63.02	(-0.85)	Regional	62.85/61.86	(1.60)
Entertainment	63.30/61.37	(3.14)	Science	78.58/79.75	(-1.46)
Government	24.44/22.88	(6.82)	Social Science	31.55/30.68	(2.84)
Health	85.42/85.27	(1.76)	Society & Culture	49.51/49.05	(0.94)

the texts are rapidly changing. We use the standard bag-of-words representation with TF-IDF weighting as our features. The dictionary size (feature dimension) is 27075. We then use 2000 web pages from Yahoo! and 2000 pages from DMOZ as training sets and the remainder as test sets. Table 5.2 and 5.3 summarise the experimental results.

From the experimental results on web directories integration, we observe the following:

- Similarly to the experiments on MNIST digits and Reuters1-v2 news articles, multitask learning always helps on *average*, i.e. the average relative improvements are positive for both Yahoo! and DMOZ web directories;

- The improvement of multitask to single task on each topic is more prominent for Yahoo! web directories and is negligible for DMOZ web directories (2.62% and 0.07%, respectively). Arguably, this can be partly explained as Yahoo! has lower average topic categorisation accuracy than DMOZ (c.f. 60.22% and 64.68 %, respectively). It seems that there is much more knowledge to be shared from DMOZ to Yahoo! in the hope to increase the latter's classification accuracies;

Table 5.3: DMOZ Top Level Categorisation Results. `STL:` single task learning accuracy; `MTL:` multi task learning accuracy; `% Imp.:` relative performance improvement. The improvement of multitask to single task on each topic is negligible for DMOZ web directories. Arguably, this can be partly explained as DMOZ has higher average topic categorisation accuracy than Yahoo! and there might be more knowledge to be shared from DMOZ to Yahoo! than vice versa.

Topic	MTL/STL	(% Imp.)	Topic	MTL/STL	(% Imp.)
Arts	57.52/57.84	(-0.5)	Reference	67.42/67.42	(0)
Business	54.02/53.05	(1.83)	Regional	28.59/28.56	(0.10)
Computers	75.08/75.72	(-0.8)	Science	42.67/42.09	(1.38)
Games	78.58/78.58	(0)	Shopping	75.20/74.62	(0.54)
Health	82.34/82.55	(-0.14)	Society	57.68/58.20	(-0.89)
Home	67.47/67.47	(0)	Sports	83.49/83.53	(-0.05)
News	61.70/62.01	(-0.49)	World	87.80/87.57	(0.26)
Recreation	58.04/58.25	(-0.36)			

- Looking closely at accuracy at each topic, the highest relative improvement at Yahoo! is for the topic of *'Computer & Internet'*, i.e. there is an increase in accuracy from 48.12% to 52.57%. Interestingly, DMOZ has a similar topic but was called *'Computers'* and it achieves accuracy of 75.72%. The improvement might be partly because our proposed method is able to discover the implicit label correlations despite the two topics being named differently;

- Regarding the worst classified categories, we have *'News & Media'* for Yahoo! and *'Regional'* for DMOZ. This is intuitive since those two topics can indeed cover a wide range of subjects. The easiest category to be classified is *'Health'* for Yahoo! and *'World'* for DMOZ. As well, this is quite intuitive as the world of health contains mostly specific jargon and the world of world has much language-specific webpage content.

5.7 Conclusion

We presented a method to learn classifiers from a collection of related tasks or data sets, in which each task has its own label set. Our method works without the need of an explicit mapping between the label spaces of the different tasks.

We formulate the problem as one of maximising the mutual information among the label sets. Our experiments on binary n-task ($n \in \{3, 5, 7, 10\}$) and multiclass 2-task problems revealed that, on average, jointly learning the multiple related tasks, albeit with different label sets, always improves the classification accuracy. We also provided experiments on a prototypical application of our method: classifying in Yahoo! and DMOZ web directories. Here we deliberately used small amounts of data–a common situation in commercial tagging and classification. This shows that classification accuracy of Yahoo! significantly increased. Given that DMOZ classification was already 4.5% better prior to the application of our method, this shows the method was able to transfer classification accuracy from the DMOZ task to the Yahoo! task. Furthermore, the experiments seem to suggest that our proposed method is able to discover implicit label correlations despite the lack of label correspondences.

Although the experiments on web directories integration is encouraging, we have clearly only touched the surface of possibilities to be explored. While we focused on the categorisation at the top level of the topic tree, it might be beneficial (and further highlight the usefulness of multitask learning, as observed in Argyriou et al. (2008); Yu et al. (2005); Ando & Zhang (2005); Evgeniou et al. (2005)) to consider categorisation at deeper levels (take for example the second level of the tree), where we have much fewer observations for each category. In the extreme case, we might consider the labels as corresponding to a directed acyclic graph (DAG) and encode the feature map associated with the label hierarchy accordingly. One instance as considered in Cai & Hofmann (2004) is to use a feature map $\phi(y) \in \mathbb{R}^k$ for k nodes in the DAG (excluding the root node) and associate with every label y the vector describing the path from the root node to y, ignoring the root node itself.

Furthermore, the application of data integration which admit a hierarchical categorisation goes beyond web related objects. With our method, it is also now possible to learn classifiers from a collection of related gene-ontology graphs (Ashburner et al., 2000) or patent hierarchies (Cai & Hofmann, 2004).

Part II

Scalable Solution for Existing Machine Learning Problems

Chapter 6

Distribution Matching for Transduction

In this chapter, we present an algorithm for learning with labelled and unlabelled data simultaneously, called transduction, by matching the distributions over the outputs on labelled and unlabelled data. The algorithm is linear in the number of data points, thus scales to very large problems. A scalable transductive learning method has diverse applications in Internet as we illustrate below.

6.1 Motivating Examples

Consider the setting of an image search service. Typically, not all the returned images from the search engine are even related to the query. In an ideal situation, we would have those not-so-related images to be filtered out. This could possibly be achieved by supervised feature extraction of the content of the images. For a given keyword and returned images, human subjects can determine whether the given images were indeed relevant images to the keyword. However, this is a costly process and at best we would only have a small number of annotated images. This leaves billions of images on the Internet un-annotated. Transduction methods are designed to harness vast amounts of unlabelled data to improve the performance of a learner trained on limited amounts of labelled data.

Similarly, the problem of learning to automatically categorise objects on the web into an ontology or directory in Chapter 5 could alternatively be casted as a transduction problem. The editors of a web directory can only annotate a very small subset of web pages in the Internet. Evidently, we can utilize another billion of un-annotated web pages via a transductive learning approach when building

the automated categoriser.

6.2 Problem Definition

Given a labelled dataset of input-output data pairs $\{(x_i, y_i)\}_{i=1}^{m} \subseteq \mathcal{X} \times \mathcal{Y}$, and an unlabelled dataset $\{x_i\}_{i=1}^{m'} \subseteq \mathcal{X}$. We would like to infer the outputs $\{y_i\}_{i=1}^{m'} \subseteq \mathcal{Y}$ of the unlabelled dataset by making use of both the labelled and unlabelled datasets. Note that, when we try to infer a function $f : \mathcal{X} \to \mathcal{Y}$ that will be able to predict the outputs of not previously seen data points instead of only the outputs of the given unlabelled dataset, the problem is called a semi-supervise learning. We focus on a transductive setting while having the function f as the byproduct of our model.

6.3 The Model

Transduction relies on the fundamental assumption that training and test data should exhibit similar behaviour. For instance, in large margin classification a popular concept is to assume that both training and test data should be separable with a large margin (Gammerman et al., 1998). A similar matching assumption is made by Joachims (1999); Zien et al. (2007) in requiring that class means are balanced between training and test set. Corresponding distributional assumptions are made for classification by Gärtner et al. (2006), for regression by Le et al. (2006), and in the context of sufficient statistics on the marginal polytope by Druck et al. (2008); Graça et al. (2007).

Such matching assumptions are well founded: after all, we assume that both training data $X = \{x_1, \ldots, x_m\}$ and test data $X' := \{x'_1, \ldots, x'_{m'}\}$ are drawn independently and identically distributed from the same distribution $p(x)$ on a domain $\mathcal{X}$. It therefore follows that for any function (or set of functions) $f : \mathcal{X} \to \mathbb{R}$ the distribution of $f(x)$ where $x \sim p(x)$ should also behave in the same way on both training and test set. Note that this is not automatically true if we get to choose f after seeing X and X'.

Rather than indirectly incorporating distributional similarity, e.g. by a large margin heuristic, we cast this goal as a two-sample problem which will allow us to draw on a rich body of literature for comparing distributions. One advantage of our setting is its full generality. That is, it is applicable to a variety of estimation problems: not only classification problems but also regression and even structured

estimation without much need for customisation.

At its heart it uses the following: rather than minimising only the empirical risk, regularised risk, log-posterior, or related quantities obtained only on the training set, we will add a divergence term characterising the *mismatch* in distributions between training and test set. We show that the kernel-based distance between distribution (Section 2.3.1) is a suitable quantity for this purpose. Moreover, we show that for certain choices of kernels we are able to recover a number of existing transduction constraints as a special case.

Note that our setting is entirely complementary to the notion of modifying the function space due to the availability of additional data. The latter stream of research led to the use of graph kernels and similar density-related algorithms (Chapelle et al., 2006). It is often referred to as the cluster assumption in semi-supervised learning. In other words, both methods can be combined as needed. That said, while distribution matching always holds thus making our method always applicable, it is not entirely clear whether the cluster assumption is always satisfied (e.g. assume a noisy classification problem).

Distribution matching, however, comes with a nontrivial price: the objective of the optimisation problem ceases to be convex except for rather special cases (which correspond to algorithms that have been proposed as previous work). While this is a downside, it is a property inherent in most transduction algorithms — after all, we are dealing with algorithms to obtain self-consistent labellings, predictions, or regressions on the data and there may exist more than one potential solution.

6.3.1 Supervised Learning

Denote by $\mathcal{X}$ and $\mathcal{Y}$ the domains of data and labels and let $\Pr(x, y)$ be a distribution on $\mathcal{X} \times \mathcal{Y}$ from which we are drawing observations. Moreover, denote by X, Y sets of data and labels of the training set and by X', Y' test data and labels respectively. Recall in Section 2.1.5, when designing an estimator one attempts to minimise some regularised risk functional

$$R_{\text{reg}}[f, X, Y] := \frac{1}{m} \sum_{i=1}^{m} l(x_i, y_i, f) + \lambda \Omega[f] \tag{6.1}$$

or alternatively (in a Bayesian setting) one deals with a log-posterior probability

$$\log p(f|X, Y) = \sum_{i=1}^{m} \log p(y_i|x_i, f) + \log p(f) + \text{const.} \tag{6.2}$$

Here $p(f)$ is the prior of the parameter choice f and $p(y_i|x_i, f)$ denotes the likelihood. f typically is a mapping $\mathcal{X} \to \mathbb{R}$ (for scalar problems such as regression or classification) or $\mathcal{X} \to \mathbb{R}^d$ (for multivariate problems such as named entity tagging, image annotation, matching, ranking, or more generally the clique potentials of graphical models). Note that we are free to choose f from one of many function classes such as decision trees, neural networks, or (nonparametric) linear models. The specific choice boils down to the ability to control the complexity of f efficiently, to one's prior knowledge of what constitutes a simple function, to runtime constraints, and to the availability of scalable algorithms. In general, we will denote the training-data dependent term by

$$R_{\text{train}}[f, X, Y] \tag{6.3}$$

and we assume that finding some f for which $R_{\text{train}}[f, X, Y]$ is small is desirable.

6.3.2 Distribution Matching

Denote by $f(X) := \{f(x_1), \ldots, f(x_m)\}$ and by $f(X') := \{f(x'_1), \ldots, f(x'_{m'})\}$ the applications of our estimator (and any related quantities) to training and test set respectively. For f chosen a-priori, the distributions from which $f(X)$ and $f(X')$ are drawn coincide. Clearly, this should also hold whenever f is chosen by an estimation process. After all, we want that the empirical risk on the training and test sets match. While this cannot be checked directly, we can at least check closeness between the distributions of $f(x)$. This reasoning leads us to the following additional term for the objective function of a transduction problem:

$$D(f(X), f(X')) \tag{6.4}$$

Here $D(f(X), f(X'))$ denotes the distance between the two distributions $f(X)$ and $f(X')$, and $f(X)$ denotes the application of f to the random training variable $x \sim p(x)$. This leads to an overall objective for learning

$$R_{\text{train}}[f, X, Y] + \gamma D(f(X), f(X')) \text{ for some } \gamma > 0 \tag{6.5}$$

when performing transductive inference. For instance, we could use the Kolmogorov-Smirnov statistic between both sets as our criterion, that is, we could use

$$D(f(X), f(X')) = \|F(f(X)) - F(f(X'))\|_\infty \tag{6.6}$$

the L_∞ norm between the cumulative distribution functions F associated with the empirical distributions $f(X)$ and $f(X')$ to quantify the differences between

both distributions. The problem with the above choice of distance is that it is not easily computable: we first need to evaluate f on both X and X', then sort the arguments, and finally compute the largest deviation between both sets before we can even attempt computing gradients or using a similar optimisation procedure. Such a choice is clearly computationally undesirable.

Instead, we propose to use kernel-based distance between distribution (Section 2.3.1). It is possible to design online estimates of the distance quantity which can be used for fast two-sample tests between $\mu[X]$ and $\mu[X']$. Details on how this can be achieved are deferred to Section 6.5.

6.4 Special Cases

Before discussing a specific algorithm let us consider a number of special cases to show that this basic idea is rather common in the literature (albeit not as explicit as in the present chapter).

