

**THEORY AND PRACTICE OF  
TRANSDUCTIVE LEARNING**

**DMITRY PECHYONY**



**THEORY AND PRACTICE OF  
TRANSDUCTIVE LEARNING**

RESEARCH THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE  
REQUIREMENTS  
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

**DMITRY PECHYONY**

SUBMITTED TO THE SENATE OF THE TECHNION — ISRAEL INSTITUTE OF TECHNOLOGY

TISHREI, 5769

HAIFA

OCTOBER, 2008



THIS RESEARCH THESIS WAS DONE UNDER THE SUPERVISION OF  
ASSOC. PROF. RAN EL-YANIV IN THE DEPARTMENT OF COMPUTER  
SCIENCE

## ACKNOWLEDGMENTS

This thesis would be impossible without a wise guidance of my advisor, Prof. Ran El-Yaniv. The path to the results presented in this thesis was long and I thank Ran for never losing the faith in the final success. In both peaceful and stressful times, Ran constantly supported and navigated me towards stronger results.

Many thanks go to my coauthors during the thesis period: Ron Begleiter, Dr. Corinna Cortes, Prof. Mehryar Mohri, Dr. Ashish Rastogi, Dr. Elad Yom-Tov and Prof. Vladimir Vapnik. I learned a lot of new things during the collaboration with all of you. I am indebted to Vladimir for offering me postdoc position in his group.

I would like to thank Prof. Ron Meir and Dr. Saharon Rosset for serving on my thesis exam. Also I thank Prof. Alon Itai, Prof. Eyal Kushilevich, Prof. Shaul Markovich, Prof. Ron Meir and Dr. Yoel Ratsaby for serving on my candidacy exam.

I am grateful to the Technion and PASCAL - European Network of Excellence for their financial and technical support.

On the personal side, I would like to thank my parents Grigory and Natalya and my parents-in-law Mark and Liliya for supporting and encouraging me during the thesis period. I am grateful to my wife Valeria for tolerating me during the tough periods of the thesis and for taking care of my social life. Finally, thanks to my daughter Celine, who took care of my emotional support during the last year.

THE GENEROUS FINANCIAL HELP OF THE TECHNION IS  
GRATEFULLY ACKNOWLEDGED



# Contents

<b>Abstract</b>	<b>1</b>
<b>1 Introduction</b>	<b>5</b>
1.1 Background	5
1.2 Applications of Transductive Learning	7
1.2.1 Text classification	7
1.2.2 Image processing	7
1.2.3 Lossy compression	8
1.2.4 Graph reconstruction	9
1.2.5 Natural language processing	9
1.3 Thesis Structure	10
<b>2 Survey of Transductive Learning</b>	<b>11</b>
2.1 Theory of Transductive Learning	11
2.1.1 Two models of transductive learning	11
2.1.2 Upper risk bounds	13
2.1.3 Consistency	27
2.1.4 Lower bounds	28
2.1.5 Relation to other learning models	30
2.1.6 When transduction is better than induction and on the value of unlabeled examples	31
2.1.7 Related learning models	32
2.1.8 Summary	36
2.2 Transductive Algorithms	36
2.2.1 Large-margin methods	37
2.2.2 Graph-based methods	39
2.2.3 Mixed large margin and graph-based methods	46
2.2.4 Volume regularization	46
2.2.5 Gaussian processes	47
2.2.6 Boosting	48
2.2.7 Methods based on the minimization of risk bounds	49
2.2.8 Statistical physics methods	50
2.2.9 Self-training methods	51

2.2.10	Agreement-based methods . . . . .	53
2.2.11	Scalability issues . . . . .	54
2.2.12	Empirical comparison of algorithms . . . . .	55
2.2.13	Summary . . . . .	55
2.3	Proof of Lemma 1 . . . . .	55
<b>3</b>	<b>Concentration Inequalities for Functions over Partitions</b>	<b>57</b>
3.1	Inequality based on strong permutation stability . . . . .	59
3.2	Inequality based on weak permutation stability . . . . .	60
3.3	Concluding Remarks . . . . .	61
3.4	Proofs . . . . .	62
3.4.1	Proof of Theorem 1 . . . . .	62
3.4.2	Proof of Lemma 4 . . . . .	63
3.4.3	Proof of Theorem 2 . . . . .	65
<b>4</b>	<b>Transductive Stability</b>	<b>68</b>
4.1	Introduction . . . . .	68
4.2	Related Work . . . . .	68
4.3	Definitions . . . . .	69
4.4	Uniform Stability Bound . . . . .	70
4.5	Weak Stability Bound . . . . .	73
4.6	High Confidence Stability Estimation . . . . .	75
4.6.1	Quantile Estimation . . . . .	75
4.6.2	Stability Estimation Algorithm . . . . .	76
4.6.3	Stability Estimation Examples . . . . .	77
4.7	Concluding Remarks . . . . .	79
<b>5</b>	<b>Transductive Rademacher Complexity and its Applications</b>	<b>81</b>
5.1	Introduction . . . . .	81
5.1.1	Related Work . . . . .	82
5.2	Definitions . . . . .	83
5.2.1	Learning Model . . . . .	83
5.2.2	Transductive Rademacher complexity . . . . .	84
5.3	Uniform Rademacher error bound . . . . .	86
5.3.1	Uniform concentration inequality for a set of vectors . . . . .	86
5.3.2	Contraction of Rademacher complexity . . . . .	89
5.3.3	Risk bound and comparison with related results . . . . .	90
5.4	Unlabeled-Labeled Representation (ULR) of transductive algorithms	
93		
5.4.1	Generic bound on transductive Rademacher complexity . . . . .	94
5.4.2	Kernel ULR . . . . .	95
5.4.3	Monte-Carlo Rademacher bounds . . . . .	97
5.5	Applications: Explicit bounds for specific algorithms . . . . .	98

5.5.1	The Spectral Graph Transduction (SGT) algorithm of Joachims (2003)	98
5.5.2	Kernel-ULR of the algorithm of Belkin et al. (2004)	99
5.5.3	The Consistency Method of Zhou et al. (2004)	102
5.6	PAC-Bayesian bound for transductive mixtures	104
5.7	Concluding remarks	105
5.8	Proofs	106
5.8.1	Proof of Lemma 9	106
5.8.2	Proof of Lemma 10	108
5.8.3	Proof of Lemma 11	112
5.8.4	Proof of Lemma 12	113
5.8.5	Proofs from Section 5.5.2	114
5.8.6	Proof of Lemma 15	116
5.8.7	Proofs from Section 5.6	116
<b>6</b>	<b>Large Margin versus Large Volume in Transductive Learning</b>	<b>119</b>
6.1	Introduction	119
6.2	The transductive setting	120
6.3	Transductive maximum power inference	121
6.4	On priors and powers	122
6.5	A large volume principle	123
6.6	Transductive learning using the large volume principle	124
6.6.1	Volume approximation	125
6.6.2	Approximate Volume Regularization (AVR) algorithm	126
6.7	Global optimum AVR optimization	127
6.8	A risk bound	129
6.9	Experimental results	130
6.9.1	On the AVR hyperparameters	131
6.9.2	Results	133
6.9.3	Analysis of results	134
6.10	Concluding remarks	136
	<b>Bibliography</b>	<b>137</b>
	<b>References</b>	<b>137</b>
	<b>Hebrew Abstract</b>	<b>i</b>

# List of Figures

2.1	AnyBoost family of algorithms . . . . .	49
4.1	Stability estimates and the corresponding empirical/true errors . .	80
5.1	A comparison of transductive Rademacher bounds . . . . .	103
6.1	Large-margin vs. large-volume prior . . . . .	120
6.2	Visualization of hypothesis space . . . . .	125
6.3	Structure of the function $f(\rho)$ . . . . .	129
6.4	Accuracy of the eigenvectors . . . . .	135
6.5	Comparison of AVR versus TSVM . . . . .	135

# List of Tables

6.1	Positive results . . . . .	133
6.2	Negative results . . . . .	134



# Abstract

In transductive learning we are given a training set of labeled examples and a test set of unlabeled examples. The goal is to guess an accurate labeling of the test points. In this thesis we present several results for learning within transductive setting, focusing on both theoretical and practical aspects. The guiding line of the thesis is to connect the theoretical results with the practical and efficient algorithm. The results of the thesis include the development of performance guarantees for the existing algorithms. Also we develop a new learning algorithm based on the existing generic performance guarantees.

We present two novel techniques for deriving explicit data-dependent error bounds for transductive algorithms. The first technique is based on the transductive notion of uniform and weak stability of learning algorithm, and the second technique is based on transductive Rademacher complexity. These bounding techniques are applicable to many transductive algorithms and we demonstrate their effectiveness by deriving bounds for several known ones.

We also consider a large volume learning principle for transduction that prioritizes transductive equivalence classes according to the “volume” they occupy in the hypothesis space. We approximate volume computation using a geometric interpretation of the hypothesis space. The resulting transductive learning algorithm is a non-convex optimization problem that can be solved efficiently. A comparison of our algorithm with large-margin methods (SVM and TSVM) over large number of datasets demonstrates an overwhelming advantage in several interesting domains but a clear disadvantage in others.



# Abbreviations and Notations

AVR	Approximate Volume Regularization
CM	Consistency Method
EM	Expectation Maximization
MST	Minimal Spanning Tree
NN	Nearest Neighbor
PAC	Probably Approximate Correct
RBF	Radial Basis Function
RKHS	Reproducing Kernel Hilbert Space
SGT	Spectral Graph Transducer
SVM	Support Vector Machine
TSVM	Transductive Support Vector Machine
TF-IDF	Term Frequency-Inverse Document Frequency
ULR	Unlabeled-Labeled Representation
$\mathcal{X}$	Space of unlabeled examples
$\mathcal{Y}$	Space of class labels
$\mathcal{D}$	Distribution over $\mathcal{X} \times \mathcal{Y}$
$x, x_i$	Unlabeled example
$\langle x, y \rangle, \langle x_i, y_i \rangle$	Labeled example
$m$	Training set size
$u$	Test set size
$S_m$	Labeled training set
$X_u$	Unlabeled test set
$S_u$	Labeled test set
$S_{m+u}$	Labeled full sample
$X_{m+u}$	Unlabeled full sample
$Y_m$	Labels of training examples
$Y_u$	Labels of test examples
$Y_{m+u}$	Full sample labels
$\mathcal{A}$	Transductive algorithm

$\ell$	0/1 loss function
$\ell_\gamma$	$\gamma$ -margin loss function
$\mathcal{L}_u(\mathcal{A})$	test error of $\mathcal{A}$ w.r.t. 0/1 loss
$\mathcal{L}_m(\mathcal{A})$	training error of $\mathcal{A}$ w.r.t. 0/1 loss
$\mathcal{L}_u^\gamma(\mathcal{A})$	test error of $\mathcal{A}$ w.r.t. $\gamma$ -margin loss
$\mathcal{L}_m^\gamma(\mathcal{A})$	training error of $\mathcal{A}$ w.r.t. $\gamma$ -margin loss
$\mathcal{H}$	Hypothesis space
$\mathcal{H}_{\mathcal{A}}$	Actual hypothesis space of algorithm $\mathcal{A}$
$\mathbf{h}$	Transductive hypothesis

# Chapter 1

## Introduction

### 1.1 Background

Our era is characterized by vast amounts of data. The data is generated by various sources and in different forms, for example, biological data, medical data, financial data, customer data and surveillance data. Also, all of us are data consumers that are required, on a daily basis, to filter a large amount of data and to find the data most relevant to us at any given moment. The field of machine learning is concerned with the intelligent analysis of the data. Such analysis can potentially reveal hidden information and regularities, which may be of high interest to data consumers. Also, the intelligent analysis may result in the replacement of humans with automated systems. For example, due to machine learning-based system that recognizes handwritten text, millions of checks that are deposited through automatic teller machines in the USA are processed fully automatically, without any intervention from a human teller (LeCun et al., 1998).

This thesis is concerned with a particular approach to machine learning, that performs learning from examples. In this approach the data is represented as a (possibly infinite) set of examples. A number of models for learning from examples exist. The most widespread model is supervised learning. Recently, also semi-supervised and transductive models have received significant attention. The latter learning model is the topic of this thesis.

In the traditional supervised learning model the data space is a set of all conceivable examples and each example consists of a finite set of attributes (called *features*) and a *target value*. An example along with its target value is called a *labeled example*. A finite set of labeled examples, called a *training set*, is sampled from the data space and given to the learning algorithm as input. The goal of the learning algorithm is to produce a *hypothesis*. The hypothesis should be a function from data space examples to possible target values. The quality of a hypothesis is measured via a loss function that quantifies the discrepancy

between predicted targets and true target values. The average discrepancy between predicted and true target values, measured over the unknown distribution of data space examples, is called a *generalization error*. The goal of the learning algorithm is to select a hypothesis that minimizes the generalization error. This learning model is called ‘supervised learning’ because only labeled examples are used by the algorithm to generate a hypothesis.

While supervised learning has been the main concern of statistical learning for many years, it turns out that in many practical applications the learner may have access to unlabeled examples during training. In *semi-supervised learning* the training set consists of both labeled and unlabeled examples. The goal in semi-supervised learning is the same as in supervised learning, namely to produce a hypothesis with a small generalization error. Over the last few years semi-supervised learning has become a very active research area. A partial list of applications of semi-supervised learning includes text and web page classification (Nigam et al., 2000; Blum & Mitchell, 1998), surveillance (Balcan et al., 2005), intrusion detection and computer security (Lane, 2006), drug design (Weston et al., 2003), protein classification (Weston et al., 2005), cancer diagnostics (Bair & Tibshirani, 2004) and ECG classification (Hughes et al., 2004). In all these and many other practical scenarios, labeled examples are scarce and hard to obtain, while unlabeled examples are abundant. Useful semi-supervised learning algorithms are, therefore, of great importance.

Both supervised and semi-supervised learning are *inductive* learning models. In inductive learning the learning algorithm receives as input a training set and should generate a hypothesis predicting the label of *any* example from the data space. In contrast, in *transductive learning* the learner receives as input a sample consisting of both a training and test sets and the goal is to predict only the labels of examples in the test set.<sup>1</sup> This sample, consisting of both labeled and unlabeled examples, is called the *full sample*<sup>2</sup>. The transductive error is defined via some loss function that measures the discrepancy between predicted and true labels of examples in the test set. Note that the usage of unlabeled examples is a common property of both transductive and semi-supervised learning<sup>3</sup>.

Clearly, any supervised learning algorithm can be straightforwardly applied to solve the transductive learning problem: simply generate a hypothesis from

---

<sup>1</sup>An informal analogy can be made between transductive learning and a take-home exam, and between inductive learning and a classroom exam.

<sup>2</sup>Existing transductive learning results assume a labeled training set and an unlabeled test set, but in general the training set can include both labeled and unlabeled examples.

<sup>3</sup>In the machine learning community there still exists a confusion between the definitions of transductive and semi-supervised learning. In particular, in a number of papers (see, for example, Belkin & Niyogi, 2004; Belkin et al., 2004, 2006; Chapelle et al., 2003; Chapelle & Zien, 2005; Zhu et al., 2003) transductive learning (as defined here) is called semi-supervised learning and semi-supervised learning is called out-of-sample (or inductive) semi-supervised learning. The definition of transductive learning, which we use, was introduced by Vapnik (1982).

the training set and use this hypothesis to label all test set examples. Since one can utilize the available unlabeled examples in the test set for selecting a hypothesis, a better solution may be obtained by using a semi-supervised learning algorithm that may also exploit the unlabeled examples. The outcome in both cases is a general hypothesis that can be applied to any example. However, in transduction one is not interested in inferring a general rule, but rather only in labeling the unlabeled set as accurately as possible. In this sense, transductive learning appears at the outset to be the easier problem, and as pointed out by Vapnik (1982, 1998), it makes little sense to solve what appears to be an easy problem by ‘reducing’ it to a more difficult one. It is, therefore, sensible that an efficient learning algorithm for transduction should not generate an hypothesis but rather “transfer” the relevant information from the training set to the test set.

## 1.2 Applications of Transductive Learning

In this section we show a number of applications of the transductive setting in various domains. The common characteristic of the problems described below is that they are of a ‘transductive nature’. Namely, in all these problems the entire full sample is known to the learner prior to the learning process, and it is unlikely that the learner will be asked to predict the labels of the points that are not in the full sample.

### 1.2.1 Text classification

Suppose we need to organize a fixed collection of documents. The simplest organization would be to divide the documents into a number of categories. More complicated ways of organization (e.g., hierarchy of categories) may also be considered. In this problem each document is an example. Since the collection of documents is fixed, no other documents will appear and hence the problem has a transductive nature. The experiments with transductive text classification algorithms can be found in (Ifrim & Weikum, 2006; El-Yaniv et al., 2008).

### 1.2.2 Image processing

Each pixel in an image can be considered as an example. An entire image, consisting of a finite set of pixels, represents a full sample. Since an image has a constant number of pixels and they are known in advance, this setting has a transductive nature. Hence the image processing field potentially contains a large amount of transductive learning problems. The example of the transductive problem in image processing is the following image colorization problem (Levin et al., 2004). Suppose we have the gray-valued color of all pixels and, in addition,

for a small number of pixels we know their RGB values. The problem is to find the true RGB color of all gray pixels. In this problem the attributes of the example are the location of the corresponding pixel and its gray color. The label of the pixel is its RGB color.

Another transductive learning problem in image processing is user-guided image segmentation (e.g., see Blake et al. (2004) and the references therein). Given an image showing a number of objects, we would like to find pixels associated with the same object. The additional input is the approximate (and probably inaccurate) boundary between the objects. In this problem the positive examples are the pixels that lie on the boundary and the negative examples, and the pixels located within the objects. The attributes of each example are its RGB color, the color of its neighbors, and the geometric location of the corresponding pixel.

### 1.2.3 Lossy compression

Suppose we have an object consisting of a finite number of discrete elements such that each such element has a number attributes and a number of labels. We would like to compress such an object, namely to find a small number of representative elements such that based on their labels we will be able to infer the labels of other elements. We now show two instantiations of this problem to particular domains.

The first domain is image compression (Cheng & Vishvanathan, 2007). In this domain each element is a pixel and the labels are its RGB colors. The image is represented by a neighborhood graph of pixels. Given the size of the image, this graph is known to both encoder and decoder. The encoder finds the most representative pixels of the image, such that given the color of these pixels, the color of other pixels could be learned by some fixed transductive algorithm (that is known to both encoder and decoder). The compressed image consists of the location and the color of the representative pixels. The decoder receives these pixels and runs transductive learning algorithm in order to learn the colors of other pixels.

The second domain is the compression of 3D mesh models in computer graphics (Mahadevan, 2007). In this domain the elements are mesh nodes, which are interconnected by weighted edges. Hence the mesh nodes constitute a weighted graph. The labels of the nodes are the 3D location and RGB colors of the nodes. Unlike the domain of images, in the 3D mesh model domain, the graph of mesh nodes is known to the encoder but is unknown to the decoder. The encoder compresses the graph of the mesh nodes (e.g., using spectral techniques). This compression is lossy and the decoder may only be able to obtain the approximate version of the original graph of the mesh nodes. In addition, the encoder finds in this approximate graph the most representative nodes such that given the 3D coordinates and RGB color of these nodes the decoder would be able to learn, using a fixed transductive algorithm, the 3D location and RGB colors of the rest of the nodes.

## 1.2.4 Graph reconstruction

Suppose we are given a graph  $G = (V, E)$  with vertices  $V$  and edges  $E \subset V \times V$ . We are also given a set of pairs of vertices  $D \subset V \times V$ ,  $D \cap E = \emptyset$ , such that for any pair  $(v_1, v_2) \in D$  we know that the vertices  $v_1$  and  $v_2$  are not similar to each other. We refer to  $D$  as a set of dissimilarity edges. The problem is to predict for any pair  $(v_1, v_2) \in V \times V \setminus (E \cup D)$  if there is an edge between  $v_1$  and  $v_2$ . In this setting the positive examples are the edges in  $E$  and the negative examples are the edges in  $D$ . The attributes of examples (edges) are the additional information that we have about the adjacent vertices. The graph reconstruction problem has a number of instantiations in several domains.

The graph examples in the bioinformatics domain (Vert & Yamanishi, 2005) include protein interaction networks, gene regulatory networks and metabolic networks. In these graphs the edges  $E$  and  $D$  represent the presence and absence of the interaction between biological entities. In the social networks domain the graph is a network of people. The edges  $E$  and  $D$  of this graph tell us about the presence/absence of friendship between individuals. In the collaborative filtering domain the graph is bipartite graph  $G = (V_1, V_2, E)$ . The first set  $V_1$  of vertices represents people and the second set  $V_2$  of vertices represents movies (or books). In this graph the edges may appear only between  $V_1$  and  $V_2$ , but not inside  $V_1$  or  $V_2$ . The presence/absence of the edge between  $(v_1, v_2) \in V_1 \times V_2$  indicates whether the user  $v_1$  liked/disliked the movie  $v_2$ .

## 1.2.5 Natural language processing

There is a number of transductive learning problems in the natural language processing field. These problems appear in particular when dealing with resource-poor languages, for which there is a single or very small number of datasets. We give examples for two such problems.

The first one is part-of-speech (POS) tagging of the corpus of documents (Duh & Kirchhoff, 2006). In this problem the examples are the words, and their labels are POS tags. Initially the learner is given the tags of a small fraction of the words and the goal is to find the POS tags of the rest of the words.

The second problem is statistical machine translation (Ueffing et al., 2007). In this problem we are given the set of documents in the source language. For a small fraction of source documents we also know their translation into the target language. The goal is to translate the rest of the documents from source into target language. In this problem the examples are the individual words in the source language and the labels are their translations in the target language.

## 1.3 Thesis Structure

The thesis is organized in the following way. In Chapter 2 we give a formal definition of transductive learning model and survey the existing theoretical and empirical results in transductive learning. This survey includes the survey of our own results appearing in Chapters 3-6 of the thesis. In Chapter 3 we develop concentration inequalities for functions over random partitions of finite set of elements. These concentration inequalities are the basis for the risk bounds developed in Chapters 4 and 5. In Chapter 4 we develop transductive risk bounds that are based on the transductive notion of uniform and weak stability. We also show stability bounds for several transductive algorithms. In Chapter 5 we develop transductive risk bounds that are based on the transductive Rademacher complexity. We also show a method of bounding Rademacher complexity of any transductive algorithm. The results of Chapters 3-5 are surveyed in Section 2.1.2. In Chapter 6 we develop a novel transductive algorithm that implements a large volume learning principle. This algorithm is also surveyed in Section 2.2.4.

The results of Chapter 3 were published in (El-Yaniv & Pechyony, 2006, 2007). The risk bounds of Chapters 4 and 5 were published in (El-Yaniv & Pechyony, 2006) and (El-Yaniv & Pechyony, 2007) respectfully. Finally, the results of Chapter 6 were published in (El-Yaniv et al., 2008).

# Chapter 2

## Survey of Transductive Learning

### 2.1 Theory of Transductive Learning

#### 2.1.1 Two models of transductive learning

Let  $\mathcal{X}$  be a space of unlabeled examples (e.g.,  $d$ -dimensional vectors in  $\mathbb{R}^d$ ) and  $\mathcal{Y}$  be a space of class labels. We denote by  $\langle x, y \rangle \in \mathcal{X} \times \mathcal{Y}$  a tuple of unlabeled example  $x$  along with its label  $y$ . Transductive learning algorithm accepts as an input a *labeled training set*  $\{\langle x_i, y_i \rangle\}_{i=1}^m$  and an *unlabeled test set*  $\{x_i\}_{i=m+1}^{m+u}$ . We denote by  $\mathcal{A}(x_i) \in \mathcal{Y}$ ,  $1 \leq i \leq m+u$ , the labeling given to example  $x_i$  by transductive learner  $\mathcal{A}$ . Let  $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$  be a fixed loss function. The goal of  $\mathcal{A}$  is to generate an accurate labeling of the unlabeled test examples so as to minimize the *test error*

$$\mathcal{L}_u(\mathcal{A}) \triangleq \frac{1}{u} \sum_{i=m+1}^{m+u} \ell(\mathcal{A}(x_i), y_i) ,$$

w.r.t.  $\ell$ . Unless otherwise stated we assume that  $\ell$  is a 0/1 loss function. Two commonly used settings for generating training and test sets are the ones defined by Vapnik (1982).

#### **Setting 1**

- (i) A full sample  $S_{m+u} \triangleq \{\langle x_i, y_i \rangle\}_{i=1}^{m+u}$  of  $m+u$  labeled examples is fixed.
- (ii) Training examples are sampled uniformly without replacement from  $S_{m+u}$ . We assume w.l.o.g. that the sampled training examples have indices from 1 to  $m$ , namely the training set is  $S_m \triangleq \{\langle x_i, y_i \rangle\}_{i=1}^m$ . The induced labeled test examples are  $S_u \triangleq S_{m+u} \setminus S_m$ . The input of transductive learner is the labeled training set  $S_m$  and the unlabeled test set  $X_u \triangleq \{x_i\}_{i=m+1}^{m+u}$ .

#### **Setting 2**

- (i) Let  $\mathcal{D}$  be a distribution over  $\mathcal{X} \times \mathcal{Y}$  with probability density function  $\mu(x, y)$ . Labeled training and test examples,  $S_m \triangleq \{(x_i, y_i)\}_{i=1}^{m+u}$  and  $S_u \triangleq \{(x_i, y_i)\}_{i=m+1}^{m+u}$ , respectively, are sampled independently from  $\mathcal{D}^{m+u}$ .
- (ii) The learning algorithm is given the training set  $S_m$  and the test set  $X_u \triangleq \{x_i\}_{i=m+1}^{m+u}$ .

In both settings each example  $x_i$  from the full sample  $S_{m+u}$  has a unique label  $y_i$ . However, both models allow the examples  $x_i$  and  $x_j$ ,  $i \neq j$ , to be the same but to have different labels  $y_i$  and  $y_j$ . Thus, without any assumption about the full sample  $S_{m+u}$  (in Setting 1) and about the distribution  $\mathcal{D}$  (in Setting 2), the test error  $\mathcal{L}_u(\mathcal{A})$  can be as large as  $1/2$  for any transductive learner  $\mathcal{A}$ .

Settings 1 and 2 have a number of differences and similarities. In both settings training and test sets are generated by sampling from some distribution. In Setting 1 this distribution is a uniform distribution and thus is known to the learner. In this regard Setting 2 is harder, since here the learner does not know the underlying distribution. Another difference between the settings is in the interdependence between examples. While in Setting 2 they are independent, in Setting 1, because of the sampling without replacement, they are dependent.

Settings 1 and 2 are concerned with a passive offline model of learning. In this model the learner has no influence on the contents of the training set, knows the entire training set at the beginning of the learning process and needs only to predict the labels of the unlabeled examples. Other transductive learning models, with different mechanisms of training set generation and different learning goals, are considered in Section 2.1.7.

The definitions of Settings 1 and 2 are very general and can be further instantiated to obtain settings for particular models of passive transductive learning. For example, if  $\mathcal{Y} = \{\pm 1\}$  and  $\ell$  is a 0/1 loss, then we obtain a binary classification setting. With  $\mathcal{Y} \subseteq \mathbb{R}$  and  $\ell$  being a squared loss we obtain a setting of regression. Finally,  $\mathcal{Y}$  can be some (possibly infinite) set of objects, in which case we obtain the setting of structured predictions. In this survey, unless otherwise stated, we assume a binary classification setting. Occasionally we also mention several results for the regression setting.

We now give examples for two applications, each one employing different transductive settings. Suppose we want to classify a large collection of documents. We may randomly choose a small subset of documents and give them to an expert for manual classification. Using this classification we may attempt to classify the rest of the documents. This application corresponds to Setting 1.

The example of application of Setting 2 is the following news classification problem. Suppose we have labeled news messages that have arrived during the last week and our goal is to label the messages that will arrive today. At the end of the day, after accumulating today's news, we may transductively classify them.

The notable difference between these two examples is that in the first one the learner observes the entire unlabeled full sample  $\{x_i\}_{i=1}^{m+u}$  before obtaining the actual training/test set partition. In contrast, in the second example the learner has no idea about the test points before actually obtaining them. This difference is crucial. The learner in Setting 2 can define its hypothesis space only in a data-independent way, before observing any (even unlabeled) example from the full sample  $S_{m+u}$ . On the other hand, due to the ability to observe the unlabeled full sample before obtaining the actual training/test partition, the learner in Setting 1 is allowed to define its hypothesis space in a data-dependent way. Such a data-dependent definition of hypothesis space can potentially tighten transductive risk bounds. This feature of Setting 1 is considered further in Section 2.1.2.

### 2.1.2 Upper risk bounds

Upper risk bounds (or simply risk bounds) bound the test error of transductive algorithms. These bounds can be used in two ways:

1. Risk bounds provide an estimate of the test error of existing transductive algorithms. Hence by comparing the values of the risk bounds one can choose which algorithm (or which hyperparameter) is preferable.
2. Risk bounds can guide the design of new learning algorithms. Commonly these algorithms minimize the objective function that is related to the expressions appearing in the risk bound.

In later sections we give examples of these two ways of using risk bounds.

Risk bounds commonly depend on the probabilistic model (e.g., Setting 1 or Setting 2) for generating training and test sets. The following lemma (proved in Appendix 2.3), which is a slight generalization of Theorem 10.1 in (Vapnik, 1982), shows that many risk bounds for Setting 1 hold also in Setting 2.

**Lemma 1** *Let  $\mathcal{U}(S_{m+u})$  be a uniform distribution over partitions of  $S_{m+u}$  to labeled training set  $S_m$  and unlabeled test set  $X_u$ .*

1. *Suppose we have a probabilistic risk bound for Setting 1, stating that for any  $S_{m+u}$  and  $\delta > 0$ ,*

$$\mathbf{P}_{(S_m, X_u) \sim \mathcal{U}(S_{m+u})} \{ \mathcal{L}_u(\mathcal{A}) \leq B_1(\delta) \} \geq 1 - \delta . \quad (2.1)$$

*Then, in Setting 2, for any  $\delta > 0$ , the risk bound*

$$\mathbf{P}_{(S_m, X_u) \sim \mathcal{D}^{m+u}} \{ \mathcal{L}_u(\mathcal{A}) \leq B_1(\delta) \} \geq 1 - \delta \quad (2.2)$$

*holds true.*

2. Suppose we have an average risk bound for Setting 1, stating that for any  $S_{m+y}$ ,

$$\mathbf{E}_{(S_m, X_u) \sim \mathcal{U}(S_{m+u})} \mathcal{L}_u(\mathcal{A}) \leq B_2 . \quad (2.3)$$

Then, in Setting 2, the risk bound

$$\mathbf{E}_{(S_m, X_u) \sim \mathcal{D}^{m+u}} \mathcal{L}_u(\mathcal{A}) \leq B_2$$

holds true.

We note that all known transductive risk bounds can be attributed to either the first or the second part of Lemma 1.

The consequence of Lemma 1 is that in order to develop risk bounds for both transductive settings it is sufficient to develop the bounds for Setting 1. This argument guided the development of transductive risk bounds over the last few years. As a result, all existing transductive risk bounds are first developed for Setting 1 and then ‘automatically’ translated to Setting 2. The problem of developing (potentially tighter) bounds for Setting 2 directly, without the use of Lemma 1, is open. In the next sections we survey a number of approaches for developing risk bounds for Setting 1.

The forthcoming risk bounds consider two types of transductive algorithms:

**Type 1** The output of transductive algorithm  $\mathcal{A}$  is a *hard classification* vector  $\mathbf{h} \in \{\pm 1\}^{m+u}$ . The  $i$ th component  $h_i$  of  $\mathbf{h}$  is a binary classification of  $x_i$ .

**Type 2** The output of transductive algorithm  $\mathcal{A}$  is a *soft classification* vector  $\mathbf{h} \in [-1, 1]^{m+u}$ . The  $i$ th component  $h_i \in [-1, 1]$  of  $\mathbf{h}$  is a soft classification of  $x_i$ . The binary classification of  $x_i$  is  $\text{sign}(h_i)$ .

Note that in our notation, if  $\mathbf{h}$  is an output of  $\mathcal{A}$ , then  $h_i$  and  $\mathcal{A}(x_i)$  refer to the same value. In later sections we use both these terms interchangeably.

We define the *actual hypothesis space*  $\mathcal{H}_{\mathcal{A}}$  as a set of vectors  $\mathbf{h}$  generated by  $\mathcal{A}$  when operated on all possible training/test set partitions. The prominent feature of risk bounds in Setting 1 is that they allow the definition of  $\mathcal{H}_{\mathcal{A}}$  to be dependent on the unlabeled full sample  $X_{m+u}$ . However, the definition of  $\mathcal{H}_{\mathcal{A}}$  should still be independent of the actual training/test partition and the labels  $Y_{m+u}$ . Hence, unless we have very strong prior knowledge about the labels  $Y_{m+u}$ , it is very hard to identify the actual hypothesis space. To overcome this issue, the risk bounds are commonly used with some superset of  $\mathcal{H}_{\mathcal{A}}$ . We refer to this superset as the *hypothesis space*  $\mathcal{H}$ . The choice of  $\mathcal{H}$  depends on the structure of  $\mathcal{A}$  and reflects our prior knowledge about the domain  $\mathcal{X} \times \mathcal{Y}$ .

The development of many transductive risk bounds was inspired by the existence of the similar bounds in the inductive setting. For example, transductive PAC-Bayesian bounds were inspired by (McAllester, 2003), transductive stability bounds were inspired by (Bousquet & Elisseeff, 2002), and transductive bounds

based on Rademacher complexity were inspired by (Bartlett & Mendelson, 2002)). We note that there are a number of technical and conceptual differences between transductive risk bounds (in Setting 1) and their inductive counterparts:

1. Inductive risk bounds assume that training examples are independent. This assumption does not hold in Setting 1 and hence transductive risk bounds assume that training examples are dependent.
2. Inductive risk bounds bound the generalization error, which is inductive analogue of the *full sample error*  $\frac{1}{m+u} \sum_{i=1}^{m+u} \ell(\mathcal{A}(x_i), y_i)$ . The generalization error counts the errors over the training examples. In contrast, transductive risk bounds bound the test error  $\frac{1}{u} \sum_{i=u+1}^{m+u} \ell(\mathcal{A}(x_i), y_i)$ , which does not count the errors over the test examples.
3. As indicated above, transductive risk bounds allow the hypothesis space to be defined in a data-dependent way. This option is absent in the inductive bounds.

Almost all forthcoming risk bounds have the following form: for any hypothesis space  $\mathcal{H}$ , any full sample  $S_{m+u}$  and any  $\delta > 0$ , with probability of at least  $1 - \delta$  over the random training/test set partition, for any  $\mathbf{h} \in \mathcal{H}$ ,

$$\mathcal{L}_u(\mathbf{h}) \leq \mathcal{L}_m(\mathbf{h}) + \text{slack term}(m, u, \delta, \mathcal{H}) , \quad (2.4)$$

where the slack term depends on  $m$ ,  $u$ ,  $\delta$  and  $\mathcal{H}$ . In some risk bounds the slack term also depends on  $\mathcal{L}_m(\mathbf{h})$ . In all risk bounds of the form (2.5), under some conditions the slack term converges to zero as  $m$  and  $u$  increase. We refer to the bounds of the form (2.5) as *absolute risk bounds*.

Since  $\mathcal{H}_{\mathcal{A}} \subseteq \mathcal{H}$ , the bound (2.4) implies that the bound

$$\mathcal{L}_u(\mathcal{A}) \leq \mathcal{L}_m(\mathcal{A}) + \text{slack term}(m, u, \delta, \mathcal{H}) , \quad (2.5)$$

holds under the same conditions as (2.4).

Without any assumption on the hypothesis space<sup>1</sup> the rate of convergence of the slack term in the bounds in this section is  $\frac{1}{\sqrt{\min(m,u)}}$  or  $\max\left(\frac{1}{\min(m,u)}, \sqrt{\frac{\mathcal{L}_m(\mathcal{A})}{\min(m,u)}}\right)$ . Hence the rate of convergence is proportional to  $\min(m, u)$ . At first sight, this fact seems to be surprising, at least for the case of  $m \gg u$ . Indeed, in this case we have a lot of labeled information in the training set and need to perform a very small number of predictions. Thus this setting looks easy for the learner and it seems that the risk bound should converge very

---

<sup>1</sup>An example of such an assumption is that the hypothesis space contains an hypothesis with a zero error on the full sample. We consider such a *realizable case* towards the end of this section.

quickly. On the other hand, the slack term in (2.5) can be considered as a confidence interval in the estimation of the test error by the training error. If  $u \ll m$ , then the mean  $\tau$  (i.e., the test error) of  $u$  elements, drawn from  $m + u$  elements, has a large variance. Hence, in this case any high-confidence interval (and thus the slack term) for the estimation of  $\tau$  will be large.

The risk bounds can be largely divided into two groups: implicit and explicit. Implicit risk bounds contain terms that are given via a computational procedure, while explicit bounds consist only of closed-form expressions. Hence the ‘language’ of implicit bounds is more powerful than that of explicit bounds, and consequently implicit bounds are tighter than explicit ones. We start with surveying implicit bounds and then proceed to the various types of explicit ones.

## Implicit bounds

The first transductive implicit bound was published by Vapnik (1982). This bound is based on the following observation. Let  $\mathcal{A}$  be a transductive algorithm of Type 1 and  $\mathbf{h} \in \{\pm 1\}^{m+u}$  be a fixed hard labeling of the full sample  $X_{m+u}$ . Let  $e_m(\mathbf{h})$  be a number of errors made by  $\mathbf{h}$  on the randomly chosen (without replacement)  $m$  training examples, and  $e_u(\mathbf{h})$  be a number of errors made by  $\mathbf{h}$  on the induced test examples.  $e_m(\mathbf{h})$  is a random variable with hypergeometric distribution. Hence for any  $k > 0$  we can compute directly the probability mass of the event,

$$|e_m(\mathbf{h}) - e_u(\mathbf{h})| > k . \quad (2.6)$$

The labeling vector  $\mathbf{h}$  is the output transductive algorithm  $\mathcal{A}$  when it is run on a single training/test set partition. When run on all possible training/test partitions, the algorithm  $\mathcal{A}$  will generate a number of different  $\mathbf{h}$ 's that constitute the actual hypothesis space  $\mathcal{H}_{\mathcal{A}}$ . Hence to obtain the risk bound for  $\mathcal{A}$  we should bound the probability mass of the event (2.6) uniformly for all  $\mathbf{h} \in \mathcal{H}_{\mathcal{A}}$ . The resulting risk bound (which was stated explicitly by El-Yaniv and Gerzon (2005)) has the following form: for any  $\delta > 0$ , with probability of at least  $1 - \delta$  over the random training/test set partitioning,

$$\mathcal{L}_u(\mathcal{A}) \leq \mathcal{L}_m(\mathcal{A}) + \varepsilon_a , \quad (2.7)$$

where  $\varepsilon_a$  is proportional to the size of hypothesis space  $\mathcal{H}$  (such that  $\mathcal{H}_{\mathcal{A}} \subseteq \mathcal{H}$ ) and depends on the properties of the hypergeometric distribution of  $e_m(\mathbf{h})$ . The value of  $\varepsilon_a$  can be obtained by an efficient computational routine. The bound (2.7) can be improved (see the analysis at the end of the next section) if instead of the event of absolute deviation in (2.6) we consider the event of the relative deviation:

$$\frac{|e_m(\mathbf{h}) - e_u(\mathbf{h})|}{\sqrt{e_{m+u}(\mathbf{h})}} > k , \quad (2.8)$$

where  $e_{m+u}(\mathbf{h})$  is the number of errors made by  $\mathbf{h}$  on the full sample  $X_{m+u}$ . In this case the resulting risk bound states that for any  $\delta > 0$ , with probability of at least  $1 - \delta$  over the random training/test set partitioning,

$$\mathcal{L}_u(\mathcal{A}) \leq \mathcal{L}_m(\mathcal{A}) + \frac{\varepsilon_r^2 u}{2(m+u)} + \varepsilon_r \sqrt{\mathcal{L}_m(\mathcal{A}) + \left(\frac{u\varepsilon_r}{2(m+u)}\right)^2}, \quad (2.9)$$

where  $\varepsilon_r$  is proportional to the size of hypothesis space  $\mathcal{H}$  (such that  $\mathcal{H}_{\mathcal{A}} \subseteq \mathcal{H}$ ) and depends on the properties of the hypergeometric distribution of  $e_m(\mathbf{h})$ . As  $\varepsilon_r$ , the value of  $\varepsilon_r$  can be obtained by an efficient computational routine. Blum and Langford (2003) derived a similar implicit bound that is slightly tighter than (2.9).

The recent implicit risk bound of Bax and Callejas (2008) has a different form. Let  $\tilde{y} \in \{\pm 1\}^u$  be a possible labeling of test examples  $X_u$ . Let  $S_{m+u}(\tilde{y})$  be the full sample  $S_{m+u}$ , with the original test labels  $Y_u$  being replaced by  $\tilde{y}$ . Moreover, let  $\mathcal{L}_u(\mathcal{A}, (S'_m, X'_u))$  be the test error of  $\mathcal{A}$  when operated on the training/test set partition  $(S'_m, X'_u)$  with the test labels  $\tilde{y}$ . Note that the labels  $\tilde{y}$  are of the examples  $X_u$ . Under partition  $(S'_m, X'_u)$ , some of the examples in  $X_u$  will also be in  $S'_m$  and for these examples the corresponding entries of  $\tilde{y}$  will be taken as training labels. For any  $\delta > 0$ , let  $\tilde{Y}$  be a set of labelings of  $X_u$  for which the test error of  $\mathcal{A}$ , when operated on  $(S_m, X_u)$ , is smaller than the one when  $\mathcal{A}$  operates on many other training/test set partitions:

$$\tilde{Y} = \left\{ \tilde{y} \in \{\pm 1\}^u \mid \mathbf{P}_{(S'_m, X'_u) \sim \mathcal{U}(S_{m+u}(\tilde{y}))} \{ \mathcal{L}_u(\mathcal{A}, (S'_m, X'_u)) \geq \mathcal{L}_u(\mathcal{A}, \tilde{y}) \} > \delta \right\}. \quad (2.10)$$

The bound of Bax and Callejas states that for any  $\delta > 0$ , with probability of at least  $1 - \delta$  over training/test set partition,

$$\mathcal{L}_u(\mathcal{A}) \leq \max_{\tilde{y} \in \tilde{Y}} \mathcal{L}_u(\mathcal{A}, \tilde{y}). \quad (2.11)$$

In general, the computation of the maximum in (2.11) takes exponential time, since according to (2.10) the size of  $\tilde{Y}$  is exponential in  $u$ . Bax and Callejas showed an efficient procedure (having polynomial complexity) that computes the value of the maximum in (2.11) when  $\mathcal{A}$  is a 1-nearest neighbor classifier. The challenging open question is to generalize this result to more powerful transductive learning algorithms and then to compare (2.11) with the implicit bounds of Vapnik (1982) and Blum and Langford (2003).

### Bounds based on VC-dimension

As mentioned previously in the description of the implicit bound of Vapnik (1982), the risk bound should hold uniformly for all hypotheses in the (actual) hypothesis space. Consequently the resulting risk bounds (e.g., (2.9)) depend (either explicitly or implicitly) on the size of the hypothesis space. Several measures of the size

of the hypothesis space have been developed during the last decades. The most straightforward one, the number of hypotheses in the hypothesis space, is used in (2.9). Unfortunately, in many applications it is hard to compute this number exactly and one has to resort to some technique of upper bounding the size of the hypothesis space. Examples of bounds on the size of various transductive hypothesis spaces can be found in (Derbeko et al., 2004; Hanneke, 2006; Pelckmans et al., 2007). The bounding methods used in these papers are ad-hoc and rely heavily on the structure of the hypothesis space.

The first known general technique for bounding the size of the hypothesis space is the one using the VC-dimension. This technique traces its roots to the seminal paper of Vapnik and Chervonenkis (1971) and is also widely used in inductive risk bounds. The definition of the VC-dimension that we give below is an adaptation of the original definition of VC-dimension to the transductive setting. A similar, but less general definition of the VC dimension in the transductive setting has appeared in (Pelckmans et al., 2006) under the name ‘Kingdom capacity of graph’.