6.4.1 Mean Matching for Classification

Joachims (1999) uses the following balancing constraint in the objective function of a binary classifier where $\hat{y}(x) = \operatorname{sgn}(f(x))$ for $f(x) = \langle w, x\rangle$. In order to balance the outputs between training and test set, Joachims (1999) imposes the linear constraint

$$\frac{1}{m}\sum_{i=1}^{m} f(x_i) = \frac{1}{m'}\sum_{i=1}^{m'} f(x'_i). \tag{6.7}$$

Assuming a linear kernel k on $\mathbb{R}$ this constraint is equivalent to requiring that

$$\mu[f(X)] = \frac{1}{m}\sum_{i=1}^{m} \langle f(x_i), \cdot\rangle = \frac{1}{m'}\sum_{i=1}^{m'} \langle f(x'_i), \cdot\rangle = \mu[f(X')]. \tag{6.8}$$

Note that Joachims (1999) uses the margin distribution as an additional criterion which will be discussed later.

This setting can be extended to multiclass categorisation and estimation with structured random variables in a straightforward fashion (Zien et al., 2007) simply by requiring a constraint corresponding to (6.8) to be satisfied for all possible values of y via

$$\frac{1}{m}\sum_{i=1}^{m} \langle f(x_i, y), \cdot\rangle = \frac{1}{m'}\sum_{i=1}^{m'} \langle f(x'_i, y), \cdot\rangle \text{ for all } y \in \mathcal{Y}. \tag{6.9}$$

This is equivalent to a linear kernel on $\mathbb{R}^{\mathcal{Y}}$ and the requirement that the distributions of the values $f(x, y)$ match for all y.

6.4.2 Distribution Matching for Classification

Gärtner et al. (2006) propose to perform transduction by requiring that the conditional class probabilities on training and test set match. That is, for classifiers generating a distribution of the form $y'_i \sim p(y'_i|x'_i, w)$ they require that the marginal class probability on the test set matches the empirical class probability on the training set. Again, this can be cast in terms of distribution matching via

$$\mu[g \circ f(X)] = \frac{1}{m}\sum_{i=1}^{m} \langle g \circ f(x_i), \cdot \rangle = \frac{1}{m'}\sum_{i=1}^{m'} \langle g \circ f(x'_i), \cdot \rangle = \mu[g \circ f(X')]$$

Here $g(\chi) = \frac{1}{1+e^{-\chi}}$ denotes the likelihood of $y = 1$ in logistic regression for the model $p(y|\chi) = \frac{1}{1+e^{-y\chi}}$. Note that instead of choosing the logistic transform g we could have picked a large number of other transformations. Indeed, we may strengthen the requirement above to hold for all g in some given function class $\mathcal{G}$ as follows:

$$D(f(X), f(X')) := \sup_{g \in \mathcal{G}} \left[\frac{1}{m}\sum_{i=1}^{m} g \circ f(x_i) - \frac{1}{m'}\sum_{i=1}^{m'} g \circ f(x'_i) \right] \tag{6.10}$$

If we restrict ourselves to g having bounded norm in a Reproducing Kernel Hilbert Space we obtain exactly the criterion (2.26). Gretton et al. (2008) show by duality that this is equivalent to the distance proposed in (6.10). In other words, generalising distribution matching to apply to transforms other than the logistic leads us directly to our new transduction criterion.

6.4.3 Distribution Matching for Regression

A similar idea for transduction was proposed by Le et al. (2006) in the context of regression: requiring that both means and predictive variances of the estimate agree between training and test set. For a heteroscedastic regression estimate this constraint between training and test set is met simply by ensuring that the distributions over first and second order moments of a Gaussian exponential family distribution match. The same goal can be achieved by using a polynomial kernel of second degree on the estimates, which shows that regression transduction can be viewed as a special case.

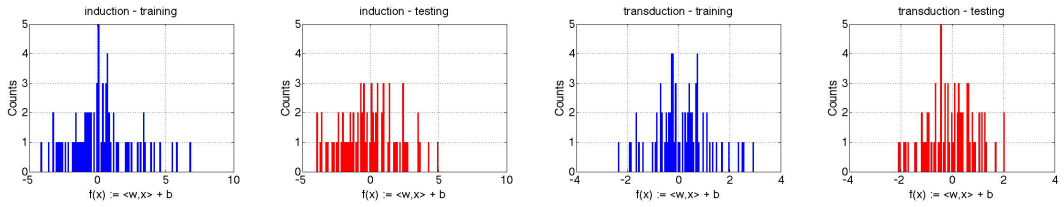


Figure 6.1: Score distribution of $f(x) = \langle w, x \rangle + b$ on the 'iris' toy dataset. From left to right: induction scores on the training set; test set; transduction scores on the training set; test set; Note that while the margin distributions on training and test set are very different for induction, the ones for transduction match rather well. It results in a 10% reduction of the misclassification error.

6.4.4 Large Margin Hypothesis

A key assumption in transduction is that a good hypothesis is characterised by a large margin of separation on both training and test set. Typically, the latter is enforced by some nonconvex function, e.g. of the form $\max(0, 1 - |f(x)|)$, thus leading to a nonconvex optimisation problem. Generalisations of this approach to multiclass and structured estimation settings is not entirely trivial and requires a number of heuristic choices (e.g. how to define the equivalent of the hat function $\max(0, 1 - |\chi|)$ that is commonly used in binary transduction).

Instead, if we require that the distribution of values $f(x, \cdot)$ on X' match those on X, we automatically obtain a loss function which enforces the large margin hypothesis whenever it is actually achievable on the training set. After all, assume that $f(X)$ exhibits a large margin of separation whereas $f(X')$ does not. In this case, $D(f(X), f(X'))$ is large and we obtain better risk minimisers by minimising the discrepancy of the distributions. The key point is that by using a two-sample criterion it is possible to obtain such criteria automatically without the need for heuristic choices. See Figure 6.1 for illustrations of this idea.

6.5 Algorithm

Streaming Approximation In general, minimising $D(f(X), f(X'))$ is computationally infeasible since the estimation of the distributional distance requires access to $f(X)$ and $f(X')$ rather than evaluations on a small sample. However, for Hilbert-Space based distance measures it is possible to find an online estimate

of D as follows:

$$D(p,p') := \|\mu[p] - \mu[p']\|^2 = \left\|\mathbf{E}_{x\sim p(x)}[k(x,\cdot)] - \mathbf{E}_{x'\sim p'(x')}[k(x',\cdot)]\right\| \tag{6.11}$$

$$= \mathbf{E}_{x,\tilde{x}\sim p}\mathbf{E}_{x',\tilde{x}'\sim p'}[k(x,\tilde{x}) - k(x,\tilde{x}') - k(\tilde{x},x') + k(x',\tilde{x}')] \tag{6.12}$$

The symbol $\tilde{(.)}$ denotes a second set of observations drawn from the same distribution. Note that (6.12) decomposes into a sum over 4 kernel functions, each of which takes as arguments a pair of instances drawn from p and p' respectively. Hence we can find an unbiased estimate via

$$\hat{D} := \frac{1}{m}\sum_{i=1}^{m} D_i \text{ where}$$

$$D_i := [k(f(x_i), f(x_{i+1})) - k(f(x_i), f(x'_{i+1})) - k(f(x_{i+1}), f(x'_i)) + k(f(x'_i), f(x'_{i+1}))] \tag{6.13}$$

under the assumption that X and X' contain i.i.d. data. Note that the assumption automatically fails if there is sequential dependence within the sets X or X' (e.g. we see all positive labels before we see the negative ones). In this case it is necessary to randomise X and X'.

Stochastic Gradient Descent The fact that the estimator of the distance $\hat{D}$ decomposes into an average over a function of pairs from the training and test set respectively means that we can use D_i as a stochastic approximation. Applying the same reasoning to the loss function in the regularised risk (6.1) we obtain the following loss

$$\begin{aligned} & \bar{l}(x_i, x_{i+1}, y_i, y_{i+1}, x'_i, x'_{i+1}, f) \\ := \; & l(x_i, y_i, f) + l(x_{i+1}, y_{i+1}, f) + 2\lambda\Omega[f] + \\ & \gamma[k(f(x_i), f(x_{i+1})) - k(f(x_i), f(x'_{i+1})) - k(f(x_{i+1}), f(x'_i)) + k(f(x'_i), f(x'_{i+1}))] \end{aligned} \tag{6.14}$$

as a stochastic estimate of the objective function defined in (6.5). This suggests Algorithm 6. Note that at no time we need to store past data even for computing the distance between both distributions.

Remark: The streaming formulation does not impose any in-principle limitation regarding matching sample sizes. The only difference is that in the unmatched case we want to give samples from both distributions different weights (1/m and 1/m' respectively), e.g. by modifying the sampling procedure (see Table 3, Section 6.7).

Algorithm 6 Transduction via Distribution Matching

Input: Convex set A, objective function $\bar{l}$
Initialize $w = 0$
for $t = 1$ to N **do**
  Sample $(x_i, y_i), (x_{i+1}, y_{i+1}) \sim p(x, y)$ and $x'_i, x'_{i+1} \sim p(x)$
  Update $w \leftarrow w - \eta_t \partial_w \bar{l}(x_i, x_{i+1}, y_i, y_{i+1}, x'_i, x'_{i+1}, f)$ where $f(x) = \langle \phi(x), w \rangle$
  Project w onto A via $w \leftarrow \operatorname{argmin}_{\bar{w} \in A} \|w - \bar{w}\|$.
end for

Concave Convex Procedure Alternatively, the Concave Convex Procedure (CCCP) (Section 2.6.1) can be used to find an approximate solution of the problem in (6.5) by solving a succession of convex programs. CCCP has been used extensively in almost any other transductive algorithms to deal with non-convexity of the objective function.

In order to minimise an additively decomposable objective function as in our transductive estimation, we could use stochastic gradient descent on the convex upper bound. Note that here the convex upper bound is given by a sum over the convex upper bounds for all terms. This strategy, however, is deficient in a significant aspect: the convex upper bounds on each of the loss terms become increasingly loose as we move f away from the current point of approximation. It would be considerably better if we updated the upper bound after every stochastic gradient descent step. This variant, however, is identical to stochastic gradient descent on the original objective function due to the following:

$$\partial_x F(x)|_{x=x_0} = \partial_x \bar{F}(x, x_0)|_{x=x_0} = \partial_x G(x)|_{x=x_0} - \partial_x H(x)|_{x=x_0} \text{ for all } x_0. \quad (6.15)$$

In other words, in order to compute the gradient of the upper bound we need not compute the upper bound itself. Instead we may use the nonconvex objective directly, hence we did not pursue CCCP approach and Algorithm 6 applies.

6.6 Related Work

The concept of distribution matching has also been exploited in covariate shift or domain adaptation setting (Huang et al., 2007; Bickel et al., 2007; Sugiyama et al., 2008; Nguyen et al., 2008; Kanamori et al., 2009). Domain adaptation deals with the problem when data distribution in the test (target) domain is different from the one in the training (source) domain. If we consider labelled instances of the

source domain as the labelled training data and unlabelled instances of the target domain as the unlabelled test data, we end up at the transductive setting. We could emphasize subtle differences between transductive and domain adaptation settings, for example, the amount of labelled data in transductive learning is small but large in domain adaptation and the amount of unlabelled data in transductive learning could be arbitrarily large. Noting the closeness between transductive and domain adaptation learning settings, it is not surprising that there has been some work extending transductive learning methods for domain adaptation and vice versa.

6.7 Experiments

To demonstrate the applicability of our approach, we apply transduction to binary and multiclass classification both on toy datasets from the UCI repository [1] and the LibSVM site [2], plus a larger scale multi-category classification dataset with $3.2 \cdot 10^6$ observations. We also perform experiments on a structured estimation problem, i.e. Japanese named entity recognition task and CoNLL-2000 base NP chunking task.

Algorithms Since we are not aware of other transductive algorithms which can be applied easily to all the problems we consider, we choose problem-specific transduction algorithms as competitors. Multi Switch Transductive SVM (**MultiSwitch**) is used for binary classification (Sindhwani & Keerthi, 2006). This method is a variant of transductive SVM algorithm (Joachims, 1999) tailored for linear semi-supervised binary classification on large and sparse datasets and involves switching of more than a single pair of labels at a time. For multiclass categorisation we pick a Gaussian processes based transductive algorithm with distribution matching term (**GPDistMatch**) (Gärtner et al., 2006).

We use stochastic gradient descent for optimisation in both inductive and transductive settings for binary and multiclass losses. More specifically, for transduction we use the Gaussian RBF kernel to compare distributions in (6.13). Note that, in the multiclass case, the additional distribution matching term measures the distance between multivariate functions.

[1] http://archive.ics.uci.edu/ml/

[2] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/

Small Scale Experiments We used the following datasets: binary (breast-cancer, derm, optdigits, wdbc, ionosphere, iris, specft, pageblock, tae, heart, splice, adult, australian, bupa, cmc, german, pima, tic, yeast, sonar, cleveland, svmguide3 and musk) from the UCI repository and multiclass (usps, satimage, segment, svmguide2, vehicle). The data was preprocessed to have zero mean and unit variance.

Since we anticipate the relevant length scale in the margin distribution to be in the order of 1 (after all, we use a loss function, i.e. a hinge loss, which uses a margin of 1) we pick a Gaussian RBF kernel width of 0.2 for binary classification. Moreover, to take scaling in the number of classes into account we choose a kernel width of $0.1\sqrt{c}$ for multicategory classification. Here c denotes the number of classes. We could indeed vary this width but we note in our experiments that the proposed method is not sensitive to this kernel width.

We split data equally into training and test sets, performing model selection on the training set and assessing performance on the test set. In these small scale experiments, we tune hyperparameters via 5-fold cross validation on the entire training set. The whole procedure was then repeated 5 times to obtain confidence bounds. More specifically, in the model selection stage, for transduction we adjust the regularisation λ and the transductive weight term γ (obviously, for inductive inference we only need to adjust λ). For MultiSwitch Transduction the positive class fraction of unlabelled data was estimated using the training set (Sindhwani & Keerthi, 2006). Likewise, the two associated regularisation parameters were tuned on the training set. For GP transduction both the regularisation and divergence parameters were adjusted.

Results The experimental results are summarised in Figure 6.2 for a binary setting and in Table 6.1 for a multiclass problem. In 23 binary datasets, transduction outperforms the inductive setup in 20 of them. Arguably, our proposed transductive method performs on a par with state-of-the-art transductive approach for each learning problem. In the binary estimation, out of 23 datasets, our method performs significantly worse than MultiSwitch transduction algorithm in 4 datasets (adult, bupa, pima, and svmguide3) and significantly better on 2 datasets (ionosphere and pageblock), using a one-sided paired t-test with 95% confidence. Overall, both algorithms are very comparable. The advantage of our approach is that it is '*plug and play*', i.e. for different problems we only need to use the appropriate supervised loss function. The distribution matching penalty itself remains unchanged. Further, by casting the transductive solution

as an online optimisation method, our approach *scales well*.

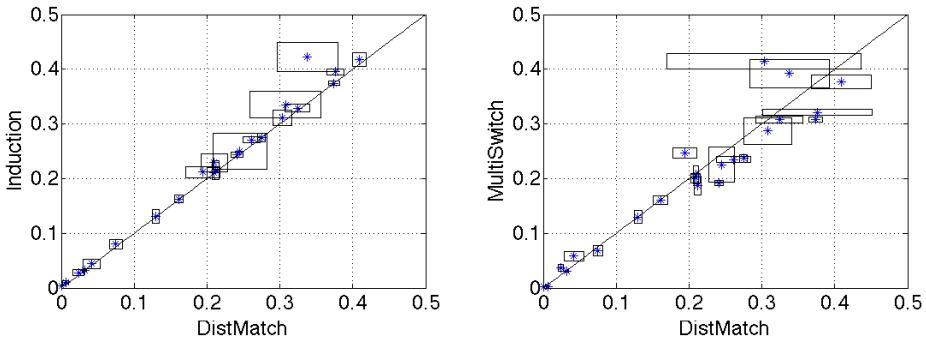


Figure 6.2: Error rate on 23 *binary* estimation problems. Left panel, DistMatch against Induction; Right panel, DistMatch against MultiSwitch. `DistMatch:` distribution matching (ours) and `MultiSwitch:` Multi switch transductive SVM, (Sindhwani & Keerthi, 2006). Height of the box encodes standard error of DistMatch and width of the box encodes standard error of Induction / MultiSwitch.

Larger Scale Experiments Since one of the key points of our approach is that it can be applied to large problems, we performed transduction on the DMOZ ontology [1] of topics. We selected the top 2 levels of the topic tree (575) and removed all but the 100 most frequent ones, since a large number of topics occurs only very rarely. This left us with 89.2% of the initial webpages. As feature vectors we used the standard bag of words representation of the web page descriptions with TF-IDF weighting. The dictionary size (and therefore the dimensionality of our features) is 1,319,489. For these larger scale experiments, we use a dataset of up to $3.2 \cdot 10^6$ observations. To our knowledge, our proposed transduction method is the only one that scales very well due to the stochastic approximation.

For each experiment, we split data into training and test sets. Model selection is perform on the training set by putting aside part of the training data as a validation set which is then used exclusively for tuning the hyperparameters. In large scale transduction two issues matter: firstly, the algorithm needs to be scalable with respect to the training set size. Secondly, we need to be able to scale the algorithm with respect to the test set. Both results can be seen in Tables 6.2 and 6.3. Note that Table 6.2 uses an equal split between training and test sets, while Table 6.3 uses an unequal split where the test set has many more observations. We see that the algorithm improves with increasing data size, both

[1] `http://www.dmoz.org/`

Table 6.1: Error rate ± standard deviation on a *multi-category* estimation problem. `DistMatch:` distribution matching (ours) and `GPDistMatch:` Gaussian Process transduction, (Gärtner et al., 2006).

dataset	m	classes	Induction	DistMatch	GPDistMatch
usps	730	10	0.143±0.021	0.125±0.019	0.140±0.034
satimage	620	6	0.190±0.052	0.186±0.037	0.212±0.034
segment	693	7	0.279±0.090	0.206±0.047	0.181±0.020
svmguide2	391	3	0.280±0.028	0.256±0.020	0.231±0.018
vehicle	423	4	0.385±0.070	0.333±0.048	0.336±0.060

Table 6.2: Error rate on the *DMOZ ontology* for increasing training / test set sizes.

training / test set size	50,000	100,000	200,000	400,000	800,000	1,600,000
induction	0.365	0.362	0.337	0.299	0.300	0.268
transduction	0.344	0.326	0.330	0.288	0.263	0.250

Table 6.3: Error rate on the *DMOZ ontology* for fixed training set size of 100,000 samples.

test set size	100,000	200,000	400,000	800,000	1,600,000
induction	0.358	0.358	0.357	0.357	0.357
transduction	0.326	0.316	0.306	0.322	0.329

Table 6.4: Accuracy, precision, recall and $F_{\beta=1}$ score on the *Japanese named entity* task.

	Accuracy	Precision	Recall	F1 Score
induction	96.82	84.15	72.49	77.89
transduction	97.13	84.46	75.30	79.62

Table 6.5: Accuracy, precision, recall and $F_{\beta=1}$ score on the *CoNLL-2000 base NP chunking* task.

	Accuracy	Precision	Recall	F1 Score
induction	95.72	90.99	90.72	90.85
transduction	96.05	91.73	91.97	91.85

for training and test sets. In the latter case, only up to some point: for the larger test sets (800,000 and 1,600,000) it decreases (although still stays better than inductive's). We suspect that a location-dependent transduction score would be useful in this context – i.e. instead of only minimising the discrepancy between decision function values on training and test set $D(f(X), f(X'))$ we could also introduce local features $D((X, f(X)), (X', f(X')))$.

Japanese Named Entity Recognition Experiments A key advantage of our transduction algorithm is it can be applied to structured estimation without modification. We used the Japanese named-entity recognition dataset provided with the CRF++ toolkit [1]. The data contains 716 Japanese sentences with 17 annotated named entities. The task is to detect and classify proper nouns and numerical information in a document into categories such as names of persons, organizations, locations, times and quantities. Conditional random fields (CRFs) (Lafferty et al., 2001) are considered to be the state-of-the-art framework for this sequential labelling problem (McCallum & Li, 2003).

As the basis of our implementation we used Leon Bottou's CRF code [2]. We use simple 1D chain CRFs with first order Markov dependency between name tags. That is, we have clique potentials joining adjacent labels (y_i, y_{i+1}), but which are independent of the text itself, and clique potentials joining words and labels (x_i, y_i). Since the former do not depend on the test data there is no need to enforce distribution matching. For the latter, though, we want to enforce that clique potentials are distributed in the same way between training and test set. The stationarity assumption in the potentials implies that this needs to hold uniformly over all such cliques.

Since the number of tokens per sentence is variable, i.e. the chain length itself is a random variable, we perform distribution matching on a per-token basis — we oversample each token 10 times in our experiments. This strikes a balance between statistical accuracy and computational efficiency. The additional distribution matching term is then measuring the distance between these over-sampled clique potentials. As before, we split data equally into training and test sets and put aside part of the training data as a validation set which is used exclusively for tuning the hyperparameters. We relied on the feature template provided in CRF++ for this task. We report results in Table 6.4, that is precision (fraction of name tags which match the reference tags), recall (fraction of reference tags re-

[1] http://crfpp.sourceforge.net/

[2] http://leon.bottou.org/projects/sgd

turned), and their harmonic mean, $F_{\beta=1}$ are reported. Transduction outperforms induction in all metrics.

CoNLL-2000 Base NP Chunking Experiments Our second structured estimation experiment is the CoNLL-2000 base NP chunking dataset (Sang & Buchholz, 2000) as provided in the CRF++ toolkit. The task is to divide text into syntactically correlated parts. The dataset has 900 sentences and the goal is to label each word with a label indicating whether the word is outside a chunk, starts a chunk, or continues a chunk.

Similarly to Japanese named entity recognition task, 1D chain CRFs with only first order Markov dependency between chunk tags are modelled. We considered binary-valued features which depend on the words, part-of-speech tags, and labels in the neighbourhood of a given word as encoded in the CRF++ feature template. The same experimental setup as in named entity experiments is used. The results in terms of accuracy, precision, recall and F1 score are summarised in Table 6.5. Again, transduction outperforms the inductive setup.

6.8 Conclusion

We proposed a transductive estimation algorithm which is a) simple, b) general c) scalable and d) works well when compared to the state of the art algorithms applied to each specific problem. Not only is it useful for classical binary and multiclass categorisation problems but it also applies to ontologies and structured estimation problems. It is not surprising that it performs well comparably to existing algorithms, since they can, in many cases, be seen as special instances of the general purpose distribution matching setting.

Extensions of distribution matching beyond simply modelling $f(X)$ and instead, modelling $(X, f(X))$, that is, the introduction of local features, obtaining good theoretical bounds on the shrinkage of the function class via the distribution matching constraint, and applications to other function classes (e.g. balancing decision trees) are subject of future research.

Chapter 7

Optimal Tiering as a Flow Problem

In this chapter, we present a scalable algorithm for performing storage and indexing management in the context of webpage tiering. The goal is to allocate pages to caches such that the most frequently accessed pages reside in the caches with the smallest latency whereas the least frequently retrieved pages are stored in the backtiers of the caching system. This indexing and storage problem is closely linked with a larger class of parametric flow problems, as we illustrate below.

7.1 Motivating Examples

Parametric flow problems have been well-studied in operations research (Gusfield & Tardos, 1994). It has received a significant amount of contributions and has been applied in many problem areas such as database record segmentation (Eisner & Severance, 1976), energy minimisation for computer vision (Kolmogorov et al., 2007), critical load factor determination in two-processor systems (Stone, 1978), end-of-session baseball elimination (Gusfield & Martel, 1992), and most recently by Zhang et al. (2004, 2005a,b) in product portfolio selection. In other words, it is a key technique for many estimation and assignment problems. Unfortunately many algorithms proposed in the literature are geared towards thousands to millions of objects rather than billions, as is common in Internet-scale problems.

Our motivation for solving parametric flow is the problem of webpage tiering for search engine indices. While our methods are general and could be applied to a range of other machine learning and optimisation problems, we focus on webpage tiering as the illustrative example in this chapter. The rationale for choosing this

application is threefold: firstly, it is a real problem in search engines. Secondly, it provides very large datasets. Thirdly, in doing so we introduce a new problem to the machine learning community. That said, our approach would also be readily applicable to very large scale versions of the problems described in Eisner & Severance (1976); Stone (1978); Gusfield & Martel (1992); Zhang et al. (2004).

The specific problem that will provide our running example is that of assigning webpages to several tiers of a search engine cache such that the time to serve a query is minimised. For a given query, a search engine returns a number of documents (typically 10). The time it takes to serve a query depends on where the documents are located. The first tier (or cache) is the fastest (using premium hardware, etc. thus also often the smallest) and retrieves its documents with little latency. If even just a single document is located in a back tier, the delay is considerably increased since now we need to search the larger (and slower) tiers until the desired document is found. Hence it is our goal to assign the most popular documents to the fastest tiers while taking the interactions between documents into account.

7.2 Problem Definition

We would like to allocate documents $d \in D$ into k tiers of storage at our disposal. Moreover, let $q \in Q$ be the queries arriving at a search engine, with finite values $v_q > 0$ (e.g. the probability of the query, possibly weighted by the relevance of the retrieved results), and a set of documents D_q retrieved for the query. This input structure is stored in a bipartite graph G with vertices $V = D \cup Q$ and edges $(d, q) \in E$ whenever document d should be retrieved for query q. In the following we assume that there is a *unique* list of r pages that are retrieved for a query. This is decidedly *not* true in practice. Instead, we can expect to see some change in the results for a given query, due to localization, user customisation, browser capabilities, a change in page relevance, different versions of the search engine, among others. Such a case is not a problem. All that is required is to treat each instance of the query that returns a *different* result set as if it were a different query. Since our algorithm does not require the actual query ID this can be accomplished easily.

The k tiers, with tier 1 as the most desirable and k the least (most costly for retrieval), form an increasing sequence of *cummulative* capacities C_t, with C_t indicating how many pages can be stored by tiers $t' \leq t$ together. Without

loss of generality, assume $C_{k-1} < |D|$ (that is, the last tier is required to hold all documents, or the problem can be reduced). Finally, for each $t \geq 2$ we assume that there is a penalty $p_{t-1} > 0$ incurred by a tier-miss at level t (known as "fallthrough" from tier $t-1$ to tier t). And since we have to access tier 1 regardless, we set $p_0 = 0$ for convenience. For instance, retrieving a page in tier 3 incurs a total penalty of $p_1 + p_2$.

7.2.1 Related Work

Optimisation of index structures and data storage is a key problem in building an efficient search engine. Much work has been invested into building efficient inverted indices which are optimised for query processing (Yan et al., 2009; Fagin, 1996). These papers all deal with the issue of optimizing the data *representation* for a given query and how an inverted index should be stored and managed for general queries. In particular, Fagin (1996); Persin et al. (1996) address the problem of computing the top-k results without scanning over the entire inverted lists. Recently, machine learning algorithms have been proposed (Goel et al., 2008) to improve the ordering within a given collection beyond the basic inverted indexing setup (Fagin, 1996).

A somewhat orthogonal strategy to this is to decompose the collection of webpages into a number of disjoint tiers (Risvik et al., 2003) ordered in decreasing level of relevance. That is, documents are partitioned according to their relevance for answering queries into different tiers of (typically) increasing size. This leads to putting the most frequently retrieved or the most relevant (according to the value of query, the market or other operational parameters) pages into the top tier with the smallest latency and relegating the less frequently retrieved or the less relevant pages into bottom tiers. Since queries are often carried out by *sequentially* searching this hierarchy of tiers, an improved ordering minimises latency, improves user satisfaction, and it reduces computation.