The VC-dimension technique considers hard classification hypotheses<sup>2</sup>  $\mathbf{h}$ , which potentially can be generated by transductive algorithms of Type 1. We say that the set of examples  $x_1, \dots, x_n \in X_{m+u}$  is shattered by  $\mathcal{H}$  if the hypotheses in  $\mathcal{H}$  label these examples in all possible  $2^n$  ways. The VC-dimension of  $\mathcal{H}$  is a maximal number  $n$  such that there exist  $n$  points  $x_1, \dots, x_n \in X_{m+u}$  that are shattered by  $\mathcal{H}$ . A remarkable result of Vapnik and Chervonenkis (1971), when adapted to our setting, states that

$$|\mathcal{H}_{\mathcal{A}}| \leq \left( \frac{(m+u)e}{d} \right)^d . \quad (2.12)$$

By using (2.12) and explicitly bounding  $\varepsilon_a$  in (2.7) and  $\varepsilon_r$  in (2.9), Cortes and Mohri (2007) obtained<sup>3</sup> that for any  $\delta > 0$ , with probability of at least  $1 - \delta$  over the training/test set partition,

$$\mathcal{L}_u(\mathcal{A}) \leq \mathcal{L}_m(\mathcal{A}) + O \left( \sqrt{\left( \frac{1}{m} + \frac{1}{u} \right) d \log(m+u) \ln \frac{1}{\delta}} \right) \quad (2.13)$$

and

$$\begin{aligned} \mathcal{L}_u(\mathcal{A}) \leq & \mathcal{L}_m(\mathcal{A}) + O \left( \left( \frac{1}{m} + \frac{1}{u} \right) d \log(m+u) \ln \frac{1}{\delta} \right) \\ & + O \left( \sqrt{\mathcal{L}_m(\mathcal{A}) \left( \frac{1}{m} + \frac{1}{u} \right) d \log(m+u) \ln \frac{1}{\delta}} \right) . \end{aligned} \quad (2.14)$$

---

<sup>2</sup>The definition of the VC-dimension can also be extended to soft classification vectors; see Section 5.2.3 in (Vapnik, 1998).

<sup>3</sup>The bound (2.13) does not appear in (Cortes & Mohri, 2007) but can be immediately obtained using the results of that paper.

The weaker versions of (2.13) and (2.14), for the case of  $m = u$ , appeared in (Bottou et al., 1994).

Let  $\beta \triangleq \frac{d \log(m+u)}{\min(m,u)}$ . If  $\log(\max(m, u)) < \min(m, u)$ , then as  $m$  and  $u$  increase, the second summand in (2.13) converges to zero with the rate  $\sqrt{\beta}$ , and the second and third summands in (2.14) converge to zero with the rates  $\beta$  and  $\sqrt{\mathcal{L}_m(\mathcal{A})\beta}$ , respectively. Hence the bound (2.13) converges with the rate  $\sqrt{\beta}$ . The rate of convergence of (2.14) is controlled by the empirical error  $\mathcal{L}_m(\mathcal{A})$ . If  $\mathcal{L}_m(\mathcal{A})$  is very small (e.g., of the order of  $\beta$ ), then (2.14) converges with the fast rate of  $\beta$ . Otherwise it converges with the slower rate of  $\sqrt{\beta}$ . Consequently, for large  $m$  and  $u$  the bound (2.14) is tighter than (2.13) if the empirical error  $\mathcal{L}_m(\mathcal{A})$  is very small.

Any transductive hypothesis space  $\mathcal{H}$  can be represented in a functional form, namely there exists a space of functions  $\mathcal{F}$  from  $X_{m+u}$  to  $\pm 1$  such that for any  $\mathbf{h} \in \mathcal{H}$  there exists  $f \in \mathcal{F}$  such that for any  $1 \leq i \leq m + u$ ,  $h_i = f(x_i)$ . The examples of  $\mathcal{F}$  are hyperplanes and polynomials. The VC dimension of transductive hypothesis space  $\mathcal{H}$  is the same as the one of its functional representation. Hence if the functional representation  $\mathcal{F}$  of  $\mathcal{H}$  is known, then many of the results (e.g., see Vapnik, 1998) from bounding the VC-dimension of the function spaces can be applied to bound the VC-dimension of  $\mathcal{H}$  in both explicit (with analytical expressions) and implicit (with computational routine) ways. We refer to such bounds on the VC-dimension of  $\mathcal{H}$  as indirect VC bounds.

If the functional form of  $\mathcal{H}$  is not known (or is very complicated), then one can try to bound the VC-dimension of  $\mathcal{H}$  directly. Pelckmans et al. (2006) showed an implicit way of directly bounding the VC-dimension of the transductive hypothesis space consisting of the ‘smooth’ labelings of the full sample. Currently, there are no direct explicit bounds on the VC-dimension of transductive hypothesis spaces.

## PAC-Bayesian bounds

PAC-Bayesian (or Occam) risk bounds, introduced in the transductive context by Blum and Langford (2003) and Derbeko et al. (2004), bound the test error of the transductive algorithm in terms of its empirical error and the prior probability  $p(\mathbf{h})$  of the hypothesis  $\mathbf{h} \in \mathcal{H}_{\mathcal{A}}$  generated by  $\mathcal{A}$ . In this section, unless otherwise stated, we assume that  $\mathcal{A}$  is of Type 1. The definition of the prior probability can depend on the unlabeled full sample  $X_{m+u}$ , but it should be independent of the training/test set partition and the labels  $Y_{m+u}$ . As stated previously, in general it is very hard to identify the space  $\mathcal{H}_{\mathcal{A}}$  precisely. Hence, instead of defining a prior  $p(\cdot)$  over the hypotheses in  $\mathcal{H}_{\mathcal{A}}$ , it is common to define a prior over a superset  $\mathcal{H}$  of  $\mathcal{H}_{\mathcal{A}}$ .

The derivation of implicit and explicit PAC-Bayesian bounds for general transductive algorithms is almost identical to the derivation of the bounds in (2.7), (2.9), (2.13) and (2.14). The implicit PAC-Bayesian bounds state that for any

$\delta > 0$ , with probability of at least  $1 - \delta$  over the random training/test set partitioning,

$$\mathcal{L}_u(\mathcal{A}) \leq \mathcal{L}_m(\mathcal{A}) + \varepsilon_a, \quad (2.15)$$

and

$$\mathcal{L}_u(\mathcal{A}) \leq \mathcal{L}_m(\mathcal{A}) + \frac{\varepsilon_r^2 u}{2(m+u)} + \varepsilon_r \sqrt{\mathcal{L}_m(\mathcal{A}) + \left(\frac{u\varepsilon_r}{2(m+u)}\right)^2}, \quad (2.16)$$

where both  $\varepsilon_a$  and  $\varepsilon_r$  are inversely proportional to the prior probability  $p(\mathbf{h})$  of the hypothesis generated by  $\mathcal{A}$  and depend on the properties of the hypergeometric distribution of  $e_m(\mathbf{h})$ . The bounds (2.15) and (2.16) were introduced by El-Yaniv and Gerzon (2005) and Derbeko et al. (2004) respectively. Similarly, explicit PAC-Bayesian bounds state that for any  $\delta > 0$ , with probability of at least  $1 - \delta$  over the random training/test set partitioning,

$$\mathcal{L}_u(\mathcal{A}) \leq \mathcal{L}_m(\mathcal{A}) + O\left(\sqrt{\left(\frac{1}{m} + \frac{1}{u}\right) \ln \frac{1}{p(\mathbf{h})\delta}}\right) \quad (2.17)$$

and

$$\mathcal{L}_u(\mathcal{A}) \leq \mathcal{L}_m(\mathcal{A}) + O\left(\left(\frac{1}{m} + \frac{1}{u}\right) \ln \frac{1}{p(\mathbf{h})\delta}\right) + O\left(\sqrt{\mathcal{L}_m(\mathcal{A}) \left(\frac{1}{m} + \frac{1}{u}\right) \ln \frac{1}{p(\mathbf{h})\delta}}\right). \quad (2.18)$$

The bound (2.17) appeared in (Derbeko et al., 2004), while the bound (2.18) is a novel one (although its derivation is almost the same as that of (2.14)).<sup>4</sup>

The tightness of the bounds (2.15)-(2.18) depends on the quality of the prior  $p(\mathbf{h})$ . If the prior is good, e.g.  $\ln \frac{1}{p(\mathbf{h})} = O(1)$  for any  $\mathbf{h} \in \mathcal{H}_{\mathcal{A}}$ , then the bounds (2.17) and (2.18) converge with the rates of  $\frac{1}{\sqrt{\min(m,u)}}$  and  $\max\left(\frac{1}{\min(m,u)}, \sqrt{\frac{\mathcal{L}_m(\mathcal{A})}{\min(m,u)}}\right)$ , respectively. These rates improve by factor  $\log(m+u)$  the corresponding rates of the bounds (2.13) and (2.14), which are based on the VC-dimension.

Several types of prior  $p(\mathbf{h})$ 's were introduced by Derbeko et al. (2004), El-Yaniv and Gerzon (2005) and Hanneke (2006). The compression prior  $p_{\text{comp}}(\mathbf{h})$ , defined in (Derbeko et al., 2004), gives a larger prior to hypotheses that can be obtained by operating  $\mathcal{A}$  on small training sets. Formally, if  $\mathbf{h}^{(1)}$  is obtained by operating  $\mathcal{A}$  on partition  $(S_{m_1}, X_{u_1})$ , and  $\mathbf{h}^{(2)}$  is obtained by operating  $\mathcal{A}$  on partition  $(S_{m_2}, X_{u_2})$  and  $m_1 < m_2$ , then  $p_{\text{comp}}(\mathbf{h}^{(1)}) > p_{\text{comp}}(\mathbf{h}^{(2)})$ . The clustering prior  $p_{\text{clust}}(\mathbf{h})$ , defined in (Derbeko et al., 2004) and (El-Yaniv &

---

<sup>4</sup>A bound similar to (2.18) appeared in (Derbeko et al., 2004). However, there it contains a  $O(\log(m+u))$  factor in the second and third summands in (2.18) and thus is weaker than (2.18).

Gerzon, 2005), gives a larger prior to hypotheses that can be clustered into a small number of clusters by some fixed clustering algorithm. This prior actually implies the transductive learning algorithm that is described in Section 2.2.7. Finally, the cut prior  $p_{\text{cut}}(\mathbf{h})$ , defined in (Hanneke, 2006), gives a larger prior to the hypotheses that induce small cuts on some fixed graph  $G$ . This prior justifies the cut-based transductive algorithms described in Section 2.2.2.

The bounds (2.15)-(2.18) are valid for any transductive algorithm  $\mathcal{A}$ . We now present two PAC-Bayesian bounds for transductive algorithms having some particular structure. These bounds assume that  $\mathcal{A}$  is of Type 2.

Let  $Q$  be a distribution over  $\mathcal{H}$  with probability density  $q(\cdot)$ . If for the classification of example  $x_i \in X_{m+u}$ , transductive algorithm  $\mathcal{A}$  picks randomly, according to  $Q$ , a hypothesis  $\mathbf{h} \in \mathcal{H}$  and classifies  $x_i$  as  $h_i$ , then we refer to  $\mathcal{A}$  as a *Gibbs algorithm* and denote it by  $\mathcal{A}_Q^{(G)}$ . The distribution  $Q$  can depend on all available information, including the training/test partition and the training labels  $Y_m$ . Hence we refer to  $Q$  as a *posterior distribution*. Let  $P$  be a *prior distribution* (with probability density  $p(\cdot)$ ) over  $\mathcal{H}$  and  $KL(Q\|P) \triangleq \int_{\mathbf{h} \in \mathcal{H}} q(\mathbf{h}) \log \frac{q(\mathbf{h})}{p(\mathbf{h})} d\mathbf{h}$  be a Kullback-Leibler divergence between  $Q$  and  $P$ . Derbeko et al. (2004) showed that for any  $\delta > 0$ , with probability of at least  $1 - \delta$  over the random training/test set partitioning,

$$\begin{aligned} \mathcal{L}_u \left( \mathcal{A}_Q^{(G)} \right) &\leq \mathcal{L}_m \left( \mathcal{A}_Q^{(G)} \right) + O \left( \frac{KL(Q\|P) + \log \left( \frac{m+u}{\delta} \right)}{m} \right) \\ &\quad + O \left( \sqrt{\mathcal{L}_m \left( \mathcal{A}_Q^{(G)} \right) \left( \frac{1}{m} + \frac{1}{u} \right) \left( KL(Q\|P) + \log \left( \frac{m+u}{\delta} \right) \right)} \right). \end{aligned} \quad (2.19)$$

If we set  $q(\mathbf{h}) = 1$  for some  $\mathbf{h} \in \mathcal{H}$  and  $q(\mathbf{h}') = 0$  for all other hypotheses  $\mathbf{h}'$ , then  $KL(Q\|P) = \ln \frac{1}{p(\mathbf{h})}$  and (2.19) recovers (2.18) up to the factor of  $O(\log(m+u))$ . In general, the difference between (2.19) and (2.18) is that while (2.18) depends of the prior probability of a single hypothesis  $\mathbf{h}$ , the bound (2.19) depends on the divergence between prior and posterior probability distributions. Thus (2.19) is more robust than (2.18) in the sense that (2.19) is less sensitive to a bad prior on a single hypothesis. However, the robustness of (2.19) comes at the price of multiplicative factor  $O(\log(m+u))$  at the second and third summands of (2.19).

Since the posterior distribution  $Q$  can be dependent on the training/test set partition and the training labels, we can choose  $Q$  minimizing (2.19). The naïve choice of  $Q$  would be  $Q \triangleq P$ . In this case  $KL(Q\|P) = 0$ . However, if  $Q = P$ , then  $Q$  is independent of the actual training/test set partition and training labels. As a result, the training error  $\mathcal{L}_m \left( \mathcal{A}_Q^{(G)} \right)$  may be very large. Hence to minimize (2.19) the posterior distribution  $Q$  should balance between minimization of the divergence with the prior distribution  $P$  and minimization of the training error  $\mathcal{L}_m \left( \mathcal{A}_Q^{(G)} \right)$ .

Let  $\tilde{\mathbf{h}} \triangleq \int_{\mathbf{h} \in \mathcal{H}} q(\mathbf{h}) \mathbf{h} d\mathbf{h} \in [-1, 1]^{m+u}$ . If for any  $x_i \in X_{m+u}$ ,  $\mathcal{A}(x_i) = \tilde{h}_i$ , namely the label given by  $\mathcal{A}$  to  $x_i$  is the weighed sum of the  $i$ th components of all hypotheses in  $\mathcal{H}$ , then we refer to  $\mathcal{A}$  as a *Bayesian mixture algorithm* and denote it by  $\mathcal{A}_Q^{(B)}$ . The PAC-Bayesian risk bound for transductive Bayesian mixtures was introduced by El-Yaniv and Pechyony (2007) and has the following form: for any fixed  $\delta > 0$  and  $\gamma > 0$ , with probability of at least  $1 - \delta$  over the random training/test set partitioning,

$$\mathcal{L}_u \left( \mathcal{A}_Q^{(B)} \right) \leq \mathcal{L}_m^\gamma \left( \mathcal{A}_Q^{(B)} \right) + \frac{1}{\gamma} O \left( \sqrt{\frac{KL(Q\|P) \cdot (\sup_{\mathbf{h} \in \mathcal{H}} \|\mathbf{h}\|_2^2) \cdot \ln \frac{1}{\delta}}{\min(m, u)}}} \right), \quad (2.20)$$

where  $\mathcal{L}_m^\gamma \left( \mathcal{A}_Q^{(B)} \right) \triangleq \frac{1}{m} \sum_{i=1}^m \ell_\gamma(\tilde{h}_i, y_i)$  is an empirical error w.r.t. to the  $\gamma$ -margin loss function  $\ell_\gamma(y_1, y_2) = \min\{1, 1 - y_1 y_2 / \gamma\}$ . The derivation of this bound is based on the technique of Rademacher complexity that is described at the next section.

The parameter  $\gamma$  is related to other terms in the bound, in particular to  $\mathcal{L}_m^\gamma \left( \mathcal{A}_Q^{(B)} \right)$  and  $\sup_{\mathbf{h} \in \mathcal{H}} \|\mathbf{h}\|_2^2$ . As  $\gamma$  increases, the second summand in (2.20) decreases, but the empirical error  $\mathcal{L}_m^\gamma \left( \mathcal{A}_Q^{(B)} \right)$  increases. Moreover, if we scale down all hypotheses in  $\mathcal{H}$  such that  $\sup_{\mathbf{h} \in \mathcal{H}} \|\mathbf{h}\|_2^2$  is small, then to obtain a small empirical error  $\mathcal{L}_m^\gamma \left( \mathcal{A}_Q^{(B)} \right)$ , the value of  $\gamma$  should also be very small. However, the small value of  $\gamma$  will increase the second term of (2.20). Hence it is desirable to choose the value of  $\gamma$  that minimizes (2.20). Using the technique of Bousquet and Elisseeff (2002) it is possible to derive the variant of bound (2.20) that holds uniformly for all possible  $\gamma$ 's. Then the optimal value of  $\gamma$  can be found by minimizing the uniform bound.

If  $\sup_{\mathbf{h} \in \mathcal{H}} \|\mathbf{h}\|_2^2$  remains constant when  $m$  and  $u$  increase, then the bound (2.20) converges with the rate  $\frac{1}{\sqrt{\min(m, u)}}$ . If the empirical error  $\mathcal{L}_m^\gamma \left( \mathcal{A}_Q^{(B)} \right)$  is small, then this rate is slower than the convergence rate  $\max \left( \frac{1}{\min(m, u)}, \sqrt{\frac{\mathcal{L}_m(\mathcal{A})}{\min(m, u)}} \right)$  of the bound (2.19) for a Gibbs distribution. The derivation of the PAC-Bayesian bound for Bayesian mixtures having the latter rate is an open problem. Finally, we note that algorithmic applications of the bounds (2.19) and (2.20) have yet to be developed.

## Bounds based on Rademacher complexity

Rademacher complexity measures the amount of correlation of hypotheses in  $\mathcal{H}$  with the random labeling hypotheses. As does the VC-dimension, Rademacher complexity measures the size of the hypothesis space. Roughly, the hypothesis space is large if any random labeling is highly correlated with some hypothesis in

$\mathcal{H}$ . However, while the VC-dimension provides us with the upper bound (2.12) on the size of the hypothesis space, Rademacher complexity is a rather indirect measure of the size. The transductive variant of Rademacher complexity is defined as follows:

**Definition 1 (El-Yaniv & Pechyony, 2007)** Let  $\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_{m+u})$  be a vector of i.i.d. random variables such that  $\sigma_i = 1$  with probability  $\frac{mu}{(m+u)^2}$ ,  $\sigma_i = -1$  with probability  $\frac{mu}{(m+u)^2}$  and  $\sigma_i = 0$  with probability  $1 - \frac{2mu}{(m+u)^2}$ . The transductive Rademacher complexity is  $R_{m+u}(\mathcal{H}) \triangleq \left(\frac{1}{m} + \frac{1}{u}\right) \cdot \mathbf{E}_{\boldsymbol{\sigma}} \left\{ \sup_{\mathbf{h} \in \mathcal{H}} \boldsymbol{\sigma} \cdot \mathbf{h} \right\}$ .

While the definition of Rademacher complexity is independent of the type of transductive algorithm  $\mathcal{A}$ , in the forthcoming Rademacher-based bounds we assume that  $\mathcal{H}$  is of Type 2. El-Yaniv and Pechyony (2007) derived the following Rademacher-based risk bound: for any  $\delta > 0$ , with probability of at least  $1 - \delta$  over the random training/test set partitioning,

$$\mathcal{L}_u(\mathcal{A}) \leq \mathcal{L}_m^\gamma(\mathcal{A}) + \frac{R_{m+u}(\mathcal{H}_{\mathcal{A}})}{\gamma} + \frac{1}{\gamma} O \left( \sqrt{\frac{\ln \frac{1}{\delta}}{\min(m, u)}} \right), \quad (2.21)$$

where  $\mathcal{L}_m^\gamma(\mathcal{A})$  is a  $\gamma$ -margin training error of the soft hypothesis  $\hat{\mathbf{h}}$  generated by  $\mathcal{A}$ .

The rate of convergence of (2.21) depends largely on the value of transductive Rademacher complexity  $R_{m+u}(\mathcal{H}_{\mathcal{A}})$ . El-Yaniv and Pechyony (2007) developed a technique for bounding the Rademacher complexity of any transductive algorithm. This technique is based on the observation that the output of any transductive algorithm can be represented as

$$\mathbf{h} = U\boldsymbol{\alpha}, \quad (2.22)$$

where  $U$  is an  $(m+u) \times r$  matrix depending only on  $X_{m+u}$  and  $\boldsymbol{\alpha}$  is an  $r \times 1$  vector that may depend on both  $S_m$  and  $X_u$ . We refer to (2.22) as an unlabeled-labeled representation (ULR) of  $\mathcal{A}$ . Note that since  $U$  depends only on  $X_{m+u}$ , each hypothesis  $\mathbf{h} \in \mathcal{H}_{\mathcal{A}}$  is completely characterized by its  $\boldsymbol{\alpha}$ . Let  $\Upsilon(\mathcal{H}_{\mathcal{A}})$  be a set of all  $\boldsymbol{\alpha}$ 's that correspond to all  $\mathbf{h} \in \mathcal{H}_{\mathcal{A}}$ . Given ULR, the Rademacher complexity of  $\mathcal{A}$  can be bounded in terms of several properties of  $U$  and  $\boldsymbol{\alpha}$ . The resulting risk bound states that for any fixed  $\delta > 0$  and  $\gamma > 0$ , with probability of at least  $1 - \delta$  over the random training/test set partitioning,

$$\mathcal{L}_u(\mathcal{A}) \leq \mathcal{L}_m^\gamma(\mathcal{A}) + \frac{\sup_{\boldsymbol{\alpha} \in \Upsilon(\mathcal{H}_{\mathcal{A}})} \|\boldsymbol{\alpha}\|_2}{\gamma} \sqrt{\frac{2}{mu} \sum_{i=1}^r \lambda_i^2} + \frac{1}{\gamma} O \left( \sqrt{\frac{\ln \frac{1}{\delta}}{\min(m, u)}} \right), \quad (2.23)$$

where  $\{\lambda_i\}_{i=1}^r$  are the singular values of the matrix  $U$ . We note that the parameter  $\gamma$  is related (similarly to (2.20)) to other terms of (2.23). In particular,

if  $\sup_{\alpha \in \Upsilon(\mathcal{H}_{\mathcal{A}})} \|\alpha\|_2$  is small, then in order to make  $\mathcal{L}_m^\gamma(\mathcal{A})$  small, the value of  $\gamma$  should be also small.

Any transductive algorithm has an infinite number of ULRs. However, the non-trivial ULRs are the ones minimizing the bound (2.23) or at least resulting in a value of (2.23) less than 0.5. It is shown in (El-Yaniv & Pechyony, 2007) that several graph-based transductive algorithms (discussed in Section 2.2.2) do have nontrivial ULRs.

If  $\sup_{\alpha \in \Upsilon(\mathcal{H}_{\mathcal{A}})} \|\alpha\|_2 = O(\sqrt{m+u})$  is of order of  $\sqrt{m+u}$  and  $\sum_{i=1}^r \lambda_i^2 = O(1)$  (this is a common case in the algorithms considered in (El-Yaniv & Pechyony, 2007)), then the bound (2.23) converges with the rate  $\frac{1}{\min(m,u)}$ . The derivation of a Rademacher-based bound with a faster rate (e.g.,  $\max\left(\frac{1}{\min(m,u)}, \sqrt{\frac{\mathcal{L}_m(\mathcal{A})}{\min(m,u)}}\right)$  as in (2.14)) is an open problem.

## Bounds based on stability

Stability is another way of indirectly measuring the size of the hypothesis space. As in previous sections, we assume that  $\mathcal{A}$  is operated on training/test set partition  $(S_m, X_u)$ . In this section we consider additional run of the algorithm, after the  $i$ th example in the training set and the  $j$ th example in the test set are exchanged. As a result of this exchange the label of the  $i$ th example is hidden and the one of the  $j$ th example is revealed. For the sake of brevity, we overload the notation and denote the former run as  $\mathcal{A}$  and the latter run as  $\mathcal{A}^{ij}$ . In this section we assume that  $\mathcal{A}$  is of Type 2. The *uniform stability* of  $\mathcal{A}$  is

$$\max_{(S_m, X_u), 1 \leq k \leq m+u, 1 \leq i \leq m, m+1 \leq j \leq m+u} |\mathcal{A}(x_k) - \mathcal{A}^{ij}(x_k)| ; \quad (2.24)$$

namely, the worst-case change of a soft classification of any example (from  $X_{m+u}$ ) when a pair of training/test examples is exchanged. We denote by  $\beta$  a bound on the uniform stability. If  $\beta$  is small, then all soft hypotheses generated by  $\mathcal{A}$  are almost the same. In this case the effective number of dichotomies that can be generated by  $\mathcal{A}$  is very small.

Stability can be thought of as a ‘Lipschitz constant’ of  $\mathcal{A}$ . Indeed, if we consider  $\mathcal{A}$  as a function of random training/test set partition  $(S_m, X_u)$  and of the example  $x \in X_{m+u}$ , then (2.24) is a definition of the Lipschitz constant of this function. There exists a large number of *concentration inequalities* (e.g., see Ledoux, 2001) showing that the difference between a function of random variables and its expectation depends on the Lipschitz constant of this function. The forthcoming stability-based risk bounds can be considered as an adaptation of the above general concentration inequalities to the transductive setting.

The following risk bound of El-Yaniv and Pechyony (2006) bounds the 0/1-test error of transductive classification algorithm  $\mathcal{A}$  in terms of its uniform stability: for any fixed  $\delta > 0$  and  $\gamma > 0$ , with probability of at least  $1 - \delta$  over the random

training/test set partitioning,

$$\mathcal{L}_u(\mathcal{A}) \leq \mathcal{L}_m^\gamma(\mathcal{A}) + \frac{1}{\gamma} O\left(\beta \sqrt{\frac{mu \ln \frac{1}{\delta}}{m+u}}\right) + O\left(\sqrt{\left(\frac{1}{m} + \frac{1}{u}\right) \ln \frac{1}{\delta}}\right). \quad (2.25)$$

A similar bound, for the setting of regression with a squared loss, has appeared in (Cortes et al., 2008). In order for the bound (2.25) to converge with the rate  $\frac{1}{\sqrt{\min(m,u)}}$  (as Rademacher-based bounds), the value of  $\beta$  should be  $O\left(\sqrt{\left(\frac{1}{m} + \frac{1}{u}\right) \frac{1}{\min(m,u)}}\right)$ . We show in Chapter 2 of this thesis that such a small  $\beta$  value can be achieved for transductive algorithms performing regularization in reproducing kernel Hilbert space (RKHS). Examples of such algorithms can be found in Section 2.1.3.

The measure of the size of hypothesis space by uniform stability is rather conservative, since it considers the worst case. Another measure of the size of the hypothesis space is by means of transductive *weak stability*, which is defined as follows:

**Definition 2** *The algorithm  $\mathcal{A}$  has weak stability  $(\beta, \beta_1, \beta_2, \delta_1^a, \delta_1^b, \delta_2)$  if its uniform stability is bounded by  $\beta$  and the following conditions hold:*

$$\mathbf{P}_{(S_m, X_u), i \sim I_1^m, j \sim I_{m+1}^{m+u}} \left\{ \mathbf{P}_{x \sim X_{m+u}} \left\{ |\mathcal{A}(x) - \mathcal{A}^{ij}(x)| \leq \beta_1 \right\} \geq 1 - \delta_1^a \right\} \geq 1 - \delta_1^b,$$

$$\mathbf{P}_{(S_m, X_u), i \sim I_1^m, j \sim I_{m+1}^{m+u}} \left\{ |\mathcal{A}(x) - \mathcal{A}^{ij}(x)| \leq \beta_2 \right\} \geq 1 - \delta_2,$$

where  $i \sim I_1^m$  refers to the uniform draw of  $i$  from  $\{1, \dots, m\}$ ,  $j \sim I_{m+1}^{m+u}$  refers to the uniform draw of  $j$  from  $\{m+1, \dots, m+u\}$  and  $x \sim X_{m+u}$  refers to the uniform draw of  $x$  from  $X_{m+u}$ .

The weak stability, as defined above, considers most of the changes of soft classifications of all examples (from  $X_{m+u}$ ) when a pair of training/test examples is exchanged. In opposition to uniform stability, weak stability does not consider “outlier” exchanges of training/test examples that cause large changes in soft classification. As such, the weak stability bounds  $\beta_1$  and  $\beta_2$  may be much smaller than the uniform stability bound  $\beta$ .

The following improved version (derived in Chapter 2 of this thesis) of the risk bound of El-Yaniv and Pechyony (2006) bounds the test error of  $\mathcal{A}$  in terms of its weak stability: for any fixed  $\delta > 0$ ,  $\gamma > 0$  and  $0 < \theta < 1$ , with probability of at least  $(1 - \delta) \left(1 - \frac{\delta_1^b m}{\theta}\right)$  over the random training/test set partitioning,

$$\mathcal{L}_u(\mathcal{A}) \leq \mathcal{L}_m^\gamma(\mathcal{A}) + \left[ (1 - \delta_2) \frac{\beta_2}{\gamma} + \delta_2 \frac{\beta}{\gamma} \right] + \sqrt{\left( (1 - \theta) \tilde{\beta}_1 + \theta \tilde{\beta} \right)^2 \cdot O\left(\frac{mu}{m+u}\right) \cdot \ln \frac{1}{\delta}}, \quad (2.26)$$

where  $\tilde{\beta}_1 = O\left(\frac{\beta_1}{\gamma} + \frac{\delta_1^a(m+u)}{\gamma \min(m,u)}\right)$  and  $\tilde{\beta} = O\left(\frac{1}{\min(m,u)} + \frac{\beta}{\gamma}\right)$ .

If  $\theta$ ,  $\beta_1$ ,  $\beta_2$  and  $\delta_2$  are all  $O\left(\frac{1}{\sqrt{\min(m,u)}}\right)$ ,  $\delta_1^a = O\left(\frac{1}{\sqrt{(m+u)\max(m,u)}}\right)$ ,  $\delta_1^b = O\left(\frac{1}{(\min(m,u))^2}\right)$  and  $\beta = O(1)$ , then the bound (2.26) holds with probability of at least  $(1 - \delta)\left(1 - \frac{1}{\sqrt{\min(m,u)}}\right)$  and converges with the rate  $\frac{1}{\sqrt{\min(m,u)}}$ . Note that this rate of convergence is also achieved in the uniform stability bound (2.25), but with much smaller  $\beta$  values.

El-Yaniv and Pechyony (2006) provided a computational routine for bounding the weak stability of several graph-based transductive algorithms (which are described in Section 2.2.2). Also, in Chapter 2 of this thesis we present an analytical explicit bound on the weak stability of algorithms performing regularization in RKHS. Examples of such algorithms can be found in Section 2.2.2.

We indicated above the conditions under which the bounds (2.25) and (2.26) converge at a rate of  $\frac{1}{\sqrt{\min(m,u)}}$ . An interesting open problem is to develop stability-based bounds such that under the same conditions the convergence rate is  $\max\left(\frac{1}{\sqrt{\min(m,u)}}, \sqrt{\frac{\mathcal{L}_m^\gamma(\mathcal{A})}{\min(m,u)}}\right)$ .

## Realizable case

The risk bounds that we showed in the previous sections converge with the rates  $\frac{1}{\sqrt{\min(m,u)}}$  or  $\max\left(\frac{1}{\min(m,u)}, \sqrt{\frac{\mathcal{L}_m(\mathcal{A})}{\min(m,u)}}\right)$ . In this section we show that, under some assumption on the hypothesis space, faster convergence rates can be achieved.

Suppose that there exists a hypothesis  $\mathbf{h}^* \in \mathcal{H}$  with zero error on the full sample, namely, a zero error is *realizable* by  $\mathcal{H}$ . We assume that  $\mathcal{A}$  is of Type 1 and for any training/test set partition  $\mathcal{A}$  outputs a hypothesis  $\mathbf{h} \in \mathcal{H}_{\mathcal{A}} \subseteq \mathcal{H}$  that is consistent with the training labels (i.e.  $\mathcal{L}_m(\mathcal{A}) = 0$ ). Let  $p(\cdot)$  be a prior probability over  $\mathcal{H}$  (as in PAC-Bayesian bounds). Blum and Langford (2003) proved that for any  $\delta > 0$ , with probability of at least  $1 - \delta$ ,

$$\mathcal{L}_u(\mathcal{A}) \leq \mathcal{L}_m(\mathcal{A}) + \frac{\ln \frac{1}{p(\mathbf{h})} + \ln \frac{1}{\delta}}{u \ln \left(1 + \frac{m}{u}\right)}. \quad (2.27)$$

Suppose that the prior  $p(\cdot)$  is good, namely  $\ln \frac{1}{p(\mathbf{h})} = O(1)$  for any  $\mathbf{h} \in \mathcal{H}_{\mathcal{A}}$ . If  $u \gg m$ , then  $u \ln \left(1 + \frac{m}{u}\right) \approx u \frac{m}{u} = m$  and the bound (2.27) converges at a rate of  $\frac{1}{m}$ . This result is expected, since if  $u \gg m$ , then the transductive setting resembles the inductive one with a training set of size  $m$ , and it is known (e.g., see Vapnik, 1998, Section 4.2) that in the realizable case inductive risk bounds converge at a rate of  $\frac{1}{m}$ . Also the convergence rate of  $\frac{1}{m}$  is attained by (2.27) if  $m = \theta(u)$ . However, if  $m \gg u$ , then the bound (2.27) converges at a rate of

$\frac{1}{u \ln(m)}$ . Yet since  $m \gg u$ , we can assume that  $m$  is exponentially larger than  $u$ . Under such an assumption the bound (2.27) converges with the rate  $\frac{1}{u}$ . Hence, summarizing the three cases of  $m \gg u$ ,  $m \ll u$  and  $m = \theta(u)$ , we determine that the bound (2.27) converges at a rate of  $\frac{1}{\min(m,u)}$ .

### 2.1.3 Consistency

The risk bounds presented in Section 2.1.2 show that under some conditions, the difference  $\mathcal{L}_u(\mathcal{A}) - \mathcal{L}_m(\mathcal{A})$  between the test and training errors of  $\mathcal{A}$  converges to zero. However, in general, this convergence result does not indicate that the test error of  $\mathcal{A}$  converges to the minimal possible one as  $m$  and  $u$  increase. In other words, it does not mean that  $\mathcal{A}$  is *consistent*. The consistency of  $\mathcal{A}$  could be proved if we were able to show that the empirical error of  $\mathcal{A}$  converges to zero as  $m$  and  $u$  increase. Currently, however, no such results are known.

The study of the consistency of transductive learning algorithms is motivated by the following example. Consider the problem of denoising the digital picture of some fixed resolution. In this problem each pixel can be thought of as example and the set of all pixels is a full sample  $X_{m+u}$ . Suppose that for some set of pixels we know that they are ‘clean’ and their colors do not contain noise. These pixels constitute the training set. Our goal is to find the true colors of the rest of the pixels, based on their noisy colors and their geometric location in the picture. These noisy pixels constitute the test set. Thus the denoising task can be stated as a transductive learning problem. Given some transductive algorithm  $\mathcal{A}$  we would like to know if its test set’s accuracy will improve when the resolution of the picture increases. Since the increase in resolution is equal to the increase in the size of the full sample, the improvement of the test set’s accuracy is equal to the algorithm’s consistency.

Formally, we define the consistency of transductive algorithm  $\mathcal{A}$  as follows:

**Definition 3 (Transductive consistency)** *Let  $r \triangleq \frac{m}{u}$  be fixed and  $\mathcal{D} \triangleq \langle x_1, y_1 \rangle, \langle x_2, y_2 \rangle, \dots$  be an infinite sequence of labeled examples. Let  $S_{m+u}$  be the first  $m+u$  examples in  $\mathcal{D}$  and  $\Pi(S_{m+u})$  be the uniform distribution over the partitions  $(S_m, X_u)$  of  $S_{m+u}$  into  $m$  labeled and  $u$  unlabeled examples. Transductive algorithm  $\mathcal{A}$  is consistent if for any such  $\mathcal{D}$ ,*

$$\lim_{m+u \rightarrow \infty} \mathbf{E}_{(S_m, X_u) \sim \Pi(S_{m+u})} \{ \mathcal{L}_u(\mathcal{A}) - \mathcal{L}_u^* \} = 0 \quad , \quad (2.28)$$

where  $\mathcal{L}_u^*$  is the minimal test set error that can be achieved by any hypothesis.<sup>5</sup> Definition (3) is a transductive variant of the (inductive) definition of universal consistency of Devroye et al. (1996, Definition 6.2).

To prove (2.28) it is sufficient to prove that

$$\mathbf{E}_{(S_m, X_u) \sim \Pi(S_{m+u})} \{ \mathcal{L}_u(\mathcal{A}) - \mathcal{L}_u^* \} \leq g(m, u, \mathcal{A}) \quad , \quad (2.29)$$

---

<sup>5</sup>If the test set does not contain duplicate examples with opposite labels, then  $\mathcal{L}_u^* = 0$ .

where  $\lim_{m+u \rightarrow \infty} g(m, u, \mathcal{A}) = 0$ . We refer to the bounds of the form (2.29) as *excess risk bounds*.

The important problem of the consistency of transductive algorithms has yet only been considered by Johnson and Zhang (2007). Their paper considered a transductive algorithm performing regularization in RKHS. The hypothesis  $\hat{\mathbf{h}}$  generated by this algorithm is defined as

$$\hat{\mathbf{h}} = \arg \min_{\mathbf{h} \in \mathbb{R}^{m+u}} \mathcal{L}_m(\mathbf{h}) + \lambda \mathbf{h}^T Q \mathbf{h} , \quad (2.30)$$

where  $Q$  is an  $(m+u) \times (m+u)$  positive definite matrix. We assume that  $Q$  depends on the normalized Laplacian of the underlying graph  $G$  (see a detailed description of graph-based algorithms in Section 2.2.2). The vertices of  $G$  are the full sample examples and the weights of the edges between the vertices are set according to some rule.<sup>6</sup> The entire procedure for constructing  $G$  depends only on  $X_{m+u}$  and is independent of the labels and of the training/test set partition. The true labels of the full sample examples partition  $G$  into a number of disjoint pure components. Pure components are connected subgraphs of  $G$ . Moreover, all vertices in the same pure component have the same label. Let  $q$  be the number of pure components. Note that we do not know this number, since it depends on the unknown test labels. Johnson and Zhang proved that for some specific choice of  $\lambda$ ,

$$\mathbf{E}_{(S_m, X_u) \sim \Pi(S_{m+u})} \{ \mathcal{L}_u(\mathcal{A}) - \mathcal{L}_u^* \} \leq O \left( \sqrt{\frac{q}{m}} \right) . \quad (2.31)$$

Hence the algorithm (2.30) is consistent and the convergence rate to the best possible test error depends on the number of pure components of  $G$ . Nevertheless we do not know the true test labels when constructing  $G$ . Hence we can only utilize our prior knowledge about the true labels in order to build  $G$  with the small number of pure components. Johnson and Zhang also proved that if the number of pure components is very small and they are balanced and have roughly the same size, then the faster convergence rate of  $O(\frac{1}{m})$  can be proved in (2.31).

The disadvantage of the results of Johnson and Zhang is that they achieve the above convergence rates by choosing  $\lambda$  in (2.30) to be dependent on the unknown test labels. Clearly, such a choice for  $\lambda$  cannot happen in practice. The proof of the consistency of (2.30) with  $\lambda$  being independent of the labels is a challenging open problem.

## 2.1.4 Lower bounds

In the previous two sections we presented a number of the upper bounds on the test error of transductive algorithms. We have seen that the typical rates

---

<sup>6</sup>The example of such a rule is to set the weights to be inversely proportional to the distances between examples; see more examples in Section 2.2.2.

of convergence are  $\frac{1}{\sqrt{\min(m,u)}}$  and  $\sqrt{\frac{\mathcal{L}_u(\mathcal{A})}{\min(m,u)}}$ . The natural question is if these convergence rates are tight. This question can be answered by considering lower bounds.

Transductive lower bounds may be of one of the following two types:

**Lower absolute risk bounds** These bounds provide an upper bound on the confidence achieved by the bounds of the form (2.5). The lower absolute risk bounds have the following form: For any full sample  $S_{m+u}$ , any hypothesis space  $\mathcal{H}$  and any  $\delta > 0$ , with probability of *at most*  $1 - \delta$  over the training/test partition, for all  $\mathbf{h} \in \mathcal{H}$ ,

$$\mathcal{L}_u(\mathbf{h}) \leq \mathcal{L}_m(\mathbf{h}) + \text{slack term}(m, u, \delta, \mathcal{H}) , \quad (2.32)$$

where the slack term depends on  $m$ ,  $u$ ,  $\delta$  and  $\mathcal{H}$ . The lower bound (2.32) is equivalent to the statement that for any full sample  $S_{m+u}$ , any hypothesis space  $\mathcal{H}$  and any  $\delta > 0$ ,

$$\mathbf{P}_{(S_m, X_u) \sim \Pi(S_{m+u})} \{ \exists \mathbf{h} \in \mathcal{H} : \mathcal{L}_u(\mathbf{h}) > g(\mathcal{L}_u(\mathbf{h}), m, u, \delta, \mathcal{H}) \} \geq \delta , \quad (2.33)$$

where  $g$  is a function of  $\mathcal{L}_u(\mathbf{h})$ ,  $m$ ,  $u$ ,  $\delta$  and  $\mathcal{H}$ . Vapnik (1998, Section 14.7) proved the existence of the bound (2.33) for the case of  $m = u$ , but did not give an explicit expression for  $g(\mathcal{L}_u(\mathbf{h}), m, u, \delta, \mathcal{H})$ . The explicit derivation of (2.33) for the case of  $m \neq u$  is an open problem.

We also note that the lower bound similar to (2.33) was published by Blum and Langford (2003). However Blum and Langford proved (2.33) only for some specific full sample and hypothesis space.<sup>7</sup> As such, the result of Blum and Langford does not satisfy our requirement for the lower bound to be valid for any full sample and any hypothesis space.

**Lower excess risk bounds** These bounds provide a lower bound on the expectation of the difference between the test error of *any* transductive algorithm  $\mathcal{A}$  and the best possible error  $\mathcal{L}_u^*$ . Formally, the lower bound has the following form:

Let  $r \triangleq \frac{m}{u}$  be fixed. There exists an infinite sequence  $\mathcal{D} \triangleq \langle x_1, y_1 \rangle, \langle x_2, y_2 \rangle, \dots$  of labeled examples, such that for any transductive algorithm  $\mathcal{A}$  with hypothesis space  $\mathcal{H}$ ,

$$\mathbf{E}_{(S_m, X_u) \sim \Pi(S_{m+u})} \{ \mathcal{L}_u(\mathcal{A}) - \mathcal{L}_u^* \} \geq g(m, u, \mathcal{H}) ,$$

---

<sup>7</sup>In particular, as it was acknowledged by the authors of (Blum & Langford, 2003), their bound can be trivially proved by considering the hypothesis space containing a hypothesis with error 1 on the full sample. For such a hypothesis space the bound (2.33) holds with confidence  $\delta = 1$ .

where  $S_{m+u}$  are the first  $m + u$  examples in  $\mathcal{D}$  and  $g$  is a function of  $m$ ,  $u$  and  $\mathcal{H}$ .

Currently, there are no known lower excess risk bounds (as we noted above, development of the upper excess risk bounds is also just beginning). Examples of the lower excess risk bounds for an inductive setting can be found in (Devroye et al., 1996, Section 14).

### 2.1.5 Relation to other learning models

In this section we show how the transductive learning model is related, from the theoretical point of view, to inductive learning models. We consider supervised and semi-supervised inductive learning models, presented informally in Chapter 1. We present two results. The first one shows that an upper transductive risk bound implies an upper inductive risk bound. The same holds also for lower bounds. The second result shows that the combination of upper risk bounds for supervised learning and transductive learning implies an upper risk bound for semi-supervised learning.

#### Transductive risk bounds imply the bounds for supervised learning

In this section we consider Setting 2 of transductive learning, defined in Section 2.1.1. We also assume that in the transductive setting the sizes of the training and test sets are equal, namely  $u = m$ . Thus the training examples have indices  $1, \dots, 2m$  and the test examples have indices  $m + 1, \dots, 2m$ . We abbreviate  $(X, Y)_i \triangleq \langle x_1, y_1 \rangle, \dots, \langle x_i, y_i \rangle$ . We now formally define the supervised learning model. Let  $\mathcal{F}$  be a space of functions  $f : \mathcal{X} \rightarrow \{\pm 1\}$ . As in the transductive definition, the training error of  $f$  is defined as  $\mathcal{L}_m(f) = \frac{1}{m} \sum_{i=1}^m \ell(f(x_i), y_i)$ . We define the generalization error of  $f$  as  $\mathcal{L}(f) = \mathbf{E}_{(x,y) \sim \mathcal{D}} \{\ell(f(x), y)\}$ . The goal of the supervised learning algorithm is, based on the training examples  $(X, Y)_m$ , to find a function  $f \in \mathcal{F}$  with a low generalization error. The two-sided symmetrization lemma of Vapnik and Chervonenkis (1991) states that

$$\begin{aligned} \mathbf{P}_{(X,Y)_{2m}} \left\{ \sup_{f \in \mathcal{F}} |\mathcal{L}_m(f) - \mathcal{L}_u(f)| > 2\epsilon \right\} &\leq \mathbf{P}_{(X,Y)_m} \left\{ \sup_{f \in \mathcal{F}} |\mathcal{L}(f) - \mathcal{L}_m(f)| > \epsilon \right\} \\ &\leq 2\mathbf{P}_{(X,Y)_{2m}} \left\{ \sup_{f \in \mathcal{F}} |\mathcal{L}_m(f) - \mathcal{L}_u(f)| > \frac{\epsilon}{2} \right\}. \end{aligned}$$

These inequalities say that the uniform (over all functions in  $\mathcal{F}$ ) convergence of the training error to the test error in transductive Setting 2 with  $m = u$  is a necessary and sufficient condition for the uniform convergence of the training error to the generalization error in the supervised setting. The function space  $\mathcal{F}$  can be a functional form of the transductive hypothesis space  $\mathcal{H}$ . In this case the two-sided symmetrization lemma states that upper (lower) transductive risk bounds in Setting 2 for  $m = u$  imply upper (lower) inductive risk bounds.

## Transductive risk bounds + the bounds for supervised learning = the bounds for semi-supervised learning

We consider the following semi-supervised learning setting. A training set  $\{\{x_i, y_i\}_{i=1}^m \cup \{x_i\}_{i=m+1}^{m+u}\}$ , consisting of  $m$  labeled and  $u$  unlabeled examples, is given. Training examples are sampled i.i.d. from an (unknown) distribution  $\mathcal{D}$  over  $\mathcal{X} \times \mathcal{Y}$ .<sup>8</sup> A semi-supervised learning algorithm utilizes both labeled and unlabeled training examples and produces a hypothesis  $f \in \mathcal{F}$ . The goal is to generate a hypothesis  $f$  with a small generalization error  $\mathcal{L}(f) \triangleq \mathbf{E}_{(x,y) \sim \mathcal{D}}\{\ell(f(x), y)\}$ . The *full sample training error* of  $f$  is defined to be  $\mathcal{L}_{m+u}(f) \triangleq \frac{1}{m+u} \sum_{i=1}^{m+u} \ell(f(x_i), y_i)$ . Note that since the labels  $\{y_i\}_{i=m+1}^{m+u}$  are unknown, the training error cannot be computed, but only bounded. The *labeled training error* of  $f$  is defined to be  $\mathcal{L}_m(f) \triangleq \frac{1}{m} \sum_{i=1}^m \ell(f(x_i), y_i)$  and the *unlabeled training error* is defined to be  $\mathcal{L}_u(f) \triangleq \frac{1}{u} \sum_{i=m+1}^{m+u} \ell(f(x_i), y_i)$ .

Upper risk bounds for the semi-supervised setting can be derived using the following high-level scheme:

1. Take any known probabilistic generalization bounds for supervised learning (e.g., see Boucheron et al., 2005) of the form  $\mathcal{L}(f) \leq \mathcal{L}_{m+u}(f) +$  the slack term.
2. Decompose  $\mathcal{L}_{m+u}(f)$  into  $\frac{m\mathcal{L}_m(f) + u\mathcal{L}_u(f)}{m+u}$  and bound  $\mathcal{L}_u(f)$  using the known transductive risk bounds for Setting 2.

By Lemma 1 any transductive risk bound in Setting 1 is also valid in Setting 2. Thus we can use any risk bound from Section 2.1.2 to bound  $\mathcal{L}_u(f)$ . Note that in this case the transductive risk bounds are operated with the transductive hypothesis space  $\mathcal{H}$  whose functional representation is  $\mathcal{F}$ .

### 2.1.6 When transduction is better than induction and on the value of unlabeled examples

As noted in Chapter 1, transductive learning is aimed at improving the performance of inductive learning by utilizing the known unlabeled test examples. In some learning problems the training sets are sufficiently good for inductive learning algorithms to exhibit excellent performance. Clearly, in such problems we cannot hope to develop a transductive algorithm that will outperform an inductive one. Moreover, as we demonstrate empirically in Chapter 5, there are learning problems where the knowledge of the test set probably deteriorates the algorithm's performance. It would be interesting, from both a theoretical and practical point of view, to characterize the problems where the transductive

---

<sup>8</sup>The unlabeled examples  $x_{m+1}, \dots, x_{m+u}$  are sampled from the marginal distribution over  $\mathcal{X}$

setting is provably better than the inductive one. In particular, it would be interesting to characterize the full samples where the following two conditions will hold:

- Most of the training/test set partitions are easy, in the sense that there exists a transductive algorithm provably achieving a small test error on them.
- The lower bounds on the error (over the test examples) of any inductive (both supervised and semi-supervised) learning algorithm is larger than the upper bound on the test error of the above transductive algorithm.

Currently this research direction is completely open. We note that there is a large body of research (e.g., see Balcan & Blum, 2006; Ben-David et al., 2008; Niyogi, 2008; Sinha & Belkin, 2008 and the references therein) about the usefulness of unlabeled examples in the semi-supervised setting. In this setting the unlabeled examples are used mainly to estimate the density function of the marginal distribution over  $\mathcal{X}$ . By assuming the spatial relatedness between examples and labels (i.e., that  $p(y|x) \neq p(y)$ ), such an estimate can potentially help in generating function  $f \in \mathcal{F}$  with a good generalization error. As noted in Chapter 1, in the transductive setting the unlabeled examples can be helpful when they induce easy training and test sets. Thus, while the above results about the helpfulness of the unlabeled examples in semi-supervised setting can be useful, they cannot be directly applied to the transductive one.

### 2.1.7 Related learning models

In this section we survey a number of learning models that do not fit into the framework of transductive Settings 1 and 2, but, nevertheless, have a transductive nature.

#### Active transductive learning

In the transductive setting described in Section 2.1.1 the training set is drawn uniformly at random among all possible training sets. As a result, the training set may not be representative of the test examples. In this *passive* model the learner has no control over the generation of the training set. In contrast, in the *active* model the learner has full control over the contents of the training set. We now give a formal definition of the active model within transductive Setting 1.

At the beginning of the learning process the active learner is given an unlabeled full sample  $X_{m+u}$  and the budget  $m$  of the number of labeled examples. Then the learner performs  $m$  iterations to build its training set. At the  $i$ th iteration the learner *queries* the label of some unlabeled example from the pool of unlabeled ones and receives its label. Initially the pool of unlabeled examples

is  $X_{m+u}$  and it is reduced by one example after each iteration. After the  $m$ th iteration the learner has  $m$  labeled and  $u$  unlabeled examples. As in the passive model, the labeled examples constitute the training set and the unlabeled examples constitute the test set. After constructing this training/test set partition the goal of the active learning is the same as that of the passive one, namely the active learner needs to find an accurate labeling of the test examples.

Note that in the active model within transductive Setting 1, when the active learner constructs its training set, it also simultaneously constructs its test set. Hence the goal of such an active learner is to construct a test set that is easy, given the training set. Since the active learner has more ‘power’ than the passive one, we would like the active learner to achieve better accuracy on the test set than the passive learner.<sup>9</sup>

There is little theoretical research in the above active transductive model. The only work whose setting resembles the one described above is done by Dasgupta (2005). In the setting of Dasgupta the learner assumes that some hypothesis  $\mathbf{h}^*$  in the hypothesis set<sup>10</sup>  $\mathcal{H}$  has a zero full sample error. Under such an assumption, Dasgupta analyzed the size of the training set (i.e., the bound on the number of labeled examples) that is needed in order to identify  $\mathbf{h}^*$ . Note that once  $\mathbf{h}^*$  is identified, we find a perfect labeling (with zero error) of the test points.<sup>11</sup> Dasgupta gave example of full sample  $X_{m+u}$  and hypothesis set  $\mathcal{H}$  such that the number of labeled examples needed by any query strategy to identify  $\mathbf{h}^*$  from  $\mathcal{H}$  is  $\Omega(m+u)$ . This example provides evidence that, in general, the active learner cannot always be better than the passive one. On the positive side, Dasgupta showed a querying strategy (for the identification of  $\mathbf{h}^*$ ) that for any full sample  $X_{m+u}$  and any hypothesis set  $\mathcal{H}$  performs a number of queries at most four times the minimal number of queries needed to identify  $\mathbf{h}^*$ .

In general, the above assumption that there exists an  $\mathbf{h}^* \in \mathcal{H}$  with a zero full sample error is very strong. The interesting research direction is to generalize the results of (Dasgupta, 2005) to the case when the full sample error of  $\mathbf{h}^*$  is not zero, but some  $\epsilon > 0$ .

## Transductive online learning

Transductive online learning is a variant of the standard online learning model in which the learner knows in advance the unlabeled sequence of examples that it will receive. The formal definition of the transductive online model is as follows.

---

<sup>9</sup>Note that the test sets of active and passive learners will probably be different.

<sup>10</sup>Here we intentionally refer to  $\mathcal{H}$  as hypothesis set in order not to confuse it with the hypothesis space of passive algorithms, defined in Section 2.1.2.

<sup>11</sup>In the context of transductive Setting 1 (with the known full sample  $X_{m+u}$ ) the model of (Dasgupta, 2005) is the same as the models of exact learning with membership queries (Angluin, 1988) and of self-directed learning (Yin, 1995). However, traditionally the last two models are attributed to the learning of boolean functions and simple geometric concepts. In contrast, the results of (Dasgupta, 2005) apply to more general domains.

Initially the learner receives the unlabeled full sample  $X_n = \{x_1, \dots, x_n\}$ . Each example in  $X_n$  has a unique label. The learning is done iteratively. At the  $i$ th iteration ( $1 \leq i \leq n$ ) the learner knows the labels of  $x_1, \dots, x_{i-1}$  and predicts the label of  $x_i$ . This prediction can depend on both the past labeled examples  $\langle x_1, y_1 \rangle, \dots, \langle x_{i-1}, y_{i-1} \rangle$  and the future unlabeled ones  $x_{i+1}, \dots, x_n$ . Then the true label of  $x_i$  is revealed. The goal of the learner is to minimize the number of mistakes made during  $n$  iterations.

Before surveying the existing results for transductive online learning we give an example that uses this learning model. Suppose we are managers of a TV channel transmitting Olympic games<sup>12</sup> and we want to predict the size of the TV audience for each competition. Such a prediction can be useful, for example, for selling advertisements. Each competition is an unlabeled example and all Olympic competitions constitute a full sample. The features of each competition are its transmission time, type (athletics/swimming/etc.), stage (final/semi-final), the types of competitions that are transmitted at the same time, the content of other TV channels during the competition, etc. At the beginning of each competition we predict the size of the audience. This prediction can be based both on the past competition (the number of people who watched the previous competition) and on the future ones. At the end of each competition we compute the loss, which is a function of the predicted and the actual number of people watching.

Transductive online learning was introduced by Ben-David et al. (1997). In their paper the authors quantified the gap between the number of mistakes  $M_{\text{trans}}$  made by any transductive online algorithm and the number of mistakes  $M_{\text{ind}}$  made using any standard online learning algorithm. In particular, it is shown in (Ben-David et al., 1997) that there exists a full sample such that  $M_{\text{trans}} = \Omega(\sqrt{\log M_{\text{ind}}})$ . This result provided an initial characterization of the gap between transductive and the standard online learning models. It would be interesting to find out if this result is tight, and to show, for some fixed transductive online algorithm  $\mathcal{A}$ , an upper bound on  $M_{\text{trans}}(\mathcal{A})$  in terms of  $M_{\text{ind}}$ .

A number of transductive online algorithms with a bounded number of mistakes has appeared in (Kakade & Kalai, 2006; Herbster et al., 2005; Herbster & Pontil, 2007; Herbster, 2008). Kakade and Kalai (2006) showed a simple transductive online algorithm, denoted by  $\mathcal{A}_2$ , that uses as a component a supervised learning algorithm. At the  $i$ th iteration  $\mathcal{A}_2$  randomly labels the unlabeled examples  $x_i, \dots, x_{m+u}$ , replicates each one of these examples and runs a supervised learning algorithm on the known labeled examples  $\langle x_1, y_1 \rangle, \dots, \langle x_{i-1}, y_{i-1} \rangle$  and the replicated ones. The function  $f$  generated by the supervised learning algorithm is used to label  $x_i$ . Let  $\mathcal{F}$  be the hypothesis space of the supervised learning algorithm. Kalai and Kakade showed that under some conditions on the

---

<sup>12</sup>This survey was written during Beijing 2008 Olympic games.

supervised learning algorithm,

$$\mathbf{E}\{\#\text{of mistakes}(\mathcal{A}_2)\} \leq \min_{f \in \mathcal{F}} \#\text{of mistakes}(f) + O(n^{3/4} \sqrt{d \log n}) , \quad (2.34)$$

where  $d$  is a VC-dimension of  $\mathcal{F}$  and the expectation is taken over the random draws made by  $\mathcal{A}_2$ . It would be interesting to know if the slack term  $O(n^{3/4} \sqrt{d \log n})$  in (2.34) is tight.

In a series of papers Herbster et al. (2005), Herbster and Pontil (2007), Herbster (2008) showed a number of transductive online algorithms that operate in the kernel space induced by the Laplacian of the graph that represents the full sample  $X_{m+u}$  (see a detailed description of such graphs in Section 2.2.2). All these algorithms have the same structure as the well-known perceptron algorithm for online learning. Moreover, these algorithms benefit from the bounds on the number of mistakes. These bounds depend on various properties of the underlying graph, e.g graph diameter, resistance diameter, cut size induced by the true labels, cluster structure etc.

## Transductive Ranking

In transductive ranking the goal of the learner is to find an accurate ranking of the full sample examples. Such ranking can be achieved by assigning real-valued labels to full sample examples and sorting them according to the label's values. As in the classification setting, in ranking the learner is given a full sample containing some labeled examples. Note that in transductive ranking the definition of the test set is different from the one used in the classification setting. While in the classification setting the test set is the set of all unlabeled examples, in ranking the test set is the entire full sample, including all available labeled examples. In the ranking setting the labels can have two forms. In the first one (Agarwal, 2008) the label of each example is a real number that is proportional to the example's rank. In the second form (Agarwal & Chakrabarti, 2007) the labels are orderings of pairs of examples. Each ordering indicates whether the first examples within the pair should be ranked higher or lower than the second one.

Let  $f_i$  be the real-valued label given to  $x_i$  by the learning algorithm and  $y_i$  be the discrete true rank of  $x_i$ . One of the loss functions used by ranking algorithms is the absolute loss (Agarwal, 2008):

$$\ell(h_i, h_j, y_i, y_j) = |y_i - y_j| \left( \mathbb{I}_{(y_i - y_j)(f_i - f_j) < 0} + \frac{1}{2} \mathbb{I}_{f_i = f_j} \right) , \quad (2.35)$$

where  $\mathbb{I}_E$  is an indicator of the event  $E$ . The absolute loss has a non-zero value if the ranking of a pair of examples  $x_i$  and  $x_j$  according to  $f_i$  and  $f_j$  is different from their true ranking (according to  $y_i$  and  $y_j$ ). A similar ranking loss function, called the ranking hinge loss, is considered by Agarwal and Chakrabarti (2007). The disadvantage of the absolute loss (and also the hinge loss) function is that it

has the same values when the error occurs at the top and at the bottom of the ranked list of examples. There exist variants of ranking loss functions that do not suffer from this drawback (e.g., see Agarwal & Chakrabarti, 2007).

Agarwal and Chakrabarti (2007) and Agarwal (2008) developed explicit risk bounds for a number of graph-based transductive ranking algorithms. These bounds are based on the notion of the stability of ranking algorithms and are similar to the stability bounds presented in Section 2.1.2. The only significant difference is that while the bounds in Section 2.1.2 bound the error over unlabeled examples, the ranking bounds bound the error over the entire full sample.

## 2.1.8 Summary

During the last five years transductive learning theory has advanced significantly. In this survey we presented most of theoretical results in transductive learning that have been published at the major conferences and in the top journals. As it can be seen from the relative sizes of the sections, most of the recent research was concentrated in the development of upper risk bounds for various transductive learning models. While a large amount of research has already been done, even larger areas of theoretical research in transductive learning remain relatively unexplored. In the following list we summarize the main directions for the future development of transductive learning theory:

1. Derivation of new transductive algorithms from the existing upper risk bounds. Some results in this direction can be found in Sections 2.2.4 and 2.2.7. In particular, it would be interesting to derive new algorithms from the risk bounds based on stability and on Rademacher complexity.
2. Development of the excess risk bounds and proof of the (in)consistency of the existing transductive algorithms.
3. Development of lower risk bounds for the convergence rate of absolute and excess risk bounds.
4. Characterization of the problems where transductive algorithms are provably better than the inductive ones; also, characterization of the problems where transduction has no advantage over induction.
5. Development of other transductive learning models (e.g., transductive active, transductive online and transductive ranking models).

## 2.2 Transductive Algorithms

Before actually describing transductive algorithms, we present their definition and the scope of this section. A transductive algorithm is a learning algorithm

that operates within a transductive model (either within Setting 1 or 2). Namely, a transductive algorithm receives a labeled training set and an unlabeled test set. The output of a transductive algorithm is a labeling of the test points.

The above definition is very broad. In fact, this definition applies to all supervised and semi-supervised algorithms. Indeed, we can run supervised algorithms on the labeled training set and obtain a hypothesis that will label the unlabeled examples. Also, we may run a semi-supervised algorithm on the labeled training set and the unlabeled test set, obtain a hypothesis and with it label the unlabeled test examples. Both supervised and semi-supervised algorithms generate a general hypothesis that can label examples that do not appear in the input. We say that a learning algorithm is *purely transductive* if it generates a hypothesis that can only label the examples from the test set. In this section we survey learning algorithms that explicitly utilize unlabeled examples in the learning process. Hence we cover both semi-supervised and purely transductive learning algorithms. For the remainder of this section, when we mention transductive algorithms we refer to these two classes of algorithms.

During the last decade more than 100 transductive algorithms have been developed. It is almost infeasible to describe each one. In this section we present the main families of transductive algorithms and describe the ideas behind them. There exist several other surveys of transductive algorithms, e.g. (Zhu, 2008; Seeger, 2000; Chapelle et al., 2006; Haffari, 2006). Our survey partially overlaps with them. On the other hand, we present several families of transductive algorithms that are not included in the existing surveys.

### 2.2.1 Large-margin methods

Large-margin is probably the first approach to incorporate unlabeled examples into the learning process. This approach extended the highly successful supervised large-margin methods, in particular the Support Vector Machine (e.g., see Vapnik, 1998), to input consisting of both labeled and unlabeled examples. In large margin methods the hypothesis is a hyperplane. For simplicity we will assume that this hyperplane lies in the same space as input examples. The *margin* of the hyperplane is a minimum distance from it to any input example.

The most popular transductive large-margin method is the Transductive Support Vector Machine (TSVM) (Vapnik, 1998). TSVM tries to find the hyperplane that has a low error on the labeled examples and a large margin. The underlying assumption of TSVM is that the positive examples from the labeled full sample can be separated from the negative examples by the hyperplane with a large margin. When this assumption does not hold, the performance of TSVM can be bad (see an example in Zhang & Oles, 2000).

TSVM has a solid theoretical justification. As Vapnik (1998, Theorem 10.3) showed, a VC dimension of a set of hyperplanes with a margin of at least  $\gamma$  is

bounded by  $O\left(\frac{1}{\gamma}\right)$ . Hence if we find a hyperplane with a large margin, then it belongs to a hypothesis space with a small VC dimension. Thus by minimizing the training error and maximizing the margin, TSVM effectively finds a hyperplane whose corresponding transductive hypothesis  $\mathbf{h} = (h_1, \dots, h_{m+u})$  has a low value of the risk bound (2.14).

This hyperplane that is generated by TSVM is characterized by the normal vector  $\mathbf{w}$  and the offset  $b$ , and is a solution of the following optimization problem:

$$\min_{\mathbf{w}, b, y_{m+1}, \dots, y_{m+u}} \|\mathbf{w}\|_2^2 + c_1 \sum_{i=1}^m (1 - y_i (\langle \mathbf{w}, x_i \rangle + b))_+ + c_2 \sum_{i=m+1}^{m+u} (1 - y_i (\langle \mathbf{w}, x_i \rangle + b))_+ , \quad (2.36)$$

where  $c_1, c_2 > 0$  are regularization constants, and for any  $t \in \mathbb{R}$ ,  $(t)_+ = t$  if  $t \geq 0$  and  $(t)_+ = 0$  otherwise. The full sample examples are then classified using the hyperplane  $(\mathbf{w}, b)$ , namely for any  $x \in X_{m+u}$  its soft-classification is  $\langle \mathbf{w}, x \rangle + b$ . The expression  $(1 - y_i (\langle \mathbf{w}, x_i \rangle + b))_+$  is a *hinge loss* of example  $x_i$ . If the hinge loss of all full sample examples is zero, then  $\|\mathbf{w}\|_2^2$  is inversely proportional to the margin of the hyperplane characterized by  $(\mathbf{w}, b)$ .

The optimization problem (2.36) operates over the input space of the full sample examples  $X_{m+u}$ . It is also possible to obtain a kernelized version of (2.36) that operates over a high-dimensional space to which the examples are transformed. In fact, the kernelized version of (2.36) is one of the most popular transductive algorithms.

When operating (2.36) in high-dimensional space, it is possible (Joachims, 1999) that there exists a hyperplane that gives the same hard classification for all test examples, and has a large margin and low training error. Such an undesirable effect can be prevented by adding a balancing constraint to (2.36), assuring that the fraction of positive labels (found after solving (2.36)) in the test set is equal to the fraction of positive labels in the training set.

The optimization problem (2.36) is defined over both continuous ( $\mathbf{w}$  and  $b$ ) and discrete variables ( $y_{m+1}, \dots, y_{m+u}$ ). Hence it cannot be solved using the standard methods in continuous optimization and discrete optimization methods should be employed. Note that when the labels  $y_{m+1}, \dots, y_{m+u}$  are fixed, the optimization problem (2.36) is convex and we can easily find its global minimum. Accordingly, we can pass through all possible fixed labelings  $y_{m+1}, \dots, y_{m+u}$ , for each one solve (2.36) and then output the labeling, achieving the minimum of (2.36). However, since the number of possible labelings is  $2^u$ , such an approach is computationally infeasible. Bennett and Demiriz (1999) and Chapelle et al. (2007) demonstrated global optimization techniques provably obtaining a global minimum of (2.36). Nevertheless the worst case complexity of these techniques is still exponential. For two small datasets of up to couple of hundred points in size, Chapelle et al. showed that their optimization procedure is dramatic improvement over many other known methods for solving (2.36). However, due

to the exponential complexity, the techniques of (Chapelle et al., 2007) and (Bennett & Demiriz, 1999) are infeasible for larger datasets.

The optimization problem (2.36) is equivalent to

$$\min_{\mathbf{w}, b} \|\mathbf{w}\|_2^2 + c_1 \sum_{i=1}^m (1 - y_i (\langle \mathbf{w}, x_i \rangle + b))_+ + c_2 \sum_{i=m+1}^{m+u} (1 - |\langle \mathbf{w}, x_i \rangle + b|)_+ . \quad (2.37)$$

As in (2.36), there also exists a kernelized version of (2.37). While the optimization problem (2.37) is defined over continuous variables, it is not convex. A number of optimization methods (e.g., see Chapelle et al., 2008; Wang, 2007; Xu et al., 2008; Zhao et al., 2008 and the references therein) for finding a good local minimum of (2.37) exist. However there is no convincing evidence that one of them is significantly better than the others and as Chapelle et al. (2007) showed, most of them are significantly inferior to the global optimization methods. Currently, the fastest methods (Collobert et al., 2006; Sindhvani & Keerthi, 2006; Zhao et al., 2008) for solving (2.37) are able to process up to tens of thousands of examples.

An interesting variation of large-margin transductive methods is presented by Wang and Shen (2007). Their algorithm finds two hyperplanes, the first one separating training examples and the second one separating the test examples. The second hyperplane must have a large margin w.r.t. test examples and these two hyperplanes must not differ too much. The exact formulation of the algorithm of (Wang & Shen, 2007) is

$$\min_{\mathbf{w}, b, \mathbf{w}', b', y_{m+1}, \dots, y_{m+u}} \|\mathbf{w}'\|_2^2 + c_0 \|\mathbf{w}' - \mathbf{w}\|_2^2 + c_1 \sum_{i=1}^m (1 - y_i (\langle \mathbf{w}, x_i \rangle + b))_+ + c_2 \sum_{i=m+1}^{m+u} (1 - y_i (\langle \mathbf{w}', x_i \rangle + b))_+ , \quad (2.38)$$

where  $c_0, c_1, c_2 > 0$  are regularization constants. The full sample examples are then classified using the hyperplane  $(\mathbf{w}', b)$ ; namely, for any  $x \in X_{m+u}$  its soft-classification is  $\langle \mathbf{w}', x \rangle + b$ . Wang and Shen also presented a kernelized version of (2.38). The approach of (Wang & Shen, 2007) with two hyperplanes can be useful when the training set does not represent the test set well. In this case by setting  $c_0$  to be small, the optimization problem (2.38) will find a hyperplane with a large margin only w.r.t. unlabeled examples.

## 2.2.2 Graph-based methods

Transductive graph-based methods have their roots in the probabilistic model of Markov random fields (Kindermann & Snell, 1980). The idea of a Markov random field model is that the label of each example depends only on the labels of the examples that lie in close proximity. The model of Markov random fields is derived from the Ising model in statistical physics.

Graph-based methods are probably the most popular approach for transductive learning. These methods consist of two steps. In the first step the graph  $G = (V, E)$ , representing the full sample, is constructed. Each vertex  $v \in V$  has a one-to-one correspondence with an example  $x \in X_{m+u}$  and  $|V| = m + u$ .<sup>13</sup> The edges of  $G$  are set according to some rule. In this section we survey a number of methods for constructing  $G$ . In the second step, based on  $G$ , graph-based methods infer the labels of the test examples in  $X_{m+u}$ .

## Constructing a graph

There are applications where a representation graph can be defined easily using domain knowledge. Consider, for example, the task of Web page classification. We can place an edge between two vertices if there is a link from one Web page to another. However, in general, there is no obvious rule for graph construction and some prior domain knowledge should be used.

In general, graph construction is guided by the following rule. There is an edge between two vertices if one of them is similar to the other. The weight of this edge is the similarity value between corresponding examples. Similarity functions that are commonly used in transductive algorithms are cosine similarity, linear kernel and RBF kernel. Other kernel functions (see Schölkopf & Smola, 2002 for more examples) may also be used. Thus once we have chosen the similarity function, to construct the graph we need a rule for connecting the vertices by edges.

The examples of possible ways of setting edges are listed below:

- A full weighted graph;
- An MST graph: obtained by building the minimum spanning tree of a full weighted graph. Carreira-Perpiñán and Zemel (2005) presented several variations of an MST graph;
- A  $k$ -NN graph: two examples are connected by the edge if one of them is among the  $k$  most similar neighbors of the other;
- An  $\epsilon$ -NN graph: two examples are connected by a (possibly weighted) edge if the similarity between them is at least  $\epsilon$ .

Wang et al. (2007) extended the above graph construction methods to transductive multi-task setting. Initially, for each task a graph is constructed using the above methods. Then these graphs are connected based on the similarity of examples from different tasks.

There is also a number of more complicated ways of constructing a graph. Wu and Schölkopf (2007) and Wang and Zhang (2008) constructed  $k$ -NN graph

---

<sup>13</sup>Sometimes auxiliary nodes are also added to  $G$  and then  $|V| > m + u$ .

and then set the weights  $w_{ij}$  of the edges between  $x_i$  and  $x_j$  in such a way that for any  $1 \leq i \leq m + u$ ,  $x_i \approx \sum_{x_j \in \mathcal{N}(x_i)} w_{ij} x_j$ , where  $\mathcal{N}(x_i)$  is a set of neighbors of  $x_i$  in the graph. Hein and Maier (2007) found an approximate low-dimensional manifold at which the full sample points are situated and utilized this manifold in order to define similarity between the points.

The above methods of constructing a graph are independent of the learning algorithms. There are also several methods for constructing a graph and learning over the graph simultaneously. These methods are described in the next section.

## Learning over graph

The first graph-based transductive algorithm was published by Blum and Chawla (2001). This algorithm adds to  $G$  two vertices,  $v_+$  and  $v_-$ . The vertex  $v_+$  is connected to all positive training vertices with the edges of infinitely high weight. Similarly,  $v_-$  is connected to all negative training vertices with the edges of infinitely high weight. The size of the graph cut is the sum of the weights of the edges that cross the cut. The algorithm of Blum and Chawla finds a minimal size cut (i.e., *minimal cut*) of  $G$  that separates  $v_+$  from  $v_-$ . The vertices that are on the side of  $v_+$  are classified as positive and the vertices on the side of  $v_-$  are classified as negative. Note that since the weights of the edges adjacent to  $v_+$  and  $v_-$  are infinitely high, the algorithm of Blum and Chawla effectively separates all positive training vertices from the negative ones. Hence the empirical error of the hypothesis generated by this algorithm is zero.

The drawback of the minimal cut approach is that it frequently generates a highly unbalanced cut, with one of the parts being much larger than the other. For example, it is possible that the first part contains only labeled positive examples and the second part contains both labeled negative examples and all the unlabeled test examples. Blum et al. (2004) used randomization to partially overcome this issue. Another way to overcome the issue of unbalanced partitions is to find a cut of a minimal normalized size (i.e., *minimal normalized cut*). Let  $|C|$  be the size of the cut  $C$ . The cut  $C$  divides  $V$  into two disjoint sets,  $V_1$  and  $V_2$ . The normalized size of  $C$  is  $\frac{|C|}{|V_1| \cdot |V_2|}$ . Thus in general the minimal normalized cut has a small cut size and partitions  $G$  into two roughly equal parts. The problem of finding a minimal normalized cut is NP-hard. There exists several relaxations of the minimal normalized cut problem. Two such relaxations have been developed into transductive learning algorithms. The spectral relaxation of the minimal normalized cut problem was implemented by Joachims (2003). The resulting transductive algorithm, named the Spectral Graph Transducer (SGT), is considered one of the most powerful transductive algorithms. Another relaxation of the minimal normalized cut problem, called Semidefinite Programming (SDP) relaxation, was implemented by Bie and Cristianini (2006).

The cut-based approach described above considers *hard cuts*. In the hard cut each vertex is labeled  $+1$  or  $-1$  and the edges with differently labeled adjacent

vertices constitute a cut. If the vertices have soft classification, e.g., in the range of  $[-1, 1]$ , then the resulting cut is a *soft cut*. The *size of the soft cut* induced by the soft labeling  $\mathbf{h}$  is

$$\frac{1}{4} \sum_{w_{ij} \in E} w_{i,j} (h_i - h_j)^2 = \mathbf{h}^T L \mathbf{h} \quad , \quad (2.39)$$

where  $w_{ij}$  is the weight of the edge connecting the  $i$ th and  $j$ th examples and  $L$  is a normalized Laplacian of  $G$ . If there is no such edge then  $w_{ij} = 0$ . Thus in the soft cut the edges that cross the cut are the ones with adjacent vertices having different soft classifications. It can be verified that if  $\mathbf{h}$  is a hard classification vector in  $\{\pm 1\}^{m+u}$ , then (2.39) is the size of the hard cut induced by  $\mathbf{h}$ . Hence the notion of soft cut is a generalization of the notion of hard cut.

Another measure of the size of the soft cut induced by the soft labeling  $\mathbf{h}$  is

$$\frac{1}{4} \sum_{w_{ij} \in E} w_{i,j} \left( \frac{h_i}{d_i} - \frac{h_j}{d_j} \right)^2 = \mathbf{h}^T L_N \mathbf{h} \quad , \quad (2.40)$$

where  $d_i = \sum_{k=1}^{m+u} w_{ik}$  is the sum of the weights of the edges that are adjacent to the  $i$ th vertex and  $L_N$  is a normalized Laplacian of  $G$ . The difference between (2.40) and (2.39) is that in (2.40) the soft classification  $h_i$  of the  $i$ th example is normalized by the weight  $d_i$  of the  $i$ th vertex. We refer to (2.40) as a *normalized size* of the soft cut. The idea of the normalization in (2.40) is as follows. We assume that for any example  $x_i \in X_{m+u}$ , most of its neighbors in  $G$  have the same hard classification as  $x_i$ . The soft classification  $h_i$  of  $x_i$  depends on the soft classifications of the neighbors of  $x_i$  in  $G$ . It is assumed (as in the Markov random field model) that each such neighbor  $x_j$  contributes to the value of  $h_i$ . The amount of contribution is proportional to  $w_{ij}$ . Thus, if  $d_i$  is large, then  $x_i$  will obtain a large number of such contributions from its neighbors in  $G$ . As a result, if the neighbors of  $x_i$  have similar soft classifications (and this is definitely possible under the above assumption), then the value of  $|h_i|$  will be large. Hence by normalizing  $h_i$  to  $\frac{h_i}{d_i}$  we bring all soft classification values to the same scale. Another method of normalization, motivated by risk bounds, can be found in (Johnson & Zhang, 2007).

There exist a number of transductive algorithms that find small (but not necessary minimal) smooth cuts in order to label test examples. In this survey we focus on one such algorithm, due to Wang and Zhang (2008), which in turn is a slight modification of the algorithm of Zhou et al. (2004). Examples of other algorithms that find smooth cuts of small normalized/unnormalized size can be found in (Zhu et al., 2003; Belkin et al., 2004; Johnson & Zhang, 2007; Pelckmans et al., 2007; Culp & Michailidis, 2008; Wang et al., 2008). The prominent feature of the algorithm of Zhou et al. is that it can be interpreted in three different ways: as the algorithm finding a small soft cut, as a label propagation algorithm and

as a random walk algorithm. We start with the presentation of the algorithm of Wang and Zhang (2008) as the one finding a small soft cut.

The algorithm of Wang and Zhang constructs an undirected graph<sup>14</sup> such that for any  $1 \leq i \leq m + u$ ,  $\sum_{i=j}^{m+u} w_{ij} = 1$ . Then this algorithm finds a soft classification vector  $\hat{\mathbf{h}}$  minimizing

$$\sum_{i=1}^{m+u} (h_i - y_i)^2 + c \sum_{w_{ij} \in E} w_{i,j} (h_i - h_j)^2, \quad (2.41)$$

where  $c > 0$  is a regularization constant and the vector  $\mathbf{y} \in \{\pm 1, 0\}^{m+u}$  has the following structure. If  $x_i$  is in the training set, then  $y_i$  is the true label of  $x_i$ . Otherwise, if  $x_i$  is in the test set, then  $y_i = 0$ .

The first sum in (2.41) is an *empirical error* of  $\mathbf{h}$ . Note this error depends on both the training error of  $\mathbf{h}$  on the labeled examples and on the error of  $\mathbf{h}$  on the unlabeled ones. While the dependence on the training error is common to many learning algorithms, the dependence on the error over the unlabeled examples is a unique feature of (2.41). Since for unlabeled examples  $y_i = 0$ , the error of unlabeled example  $x_i$  is  $h_i^2$ . By minimizing this error the algorithm of Wang and Zhang tries to find a soft classification of  $x_i$  that is close to zero. Suppose that  $\|h_i\|$  expresses our confidence in labeling  $x_i$  with  $\text{sign}(h_i)$ . The rationale behind the above error over unlabeled examples is that since  $x_i$  is unlabeled, we should not be too confident about its label. The second sum in (2.41) is the normalized size of the soft cut induced by  $\mathbf{h}$ . Thus the algorithm of Wang and Zhang finds a soft classification vector  $\mathbf{h}$  having low empirical error and a small normalized size of the induced cut. The balance between the low empirical error and the normalized size of the induced cut is controlled by the constant  $c$ .

Let  $\alpha \triangleq \frac{c}{c+1}$ . The solution of (2.41) is the following closed-form expression:

$$\hat{\mathbf{h}} = (1 - \alpha) (I - \alpha W)^{-1} \mathbf{y}, \quad (2.42)$$

where  $D$  is a diagonal matrix with the  $i$ th entry being  $d_i$  and  $W$  is an adjacency matrix of  $G$  with the  $(i, j)$ th entry being  $w_{ij}$ .

The computational complexity of inverting the matrix in (2.42) is  $\Omega((m+u)^3)$ . Such a large complexity prevents the algorithm (2.41) from being applied to large full samples of at least tens of thousands of examples. In the last years a number of methods have been published that reduced the complexity of (2.42). Galeano and Herbster (2007) showed that if the underlying graph is a tree then the matrix inversion in (2.42) can be done in the linear time  $O(m + u)$ . Zhu and Lafferty (2005) assumed that full sample points are samples from a mixture model (e.g.

<sup>14</sup>The algorithms (2.41) of Wang and Zhang (2008) and of Zhou et al. (2004) can be extended to directed graphs (Zhou et al., 2005, 2005) and to hypergraphs (Agarwal et al., 2006; Zhou et al., 2007). Also Wang et al. (2007) showed an extension of (2.41) to the transductive multi-task setting.

Gaussian mixtures). Under such an assumption, the complexity of the matrix inversion in (2.42) is cubic in the number of mixtures.

Another approach to reducing the complexity of (2.42) was considered by Delalleau et al. (2005). In this paper, using  $k$ -farthest first search, the authors found a small number of the most informative unlabeled examples. Then they minimized the objective function (2.41) for the original training set and the reduced test set, consisting of the informative unlabeled examples. The original training set and the reduced test set, along with the latter's new labels, are used to find the labels of unlabeled examples that are not in the reduced test set. This is done using a very fast Nadaraya-Watson estimator.

The expression (2.42) allows two additional interpretations of the algorithm of (Wang & Zhang, 2008). The first interpretation is in terms of label propagation. Consider the following iterative process. Set  $\mathbf{h}(0) = \mathbf{y}$  and for any integer  $t > 0$ ,

$$\mathbf{h}(t+1) = \alpha W \mathbf{h}(t) + (1 - \alpha) \mathbf{y} . \quad (2.43)$$

It can be verified that  $\lim_{t \rightarrow \infty} \mathbf{h}(t) = \hat{\mathbf{h}}$ . The equation (2.43) describes the following iterative label propagation process. At each iteration a vertex  $v_i$  in  $G$  propagates its current soft label to its neighbors. The soft label  $h_i(t)$  of  $v_i$ , when propagated to  $v_j$  through the edge with the weight  $w_{ij}$ , diminishes its value by the factor  $w_{ij}$ . When the vertex  $v_j$  receives the scaled soft labels from its neighbors, it sums them to obtain a temporary soft classification  $\tilde{h}_j$ . The final soft classification  $h_j(t+1)$  of  $x_j$  is a weighted average of  $\tilde{h}_j$  and its initial label  $y_j$ . There exists a number of other label propagation mechanisms in graph-based algorithms. For additional examples see (Zhu et al., 2003; Macskassy & Provost, 2007; Culp & Michailidis, 2008; Galstyan & Cohen, 2008).

The second interpretation of (2.42) is in terms of a random walk over  $G$ . By Taylor expansion of  $(I - \alpha W)^{-1}$  we obtain that (2.42) is equivalent to

$$\hat{\mathbf{h}} = (1 - \alpha) \left( \sum_{i=0}^{\infty} (\alpha W)^i \right) \mathbf{y} . \quad (2.44)$$

Since for any  $1 \leq i \leq m+u$ ,  $\sum_j w_{ij} = 1$ , the matrix  $W$  is a transition probability matrix of the random walk over  $G$  and  $w_{ij}$  is a probability of a random transition from  $v_i$  to  $v_j$ . Hence the matrix  $\alpha W$  corresponds to a *lazy random walk*, in which at each step with probability  $(1 - \alpha)$  we stay at the current vertex and with probability  $\alpha$  we move according to  $W$ . Therefore, for any  $i \geq 0$ , the  $(s, t)$ th entry of  $(\alpha W)^i$  is a probability of arriving from  $v_s$  to  $v_t$  in exactly  $i$  steps. Consequently, the  $(s, t)$ th entry of  $\sum_{i=0}^{\infty} (\alpha W)^i$  is the probability of arriving from  $v_s$  to  $v_t$  using the above lazy random walk. Finally, recalling the structure of  $\mathbf{y}$ , we obtain that  $\text{sign}(\hat{h}_i) = 1$  if the probability of arriving from  $v_i$  to any positive training vertex is larger than the probability of arriving from  $v_i$  to any negative training vertex. Otherwise  $\text{sign}(\hat{h}_i) = -1$ . There exists a number of other graph-based algorithms performing similar random walks. For additional examples see

(Azran, 2007; Callut et al., 2008; Szummer & Jaakkola, 2002; Zhu et al., 2003; Zhou & Burges, 2007; Kim & Choi, 2007).

The learning algorithms described above assume that the weights of graph edges are proportional to the similarity between corresponding points. Tong and Jin (2007) showed graph-based algorithm that considers two graphs. In the first one the weights of the graph edges are proportional to the similarity between corresponding points, as previously. But in the second one the weights of the graph edges are inversely proportional to the similarity between corresponding points. Hence unlike all other graph-based algorithms, the algorithm of Tong and Jin is able to incorporate both similarity and dissimilarity information.

A useful method (Chapelle et al., 2003) to improve the performance of transductive algorithms is to modify the eigenvalues of Laplacians  $L$  and  $L_N$ . The modified Laplacian is substituted back into (2.39), (2.40) or (2.41) and is used for learning over the graph. The example of the commonly used spectral transformation is to raise the Laplacian eigenvalues to some fixed power (e.g., 2). It can be shown (e.g., see Johnson & Zhang, 2008) that such spectral transformation is effectively equivalent to the projection of the full sample points into its principal components and cleaning the noise in the coordinates of full sample points. For more examples of empirically successful spectral transformations see (Joachims, 2003; Johnson & Zhang, 2008) and the references therein. These transformations are based solely on the unlabeled data. There are also methods (see Zhu et al., 2006 and the references therein) that perform spectral transformations that depend on training/test set partition and on the training labels.

Shin et al. (2006) and Szlam et al. (2008) considered a jointly minimization of (2.41) over both  $\mathbf{h}$  and the graph weights  $w_{ij}$ . The resulting optimization problem is non-convex. Both these papers used heuristic methods for minimizing (2.41) over both  $\mathbf{h}$  and  $w_{ij}$ . Another approach for the joint minimization is to construct a number of base Laplacians  $L^{(1)}, \dots, L^{(t)}$  and to search for a good linear combination  $\sum_{j=1}^t \alpha_j L^{(j)}$  of them. The joint optimization over both  $\mathbf{h}$  and  $\alpha_j$  is considered in (Lanckriet et al., 2004; Argyriou et al., 2006).

Belkin and Niyogi (2004) considered another method for learning over graphs, which is motivated by spectral clustering (Ng et al., 2002). The method of Belkin and Niyogi considers  $k$  eigenvectors that correspond to the  $k$  smallest eigenvalues of Laplacian. Each example  $x_i$  from the full sample is represented by  $i$ th coordinates of  $k$  eigenvectors. Namely, full sample examples are embedded into  $k$ -dimensional space generated by  $k$  leading eigenvectors. Then the algorithm of Belkin and Niyogi finds a hyperplane in this space that minimizes the squared error over the labeled examples.<sup>15</sup> This hyperplane is then used to classify the test examples.

---

<sup>15</sup>Spectral clustering algorithms run k-means clustering in this space.

### 2.2.3 Mixed large margin and graph-based methods

While large-margin methods consider a global geometry of the full sample, graph-based methods mainly consider the local geometry. There exists a number of methods that combine these approaches, thus attempting to consider both global and local aspects of the full sample geometry. The first approach is to construct the kernel that is based on the graph used in graph-based algorithms. Such a kernel may then be used (e.g., see Chapelle & Zien, 2005; Dai & Yeung, 2007) in the kernelized version of (2.37). Another approach is to add the regularization term (2.39) to (2.37). The resulting algorithm, called Laplacian SVM, was introduced by Belkin et al. (2006).

### 2.2.4 Volume regularization

The size of the equivalence class is defined as the number of soft hypotheses in it. If this number is infinite (as in many hypothesis spaces), then we define the size of equivalence class to be its volume. This volume is computed in the dual space, with the soft hypotheses being points. The goal of volume regularization methods is to find a soft hypothesis that has a low empirical error and belongs to the equivalence class of large volume.