A naive implementation of this approach would simply assign a value to each page in the index and arrange them such that the most frequently accessed pages reside in the highest levels of the cache. Unfortunately this approach is suboptimal: in order to answer a given query well a search engine typically does not only return a *single* page as a result but rather returns a *list* of r (typically $r = 10$) pages. This means that if even just one of these pages is found at a much lower tier, we either need to search the backtiers to retrieve this page or alternatively we need to sacrifice result relevance.

At first glance, the problem is daunting: we need to take all correlations among pages induced by user queries into account. Moreover, for reasons of practicality we need to design an algorithm which is linear in the amount of data presented (i.e. the number of queries) and whose storage requirements are only linear in the number of pages. Finally, we would like to obtain guarantees in terms of performance for the assignment that we obtain from the algorithm. Our problem, even for $r = 2$, is closely related to the weighted k-densest subgraph problem, which is NP hard (Papadimitriou & Steiglitz, 1982).

7.2.2 Optimisation Problem

Since the problem we study is somewhat more general than the parametric flow problem we give a self-contained derivation of the problem and derive the more general version beyond Gusfield & Tardos (1994).

We denote the result set for query q by $D_q := \{d : (d, q) \in G\}$, and similarly, the set of queries seeking for a document d by $Q_d := \{q : (d, q) \in G\}$. For a document d we denote by $z_d \in \{1, \dots, k\}$ the tier storing d. Define

$$u_q := \max_{d \in D_q} z_d \tag{7.1}$$

as the number of cache levels we need to traverse to answer query q. In other words, it is the document found in the worst tier which determines the cost of access. Integrating the optimisation over u_q we may formulate the tiering problem as an integer program:

$$\text{minimise}_{z,u} \sum_{q \in Q} v_q \sum_{t=1}^{u_q - 1} p_t \tag{7.2a}$$

$$\text{subject to } z_d \leq u_q \leq k \text{ for all } (q, d) \in G \tag{7.2b}$$

$$\sum_{d \in D} \{z_d \leq t\} \leq C_t \ \forall \ t. \tag{7.2c}$$

Note that we replaced the maximisation condition (7.1) by a linear inequality in preparation for a reformulation as an integer linear program. Obviously, the optimal u_q for a given z will satisfy (7.1).

Lemma 44 *Assume that $C_k \geq |D| > C_{k-1}$. Then there exists an optimal solution of (7.2) such that $\sum_d \{z_d \leq t\} = C_t$ for all $1 \leq t < k$.*

Proof Assume that z^*, v^* is the optimal solution. Note that the objective function only depends on v^* directly. If the capacity constraint is not met with

equality we may decrease the tiers of an arbitrary set of pages until the constraints are met. Since this only relaxes the constraints on v^* further while not increasing the objective function, the solution is still optimal. ■

In the following we address several issues associated with the optimisation problem:

A) Eq. (7.2) is an *integer* program and consequently it is discrete and nonconvex. We show that there exists a convex reformulation of the problem.

B) It is at a formidable scale (often $|D| > 10^9$). Section 7.3.4 presents a stochastic gradient descent procedure to solve the problem in few passes through the database.

C) We have insufficient data for an accurate tier assignment for pages associated with tail queries. Section 7.5.2 addresses the problem by a smoothing estimator for the tier index of a page.

7.2.3 Integer Linear Program

We now replace the selector variables z_d and u_q by binary variables via a "thermometer" code. Let

$$x \in \{0;1\}^{D\times(k-1)} \text{ subject to } x_{dt} \geq x_{d,t+1} \text{ for all } d,t \tag{7.3a}$$
$$y \in \{0;1\}^{Q\times(k-1)} \text{ subject to } y_{qt} \geq y_{q,t+1} \text{ for all } q,t \tag{7.3b}$$

be index variables. Thus we have the one-to-one mapping $z_d = 1 + \sum_t x_{dt}$ and $x_{dt} = \{z_d > t\}$ between z and x. For instance, for $k = 5$, a middle tier $z = 3$ maps into $x = (1,1,0,0)$ (requiring two fallthroughs), and the best tier $z = 1$ corresponds to $x = (0,0,0,0)$. The mapping between u and y is analogous. The constraint $u_q \geq z_d$ can simply be rewritten coordinate-wise $y_{qt} \geq x_{dt}$.

Finally, the capacity constraints assume the form $\sum_d x_{dt} \geq |D| - C_t$. That is, the number of pages allocated to higher tiers are at least $|D| - C_t$. Define *remaining* capacities $\bar{C}_t := |D| - C_t$ and use the variable transformation (7.1) we

have the following integer linear program:

$$\underset{x,y}{\text{minimise}}\ v^\top yp \tag{7.4a}$$

$$\text{subject to } x_{dt} \geq x_{d,t+1} \text{ and } y_{qt} \geq y_{q,t+1} \text{ and } y_{qt} \geq x_{dt} \text{ for all } (q,d) \in G \tag{7.4b}$$

$$\textstyle\sum_d x_{dt} \geq \bar{C}_t \text{ for all } 1 \leq t \leq k-1 \tag{7.4c}$$

$$x \in \{0;1\}^{D\times(k-1)}; y \in \{0;1\}^{Q\times(k-1)} \tag{7.4d}$$

where $p = (p_1, \ldots, p_{k-1})^\top$ and $v = (v_1, \ldots, v_{|Q|})^\top$ are column vectors, and y a matrix (y_{qt}). The advantage of (7.4) is that while still discrete, we now have *linear* constraints and a *linear* objective function. The only problem is that the variables x and y need to be binary.

Lemma 45 *The solutions of (7.2) and (7.4) are equivalent.*

Proof Firstly, the variable sets (z,u) and (x,y) are equivalent (we have an explicit bijection). The same applies to the constraints between them — eq. (7.4b) implies that the retrieval tier for query q needs to be at least as high as that of the highest page. Finally, the objective function sums over all tier levels from 2 to k such that a document found at tier t will contribute via $p_2 + \ldots + p_t$. Hence equality holds. ■

7.2.4 Hardness

Before discussing convex relaxations and approximation algorithms it is worthwhile to review the hardness of the problem: consider only two tiers, and a case where we retrieve only *two* pages per query. The corresponding graph has vertices D and edges $(d,d') \in E$, whenever d and d' are displayed together to answer a query. In this case the tiering problem reduces to one of finding a subset of vertices $D' \subset D$ such that the induced subgraph has the largest number (possibly weighted) of edges subject to the capacity constraint $|D'| \leq C$.

For the case of k pages per query, simply assume that $k-2$ of the pages are always the same. Hence the problem of finding the best subset reduces to the case of 2 pages per query. This problem is identical to the k-densest subgraph problem which is known to be NP hard (Papadimitriou & Steiglitz, 1982).

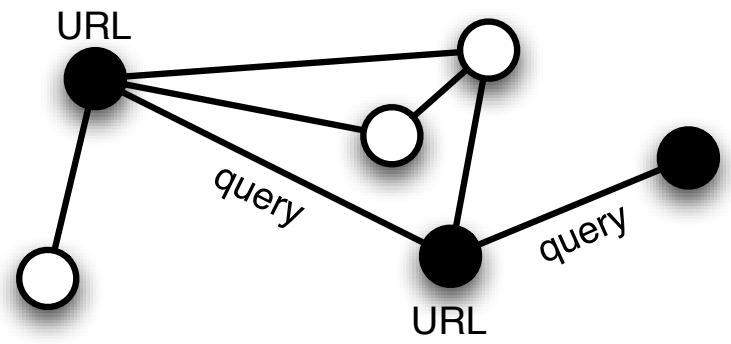


Figure 7.1: k-densest subgraph reduction. Vertices correspond to URLs and queries correspond to edges. Queries can be served whenever the corresponding URLs are in the cache. This is the case whenever the induced subgraph contains the edge.

7.3 The Model

The key idea in solving (7.4) is to relax the capacity constraints for the tiers. This renders the problem totally unimodular and therefore amenable to a solution by a linear program. We replace the capacity constraint by a partial Lagrangian. This does *not* ensure that we will be able to meet the capacity constraints *exactly* anymore. Instead, we will only be able to state ex-post that the relaxed solution is optimal for the *observed* capacity distribution. Moreover, we are still able to control capacity by a suitable choice of the associated Lagrange multipliers.

7.3.1 Linear Program

Instead of solving (7.4) we study the linear program:

$$\begin{aligned} \underset{x,y}{\text{minimise}}\ v^\top y p - \mathbf{1}^\top x \lambda \text{ subject to } x_{dt} \geq x_{d,t+1} \text{ and } y_{qt} \geq y_{q,t+1} \\ y_{qt} \geq x_{dt} \text{ for } (q,d) \in G \text{ and } x_{dt}, y_{qt} \in [0,1] \end{aligned} \tag{7.5}$$

Here $\lambda = (\lambda_1, \ldots, \lambda_{k-1})^\top$ act as Lagrange multipliers $\lambda_t \geq 0$ for enforcing capacity constraints and $\mathbf{1}$ denotes a column of $|D|$ ones. We now relate the solution of (7.5) to that of (7.4).

Lemma 46 *For any choice of λ with $\lambda_t \geq 0$ the linear program (7.5) has an integral solution, i.e. there exists some x^*, y^* satisfying $x^*_{dt}, y^*_{qt} \in \{0;1\}$ which minimise (7.5). Moreover, for $\bar{C}_t = \sum_d x^*_{dt}$ the solution (x^*, y^*) also solves (7.4).*

Proof We first show that (7.5) has an integral solution for all choices of λ. This holds since constraints are totally unimodular: the constraint matrix has only one 1 and one -1 entry per row. Integrality follows Heller & Tompkins (1956).

By construction, for the choice of $\bar{C}_t = \sum_d x^*_{dt}$ the condition (7.4c) is met with equality, hence the integral solution of (7.5) is also the solution of a linear program arising from a relaxation of the integer linear program (7.4) to a linear program. However, since the relaxation has an integral solution it follows that (x^*, y^*) are also optimal for (7.4). ■

We have succeeded in reducing the complexity of the problem to that of a linear program, yet it is still formidable and it needs to be solved to optimality for an accurate caching prescription. Moreover, we need to adjust λ such that we satisfy the desired capacity constraints (approximately).

Lemma 47 *Denote by $L^*(\lambda)$ the value of (7.5) at the solution of (7.5) and let $L(\lambda) := L^*(\lambda) + \sum_t \bar{C}_t \lambda_t$. Hence $L(\lambda)$ is concave in λ and moreover, $L(\lambda)$ is maximised for a choice of λ where the solution of (7.5) satisfies the constraints of (7.4).*

Proof Subtracting $\sum_t \bar{C}_t \lambda_t$ from the objective of (7.5) yields a reduced Lagrange function which enforces the constraint $\sum_d x_{dt} \geq \bar{C}_t$. As such, it is concave in λ and at its maximum the capacity constraint is satisfied. ■

Note that while the above two lemmas provide us with a guarantee that for every λ and for every associated integral solution of (7.5) there exists a set of capacity constraints for which this is optimal and that such a capacity satisfying constraint can be found efficiently by concave maximisation, they do not guarantee the converse: not every capacity constraint can be satisfied by the convex relaxation.

7.3.2 Graph Cut Equivalence

It is well known that the case of two tiers ($k = 2$) can be relaxed to a min-cut, max-flow problem (Gusfield & Tardos, 1994; Ford & Fulkerson, 1956). The transformation works by designing a bipartite graph between queries q and documents d. All documents are connected to the source s by edges with capacity λ and queries are connected to the sink t with capacity $(1 - v_q)$. Documents d retrieved for a query q are connected to q with capacity ∞.

Figure 7.2 provides an example of such a maximum-flow, minimum-cut graph from source s to sink t. The conversion to several tiers is slightly more involved.

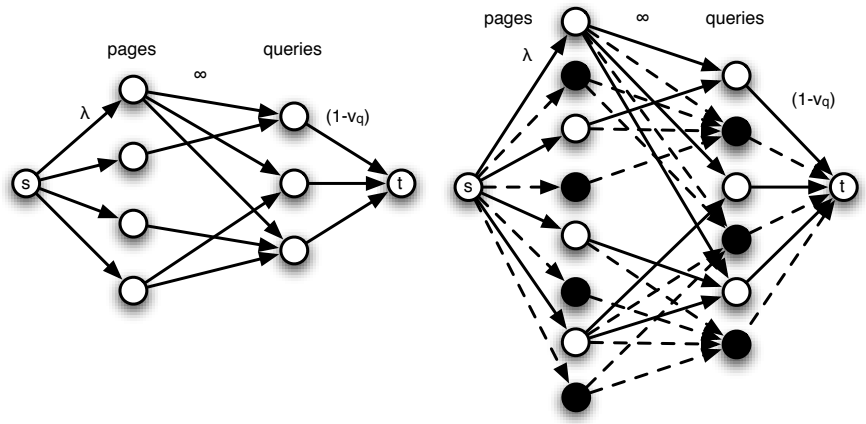


Figure 7.2: Left: maximum flow problem for a problem of 4 pages and 3 queries. The minimum cut of the directed graph needs to sever all pages leading to a query or alternatively it needs to sever the corresponding query incurring a penalty of $(1 - v_q)$. This is precisely the tiering objective function for the case of two tiers. Right: the same query graph for three tiers. Here the black nodes and dashed edges represent a copy of the original graph — additionally each page in the original graph also has an infinite-capacity link to the corresponding query in the additional graph.

Denote by v_{di} vertices associated with document d and tier i and moreover, denote by w_{qi} vertices associated with a query q and tier i. Then the graph is given by edges (s, v_{di}) with capacities λ_i; edges $(v_{di}, w_{qi'})$ for all (document, query) pairs and for all $i \leq i'$, endowed with infinite capacity; and edges (w_{qi}, t) with capacity $(1 - v_q)$.

As with the simple caching problem, we need to impose a cut on any query edge for which not all incoming page edges have been cut. The key difference is that in order to benefit from storing pages in a better tier we need to guarantee that the page is contained in the lower tier, too.