In general, the computation of the volume is very time-consuming.<sup>16</sup> Hence volume regularization methods actually try to approximate the volume of the equivalence class. Currently, all the existing volume approximations are heuristic, with no theoretical guarantees. The first approach for volume regularization was published by Vapnik (1982, Section 10.5). Vapnik's approach used a hypothesis space of hyperplanes and approximated the volume of the equivalence class by the minimal distance between positively and negatively classified examples.<sup>17</sup> Based on this approximation, Vapnik (1982, Section 10.5) constructed a structural risk minimization scheme that attempts to find a hyperplane with small empirical error and large margin. This scheme employs an empirical risk minimization procedure and hence in general is NP-hard. However, it can be approximated by the transductive SVM algorithm that is considered in Section 2.2.1.

Another attempt to approximate the volume of an equivalence class was made by Graepel et al. (2000). As in (Vapnik, 1982), Graepel et al. used the hypothesis space of hyperplanes. In addition, the algorithm of Graepel et al. assumes that in the kernel space the training examples can be separated, with no empirical error, by some hyperplane.<sup>18</sup> Graepel et al. considered equivalence classes

---

<sup>16</sup>While for convex bodies there exist algorithms (e.g., see Lovász & Vempala, 2006) for the computation of volume in polynomial time, their complexity is still too high for most machine learning applications.

<sup>17</sup>This distance is exactly twice the margin of the hyperplane.

<sup>18</sup>This assumption is mild, since for any set of labeled points there exists a kernel space where the training examples can be separated, with no empirical error, by some hyperplane.

induced by only  $m$  training examples and a single test example. Their algorithm has the following structure. The algorithm performs  $u$  iterations. At each iteration two possible labelings of the  $i$ th unlabeled example  $x_i$  are considered. If the (approximated) volume of the equivalence class of hyperplanes that has zero training error and label  $x_i$  with  $+1$  is larger than the one of the equivalence class of hyperplanes that has zero training error and label  $x_i$  with  $-1$ , then  $x_i$  is labeled with  $+1$ , otherwise it is labeled with  $-1$ . The heuristic approximation of the volume is done by the kernel billiard algorithm.

Recently, El-Yaniv et al. (2008) showed yet another way of approximating the volume of an equivalence class for the hypothesis spaces defined as  $\mathcal{H}_Q = \{\mathbf{h} : \mathbf{h}^T Q \mathbf{h} \leq 1\}$ , where  $Q$  is an  $(m+u) \times (m+u)$  positive definite matrix and is a hyperparameter. Their approximation and the resulting learning algorithm are based on the geometric interpretation of the space  $\mathcal{H}_Q$ . This algorithm is described in details in Chapter 6 of this thesis.

## 2.2.5 Gaussian processes

We start with a brief introduction to Gaussian processes (see Rasmussen & Williams, 2006 for more details) and then survey several approaches to incorporating unlabeled examples in them. Gaussian process (GP) is an infinite sequence of random variables such that each finite subset of them has multivariate Gaussian distribution. Gaussian process is characterized by the mean function  $m(\cdot)$  and covariance function  $k(\cdot, \cdot)$ . A common assumption in learning with Gaussian processes is that  $m(x) \equiv 0$ . Gaussian process regression assumes that the real-valued label  $y(x)$  of example  $x$  may be decomposed as

$$y(x) = f(x) + \epsilon, \quad (2.45)$$

where  $f$  is a Gaussian process and  $\epsilon$  is a Gaussian noise with mean 0 and variance  $\sigma^2$ . Let

$$K_{m+u} = \begin{pmatrix} K_{mm} & K_{mu} \\ K_{um} & K_{uu} \end{pmatrix}$$

be an  $(m+u) \times (m+u)$  matrix with the  $(i, j)$ th entry referred to as  $k(x_i, x_j)$ . It follows from (2.45) that given a full sample  $X_{m+u}$ , the vector of labels  $Y_{m+u}$  has a multivariate Gaussian distribution with mean 0 and covariance matrix  $K_{m+u} + \sigma I$ . Using the standard derivation from probability theory we obtain that

$$Y_u | X_m, Y_m, X_u \sim \mathcal{N}(K_{um} (K_{mm} + \sigma^2 I)^{-1} Y_m, K_{uu} + \sigma^2 I - K_{um} (K_{mm} + \sigma^2 I)^{-1} K_{mu}). \quad (2.46)$$

Hence in Gaussian process regression,  $Y_u = K_{um} (K_{mm} + \sigma^2 I)^{-1} Y_m$ . The matrix inversion in the last expression is very expensive and takes  $O(m^3)$  time. A

number of approximate GP regression methods were developed (see Rasmussen & Williams, 2006) that try to alleviate this issue.

Gaussian process classification assumes that the binary label  $y(x)$  of example  $x$  can be decomposed as

$$y(x) = \text{sign}(f(x) + \epsilon) , \quad (2.47)$$

where  $f$  and  $\epsilon$  are the same as in (2.45). Because of the sign function, in the classification case the vector  $Y_{m+u}$  does not have a multivariate Gaussian distribution and there is no closed form explicit expression for  $Y_u | X_m, Y_m, X_u$ . Instead, several approximation techniques are used (see Rasmussen & Williams, 2006) to compute  $Y_u$ . Thus Gaussian process regression is much easier than Gaussian process classification.

While the solution (2.46) is formulated for the transductive setting, its prediction for each test example is based solely on training examples and is independent of the unlabeled examples. Transduction GP regression methods try to alleviate this issue so that the prediction for each test example would depend on both labeled training examples  $(X_m, Y_m)$  and the unlabeled test examples  $X_u$ . Schwaighofer and Tresp (2003) showed a fast (and equivalent to (2.46)) way of computing  $Y_u$ , which is based on the knowledge of the entire test set. Zhu et al. (2003), Sindhwani et al. (2007) and Yu et al. (2008) defined the covariance function  $k(\cdot, \cdot)$  such that its value  $k(x_i, x_j)$  for any two training examples also depends on the entire set of  $X_u$  of the unlabeled examples. In addition, Gärtner et al. (2006) and Le et al. (2006) showed how to find good values for the hyperparameters of covariance function  $k(\cdot, \cdot)$  using the unlabeled examples. Lawrence and Jordan (2006) modified (2.47) by replacing the binary-output sign function with the ternary-output function. One the values of that function indicates that the example is unlabeled. Based on this modification, Lawrence and Jordan developed another transductive version of the GP classification algorithm.

## 2.2.6 Boosting

The idea of boosting is to combine a number of base hypotheses into an ensemble such that the accuracy of the ensemble is much higher than the accuracy of the individual hypotheses. There exist a number of boosting algorithms. In this section we consider the AnyBoost family of boosting algorithms (Mason et al., 2000) and survey its extensions that incorporate unlabeled examples. The AnyBoost family is very large and in particular it includes the well-known AdaBoost boosting algorithm (Freund & Schapire, 1997).

Let  $\mathcal{H} \subseteq \mathbb{R}^{m+u}$  be a space of base hypotheses and  $\text{lin}(\mathcal{H}) \subseteq \mathbb{R}^{m+u}$  be a space of linear combinations of base hypotheses from  $\mathcal{H}$ . Let  $\mathbf{h} \in \mathcal{H}$  be a base hypothesis and  $\tilde{\mathbf{h}} \in \text{lin}(\mathcal{H})$  be an ensemble of base hypotheses. Let  $\ell : \mathbb{R} \rightarrow \mathbb{R}$  be a differentiable loss function and  $\mathcal{L}(\tilde{\mathbf{h}}) \triangleq \frac{1}{m} \sum_{i=1}^m \ell(y_i \tilde{h}_i)$  be a training error of

**Input:** An algorithm  $\mathcal{A}(\tilde{\mathbf{h}})$  that accepts  $\tilde{\mathbf{h}} \in \text{lin}(\mathcal{H})$  and returns  $\mathbf{h} \in \mathcal{H}$  such that  $\langle -\nabla \mathcal{L}(\tilde{\mathbf{h}}), \mathbf{h} \rangle > 0$ .

**Output:** An ensemble  $\tilde{\mathbf{h}} \in \text{lin}(\mathcal{H})$  of base hypotheses.

```

for  $i = 1$  to  $T$  do
  Let  $\mathbf{h}(i) = \mathcal{A}(\tilde{\mathbf{h}})$ .
  if  $\langle -\nabla \mathcal{L}(\tilde{\mathbf{h}}), \mathbf{h}(i) \rangle \leq 0$  then
    Return  $\tilde{\mathbf{h}}$ .
  end if
  Choose weight  $\alpha_i$ .
  Set  $\tilde{\mathbf{h}} = \tilde{\mathbf{h}} + \alpha_i \mathbf{h}(i)$ .
end for

```

**Figure 2.1:** AnyBoost family of algorithms

$\tilde{h}$ . The AnyBoost family of algorithms is depicted in Figure 2.1. The algorithms in this family construct an ensemble  $\tilde{\mathbf{h}}$  by trying to minimize a loss  $\mathcal{L}(\tilde{\mathbf{h}})$ . The minimization is done by using the gradient descent method. At each iteration we would like to add to  $\tilde{\mathbf{h}}$  the negative gradient  $-\nabla \mathcal{L}(\tilde{\mathbf{h}})$  of the current ensemble, in order to to maximally reduce  $\mathcal{L}(\tilde{\mathbf{h}})$ . However, since  $-\nabla \mathcal{L}(\tilde{\mathbf{h}})$  is not necessary in  $\mathcal{H}$ , we add to  $\tilde{\mathbf{h}}$  a base hypothesis  $\mathbf{h}$ , which roughly has the same direction as the negative gradient.

In the inductive setting the loss  $\mathcal{L}(\tilde{\mathbf{h}})$  of the ensemble  $\tilde{\mathbf{h}}$  depends only on the labeled training examples. The extension of AnyBoost to the transductive setting is straightforward. Indeed, we only need to add to  $\mathcal{L}(\tilde{\mathbf{h}})$  the term that indicates the loss on the unlabeled examples. Two obvious extensions of  $\mathcal{L}(\tilde{\mathbf{h}})$  are the addition of the term that depends on the margin w.r.t. unlabeled examples (e.g. see Bennett et al., 2002 and the references therein) and the addition of the term that depends on the graph cut induced by the labeling of unlabeled examples (e.g., see Loeff et al., 2008 and the references therein). The modified loss function is then used to develop transductive instantiations of the AnyBoost scheme.

## 2.2.7 Methods based on the minimization of risk bounds

One approach for transductive learning is to develop an algorithm directly minimizing the risk bound. To date, all existing attempts to develop such algorithms considered PAC-Bayesian bounds (described in Section 2.1.2). The first such algorithm, developed by El-Yaniv and Gerzon (2005), is essentially a clustering algorithm with the minimum of the risk bounds (2.16) and (2.15) used to select the number of clusters. In particular, the algorithm of El-Yaniv and Gerzon constructs a prior over the hypothesis space of all possible clusterings with the

number of clusters up to  $k$  ( $k$  is a predefined constant). In this construction the hypotheses corresponding to clusterings with the small number of clusters have larger priors than the hypotheses corresponding to clusterings with the large number of clusters. For each number of clusters the algorithm of El-Yaniv and Gerzon constructs (in the unsupervised way) a clustering and then labels the points within each cluster according to the majority of the training labels that appear in it. Finally, the algorithm of El-Yaniv and Gerzon chooses among  $k$  clusterings the one that minimizes the minimum of (2.16) and (2.15) with the prior  $p(\mathbf{h})$  as defined above.

Another algorithm, based on the minimization of implicit risk bound that is similar to (2.16), was also developed by Banerjee and Langford (2004).

## 2.2.8 Statistical physics methods

One approach to labeling the test points is to represent the full sample as a system of magnets. In this system each example is a spin. The upside orientation of the spin corresponds to the positive label of the point and the downside orientation corresponds to the negative label. The configuration of the spins is a full sample hard labeling  $\mathbf{h} \in \{\pm 1\}^{m+u}$ . In this section we refer to  $\mathbf{h}$  as both a configuration and the full sample hard labeling. The system in configuration  $\mathbf{h}$  has an energy

$$E(\mathbf{h}) \triangleq - \sum_{i,j=1}^{m+u} J_{ij} h_i h_j - \sum_{i=1}^{m+u} \theta_i h_i , \quad (2.48)$$

where  $J_{ij}$  is a symmetric interaction energy between spins  $i$  and  $j$  and  $\theta_i$  is an energy of spin  $i$  induced by the external field. In the transductive model we assume that there is no external field over the unlabeled spins and hence  $\theta_i = 0$  for  $m+1 \leq i \leq m+u$ . For the labeled spins we set  $\theta_i$  to the value of the corresponding label. The value  $J_{ij}$  of the symmetrical interaction energy between spins  $i$  and  $j$  may be set to the weight  $w_{ij}$  of the edge between  $x_i$  and  $x_j$  in the underlying graph (see Section 2.2.2 for various methods of constructing such graphs). With such choices of  $\theta_i$  and  $J_{ij}$  the energy (2.48) is very similar to the objective function (2.41) used in graph-based transductive algorithms. However, in the statistical physics approach we do not find the labeling  $\mathbf{h}$  minimizing (2.48) (as in graph-based algorithms). Instead, we find the labeling  $\mathbf{h}$  in the following way.

Let  $T > 0$  be a hyperparameter, which is commonly interpreted as a temperature. We assume that the configuration  $\mathbf{h}$  is a random variable with Boltzmann distribution. Namely, the probability that the spins have configuration  $\mathbf{h}$  at temperature  $T$  is

$$\mathbf{P}_T(\mathbf{h}) = \exp\left(-\frac{\beta}{T} E(\mathbf{h})\right) , \quad (2.49)$$

where  $\beta$  is a Boltzmann constant. Then the marginal probability of the  $i$ th spin to be in state  $y_i$  is

$$\mathbf{P}_T(h_i = y_i) = \frac{1}{2} \sum_{\mathbf{s} \in \{\pm 1\}^{m+u}} \mathbf{P}_T(h_1 = s_1, \dots, h_{i-1} = s_{i-1}, h_i = y_i, h_{i+1} = s_{i+1}, \dots, h_{m+u} = s_{m+u}) , \quad (2.50)$$

and the mean value of the state of the  $i$ th spin at temperature  $T$  is

$$\langle h_i \rangle_T = \mathbf{P}_T(h_i = 1) - \mathbf{P}_T(h_i = -1) .$$

Statistical physics-motivated transductive algorithms aim to output a mean state vector,

$$\langle \mathbf{h} \rangle_T = (\langle h_1 \rangle_T, \dots, \langle h_{m+u} \rangle_T) ,$$

of soft classifications of the full sample. However, the direct computation of  $\langle \mathbf{h} \rangle_T$  is infeasible, since it involves the sum (2.50) over  $2^{m+u}$  elements. There are two attempts to compute  $\langle \mathbf{h} \rangle_T$  approximately. Getz et al. (2005) approximated (2.50) using Markov chain Monte-Carlo sampling. Wang et al. (2007) approximated the Boltzmann probability distribution (2.49) by another distribution, for which (2.50) is easy to compute.

## 2.2.9 Self-training methods

In self-training methods the learner learns in the iterative way. At each iteration the learner labels the subset of the currently unlabeled data and adds it to the training set that is used at the next iteration. Historically, self-training methods were one of the first approaches to utilize the unlabeled data. Currently self-training methods are largely divided into four groups, which are surveyed below.

### EM methods

In the transductive context Expectation-Maximization (EM) methods have the following high-level scheme. Suppose that transductive algorithm  $\mathcal{A}$  depends on some hyperparameter  $\alpha$ . For example,  $\alpha$  may be a hyperparameter of the distance metric. Let  $\mathbf{P}(\alpha)$  be a prior probability distribution over all possible values of  $\alpha$ . We denote by  $S'_u$  the unlabeled test set  $X_u$  along with the test labels  $Y'_u$  obtained at the last run of  $\mathcal{A}$ . Let  $\mathbf{P}(S_m, S'_u | \alpha)$  be the posterior probability distribution of obtaining the full sample  $S_m \cup S'_u$ , given  $\alpha$ . The functional form of the both prior and posterior probabilities should be specified after obtaining the unlabeled full sample  $X_{m+u}$  but before obtaining the actual training/test set partition and the labeling information.<sup>19</sup> EM methods seek the test labeling  $Y'_u$  and the value

<sup>19</sup>For examples, prior and posterior distributions may be two Gaussians depending on  $\alpha$ , and on both  $\alpha$  and  $S_m \cup S'_u$  respectfully.

of  $\alpha$  that maximize

$$\mathbf{P}(S_m, X_u \cup Y'_u \mid \alpha) \mathbf{P}(\alpha) . \quad (2.51)$$

Since this optimization problem is commonly non-convex, EM methods estimate the correct value of  $\alpha$  and find the labeling  $Y'_u$  of unlabeled examples in two steps, which are performed interchangeably. At the  $E$ -step the algorithm  $\mathcal{A}$  is run with the current value of  $\alpha$  and labels all the unlabeled examples. At the  $M$ -step the learner finds a value  $\hat{\alpha} = \arg \max_{\alpha'} \mathbf{P}(S_m, S'_u \mid \alpha') \mathbf{P}(\alpha')$  maximizing the posterior probability of obtaining the labeled full sample  $S_m \cup S'_u$ , and sets  $\alpha \triangleq \hat{\alpha}$ . The algorithm stops when there is no significant change in the value of  $\alpha$  between two consecutive  $M$ -steps. A number of instantiations of this scheme, with  $\mathcal{A}$  being a naïve Bayes classifier, can be found in (Nigam et al., 2000). The common disadvantage of the EM methods is that the solutions  $Y'_u$  and  $\alpha$  that they find are not guaranteed to be a global maximum of (2.51).

## Bootstrapping

Bootstrapping (Yarovsky, 1995) is probably one of the simplest methods for the usage of unlabeled examples. While originally bootstrapping was developed for semi-supervised setting, it also can be easily applied to the transductive one. In bootstrapping the labeling of unlabeled examples is done iteratively. At each iteration the learner runs a supervised learning algorithm  $\mathcal{A}$  to label the unlabeled examples. For each unlabeled example, along with the label,  $\mathcal{A}$  should output its confidence in this label. Then the learner adds to the training set the unlabeled examples, along with their new labels, which are labeled with high confidence, and proceeds to the next iteration. This iteration is repeated until there are no unlabeled examples that are labeled with high confidence. At this stage the algorithm  $\mathcal{A}$  is run on the training set consisting of the original one and the unlabeled examples that were labeled with high confidence. The resulting hypothesis is used to label the remaining unlabeled examples.

The bootstrapping scheme has many instantiations to particular algorithms. See, for example, (Taira & Haruno, 2001; Haffari & Sarkar, 2007) and the references therein. There is a number of results (e.g., see Haffari & Sarkar, 2007 and the references therein) showing when the bootstrapping scheme has an advantage, in the semi-supervised setting, over the underlying supervised algorithm  $\mathcal{A}$ . It would be interesting to obtain similar results for the transductive setting.

## Co-training with a single view

Co-training methods with a single view (Goldman & Zhou, 2000; Zhou & Li, 2007), originally motivated by bootstrapping, utilize two supervised learning algorithms,  $\mathcal{A}_1$  and  $\mathcal{A}_2$ . These algorithms are assumed to be different.<sup>20</sup>  $\mathcal{A}_1$  and

<sup>20</sup>If  $\mathcal{A}_1 = \mathcal{A}_2$ , then co-training with a single view reduces to the bootstrapping scheme.

$\mathcal{A}_2$  operate mostly on different training sets. Initially, the training sets of  $\mathcal{A}_1$  and  $\mathcal{A}_2$  are the same and consist of the labeled examples  $S_m$ . As other self-learning algorithms, co-training operates in an iterative way. At each iteration both  $\mathcal{A}_1$  and  $\mathcal{A}_2$  are run on their training sets and label the unlabeled examples. Then the examples that were labeled by  $\mathcal{A}_1$  with high-confidence are added, along with their new labels, to the training set of  $\mathcal{A}_2$ . Similarly, the examples that were labeled by  $\mathcal{A}_2$  with high-confidence are added, along with their new labels, to the training set of  $\mathcal{A}_1$ . This iteration is repeated until there are no unlabeled examples that are labeled with high confidence. At this stage the remaining unlabeled examples are labeled by a weighted average of  $\mathcal{A}_1$  and  $\mathcal{A}_2$ .

### Co-training with two views

Co-training with two views (Blum & Mitchell, 1998) is a generalization of co-training with a single view to the setting when each example has two different sets of features. Each such set is called a *view*. For example, suppose that the examples are web pages. Then the first view contains the textual information on the page and the second view contains the information about the hyperlinks pointing from/to the page.

Historically, the first algorithm for co-training with two views (Blum & Mitchell, 1998) was published two years before the first algorithm for co-training with a single view (Goldman & Zhou, 2000). The difference between the single view and the two views settings is that in the two views setting  $\mathcal{A}_1$  operates on the features from the first view and  $\mathcal{A}_2$  operates on the features from the second view.

Recently, co-training algorithms have been extended (Balcan & Blum, 2006; Zhou et al., 2007) to deal with a training set consisting of a single labeled example. It would be interesting to find out if similar extensions can be done with other transductive algorithms.

There is a number of results (e.g., see Balcan & Blum, 2006 and the references therein) showing when the co-training scheme with two views has an advantage, in the semi-supervised setting, over the underlying supervised algorithm  $\mathcal{A}$ . It would be interesting to obtain similar results for the transductive setting.

#### 2.2.10 Agreement-based methods

Agreement-based methods assume that there exist two hypothesis spaces  $\mathcal{H}_1$  and  $\mathcal{H}_2$ . These spaces may correspond to different views of the examples (as in co-training with two views), or they may operate on the same view but be very different. The example for the latter case is when both  $\mathcal{H}_1$  and  $\mathcal{H}_2$  are the spaces of large-margin hyperplanes but rely on different metrics (e.g.  $\mathcal{H}_1$  may use an RBF kernel and  $\mathcal{H}_2$  may use a polynomial kernel). Agreement-based methods, introduced by de Sa (1994), find two hypotheses  $\mathbf{h}^{(1)} \in \mathcal{H}_1$  and  $\mathbf{h}^{(2)} \in \mathcal{H}_2$

such that both these hypotheses have a low empirical error and also  $\mathbf{h}^{(1)}$  and  $\mathbf{h}^{(2)}$  mostly agree in their labeling of unlabeled examples. For example, with the squared error loss,  $\mathbf{h}^{(1)}$  and  $\mathbf{h}^{(2)}$  minimize the following objective function:

$$\sum_{i=1}^2 \sum_{j=1}^m \left( h_j^{(i)} - y_j \right)^2 + c \cdot \sum_{j=m+1}^{m+u} \left( h_j^{(1)} - h_j^{(2)} \right)^2, \quad (2.52)$$

where  $c > 0$  is a regularization constant. In practice, to achieve better empirical results, it is common (e.g., see Brefeld et al., 2006) to set  $\mathcal{H}_1$  and  $\mathcal{H}_2$  to be reproducing kernel Hilbert spaces (RKHS) and to add to (2.52) a regularization term  $\mu \sum_{i=1}^2 \|\mathbf{h}\|_{\mathcal{H}_i}$ , where  $\mu > 0$  is a regularization constant and  $\|\cdot\|_{\mathcal{H}_i}$  is a norm in the RKHS  $\mathcal{H}_i$ :

$$\left( \tilde{\mathbf{h}}^{(1)}, \tilde{\mathbf{h}}^{(2)} \right) = \arg \min_{\mathbf{h}^{(1)}, \mathbf{h}^{(2)}} \sum_{i=1}^2 \sum_{j=1}^m \left( h_j^{(i)} - y_j \right)^2 + c \cdot \sum_{j=m+1}^{m+u} \left( h_j^{(1)} - h_j^{(2)} \right)^2 + \mu \sum_{i=1}^2 \|\mathbf{h}\|_{\mathcal{H}_i}. \quad (2.53)$$

The final labeling generated by the agreement-based method of (2.53) is  $\frac{1}{2} \left( \tilde{\mathbf{h}}^{(1)} + \tilde{\mathbf{h}}^{(2)} \right)$ . Recently Sindhvani and Rosenberg (2008) showed that there exists a third RKHS  $\mathcal{H}_3$  such that the agreement-based optimization problem (2.53) is equivalent to the standard norm regularization in  $\mathcal{H}_3$ :

$$\tilde{\mathbf{h}} = \arg \min_{\mathbf{h} \in \mathcal{H}_3} \sum_{j=1}^m (h_j - y_j)^2 + \mu \|\mathbf{h}\|_{\mathcal{H}_3}. \quad (2.54)$$

The idea of the regularization by agreement on the unlabeled examples can also be applied to boosting (Leskes & Torenvliet, 2008) and Gaussian processes (Yu et al., 2008).

Agreement-based methods trace their roots to the co-training scheme. While performing a theoretical analysis of co-training with two views, Dasgupta et al. (2002) found that in the semi-supervised setting, under certain conditions, the maximum of the generalization error of any two hypotheses is bounded by their disagreement on the unlabeled data. The bound on the same nature has also appeared in (Leskes & Torenvliet, 2008). We conjecture that similar bounds can be also derived for the transductive setting.

### 2.2.11 Scalability issues

The common scenario of many applications of transductive learning is tens or hundreds of labeled examples and a huge, up to a million, number of unlabeled examples. We are aware of only two papers (Tsang & Kwok, 2007; Karlen et al., 2008) that present algorithms that are capable of processing such amounts of data. We also note that the scalability of most of the algorithms described in the previous section is limited to thousands of examples.

### 2.2.12 Empirical comparison of algorithms

As we showed in this survey, there is a large number of transductive algorithms. It is almost impossible to perform their extensive comparison in some unified setting of experiments. To our knowledge the largest comparison of a number of transductive algorithms was done in (Chapelle et al., 2006). In that experiment 11 transductive algorithms were compared using eight datasets. In addition, the experiment included the comparison with the inductive  $k$ -nearest neighbors and SVM algorithms. Among eight datasets, seven were small-scale ones, with up to 1500 examples, and one dataset was medium-scale, with 83000 examples. Only four out of 11 transductive algorithms were able to complete the experiment on the medium-scale dataset.

The experiment was run in a distributed way by the algorithms' authors. Then the result of the experiments were reported to the authors of (Chapelle et al., 2006). This distributed setting caused the differences between the computational efforts invested by different authors in the experiment. In particular, the algorithms were run on hyperparameter grids of different sizes. Also for some of the algorithms, the authors did not do model selection and reported only the errors for the best hyperparameters in hindsight. Ignoring these differences the best performance in this experiment was achieved by the SGT algorithm of Joachims (2003).

### 2.2.13 Summary

In this section we surveyed a number of algorithmic approaches for transductive learning. While the number of transductive algorithms has grown tremendously during the last five years, this field is still at its infancy and has yet to mature. Currently, there is no clear evidence which algorithms are the best ones for particular domains and applications. Moreover, there is only a partial empirical evidence about the domains where these algorithms are better than the best algorithms in the inductive model (e.g., SVM, inductive boosting, inductive Gaussian processes et al.). Finally, most of the described transductive algorithms have little, if any, theoretical basis. An interesting research direction would be to try to fill this gap.

## 2.3 Proof of Lemma 1

1. Let  $\mathbb{I}(\mathcal{A})$  be an indicator random variable, having value 1 if  $\mathcal{L}_u(\mathcal{A}) > B_1(\delta)$  and 0 otherwise. If the bound (2.1) holds, then for any full sample  $S_{m+u}$ ,  $\mathbf{E}_{(S_m, X_u) \sim \mathcal{U}(S_{m+u})} \mathbb{I}(\mathcal{A}) < \delta$ . In order to prove (2.2) it is sufficient to prove

that  $\mathbf{E}_{(S_m, X_u) \sim \mathcal{D}^{m+u}} \mathbb{I}(\mathcal{A}) < \delta$ . We have that

$$\begin{aligned} \mathbf{E}_{(S_m, X_u) \sim \mathcal{D}^{m+u}} \mathbb{I}(\mathcal{A}) &= \mathbf{E}_{S_{m+u} \sim \mathcal{D}^{m+u}} \mathbf{E}_{(S_m, X_u) \sim \mathcal{U}(S_{m+u})} \mathbb{I}(\mathcal{A}) \\ &< \mathbf{E}_{S_{m+u} \sim \mathcal{D}^{m+u}} \delta = \delta . \end{aligned}$$

2. The proof is very similar to the proof of the first part of the lemma. We have that

$$\begin{aligned} \mathbf{E}_{(S_m, X_u) \sim \mathcal{D}^{m+u}} \mathcal{L}_u(\mathcal{A}) &= \mathbf{E}_{S_{m+u} \sim \mathcal{D}^{m+u}} \mathbf{E}_{(S_m, X_u) \sim \mathcal{U}(S_{m+u})} \mathcal{L}_u(\mathcal{A}) \\ &< \mathbf{E}_{S_{m+u} \sim \mathcal{D}^{m+u}} B_2 = B_2 . \end{aligned}$$

# Chapter 3

## Concentration Inequalities for Functions over Partitions

In this chapter we develop novel concentration inequalities for functions over partitions and compare them to the several known ones. Our concentration inequalities are utilized in the derivation of the risk bounds in Chapters 4 and 5.

Denote by  $I_r^s$  the set of natural numbers  $\{r, r + 1, \dots, s\}$  ( $r < s$ ). Let  $\mathbf{Z} \triangleq \mathbf{Z}_1^{m+u} \triangleq (Z_1, \dots, Z_{m+u})$  be a *random permutation vector* where the variable  $Z_k$ ,  $k \in I_1^{m+u}$ , is the  $k$ th component of a permutation of  $I_1^{m+u}$  that is chosen uniformly at random. Any function  $f$  on permutations of  $I_1^{m+u}$  is called  $(m, u)$ -*permutation symmetric* if  $f(\mathbf{Z}) \triangleq f(Z_1, \dots, Z_{m+u})$  is symmetric on  $Z_1, \dots, Z_m$  as well as on  $Z_{m+1}, \dots, Z_{m+u}$ . Since  $\mathbf{Z}$  is a random permutation vector,  $f(\mathbf{Z})$  is a function over dependent random variables.

In this chapter we present novel concentration inequalities for  $(m, u)$ -permutation symmetric functions. Note that an  $(m, u)$ -permutation symmetric function is essentially a function over the partition of  $m + u$  items into sets of sizes  $m$  and  $u$ . Thus, the forthcoming concentration inequalities, while being stated for  $(m, u)$ -permutation symmetric functions, also hold in exactly the same form for functions over partitions. Conceptually it is more convenient to view our results as concentration inequalities for functions over partitions. However, from a technical point of view we find it more convenient to consider  $(m, u)$ -permutation symmetric functions.

Let  $\mathbf{Z}^{ij}$  be a perturbed permutation vector obtained by exchanging the values of  $Z_i$  and  $Z_j$  in  $\mathbf{Z}$ . The concentration inequalities developed in this chapter depend on the bound on the difference

$$|f(\mathbf{Z}) - f(\mathbf{Z}^{ij})| . \tag{3.1}$$

In Section 3.1 we develop concentration inequality depending on the bound on (3.1) which holds uniformly for all possible random permutations  $\mathbf{Z}$  and the choices of  $i$  and  $j$ . We denote this bound *strong permutation stability*. In Section 3.2 we relax this dependence and develop concentration inequality depending

on the bound on (3.1) which holds for most (but not for all) choices of  $\mathbf{Z}$ ,  $i$  and  $j$ . We denote this bound *weak permutation stability*.

The proofs of both inequalities utilize the martingale technique. This technique is in particular very convenient for the development of concentration inequalities for function of dependent random variables. We require the following standard definitions and facts about martingales.<sup>1</sup> Let  $\mathbf{B}_1^n \triangleq (B_1, \dots, B_n)$  be a sequence of random variables and  $\mathbf{b}_1^n \triangleq (b_1, \dots, b_n)$  be their respective values. The sequence  $\mathbf{W}_0^n \triangleq (W_0, W_1, \dots, W_n)$  is called a *martingale* w.r.t. the *underlying* sequence  $\mathbf{B}_1^n$  if for any  $i \in I_1^n$ ,  $W_i$  is a function of  $\mathbf{B}_1^i$  and  $\mathbf{E}_{B_i} \{W_i | \mathbf{B}_1^{i-1}\} = W_{i-1}$ . The sequence of random variables  $\mathbf{d}_1^n = (d_1, d_2, \dots, d_n)$ , where  $d_i \triangleq W_i - W_{i-1}$ , is called the *martingale difference sequence* of  $\mathbf{W}_n$ . An elementary fact is that  $\mathbf{E}_{B_i} \{d_i | \mathbf{B}_1^{i-1}\} = 0$ .

Let  $f(\mathbf{X}_1^n) \triangleq f(X_1, \dots, X_n)$  be an arbitrary function of  $n$  (possibly dependent) random variables. Let  $W_0 \triangleq \mathbf{E}_{\mathbf{X}_1^n} \{f(\mathbf{X}_1^n)\}$  and  $W_i \triangleq \mathbf{E}_{\mathbf{X}_1^n} \{f(\mathbf{X}_1^n) | \mathbf{X}_1^i\}$  for any  $i \in I_1^n$ . An elementary fact is that  $\mathbf{W}_0^n$  is a martingale w.r.t. the underlying sequence  $\mathbf{X}_1^n$ . Thus we can obtain a martingale from any function of (possibly dependent) random variables. This routine of obtaining a martingale from an arbitrary function is called *Doob's martingale process*. By the definition of  $W_n$  we have  $W_n = \mathbf{E}_{\mathbf{X}_1^n} \{f(\mathbf{X}_1^n) | \mathbf{X}_1^n\} = f(\mathbf{X}_1^n)$ . Consequently, to bound the deviation of  $f(\mathbf{X}_1^n)$  from its mean it is sufficient to bound the difference  $W_n - W_0$ . A fundamental inequality, providing such a bound, is Azuma inequality:

**Lemma 2 (Azuma (1967))** *Let  $\mathbf{W}_0^n$  be a martingale w.r.t.  $\mathbf{B}_1^n$  and  $\mathbf{d}_1^n$  be its difference sequences. Suppose that for all  $i \in I_1^n$ ,*

$$|d_i| \leq b_i . \quad (3.2)$$

Then

$$\mathbf{P}_{\mathbf{B}_1^n} \{W_n - W_0 > \epsilon\} < \exp \left( -\frac{\epsilon^2}{2 \sum_{i=1}^n b_i^2} \right) . \quad (3.3)$$

A different version of Azuma inequality was developed by McDiarmid:

**Lemma 3 (McDiarmid, 1989, Corollary 6.10)** *Let  $\mathbf{W}_0^n$  be a martingale w.r.t.  $\mathbf{B}_1^n$ . Let  $\mathbf{b}_1^n = (b_1, \dots, b_n)$  be the vector of possible values of the random variables  $B_1, \dots, B_n$ . Let*

$$r_i(\mathbf{b}_1^{i-1}) \triangleq \sup_{b_i} \{W_i : \mathbf{B}_1^{i-1} = \mathbf{b}_1^{i-1}, B_i = b_i\} - \inf_{b_i} \{W_i : \mathbf{B}_1^{i-1} = \mathbf{b}_1^{i-1}, B_i = b_i\} . \quad (3.4)$$

Let  $r^2(\mathbf{b}_1^n) \triangleq \sum_{i=1}^n (r_i(\mathbf{b}_1^{i-1}))^2$  and  $\widehat{r}^2 \triangleq \sup_{\mathbf{b}_1^n} r^2(\mathbf{b}_1^n)$ . Then,

$$\mathbf{P}_{\mathbf{B}_1^n} \{W_n - W_0 > \epsilon\} < \exp \left( -\frac{2\epsilon^2}{\widehat{r}^2} \right) . \quad (3.5)$$

---

<sup>1</sup>See, e.g., Chapter 12 of Grimmett and Stirzaker (1995), and Section 9.1 of Devroye et al. (1996) for more details.

Lemmas 2 and 3 differ in their preconditions, (3.2) vs. (3.4). Another difference is in constant 2, appearing in the exponent. We derive the weak permutation stability-based concentration inequality using Azuma inequality. The strong permutation stability-based concentration inequality may be derived using either Azuma or McDiarmid inequality. The last derivation given slightly better results (in terms of constants) and is presented in this chapter. The derivation of the weak permutation stability-based concentration inequality using McDiarmid inequality is an open question.

### 3.1 Inequality based on strong permutation stability

We start with the definition of strong permutation stability.

**Definition 4 (Strong Permutation Stability)** *Let  $\mathbf{Z}$  be a random permutation vector. A function  $f(\mathbf{Z})$  has strong permutation stability  $\beta$  if for any permutation  $\mathbf{Z}$ ,  $i \in I_1^m$  and  $j \in I_{m+1}^{m+u}$*

$$|f(\mathbf{Z}) - f(\mathbf{Z}^{ij})| \leq \beta . \quad (3.6)$$

The following theorem, which is a main result of this section, expresses the concentration of  $(m, u)$ -permutation symmetric function in terms of its strong permutation stability.

**Theorem 1** *Let  $\mathbf{Z}$  be a random permutation vector over  $I_1^{m+u}$ . Let  $f(\mathbf{Z})$  be an  $(m, u)$ -permutation symmetric function with strong permutation stability  $\beta$ . Then*

$$\mathbf{P}_{\mathbf{Z}} \{f(\mathbf{Z}) - \mathbf{E}_{\mathbf{Z}} \{f(\mathbf{Z})\} \geq \epsilon\} \leq \exp \left( -\frac{2\epsilon^2(m+u-1/2)}{mu\beta^2} \left( 1 - \frac{1}{2\max(m,u)} \right) \right) . \quad (3.7)$$

The proof of Theorem 1 is given in Section 3.4.1.

The right hand side of (3.7) is approximately  $\exp \left( -\frac{2\epsilon^2}{\beta^2} \left( \frac{1}{m} + \frac{1}{u} \right) \right)$ . There is a number of previously published concentration inequalities that are comparable with (3.7). These inequalities are valid for function classes that encompass the class of  $(m, u)$ -permutation symmetric functions. In the rest of this section we compare 1 with the previously published results.

A similar, but less tight inequality can be obtained by the reduction from the drawing of the permutation to the drawing of  $\min(m, u)$  independent random variables and application of the bounded difference inequality of McDiarmid (1989):

**Lemma 4** *Suppose that the conditions of Theorem 1 hold. Then*

$$\mathbf{P}_{\mathbf{Z}} \{f(\mathbf{Z}) - \mathbf{E}_{\mathbf{Z}} \{f(\mathbf{Z})\} \geq \epsilon\} \leq \exp \left( -\frac{2\epsilon^2}{\beta^2 \min(m, u)} \right) . \quad (3.8)$$

The proof of Lemma 4 appears in Section 3.4.2.

**Remark 1** *The inequalities developed in Section 5 of Talagrand (1995) imply a concentration inequality that is similar to (3.8), but with worse constants.*

The inequality (3.7) is defined for any  $(m, u)$ -permutation symmetric function  $f$ . By specializing  $f$  we obtain the following two concentration inequalities:

**Remark 2** *If  $g : I_1^{m+u} \rightarrow \{0, 1\}$  and  $f(\mathbf{Z}) = \frac{1}{u} \sum_{i=m+1}^{m+u} g(Z_i) - \frac{1}{m} \sum_{i=1}^m g(Z_i)$ , then  $\mathbf{E}_{\mathbf{Z}}\{f(\mathbf{Z})\} = 0$ . Moreover, for any  $i \in I_1^m$ ,  $j \in I_{m+1}^{m+u}$ ,  $|f(\mathbf{Z}) - f(\mathbf{Z}^{ij})| \leq \frac{1}{m} + \frac{1}{u}$ . Therefore, by specializing (3.7) for such  $f$  we obtain*

$$\mathbf{P}_{\mathbf{Z}} \left\{ \frac{1}{u} \sum_{i=m+1}^{m+u} g(Z_i) - \frac{1}{m} \sum_{i=1}^m g(Z_i) \geq \epsilon \right\} \leq \exp \left( -\frac{2\epsilon^2 m u (m+u-1/2)}{(m+u)^2} \left( 1 - \frac{1}{2 \max(m, u)} \right) \right). \quad (3.9)$$

*The right hand side of (3.9) is approximately  $\exp \left( -\frac{2\epsilon^2 m u}{m+u} \right)$ . The inequality (3.9) is an explicit (and looser) version of Vapnik's absolute bound (see El-Yaniv & Gerzon, 2005). We note that using (3.7) we were unable to obtain an explicit version of Vapnik's relative bound (inequality 10.14 of Vapnik, 1982).*

**Remark 3** *If  $g : I_1^{m+u} \rightarrow \{0, 1\}$  and  $f(\mathbf{Z}) = \frac{1}{m} \sum_{i=1}^m g(Z_i)$ , then  $\mathbf{E}_{\mathbf{Z}}\{f(\mathbf{Z})\} = \frac{1}{m+u} \sum_{i=1}^{m+u} g(Z_i)$ . Moreover, for any  $i \in I_1^m$ ,  $j \in I_{m+1}^{m+u}$ ,  $|f(\mathbf{Z}) - f(\mathbf{Z}^{ij})| \leq \frac{1}{m}$ . Therefore, by specializing (3.7) for such  $f$  we obtain*

$$\mathbf{P}_{\mathbf{Z}} \left\{ \frac{1}{m} \sum_{i=1}^m g(Z_i) - \frac{1}{m+u} \sum_{i=1}^{m+u} g(Z_i) \geq \epsilon \right\} \leq \exp \left( -\frac{2\epsilon^2 (m+u-1/2)m}{u} \left( 1 - \frac{1}{2 \max(m, u)} \right) \right). \quad (3.10)$$

*The right hand side of (3.10) is approximately  $\exp \left( -\frac{2\epsilon^2 (m+u)m}{u} \right)$  for sufficiently large values of  $m$  and  $u$ . Hence for large  $m$  and  $u$ , (3.10) is slightly tighter than the following inequality, which was developed by Serfling (1974):*

$$\mathbf{P}_{\mathbf{Z}} \left\{ \frac{1}{m} \sum_{i=1}^m g(Z_i) - \frac{1}{m+u} \sum_{i=1}^{m+u} g(Z_i) \geq \epsilon \right\} \leq \exp \left( -\frac{2\epsilon^2 (m+u)m}{u+1} \right).$$

*However for small values of  $m$  and  $u$  Serfling's inequality is slightly tighter than ours.*

## 3.2 Inequality based on weak permutation stability

We start with the definition of weak permutation stability:

**Definition 5 (Weak Permutation Stability)** Let  $\mathbf{Z}$  be a random permutation vector. A function  $f(\mathbf{Z})$  has weak permutation stability  $(\beta, \beta_1, \delta_1)$  if  $f$  has strong permutation stability  $\beta$  and

$$\mathbf{P}_{\mathbf{Z}, i \sim I_1^m, j \sim I_{m+1}^{m+u}} \{|f(\mathbf{Z}) - f(\mathbf{Z}^{ij})| \leq \beta_1\} \geq 1 - \delta_1 \quad , \quad (3.11)$$

where  $i \sim I$  denotes a choice of  $i \in I$  uniformly at random.

This weaker notion of stability only requires that  $|f(\mathbf{Z}) - f(\mathbf{Z}^{ij})|$  be bounded with respect to most exchanges, allowing for a  $\delta_1$ -fraction of outliers.

The following theorem, which is a main result of this section, expresses the concentration of  $(m, u)$ -permutation symmetric function in terms of its weak permutation stability.

**Theorem 2** Let  $\mathbf{Z}$  be a random permutation vector and  $f(\mathbf{Z})$  be an  $(m, u)$ -symmetric permutation function. Suppose that  $f(\mathbf{Z})$  has weak permutation stability  $(\beta, \beta_1, \delta_1)$ . Let  $\delta \in (0, 1)$  be given, and for  $i \in I_1^m$ , let  $\theta_i \in (0, 1)$ ,  $\Psi \triangleq \delta_1 \sum_{i=1}^m 1/\theta_i$  and  $b_i \triangleq \frac{((1-\theta_i)\beta_1 + \theta_i\beta)}{(m+u-i+1)(1-\Psi)}$ . If  $\Psi < 1$ , then with probability at least  $(1 - \delta) \cdot (1 - \Psi)$  over the choices of  $f$ ,

$$f(\mathbf{Z}) \leq \mathbf{E}_{\mathbf{Z}} \{f(\mathbf{Z})\} + u \sqrt{\frac{1}{2} \sum_{i=1}^m b_i^2 \ln \frac{1}{\delta}} \quad . \quad (3.12)$$

Note that the confidence level can be made arbitrarily small by selecting appropriate  $\theta_i$  and  $\delta_1$  (thus trading-off  $\beta_1$ ).

It follows from Definition 5 that  $\beta_1$  depends on  $\delta_1$ . Hence, the bound (3.12) depends on the following parameters:  $\delta_1, \theta_i, i \in I_1^m$ . It can be shown that if  $\beta_1 = O(1/m)$ ,  $\delta_1 = O(1/m^2)$  and  $\theta_i = O(1/m)$  for all  $i \in I_1^m$ , then the slack term in (3.12) is  $O(\sqrt{\ln(1/\delta)}/m)$  and the bound's confidence can be made arbitrarily close to 1.

### 3.3 Concluding Remarks

Concentration inequalities are the main building blocks in the development of risk bounds. In this section we derived two new concentration inequalities for  $(m, u)$ -permutation symmetric functions. These inequalities show that under some conditions,

$$\mathbf{P}_{\mathbf{Z}} \{f(\mathbf{Z}) - \mathbf{E}_{\mathbf{Z}} \{f(\mathbf{Z})\} > \epsilon\} \leq O(\exp(-\epsilon^2)) \quad . \quad (3.13)$$

Very interesting and challenging direction for a future work is to find a tighter upper bound on the left hand side of (3.13). In particular it is interesting to find out if it is possible to obtain that under certain conditions,

$$\mathbf{P}_{\mathbf{Z}} \{f(\mathbf{Z}) - \mathbf{E}_{\mathbf{Z}} \{f(\mathbf{Z})\} > \epsilon\} \leq O(\exp(-\epsilon)) \quad . \quad (3.14)$$

One possible approach to attacking this problem is to consider the variance information about the martingale difference sequences.

## 3.4 Proofs

### 3.4.1 Proof of Theorem 1

The proof of Lemma 1 is inspired by McDiarmid's proof of the bounded difference inequality for permutation graphs (McDiarmid, 1998, Section 3). Let  $\mathbf{W}_0^{m+u}$  be a martingale obtained from  $f(\mathbf{Z})$  by Doob's martingale process, namely  $W_0 \triangleq \mathbf{E}_{\mathbf{Z}_1^{m+u}} \{f(\mathbf{Z}_1^{m+u})\}$  and  $W_i \triangleq \mathbf{E}_{\mathbf{Z}_1^{m+u}} \{f(\mathbf{Z}_1^{m+u}) | \mathbf{Z}_1^i\}$ . We compute the upper bound on  $\hat{r}^2$  and apply Lemma 3.

Fix  $i$ ,  $i \in I_1^m$ . Let  $\boldsymbol{\pi}_1^{m+u} = \pi_1, \dots, \pi_{m+u}$  be a specific permutation of  $I_1^{m+u}$  and  $\pi'_i \in \{\pi_{i+1}, \dots, \pi_{m+u}\}$ . Let  $p_1 \triangleq \mathbf{P}_{j \sim I_{i+1}^{m+u}} \{j \in I_{i+1}^m\} = \frac{m-i}{m+u-i}$  and  $p_2 \triangleq \mathbf{P}_{j \sim I_{i+1}^{m+u}} \{j \in I_{m+1}^{m+u}\} = 1 - p_1 = \frac{u}{m+u-i}$ . We have

$$r_i(\boldsymbol{\pi}_1^{i-1}) = \sup_{\pi_i} \{W_i : \mathbf{B}_1^{i-1} = \boldsymbol{\pi}_1^{i-1}, B_i = \pi_i\} - \inf_{\pi_i} \{W_i : \mathbf{B}_1^{i-1} = \boldsymbol{\pi}_1^{i-1}, B_i = \pi_i\} \quad (3.15)$$

$$\begin{aligned} &= \sup_{\pi_i, \pi'_i} \left\{ \mathbf{E}_{\mathbf{Z}} \{f(\mathbf{Z}) \mid \mathbf{Z}_1^{i-1} = \boldsymbol{\pi}_1^{i-1}, Z_i = \pi_i\} \right. \\ &\quad \left. - \mathbf{E}_{\mathbf{Z}} \{f(\mathbf{Z}) \mid \mathbf{Z}_1^{i-1} = \boldsymbol{\pi}_1^{i-1}, Z_i = \pi'_i\} \right\} \\ &= \sup_{\pi_i, \pi'_i} \left\{ \mathbf{E}_{j \sim I_{i+1}^{m+u}} \mathbf{E}_{\mathbf{Z}} \{f(\mathbf{Z}) \mid \mathbf{Z}_1^{i-1} = \boldsymbol{\pi}_1^{i-1}, Z_i = \pi_i, Z_j = \pi'_i\} \right. \\ &\quad \left. - \mathbf{E}_{j \sim I_{i+1}^{m+u}} \mathbf{E}_{\mathbf{Z}} \{f(\mathbf{Z}^{ij}) \mid \mathbf{Z}_1^{i-1} = \boldsymbol{\pi}_1^{i-1}, Z_i = \pi_i, Z_j = \pi'_i\} \right\} \\ &= \sup_{\pi_i, \pi'_i} \left\{ \mathbf{E}_{j \sim I_{i+1}^{m+u}} \mathbf{E}_{\mathbf{Z}} \{f(\mathbf{Z}) - f(\mathbf{Z}^{ij}) \mid \mathbf{Z}_1^{i-1} = \boldsymbol{\pi}_1^{i-1}, Z_i = \pi_i, Z_j = \pi'_i\} \right\} \end{aligned} \quad (3.16)$$

$$\begin{aligned} &= \sup_{\pi_i, \pi'_i} \left\{ p_1 \cdot \mathbf{E}_{\mathbf{Z}, j \sim I_{i+1}^m} \{f(\mathbf{Z}) - f(\mathbf{Z}^{ij}) \mid \mathbf{Z}_1^{i-1} = \boldsymbol{\pi}_1^{i-1}, Z_i = \pi_i, Z_j = \pi'_i\} \right. \\ &\quad \left. + p_2 \cdot \mathbf{E}_{\mathbf{Z}, j \sim I_{m+1}^{m+u}} \{f(\mathbf{Z}) - f(\mathbf{Z}^{ij}) \mid \mathbf{Z}_1^{i-1} = \boldsymbol{\pi}_1^{i-1}, Z_i = \pi_i, Z_j = \pi'_i\} \right\} \end{aligned} \quad (3.17)$$

Since  $f(\mathbf{Z})$  is  $(m, u)$ -permutation symmetric function, the expectation in (3.17) is zero. Therefore,

$$\begin{aligned} r_i(\boldsymbol{\pi}_1^{i-1}) &= \sup_{\pi_i, \pi'_i} \left\{ p_2 \cdot \mathbf{E}_{\mathbf{Z}, j \sim I_{m+1}^{m+u}} \{f(\mathbf{Z}) - f(\mathbf{Z}^{ij}) \mid \mathbf{Z}_1^{i-1} = \boldsymbol{\pi}_1^{i-1}, Z_i = \pi_i, Z_j = \pi'_i\} \right\} \\ &\leq \frac{u\beta}{m+u-i}. \end{aligned}$$

Since  $f(\mathbf{Z})$  is  $(m, u)$ -permutation symmetric, it also follows from (3.16) that for  $i \in I_{m+1}^{m+u}$ ,  $r_i(\boldsymbol{\pi}_1^{i-1}) = 0$ . It can be verified that for any  $j > 1/2$ ,  $\frac{1}{j^2} \leq \int_{j-1/2}^{j+1/2} \frac{1}{t^2} dt$ ,

and therefore,

$$\begin{aligned}\hat{r}^2 &= \sup_{\boldsymbol{\pi}_1^{m+u}} \sum_{i=1}^{m+u} (r_i(\boldsymbol{\pi}_1^{i-1}))^2 \leq \sum_{i=1}^m \left( \frac{u\beta}{m+u-i} \right)^2 = u^2\beta^2 \sum_{j=u}^{m+u-1} \frac{1}{j^2} \\ &\leq u^2\beta^2 \int_{u-1/2}^{m+u-1/2} \frac{1}{t^2} dt = \frac{mu^2\beta^2}{(u-1/2)(m+u-1/2)}.\end{aligned}\quad (3.18)$$

By applying Lemma 3 with the bound (3.18) we obtain

$$\mathbf{P}_{\mathbf{Z}} \{f(\mathbf{Z}) - \mathbf{E}_{\mathbf{Z}} \{f(\mathbf{Z})\} \geq \epsilon\} \leq \exp\left(-\frac{2\epsilon^2(u-1/2)(m+u-1/2)}{mu^2\beta^2}\right). \quad (3.19)$$

The entire derivation is symmetric in  $m$  and  $u$ . Therefore, we also have

$$\mathbf{P}_{\mathbf{Z}} \{f(\mathbf{Z}) - \mathbf{E}_{\mathbf{Z}} \{f(\mathbf{Z})\} \geq \epsilon\} \leq \exp\left(-\frac{2\epsilon^2(m-1/2)(m+u-1/2)}{m^2u\beta^2}\right). \quad (3.20)$$

By taking the tightest bound from (3.19) and (3.20) we obtain the statement of the lemma.

### 3.4.2 Proof of Lemma 4

We consider the following algorithm<sup>2</sup> (named **RANDPERM**) for drawing the first  $m$  elements  $\{Z_i\}_{i=1}^m$  of the random permutation  $\mathbf{Z}$  of  $I_1^{m+u}$ :

---

**Algorithm 1** **RANDPERM** - draw the first  $m$  elements of the random permutation of  $m+u$  elements.

---

Let  $Z_i = i$  for any  $i \in I_1^{m+u}$ .  
**for**  $i = 1$  to  $m$  **do**  
    Draw  $d_i$  uniformly from  $I_i^{m+u}$ .  
    Swap the values of  $Z_i$  and  $Z_{d_i}$ .  
**end for**

---

The algorithm **RANDPERM** is an abridged version of the procedure of drawing a random permutation of  $n$  elements by drawing  $n-1$  non-identically distributed independent random variables, presented in Section 5 of Talagrand (1995) (which according to Talagrand is due to Maurey (1979)).

---

<sup>2</sup>Another algorithm for generating random permutation from independent draws was presented in Appendix B of Lanckriet et al. (2004). This algorithm draws a random permutation by means of drawing  $m+u$  independent random variables. Since we only deal with  $(m, u)$ -permutation symmetric functions, we are only interested in the first  $m$  elements of the random permutation. The algorithm of Lanckriet et al. needs  $m+u$  draws of independent random variables to define the above  $m$  elements. The algorithm **RANDPERM**, presented in this section, needs only  $m$  draws. If we use the algorithm of Lanckriet et al. instead of **RANDPERM**, the forthcoming bound (3.23) would have the term  $m+u$  instead of  $m$ . This change, in turn, would result in a non-convergent risk bound being derived using our techniques.

**Lemma 5** *The algorithm RANDPERM performs a uniform draw of the first  $m$  elements  $Z_1, \dots, Z_m$  of the random permutation  $\mathbf{Z}$ .*

**Proof:** The proof is by induction on  $m$ . If  $m = 1$ , then a single random variable  $d_1$  is uniformly drawn among  $I_{m+u}$ , and therefore,  $Z_1$  has a uniform distribution over  $I_1^{m+u}$ . Let  $\mathbf{d}_1^m \triangleq d_1, \dots, d_m$ . Suppose the claim holds for all  $m_1 < m$ . For any two possible values  $\boldsymbol{\pi}_1^m \triangleq \pi_1, \dots, \pi_m$  and  $\boldsymbol{\pi}'_1^m \triangleq \pi'_1, \dots, \pi'_m$  of  $Z_1, \dots, Z_m$ , we have

$$\begin{aligned}
\mathbf{P}_{\mathbf{d}_1^m} \{ \mathbf{Z}_1^m = \boldsymbol{\pi}_1^m \} &= \mathbf{P}_{\mathbf{d}_1^{m-1}} \{ \mathbf{Z}_1^{m-1} = \boldsymbol{\pi}_1^{m-1} \} \cdot \mathbf{P}_{d_m} \{ Z_m = \pi_m \mid \mathbf{Z}_1^{m-1} = \boldsymbol{\pi}_1^{m-1} \} \\
&= \mathbf{P}_{\mathbf{d}_1^{m-1}} \{ \mathbf{Z}_1^{m-1} = \boldsymbol{\pi}'_1^{m-1} \} \cdot \frac{1}{u+1} \\
&= \mathbf{P}_{\mathbf{d}_1^{m-1}} \{ \mathbf{Z}_1^{m-1} = \boldsymbol{\pi}'_1^{m-1} \} \cdot \mathbf{P}_{d_m} \{ Z_m = \pi'_m \mid \mathbf{Z}_1^{m-1} = \boldsymbol{\pi}'_1^{m-1} \} \\
&= \mathbf{P}_{\mathbf{d}_1^m} \{ \mathbf{Z}_1^m = \boldsymbol{\pi}'_1^m \} .
\end{aligned} \tag{3.21}$$

The equality (3.21) follows from the inductive assumption and the definition of  $d_m$ .  $\square$

Consider any  $(m, u)$ -permutation symmetric function  $f = f(\mathbf{Z})$  over random permutations  $\mathbf{Z}$ . Using the algorithm RANDPERM we can represent any random permutation  $\mathbf{Z}$  as a function  $g(\mathbf{d})$  of  $m$  independent random variables. The function  $g(\mathbf{d})$  can be considered as an operation of the algorithm RANDPERM with the values of random draws given by  $\mathbf{d}$ . The next lemma relates the Lipschitz constant of the function  $f(g(\mathbf{d}))$  to the Lipschitz constant of  $f(\mathbf{Z})$ :

**Lemma 6** *Let  $f(\mathbf{Z})$  be an  $(m, u)$ -permutation symmetric function of random permutation  $\mathbf{Z}$ . Suppose that for all  $i \in I_1^m$ ,  $j \in I_{m+1}^{m+u}$ ,  $|f(\mathbf{Z}) - f(\mathbf{Z}^{ij})| \leq \beta$ . Let  $d'_i$  be an independent draw of the random variable  $d_i$ . Then for any  $i \in I_1^m$ ,*

$$|f(g(d_1, \dots, d_{i-1}, d_i, d_{i+1}, \dots, d_m)) - f(g(d_1, \dots, d_{i-1}, d'_i, d_{i+1}, \dots, d_m))| \leq \beta . \tag{3.22}$$

**Proof:** The values of  $\mathbf{d} \triangleq (d_1, \dots, d_i, \dots, d_m)$  and  $\mathbf{d}' \triangleq (d_1, \dots, d'_i, \dots, d_m)$  induce, respectively, the first  $m$  values<sup>3</sup>  $\mathbf{Z}_1^m = \{Z_1, \dots, Z_m\}$  and  $\mathbf{Z}'_1^m = \{Z'_1, \dots, Z'_m\}$  of the two dependent permutations of  $I_1^{m+u}$ . Since  $f$  is  $(m, u)$ -permutation symmetric, its value is uniquely determined by the value of  $\mathbf{Z}_1^m$ . We prove that the change of  $d_i$  by  $d'_i$  results in a change of a single element in  $\mathbf{Z}_1^m$ . Combined with the property of  $|f(\mathbf{Z}) - f(\mathbf{Z}^{ij})| \leq \beta$ , this will conclude the proof of (3.22).

We refer to  $\mathbf{d}$  and  $\mathbf{d}'$  as, respectively, ‘old’ and ‘new’ draws. Consider the operation of RANDPERM with the draws  $\mathbf{d}$  and  $\mathbf{d}'$ . Let  $\pi_i, \pi_{d_i}$  and  $\pi_{d'_i}$  be the values

---

<sup>3</sup>For notational convenience in this section, we refer to  $\mathbf{Z}_1^m$  as a set of values and not as a vector of values (as is done in other sections).

of, respectively,  $Z_i$ ,  $Z_{d_i}$  and  $Z_{d'_i}$  just *before* the  $i$ th iteration of RANDPERM. Note that  $d_i \geq i$  and  $d'_i \geq i$ . In the old permutation, *after* the  $i$ th iteration  $Z_i = \pi_{d_i}$ ,  $Z_{d_i} = \pi_i$  and  $Z_{d'_i} = \pi_{d'_i}$ . In the new permutation, *after* the  $i$ th iteration  $Z_i = \pi_{d'_i}$ ,  $Z_{d_i} = \pi_{d_i}$  and  $Z_{d'_i} = \pi_i$ . After the  $i$ th iteration of RANDPERM the value of  $Z_i$  remains intact. However the values of  $Z_{d_i}$  and  $Z_{d'_i}$  may change. In particular the values of  $\pi_{d_i}$  and  $\pi_i$  may be among  $Z_{i+1}, \dots, Z_m$  at the end of the run of RANDPERM. We have four cases:

**Case 1** If  $\pi_{d'_i} \notin \mathbf{Z}_1^m$  and  $\pi_i \notin \mathbf{Z}_1^m$  then  $\pi_{d_i} \notin \mathbf{Z}_1^m$ ,  $\pi_i \notin \mathbf{Z}_1^m$  and  $\mathbf{Z}_1^m = \mathbf{Z}_1^m \setminus \{\pi_{d_i}\} \cup \{\pi_{d'_i}\}$ .

**Case 2** If  $\pi_{d'_i} \in \mathbf{Z}_1^m$  and  $\pi_i \in \mathbf{Z}_1^m$  then  $\pi_{d_i} \in \mathbf{Z}_1^m$ ,  $\pi_i \in \mathbf{Z}_1^m$  and  $\mathbf{Z}_1^m = \mathbf{Z}_1^m$ .

**Case 3** If  $\pi_i \in \mathbf{Z}_1^m$  and  $\pi_{d'_i} \notin \mathbf{Z}_1^m$  then  $\pi_{d_i} \in \mathbf{Z}_1^m$ ,  $\pi_i \notin \mathbf{Z}_1^m$  and  $\mathbf{Z}_1^m = \mathbf{Z}_1^m \setminus \{\pi_i\} \cup \{\pi_{d'_i}\}$ .

**Case 4** If  $\pi_{d'_i} \in \mathbf{Z}_1^m$  and  $\pi_i \notin \mathbf{Z}_1^m$  then  $\pi_i \in \mathbf{Z}_1^m$ ,  $\pi_{d_i} \notin \mathbf{Z}_1^m$  and  $\mathbf{Z}_1^m = \mathbf{Z}_1^m \setminus \{\pi_{d_i}\} \cup \{\pi_i\}$ .

□

We apply a bounded difference inequality of McDiarmid (1989) to  $f(g(\mathbf{d}))$  and obtain

$$\mathbf{P}_{\mathbf{d}} \{f(g(\mathbf{d})) - \mathbf{E}_{\mathbf{d}} \{f(g(\mathbf{d}))\} \geq \epsilon\} \leq \exp\left(-\frac{2\epsilon^2}{\beta^2 m}\right). \quad (3.23)$$

Since  $f(\mathbf{Z})$  is a  $(m, u)$ -permutation symmetric, it follows from (3.23) that

$$\mathbf{P}_{\mathbf{Z}} \{f(\mathbf{Z}) - \mathbf{E}_{\mathbf{Z}} \{f(\mathbf{Z})\} \geq \epsilon\} \leq \exp\left(-\frac{2\epsilon^2}{\beta^2 m}\right). \quad (3.24)$$

Since the entire derivation is symmetric in  $m$  and  $u$  we also have

$$\mathbf{P}_{\mathbf{Z}} \{f(\mathbf{Z}) - \mathbf{E}_{\mathbf{Z}} \{f(\mathbf{Z})\} \geq \epsilon\} \leq \exp\left(-\frac{2\epsilon^2}{\beta^2 u}\right). \quad (3.25)$$

The proof of Lemma 4 is completed by taking the minimum of the bounds (3.24) and (3.25).

### 3.4.3 Proof of Theorem 2

Let  $\mathbf{W}_0^{m+u}$  be a martingale generated from  $f(\mathbf{Z})$  by Doob's process. We derive bounds on the martingale differences  $d_i$ ,  $i \in I_1^{m+u}$ , and apply Lemma 3.

Let  $\boldsymbol{\pi}_1^{m+u} = \pi_1, \dots, \pi_{m+u}$  be a specific permutation of  $I_1^{m+u}$ . In the proof we use the following shortcut:  $\mathbf{Z}_1^r = \boldsymbol{\pi}_1^r$  abbreviates the  $r$  equalities  $Z_1 = \pi_1, \dots, Z_r = \pi_r$

$\pi_r$ . Let  $\theta_i$  be given. For  $r \in I_1^m$ , we say that the prefix  $\pi_1^r$  of a permutation  $\pi_1^{m+u}$  is  $(r, \theta_r)$ -admissible (w.r.t. a fixed  $\beta_1$ ) if it guarantees that

$$\mathbf{P}_{\mathbf{Z}, j \sim I_{m+1}^{m+u}} \{ |f(\mathbf{Z}) - f(\mathbf{Z}^{rj})| \leq \beta_1 \mid \mathbf{Z}_1^r = \pi_1^r \} \geq 1 - \theta_r . \quad (3.26)$$

If the prefix  $\pi_1^r$  does not satisfy (3.26), we say that it is not  $(r, \theta_r)$ -admissible. Let  $\zeta(r, \theta_r)$  be the probability that  $\mathbf{Z}_1^r$  is not  $(r, \theta_r)$ -admissible. Our goal is to bound  $\zeta(r, \theta_r)$ . For any fixed  $1 \leq r \leq m$  we have,

$$\begin{aligned} t(r) &\stackrel{\Delta}{=} \mathbf{P}_{\mathbf{Z}, j \sim I_{m+1}^{m+u}} \{ |f(\mathbf{Z}) - f(\mathbf{Z}^{rj})| > \beta_1 \} \\ &= \sum_{\substack{\text{all possible} \\ \text{prefixes } \pi_1^r}} \left( \mathbf{P}_{\mathbf{Z}, j \sim I_{m+1}^{m+u}} \{ |f(\mathbf{Z}) - f(\mathbf{Z}^{rj})| > \beta_1 \mid \mathbf{Z}_1^r = \pi_1^r \} \cdot \mathbf{P}_{\mathbf{Z}} \{ \mathbf{Z}_1^r = \pi_1^r \} \right) \\ &\geq \sum_{\substack{\text{non-} \\ \text{admissible} \\ \text{prefixes } \pi_1^r}} \left( \mathbf{P}_{\mathbf{Z}, j \sim I_{m+1}^{m+u}} \{ |f(\mathbf{Z}) - f(\mathbf{Z}^{rj})| > \beta_1 \mid \mathbf{Z}_1^r = \pi_1^r \} \cdot \mathbf{P}_{\mathbf{Z}} \{ \mathbf{Z}_1^r = \pi_1^r \} \right) \\ &\geq \theta_r \cdot \sum_{\substack{\text{non-admissible} \\ \text{prefixes } \pi_1^r}} \mathbf{P}_{\mathbf{Z}} \{ \mathbf{Z}_1^r = \pi_1^r \} = \theta_r \zeta(r, \theta_r) . \end{aligned} \quad (3.27)$$

Since  $f(\mathbf{Z})$  is  $(m, u)$ -permutation symmetric,  $t(r) = t$  is constant. Since  $f(\mathbf{Z})$  has weak permutation stability  $(\beta, \beta_1, \delta_1)$ ,

$$\delta_1 \geq \mathbf{P}_{\mathbf{Z}, i \sim I_1^m, j \sim I_{m+1}^{m+u}} \{ |f(\mathbf{Z}) - f(\mathbf{Z}^{ij})| > \beta_1 \} = \sum_{r=1}^m \frac{1}{m} \cdot t(r) = t \geq \theta_r \zeta(r, \theta_r) . \quad (3.28)$$

Consequently,  $\zeta(r, \theta_r) \leq \delta_1 / \theta_r$ . Our next goal is to bound  $d_r$  for  $(r, \theta_r)$ -admissible prefixes. For any  $1 \leq r \leq m + u$  we have

$$\begin{aligned} |d_r| &= |W_r - W_{r-1}| = |\mathbf{E}_{\mathbf{Z}} \{ f(\mathbf{Z}) \mid \mathbf{Z}_1^r = \pi_1^r \} - \mathbf{E}_{\mathbf{Z}} \{ f(\mathbf{Z}) \mid \mathbf{Z}_1^{r-1} = \pi_1^{r-1} \}| \\ &= |\mathbf{E}_{\mathbf{Z}} \{ f(\mathbf{Z}^{il(k)}) - f(\mathbf{Z}) \mid \mathbf{Z}_1^{r-1} = \pi_1^{r-1} \}| \\ &= \left| \mathbf{E}_{\mathbf{Z}, j \sim I_i^{m+u}} \{ f(\mathbf{Z}) - f(\mathbf{Z}^{rj}) \mid \mathbf{Z}_1^r = \pi_1^r \} \right| \end{aligned} \quad (3.29)$$

$$\leq \mathbf{E}_{\mathbf{Z}, j \sim I_r^{m+u}} \{ |f(\mathbf{Z}^{rj}) - f(\mathbf{Z})| \mid \mathbf{Z}_1^r = \pi_1^r \} \quad (3.30)$$

$$\begin{aligned} &= \mathbf{P}_{j \sim I_r^{m+u}} \{ j \in I_r^m \} \cdot \mathbf{E}_{\mathbf{Z}, j \sim I_r^m} \{ |f(\mathbf{Z}^{rj}) - f(\mathbf{Z})| \mid \mathbf{Z}_1^r = \pi_1^r \} \\ &\quad + \mathbf{P}_{j \sim I_r^{m+u}} \{ j \in I_{m+1}^{m+u} \} \cdot \mathbf{E}_{\mathbf{Z}, j \sim I_{m+1}^{m+u}} \{ |f(\mathbf{Z}^{rj}) - f(\mathbf{Z})| \mid \mathbf{Z}_1^r = \pi_1^r \} \\ &= \mathbf{P}_{j \sim I_r^{m+u}} \{ j \in I_{m+1}^{m+u} \} \cdot \mathbf{E}_{\mathbf{Z}, j \sim I_{m+1}^{m+u}} \{ |f(\mathbf{Z}^{rj}) - f(\mathbf{Z})| \mid \mathbf{Z}_1^r = \pi_1^r \} . \end{aligned} \quad (3.31)$$

The equality (3.31) follows because the expectation in (3.30) is zero since  $f$  is  $(m, u)$ -permutation symmetric. Since  $f(\mathbf{Z})$  is  $(m, u)$ -permutation symmetric, it follows from (3.29) that for any  $r \in I_{m+1}^{m+u}$ ,  $d_r = 0$ . If  $r \in I_1^m$  and  $\pi_1^r$  is  $(r, \theta_r)$ -admissible then the expectation in (3.31) is bounded by  $(1 - \theta_r)\beta_1 + \theta_r\beta$ . Hence

for all  $(r, \theta_r)$ -admissible prefixes  $\boldsymbol{\pi}_1^r$ ,  $r \in I_1^m$ ,

$$|d_r| \leq \frac{u((1-\theta_r)\beta_1 + \theta_r\beta)}{m+u-r+1} . \quad (3.32)$$

A permutation  $\boldsymbol{\pi}_1^{m+u}$  is *good* if for all  $r \in I_1^m$  its  $r$ -prefixes,  $\boldsymbol{\pi}_1^r$ , are admissible (w.r.t. the corresponding  $\theta_r$ ). Since  $\zeta(r, \theta_r) \leq \delta_1/\theta_r$ , we have

$$\mathbf{P}_{\mathbf{Z}} \{\mathbf{Z} \text{ not good}\} \leq \sum_{r=1}^m \mathbf{P}_{\mathbf{Z}} \{\mathbf{Z}_1^r \text{ not admissible}\} = \sum_{r=1}^m \zeta(r, \theta_r) \leq \sum_{r=1}^m \frac{\delta_1}{\theta_r} = \Psi . \quad (3.33)$$

Thus, with probability at least  $1 - \Psi$ , the random permutation  $\mathbf{Z}$  is good, in which case we have  $|d_r| \leq b_r$  for all  $r \in I_1^m$ .

Consider the space  $\mathcal{G}$  of all good permutations. Let  $\mathbf{V}_0^{m+u}$  be a martingale obtained by Doob's process operated on  $f$  and  $\mathcal{G}$ . Then, using (3.32) we bound the martingale difference sequence  $\mathbf{d}'_1^{m+u}$  of  $\mathbf{V}_0^{m+u}$  as follows.

$$\begin{aligned} |d'_r| &\leq \mathbf{P}_{j \sim I_r^{m+u}} \{j \in I_{m+1}^{m+u}\} \times \\ &\quad \mathbf{E}_{\mathbf{Z} \in \mathcal{G}, j \sim I_{m+1}^{m+u}} \{|f(\mathbf{Z}^{rj}) - f(\mathbf{Z})| \mid \mathbf{Z}_1^r = \boldsymbol{\pi}_1^r, \boldsymbol{\pi}_1^r \text{ is admissible}\} \quad (3.34) \\ &\leq \mathbf{P}_{j \sim I_r^{m+u}} \{j \in I_{m+1}^{m+u}\} \times \\ &\quad \frac{\mathbf{E}_{\mathbf{Z}, j \sim I_{m+1}^{m+u}} \{|f(\mathbf{Z}^{rj}) - f(\mathbf{Z})| \mid \mathbf{Z}_1^r = \boldsymbol{\pi}_1^r, \boldsymbol{\pi}_1^r \text{ is admissible}\}}{\mathbf{P}_{\mathbf{Z}} \{\mathbf{Z} \in \mathcal{G}\}} \\ &\leq \frac{u((1-\theta_r)\beta_1 + \theta_r\beta)}{(m+u-r+1)(1-\Psi)} \triangleq b_r . \quad (3.35) \end{aligned}$$

Similarly to what we had showed previously, since  $f(\mathbf{Z})$  is  $(m, u)$ -permutation symmetric, for any  $r \in I_{m+1}^{m+u}$ ,  $d'_r = 0$ . Therefore, we can apply Azuma's inequality (Lemma 2) to the martingale  $\mathbf{V}_0^{m+u}$ . We obtain a bound on the deviation of  $V_{m+u} - V_0 = f(\mathbf{Z}) - \mathbf{E}_{\mathbf{Z}} \{f(\mathbf{Z})\}$ . Our result (3.12) is completed by equating the resulting bound to  $\delta$  and isolating  $\epsilon$ .

# Chapter 4

## Transductive Stability

### 4.1 Introduction

In this chapter we present novel transductive error bounds that are based on new notions of *transductive stability*. The *uniform stability* of a transductive algorithm is its worst case sensitivity for an exchange of two points, one from the labeled training set and one from the test set. Our uniform stability result is a rather straightforward adaptation of the results of Bousquet and Elisseeff (2002) for inductive learning. Unfortunately, this new bound is of limited practical merit because the required stability rates, which enable a non-vacuous bound, are not met by many transductive algorithms.

We, therefore, follow the approach taken by Kutin and Niyogi (2002) in induction and define a notion of *weak transductive stability* that requires overall stability ‘almost everywhere’ but still allows the algorithm to be sensitive to some fraction of the possible input exchanges. To utilize this weak transductive stability we develop a novel concentration inequality for symmetric functions of permutations based on Azuma’s martingale bound. We show that for sufficiently stable algorithms, their empirical error is concentrated near their transductive error and the slack term is a function of their weak stability parameters. The resulting error bound is potentially applicable to any transductive algorithm.

To apply our transductive error bound to a specific algorithm, it is necessary to know a bound on the weak stability of the algorithm. To this end, we develop a data-dependent estimation technique based on sampling that provides high probability estimates of the algorithm’s weak stability parameters. We apply this routine on the algorithm of Zhou et al. (2004).

### 4.2 Related Work

Exponential concentration bounds in terms of *uniform stability* were first considered by Bousquet and Elisseeff (2002) in the context of induction. Quite a few

variations of the inductive stability concept were defined and studied in (Bousquet & Elisseeff, 2002; Kutin & Niyogi, 2002; Mukherjee et al., 2004). It is not clear, however, what is the precise relation between these definitions and the associated error bounds. It is noted in (?; Mukherjee et al., 2004) that many important learning algorithms (e.g., SVM) are not stable under any of the stability definitions, including the significantly relaxed notion of weak stability introduced by Kutin (2002) and Kutin and Niyogi (2002). ? (?) attempted to remedy this by considering ‘graphical algorithms’ and a new geometrical stability definition, which captures a modified SVM (see also Bousquet & Elisseeff, 2002).

Stability was first considered in the context of transductive learning by Belkin et al. (2004). There the authors applied uniform inductive stability notions and results of Bousquet and Elisseeff (2002) to a specific graph-based transductive learning algorithm.<sup>1</sup>

We present general bounds for transduction based on particularly designed definitions of transductive stability, which we believe are better suited for capturing practical algorithms. Our weak stability bounds have relatively “standard” form of empirical error plus a slack term (unlike most weak stability bounds for induction (Kutin & Niyogi, 2002; Mukherjee et al., 2004; Rakhlin et al., 2005)). Kearns and Ron (1999) were the first to develop standard risk bounds based on weak stability. Their bounds are “polynomial”, depending on  $1/\delta$ , unlike the “exponential” bounds we develop here (depending on  $\ln 1/\delta$ ).

### 4.3 Definitions

We consider the following transductive setting (Vapnik, 1982). A *full sample*  $X_{m+u} = \{x_i\}_{i=1}^{m+u}$  consisting of  $m + u$  unlabeled examples in some space  $\mathcal{X}$  is given. For each point  $x_j \in X_{m+u}$ , let  $y_j \in \{\pm 1\}$  be its unknown deterministic label. A *training set*  $S_m$  consisting of  $m$  labeled points is generated as follows. Sample a subset of  $m$  points  $X_m \subset X_{m+u}$  uniformly at random from all  $m$ -subsets of the full sample. For each point  $x_i \in X_m$ , obtain its uniquely determined label  $y_i$  from the teacher. Then,  $S_m = (X_m, Y_m) = (z_i = \langle x_i, y_i \rangle)_{i=1}^m$ . The set of remaining  $u$  (unlabeled) points  $X_u = X_{m+u} \setminus X_m$  is called the *test set*. We use the notation  $I_r^s$  for the set of (indices)  $\{r, \dots, s\}$  (for integers  $r < s$ ). For simplicity we abuse notation, and unless otherwise stated, the indices  $I_1^m$  are reserved for training set points and the indices  $I_{m+1}^{m+u}$  for test set points.

The goal of the transductive learning algorithm  $\mathcal{A}$  is to utilize both the labeled training points  $S_m$  and the unlabeled test points  $X_u$  and generate a *soft classification*  $\mathcal{A}_{S_m, X_u}(x_i) \in [-1, 1]$  for each (test) point  $x_i$  so as to minimize its

---

<sup>1</sup>There is still some disagreement between authors about the definitions of ‘semi-supervised’ and ‘transductive’ learning. Belkin et al. (2004) study a transductive setting (according to the terminology presented here) but call it ‘semi-supervised’.

transductive error with respect to some loss function  $\ell$ ,

$$R_u(\mathcal{A}) \triangleq R_u(\mathcal{A}_{S_m, X_u}) \triangleq \frac{1}{u} \sum_{i=m+1}^{m+u} \ell(\mathcal{A}_{S_m, X_u}(x_i), y_i) .$$

The *empirical error* of  $\mathcal{A}$  is  $\widehat{R}_m(\mathcal{A}) \triangleq \widehat{R}_m(\mathcal{A}_{S_m, X_u}) \triangleq \frac{1}{m} \sum_{i=1}^m \ell(\mathcal{A}_{S_m, X_u}(x_i), y_i)$ . We consider the standard 0/1-loss and margin-loss functions denoted by  $\ell$  and  $\ell_\gamma$ , respectively.<sup>2</sup> In applications of the 0/1 loss function we always apply the sign function on the soft classification  $\mathcal{A}_{S_m, X_u}(x)$ . When using the margin loss function we denote the training and transductive errors of  $\mathcal{A}$  by  $\widehat{R}_m^\gamma(\mathcal{A})$  and  $R_u^\gamma(\mathcal{A})$ , respectively.

Note that in this transductive setting there is no underlying distribution as in (semi-supervised) inductive models.<sup>3</sup> Also, training examples are *dependent* due to the sampling without replacement of the training set from the full sample.

## 4.4 Uniform Stability Bound

Given a training set  $S_m$  and a test set  $X_u$  and two indices  $i \in I_1^m$  and  $j \in I_{m+1}^{m+u}$ , let  $S_m^{ij} \triangleq S_m \setminus \{z_i\} \cup \{z_j = \langle x_j, y_j \rangle\}$  and  $X_u^{ij} \triangleq X_u \setminus \{x_j\} \cup \{x_i\}$  (e.g.,  $S_m^{ij}$  is  $S_m$  with the  $i$ th example (from the training set) and  $j$ th example (from the test set) exchanged). The following definition of stability is a straightforward adaptation of the uniform stability definition from (Bousquet & Elisseeff, 2002) to our transductive setting.

**Definition 6 (Uniform Transductive Stability)** *A transductive learning algorithm  $\mathcal{A}$  has uniform transductive stability  $\beta$  if for all choices of  $S_m \subset S_{m+u}$ , for all  $i \in I_1^m$ ,  $j \in I_{m+1}^{m+u}$ ,*

$$\max_{1 \leq k \leq m+u} \left| \mathcal{A}_{S_m, X_u}(x_k) - \mathcal{A}_{S_m^{ij}, X_u^{ij}}(x_k) \right| \leq \beta . \quad (4.1)$$

Let  $\Delta(i, j, s, t) \triangleq \ell_\gamma(\mathcal{A}_{S_m^{ij}, X_u^{ij}}(x_t), y_t) - \ell_\gamma(\mathcal{A}_{S_m, X_u}(x_s), y_s)$ . For the proof of the forthcoming error bound we need the following technical lemma.

**Lemma 7**  $\mathbf{E}_{(S_m, X_u)} \left\{ R_u^\gamma(\mathcal{A}) - \widehat{R}_m^\gamma(\mathcal{A}) \right\} = \mathbf{E}_{(S_m, X_u), i \sim I_1^m, j \sim I_{m+1}^{m+u}} \left\{ \Delta(i, j, i, i) \right\}$ .

<sup>2</sup>For a positive real  $\gamma$ ,  $\ell_\gamma(y_1, y_2) = 0$  if  $y_1 y_2 \geq \gamma$  and  $\ell_\gamma(y_1, y_2) = \min\{1, 1 - y_1 y_2 / \gamma\}$  otherwise.

<sup>3</sup>As discussed earlier, Vapnik also considers a second transductive setting where examples are drawn from some unknown distribution; see Chapter 8 in (Vapnik, 1998). Results in the model we study here apply to the other model (Theorem 8.1 in (Vapnik, 1998)).

**Proof:** Using the linearity of expectation we obtain

$$\mathbf{E}_{S_m} \left\{ \hat{R}_m^\gamma(\mathcal{A}) \right\} = \frac{1}{m} \sum_{i=1}^m \mathbf{E}_{S_m} \left\{ \ell_\gamma(\mathcal{A}_{S_m, X_u}(x_i), y_i) \right\} \quad (4.2)$$

$$= \frac{1}{m} \sum_{i=1}^m \mathbf{E}_{S_m, k \sim I_1^{m+u}} \left\{ \ell_\gamma(\mathcal{A}_{S_m^{ik}, X^{ik}}(x_k), y_k) \right\} . \quad (4.3)$$

The last equality holds since both expectations in (4.2) and (4.3) are the average loss of the algorithm  $\mathcal{A}$  on the  $i$ -th example and the average is taken over all possible permutations. Since  $\mathcal{A}$  is symmetric on the training set, the expectation in (4.3) is the same for all  $i$ . Therefore, for all  $i \in I_1^m$ ,

$$\mathbf{E}_{(S_m, X_u)} \left\{ \hat{R}_m^\gamma(\mathcal{A}_{S_m, X_u}) \right\} = \mathbf{E}_{(S_m, X_u), k \sim I_1^{m+u}} \left\{ \ell_\gamma(\mathcal{A}_{S_m^{ik}, X^{ik}}(x_k), y_k) \right\} . \quad (4.4)$$

Likewise for all  $j \in I_{m+1}^{m+u}$ :

$$\mathbf{E}_{(S_m, X_u)} \left\{ R_u^\gamma(\mathcal{A}_{S_m, X_u}) \right\} = \mathbf{E}_{(S_m, X_u), k \sim I_1^{m+u}} \left\{ \ell_\gamma(\mathcal{A}_{S_m^{kj}, X^{kj}}(x_k), y_k) \right\} . \quad (4.5)$$

We abbreviate

$$R_{\text{diff}} = R_u^\gamma(\mathcal{A}) - \hat{R}_m^\gamma(\mathcal{A}) . \quad (4.6)$$

For any  $i \in I_1^m$ ,  $j \in I_{m+1}^{m+u}$ , it follows from (4.4) and (4.5) that

$$\begin{aligned} \mathbf{E}_{(S_m, X_u)} \left\{ R_{\text{diff}} \right\} &= \\ &= \mathbf{E}_{(S_m, X_u), k \sim I_1^{m+u}} \left\{ \ell_\gamma(\mathcal{A}_{S_m^{kj}, X^{kj}}(x_k), y_k) - \ell_\gamma(\mathcal{A}_{S_m^{ik}, X^{ik}}(x_k), y_k) \right\} \\ &= \mathbf{E}_{(S_m, X_u), k \sim I_1^{m+u}} \left\{ \ell_\gamma(\mathcal{A}_{S_m^{kj}, X^{kj}}(x_k), y_k) - \ell_\gamma(\mathcal{A}_{S_m, X_u}(x_k), y_k) \right\} \\ &\quad + \mathbf{E}_{(S_m, X_u), k \sim I_1^{m+u}} \left\{ \ell_\gamma(\mathcal{A}_{S_m, X_u}(x_k), y_k) - \ell_\gamma(\mathcal{A}_{S_m^{ik}, X^{ik}}(x_k), y_k) \right\} . \end{aligned}$$

Therefore, since  $\mathcal{A}$  is symmetric on  $X_m$  and  $X_u$ ,

$$\begin{aligned} \mathbf{E}_{(S_m, X_u)} \left\{ R_{\text{diff}} \right\} &= \\ &= \mathbf{E}_{(S_m, X_u), j \sim I_{m+1}^{m+u}, k \sim I_1^{m+u}} \left\{ \ell_\gamma(\mathcal{A}_{S_m^{kj}, X^{kj}}(x_k), y_k) - \ell_\gamma(\mathcal{A}_{S_m, X_u}(x_k), y_k) \right\} \\ &\quad + \mathbf{E}_{(S_m, X_u), i \sim I_1^m, k \sim I_1^{m+u}} \left\{ \ell_\gamma(\mathcal{A}_{S_m, X_u}(x_k), y_k) - \ell_\gamma(\mathcal{A}_{S_m^{ik}, X^{ik}}(x_k), y_k) \right\} \\ &= \frac{m}{m+u} \mathbf{E}_{(S_m, X_u), i \sim I_1^m, j \sim I_{m+1}^{m+u}} \left\{ \ell_\gamma(\mathcal{A}_{S_m^{ij}, X^{ij}}(x_i), y_i) - \ell_\gamma(\mathcal{A}_{S_m, X_u}(x_i), y_i) \right\} \\ &\quad + \frac{u}{m+u} \mathbf{E}_{(S_m, X_u), i \sim I_1^m, j \sim I_{m+1}^{m+u}} \left\{ \ell_\gamma(\mathcal{A}_{S_m, X_u}(x_j), y_j) - \ell_\gamma(\mathcal{A}_{S_m^{ij}, X^{ij}}(x_j), y_j) \right\} \\ &= \mathbf{E}_{(S_m, X_u), i \sim I_1^m, j \sim I_{m+1}^{m+u}} \left\{ \Delta(i, j, i, i) \right\} . \end{aligned}$$

□

Our first transductive error bound is obtained by applying Theorem 1 to the function  $R_u^\gamma(\mathcal{A}) - \hat{R}_m^\gamma(\mathcal{A})$  and bounding  $\mathbf{E}\{R_u^\gamma(\mathcal{A}) - \hat{R}_m^\gamma(\mathcal{A})\}$  using an adaptation of Lemma 7 from (Bousquet & Elisseeff, 2002) to our setting.