7.3.3 Variable Reduction

We now simplify the relaxed problem (7.5) further by reducing the number of variables, without sacrificing integrality of the solution. A first step is to substitute $y_{qt} = \max_{d \in D_q} x_{dt}$, to obtain an optimisation problem over the documents

alone:

$$\underset{x}{\text{minimise}}\ v^\top \Big(\max_{d\in D_q} x_{dt}\Big) p - 1^\top x\lambda \text{ subject to } x_{dt} \geq x_{dt'} \text{ for } t' > t \text{ and } x_{dt} \in [0,1] \tag{7.6}$$

Note that the monotonicity condition $y_{qt} \geq y_{qt'}$ for $t' > t$ is automatically inherited from that of x. The solution of (7.6) is still integral since the problem is equivalent to one with integral solution.

Lemma 48 *We may scale p_t and λ_t together by constants $\beta_t > 0$, such that $p'_t/p_t = \beta_t = \lambda'_t/\lambda_t$. The resulting solution of this new problem (7.6) with (p', λ') is unchanged.*

Proof We introduce Lagrange multipliers γ_{dt} due to constraints of the form $\sum_{t=1}^{k-2} \gamma_{dt}(x_{dt} - x_{d,t+1})$, which can be rewritten as $\sum_{t=1}^{k-1} \alpha_{dt} x_{dt}$. At optimality we know for a given (p, λ) that the gradient of (7.6) needs to match the Lagrange multipliers (α_{dt}). Denote by x^* and α^* the solution of the optimisation problem and the corresponding Lagrange multipliers. Rescaling λ and p as per assumption we see that by rescaling α the optimality conditions still hold. Hence x^* must also solve (7.5) for (p', λ'). ■

Essentially, problem (7.5) as parameterised by (p, λ) yields solutions which form equivalence classes. Consequently for the convenience of solving (7.5), we may assume $p'_t = 1$ for $t \geq 1$. We only need to consider the original p for evaluating the objective using solution z (thus, same observed capacities C_t).

Since (7.5) is a relaxation of (7.4) this reformulation can be extended to the integer linear program, too. Moreover, under reasonable conditions on the capacity constraints, there is more structure in λ.

Lemma 49 *Assume that $\bar{C}_t$ is monotonically decreasing and that $p_t = 1$ for $t \geq 1$. Then any choice of λ satisfying the capacity constraints is monotonically non-increasing.*

Proof If $\lambda_t = \lambda_{t+1}$ we arrive at a solution where $x_{dt} = x_{d,t+1}$ since in this case the functions concerning both variables are identical. Moreover, choosing $\lambda_{t+1} > \lambda_t$ can only lead to an increase in $x_{d,t+1}$. However, since $x_{dt} \geq x_{d,t+1}$ by constraint, this means that for any $\lambda_{t+1} \geq \lambda_t$ we have $x_{d,t+1} = x_{dt}$.

Then we may choose $\lambda'_t = \lambda'_{t+1} = \frac{\lambda_t + \lambda_{t+1}}{2}$ and obtain the same solution with a nonincreasing sequence of λ_t: it has the same value of the objective function and moreover the joint subgradients are identical since terms in λ_t and λ_{t+1} are

added. A recursive averaging procedure generates a nonincreasing sequence of equivalent values for λ which completes the proof. ■

One interpretation of this is that, unless the tiers are increasingly inexpensive, the optimal solution would assign pages in a fashion yielding empty middle tiers (the remaining capacities $\bar{C}_t$ not strictly decreasing). This monotonicity simplifies the problem. Consequently, we exploit this fact to complete the variable reduction.

Define $\delta\lambda_i := \lambda_i - \lambda_{i+1}$ for $i \geq 1$ (all non-negative by virtue of Lemma 49) and

$$f_\lambda(\chi) := -\lambda_1\chi + \sum_{i=1}^{k-2} \delta\lambda_i \max(0, i - \chi) \text{ for } \chi \in [0, \text{k-1}]. \tag{7.7}$$

Note that by construction $\partial_\chi f_\lambda(\chi) = -\lambda_i$ whenever $\chi \in (i-1, i)$. The function f_λ is clearly convex, which helps describe our tiering problem via the following convex program

$$\underset{z}{\text{minimise}}\, v^\top \left(\max_{d \in D_q} z_d\right) + \sum_d f_\lambda(z_d - 1) \text{ for } z_d \in [1, k] \tag{7.8}$$

We now use only one variable per document. Moreover, the convex constraints are simple box constraints. This simplifies convex projections, as needed for online programming.

Lemma 50 *The solution of (7.8) is equivalent to that of (7.5).*

Proof We use the following three properties:

(A) (7.8) is convex, has a unique minimum value.

(B) There is an injective mapping from any set of variables in (7.8) to the thermometer code of (7.5) with the property that the values of the objective function coincide in this case. From this it follows that the minimum of (7.8) cannot exceed the minimum of (7.5).

(C) For an integral set of variables in (7.5) there is an injective map to (7.8) such that, again, the objective functions coincide. From this it follows that the minimum of (7.5) cannot exceed the minimum of (7.8).

Combination of (B) and (C) proves the claim. ■

Algorithm 7 Tiering Optimisation

Initialize all $z_d = 0$
Initialize $n = 100$
for $i = 1$ **to** MAXITER **do**
  for all $q \in Q$ **do**
    $\eta = \frac{1}{\sqrt{n}}$ (learning rate)
    $n \leftarrow n + 1$ (increment counter)
    Update $z \leftarrow z - \eta \partial_x \ell_q(z)$
    Project z to $[1, k]^D$ via
    $z_d \leftarrow \max(1, \min(k, z_d))$
  end for
end for

7.3.4 Online Algorithm

We now turn our attention to a fast algorithm for minimising (7.8). While greatly simplified relative to (7.2) it still remains a problem of billions of variables. The key observation is that the objective function of (7.8) can be written as sum over the following loss functions

$$l_q(z) := v_q \max_{d \in D_q} z_d + \frac{1}{|Q|} \sum_d f_\lambda(z_d - 1) \tag{7.9}$$

where $|Q|$ denotes the cardinality of the query set. The transformation suggests a simple stochastic gradient descent optimisation algorithm: traverse the input stream by queries, and update the values of x_d of all those documents d that would need to move into the next tier in order to reduce service time for a query. Subsequently, perform a projection of the page vectors to the set $[1, k]$ to ensure that we do not assign pages to non-existent tiers.

Algorithm 7 proceeds by processing the input query-result records (q, v_q, D_q) as a stream comprising the set of pages that need to be displayed to answer a given query. More specifically, it updates the tier preferences of the pages that have the lowest tier scores for each level and it decrements the preferences for all other pages. We may apply results for online optimisation algorithms (Zinkevich, 2003) to show that a small number of passes through the dataset suffice.

Lemma 51 *The solution obtained by Algorithm 7 converges at rate $O(\sqrt{T})$ to its minimum value. Here T is the number of queries processed.*

Algorithm 8 Deferred updates

Observe current time n'
Read timestamp n for document d
Compute update steps $\delta = \delta(n', n)$
repeat
  $j = \lfloor z_d + 1 \rfloor$ (next largest tier)
  $t = (j - z_d)/\lambda_j$ (change needed to reach next tier)
  if $t > \delta$ **then**
    $\delta = 0$ and $z_d \leftarrow z_d + \lambda_j \delta$ (partial step; we are done)
  else
    $\delta \leftarrow \delta - t$ and $z_d \leftarrow z_d + 1$ (full step; next tier)
  end if
until $\delta = 0$ (no more updates) or $z_d = k - 1$ (bottom tier)

7.4 Practical Issues

7.4.1 Deferred and Approximate Updates

The naive implementation of algorithm 7 is infeasible as it would require us to update all $|D|$ coordinates of x_d for each query q. However, it is possible to defer the updates until we need to inspect z_d directly. The key idea is to exploit that for all z_d with $d \notin D_q$ the updates only depend on the value of z_d at update time and that f_λ is piecewise linear and monotonically decreasing. Assume that we updated z_d at iteration n and we revisit it at iteration n'. This means that z_d at iteration n' is given by applying gradients of $f_\lambda(z_d)$ repeatedly and by moving η in the negative gradient direction. We may compute the aggregate result of all steps by simply adding up the steplengths for each segment, rescaled by the slope λ_j. Denote by

$$s(n) := \sum_{j=1}^{n} \eta_j \text{ and let } \delta(n', n) := s(n') - s(n) \tag{7.10}$$

the aggregate steps lengths from time n to time n'. Note that λ_t^{-1} is the aggregate steplength required to cross the interval $[t-1, t]$. Algorithm 8 carries out the deferred updates by moving step by step down the slope of f_λ. This is required for invoking the gradient computation and update step of Algorithm 7.

While precomputing the steplength is a significant computational improvement, storing (7.10) is substantial: a billion steps translate into 4GB of data.

This can be remedied by an integral approximation

$$\delta(n', n) = \sum_{j=n+1}^{n'} \eta_j = \sum_{j=n+1}^{n'} \frac{1}{\sqrt{j+n_0}} \approx 2\left[\sqrt{n'+n_0} - \sqrt{n+n_0}\right]$$

which becomes increasingly accurate for large $|n' - n|$. It allows us to obtain values for $\delta(n', n')$ in constant time without any storage.

7.4.2 Data Reduction and Max/Sum Heuristics

The amount of data used in the optimisation problem can be reduced significantly by eliminating documents and queries which are definitely assigned to particular tiers.

Consider the case of only two tiers (we only have λ_1): any query occurring more frequently v_q than λ_1 will automatically ensure that the associated pages are cached. Consequently we may remove this query from the dataset, assign all related pages to the first tier $x_d = 0$ and *remove* them from all remaining queries. Secondly, any document d for which $\sum_{q \in Q_d} v_q$ is displayed less than λ_1 will definitely *not* be in the cache. Consequently all queries using d will by default fail and can be removed from the dataset. Note that this thresholding procedure can be repeated with the remaining (so far undetermined) documents and queries.

An analogous reasoning applies to multiple tiers: for any query q with weight $v_q \geq \lambda_t$ we know that all $d \in D_q$ will definitely be stored in tier t or lower — the subgradients with respect to z_d are at least v_q at this tier. Any document which, accumulated over all queries $q \in Q_d$ is not requested more than λ_t times cannot be displayed at t or higher. An appealing side-effect of this data reduction is that the gradients of the remaining functions l_q cover a much smaller dynamic range. This accelerates convergence (Nesterov & Vial, 2000) since optimisation progress inversely depends on the gradient range.

Furthermore, both $s_d := \sum_{q \in Q_d} v_q$ and $m_d := \max_{q \in Q_d} v_q$ are good tiering heuristics in their own right. If we had only one page per query the optimal solution would be to sort according to s_d. On the other hand, for large $|D_q|$ ordering documents according to m_d proves near optimal as we see on both synthetic and real data. This suggests a very simple heuristic for obtaining near-optimal tiering, namely to sort based on m_d. Empirically we found that a good initialization for the page variables z_d to be $-(0.9 \log m_d + 0.1 \log s_d)$ scaled and shifted to fit the $[1, k]$ range, which helps convergence.

7.5 Extensions

We describe three types of extensions on our proposed tiering approach: beyond hit and miss, smoothing and robustness. We will discuss those in turn.

7.5.1 Beyond Hit and Miss

So far we only discussed a rather primitive model of penalties per query, namely that we would incur a penalty $v_q p_t$ for not serving a query at level t. The motivation for this simplification was twofold — we were interested in finding the optimal tier arrangement for a given set of pages to be retrieved per query and moreover, we did not distinguish between the value of different pages or the possibility of retrieving only a partial set of results per query. In the following we show that considerably more sophisticated score functions still lead to integral solutions.

Lemma 52 *Denote by $\mathcal{S}$ a collection of sets, and by $\lambda_{St}, \gamma_{St} \geq 0$ and $\eta_S \in \mathbb{R}$ weighting coefficients. Then, the optimisation problem obtained by replacing $\sum_q v_q \max_{d:(d,q)\in G} z_d$ with*

$$\sum_{S\in\mathcal{S}} \max_{d\in S} \left[\eta_S z_d + \sum_t \lambda_{St} \max(0, t - z_d) + \gamma_{St} \max(0, z_d - t) \right]$$

has an integral solution.

Proof [sketch only] We treat each $S \in \mathcal{S}$ as if it were a query of its own with documents $d \in S$ associated with it. Within each set S note that the score function is piecewise linear with discontinuities occurring only at integers. Hence we may use the same thermometer code decomposition as discussed in Section 7.2.3 to rewrite the problem in terms of $[0, 1]$ valued variables with totally unimodular constraints. The overall problem has an integral solution. ■

7.5.2 Smoothing

The approach we discussed so far works well whenever the number of queries significantly exceeds the number of pages in the cache. While the query stream of search engines is obviously tremendous, the above assumption is no longer satisfied when optimizing over hundreds of billions of pages (this would require nearly a Trillion queries to obtain good statistics in the tails).

Assume that each document d comes with a set of features ϕ_d, e.g. its relevance in the Hubs and Spokes model, or alternatively PageRank (Kleinberg, 1999; Page et al., 1998), the indegrees/outdegrees of a page, the likelihood that it is spam, or other content-related information. In this case, one would expect that such information ought to be valuable in deciding at which tier to store a page. We can take advantage of this by modelling $z_d = \langle \phi_d, w \rangle$ for a suitable parameter vector w and a page-feature vector ϕ_d. The resulting optimisation problem is convex in w and we can use the same algorithm we used for z_d to optimise over w. Focusing only ϕ_d exclusively, though, is ineffective since it ignores the fact that certain pages simply happen to be popular whereas others simply happen not to be popular at all despite meaningful features ϕ_d. Replacing ϕ_d by $(\phi_d, \nu_d e_d)$, where e_d is the unit vector for document d and ν_d is an indicator variable which characterizes an a-priori estimate of the importance of a page, allows us to have a page-specific weight for common pages whereas for infrequent pages we simply smooth over the prior coefficients.