**Theorem 3** *Let  $\mathcal{A}$  be a transductive learning algorithm with transductive uniform stability  $\beta$ . Let  $\tilde{\beta} \triangleq \frac{(u-1)\beta}{u\gamma} + \frac{(m-1)\beta}{m\gamma} + \frac{1}{m} + \frac{1}{u}$  and  $K \triangleq \frac{mu}{2(m+u)-1} \left(1 - \frac{1}{2\max(m,u)}\right)^{-1}$ . Then, for all  $\gamma > 0$  and  $\delta \in (0, 1)$ , with probability at least  $1 - \delta$  over the draw of the training/test sets  $(S_m, X_u)$ ,*

$$R_u(\mathcal{A}) \leq \hat{R}_m^\gamma(\mathcal{A}) + \beta/\gamma + \tilde{\beta}\sqrt{K(m, u) \ln(1/\delta)} . \quad (4.7)$$

**Proof:** We derive a bound on the strong permutation stability of the function  $f(S_m, X_u) \triangleq R_u^\gamma(\mathcal{A}) - \hat{R}_m^\gamma(\mathcal{A})$  and its expected value. Then we apply Theorem 1. Abbreviate  $\mathcal{A}^{ij} \triangleq \mathcal{A}_{S_m^{ij}, X_u^{ij}}$ . For  $i \in I_1^m, j \in I_{m+1}^{m+u}$ , we have (by expanding the risk expressions),

$$\begin{aligned} & \left| R_u^\gamma(\mathcal{A}) - \hat{R}_m^\gamma(\mathcal{A}) - \left( R_u^\gamma(\mathcal{A}^{ij}) - \hat{R}_m^\gamma(\mathcal{A}^{ij}) \right) \right| \leq \\ & \frac{1}{u} \sum_{\substack{k=m+1, \\ k \neq j}}^{m+u} |\Delta(i, j, k, k)| + \frac{1}{u} |\Delta(i, j, i, j)| + \frac{1}{m} \sum_{\substack{k=1, \\ k \neq i}}^m |\Delta(i, j, k, k)| + \frac{1}{m} |\Delta(i, j, j, i)| . \end{aligned} \quad (4.8)$$

Since  $\ell_\gamma$  has Lipschitz constant  $\gamma$ , it follows from (4.1) that

$$\max_{1 \leq k \leq m+u} |\Delta(i, j, k, k)| \leq \frac{\beta}{\gamma} . \quad (4.9)$$

Hence (4.8) is bounded by  $\tilde{\beta}$ . Therefore the function  $f(S_m, X_u)$  has transductive classification stability  $\tilde{\beta}$ . By applying Theorem 1 to  $f(S_m, X_u)$ , equating the resulting bound to  $\delta$  and isolating  $\epsilon$  we obtain that with probability at least  $1 - \delta$ ,

$$R_u^\gamma(\mathcal{A}) \leq \hat{R}_m^\gamma(\mathcal{A}) + \mathbf{E}_{(S_m, X_u)} \left\{ R_u^\gamma(\mathcal{A}) - \hat{R}_m^\gamma(\mathcal{A}) \right\} + \tilde{\beta}\sqrt{K(m, u) \ln \frac{1}{\delta}} . \quad (4.10)$$

It follows from (4.9) that the right hand side of the equality in Lemma 7 is bounded by  $\beta/\gamma$ . By substituting this bound to (4.10) and using the inequality  $R_u(\mathcal{A}) \leq R_u^\gamma(\mathcal{A})$ , we obtain (4.7).  $\square$

The tightness of the bound (4.7) depends on the transductive uniform stability  $\beta$  of algorithm  $\mathcal{A}$ . If  $\beta = O(1/m)$  and  $u = \Omega(m)$ , then the slack terms in (4.7) amount to  $O(\sqrt{\ln(1/\delta)/m}/\gamma)$ . However, in our experience this stability rate is never met by useful transductive algorithms.

## 4.5 Weak Stability Bound

The impractical requirement of the uniform stability concept motivates a weaker notion of stability that we develop here. In this section we derive an error bound for transductive algorithms by utilizing the weak stability notion. To this end, we now define weak transductive stability for algorithms. The following definition, which contains three conditions and six parameters, is somewhat cumbersome but we believe it facilitates tighter bounds than can possibly be achieved using a simpler definition (that only includes condition (4.12) below); see also the discussion that follows this definition. For a fixed full sample, we abbreviate  $\mathcal{A}^{ij}(x, (S_m, X_u)) \triangleq |\mathcal{A}_{S_m, X_u}(x) - \mathcal{A}_{S_m^i, X_u^j}(x)|$ .

**Definition 7 (Weak Transductive Stability)** *A transductive learning algorithm  $\mathcal{A}$  has weak transductive stability  $(\beta, \beta_1, \beta_2, \delta_1^a, \delta_1^b, \delta_2)$  if it has uniform transductive stability  $\beta$  and the following conditions (4.11) and (4.12) hold.*

$$\mathbf{P}_{(S_m, X_u), i \sim I_1^m, j \sim I_{m+1}^{m+u}} \left\{ \mathbf{P}_{x \sim X_{m+u}} \left\{ \mathcal{A}^{ij}(x, (S_m, X_u)) \leq \beta_1 \right\} \geq 1 - \delta_1^a \right\} \geq 1 - \delta_1^b . \quad (4.11)$$

$$\mathbf{P}_{(S_m, X_u), i \sim I_1^m, j \sim I_{m+1}^{m+u}} \left\{ \mathcal{A}^{ij}(x_i, (S_m, X_u)) \leq \beta_2 \right\} \geq 1 - \delta_2 . \quad (4.12)$$

While in (4.11) we quantify the sensitivity of the algorithm w.r.t. all examples in  $X_{m+u}$ , in (4.12) only the exchanged examples are considered. A number of weak stability definitions for induction is given in (Kearns & Ron, 1999; Kutin & Niyogi, 2002; Mukherjee et al., 2004). Ignoring the differences between induction and transduction, our condition (4.11) poses a qualitatively weaker constraint than the ‘weak hypothesis stability’ (Definition 3.5 in (Kutin & Niyogi, 2002)), and a stronger constraint than the ‘weak error stability’ (Definition 3.8 in (Kutin & Niyogi, 2002)). Our condition (4.12) is a straightforward adaptation of the ‘cross-validation stability’ (Definition 3.12 in (Kutin & Niyogi, 2002)) to our transductive setting.

It should be possible to show, using a technique similar to the one used in the proof of Theorem 3.16 in (Kutin & Niyogi, 2002), that (4.12) implies (4.11). In this case a simpler weak stability definition may suffice but, using our techniques, the resulting error bound would be looser.

**Theorem 4** *Let  $\mathcal{A}$  be an algorithm with weak transductive classification stability  $(\beta, \beta_1, \beta_2, \delta_1^a, \delta_1^b, \delta_2)$ . Suppose that  $u \geq m$  and  $\delta_1^a < \frac{m}{m+u}$ .<sup>4</sup> Let  $\gamma > 0$ ,  $\delta \in (0, 1)$  be given and set*

$$\tilde{\beta}_1 \triangleq \frac{u-1}{u} \cdot \frac{\beta_1}{\gamma} + \frac{\delta_1^a(m+u)\beta + [m-1-\delta_1^a(m+u)]\beta_1}{m\gamma} + \frac{1}{m} + \frac{1}{u} , \quad (4.13)$$

<sup>4</sup>The proof for the cases  $\delta_1^a > \frac{m}{m+u}$  and  $m > u$  is very similar to the proof given below and is omitted.

$$\tilde{\beta} \triangleq \frac{u-1}{u} \cdot \frac{\beta}{\gamma} + \frac{m-1}{m} \cdot \frac{\beta}{\gamma} + \frac{1}{m} + \frac{1}{u} . \quad (4.14)$$

For any  $\theta_i \in (0, 1)$ ,  $i \in I_1^m$ , set  $\Psi \triangleq \sum_{i=1}^m \frac{\delta_1^b}{\theta_i}$  and  $b_i \triangleq \frac{u((1-\theta_i)\tilde{\beta}_1 + \theta_i\tilde{\beta})}{(m+u-i+1)(1-\Psi)}$ . If  $\Psi < 1$ , then with probability at least  $(1-\delta) \cdot (1-\Psi)$  over the draw of the training/test sets  $(S_m, X_u)$ ,

$$R_u(\mathcal{A}) \leq \hat{R}_m^\gamma(\mathcal{A}) + \left[ (1-\delta_2) \frac{\beta_2}{\gamma} + \delta_2 \frac{\beta}{\gamma} \right] + \sqrt{2 \sum_{i=1}^m b_i^2 \ln \frac{1}{\delta}} . \quad (4.15)$$

**Proof:** We derive bounds on the weak permutation stability of the function  $f(S_m, X_u) \triangleq R_u^\gamma(\mathcal{A}) - \hat{R}_m^\gamma(\mathcal{A})$  and its expected value. Then we apply Lemma 2. As in the proof of Theorem 3 we have (by expanding the risk expressions) that for  $i \in I_1^m$ ,  $j \in I_{m+1}^{m+u}$ ,

$$\begin{aligned} & \left| R_u^\gamma(\mathcal{A}) - \hat{R}_m^\gamma(\mathcal{A}) - \left( R_u^\gamma(\mathcal{A}_{S_m^{ij}, X_u^{ij}}) - \hat{R}_m^\gamma(\mathcal{A}_{S_m^{ij}, X_u^{ij}}) \right) \right| \leq \\ & \frac{1}{u} \sum_{\substack{k=m+1, \\ k \neq j}}^{m+u} |\Delta(i, j, k, k)| + \frac{1}{u} |\Delta(i, j, j, i)| + \frac{1}{m} \sum_{\substack{k=1, \\ k \neq i}}^m |\Delta(i, j, k, k)| + \frac{1}{m} |\Delta(i, j, i, j)| . \end{aligned} \quad (4.16)$$

Since  $\ell_\gamma$  has Lipschitz constant  $\gamma$ , it follows from (4.11) that

$$\mathbf{P}_{(S_m, X_u), i \sim I_1^m, j \sim I_{m+1}^{m+u}} \left\{ \mathbf{P}_{k \sim I_1^{m+u}} \{ |\Delta(i, j, k, k)| \leq \beta_1/\gamma \} \geq 1 - \delta_1^a \right\} \geq 1 - \delta_1^b . \quad (4.17)$$

We say that the example  $x_k$  is *bad* if  $|\Delta(i, j, k, k)| > \beta_1/\gamma$ . According to (4.17), with probability at least  $1 - \delta_1^b$  over the choices of  $((S_m, X_u), i, j)$ , there are at most  $(1 - \delta_1^a)(m+u)$  bad examples. If  $u \geq m$ , the terms in the second summation in (4.16) have greater weight (which is  $1/m$ ) than the terms in the first summation (weighted by  $1/u$ ). In the worst case all bad examples appear in the second summation in which case (4.16) is bounded by (4.13) with probability at least  $1 - \delta_1^b$  over the choices of  $((S_m, X_u), i, j)$ .

The right hand side of (4.16) is always bounded by  $\tilde{\beta}$ . Therefore, the function  $f(S_m, X_u)$  has weak permutation stability  $(\tilde{\beta}, \tilde{\beta}_1, \delta_1^b)$ . By applying Lemma 2 to  $f(S_m, X_u)$ , we obtain that with probability at least  $(1-\delta)(1-\Psi)$ ,

$$R_u^\gamma(\mathcal{A}) \leq \hat{R}_m^\gamma(\mathcal{A}) + \mathbf{E}_{(S_m, X_u)} \left\{ R_u^\gamma(\mathcal{A}) - \hat{R}_m^\gamma(\mathcal{A}) \right\} + \sqrt{2 \sum_{i=1}^m b_i^2 \ln \frac{1}{\delta}} . \quad (4.18)$$

Since  $\ell_\gamma$  has Lipschitz constant  $\gamma$ , it follows from (4.12) that

$$\mathbf{P}_{(S_m, X_u), i \sim I_1^m, j \sim I_{m+1}^{m+u}} \{ |\Delta(i, j, i, i)| \leq \beta_2/\gamma \} \geq 1 - \delta_2 . \quad (4.19)$$

Therefore, the right hand side of the equality in Lemma 7 is bounded from above by  $\beta_2(1 - \delta_2)/\gamma + \beta\delta_2/\gamma$ . By substituting this bound to (4.18) and using the inequality  $R_u^\gamma(\mathcal{A}) \geq R_u(\mathcal{A})$ , we obtain (4.15).  $\square$

It follows from Definition 7 that  $\beta_1$  depends on  $\delta_1^a$  and  $\delta_1^b$ , and that  $\beta_2$  depends on  $\delta_2$ . Hence the bound (4.15) depends on the parameters  $\delta_1^a, \delta_1^b, \delta_2, \theta_i, i \in I_1^m$ . It is possible to show that if  $u = \Omega(m)$ ,  $\delta_1^a = O(1/m + u)$ ,  $\delta_1^b = O(1/m^2)$  and  $\beta_1, \beta_2, \delta_2, \theta_i$  are each  $O(1/m)$ , then the slack term in (4.15) is  $O(\sqrt{\ln(1/\delta)/m/\gamma})$  and the bound's confidence can be made arbitrarily close to 1.

## 4.6 High Confidence Stability Estimation

In this section we describe a routine that can generate useful upper bounds on the weak stability parameters (Definition 7) of transductive algorithms. The routine generates these estimates with arbitrarily high probability and is based on a sampling-based quantile estimation technique. Given a particular learning algorithm, our stability estimation routine relies on an ‘‘oracle’’ that bounds the sensitivity of the transductive algorithm with respect to a small change in the input. We present such an oracle for a familiar practical algorithm. In Sec. 4.6.1 we describe the quantile estimation method, which is similar to the one presented in (Manku, Rajagopalan, & Lindsay, 1998); in Sec. 4.6.2 we present the bounding algorithm, and in Sec. 4.6.3 we consider a known transductive algorithm and present a few numerical examples of the application of these methods.

### 4.6.1 Quantile Estimation

Consider a very large set  $\Omega$  of  $N$  numbers. Define the  $q$ -quantile of  $\Omega$  to be the  $\lceil qN \rceil$ -th smallest element of  $\Omega$  (i.e., it is the  $\lceil qN \rceil$ -th element in an increasing order sorted list of all elements in  $\Omega$ ). Our goal is to bound the  $q$ -quantile  $x_q$  from above as tightly as possible, with high confidence, by sampling a ‘‘small’’ number  $k \ll N$  of elements. For any  $\epsilon \in (0, 1)$  we generate a bound  $\beta$  such that  $\mathbf{P}\{x_q \leq \beta\} \geq 1 - \epsilon$ . The idea is to sample  $k = k(q, \epsilon)$  elements from  $\Omega$  uniformly at random, compute their exact  $(\bar{q} \triangleq q + \frac{1-q}{2})$ -quantile  $x_{\bar{q}}$ , and output  $\beta \triangleq x_{\bar{q}}$ . Denote by  $\mathbf{quantile}(q, \epsilon, \Omega)$  the resulting routine whose output is  $\beta = x_{\bar{q}}$ .

**Lemma 8** *For any  $q, \epsilon \in (0, 1)$ . If  $k = k(q, \epsilon) = \frac{2 \ln(1/\epsilon)}{(1-q)^2}$ , then*

$$\mathbf{P}\{x_q \leq \mathbf{quantile}(q, \epsilon, \Omega)\} \geq 1 - \epsilon . \quad (4.20)$$

**Proof:** For  $i \in I_1^k$  let  $X_i$  be the indicator random variable obtaining 1 if the  $i$ th drawn element (from  $\Omega$ ) is smaller than  $x_q$ , and 0 otherwise. Set  $Q = \frac{1}{k} \sum_{i=1}^k X_i$ .

Clearly,  $\mathbf{E}Q \leq q$ . By Hoeffding's inequality and using the definition of  $\bar{q}$ , we get

$$\begin{aligned} \mathbf{P}\{Q > \bar{q}\} &= \mathbf{P}\left\{Q - q > \frac{1 - q}{2}\right\} \\ &\leq \mathbf{P}\left\{Q - \mathbf{E}Q > \frac{1 - q}{2}\right\} \leq \exp\left(-\frac{k(1 - q)^2}{2}\right). \end{aligned} \quad (4.21)$$

Therefore, with ‘‘high probability’’ the number  $kQ$  of sample points that are smaller than  $x_q$  is smaller than  $k\bar{q}$ . Hence, at least  $(1 - \bar{q})k$  points in the sample are larger than  $x_q$ . `quantile` returns the smallest of them. Equating the right hand side of (4.21) to  $\epsilon$  and solving for  $k$  yields the stated sample size.  $\square$

## 4.6.2 Stability Estimation Algorithm

Let  $\mathcal{A}$  be a transductive learning algorithm. We assume that some (rough) bound on  $\mathcal{A}$ 's uniform stability  $\beta$  is known. If no tight bound is known, we take the maximal default value, which is 2, as can be seen in Definition 6. Our goal is to find useful bounds for the weak stability parameters of Definition 7. Let the values of  $\delta_1^a$ ,  $\delta_1^b$  and  $\delta_2$  be given. We aim at finding upper bounds on  $\beta_1$  and  $\beta_2$ .

**Definition 8 (The diff Oracle)** Consider a fixed labeled training set  $S_m = (X_m, Y_m)$  given to the learning algorithm. Let  $\mathbf{diff}(\tilde{X}_m, \tilde{X}_u, i, j, r | S_m)$  be an ‘‘oracle’’ function defined for any possible partition  $(\tilde{X}_m, \tilde{X}_u)$  of the full sample and indices  $i \in I_1^m$ ,  $j \in I_{m+1}^{m+u}$  and  $r \in I_1^{m+u}$ . `diff` provides an upper bound on

$$\left| \mathcal{A}_{\tilde{S}_m, \tilde{X}_u}(x_r) - \mathcal{A}_{\tilde{S}_m, \tilde{X}_u}^{ij}(x_r) \right|, \quad (4.22)$$

where  $\tilde{S}_m$  is any possible labeling of  $\tilde{X}_m$  that ‘‘agrees’’ with  $S_m$  on points in  $X_m \cap \tilde{X}_m$ . Note that here we assume that  $I_1^m$  is the set indices of points in  $\tilde{X}_m$  (and indices in  $X_m$  are not specified and can be arbitrary indices in  $I_1^{m+u}$ ).

We assume that we have an access to a useful  $\mathbf{diff}(\tilde{X}_m, \tilde{X}_u, i, j, r | S_m)$  function that provides a tight upper bound on (4.22). We now describe our stability estimation algorithm that applies `diff`.

Let  $K$  be the set of all possible quadruples  $(\tilde{X}_m, \tilde{X}_u, i, j)$  as in Definition 8. Define  $\Omega_1 = \{\omega(t) : t \in K\}$ , where  $\omega(t) = \omega(\tilde{X}_m, \tilde{X}_u, i, j)$  is a  $(1 - \delta_1^a)$ -quantile of the set

$$\left\{ \mathbf{diff}(\tilde{X}_m, \tilde{X}_u, i, j, r | S_m), r = 1, \dots, m + u \right\}.$$

It is not hard to see that for any  $\epsilon \in (0, 1)$ , with probability at least  $1 - \epsilon$  (over random choices made by the `quantile` routine), `quantile`( $1 - \delta_1^b, \epsilon, \Omega_1$ ) is an upper bound on the weak stability parameter  $\beta_1$  of Definition 7. Likewise, let  $\Omega_2 = \{\omega(t) : t \in K\}$ , but now  $\omega(t) = \omega(\tilde{X}_m, \tilde{X}_u, i, j) = \mathbf{diff}(\tilde{X}_m, \tilde{X}_u, i, j, i)$ . It is

not hard to see that for any  $\epsilon$ , with probability at least  $1 - \epsilon$ ,  $\text{quantile}(1 - \delta_2, \epsilon, \Omega_2)$  is an upper bound on the weak stability parameter  $\beta_2$  of Definition 7.

Thus, our weak stability estimation algorithm simply applies `quantile` twice with appropriate parameters. To actually draw the samples, `quantile` utilizes the `diff` oracle. Let  $v$  be the time complexity of computing `diff` oracle. By Lemma 8 the number of samples that should be drawn, in order to obtain with probability at least  $1 - \epsilon$  the bound on  $q$ -quantile, is  $O(\ln(1/\epsilon)/(1 - q)^2)$ . It can be verified that the complexity of our stability estimation algorithm is  $O(\ln(1/\epsilon)(m + u)v / \min\{(\delta_1^b)^2, (\delta_2)^2\})$ . As discussed after Theorem 4,  $\delta_1^b$  should be  $O(1/m^2)$  to ensure that the bound (4.15) has arbitrarily high confidence. This constraint entails a time complexity of  $\Omega(m^4(m + u))$ . Therefore, at this stage our ability to use the stability estimation routine in conjunction with the transductive error bound is limited to very small values of  $m$ .

### 4.6.3 Stability Estimation Examples

In this section we consider the transductive learning algorithm of Zhou et al. (Zhou et al., 2004) and demonstrate a data-dependent estimation of its weak stability parameters using our method. While currently there is no comprehensive empirical comparison between all available transductive algorithms, this algorithm appears to be among the more promising ones (Huang & Kecman, 2005). We chose this algorithm, denoted by **CM** (stands for ‘Consistency Method’; see (Huang & Kecman, 2005)), because we could easily develop a useful `diff` “oracle” for it. We were also able to efficiently implement `diff` “oracle” for the algorithm of Zhu et al. (Zhu et al., 2003), which will be presented elsewhere.

We start with the brief description of the **CM** algorithm. Let  $W$  be a symmetric  $(m + u) \times (m + u)$  affinity matrix of the full sample  $X_{m+u}$ . We assume that  $W_{ii} = 0$ . In this paper we use RBF kernels, parameterized by  $\sigma$ , to construct  $W$ . Let  $D$  be a diagonal matrix, whose  $(i, i)$ -element is the sum of the  $i$ th row in  $W$ . A normalized Laplacian of  $W$  is  $L = D^{-1/2}WD^{-1/2}$ . Let  $\alpha$  be a parameter in  $(0, 1)$ . Let  $Y$  be an  $(m + u) \times 1$  vector of available full sample labels, where the entries corresponding to training examples are  $\pm 1$  and entries of unlabeled examples are 0. We assume w.l.o.g. that the first  $m$  entries in  $Y$  correspond to the  $m$  labeled training examples. Let  $P = (I - \alpha L)^{-1}$ . The **CM** algorithm produces soft-classification  $F = P \cdot Y$ . In other words, if  $p_{ij}$  is the  $(i, j)$ th entry of  $P$  and  $f_i$  is the  $i$ th entry of  $F$ , the point  $x_i$  receives the soft-classification

$$f_i = \sum_{j=1}^m p_{ij} y_j . \quad (4.23)$$

To obtain useful bounds on the (weak) stability of **CM** we require the following benign technical modifications of **CM** that would not change the *hard* classification it generates over test set examples.

1. We prevent over-fitting to the training set by setting  $p_{ii} = 0$ .
2. To enable a comparison between stability values corresponding to different settings of the parameters  $\alpha$  and  $\sigma$ , we ensure that the dynamic range of  $f_i$  is normalized w.r.t. different values of  $\alpha$  and  $\sigma$ . That is, instead of using (4.23) for prediction we use

$$f_i = \frac{\sum_{j=1}^m p_{ij} y_j}{\sum_{j=1}^m p_{ij}} . \quad (4.24)$$

The first modification prevents possible over-fitting to the training set since for any  $i \in I_1^{m+u}$ , in the original CM the value of  $p_{ii}$  is much larger than any of the other  $p_{ij}$ ,  $j \neq i$ , and therefore, the soft classification of the training example  $x_i$  is almost completely determined by its given label  $y_i$ . Hence by (4.23), when  $x_i$  is exchanged with some test set example  $x_j$ , the soft classification change of  $x_i$  will probably be large. Therefore, the stability condition (4.12) cannot be satisfied with small values of  $\beta_2$ . By setting  $p_{ii} = 0$  we prevent this problem and only affect the soft and hard classification of training examples (and keep the soft classifications of test points intact). The second modification clearly changes the dynamic range of all soft classifications but does not alter any hard classification.

To use our stability estimation algorithm one should provide an implementation of `diff`. We show that for the CM algorithm `diff`( $\tilde{X}_m, \tilde{X}_u, i, j, r | S_m$ ) can be effectively implemented as follows. For notational convenience we assume here (see also Definition 8 where we use this convention) that examples in  $\tilde{X}_m$  have indices in  $I_1^m$ . Let  $\tau(r) = \sum_{k=1, k \neq i}^m p_{rk}$  and  $\tau_y(r) = \sum_{k=1, k \neq i}^m p_{rk} y_k$ . It follows from (4.24) that

$$\begin{aligned} \left| \mathcal{A}_{\tilde{S}_m, \tilde{X}_u}(x_r) - \mathcal{A}_{\tilde{S}_m^{ij}, \tilde{X}_u^{ij}}(x_r) \right| &= \left| \frac{\tau_y(r) + p_{ri} y_i}{\tau(r) + p_{ri}} - \frac{\tau_y(r) + p_{rj} y_j}{\tau(r) + p_{rj}} \right| \\ &= \left| \frac{\tau_y(r) \cdot (p_{rj} - p_{ri}) + \tau(r) \cdot (p_{ri} y_i - p_{rj} y_j) + p_{ri} p_{rj} (y_i - y_j)}{(\tau(r) + p_{ri})(\tau(r) + p_{rj})} \right| \\ &= \left| \frac{(p_{rj} - p_{ri}) \cdot \sum_{k=1, k \neq i, x_k \in X_m}^m p_{rk} y_k + T}{(\tau(r) + p_{ri})(\tau(r) + p_{rj})} \right| , \end{aligned} \quad (4.25)$$

where  $T \triangleq (p_{rj} - p_{ri}) \cdot \sum_{k=1, k \neq i, x_k \in X_m}^m p_{rk} y_k + \tau(r) \cdot (p_{ri} y_i - p_{rj} y_j) + p_{ri} p_{rj} (y_i - y_j)$ .

To implement `diff`( $\tilde{X}_m, \tilde{X}_u, i, j, r | S_m$ ) we should upper bound (4.25). Suppose first that the values of  $y_i$  and  $y_j$  are known. Then,  $T$  is constant and the only unknowns in (4.25) are the  $y_k$ 's in the first summation. Observe that (4.25) is maximal when all values of these  $y_k$ 's are  $-1$  or all of them are  $+1$ . Hence by taking the maximum over these possibilities we obtain an upper bound on (4.25). If  $y_i$  (or  $y_j$ ) is unknown then, similarly, for each of its possible assignments we compute (4.25) and take the maximum. In the worst case, when both  $y_i$  and

$y_j$  are unknown, we compute the maximum of (4.25) over the eight possible assignments for these two variables and the  $y_k$ 's in the first summation. It can be verified that the time complexity of the above `diff` oracle is  $O(m)$ .

We now show two numerical examples of stability estimations for the `CM` algorithm with respect to two UCI datasets. These results were obtained by implementing the modified `CM` algorithm and the stability estimation routine applied with the above implementation of `diff`. For each “experiment” we ran the modified `CM` algorithm with 21 different hyper-parameter settings for  $\alpha$  and  $\sigma$ , each resulting in a different application of the algorithm.<sup>5</sup>

We considered two UCI datasets, `musk` and `mush`. From each dataset we generated 30 random full samples  $X_{m+u}$  each consisting of 400 points. We divided each full sample instance to equally sized training and test sets uniformly at random. The high confidence (95%) estimation of stability parameter  $\beta_1$  (see Definition 7) w.r.t.  $\delta_1^a = \delta_1^b = 0.1$ , and the corresponding empirical and true risks are shown in Fig. 4.1. The graphs for the  $\beta_2$  parameter are qualitatively similar and are omitted here. Indices in the  $x$ -axis correspond to the 21 applications of `CM` and are sorted in increasing order of true risk. Each stability and error value depicted is an average over the 30 random full samples. We also depict a high confidence (95%) true stability estimates, obtained *in hindsight* by using the unknown labels in the computation of `diff`. The uniform stability graphs correspond to *lower bounds* obtained by taking the maximal soft classification change encountered while estimating the true weak stability.

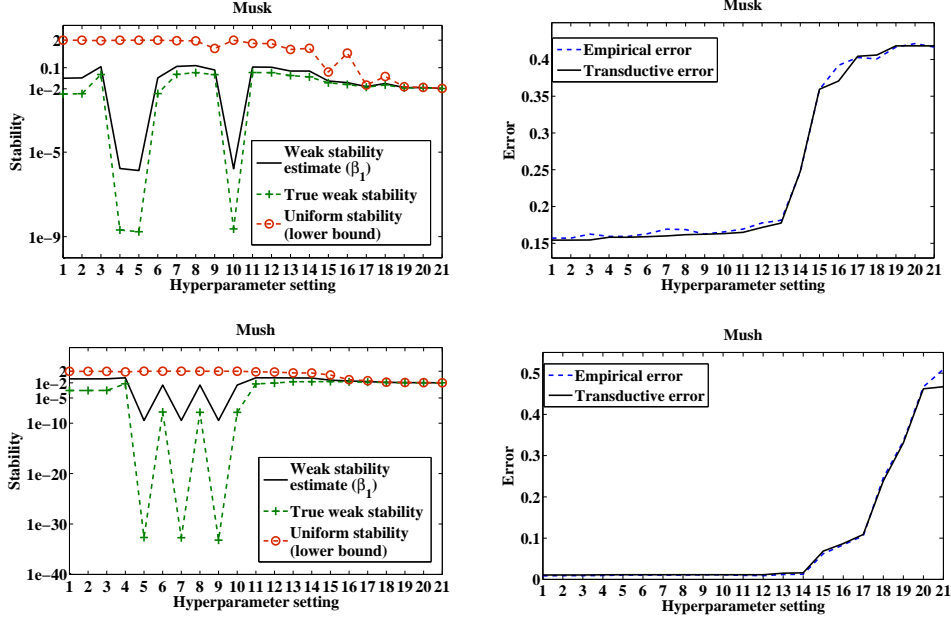
It is evident that the (true) weak stability is often significantly lower than the (lower bound on) the uniform stability. In cases where the weak and uniform stabilities are similar, the `CM` algorithm performs poorly. The estimated weak stability behaves qualitatively the same as the true weak stability. When the uniform stability obtains lower values the algorithm performs very poorly. This may indicate that a good uniform stability is correlated with degenerated behavior (similar phenomenon was observed in (Belkin et al., 2004)). In contrast, we see that very good weak stability can coincide with very high performance. Finally, we note that these graphs do not demonstrate that good weak stability is proportional to low discrepancy between the empirical and true errors.

## 4.7 Concluding Remarks

This paper has presented new error bounds for transductive learning. The bounds are based on novel definitions of uniform and weak transductive stability. We have also shown that weak transductive stability can be bounded with high confidence in a data-dependent manner and demonstrated the application of this estimation

---

<sup>5</sup>We naively took  $\alpha \in \{0.01, 0.5, 0.99\}$  and  $\sigma \in \{0.1, 0.2, 0.3, 0.4, 0.5, 1, 2\}$  and these were our first and only choices.



**Figure 4.1:** Stability estimates (left) and the corresponding empirical/true errors (right) for musk and mush datasets.

routine on a known transductive algorithm. As far as we know this is the first attempt to generate truly data-dependent high confidence stability estimates based on all available information including the labeled samples.

We note that similar risk bounds based on weak stability can be obtained for induction. However, the adaptation of Definition 7 to induction (see also inductive definitions of weak stability in (Kearns & Ron, 1999; Kutin & Niyogi, 2002; Mukherjee et al., 2004)) depends on the probability space of training sets, which is unknown in general. This prevents the estimation of weak stability using our method.

As discussed, to derive stability bounds with sufficient confidence our stability estimation routine is required to run in  $\Omega(m^4(m + u))$  time, which precluded, at this stage, an empirical evaluation of our bounds. In future work we will attempt to overcome this obstacle by tightening our bound, perhaps using the techniques from (Ledoux, 2001; Talagrand, 2005). A second direction would be to develop a more suitable weak stability definition. We also plan to consider other known transductive algorithms and develop for them a suitable implementation of the `diff` oracle.

# Chapter 5

## Transductive Rademacher Complexity and its Applications

### 5.1 Introduction

In this paper we consider transductive classification. So far, several general error bounds for transductive classification have been developed (see, e.g., Vapnik, 1982; Blum & Langford, 2003; Derbeko et al., 2004; El-Yaniv & Pechyony, 2006). We continue this fruitful line of research and develop a new technique for deriving data-dependent error bounds. Our technique consists of two components. The first component is a general error bound for transduction in terms of transductive Rademacher complexity. While this bound is syntactically similar to known inductive Rademacher bounds (see, e.g., Bartlett & Mendelson, 2002), it is different in the sense that the transductive Rademacher complexity is computed with respect to the hypothesis space that can be chosen *after* observing unlabeled training and test examples. This opportunity is unavailable in the inductive setting where the hypothesis space must be fixed *before* any example is observed.

The second component of our bounding technique is a generic method for bounding the Rademacher complexity of transductive algorithms using a special representation that we term *unlabeled-labeled representation (ULR)*. In this representation the soft classification vector generated by the algorithm is a product  $U\alpha$ , where  $U$  is a matrix that depends on the unlabeled data and  $\alpha$  is an unrestricted vector (i.e., may depend on all available information, including the labeled training set). Any transductive algorithm has infinitely many ULRs, including a trivial ULR, with  $U$  being an identity matrix. We show that many state-of-the-art algorithms have non-trivial ULR leading to non-trivial error bounds. Based on ULR representations we bound the Rademacher complexity of transductive algorithms in terms of the spectrum of the matrix  $U$  in their ULR. This bound motivates spectral transformations that are commonly done to improve the performance of transductive algorithms (e.g., see Chapelle et al., 2003; Joachims,

2003; Johnson & Zhang, 2008). We apply our method and derive error bounds for the “consistency method” of Zhou et al. (2004), the spectral graph transducer (SGT) algorithm of Joachims (2003) and the Tikhonov regularization algorithm of Belkin et al. (2004). The bounds obtained for these algorithms are explicit and can be easily computed.

We also show a simple Monte-Carlo scheme for bounding the Rademacher complexity of any transductive algorithm using its ULR. We demonstrate the efficacy of this scheme for the “consistency method” of Zhou et al. (2004). Our final contribution is a PAC-Bayesian bound for transductive mixture algorithms. This result motivates the use of ensemble methods in transduction that are yet to be explored in this setting.

The paper has the following structure. In Section 5.1.1 we survey the results that are closely related to our work. In Section 5.2 we define our learning model and the transductive Rademacher complexity. The inequality developed in Section 3.1 and the transductive Rademacher complexity are used in Section 5.3 to derive a uniform risk bound, which depends on the transductive Rademacher complexity. In Section 5.4 we introduce a generic method for bounding the Rademacher complexity of any transductive algorithm using its unlabeled-labeled representation. In Section 5.5 we apply this technique to obtain explicit risk bounds for several known transductive algorithms. Finally, in Section 5.6 we instantiate our risk bound to transductive mixture algorithms. We discuss directions for future research in Section 5.7. The technical proofs of our results are presented in Appendices 5.8.1-5.8.7.

### 5.1.1 Related Work

Vapnik (1982) presented the first general 0/1 loss bounds for transductive classification. His bounds are implicit in the sense that tail probabilities are specified in the bound as the outcome of a computational routine. Vapnik’s bounds can be refined to include prior “beliefs” as noted by Derbeko et al. (2004). Similar implicit but somewhat tighter bounds were developed by Blum and Langford (2003) for the 0/1 loss case. Explicit PAC-Bayesian transductive bounds for any bounded loss function were presented by Derbeko et al. (2004). Catoni (2004, 2007) and Audibert (2004) developed PAC-Bayesian and VC dimension-based risk bounds for the special case when the size of the test set is a multiple of the size of the training set. Unlike our PAC-Bayesian bound, the published transductive PAC-Bayesian bounds hold for deterministic hypotheses and for Gibbs classifiers. The bounds of Balcan and Blum (2006) for semi-supervised learning also hold in the transductive setting, making them conceptually similar to some transductive PAC-Bayesian bounds. General error bounds based on stability were developed by El-Yaniv and Pechyony (2006).

Effective applications of the general bounds mentioned above to particular

algorithms or “learning principles” is not automatic. In the case of the PAC-Bayesian bounds several such successful applications were presented in terms of appropriate “priors” that promote various structural properties of the data (see, e.g., Derbeko et al., 2004; El-Yaniv & Gerzon, 2005; Hanneke, 2006). Ad-hoc bounds for particular algorithms were developed by Belkin et al. (2004) and Johnson and Zhang (2008, 2007).

Error bounds based on Rademacher complexity were introduced by Koltchinskii (2001) and are a well-established topic in induction (see Bartlett & Mendelson, 2002, and references therein). The first Rademacher transductive risk bound was presented by Lanckriet et al. (2004, Theorem 24). This bound, which is a straightforward extension of the inductive Rademacher techniques of Bartlett and Mendelson (2002), is limited to the special case when training and test sets are of equal size. The bound presented here overcomes this limitation.

## 5.2 Definitions

### 5.2.1 Learning Model

In this paper we use a distribution-free transductive model, as defined by Vapnik (1982, Section 10.1). Consider a fixed set  $S_{m+u} \triangleq \{(x_i, y_i)\}_{i=1}^{m+u}$  of  $m + u$  points  $x_i$  in some space together with their labels  $y_i$ . The learner is provided with the (unlabeled) *full-sample*  $X_{m+u} \triangleq \{x_i\}_{i=1}^{m+u}$ . A set consisting of  $m$  points is selected from  $X_{m+u}$  uniformly at random among all subsets of size  $m$ . These  $m$  points together with their labels are given to the learner as a *training set*. Re-numbering the points we denote the unlabeled training set points by  $X_m \triangleq \{x_1, \dots, x_m\}$  and the labeled training set by  $S_m \triangleq \{(x_i, y_i)\}_{i=1}^m$ . The set of unlabeled points  $X_u \triangleq \{x_{m+1}, \dots, x_{m+u}\} = X_{m+u} \setminus X_m$  is called the *test set*. The learner’s goal is to predict the labels of the test points in  $X_u$  based on  $S_m \cup X_u$ .

The choice of the set of  $m$  points as described above can be viewed in three equivalent ways:

1. Drawing  $m$  points from  $X_{m+u}$  uniformly *without replacement*. Due to this draw, the points in the training and test sets are *dependent*.
2. Random permutation of the full sample  $X_{m+u}$  and choosing the first  $m$  points as a training set.
3. Random partitioning of  $m + u$  points into two disjoint sets of  $m$  and  $u$  points.

To emphasize different aspects of the transductive learning model, throughout the paper we use interchangeably these three views on the generation of the training and test sets.

This paper focuses on binary learning problems where labels  $y \in \{\pm 1\}$ . The learning algorithms we consider generate “soft classification” vectors  $\mathbf{h} = (h(1), \dots, h(m+u)) \in \mathbb{R}^{m+u}$ , where  $h(i)$  (or  $h(x_i)$ ) is the soft, or confidence-rated, label of example  $x_i$  given by the “hypothesis”  $\mathbf{h}$ . For actual (binary) classification of  $x_i$  the algorithm outputs  $\text{sgn}(h(i))$ . We denote by  $\mathcal{H}_{\text{out}} \subseteq \mathbb{R}^{m+u}$  the set of all possible soft classification vectors (over all possible training/test partitions) that are generated by the algorithm.

Based on the full-sample  $X_{m+u}$ , the algorithm selects an hypothesis space  $\mathcal{H} \subseteq \mathbb{R}^{m+u}$  of soft classification hypotheses. Note that  $\mathcal{H}_{\text{out}} \subseteq \mathcal{H}$ . Then, given the labels of training points the algorithm outputs one hypothesis  $\mathbf{h}$  from  $\mathcal{H}_{\text{out}} \cap \mathcal{H}$  for classification. The goal is to minimize its *test error*  $\mathcal{L}_u(\mathbf{h}) \triangleq \frac{1}{u} \sum_{i=m+1}^{m+u} \ell(h(i), y_i)$  w.r.t. the 0/1 loss function  $\ell$ . The *empirical error* of  $\mathbf{h}$  is  $\widehat{\mathcal{L}}_m(\mathbf{h}) \triangleq \frac{1}{m} \sum_{i=1}^m \ell(h(i), y_i)$  and the *full sample error* of  $\mathbf{h}$  is  $\mathcal{L}_{m+u}(\mathbf{h}) \triangleq \frac{1}{m+u} \sum_{i=1}^{m+u} \ell(h(i), y_i)$ . In this work we also use the margin loss function  $\ell_\gamma$ . For a positive real  $\gamma$ ,  $\ell_\gamma(y_1, y_2) = 0$  if  $y_1 y_2 \geq \gamma$  and  $\ell_\gamma(y_1, y_2) = \min\{1, 1 - y_1 y_2 / \gamma\}$  otherwise. The *empirical (margin) error* of  $\mathbf{h}$  is  $\widehat{\mathcal{L}}_m^\gamma(\mathbf{h}) \triangleq \frac{1}{m} \sum_{i=1}^m \ell_\gamma(h(i), y_i)$ . We denote by  $\mathcal{L}_u^\gamma(\mathbf{h})$  the margin error of the test set and by  $\mathcal{L}_{m+u}^\gamma(\mathbf{h})$  the margin full sample error.

## 5.2.2 Transductive Rademacher complexity

We adapt the inductive Rademacher complexity to our transductive setting but generalize it a bit to also include “neutral” Rademacher values.

**Definition 9 (Transductive Rademacher complexity)** *Let  $\mathcal{V} \subseteq \mathbb{R}^{m+u}$  and  $p \in [0, 1/2]$ . Let  $\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_{m+u})$  be a vector of i.i.d. random variables such that*

$$\sigma_i \triangleq \begin{cases} 1 & \text{with probability } p; \\ -1 & \text{with probability } p; \\ 0 & \text{with probability } 1 - 2p. \end{cases} \quad (5.1)$$

The transductive Rademacher complexity with parameter  $p$  is

$$R_{m+u}(\mathcal{V}, p) \triangleq \left( \frac{1}{m} + \frac{1}{u} \right) \cdot \mathbf{E}_{\boldsymbol{\sigma}} \left\{ \sup_{\mathbf{v} \in \mathcal{V}} \boldsymbol{\sigma} \cdot \mathbf{v} \right\}. \quad (5.2)$$

The need for this novel definition of Rademacher complexity is technical. Two main issues that lead to the new definition are:

1. The need to bound the test error  $\mathcal{L}_u(\mathbf{h}) = \frac{1}{u} \sum_{i=m+1}^{m+u} \ell(h(i), y_i)$ . Notice that in inductive risk bounds the standard definition of Rademacher complexity (see Definition 10 below), with binary values of  $\sigma_i$ , is used to bound the generalization error, which is an inductive analogue of the full sample error  $\mathcal{L}_{m+u}(\mathbf{h}) = \frac{1}{m+u} \sum_{i=1}^{m+u} \ell(h(i), y_i)$ .

2. Different sizes ( $m$  and  $u$  respectively) of training and test set.

See Section 5.3.1 for more technical details that lead to the above definition of Rademacher complexity.

For the sake of comparison we also state the inductive definition of Rademacher complexity.

**Definition 10 (Inductive Rademacher complexity, Koltchinskii, 2001)**

Let  $\mathcal{D}$  be a probability distribution over  $\mathcal{X}$ . Suppose that the examples  $X_n = \{x_i\}_{i=1}^n$  are sampled independently from  $\mathcal{X}$  according to  $\mathcal{D}$ . Let  $\mathcal{F}$  be a class of functions mapping  $\mathcal{X}$  to  $\mathbb{R}$ . Let  $\sigma = \{\sigma_i\}_{i=1}^n$  be an independent uniform  $\{\pm 1\}$ -valued random variables,  $\sigma_i = 1$  with probability  $1/2$  and  $\sigma_i = -1$  with the same probability. The empirical Rademacher complexity is<sup>1</sup>  $\widehat{R}_n^{(\text{ind})}(\mathcal{F}) \triangleq \frac{2}{n} \mathbf{E}_{\sigma} \left\{ \sup_{f \in \mathcal{F}} \sum_{i=1}^n \sigma_i f(x_i) \right\}$  and the Rademacher complexity of  $\mathcal{F}$  is  $R_n^{(\text{ind})}(\mathcal{F}) \triangleq \mathbf{E}_{X_n \sim \mathcal{D}^n} \left\{ \widehat{R}_n^{(\text{ind})}(\mathcal{F}) \right\}$ .

For the case  $p = 1/2$ ,  $m = u$  and  $n \triangleq m + u$  we have that  $R_{m+u}(\mathcal{V}) = 2\widehat{R}_{m+u}^{(\text{ind})}(\mathcal{V})$ . Whenever  $p < 1/2$ , some Rademacher variables will attain (neutral) zero values and reduce the complexity (see Lemma 9). We use this property to tighten our bounds.

Notice that the transductive complexity is an empirical quantity that does not depend on any underlying distribution, including the one over the choices of the training set. Since in distribution-free transductive model the unlabeled full sample of training and test points is fixed, in transductive Rademacher complexity we don't need the outer expectation, which appears in the inductive definition. Also, the transductive complexity depends on both the (unlabeled) training and test points whereas the inductive complexity only depends only on the (unlabeled) training points.