7.5.3 Robustness

So far we assumed that v_q is exactly observed. This can be extended to allow for deviations in v by means of robust optimisation. The following minimax problem remains convex, hence it is accessible to efficient solution:

$$\underset{z}{\text{minimise}} \underset{\epsilon \in \mathcal{E}}{\text{maximise}} \sum_q \left[(v_q + \epsilon_q) \max_{d \in D_q} z_d \right] + \sum_d f_\lambda(z_d) \tag{7.11}$$

Here $\epsilon \in \mathcal{E}$ denotes an admissible perturbation of query values, and may be any ℓ_p balls ($0 < p < \infty$) around v, thus including the case of sparse perturbation when $p < 1$.

7.6 Experiments

We perform experiments on both synthetically generated data and real query-pages data, and compare our results of our proposed tiering methods to the *max* and *sum* heuristics mentioned in Section 7.4.2.

7.6.1 Experiments on Synthetic Data

The purpose of experiments on synthetic data is to obtain a small enough dataset which allows us to compare both heuristics, the online solver, and the (much

slower) LP solution *exactly*. We generated a random bipartite query-page graph using 150 queries and 150 pages. Each query vertex has a degree of 3, and value $v_q := 10(2+q)^{-0.8}$ mimicking a power law distribution of real data.

We experimented with a 2-tier system by varying the relative size of the prime (cache) tier. We evaluate system performance in *session miss*: for each session q, a miss occurs if any one of the associated pages is not found in cache, incurring v_q misses for that session. The experimental results are summarized in Figure 7.3. Our proposed method (OPT-tier) outperforms baselines by a significant margin.

To assess the convergence properties of our online algorithm, we compare the quality of the solutions given by linear program (Section 7.3.1) and online algorithm (Section 7.3.4). From Figure 7.3, shows that the online solver (ONL OPT-tier) converges to the solution of linear programming (LP OPT-tier) within few passes over the data. Note that the LP solver is computationally costly, thus unsuitable for problems even at the scale of 1000.

We examine the same synthetic data set for a 3-tier assignment problem. Here we can vary i.e. the relative sizes of the prime tier and the second tier. We report the relative improvement of our tiering algorithm as ratios of (generalized) session misses in Figure 7.4. As before, our method consistently outperforms the max heuristic and, especially the sum heuristic. We observe that the size of the prime tier affects relative improvement more than the size of the second tier.

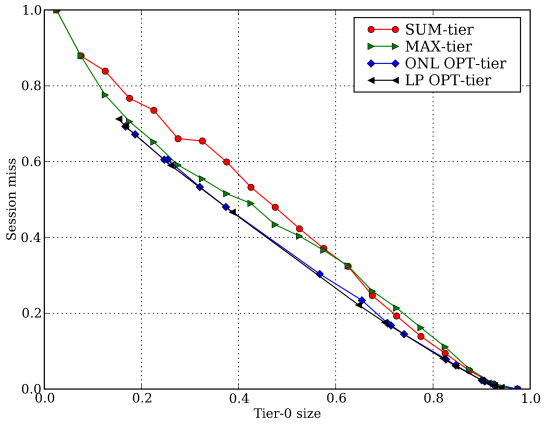


Figure 7.3: Session miss rate performance on the 150 queries-150 documents with 3 docs/query dataset. The caching performance was rescaled to yield a miss rate of 1 for a cache size of 2.5% for sessions. Our proposed method (OPT-tier) outperforms baselines by a significant margin in the total cache miss rate. The online solver (ONL OPT-tier) converges to the LP solution (LP OPT-tier).

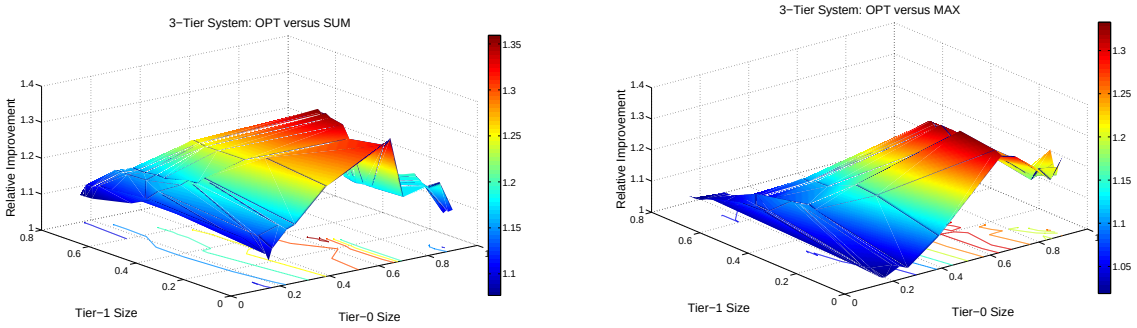


Figure 7.4: Cache performance for a set of 3 tiers. Our method consistently outperforms the baselines for all choices of both tiers. The difference is most pronounced for large tier sizes where interactions between pages matter most.

7.6.2 Real Query-Pages Data

To examine the efficacy of our algorithm at web-scale we tested it with real data from a major search engine. The results of our proposed methods are compared to those of the *max* and *sum* heuristics in Section 7.4.2. Since LP solvers are very slow, it is not feasible for web-scale problems.

We processed the logs for one week of September 2009 containing results from the top geographic regions which include a majority of the search engine's user base. To simplify the heavy processing involved for collecting such a massive data set, we only record whether a particular *result*, defined as a (query, document) pair, appears in top 10 (first result page) for a given session and we aggregate the view counts of such results, which will be used for the session value v_q once. In its entirety this subset contains about 10^8 viewed documents and $1.6 \cdot 10^7$ distinct queries. We excluded results viewed only once, yielding a final data set of $8.4 \cdot 10^7$ documents. The search results for any fixed query vary for a variety of reasons, e.g. database updates. We approximate the session graph by treating queries with different result sets as if they were different. This does not change the optimisation problem and keeps the model accurate. Moreover, we remove rare results by maintaining that the lowest count of a document is at least as large as the square root of the highest within the same session. For simplicity, our experiments are carried out for a two-tier (single cache) system such that the only design parameter is the relative size of the prime tier (the cache). Our algorithm consistently outperforms the max and sum heuristics over a large span of cache sizes (Figure 7.5).

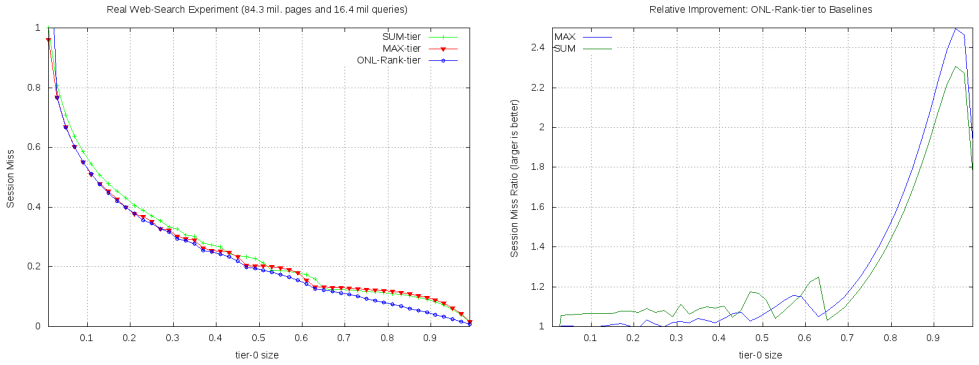


Figure 7.5: Left: Experimental results for real web-search data with $8.4 \cdot 10^7$ pages and $1.6 \cdot 10^7$ queries. Session miss rate for the online procedure, the *max* and *sum* heuristics (7.4.2). (The y-axis is normalized such that SUM-tier's first point is at 1). As seen, the max heuristic cannot be optimal for any but small cache sizes, but it performs comparably well to Online. Right: "Online" is outperforming MAX for cache size larger than 60%, sometimes more than twofold.

Direct comparison can now be made between our online procedure and the max and sum heuristics since each one induces a ranking on the set of documents. We then calculate the session miss rate of each procedure at any cache size, and report the relative improvement of our online algorithm as ratios of miss rates in Figure 7.5–Right.

We measure a throughput of approximately 0.5 million query-sessions per second (qps) for this version, and about 2 million qps for smaller problems (as they incur fewer memory page faults). Billion-scale problems can readily fit in 24GB of RAM by serializing computation one λ value at a time. We also implemented a multi-thread version utilizing 4 CPU cores, although its performance did not improve since memory and disk bandwidth limits have already been reached.

7.7 Conclusion

We showed that very large tiering and densest subset optimisation problems can be solved efficiently by a relatively simple online optimisation procedure. It came somewhat as a surprise that the max heuristic often works nearly as well as the optimal tiering solution. Since we experienced this correlation on both synthetic and real data we believe that it might be possible to prove approximation guarantees for this strategy whenever the bipartite graphs satisfy certain power-law properties.

The need for a static tiering solution might be questionable, given that data could, in theory, be reassigned between different caching tiers on the fly. The problem is that in production systems of a search engine, such reassignment of large amounts of data may not always be efficient for operational reasons (e.g. different versions of the ranking algorithm, different versions of the index, different service levels, constraints on transfer bandwidth). In addition to that, tiering is a problem not restricted to the provision of webpages. It occurs in product portfolio optimisation and other resource constrained settings. We showed that it is possible to solve such problems at several orders of magnitude larger scale than what was previously considered feasible.

Chapter 8

Conclusion and Future Directions

In an information society, information is money.
The trick is to generate value by extracting the right information from the Internet.
IDC's Digital Universe Study, 2011

In this thesis, we develop principled machine learning methods suited for challenging real-world Internet problems. The Internet has supplied an unprecedented amount of data. Synergistically with rapid progress in machine learning models and algorithms, as well as rapid rises in computing power and storage, the challenge of the 21st century consists of finding ways to transform this complex massive yet noisy and sparse Internet data coming from a variety of sources into insights in support of knowledge creation. This thesis makes contributions in addressing problems that the Internet offers some new challenges that do not naturally fit into existing machine learning methods and the Internet requires large-scale solution to problems.

The *present* work focuses on addressing Internet complexity on *output* label dimensions. The first part of this thesis deals with formulation and solution of *new* machine learning problems. The contributions of this thesis are in the following three non-standard settings forming Chapter 3, 4 and 5 of the thesis:

1. Estimating Labels from Label Proportions.
   This chapter introduces a learning setting where we are given sets of unlabelled observations, each set with known label proportions; the goal is to predict the labels of another set of observations, possibly with known label proportions. Our solution works by modelling a conditional exponential likelihood and approximating the unknown mean of sufficient statistics. The approximation is achieved: a) by exploiting the convergence properties

of a sample mean operator to its population counterpart, and b) by solving a linear system of equations formed by the known proportions.

2. Kernelised Sorting.
   This chapter introduces a learning framework where a set of data inputs and a set of data outputs are given however they are not paired; the goal of learning is to infer the latent input-output correspondences. Our solution is based on dependence maximisation between input-output pairs of observations by means of kernel embeddings based dependency measure called the Hilbert Schmidt Independence Criterion.

3. Multitask Learning without Label Correspondences.
   This chapter introduces a setting of jointly learning several related tasks where each task has potentially distinct label sets, and label correspondences are not readily available. Our solution directly maximizes the mutual information among the labels.

Traditionally, supervised machine learning settings draw inference and make prediction from a set of input objects; each of which is supervised by a desired output value. Internet poses challenges of weak label supervision (contribution #1 and #2) and label inconsistency (contribution #3).

The second part addresses refinements of existing machine learning models and algorithms to *scale* to large data. The contributions of this thesis include a streaming algorithm for the following two problems forming Chapter 6 and 7 of the thesis:

4. Distribution Matching for Transduction.
   This chapter presents a scalable algorithm for learning with labelled and unlabelled data simultaneously, based on distribution matching. We cast the goal of matched distributions over the outputs on labelled data and unlabelled data as a two-sample problem which can be solved efficiently by using a distance measure in Hilbert Space. Another advantage of our method apart from scalability is that it is 'plug and play', i.e. it is applicable to all estimation problems ranging from classification and regression to structured estimation without much need for customisation.

5. Optimal Tiering as a Flow Problem.
   This paper presents a scalable algorithm for webpage tiering. The goal is to allocate pages to caches such that the most frequently accessed pages

reside in the caches with the smallest latency. Our algorithm solves an integer linear program in an online fashion. It exploits total unimodularity of the constraint matrix and a Lagrangian relaxation to solve the problem as a convex online game. This tiering problem is related to a larger class of parametric flow problem.

The *future* work focuses on tackling Internet complexity on *input* feature dimensions. On the input dimensions, we deal not only with potentially millions of features but also the features might come from multiple modalities or data sources. Research questions arise as to how to combine those widely varied multiple modalities while taking into account the intricate *social network* representation, the Internet sparsity, and scalability, for better and more robust statistical modelling. As well, the contents of the Internet are being updated every fraction of a second. Any features defined on Internet data should be able to handle this challenge.

As mentioned, sparsity is an inherent nature of Internet data. Ideally for such a setting, Bayesian statistics provide a robust approach to drawing inferences and making predictions from very sparse information. In the framework of Bayesian probabilistic methods, all unobserved quantities would be averaged out when making predictions. Classically, Bayesian parametric approaches would require an assumption on some aspects of the model such as the mathematical form of the model. On the contrary, nonparametric models automatically adapt to the complexity of the data, making the need to specify aspects of the model no longer required. Thus nonparametric Bayesian methods are well-suited for tackling some of challenging Internet applications. It would be interesting to explore both the modelling aspect of nonparametric Bayesian methods in solving Internet challenges and in scaling up learning and inference of nonparametric Bayesian methods to handle Internet-scale data.

While a supervised machine learning is an important tool for extracting insights from the Internet, other aspects such as *user interactions* play an equally important role. The basic setup in supervised learning is that experts provide labelled examples, and the goal of learning is to build a predictor based on the predictions of those experts. The current thesis attempts to extend the basic setup to be able to handle weak label supervision and label inconsistency cases. However, all the supervised learning approaches do not interact with the world (users) and thus are not able to learn from the interactions. Internet problems, full of inherent users-systems interactions, are driving the need for interactive

machine learning approaches.

Lastly, it is likely that the statistical techniques developed in this thesis to address some of the Internet challenges could also be used in a variety of applications. For instance, contribution # 3 can be used in Genetics for integrating a collection of related gene-ontology graphs and in Bioinformatics for combining microarray measurements on different platforms (contribution # 2), among others.

Bibliography

Altun, Y. and Smola, A.J. Unifying divergence minimization and statistical inference via convex duality. In Simon, H.U. and Lugosi, G. (eds.), *Proc. Annual Conf. Computational Learning Theory (COLT)*, pp. 139–153. Springer, 2006. 37, 44, 48, 49, 52, 96, 97, 101

Ando, Rie Kubota and Zhang, Tong. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, 2005. 95, 108

Argyriou, Andreas, Evgeniou, Theodoros, and Pontil, Massimiliano. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008. 95, 102, 108

Aronszajn, N. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404, 1950. 9, 14

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., and Sherlock, G. Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nature Genetics*, 25: 25–29, 2000. 108

Asthana, Akshay, Goecke, Roland, Quadrianto, Novi, and Gedeon, Tom. Learning based automatic face annotation for arbitrary poses and expressions from frontal images only. In *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1635–1642. IEEE, 2009. 5

Bartlett, P. L. and Mendelson, S. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002. 45

Ben-David, Shai, Gehrke, Johannes, and Schuller, Reba. A theoretical framework for learning from a pool of disparate data sources. In *ACM Conf. on Knowledge Discovery and Data Mining (KDD)*, pp. 443–449. ACM, 2002. 102

Bennett, Kristin P., Bennett, P., and Parrado-Hernandez, Emilio. The interplay of optimization and machine learning research. *Journal of Machine Learning Research*, 7:1265–1281, 2006. 24

Bickel, Steffen, Brückner, Michael, and Scheffer, Tobias. Discriminative learning for differing training and test distributions. In *International Conf. on Machine Learning (ICML)*, pp. 81–88. ACM, 2007. 118

Bishop, C. M., Svensén, M., and Williams, C. K. I. GTM: The generative topographic mapping. *Neural Computation*, 10(1):215–234, 1998. 78

Borwein, J. M. and Zhu, Q. J. *Techniques of Variational Analysis*. CMS books in Mathematics. Canadian Mathematical Society, 2005. 100

Boser, B., Guyon, I., and Vapnik, V. A training algorithm for optimal margin classifiers. In Haussler, D. (ed.), *Proc. Annual Conf. Computational Learning Theory (COLT)*, pp. 144–152. ACM Press, 1992. 18

Bottou, Léon. Online algorithms and stochastic approximations. In Saad, David (ed.), *Online Learning and Neural Networks*. Cambridge University Press, Cambridge, UK, 1998. 27

Boyd, S. and Vandenberghe, L. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004. 24

Caetano, T., Caelli, T., Schuurmans, D., and Barone, D.A.C. Graphical models and point pattern matching. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(10):1646–1663, 2006. 64

Caetano, T., Cheng, L., Le, Quoc V., and Smola, A. J. Learning graph matching. In *International Conf. on Computer Vision (ICCV)*, pp. 1–8. IEEE, 2007. 63

Cai, Lijuan and Hofmann, T. Hierarchical document categorization with support vector machines. In *ACM Conf. on Information and Knowledge Management (CIKM)*, pp. 78–87. ACM Press, 2004. 108

Candes, E. and Tao, T. Decoding by linear programming. *IEEE Trans. Info Theory*, 51(12):4203–4215, 2005. 52

Caruana, R. Multitask learning. *Machine Learning*, 28:41–75, 1997. 95

Chapelle, O., Schölkopf, B., and Zien, A. (eds.). *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006. URL http://www.kyb.tuebingen.mpg.de/ssl-book. 112

Chen, B.C., Chen, L., Ramakrishnan, R., and Musicant, D.R. Learning from aggregate views. In Liu, L., Reuter, A., Whang, K.Y., and Zhang, J. (eds.), *International Conf. on Data Engineering (ICDE)*, pp. 3–12. IEEE, 2006. 54, 55

Chen, S., Donoho, D., and Saunders, M. Atomic decomposition by basis pursuit. Technical Report 479, Department of Statistics, Stanford University, May 1995. 52

Chiaia, A. C., Banta-Green, C., Power, L., Sudakin, D. L., and Field, J. A. Community burdens of methamphetamine and other illicit drugs. In *International Conf. on Pharmaceuticals and Enocrine Disrupting Chemicals in Water*, 2007. 62

Christmann, Andreas and Steinwart, Ingo. Universal kernels on non-standard input spaces. In Lafferty, J., Williams, C. K. I., Shawe-Taylor, J., Zemel, R.S., and Culotta, A. (eds.), *Advances in Neural Information Processing Systems 23 (NIPS)*, pp. 406–414. MIT Press, 2010. 14

Chung-Graham, F. *Spectral Graph Theory*. Number 92 in CBMS Regional Conference Series in Mathematics. AMS, 1997. 76

Cortes, Corinna and Vapnik, V. Support vector networks. *Machine Learning*, 20 (3):273–297, 1995. 18

Cour, T., Srinivasan, P., and Shi, J. Balanced graph matching. In Schölkopf, B., Platt, J., and Hofmann, T. (eds.), *Advances in Neural Information Processing Systems 19 (NIPS)*, pp. 313–320. MIT Press, December 2006. 63, 71

Crammer, K., Kearns, M., and Wortman, J. Learning from multiple sources. In Schölkopf, B., Platt, J., and Hofmann, T. (eds.), *Advances in Neural Information Processing Systems 19 (NIPS)*, pp. 321–328. MIT Press, 2007. 102

Csurka, Gabriella, Dance, Christopher R., Fan, Lixin, Willamowski, Jutta, and Bray, Cdric. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision*, pp. 1–22, 2004. 80

Dinh, T. Pham and An, L. Hoai. A D.C. optimization algorithm for solving the trust-region subproblem. *SIAM Journal on Optimization*, 8(2):476–505, 1988. 30, 31

Druck, G., Mann, G.S., and McCallum, A. Learning from labeled features using generalized expectation criteria. In Myaeng, S.-H., Oard, D.W., Sebastiani, F., Chua, T.-S., and Leong, M.-K. (eds.), *ACM Special Interest Group on Information Retrieval (SIGIR) Conf.*, pp. 595–602. ACM, 2008. 111

Dudík, M. and Schapire, R. E. Maximum entropy distribution estimation with generalized regularization. In Lugosi, Gábor and Simon, Hans U. (eds.), *Proc. Annual Conf. Computational Learning Theory (COLT)*. Springer, June 2006. 37, 52, 96, 97

Eisner, Mark J. and Severance, Dennis G. Mathematical techniques for efficient record segmentation in large shared databases. *Journal of the ACM*, 23(4): 619–635, 1976. 125, 126

Evgeniou, Theodoros, Micchelli, Charles A., and Pontil, Massimiliano. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6: 615–637, 2005. 102, 108

Fagin, R. Combining fuzzy information from multiple systems. In *Proc. of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pp. 216–226, 1996. 127

Fiedler, M. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(98):298–305, 1973. 76

Finke, G., Burkard, R. E., and Rendl, F. Quadratic assignment problems. *Annals of Discrete Mathematics*, 31:61–82, 1987. 65, 68

Flamary, Remi, Rakotomamonjy, Alain, Gasso, Gilles, and Canu, Stephane. Svm multi-task learning and non convex sparsity measure. In *The Learning Workshop*, 2009. 102

Ford, L. R. and Fulkerson, D. R. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956. 132

Gale, W. A. and Church, K. W. A program for aligning sentences in bilingual corpora. In *Proc. of Annual Meeting on the Association for Computational Linguistics (ACL)*, pp. 177–184, 1991. 90

Gammerman, A., Vovk, Volodya, and Vapnik, Vladimir. Learning by transduction. In *Proc. of Uncertainty in Artificial Intelligence (UAI)*, pp. 148–155. Morgan Kaufmann, 1998. 111

Gander, W., Golub, G.H., and von Matt, U. A constrained eigenvalue problem. *Linear Algebra and its Applications*, 114–115:815–839, 1989. 71

Garey, M. R. and Johnson, D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Series of Books in Mathematical Sciences. W. H. Freeman, 1979. 65

Gärtner, T., Le, Q.V., Burton, S., Smola, A. J., and Vishwanathan, S. V. N. Large-scale multiclass transduction. In Weiss, Y., Schölkopf, B., and Platt, J. (eds.), *Advances in Neural Information Processing Systems 18 (NIPS)*, pp. 411–418, Cambride, MA, 2006. MIT Press. xix, 54, 111, 115, 119, 122

Ghamrawi, Nadia and McCallum, Andrew. Collective multi-label classification. In *ACM Conf. on Information and Knowledge Management (CIKM)*. ACM Press, 2005. 97

Girosi, F. An equivalence between sparse approximation and support vector machines. A.I. Memo No. 1606, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1997. 76

Goel, S., Langford, J., and Strehl, A.L. Predictive indexing for fast search. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L. (eds.), *Advances in Neural Information Processing Systems 21 (NIPS)*, pp. 505–512. MIT Press, 2008. 127

Gold, S. and Rangarajan, A. A graduated assignment algorithm for graph matching. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(4):377–388, 1996. 63

Graça, J., Ganchev, K., and Taskar, B. Expectation maximization and posterior constraints. In Platt, J. C., Koller, D., Singer, Y., and Roweis, S. T. (eds.), *Advances in Neural Information Processing Systems 20 (NIPS)*. MIT Press, 2007. 111

Gretton, A., Bousquet, O., Smola, A.J., and Schölkopf, B. Measuring statistical dependence with Hilbert-Schmidt normmeasuring statisticals. In Jain, S., Si-

mon, H. U., and Tomita, E. (eds.), *Proc. Algorithmic Learning Theory (ALT)*, pp. 63–77. Springer-Verlag, 2005. 23

Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., and Smola, A. A kernel method for the two sample problem. Technical Report 157, MPI for Biological Cybernetics, 2008. 115

Gusfield, D. and Tardos, É. A faster parametric minimum-cut algorithm. *Algorithmica*, 11(3):278–290, 1994. 125, 128, 132

Gusfield, Dan and Martel, Charles U. A fast algorithm for the generalized parametric minimum cut problem and applications. *Algorithmica*, 7(5&6):499–519, 1992. 125, 126

Heller, I. and Tompkins, C. An extension of a theorem of dantzig's. In Kuhn, H.W. and Tucker, A.W. (eds.), *Linear Inequalities and Related Systems*, volume 38 of *Annals of Mathematics Studies*. 1956. 132

Hofmann, T., Schölkopf, B., and Smola, A. J. Kernel methods in machine learning. *Annals of Statistics*, 36:1171–1220, 2008. 11, 20, 42

Huang, J., Smola, A.J., Gretton, A., Borgwardt, K., and Schölkopf, B. Correcting sample selection bias by unlabeled data. In Schölkopf, B., Platt, J., and Hofmann, T. (eds.), *Advances in Neural Information Processing Systems 19 (NIPS)*. The MIT Press, 2007. 53, 118

Jebara, T. Kernelizing sorting, permutation, and alignment for minimum volume PCA. In *Proc. Annual Conf. on Computational Learning Theory (COLT)*, pp. 609–623. Springer, 2004. xv, 63, 64, 74, 75, 92, 93, 94

Joachims, T. Transductive inference for text classification using support vector machines. In Bratko, I. and Dzeroski, S. (eds.), *International Conf. on Machine Learning (ICML)*, pp. 200–209. Morgan Kaufmann, 1999. 111, 114, 119

Jolliffe, I. T. *Principal Component Analysis*. Springer, New York, 1986. 19

Jonker, R. and Volgenant, A. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38:325–340, 1987. 69

Kanamori, T., Hido, S., and Sugiyama, M. Efficient direct density ratio estimation for non-stationarity adaptation and outlier detection. In Koller, D.,

Schuurmans, D., Bengio, Y., and Botton, L. (eds.), *Advances in Neural Information Processing Systems 21 (NIPS)*, pp. 809–816, Cambridge, MA, USA, 2009. MIT Press. 118

Kimeldorf, G. S. and Wahba, G. Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis Applications*, 33:82–95, 1971. 16

Kleinberg, J. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999. 140

Koehn, P. Europarl: A parallel corpus for statistical machine translation. In *Machine Translation Summit X*, pp. 79–86, 2005. 89, 90

Kohonen, T. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982. 78

Kolmogorov, Vladimir, Boykov, Yuri, and Rother, Carsten. Applications of parametric maxflow in computer vision. In *International Conf. on Computer Vision (ICCV)*, pp. 1–8, 2007. 125

Kück, H. and de Freitas, N. Learning about individuals from group statistics. In *Proc. of Uncertainty in Artificial Intelligence (UAI)*, pp. 332–339. AUAI Press, 2005. 54, 57

Lafferty, J. D., McCallum, A., and Pereira, F. Conditional random fields: Probabilistic modeling for segmenting and labeling sequence data. In *International Conf. on Machine Learning (ICML)*, volume 18, pp. 282–289. Morgan Kaufmann, 2001. 123

Le, Q.V., Smola, A.J., Gärtner, T., and Altun, Y. Transductive gaussian process regression with automatic model selection. In Fürnkranz, J., Scheffer, T., and Spiliopoulou, M. (eds.), *European Conference on Machine Learning (ECML)*, volume 4212. 306-317, 2006. 111, 115