The following lemma, whose proof appears in Appendix 5.8.1, states that  $R_{m+u}(\mathcal{V}, p)$  is monotone increasing with  $p$ . The proof is based on the technique used in the proof of Lemma 5 in Meir and Zhang (2003).

**Lemma 9** For any  $\mathcal{V} \subseteq \mathbb{R}^{m+u}$  and  $0 \leq p_1 < p_2 \leq 1/2$ ,  $R_{m+u}(\mathcal{V}, p_1) < R_{m+u}(\mathcal{V}, p_2)$ .

In the forthcoming results we utilize the transductive Rademacher complexity with  $p_0 \triangleq \frac{mu}{(m+u)^2}$ . We abbreviate  $R_{m+u}(\mathcal{V}) \triangleq R_{m+u}(\mathcal{V}, p_0)$ . By Lemma 9, all our bounds also apply to  $R_{m+u}(\mathcal{V}, p)$  for all  $p > p_0$ . Since  $p_0 < \frac{1}{2}$ , the Rademacher complexity involved in our results is strictly smaller than the standard inductive Rademacher complexity defined over  $X_{m+u}$ . If transduction approaches the induction, namely  $m$  is fixed and  $u \rightarrow \infty$ , then  $\widehat{R}_{m+u}^{(\text{ind})}(\mathcal{V}) \rightarrow 2R_{m+u}(\mathcal{V})$ .

<sup>1</sup>The original definition of Rademacher complexity, as given by Koltchinskii (2001), is slightly different from the one presented here, and contains  $\sup_{f \in \mathcal{F}} |\sum_{i=1}^n \sigma_i f(x_i)|$  instead of  $\sup_{f \in \mathcal{F}} \sum_{i=1}^n \sigma_i f(x_i)$ . However, from the conceptual point of view, Definition 10 and the one given by Koltchinskii are equivalent.

## 5.3 Uniform Rademacher error bound

In this section we develop a transductive risk bound, which is based on transductive Rademacher complexity (Definition 9). The derivation follows the standard two-step scheme, as in induction<sup>2</sup>:

1. Derivation of a uniform concentration inequality for a set of vectors (or functions). This inequality depends on the Rademacher complexity of the set. After substituting to the vectors (or functions) the values of the loss functions, we obtain an error bound depending on the Rademacher complexity of the values of the loss function. This step is done in Section 5.3.1.
2. In order to bound the Rademacher complexity in terms of the properties of the hypothesis space, the Rademacher complexity is ‘translated’, using its contraction property (Ledoux & Talagrand, 1991, Theorem 4.12), from the domain of loss function values to the domain of soft hypotheses from the hypothesis space. This step is done in Section 5.3.2.

As we show in Sections 5.3.1 and 5.3.2, the adaptation of both these steps to the transductive setting is not immediate and involves several novel ideas. In Section 5.3.3 we combine the results of these two steps and obtain a transductive Rademacher risk bound. We also provide a thorough comparison of our risk bound with the corresponding inductive bound.

### 5.3.1 Uniform concentration inequality for a set of vectors

As in induction (Koltchinskii & Panchenko, 2002), our derivation of a uniform concentration inequality for a set of vectors consists of three steps:

1. Introduction of the “ghost sample”.
2. Bounding the supremum  $\sup_{\mathbf{h} \in \mathcal{H}} g(\mathbf{h})$ , where  $g(\mathbf{h})$  is some random real-valued function, with its expectation using a concentration inequality for functions of random variables.
3. Bounding the expectation of the supremum using Rademacher variables.

While we follow these three steps as in induction, the establishment of each of these steps can not be achieved using inductive techniques. Throughout this section, after performing the derivation of each step in transductive context we discuss its differences from its inductive counterpart.

Just before the derivation we make several new definitions. Let  $\mathcal{V}$  be a set of vectors in  $[B_1, B_2]^{m+u}$ ,  $B_1 \leq 0$ ,  $B_2 \geq 0$  and set  $B \triangleq B_2 - B_1$ ,  $B_{\max} =$

---

<sup>2</sup>This scheme was introduced by Koltchinskii and Panchenko (2002). The examples of other uses of this technique can be found in (Bartlett & Mendelson, 2002) and (Meir & Zhang, 2003).

$\max(|B_1|, |B_2|)$ . Consider two independent permutations of  $I_1^{m+u}$ ,  $\mathbf{Z}$  and  $\mathbf{Z}'$ . For any  $\mathbf{v} \in \mathcal{V}$  denote by

$$\mathbf{v}(\mathbf{Z}) \triangleq (v(Z_1), v(Z_2), \dots, v(Z_{m+u})) ,$$

the vector  $\mathbf{v}$  permuted according to  $\mathbf{Z}$ . We use the following abbreviations for averages of  $\mathbf{v}$  over subsets of its components:  $\mathbf{H}_k\{\mathbf{v}(\mathbf{Z})\} \triangleq \frac{1}{m} \sum_{i=1}^k v(Z_i)$ ,  $\mathbf{T}_k\{\mathbf{v}(\mathbf{Z})\} \triangleq \frac{1}{u} \sum_{i=k+1}^{m+u} v(Z_i)$  (note that  $\mathbf{H}$  stands for ‘head’ and  $\mathbf{T}$ , for ‘tail’). In the special case where  $k = m$  we set  $\mathbf{H}\{\mathbf{v}(\mathbf{Z})\} \triangleq \mathbf{H}_m\{\mathbf{v}(\mathbf{Z})\}$ , and  $\mathbf{T}\{\mathbf{v}(\mathbf{Z})\} \triangleq \mathbf{T}_m\{\mathbf{v}(\mathbf{Z})\}$ . Finally, the average component of  $\mathbf{v}$  is denoted  $\bar{v} \triangleq \frac{1}{m+u} \sum_{i=1}^{m+u} v(i)$ .

*Step 1: Introduction of the ghost sample.*

For any  $\mathbf{v} \in \mathcal{V}$  and any permutation  $\mathbf{Z}$  of  $I_1^{m+u}$  we have

$$\begin{aligned} \mathbf{T}\{\mathbf{v}(\mathbf{Z})\} &= \mathbf{H}\{\mathbf{v}(\mathbf{Z})\} + \mathbf{T}\{\mathbf{v}(\mathbf{Z})\} - \mathbf{H}\{\mathbf{v}(\mathbf{Z})\} & (5.3) \\ &\leq \mathbf{H}\{\mathbf{v}(\mathbf{Z})\} + \sup_{\mathbf{v} \in \mathcal{V}} \left[ \mathbf{T}\{\mathbf{v}(\mathbf{Z})\} - \bar{v} + \bar{v} - \mathbf{H}\{\mathbf{v}(\mathbf{Z})\} \right] \\ &= \mathbf{H}\{\mathbf{v}(\mathbf{Z})\} + \\ &\quad \sup_{\mathbf{v} \in \mathcal{V}} \left[ \mathbf{T}\{\mathbf{v}(\mathbf{Z})\} - \mathbf{E}_{\mathbf{Z}'} \mathbf{T}\{\mathbf{v}(\mathbf{Z}')\} + \mathbf{E}_{\mathbf{Z}'} \mathbf{H}\{\mathbf{v}(\mathbf{Z}')\} - \mathbf{H}\{\mathbf{v}(\mathbf{Z})\} \right] \\ &\leq \mathbf{H}\{\mathbf{v}(\mathbf{Z})\} + \\ &\quad \underbrace{\mathbf{E}_{\mathbf{Z}'} \sup_{\mathbf{v} \in \mathcal{V}} \left[ \mathbf{T}\{\mathbf{v}(\mathbf{Z})\} - \mathbf{T}\{\mathbf{v}(\mathbf{Z}')\} + \mathbf{H}\{\mathbf{v}(\mathbf{Z}')\} - \mathbf{H}\{\mathbf{v}(\mathbf{Z})\} \right]}_{\triangleq g(\mathbf{Z})} \end{aligned} \quad (5.4)$$

**Remark 4** *In this derivation the “ghost sample” is a permutation  $\mathbf{Z}'$  of  $m + u$  elements drawn from the same distribution as  $\mathbf{Z}$ . In inductive Rademacher-based risk bounds the ghost sample is a new training set of size  $m$ , independently drawn from the original one. Note that in our transductive setting the ghost sample corresponds to the independent draw of training/test set partition, which is equivalent to the independent draw of random permutation  $\mathbf{Z}'$ .*

**Remark 5** *In principle we could avoid the introduction of the ghost sample  $\mathbf{Z}'$  and consider  $m$  elements in  $\mathbf{H}\{\mathbf{v}(\mathbf{Z})\}$  as ghosts of  $u$  elements in  $\mathbf{T}\{\mathbf{v}(\mathbf{Z})\}$ . This approach would lead to a new definition of Rademacher averages (with  $\sigma_i = -1/m$  with probability  $m/(m+u)$  and  $1/u$  with probability  $u/(m+u)$ ). With this definition we can obtain Corollary 1. However, since the distribution of alternative Rademacher averages is not symmetric around zero, technically we do not know how to prove the Lemma 5 (the contraction property).*

*Step 2: Bounding the supremum with its expectation.*

Let  $S \triangleq \frac{m+u}{(m+u-1/2)(1-1/(2\max(m,u)))}$ . For sufficiently large  $m$  and  $u$ , the value of  $S$  is almost 1. The function  $g(\mathbf{Z})$  is  $(m, u)$ -permutation symmetric in  $\mathbf{Z}$ . It can be verified that  $|g(\mathbf{Z}) - g(\mathbf{Z}^{ij})| \leq B \left(\frac{1}{m} + \frac{1}{u}\right)$ . Therefore, we can apply Lemma 1 with  $\beta \triangleq B \left(\frac{1}{m} + \frac{1}{u}\right)$  to  $g(\mathbf{Z})$ . We obtain, with probability of at least  $1 - \delta$  over random permutation  $\mathbf{Z}$  of  $I_1^{m+u}$ , for all  $\mathbf{v} \in \mathcal{V}$ :

$$\mathbf{T}\{\mathbf{v}(\mathbf{Z})\} \leq \mathbf{H}\{\mathbf{v}(\mathbf{Z})\} + \mathbf{E}_{\mathbf{Z}}\{g(\mathbf{Z})\} + B\sqrt{\frac{S}{2} \left(\frac{1}{m} + \frac{1}{u}\right) \ln \frac{1}{\delta}}. \quad (5.5)$$

**Remark 6** *In induction this step is performed using an application of McDiarmid's bounded difference inequality (McDiarmid, 1989, Lemma 1.2). We cannot apply this inequality in our setting since the function under the supremum (i.e.  $g(\mathbf{Z})$ ) is not a function over independent variables, but rather over permutations. Our Lemma 1 is a substitute for the bounded difference inequality in this step.*

*Step 3: Bounding the expectation over the supremum using Rademacher random variables.*

Our goal is to bound the expectation  $\mathbf{E}_{\mathbf{Z}}\{g(\mathbf{Z})\}$ . This is done in the following lemma.

**Lemma 10** *Let  $\mathbf{Z}$  be a random permutation of  $I_1^{m+u}$ . Let  $c_0 \triangleq \sqrt{\frac{32\ln(4e)}{3}} < 5.05$ . Then*

$$\mathbf{E}_{\mathbf{Z}}\{g(\mathbf{Z})\} \leq R_{m+u}(\mathcal{V}) + c_0 B_{\max} \left(\frac{1}{u} + \frac{1}{m}\right) \sqrt{\min(m, u)}. \quad (5.6)$$

The (long) proof of this lemma appears in Appendix 5.8.2. The proof is based on ideas from the proof of Lemma 3 from Bartlett and Mendelson (2002).

**Remark 7** *The technique we use to bound the expectation of the supremum is more complicated than the one commonly used in induction (e.g., see Koltchinskii & Panchenko, 2002). This is caused by the structure of the function under the supremum (i.e.,  $g(\mathbf{Z})$ ). From a conceptual point of view, this step utilizes our novel definition of transductive Rademacher complexity.*

By combining (5.5) and Lemma 10 we obtain the next concentration inequality, which is the main result of this section.

**Theorem 5** *Let  $B_1 \leq 0$ ,  $B_2 \geq 0$  and  $\mathcal{V}$  be a (possibly infinite) set of real-valued vectors in  $[B_1, B_2]^{m+u}$ . Let  $B \triangleq B_2 - B_1$  and  $B_{\max} \triangleq \max(|B_1|, |B_2|)$ . Let  $Q \triangleq \left(\frac{1}{u} + \frac{1}{m}\right)$ ,  $S \triangleq \frac{m+u}{(m+u-1/2)(1-1/(2\max(m,u)))}$  and  $c_0 \triangleq \sqrt{\frac{32\ln(4e)}{3}} < 5.05$ . Then*

with probability of at least  $1 - \delta$  over random permutation  $\mathbf{Z}$  of  $I_1^{m+u}$ , for all  $\mathbf{v} \in \mathcal{V}$ ,

$$\mathbf{T}\{\mathbf{v}(\mathbf{Z})\} \leq \mathbf{H}\{\mathbf{v}(\mathbf{Z})\} + R_{m+u}(\mathcal{V}) + B_{\max}c_0Q\sqrt{\min(m, u)} + B\sqrt{\frac{S}{2}Q \ln \frac{1}{\delta}}. \quad (5.7)$$

We defer the analysis of the slack terms  $B_{\max}c_0Q\sqrt{\min(m, u)}$  and  $B\sqrt{\frac{S}{2}Q \ln \frac{1}{\delta}}$  to Section 5.3.3. We now instantiate the inequality (5.7) to obtain our first risk bound. The idea is to apply Theorem 5 with an appropriate instantiation of the set  $\mathcal{V}$  so that  $\mathbf{T}\{\mathbf{v}(\mathbf{Z})\}$  will correspond to the test error and  $\mathbf{H}\{\mathbf{v}(\mathbf{Z})\}$  to the empirical error. For a true (unknown) labeling of the full-sample  $Y$  and any  $\mathbf{h} \in \mathcal{H}_{\text{out}}$  we define

$$\boldsymbol{\ell}^Y(\mathbf{h}) \triangleq (\ell(h(1), y_1), \dots, \ell(h(m+u), y_{m+u}))$$

and set  $L_{\mathcal{H}} = \{\mathbf{v} : \mathbf{v} = \boldsymbol{\ell}^Y(\mathbf{h}), \mathbf{h} \in \mathcal{H}_{\text{out}}\}$ . Thus  $\boldsymbol{\ell}^Y(\mathbf{h})$  is a vector of the values of the 0/1 loss over all full sample examples, when transductive algorithm is operated on some training/test partition. The set  $L_{\mathcal{H}}$  is the set of all possible vectors  $\boldsymbol{\ell}^Y(\mathbf{h})$ , over all possible training/test partitions. We apply Theorem 5 with  $\mathcal{V} \triangleq L_{\mathcal{H}}$ ,  $\mathbf{v} \triangleq \boldsymbol{\ell}(\mathbf{h})$ ,  $B_{\max} = B = 1$  and obtain the following corollary:

**Corollary 1** *Let  $Q$ ,  $S$  and  $c_0$  be as defined in Theorem 5. For any  $\delta > 0$ , with probability of at least  $1 - \delta$  over the choice of the training set from  $X_{m+u}$ , for all  $\mathbf{h} \in \mathcal{H}_{\text{out}}$ ,*

$$\mathcal{L}_u(\mathbf{h}) \leq \widehat{\mathcal{L}}_m(\mathbf{h}) + R_{m+u}(L_{\mathcal{H}}) + B_{\max}c_0Q\sqrt{\min(m, u)} + \sqrt{\frac{S}{2}Q \ln \frac{1}{\delta}}. \quad (5.8)$$

We defer the analysis of the slack terms  $B_{\max}c_0Q\sqrt{\min(m, u)}$  and  $B\sqrt{\frac{S}{2}Q \ln \frac{1}{\delta}}$  to Section 5.3.3. While the bound (5.8) is obtained by a straightforward application of the concentration inequality (5.7), it is not convenient to deal with. That's because it is not clear how to bound the Rademacher complexity  $R_{m+u}(L_{\mathcal{H}})$  of the 0/1 loss values in terms of the properties of transductive algorithm. In the next sections we eliminate this deficiency by utilizing margin loss function.

### 5.3.2 Contraction of Rademacher complexity

The following lemma is a version of the well-known ‘contraction principle’ of the theory of Rademacher averages (see Theorem 4.12 of Ledoux & Talagrand, 1991, and Ambroladze et al., 2007). The lemma is an adaptation, which accommodates the transductive Rademacher variables, of Lemma 5 of Meir and Zhang (2003). The proof is provided in Appendix 5.8.3.

**Lemma 11** *Let  $\mathcal{V} \subseteq \mathbb{R}^{m+u}$  be a set of vectors. Let  $f$  and  $g$  be real-valued functions. Let  $\boldsymbol{\sigma} = \{\sigma_i\}_{i=1}^{m+u}$  be Rademacher variables, as defined in (5.1). If for all  $1 \leq i \leq m+u$  and any  $\mathbf{v}, \mathbf{v}' \in \mathcal{V}$ ,  $|f(v_i) - f(v'_i)| \leq |g(v_i) - g(v'_i)|$ , then*

$$\mathbf{E}_{\boldsymbol{\sigma}} \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^{m+u} \sigma_i f(v_i) \right] \leq \mathbf{E}_{\boldsymbol{\sigma}} \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^{m+u} \sigma_i g(v_i) \right]. \quad (5.9)$$

Let  $Y \in \{\pm 1\}^{m+u}$  be a true (unknown) labeling of the full-sample. Similarly to what was done in the derivation of Corollary 1, for any  $\mathbf{h} \in \mathcal{H}_{\text{out}}$  we define  $\ell_\gamma^Y(h(i)) \triangleq \ell_\gamma(h(i), y_i)$  and

$$\boldsymbol{\ell}_\gamma^Y(\mathbf{h}) \triangleq (\ell_\gamma^Y(h(1)), \dots, \ell_\gamma^Y(h(m+u)))$$

and set  $L_{\mathcal{H}}^\gamma = \{\mathbf{v} : \mathbf{v} = \boldsymbol{\ell}_\gamma^Y(\mathbf{h}), \mathbf{h} \in \mathcal{H}_{\text{out}}\}$ . Noting that  $\ell_\gamma^Y$  satisfies the Lipschitz condition  $|\ell_\gamma^Y(h(i)) - \ell_\gamma^Y(h'(i))| \leq \frac{1}{\gamma} |h(i) - h'(i)|$ , we apply Lemma 11 with  $\mathcal{V} \triangleq L_{\mathcal{H}}^\gamma$ ,  $f(v_i) \triangleq \ell_\gamma^Y(h(i))$  and  $g(v_i) \triangleq h(i)/\gamma$ , to get

$$\mathbf{E}_{\boldsymbol{\sigma}} \left\{ \sup_{\mathbf{h} \in \mathcal{H}_{\text{out}}} \sum_{i=1}^{m+u} \sigma_i \ell_\gamma^Y(h(i)) \right\} \leq \frac{1}{\gamma} \mathbf{E}_{\boldsymbol{\sigma}} \left\{ \sup_{\mathbf{h} \in \mathcal{H}_{\text{out}}} \sum_{i=1}^{m+u} \sigma_i h(i) \right\}. \quad (5.10)$$

It follows from (5.10) that

$$R_{m+u}(L_{\mathcal{H}}^\gamma) \leq \frac{1}{\gamma} R_{m+u}(\mathcal{H}_{\text{out}}). \quad (5.11)$$

### 5.3.3 Risk bound and comparison with related results

Applying Theorem 5 with  $\mathcal{V} \triangleq L_{\mathcal{H}}^\gamma$ ,  $\mathbf{v} \triangleq \boldsymbol{\ell}_\gamma(\mathbf{h})$ ,  $B_{\max} = B = 1$ , and using the inequality (5.11) we obtain<sup>3</sup>:

**Theorem 6** *Let  $\mathcal{H}_{\text{out}}$  be the set of full-sample soft labelings of the algorithm, generated by operating it on all possible training/test set partitions. The choice of  $\mathcal{H}_{\text{out}}$  can depend on the full-sample  $X_{m+u}$ . Let  $c_0 = \sqrt{\frac{32 \ln(4e)}{3}} < 5.05$ ,  $Q \triangleq \left(\frac{1}{u} + \frac{1}{m}\right)$  and  $S \triangleq \frac{m+u}{(m+u-1/2)(1-1/(2 \max(m,u)))}$ . For any fixed  $\gamma$ , with probability of at least  $1 - \delta$  over the choice of the training set from  $X_{m+u}$ , for all  $\mathbf{h} \in \mathcal{H}_{\text{out}}$ ,*

$$\mathcal{L}_u(\mathbf{h}) \leq \mathcal{L}_u^\gamma(\mathbf{h}) \leq \widehat{\mathcal{L}}_m^\gamma(\mathbf{h}) + \frac{R_{m+u}(\mathcal{H}_{\text{out}})}{\gamma} + c_0 Q \sqrt{\min(m, u)} + \sqrt{\frac{SQ}{2} \ln \frac{1}{\delta}}. \quad (5.12)$$

<sup>3</sup>This bound holds for any *fixed* margin parameter  $\gamma$ . Using the technique of the proof of Theorem 18 of Bousquet and Elisseeff (2002), we can also obtain a bound that is uniform in  $\gamma$ .

For large enough values of  $m$  and  $u$  the value of  $S$  is close to 1. Therefore the slack term  $c_0 Q \sqrt{\min(m, u)} + \sqrt{\frac{S}{2}} Q \ln \frac{1}{\delta}$  is of order  $O\left(1/\sqrt{\min(m, u)}\right)$ . The convergence rate of  $O\left(1/\sqrt{\min(m, u)}\right)$  can be very slow if  $m$  is very small or  $u \ll m$ . Slow rate for small  $m$  is not surprising, but a latter case of  $u \ll m$  is somewhat surprising. However note that if  $u \ll m$  then the mean  $\mu$  of  $u$  elements, drawn from  $m + u$  elements, has a large variance. Hence, in this case any high-confidence interval for the estimation of  $\mu$  will be large. This confidence interval is reflected in the slack term of (5.12).

We now compare the bound (5.12) with the Rademacher-based inductive risk bounds. We use the following variant of Rademacher-based inductive risk bound (Meir & Zhang, 2003):

**Theorem 7** *Let  $\mathcal{D}$  be a probability distribution over  $\mathcal{X}$ . Suppose that a set of examples  $S_m = \{(x_i, y_i)\}_{i=1}^m$  is sampled i.i.d. from  $\mathcal{X}$  according to  $\mathcal{D}$ . Let  $\mathcal{F}$  be a class of functions each maps  $\mathcal{X}$  to  $\mathbb{R}$  and  $R_m^{(\text{ind})}(\mathcal{F})$  be the inductive Rademacher complexity of  $\mathcal{F}$  (Definition 10). Let  $\mathcal{L}(f) = \mathbf{E}_{(x,y) \sim \mathcal{D}}\{\ell(f(x), y)\}$  and  $\widehat{\mathcal{L}}^\gamma(f) = \frac{1}{m} \sum_{i=1}^m \ell_\gamma(f(x_i), y_i)$  be respectively the 0/1 generalization error and empirical margin error of  $f$ . Then for any  $\delta > 0$  and  $\gamma > 0$ , with probability of at least  $1 - \delta$  over the random draw of  $S_m$ , for any  $f \in \mathcal{F}$ ,*

$$\mathcal{L}(f) \leq \widehat{\mathcal{L}}^\gamma(f) + \frac{R_m^{(\text{ind})}(\mathcal{F})}{\gamma} + \sqrt{\frac{\ln(1/\delta)}{2m}}. \quad (5.13)$$

The slack term in the bound (5.13) is of order  $O(1/\sqrt{m})$ . The bounds (5.12) and (5.13) are not quantitatively comparable. The inductive bound holds with high probability over the random selection of  $m$  examples from some distribution  $\mathcal{D}$ . This bound is on average (generalization) error of some hypothesis over the distribution  $\mathcal{D}$ . The transductive bound holds with high probability over the random selection of a training/test partition. This bound is on the test error of some hypothesis over a particular set of  $u$  points.

A kind of meaningful comparison can be obtained as follows. Using the given full (transductive) sample  $X_{m+u}$ , we define a corresponding inductive distribution  $\mathcal{D}_{\text{trans}}$  as the uniform distribution over  $X_{m+u}$ ; that is, a training set of size  $m$  will be generated by sampling from  $X_{m+u}$   $m$  times with replacements. Given an inductive hypothesis space  $\mathcal{F} = \{f\}$  of function we define the transductive hypothesis space  $\mathcal{H}_{\mathcal{F}}$  as a projection of  $\mathcal{F}$  into the full sample  $X_{m+u}$ :  $\mathcal{H}_{\mathcal{F}} = \{\mathbf{h} \in \mathbb{R}^{m+u} : \exists f \in \mathcal{F}, \forall 1 \leq i \leq m+u, h(i) = f(x_i)\}$ . By such definition of  $\mathcal{H}_{\mathcal{F}}$ ,  $\mathcal{L}(f) = \mathcal{L}_{m+u}(\mathbf{h})$ .

Our final step towards a meaningful comparison would be to translate a transductive bound of the form  $\mathcal{L}_u(\mathbf{h}) \leq \widehat{\mathcal{L}}_m^\gamma(\mathbf{h}) + \text{slack}$  to a bound on the average error

of the hypothesis<sup>4</sup>  $\mathbf{h}$ :

$$\begin{aligned} \mathcal{L}_{m+u}(\mathbf{h}) &\leq \mathcal{L}_{m+u}^\gamma(\mathbf{h}) = \frac{m\widehat{\mathcal{L}}_m^\gamma(\mathbf{h}) + u\mathcal{L}_u^\gamma(\mathbf{h})}{m+u} \leq \frac{m\widehat{\mathcal{L}}_m^\gamma(\mathbf{h}) + u(\widehat{\mathcal{L}}_m^\gamma(\mathbf{h}) + \text{slack})}{m+u} \\ &= \widehat{\mathcal{L}}_m^\gamma(\mathbf{h}) + \frac{u}{m+u} \cdot \text{slack} \end{aligned} \quad (5.14)$$

We instantiate (5.14) to the bound (5.12) and obtain

$$\mathcal{L}_{m+u}(\mathbf{h}) \leq \widehat{\mathcal{L}}_m^\gamma(\mathbf{h}) + \frac{u}{m+u} \frac{R_{m+u}(\mathcal{H}_{\mathcal{F}})}{\gamma} + \frac{u}{m+u} \left[ c_0 Q \sqrt{\min(m, u)} + \sqrt{\frac{SQ}{2} \ln \frac{1}{\delta}} \right]. \quad (5.15)$$

Now given a transductive problem we consider the corresponding inductive bound obtained from (5.13) under the distribution  $\mathcal{D}_{\text{trans}}$  and compare it to the bound (5.15).

Note that in the inductive bound (5.13) the sampling of the training set is done with replacement, while in the transductive bound (5.15) it is done without replacement. Thus, in the inductive case the actual number of distinct training examples may be smaller than  $m$ .

The bounds (5.13) and (5.15) consist of three terms: empirical error term (first summand in (5.13) and (5.15)), the term depending on the Rademacher complexity (second summand in (5.13) and (5.15)) and the slack term (third summand in (5.13) and third and fourth summands in (5.15)). The empirical error terms are the same in both bounds. It is hard to compare analytically the Rademacher complexity terms. This is because the inductive bound is derived for the setting of sampling with replacement and the transductive bound is derived for the setting of sampling without replacement. Thus, in the transductive Rademacher complexity each example  $x_i \in X_{m+u}$  appears in  $R_{m+u}(\mathcal{H}_{\text{out}})$  only once and is multiplied by  $\sigma_i$ . In contrast, due to the sampling with replacement, in the inductive Rademacher term the example  $x_i \in X_{m+u}$  can appear several times in  $R_m^{(\text{ind})}(\mathcal{F})$ , multiplied by different values of the Rademacher variables.

Nevertheless, in transduction we have a full control over the Rademacher complexity (since we can choose  $\mathcal{H}_{\text{out}}$  after observing the full sample  $X_{m+u}$ ) and can choose an hypothesis space  $\mathcal{H}_{\text{out}}$  with arbitrarily small Rademacher complexity. In induction we choose  $\mathcal{F}$  before observing any data. Hence, if we are lucky with the full sample  $X_{m+u}$  then  $R_m^{(\text{ind})}(\mathcal{F})$  is small, and if we are unlucky with  $X_{m+u}$

---

<sup>4</sup>Alternatively, to compare (5.12) and (5.13), we could try to express the bound (5.13) as the bound on the error of  $f$  on  $X_u$  (the randomly drawn subset of  $u$  examples). The bound (5.13) holds for the setting of random draws with replacement. In this setting the number of unique training examples can be smaller than  $m$  and thus the number of the remaining test examples is larger than  $u$ . Hence the draw of  $m$  training examples with replacement does not imply the draw of the subset of  $u$  test examples, as in transductive setting. Thus we cannot express the bound (5.13) as the bound on the randomly drawn  $X_u$

then  $R_m^{(\text{ind})}(\mathcal{F})$  can be large. Thus, under these provisions we can argue that the transductive Rademacher term is not larger than the inductive counterpart.

Finally, we compare the slack terms in (5.13) and (5.15). If  $m \approx u$  or  $m \ll u$  then the slack term of (5.15) is of order  $O(1/\sqrt{m})$ , which is the same as the corresponding term in (5.13). But if  $m \gg u$  then the slack term of (5.15) is of order  $O(1/(m\sqrt{u}))$ , which is much smaller than  $O(1/\sqrt{m})$  of the slack term in (5.13).

Based on the comparison of the corresponding terms in (5.13) and (5.15) our conclusion is that in the regime of  $u \ll m$  the transductive bound is significantly tighter than the inductive one.

## 5.4 Unlabeled-Labeled Representation (ULR) of transductive algorithms

Let  $r$  be any natural number and let  $U$  be an  $(m+u) \times r$  matrix depending only on  $X_{m+u}$ . Let  $\boldsymbol{\alpha}$  be an  $r \times 1$  vector that may depend on both  $S_m$  and  $X_u$ . The soft classification output  $\mathbf{h}$  of any transductive algorithm can be represented by

$$\mathbf{h} = U \cdot \boldsymbol{\alpha} . \quad (5.16)$$

We refer to (5.16) as an *unlabeled-labeled representation (ULR)*. In this section we develop bounds on the Rademacher complexity of algorithms based on their ULRs. We note that any transductive algorithm has a trivial ULR, for example, by taking  $r = m+u$ , setting  $U$  to be the identity matrix and assigning  $\boldsymbol{\alpha}$  to any desired (soft) labels. We are interested in “non-trivial” ULRs and provide useful bounds for such representations.<sup>5</sup>

In a “vanilla” ULR,  $U$  is an  $(m+u) \times (m+u)$  matrix and  $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_{m+u})$  simply specifies the given labels in  $S_m$  (where  $\alpha_i = y_i$  for labeled points, and  $\alpha_i = 0$  otherwise). From our point of view any vanilla ULR is not trivial because  $\boldsymbol{\alpha}$  does not encode the final classification of the algorithm. For example, the algorithm of Zhou et al. (2004) straightforwardly admits a vanilla ULR. On the other hand, the natural (non-trivial) ULR of the algorithms of Zhu et al. (2003) and Belkin and Niyogi (2004) are not of the vanilla type. For some algorithms it is not necessarily obvious how to find non-trivial ULRs. In Sections 5.5 we consider two such cases – in particular, the algorithms of Joachims (2003) and Belkin et al. (2004).

The rest of this section is organized as follows. In Section 5.4.1 we present a generic bound on the Rademacher complexity of any transductive algorithm based on its ULR. In Section 5.4.2 we consider a case when the matrix  $U$  is a kernel

---

<sup>5</sup>For the trivial representation where  $U$  is the identity matrix multiplied by constant we show in Lemma 12 that the risk bound (5.12), combined with the forthcoming Rademacher complexity bound (5.19), is greater than 1.

matrix. For this case we develop another bound on the transductive Rademacher complexity. Finally, in Section 5.4.3 we present a method of computing high-confidence estimate of the transductive Rademacher complexity.

### 5.4.1 Generic bound on transductive Rademacher complexity

We now present a bound on the transductive Rademacher complexity of any transductive algorithm based on its ULR. Let  $\{\lambda_i\}_{i=1}^r$  be the singular values of  $U$ . We use the well-known fact that  $\|U\|_{\text{Fro}} = \sqrt{\sum_{i=1}^r \lambda_i^2}$ , where  $\|U\|_{\text{Fro}} \triangleq \sqrt{\sum_{i,j} (U(i,j))^2}$  is the Frobenius norm of  $U$ . Suppose that  $\|\alpha\|_2 \leq \mu_1$  for some  $\mu_1$ . Let  $\mathcal{H}_{\text{out}} \triangleq \mathcal{H}_{\text{out}}(U)$  be the set of all possible outputs of the algorithm when operated on all possible training/test set partitions of the full-sample  $X_{m+u}$ . Let  $Q \triangleq \frac{1}{m} + \frac{1}{u}$ . Using the abbreviation  $U(i, \cdot)$  for the  $i$ th row of  $U$  and following the proof idea of Lemma 22 of Bartlett and Mendelson (2002), we have that

$$\begin{aligned}
R_{m+u}(\mathcal{H}_{\text{out}}) &= Q \cdot \mathbf{E}_{\sigma} \left\{ \sup_{\mathbf{h} \in \mathcal{H}_{\text{out}}} \sum_{i=1}^{m+u} \sigma_i h(x_i) \right\} \\
&= Q \cdot \mathbf{E}_{\sigma} \left\{ \sup_{\alpha: \|\alpha\|_2 \leq \mu_1} \sum_{i=1}^{m+u} \sigma_i \langle \alpha, U(i, \cdot) \rangle \right\} \\
&= Q \cdot \mathbf{E}_{\sigma} \left\{ \sup_{\alpha: \|\alpha\|_2 \leq \mu_1} \langle \alpha, \sum_{i=1}^{m+u} \sigma_i U(i, \cdot) \rangle \right\} \\
&= Q \mu_1 \mathbf{E}_{\sigma} \left\{ \left\| \sum_{i=1}^{m+u} \sigma_i U(i, \cdot) \right\|_2 \right\} \tag{5.17}
\end{aligned}$$

$$\begin{aligned}
&= Q \mu_1 \mathbf{E}_{\sigma} \left\{ \sqrt{\sum_{i,j=1}^{m+u} \sigma_i \sigma_j \langle U(i, \cdot), U(j, \cdot) \rangle} \right\} \\
&\leq Q \mu_1 \sqrt{\sum_{i,j=1}^{m+u} \mathbf{E}_{\sigma} \{ \sigma_i \sigma_j \langle U(i, \cdot), U(j, \cdot) \rangle \}} \tag{5.18}
\end{aligned}$$

$$\begin{aligned}
&= \mu_1 \sqrt{\sum_{i=1}^{m+u} \frac{2}{mu} \langle U(i, \cdot), U(i, \cdot) \rangle} = \mu_1 \sqrt{\frac{2}{mu} \|U\|_{\text{Fro}}^2} \\
&= \mu_1 \sqrt{\frac{2}{mu} \sum_{i=1}^r \lambda_i^2} . \tag{5.19}
\end{aligned}$$

where (5.17) and (5.18) are obtained using, respectively, the Cauchy-Schwarz and Jensen inequalities. Using the bound (5.19) in conjunction with Theorem 6 we

immediately get a data-dependent error bound for any algorithm, which can be computed once we derive an upper bound on the maximal length of possible values of the  $\alpha$  vector, appearing in its ULR. Notice that for any vanilla ULR (and thus for the “consistency method” of Zhou et al. (2004)),  $\mu_1 = \sqrt{m}$ . In Section 5.5 we derive a tight bound on  $\mu_1$  for non-trivial ULRs of SGT of Joachims (2003) and of the Tikhonov regularization method of Belkin et al. (2004).

The bound (5.19) is syntactically similar in form to a corresponding inductive Rademacher bound for kernel machines (Bartlett & Mendelson, 2002). However, as noted above, the fundamental difference is that in induction, the choice of the kernel (and therefore  $\mathcal{H}_{\text{out}}$ ) must be *data-independent* in the sense that it must be selected *before* the training examples are observed. In our transductive setting,  $U$  and  $\mathcal{H}_{\text{out}}$  can be selected *after* the unlabeled full-sample is observed.

The Rademacher bound (5.19), as well as the forthcoming Rademacher bound (5.23), depend on the spectrum of the matrix  $U$ . As we will see in Section 5.5, in non-trivial ULRs of some transductive algorithms (e.g., the algorithms of Zhou et al. (2004) and of Belkin et al. (2004)) the spectrum of  $U$  depends on the spectrum of the Laplacian of the graph used by the algorithm. Thus by transforming the spectrum of Laplacian we control the Rademacher complexity of the hypothesis class. There exists strong empirical evidence (see Chapelle et al., 2003; Joachims, 2003; Johnson & Zhang, 2008) that such spectral transformations improve the performance of the transductive algorithms.

The next lemma (proven in Appendix 5.8.4) shows that for “trivial” ULRs the resulting risk bound is vacuous.

**Lemma 12** *Let  $\alpha \in \mathbb{R}^{m+u}$  be a vector depending on both  $S_m$  and  $X_u$ . Let  $c \in \mathbb{R}$ ,  $U \triangleq c \cdot I$  and  $\mathcal{A}$  be transductive algorithm generating soft-classification vector  $\mathbf{h} = U \cdot \alpha$ . Let  $\mu_1$  be an upper bound on  $\|\alpha\|_2$ , and  $\{\lambda_i\}_{i=1}^k$  be singular values of  $U$ . For the algorithm  $\mathcal{A}$  the bound (5.19) in conjunction with the bound (5.12) is vacuous; namely, for any  $\gamma \in (0, 1)$  and any  $\mathbf{h}$  generated by  $\mathcal{A}$  it holds that*

$$\widehat{\mathcal{L}}_m^\gamma(\mathbf{h}) + \frac{\mu_1}{\gamma} \sqrt{\frac{2}{mu} \sum_{i=1}^k \lambda_i^2} + c_0 Q \sqrt{\min(m, u)} + \sqrt{\frac{S}{2} Q \ln \frac{1}{\delta}} \geq 1 .$$

## 5.4.2 Kernel ULR

If  $r = m + u$  and the matrix  $U$  is a kernel matrix (this holds if  $U$  is positive semidefinite), then we say that the decomposition is a *kernel-ULR*. Let  $\mathcal{G} \subseteq \mathbb{R}^{m+u}$  be the reproducing kernel Hilbert space (RKHS), corresponding to  $U$ . We denote by  $\langle \cdot, \cdot \rangle_{\mathcal{G}}$  the inner product in  $\mathcal{G}$ . Since  $U$  is a kernel matrix, by the reproducing property<sup>6</sup> of  $\mathcal{G}$ ,  $U(i, j) = \langle U(i, \cdot), U(j, \cdot) \rangle_{\mathcal{G}}$ . Suppose that the vector  $\alpha$  satisfies

<sup>6</sup>This means that for all  $\mathbf{h} \in \mathcal{G}$  and  $i \in I_1^{m+u}$ ,  $h(i) = \langle U(i, \cdot), \mathbf{h} \rangle_{\mathcal{G}}$ .

$\sqrt{\boldsymbol{\alpha}^T U \boldsymbol{\alpha}} \leq \mu_2$  for some  $\mu_2$ . Let  $\{\lambda_i\}_{i=1}^{m+u}$  be the eigenvalues of  $U$ . By similar arguments used to derive (5.19) we have:

$$\begin{aligned}
R_{m+u}(\mathcal{H}_{\text{out}}) &= Q \cdot \mathbf{E}_{\boldsymbol{\sigma}} \left\{ \sup_{\mathbf{h} \in \mathcal{H}_{\text{out}}} \sum_{i=1}^{m+u} \sigma_i h(x_i) \right\} \\
&= Q \cdot \mathbf{E}_{\boldsymbol{\sigma}} \left\{ \sup_{\boldsymbol{\alpha}} \sum_{i=1}^{m+u} \sigma_i \sum_{j=1}^{m+u} \alpha_j U(i, j) \right\} \\
&= Q \cdot \mathbf{E}_{\boldsymbol{\sigma}} \left\{ \sup_{\boldsymbol{\alpha}} \sum_{i=1}^{m+u} \sigma_i \sum_{j=1}^{m+u} \alpha_j \langle U(i, \cdot), U(j, \cdot) \rangle_{\mathcal{G}} \right\} \\
&= Q \cdot \mathbf{E}_{\boldsymbol{\sigma}} \left\{ \sup_{\boldsymbol{\alpha}} \left\langle \sum_{i=1}^{m+u} \sigma_i U(i, \cdot), \sum_{j=1}^{m+u} \alpha_j U(j, \cdot) \right\rangle_{\mathcal{G}} \right\} \quad (5.20)
\end{aligned}$$

$$\leq Q \cdot \mathbf{E}_{\boldsymbol{\sigma}} \left\{ \sup_{\boldsymbol{\alpha}} \left\| \sum_{i=1}^{m+u} \sigma_i U(i, \cdot) \right\|_{\mathcal{G}} \cdot \left\| \sum_{j=1}^{m+u} \alpha_j U(j, \cdot) \right\|_{\mathcal{G}} \right\} \quad (5.21)$$

$$\begin{aligned}
&= Q \mu_2 \mathbf{E}_{\boldsymbol{\sigma}} \left\{ \left\| \sum_{i=1}^{m+u} \sigma_i U(i, \cdot) \right\|_{\mathcal{G}} \right\} \\
&= Q \mu_2 \mathbf{E}_{\boldsymbol{\sigma}} \left\{ \sqrt{\left\langle \sum_{i=1}^{m+u} \sigma_i U(i, \cdot), \sum_{j=1}^{m+u} \sigma_j U(j, \cdot) \right\rangle_{\mathcal{G}}} \right\} \\
&= Q \mu_2 \mathbf{E}_{\boldsymbol{\sigma}} \left\{ \sqrt{\sum_{i,j=1}^{m+u} \sigma_i \sigma_j U(i, j)} \right\} \\
&\leq Q \mu_2 \sqrt{\sum_{i,j=1}^{m+u} \mathbf{E}_{\boldsymbol{\sigma}} \{ \sigma_i \sigma_j U(i, j) \}} \quad (5.22)
\end{aligned}$$

$$\begin{aligned}
&= \mu_2 \sqrt{\sum_{i=1}^{m+u} \frac{2}{mu} U(i, i)} = \mu_2 \sqrt{\frac{2 \cdot \text{trace}(U)}{mu}} \\
&= \mu_2 \sqrt{\frac{2}{mu} \sum_{i=1}^{m+u} \lambda_i} . \quad (5.23)
\end{aligned}$$

The inequalities (5.21) and (5.22) are obtained using, respectively, Cauchy-Schwarz and Jensen inequalities. Finally, the first equality in (5.23) follows from the definition of Rademacher variables (see Definition 9).

If transductive algorithm has kernel-ULR then we can use both (5.23) and (5.19) to bound its Rademacher complexity. The kernel bound (5.23) can be

tighter than its non-kernel counterpart (5.19) when the kernel matrix has eigenvalues larger than one and/or  $\mu_2 < \mu_1$ . In Section 5.5 we derive a tight bound on  $\mu_1$  for non-trivial ULRs of “consistency method” of Zhou et al. (2004) and of the Tikhonov regularization method of Belkin et al. (2004).

### 5.4.3 Monte-Carlo Rademacher bounds

We now show how to compute Monte-Carlo Rademacher bounds with high confidence for any transductive algorithm using its ULR. Our empirical examination of these bounds (see Section 5.5.3) shows that they are tighter than the analytical bounds (5.19) and (5.23). The technique, which is based on a simple application of Hoeffding’s inequality, is made particularly simple for vanilla ULRs.

Let  $\mathcal{V} \subseteq \mathbb{R}^{m+u}$  be a set of vectors,  $Q \triangleq \frac{1}{m} + \frac{1}{u}$ ,  $\boldsymbol{\sigma} \in \mathbb{R}^{m+u}$  to be a Rademacher vector (5.1), and  $g(\boldsymbol{\sigma}) = \sup_{\mathbf{v} \in \mathcal{V}} \boldsymbol{\sigma}^T \cdot \mathbf{v}$ . By Definition 9,  $R_{m+u}(\mathcal{V}) = Q \cdot \mathbf{E}_{\boldsymbol{\sigma}}\{g(\boldsymbol{\sigma})\}$ . Let  $\boldsymbol{\sigma}_1, \dots, \boldsymbol{\sigma}_n$  be an i.i.d. sample of Rademacher vectors. We estimate  $R_{m+u}(\mathcal{V})$  with high confidence by applying the Hoeffding inequality on  $\sum_{i=1}^n \frac{1}{n} g(\boldsymbol{\sigma}_i)$ . To apply the Hoeffding inequality we need a bound on  $\sup_{\boldsymbol{\sigma}} |g(\boldsymbol{\sigma})|$ , which is derived for the case where  $\mathcal{V} = \mathcal{H}_{\text{out}}$ . Namely we assume that  $\mathcal{V}$  is a set of all possible outputs of the algorithm (for a fixed  $X_{m+u}$ ). Specifically, suppose that  $\mathbf{v} \in \mathcal{V}$  is an output of the algorithm,  $\mathbf{v} = U\boldsymbol{\alpha}$ , and assume that  $\|\boldsymbol{\alpha}\|_2 \leq \mu_1$ .