Ledoux, M. and Talagrand, M. *Probability in Banach Spaces*. Springer, 1991. 44

Leung, Gilbert, Quadrianto, Novi, Smola, Alexander, and Tsioutsiouliklis, Kostas. Optimal web-scale tiering as a flow problem. In Lafferty, J., Williams, C. K. I., Shawe-Taylor, J., Zemel, R.S., and Culotta, A. (eds.), *Advances in Neural Information Processing Systems 23 (NIPS)*, pp. 1333–1341, 2010. 4

Lewis, David D., Yang, Yiming, Rose, Tony G., and Li, Fan. RCV1: A new benchmark collection for text categorization research. *The Journal of Machine Learning Research*, 5:361–397, 2004. 105

Lowe, David G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004. 80

Malcolm, W. P., Quadrianto, Novi, and Aggoun, Lakhdar. State estimation schemes for independent component coupled hidden markov models. *Stochastic Analysis and Applications*, 28(3):430–446, 2010. 5

Mann, G. and McCallum, A. Simple, robust, scalable semi-supervised learning via expectation regularization. In Ghahramani, Zoubin (ed.), *International Conf. on Machine Learning (ICML)*, pp. 593–600. Omnipress, 2007. 54

McCallum, A. and Li, W. Early results for named entity recognition with conditional random fields, feature induction and web enhanced lexicons. In *Conf. on Computational Natural Language Learning (CoNLL)*. ACL, 2003. 123

McDiarmid, C. On the method of bounded differences. In *Survey in Combinatorics*, pp. 148–188. Cambridge University Press, 1989. 67

Mendelson, S. Rademacher averages and phase transitions in glivenko-cantelli classes. *IEEE Trans. Information Theory*, 48(1):251–263, 2002. 44

Musicant, D.R., Christensen, J., and Olson, J.F. Supervised learning by training on aggregate outputs. In *International Conf. on Data Mining (ICDM)*. IEEE, 2007. 54, 55

Nesterov, Y. and Vial, J.-P. Confidence level solutions for stochastic programming. Technical Report 2000/13, Université Catholique de Louvain - Center for Operations Research and Economics, 2000. 138

Nguyen, X.L., Wainwright, M., and Jordan, M. Estimating divergence functionals and the likelihood ratio by penalized convex risk minimization. In *Advances in Neural Information Processing Systems 20 (NIPS)*. MIT Press, 2008. 118

Obozinski, G., Taskar, B., and Jordan, M. I. Multi-task feature selection. Technical report, U.C. Berkeley, 2007. 102

Page, L., Brin, S., Motwani, R., and Winograd, T. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, Stanford University, Stanford, CA, USA, November 1998. 140

Papadimitriou, C. H. and Steiglitz, K. *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, New Jersey, 1982. 128, 130

Parameswaran, Shibin and Weinberger, Kilian. Large margin multi-task metric learning. In Lafferty, J., Williams, C. K. I., Shawe-Taylor, J., Zemel, R.S., and Culotta, A. (eds.), *Advances in Neural Information Processing Systems 23 (NIPS)*, pp. 1867–1875. 2010. 102

Persin, M., Zobel, J., and Sacks-Davis, R. Filtered document retrieval with frequency-sorted indexes. *Journal of the American Society for Information Science*, 47(10):749–764, 1996. 127

Quadrianto, N. and Lampert, C.H. Learning multi-view neighborhood preserving projections. In *International Conf. on Machine Learning (ICML)*, pp. 425–432. Omnipress, 2011a. 5

Quadrianto, N., Smola, A., Caetano, T., and Le, Q. Estimating labels from label proportions. In *International Conf. on Machine Learning (ICML)*, pp. 776–783. Omnipress, 2008. 4

Quadrianto, N., Smola, A., Caetano, T., and Le, Q. Estimating labels from label proportions. *Journal of Machine Learning Research*, 10:2349–2374, 2009a. 4

Quadrianto, N., Song, L., and Smola, A. Kernelized sorting. In Koller, D., Schurmans, D., Bengio, Y., and Bottou, L. (eds.), *Advances in Neural Information Processing Systems 22 (NIPS)*, pp. 1289–1296, 2009b. 4

Quadrianto, Novi and Buntine, Wray L. Linear discriminant. In *Encyclopedia of Machine Learning*, pp. 601–603. Springer, 2010a. 6

Quadrianto, Novi and Buntine, Wray L. Linear regression. In *Encyclopedia of Machine Learning*, pp. 603–606. Springer, 2010b. 5

Quadrianto, Novi and Buntine, Wray L. Regression. In *Encyclopedia of Machine Learning*, pp. 838–842. Springer, 2010c. 5

Quadrianto, Novi and Lampert, Christoph. Kernel-based learning. In *Encyclopedia of Systems Biology*. Springer, 2011b. to appear. 5

Quadrianto, Novi, Caetano, Tiberio, Lim, John, and Schuurmans, Dale. Convex relaxation of mixture regression with efficient algorithms. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C. K. I., and Culotta, A. (eds.), *Advances in Neural Information Processing Systems 22 (NIPS)*, pp. 1491–1499, 2009c. 5

Quadrianto, Novi, Kersting, Kristian, Reid, Mark D., Caetano, Tibério S., and Buntine, Wray L. Kernel conditional quantile estimation via reduction revisited. In *International Conf. on Data Mining (ICDM)*, pp. 938–943. IEEE, 2009d. 5

Quadrianto, Novi, Petterson, James, and Smola, Alex. Distribution matching for transduction. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C. K. I., and Culotta, A. (eds.), *Advances in Neural Information Processing Systems 22 (NIPS)*, pp. 1500–1508, 2009e. 4

Quadrianto, Novi, Kersting, Kristian, Tuytelaars, Tinne, and Buntine, Wray L. Beyond 2d-grids: a dependence maximization view on image browsing. In *ACM SIGMM International Conf. on Multimedia Information Retrieval (MIR)*, pp. 339–348. ACM, 2010a. 4

Quadrianto, Novi, Kersting, Kristian, and Xu, Zhao. Gaussian process. In *Encyclopedia of Machine Learning*, pp. 428–439. Springer, 2010b. 5

Quadrianto, Novi, Smola, Alex J., Song, Le, and Tuytelaars, Tinne. Kernelized sorting. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32(10):1809–1821, 2010c. 4

Quadrianto, Novi, Smola, Alexander, Caetano, Tiberio, Vishwanathan, S.V.N., and Petterson, James. Multitask learning without label correspondences. In Lafferty, J., Williams, C. K. I., Shawe-Taylor, J., Zemel, R.S., and Culotta, A. (eds.), *Advances in Neural Information Processing Systems 23 (NIPS)*, pp. 1957–1965, 2010d. 4

Risvik, K. M., Aasheim, Y., and Lidal, M. Multi-tier architecture for web search engines. In *Conf. on Latin American Web Congress*, pp. 132–143. IEEE, 2003. 127

Roweis, S. and Saul, L. K. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000. 78

Rüping, Stefan. Svm classifier estimation from group probabilities. In *International Conf. on Machine Learning (ICML)*, pp. 911–918, 2010. 54

Sang, Erik F. Tjong Kim and Buchholz, S. Introduction to the CoNLL-2000 shared task: Chunking. In *Conf. on Computational Natural Language Learning (CoNLL)*, pp. 127–132, 2000. 124

Saunders, Craig, Gammermann, A., and Vovk, Volodya. Ridge regression learning algorithm in dual variables. In *International Conf. on Machine Learning (ICML)*, pp. 515–521. Morgan Kaufmann, 1998. 19

Schaback, R. Kernel-based meshless methods. http://num.math.uni-goettingen.de/schaback/teaching/texte/approx/Appverf_II.pdf, 2007. 14

Schölkopf, B. *Support Vector Learning*. R. Oldenbourg Verlag, 1997. http://www.kernel-machines.org. 57, 103

Schölkopf, B. and Smola, A. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002. 11, 12, 16, 20, 51

Schölkopf, B., Smola, A. J., and Müller, K.-R. Nonlinear component analysis as a kernel eigenvalue problem. Technical Report 44, Max-Planck-Institut für biologische Kybernetik, 1996. 19

Schölkopf, B., Smola, A. J., and Müller, K.-R. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998. 76

Schölkopf, B., Tsuda, K., and Vert, J.-P. *Kernel Methods in Computational Biology*. MIT Press, Cambridge, MA, 2004. 11, 20

Serfling, R. *Approximation Theorems of Mathematical Statistics*. Wiley, New York, 1980. 22

Shawe-Taylor, J. and Cristianini, N. *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, UK, 2004. 11, 12, 20

Sherman, S. On a Theorem of Hardy, Littlewood, Polya, and Blackwell. *Proceedings of the National Academy of Sciences (PNAS)*, 37(12):826–831, 1951. 65

Sindhwani, V. and Keerthi, S.S. Large scale semi-supervised linear SVMs. In *ACM Special Interest Group on Information Retrieval (SIGIR) Conf.*, pp. 477–484. ACM, 2006. xvi, 119, 120, 121

Smola, A. J., Schölkopf, B., and Müller, K.-R. The connection between regularization operators and support vector kernels. *Neural Networks*, 11(5):637–649, 1998. 76

Smola, A.J., Gretton, A., Song, L., and Schölkopf, B. A hilbert space embedding for distributions. In Takimoto, E. (ed.), *Algorithmic Learning Theory (ALT)*. Springer, 2007a. 20, 21, 22, 23, 66, 72

Smola, Alex, Vishwanathan, S. V. N., and Le, Quoc. Bundle methods for machine learning. In Koller, Daphne and Singer, Yoram (eds.), *Advances in Neural Information Processing Systems 20 (NIPS)*. MIT Press, 2007b. 57

Steinke, F., Schölkopf, B., and Blanz, V. Learning dense 3d correspondence. In Schölkopf, B., Platt, J., and Hofmann, T. (eds.), *Advances in Neural Information Processing Systems 19 (NIPS)*, pp. 1313–1320. MIT Press, 2007. 64, 77

Steinwart, I. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2:67–93, 2001. 14, 15

Stone, H. S. Critical load factors in two-processor distributed systems. *IEEE Trans. Software Engineering*, 4(3):254–258, 1978. 125, 126

Sugiyama, M., Nakajima, S., Kashima, H., von Bünau, P., and Kawanabe, M. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in Neural Information Processing Systems 20*, pp. 1433–1440, Cambridge, MA, 2008. 118

Teo, C.H., Le, Q., Smola, A.J., and Vishwanathan, S.V.N. A scalable modular convex solver for regularized risk minimization. In *ACM Conf. on Knowledge Discovery and Data Mining (KDD)*. ACM, 2007. 57

Tikhonov, A. N. On the stability of inverse problems. *Dokl. Akad. Nauk SSSR*, 39(5): 195–198, 1943. 16

Tikhonov, A. N. Solution of incorrectly formulated problems and the regularization method. *Soviet Math. Dokl.*, 4:1035–1038, 1963. 16

Torralba, Antonio, Russell, Bryan C., and Yuen, Jenny. Labelme: Online image annotation and applications. *Proceedings of The IEEE*, 98:1467–1484, 2010. 85

Tripathi, Abhishek, Klami, Arto, Oresic, Matej, and Kaski, Samuel. Matching samples of multiple views. *Data Mining and Knowledge Discovery*, 23(2):300–321, 2011. 87

Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005. 42

Vapnik, V. *The Nature of Statistical Learning Theory*. Springer, New York, 1995. 16, 19

Walder, C., Schl̈kopf, B., and Chapelle, O. Implicit surface modelling with a globally regularised basis of compact support. *Computer Graphics Forum*, 25(3):635–644, 2006. 64, 77

Wedin, Per-Åke. Perturbation theory for pseudo-inverses. *BIT Numerical Mathematics*, 13(2), 1973. 50

Weinberger, K. Q. and Saul, L. K. An introduction to nonlinear dimensionality reduction by maximum variance unfolding. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 2006. 78

Weinberger, Kilian Q. and Saul, Lawrence K. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009. 102

Yamada, Makoto and Sugiyama, Masashi. Cross-domain object matching with model selection. In *International Conf. on Artificial Intelligence and Statistics (AISTATS)*, 2011. 79

Yan, H., Ding, S., and Suel, T. Inverted index compression and query processing with optimized document ordering. In Quemada, J., León, G., Maarek, Y.S., and Nejdl, W. (eds.), *Proc. of International Conf. on World Wide Web (WWW)*, pp. 401–410. ACM, 2009. 127

Yu, Kai, Tresp, Volker, and Schwaighofer, Anton. Learning gaussian processes from multiple tasks. In *International Conf. on Machine Learning (ICML)*, pp. 1012–1019. ACM, 2005. 95, 102, 108

Yuille, A.L. and Rangarajan, A. The concave-convex procedure. *Neural Computation*, 15:915–936, 2003. 29, 31

Zhang, B., Ward, J., and Feng, Q. A simultaneous parametric maximum-flow algorithm for finding the complete chain of solutions. Technical Report HPL-2004-189, Hewlett Packard Laboratories, 2004. URL http://www.hpl.hp.com/techreports/2004/HPL-2004-189.html. 125, 126

Zhang, B., Ward, J., and Feng, A. A simultaneous maximum flow algorithm for the selection model. Technical Report HPL-2005-91, Hewlett Packard Laboratories, 2005a. URL http://www.hpl.hp.com/techreports/2005/HPL-2005-91.html. 125

Zhang, B., Ward, J., and Feng, Q. Simultaneous parametric maximum flow algorithm with vertex balancing. Technical Report HPL-2005-121, Hewlett Packard Laboratories, 2005b. URL http://www.hpl.hp.com/techreports/2005/HPL-2005-121.html. 125

Zien, A., Brefeld, U., and Scheffer, T. Transductive support vector machines for structured variables. In *International Conf. on Machine Learning (ICML)*, pp. 1183–1190, 2007. 111, 114

Zinkevich, M. Online convex programming and generalised infinitesimal gradient ascent. In *International Conf. on Machine Learning (ICML)*, pp. 928–936, 2003. 28, 136