By Definition 9, for all  $\boldsymbol{\sigma}$ ,  $\|\boldsymbol{\sigma}\|_2 \leq b \triangleq \sqrt{m+u}$ . Let  $\lambda_1 \leq \dots \leq \lambda_k$  be the singular values of  $U$  and  $\mathbf{u}_1, \dots, \mathbf{u}_k$  and  $\mathbf{w}_1, \dots, \mathbf{w}_k$  be their corresponding unit-length right and left singular vectors<sup>7</sup>. We have that

$$\sup_{\boldsymbol{\sigma}} |g(\boldsymbol{\sigma})| = \sup_{\|\boldsymbol{\sigma}\|_2 \leq b, \|\boldsymbol{\alpha}\|_2 \leq \mu_1} |\boldsymbol{\sigma}^T U \boldsymbol{\alpha}| = \sup_{\|\boldsymbol{\sigma}\|_2 \leq b, \|\boldsymbol{\alpha}\|_2 \leq \mu_1} \left| \boldsymbol{\sigma}^T \sum_{i=1}^k \lambda_i \mathbf{u}_i \mathbf{w}_i^T \boldsymbol{\alpha} \right| \leq b \mu_1 \lambda_k .$$

Applying the one-sided Hoeffding inequality on  $n$  samples of  $g(\boldsymbol{\sigma})$  we have, for any given  $\delta$ , that with probability of at least  $1 - \delta$  over the random i.i.d. choice of the vectors  $\boldsymbol{\sigma}_1, \dots, \boldsymbol{\sigma}_n$ ,

$$R_{m+u}(\mathcal{V}) \leq \left( \frac{1}{m} + \frac{1}{u} \right) \cdot \left( \frac{1}{n} \sum_{i=1}^n \sup_{\boldsymbol{\alpha}: \|\boldsymbol{\alpha}\|_2 \leq \mu_1} \boldsymbol{\sigma}_i^T U \boldsymbol{\alpha} + \mu_1 \lambda_k \sqrt{m+u} \sqrt{\frac{2 \ln \frac{1}{\delta}}{n}} \right) . \quad (5.24)$$

To use the bound (5.24), the value of  $\sup_{\boldsymbol{\alpha}: \|\boldsymbol{\alpha}\|_2 \leq \mu_1} \boldsymbol{\sigma}_i^T U \boldsymbol{\alpha}$  should be computed for each randomly drawn  $\boldsymbol{\sigma}_i$ . This computation is algorithm-dependent and in Section 5.5.3 we show how to compute it for the algorithm of Zhou et al. (2004).<sup>8</sup> In cases where we can compute the supremum exactly (as in vanilla ULRs; see below) we can also get a lower bound using the symmetric Hoeffding inequality.

<sup>7</sup>These vectors can be found from the singular value decomposition of  $U$ .

<sup>8</sup>An application of this approach in induction seems to be very hard, if not impossible. For example, in the case of RBF kernel machines we will need to optimize over (typically) infinite-dimensional vectors in the feature space.

## 5.5 Applications: Explicit bounds for specific algorithms

In this section we exemplify the use of the Rademacher bounds (5.19), (5.23) and (5.24) to particular transductive algorithms. In Section 5.5.1 we instantiate the generic ULR bound (5.19) for the SGT algorithm of Joachims (2003). In Section 5.5.2 we instantiate both the generic ULR bound (5.19) and kernel-ULR bound (5.23) for the algorithm of Belkin et al. (2004). Finally, in Section 5.5.3 we instantiate all three bounds (5.19), (5.23) and (5.24) for the algorithm of Zhou et al. (2004) and compare the resulting bounds numerically.

### 5.5.1 The Spectral Graph Transduction (SGT) algorithm of Joachims (2003)

We start with a description of a simplified version of SGT that captures the essence of the algorithm.<sup>9</sup> Let  $W$  be a symmetric  $(m + u) \times (m + u)$  similarity matrix of the full-sample  $X_{m+u}$ . The  $(i, j)$ th entry of  $W$  represents the similarity between  $x_i$  and  $x_j$ . The matrix  $W$  can be constructed in various ways, for example, it can be a  $k$ -nearest neighbors graph. In such graph each vertex represents example from the full sample  $X_{m+u}$ . There is an edge between a pair of vertices if one of the corresponding examples is among  $k$  most similar examples to the other. The weights of the edges are proportional to the similarity of the adjacent vertices (points). The examples of commonly used measures of similarity are cosine similarity and RBF kernel. Let  $D$  be a diagonal matrix, whose  $(i, i)$ th entry is the sum of the  $i$ th row in  $W$ . An unnormalized Laplacian of  $W$  is  $L = D - W$ .

Let  $\boldsymbol{\tau} = (\tau_1, \dots, \tau_{m+u})$  be a vector that specifies the given labels in  $S_m$ ; that is,  $\tau_i \in \{\pm 1\}$  for labeled points, and  $\tau_i = 0$  otherwise. Let  $c$  be a fixed constant and  $\mathbf{1}$  be an  $(m + u) \times 1$  vector whose entries are 1 and let  $C$  be a diagonal matrix such that  $C(i, i) = 1/m$  iff example  $i$  is in the training set (and zero otherwise). The soft classification  $\mathbf{h}^*$  produced by the SGT algorithm is the solution of the following optimization problem:

$$\min_{\mathbf{h} \in \mathbb{R}^{m+u}} \mathbf{h}^T L \mathbf{h} + c(\mathbf{h} - \bar{\boldsymbol{\tau}})^T C(\mathbf{h} - \bar{\boldsymbol{\tau}}) \quad (5.25)$$

$$s.t. \quad \mathbf{h}^T \mathbf{1} = 0, \quad \mathbf{h}^T \mathbf{h} = m + u. \quad (5.26)$$

It is shown by Joachims (2003) that  $\mathbf{h}^* = U\boldsymbol{\alpha}$ , where  $U$  is an  $(m + u) \times r$  matrix<sup>10</sup> whose columns are orthonormal eigenvectors corresponding to non-zero eigenvalues of the Laplacian  $L$ , and  $\boldsymbol{\alpha}$  is an  $r \times 1$  vector. While  $\boldsymbol{\alpha}$  depends on

<sup>9</sup>We omit a few heuristics that are optional in SGT. Their exclusion does not affect the error bound we derive.

<sup>10</sup> $r$  is the number of non-zero eigenvalues of  $L$ , after performing spectral transformations. Joachims set the default  $r$  to 40.

both the training and test sets, the matrix  $U$  depends only on the unlabeled full-sample. Substituting  $\mathbf{h}^* = U\boldsymbol{\alpha}$  for the second constraint in (5.26) and using the orthonormality of the columns of  $U$ , we get  $m + u = \mathbf{h}^{*T}\mathbf{h}^* = \boldsymbol{\alpha}^T U^T U \boldsymbol{\alpha} = \boldsymbol{\alpha}^T \boldsymbol{\alpha}$ . Hence,  $\|\boldsymbol{\alpha}\|_2 = \sqrt{m + u}$  and we can take  $\mu_1 = \sqrt{m + u}$ . Since  $U$  is an  $(m + u) \times r$  matrix with orthonormal columns,  $\|U\|_{\text{Fro}}^2 = r$ . We conclude from (5.19) the following bound on transductive Rademacher complexity of SGT

$$R_{m+u}(\mathcal{H}_{\text{out}}) \leq \sqrt{2r \left( \frac{1}{m} + \frac{1}{u} \right)}, \quad (5.27)$$

where  $r$  is the number of non-zero eigenvalues of  $L$ . Notice that the bound (5.27) is oblivious to the magnitude of these eigenvalues. With the small value of  $r$  the bound (5.27) is small, but, as shown by Joachims (2003) the test error of SGT is bad. If  $r$  increases then the bound (5.27) increases but the test error improves. Joachims shows empirically that the smallest value of  $r$  achieving nearly optimal test error is 40.

### 5.5.2 Kernel-ULR of the algorithm of Belkin et al. (2004)

By defining the RKHS induced by the graph (unnormalized) Laplacian, as it was done by Herbster et al. (2005), and applying a generalized representer theorem of Schölkopf et al. (2001), we show that the algorithm of Belkin et al. (2004) has a kernel-ULR. Based on this kernel-ULR we derive an explicit risk bound for this. We also derive an explicit risk bound based on generic ULR. We show that the former (kernel) bound is tighter than the latter (generic) one. Finally, we compare our kernel bound with the risk bound of Belkin et al. (2004). The proofs of all lemmas in this section appear in Appendix 5.8.5.

The algorithm of Belkin et al. (2004) is similar to the SGT algorithm, described in Section 5.5.1. Hence in this appendix we use the same notation as in the description of SGT (see Section 5.5.1). The algorithm of Belkin et al. is formulated as follows.

$$\min_{\mathbf{h} \in \mathbb{R}^{m+u}} \quad \mathbf{h}^T L \mathbf{h} + c(\mathbf{h} - \vec{\tau})^T C(\mathbf{h} - \vec{\tau}) \quad (5.28)$$

$$s.t. \quad \mathbf{h}^T \mathbf{1} = 0 \quad (5.29)$$

The difference between (5.28)-(5.29) and (5.25)-(5.26) is in the constraint (5.26), which may change the resulting hard classification. Belkin et al. developed a stability-based error bound for the algorithm based on a connected graph. In the analysis that follows we also assume that the underlying graph is connected, but as shown at the end of this section, the argument can be also extended to unconnected graphs.

We represent a full-sample labeling as a vector in the Reproducing Kernel Hilbert Space (RKHS) associated with the graph Laplacian (as described by

Herbster et al., 2005) and derive a transductive version of the generalized representer theorem of Schölkopf et al. (2001). Considering (5.28)-(5.29) we set  $\mathcal{H} = \{\mathbf{h} \mid \mathbf{h}^T \mathbf{1} = 0, \mathbf{h} \in \mathbb{R}^{m+u}\}$ . Let  $\mathbf{h}_1, \mathbf{h}_2 \in \mathcal{H}$  be two soft classification vectors. We define their inner product as

$$\langle \mathbf{h}_1, \mathbf{h}_2 \rangle_L \triangleq \mathbf{h}_1^T L \mathbf{h}_2 . \quad (5.30)$$

We denote by  $\mathcal{H}_L$  the set  $\mathcal{H}$  along with the inner product (5.30). Let  $\lambda_1, \dots, \lambda_{m+u}$  be the eigenvalues of  $L$  in the increasing order. Since  $L$  is a Laplacian of the connected graph,  $\lambda_1 = 0$  and for all  $2 \leq i \leq m+u$ ,  $\lambda_i \neq 0$ . Let  $\mathbf{u}_i$  be an eigenvector corresponding to  $\lambda_i$ . Since  $L$  is symmetric, the vectors  $\{\mathbf{u}_i\}_{i=1}^{m+u}$  are orthogonal. We assume also w.l.o.g. that the vectors  $\{\mathbf{u}_i\}_{i=1}^{m+u}$  are orthonormal and  $\mathbf{u}_1 = \frac{1}{\sqrt{m+u}} \mathbf{1}$ . Let

$$U \triangleq \sum_{i=2}^{m+u} \frac{1}{\lambda_i} \mathbf{u}_i \mathbf{u}_i^T . \quad (5.31)$$

Note that the matrix  $U$  depends only on the unlabeled full-sample.

**Lemma 13 (Herbster et al., 2005)** *The space  $\mathcal{H}_L$  is an RKHS with a reproducing kernel matrix  $U$ .*

A consequence of Lemma 13 is that the algorithm (5.28)-(5.29) performs the regularization in the RKHS  $\mathcal{H}_L$  with the regularization term  $\|\mathbf{h}\|_L^2 = \mathbf{h}^T L \mathbf{h}$  (this fact was also noted by Herbster et al., 2005). The following transductive variant of the generalized representer theorem (Schölkopf et al., 2001) concludes the derivation of the kernel-ULR of the algorithm of (Belkin et al., 2004).

**Lemma 14** *Let  $\mathbf{h}^* \in \mathcal{H}$  be the solution of the optimization problem (5.28)-(5.29), and let  $U$  be defined as above. Then, there exists  $\boldsymbol{\alpha} \in \mathbb{R}^{m+u}$  such that  $\mathbf{h}^* = U \boldsymbol{\alpha}$ .*

**Remark 8** *We now consider the case of an unconnected graph. Let  $t$  be the number of connected components in the underlying graph. Then the zero eigenvalue of the Laplacian  $L$  has multiplicity  $t$ . Let  $\mathbf{u}_1, \dots, \mathbf{u}_t$  be the eigenvectors corresponding to the zero eigenvalue of  $L$ . Let  $\mathbf{u}_{t+1}, \dots, \mathbf{u}_{m+u}$  be the eigenvectors corresponding to non-zero eigenvalues  $\lambda_{t+1}, \dots, \lambda_{m+u}$  of  $L$ . We replace constraint (5.29) with  $t$  constraints  $\mathbf{h}^T \mathbf{u}_i = 0$  and define the kernel matrix as  $U \triangleq \sum_{i=t+1}^{m+u} \frac{1}{\lambda_i} \mathbf{u}_i \mathbf{u}_i^T$ . The rest of the analysis is the same as for the case of the connected graph.*

To obtain the explicit bounds on the transductive Rademacher complexity of the algorithm of Belkin et al. it remains to bound  $\sqrt{\boldsymbol{\alpha}^T U \boldsymbol{\alpha}}$  and  $\|\boldsymbol{\alpha}\|_2$ . We start with bounding  $\sqrt{\boldsymbol{\alpha}^T U \boldsymbol{\alpha}}$ .

We substitute  $\mathbf{h} = U \boldsymbol{\alpha}$  into (5.28)-(5.29). Since  $\mathbf{u}_2, \dots, \mathbf{u}_{m+u}$  are orthogonal to  $\mathbf{u}_1 = \frac{1}{\sqrt{m+u}} \mathbf{1}$ , we have that  $\mathbf{h}^T \mathbf{1} = \boldsymbol{\alpha}^T U^T \mathbf{1} = \boldsymbol{\alpha}^T \sum_{i=2}^{m+u} \frac{1}{\lambda_i} \mathbf{u}_i \mathbf{u}_i^T \mathbf{1} = 0$ . Moreover, we have that  $\mathbf{h}^T L \mathbf{h} = \boldsymbol{\alpha}^T U^T L U \boldsymbol{\alpha} = \boldsymbol{\alpha}^T \left( I - \frac{1}{m+u} \mathbf{1} \cdot \mathbf{1}^T \right) U \boldsymbol{\alpha} = \boldsymbol{\alpha}^T U \boldsymbol{\alpha}$ . Thus (5.28)-(5.29) is equivalent to solving

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^{m+u}} \boldsymbol{\alpha}^T U \boldsymbol{\alpha} + c(U \boldsymbol{\alpha} - \bar{\tau})^T C(U \boldsymbol{\alpha} - \bar{\tau}) \quad (5.32)$$

and outputting  $\mathbf{h}^* = U\boldsymbol{\alpha}_{\text{out}}$ , where  $\boldsymbol{\alpha}_{\text{out}}$  is the solution of (5.32). Let  $\mathbf{0}$  be the  $(m+u) \times 1$  vector consisting of zeros. We have

$$\begin{aligned} \boldsymbol{\alpha}_{\text{out}}^T U \boldsymbol{\alpha}_{\text{out}} &\leq \boldsymbol{\alpha}_{\text{out}}^T U \boldsymbol{\alpha}_{\text{out}} + c(U\boldsymbol{\alpha}_{\text{out}} - \vec{\tau})^T C(U\boldsymbol{\alpha}_{\text{out}} - \vec{\tau}) \\ &\leq \mathbf{0}^T U \mathbf{0} + c(U\mathbf{0} - \vec{\tau})^T C(U\mathbf{0} - \vec{\tau}) = c . \end{aligned}$$

Thus

$$\sqrt{\boldsymbol{\alpha}_{\text{out}}^T U \boldsymbol{\alpha}_{\text{out}}} \leq \sqrt{c} \triangleq \mu_2 . \quad (5.33)$$

We proceed with the bounding of  $\|\boldsymbol{\alpha}\|_2$ . Let  $\bar{\lambda}_1, \dots, \bar{\lambda}_{m+u}$  be the eigenvalues of  $U$ , sorted in the increasing order. It follows from (5.31) that  $\bar{\lambda}_1 = 0$  and for any  $2 \leq i \leq m+u$ ,  $\bar{\lambda}_i = \frac{1}{\lambda_{m+u-i+2}}$ , where  $\lambda_1, \dots, \lambda_{m+u}$  are the eigenvalues of  $L$  sorted in the increasing order. By the Rayleigh-Ritz theorem (Horn & Johnson, 1990), since  $\boldsymbol{\alpha}_{\text{out}}^T \mathbf{1} = 0$  and  $\mathbf{1}$  is an eigenvector corresponding to  $\bar{\lambda}_1$ , we have that  $\frac{\boldsymbol{\alpha}_{\text{out}}^T U \boldsymbol{\alpha}_{\text{out}}}{\boldsymbol{\alpha}_{\text{out}}^T \boldsymbol{\alpha}_{\text{out}}} \geq \bar{\lambda}_2$ . Therefore

$$\sqrt{\boldsymbol{\alpha}_{\text{out}}^T \boldsymbol{\alpha}_{\text{out}}} \leq \sqrt{\frac{\boldsymbol{\alpha}_{\text{out}}^T U \boldsymbol{\alpha}_{\text{out}}}{\bar{\lambda}_2}} \leq \sqrt{\frac{c}{\bar{\lambda}_2}} \triangleq \mu_1 . \quad (5.34)$$

We substitute the bounds (5.34) and (5.33) into (5.19) and (5.23) respectively, and obtain that the generic bound on the Rademacher complexity is  $\sqrt{\frac{2c}{mu\bar{\lambda}_2} \sum_{i=2}^{m+u} \bar{\lambda}_i^2} = \sqrt{\frac{2c\lambda_{m+u}}{mu} \sum_{i=2}^{m+u} \frac{1}{\lambda_i^2}}$  and the kernel bound is  $\sqrt{\frac{2c}{mu} \sum_{i=2}^{m+u} \bar{\lambda}_i} = \sqrt{\frac{2c}{mu} \sum_{i=2}^{m+u} \frac{1}{\lambda_i}}$ . It is easy to verify that the kernel bound is always tighter than the non-kernel one.

Suppose that<sup>11</sup>  $\sum_{i=2}^{m+u} \frac{1}{\lambda_i} = O(m+u)$ . We substitute the kernel bound into (5.12) and obtain that with probability at least  $1-\delta$  over the random training/test partition,

$$\mathcal{L}_u(\mathbf{h}) \leq \widehat{\mathcal{L}}_m^\gamma(\mathbf{h}) + O\left(\frac{1}{\sqrt{\min(m, u)}}\right) . \quad (5.35)$$

We briefly compare this bound with the risk bound for the algorithm (5.28)-(5.29) given by Belkin et al. (2004). Belkin et al. provide the following bound for their algorithm<sup>12</sup>. With probability of at least  $1-\delta$  over the random draw of  $m$  training examples from  $X_{m+u}$ ,

$$\mathcal{L}_{m+u}(\mathbf{h}) \leq \widehat{\mathcal{L}}_m^\gamma(\mathbf{h}) + O\left(\frac{1}{\sqrt{m}}\right) . \quad (5.36)$$

<sup>11</sup>This assumption is not restricting since we can define the matrix  $L$  and its spectrum after observing the unlabeled full-sample. Thus we can set  $L$  in a way that this assumption will hold.

<sup>12</sup>The original bound of Belkin et al. is in terms of squared loss. The equivalent bound in terms of 0/1 and margin loss can be obtained by the same derivation as in (Belkin et al., 2004).

Similarly to what was done in Section 5.3.3, to bring the bounds to ‘common denominator’, we rewrite the bound (5.35) as

$$\mathcal{L}_u(\mathbf{h}) \leq \widehat{\mathcal{L}}_m^\gamma(\mathbf{h}) + \frac{u}{m+u} O\left(\frac{1}{\sqrt{\min(m, u)}}\right). \quad (5.37)$$

If  $m \ll u$  or  $m \approx u$  then the bounds (5.36) and (5.37) have the same convergence rate. However if  $m \gg u$  then the convergence rate of (5.37) (which is  $O(1/(m\sqrt{u}))$ ) is much faster than the one of (5.36) (which is  $O(1/\sqrt{m})$ ).

### 5.5.3 The Consistency Method of Zhou et al. (2004)

In this section we instantiate the bounds (5.19), (5.23) and (5.24) to the ‘consistency method’ of Zhou et al. (2004) and provide their numerical comparison.

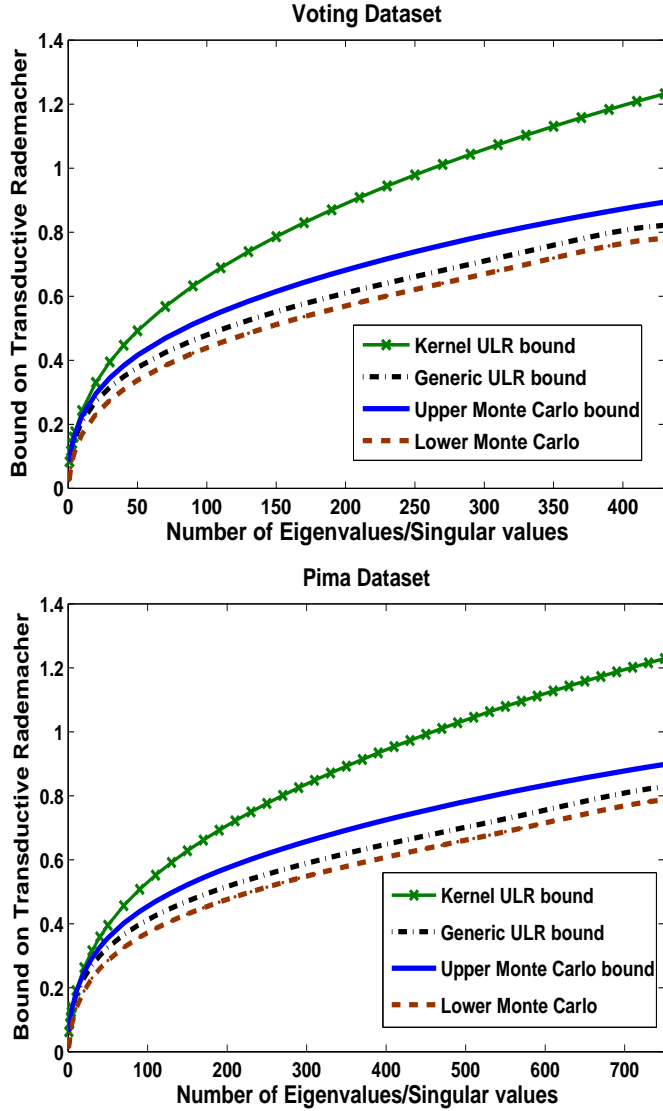
We start with a brief description of the Consistency Method (CM) algorithm of Zhou et al. (2004). The algorithm has a natural vanilla ULR (see definition at the beginning of Section 5.4), where the matrix  $U$  is computed as follows. Let  $W$  and  $D$  be matrices as in SGT (see Section 5.5.1). Let  $L \triangleq D^{-1/2}WD^{-1/2}$  and  $\beta$  be a parameter in  $(0, 1)$ . Then,  $U \triangleq (1 - \beta)(I - \beta L)^{-1}$  and the output of CM is  $\mathbf{h} = U \cdot \boldsymbol{\alpha}$ , where  $\boldsymbol{\alpha}$  specifies the given labels. Consequently  $\|\boldsymbol{\alpha}\|_2 = \sqrt{m}$ . The following lemma, proven in Appendix 5.8.6, provides a characterization of the eigenvalues of  $U$ :

**Lemma 15** *Let  $\lambda_{\max}$  and  $\lambda_{\min}$  be, respectively, the largest and smallest eigenvalues of  $U$ . Then  $\lambda_{\max} = 1$  and  $\lambda_{\min} > 0$ .*

It follows from Lemma 15 that  $U$  is a positive definite matrix and hence is also a kernel matrix. Therefore, the decomposition with the above  $U$  is a kernel-ULR. To apply the kernel bound (5.23) we compute the bound  $\mu_2$  on  $\sqrt{\boldsymbol{\alpha}^T U \boldsymbol{\alpha}}$ . By the Rayleigh-Ritz theorem (Horn & Johnson, 1990), we have that  $\frac{\boldsymbol{\alpha}^T U \boldsymbol{\alpha}}{\boldsymbol{\alpha}^T \boldsymbol{\alpha}} \leq \lambda_{\max}$ . Since by the definition of the vanilla ULR,  $\boldsymbol{\alpha}^T \boldsymbol{\alpha} = m$ , we obtain that  $\sqrt{\boldsymbol{\alpha}^T U \boldsymbol{\alpha}} \leq \sqrt{\lambda_{\max} \boldsymbol{\alpha}^T \boldsymbol{\alpha}} = \sqrt{\lambda_{\max} m}$ .

We obtained that  $\mu_1 = \sqrt{m}$  and  $\mu_2 = \sqrt{\lambda_{\max} m}$ , where  $\lambda_{\max}$  is the maximal eigenvalue of  $U$ . Since by Lemma 15  $\lambda_{\max} = 1$ , for the CM algorithm the bound (5.19) is always tighter than (5.23).

It turns out that for CM, the exact value of the supremum in (5.24) can be analytically derived. Recall that the vectors  $\boldsymbol{\alpha}$ , which induce the CM hypothesis space for a particular  $U$ , have exactly  $m$  components with values in  $\{\pm 1\}$ ; the rest of the components are zeros. Let  $\Psi$  be the set of all possible such  $\boldsymbol{\alpha}$ ’s. Let  $\mathbf{t}(\boldsymbol{\sigma}_i) = (t_1, \dots, t_{m+u}) \triangleq \boldsymbol{\sigma}_i^T U \in \mathbb{R}^{1 \times (m+u)}$  and  $|\mathbf{t}(\boldsymbol{\sigma}_i)| \triangleq (|t_1|, \dots, |t_{m+u}|)$ . Then, for any fixed  $\boldsymbol{\sigma}_i$ ,  $\sup_{\boldsymbol{\alpha} \in \Psi} \boldsymbol{\sigma}_i^T U \boldsymbol{\alpha}$  is the sum of the  $m$  largest elements in  $|\mathbf{t}(\boldsymbol{\sigma}_i)|$ . This derivation holds for any vanilla ULR.



**Figure 5.1:** A comparison of transductive Rademacher bounds.

To demonstrate the Rademacher bounds discussed in this paper we present an empirical comparison of the bounds over two datasets (`Voting`, `Pima`) from the UCI repository<sup>13</sup>. For each dataset we took  $m+u$  to be the size of the dataset (435 and 768 respectively) and we took  $m$  to be  $1/3$  of the full-sample size. The matrix  $W$  is the 10-nearest neighbors graph computed with the cosine similarity metric. We applied the CM algorithm with  $\beta = 0.5$ . The Monte-Carlo bounds (both upper and lower) were computed with  $\delta = 0.05$  and  $n = 10^5$ .

We compared upper and lower Monte-Carlo bounds with the generic ULR

<sup>13</sup>We also obtained similar results for several other UCI datasets.

bound (5.19) and the kernel-ULR bound (5.23). The graphs in Figure 5.1 compare these four bounds for each of the datasets as a function of the number of non-zero eigenvalues of  $U$ . Specifically, each point  $t$  on the  $x$ -axis corresponds to bounds computed with a matrix  $U_t$  that approximates  $U$  using only the smallest  $t$  eigenvalues of  $U$ . In both examples the lower and upper Monte-Carlo bounds tightly “sandwich” the true Rademacher complexity. It is striking that generic-ULR bound is very close to the true Rademacher complexity. In principle, with our simple Monte-Carlo method we can approximate the true Rademacher complexity up to any desired accuracy (with high confidence) at the cost of drawing sufficiently many Rademacher vectors.

## 5.6 PAC-Bayesian bound for transductive mixtures

In this section we adapt part of the results of Meir and Zhang (2003) to transduction. The proofs of all results presented in this section appear in Appendix 5.8.7.

Let  $\mathcal{B} = \{\mathbf{h}_i\}_{i=1}^{|\mathcal{B}|}$  be a finite set of *base-hypotheses*. The class  $\mathcal{B}$  can be formed after observing the full-sample  $X_{m+u}$ , but before obtaining the training/test set partition and the labels. Let  $\mathbf{q} = (q_1, \dots, q_{|\mathcal{B}|}) \in \mathbb{R}^{|\mathcal{B}|}$  be a probability vector, i.e.  $\sum_{i=1}^{|\mathcal{B}|} q_i = 1$  and  $q_i \geq 0$  for all  $1 \leq i \leq |\mathcal{B}|$ . The vector  $\mathbf{q}$  can be computed after observing training/test partition and the training labels. Our goal is to find the “posterior” vector  $\mathbf{q}$  such that the *mixture hypothesis*,  $\tilde{\mathbf{h}}_{\mathbf{q}} \triangleq \sum_{i=1}^{|\mathcal{B}|} q_i \mathbf{h}_i$  minimizing  $\mathcal{L}_u(\tilde{\mathbf{h}}_{\mathbf{q}}) = \frac{1}{u} \sum_{j=m+1}^{m+u} \ell\left(\sum_{i=1}^{|\mathcal{B}|} q_i h_i(j), y_j\right)$ .

In this section we derive a uniform risk bound for a set of  $\mathbf{q}$ ’s. This bound depends on the KL-divergence (see the definition below) between  $\mathbf{q}$  and the “prior” probability vector  $\mathbf{p} \in \mathbb{R}^{|\mathcal{B}|}$ , where the vector  $\mathbf{p}$  is defined based only on the unlabeled full-sample. Thus our forthcoming bound (see Theorem 8) belongs to the family of PAC-Bayesian bounds (e.g., see McAllester, 2003; Derbeko et al., 2004), which depend on prior and posterior information. Notice that our bound is different from the PAC-Bayesian bounds for Gibbs classifiers that minimize  $\mathbf{E}_{\mathbf{h} \sim \mathcal{B}(\mathbf{q})} \mathcal{L}_u(\mathbf{h}) = \frac{1}{u} \sum_{j=m+1}^{m+u} \mathbf{E}_{\mathbf{h} \sim \mathcal{B}(\mathbf{q})} \ell(h(j), y_j)$ , where  $\mathbf{h} \sim \mathcal{B}(\mathbf{q})$  is a random draw of the base hypothesis from  $\mathcal{B}$  according to distribution  $\mathbf{q}$ .

We assume that  $\mathbf{q}$  belongs to a domain  $\Omega_{g,A} = \{\mathbf{q} \mid g(\mathbf{q}) \leq A\}$ , where  $g : \mathbb{R}^{|\mathcal{B}|} \rightarrow \mathbb{R}$  is a predefined function and  $A \in \mathbb{R}$  is a constant. The domain  $\Omega_{g,A}$  and the set  $\mathcal{B}$  induce the class  $\tilde{\mathcal{B}}_{g,A}$  of all possible mixtures  $\tilde{\mathbf{h}}_{\mathbf{q}}$ . Recalling that  $Q \triangleq (1/m + 1/u)$ ,  $S \triangleq \frac{m+u}{(m+u-0.5)(1-0.5/\max(m,u))}$  and  $c_0 = \sqrt{32 \ln(4e)}/3 < 5.05$ , we apply Theorem 6 with  $\mathcal{H}_{\text{out}} \triangleq \tilde{\mathcal{B}}_{g,A}$  and obtain that with probability of at least  $1 - \delta$  over the training/test partition of  $X_{m+u}$ , for all  $\tilde{\mathbf{h}}_{\mathbf{q}} \in \tilde{\mathcal{B}}_{g,A}$ ,

$$\mathcal{L}_u(\tilde{\mathbf{h}}_{\mathbf{q}}) \leq \hat{\mathcal{L}}_m^\gamma(\tilde{\mathbf{h}}_{\mathbf{q}}) + \frac{R_{m+u}(\tilde{\mathcal{B}}_{g,A})}{\gamma} + c_0 Q \sqrt{\min(m, u)} + \sqrt{\frac{S}{2} Q \ln \frac{1}{\delta}}. \quad (5.38)$$

Let  $Q_1 \triangleq \sqrt{\frac{s}{2}Q (\ln(1/\delta) + 2 \ln \log_s (s\tilde{g}(\mathbf{q})/g_0))}$ . It is straightforward to apply the technique used in the proof of Theorem 10 of Meir and Zhang (2003) and obtain the following bound, which eliminates the dependence on  $A$ .

**Corollary 2** *Let  $g_0 > 0$ ,  $s > 1$  and  $\tilde{g}(\mathbf{q}) = s \max(g(\mathbf{q}), g_0)$ . For any fixed  $g$  and  $\gamma > 0$ , with probability of at least  $1 - \delta$  over the training/test set partition, for all<sup>14</sup>  $\tilde{\mathbf{h}}_{\mathbf{q}}$ ,*

$$\mathcal{L}_u(\tilde{\mathbf{h}}_{\mathbf{q}}) \leq \widehat{\mathcal{L}}_m^\gamma(\tilde{\mathbf{h}}_{\mathbf{q}}) + \frac{R_{m+u}(\tilde{\mathcal{B}}_{g,\tilde{g}(\mathbf{q})})}{\gamma} + c_0 Q \sqrt{\min(m, u)} + Q_1 . \quad (5.39)$$

We now instantiate Corollary 2 for  $g(\mathbf{q})$  being the KL-divergence and derive a PAC-Bayesian bound. Let  $g(\mathbf{q}) \triangleq D(\mathbf{q}||\mathbf{p}) = \sum_{i=1}^{|\mathcal{B}|} q_i \ln \left( \frac{q_i}{p_i} \right)$  be KL-divergence between  $\mathbf{p}$  and  $\mathbf{q}$ . Adopting Lemma 11 of Meir and Zhang (2003) to the transductive Rademacher variables, defined in (5.1), we obtain the following bound.

**Theorem 8** *Let  $g_0 > 0$ ,  $s > 1$ ,  $\gamma > 0$ . Let  $\mathbf{p}$  and  $\mathbf{q}$  be any prior and posterior distribution over  $\mathcal{B}$ , respectively. Set  $g(\mathbf{q}) \triangleq D(\mathbf{q}||\mathbf{p})$  and  $\tilde{g}(\mathbf{q}) \triangleq s \max(g(\mathbf{q}), g_0)$ . Then, with probability of at least  $1 - \delta$  over the training/test set partition, for all  $\tilde{\mathbf{h}}_{\mathbf{q}}$ ,*

$$\mathcal{L}_u(\tilde{\mathbf{h}}_{\mathbf{q}}) \leq \widehat{\mathcal{L}}_m^\gamma(\tilde{\mathbf{h}}_{\mathbf{q}}) + \frac{Q}{\gamma} \sqrt{2\tilde{g}(\mathbf{q}) \sup_{\mathbf{h} \in \mathcal{B}} \|\mathbf{h}\|_2^2} + c_0 Q \sqrt{\min(m, u)} + Q_1 . \quad (5.40)$$

Theorem 8 is a PAC-Bayesian result, where the prior  $\mathbf{p}$  can depend on  $X_{m+u}$  and the posterior can be optimized adaptively, based also on  $S_m$ . As our general bound (5.12), the bound (5.40) has the convergence rate of  $O(1/\sqrt{\min(m, u)})$ . The bound (5.40) is syntactically similar to inductive PAC-Bayesian bound for mixture hypothesis (see Theorem 10 and Lemma 11 in Meir & Zhang, 2003), having similar convergence rate of  $O(1/\sqrt{m})$ . However the conceptual difference between inductive and transductive bounds is that in transduction we can define the prior vector  $\mathbf{p}$  after observing the unlabeled full-sample and in induction we should define  $\mathbf{p}$  before observing any data.

## 5.7 Concluding remarks

We studied the use of Rademacher complexity analysis in the transductive setting. Our results include the first general Rademacher bound for soft classification algorithms, the unlabeled-labeled representation (ULR) technique for bounding the Rademacher complexity of any transductive algorithm and a bound for Bayesian

---

<sup>14</sup>In the bound (5.39) the meaning of  $R_{m+u}(\tilde{\mathcal{B}}_{g,\tilde{g}(\mathbf{q})})$  is as follows: for any  $\mathbf{q}$ , let  $A = \tilde{g}(\mathbf{q})$  and  $R_{m+u}(\tilde{\mathcal{B}}_{g,\tilde{g}(\mathbf{q})}) \triangleq R_{m+u}(\tilde{\mathcal{B}}_{g,A})$ .

mixtures. We demonstrated the usefulness of these results and, in particular, the effectiveness of our ULR framework for deriving error bounds for several advanced transductive algorithms.

It would be nice to further improve our bounds using, for example, the local Rademacher approach by Bartlett et al. (2005). However, we believe that the main advantage of these transductive bounds is the possibility of selecting a hypothesis space based on a full-sample. A clever data-dependent choice of this space should provide sufficient flexibility to achieve a low training error with low Rademacher complexity. In our opinion this opportunity can be explored and exploited much further.

It would be interesting to optimize the matrix  $U$  in the ULR explicitly (to fit the data) under a constraint of low Rademacher complexity. Also, it would be nice to find “low-Rademacher” approximations of particular  $U$  matrices. The PAC-Bayesian bound for mixture algorithms motivates the development and use of transductive mixtures, an area that has yet to be investigated.

## 5.8 Proofs

### 5.8.1 Proof of Lemma 9

The proof is based on the technique used in the proof of Lemma 5 in Meir and Zhang (2003). Let  $\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_{m+u})$  be the Rademacher random variables of  $R_{m+u}(\mathcal{V}, p_1)$  and  $\boldsymbol{\tau} = (\tau_1, \dots, \tau_{m+u})$  be the Rademacher random variables of  $R_{m+u}(\mathcal{V}, p_2)$ . Denote by  $I_r^s$  the set of natural numbers  $\{r, \dots, s\}$  ( $r < s$ ). For any real-valued function  $g(\mathbf{v})$ , for any  $n \in I_1^{m+u}$  and any  $\mathbf{v}' \in \mathcal{V}$ ,

$$\sup_{\mathbf{v} \in \mathcal{V}} [g(\mathbf{v})] = \mathbf{E}_{\tau_n} \left\{ \tau_n v'_n + \sup_{\mathbf{v} \in \mathcal{V}} [g(\mathbf{v})] \mid \tau_n \neq 0 \right\} \leq \mathbf{E}_{\tau_n} \left\{ \sup_{\mathbf{v} \in \mathcal{V}} [\tau_n v_n + g(\mathbf{v})] \mid \tau_n \neq 0 \right\}. \quad (5.41)$$

We use the abbreviation  $\tau_1^s \triangleq \tau_1, \dots, \tau_s$ . We apply (5.41) with a fixed  $\tau_1^{n-1}$  and  $g(\mathbf{v}) \triangleq f(\mathbf{v}) + \sum_{i=1}^{n-1} \tau_i v_i$ , and obtain that

$$\sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^{n-1} \tau_i v_i + f(\mathbf{v}) \right] \leq \mathbf{E}_{\tau_n} \left\{ \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^n \tau_i v_i + f(\mathbf{v}) \right] \mid \tau_n \neq 0 \right\}. \quad (5.42)$$

To complete the proof of the lemma, we prove a more general claim: for any real-valued function  $f(\mathbf{v})$ , for any  $0 \leq n \leq m+u$ ,

$$\mathbf{E}_{\boldsymbol{\sigma}} \left\{ \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^n \sigma_i v_i + f(\mathbf{v}) \right] \right\} \leq \mathbf{E}_{\boldsymbol{\tau}} \left\{ \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^n \tau_i v_i + f(\mathbf{v}) \right] \right\}. \quad (5.43)$$

The proof is by induction on  $n$ . The claim trivially holds for  $n = 0$  (in this case (5.43) holds with equality). Suppose the claim holds for all  $k < n$ . We use the

abbreviation  $\sigma_1^s \triangleq \sigma_1, \dots, \sigma_s$ . We have

$$\begin{aligned}
& \mathbf{E}_{\sigma_1^n} \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^n \sigma_i v_i + f(\mathbf{v}) \right] \\
= & \left. 2p_1 \left\{ \frac{1}{2} \mathbf{E}_{\sigma_1^{n-1}} \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^{n-1} \sigma_i v_i + v_n + f(\mathbf{v}) \right] + \frac{1}{2} \mathbf{E}_{\sigma_1^{n-1}} \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^{n-1} \sigma_i v_i - v_n + f(\mathbf{v}) \right] \right\} \right. \\
& \left. + (1 - 2p_1) \mathbf{E}_{\sigma_1^{n-1}} \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^{n-1} \sigma_i v_i + f(\mathbf{v}) \right] \right. \\
\leq & 2p_1 \left\{ \frac{1}{2} \mathbf{E}_{\tau_1^{n-1}} \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^{n-1} \tau_i v_i + v_n + f(\mathbf{v}) \right] \right. \\
& \left. + \frac{1}{2} \mathbf{E}_{\tau_1^{n-1}} \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^{n-1} \tau_i v_i - v_n + f(\mathbf{v}) \right] \right\} + (1 - 2p_1) \mathbf{E}_{\tau_1^{n-1}} \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^{n-1} \tau_i v_i + f(\mathbf{v}) \right] \\
= & \mathbf{E}_{\tau_1^{n-1}} \left\{ 2p_1 \mathbf{E}_{\tau_n} \left\{ \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^n \tau_i v_i + f(\mathbf{v}) \right] \mid \tau_n \neq 0 \right\} + (1 - 2p_1) \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^{n-1} \tau_i v_i + f(\mathbf{v}) \right] \right\} \\
= & \mathbf{E}_{\tau_1^{n-1}} \left\{ 2p_1 \left( \mathbf{E}_{\tau_n} \left\{ \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^n \tau_i v_i + f(\mathbf{v}) \right] \mid \tau_n \neq 0 \right\} - \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^{n-1} \tau_i v_i + f(\mathbf{v}) \right] \right) \right. \\
& \left. + \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^{n-1} \tau_i v_i + f(\mathbf{v}) \right] \right\} \\
\leq & \mathbf{E}_{\tau_1^{n-1}} \left\{ 2p_2 \left( \mathbf{E}_{\tau_n} \left\{ \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^n \tau_i v_i + f(\mathbf{v}) \right] \mid \tau_n \neq 0 \right\} - \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^{n-1} \tau_i v_i + f(\mathbf{v}) \right] \right) \right. \\
& \left. + \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^{n-1} \tau_i v_i + f(\mathbf{v}) \right] \right\} \\
= & \mathbf{E}_{\tau_1^{n-1}} \left\{ 2p_2 \mathbf{E}_{\tau_n} \left\{ \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^n \tau_i v_i + f(\mathbf{v}) \right] \mid \tau_n \neq 0 \right\} \right. \\
& \left. + (1 - 2p_2) \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^{n-1} \tau_i v_i + f(\mathbf{v}) \right] \right\} \\
= & \mathbf{E}_{\tau_1^n} \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^n \tau_i v_i + f(\mathbf{v}) \right].
\end{aligned} \tag{5.44}$$

$$\begin{aligned}
& \mathbf{E}_{\tau_1^{n-1}} \left\{ 2p_1 \mathbf{E}_{\tau_n} \left\{ \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^n \tau_i v_i + f(\mathbf{v}) \right] \mid \tau_n \neq 0 \right\} + (1 - 2p_1) \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^{n-1} \tau_i v_i + f(\mathbf{v}) \right] \right\} \\
& \mathbf{E}_{\tau_1^{n-1}} \left\{ 2p_1 \left( \mathbf{E}_{\tau_n} \left\{ \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^n \tau_i v_i + f(\mathbf{v}) \right] \mid \tau_n \neq 0 \right\} - \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^{n-1} \tau_i v_i + f(\mathbf{v}) \right] \right) \right. \\
& \left. + \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^{n-1} \tau_i v_i + f(\mathbf{v}) \right] \right\} \\
& \mathbf{E}_{\tau_1^{n-1}} \left\{ 2p_2 \left( \mathbf{E}_{\tau_n} \left\{ \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^n \tau_i v_i + f(\mathbf{v}) \right] \mid \tau_n \neq 0 \right\} - \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^{n-1} \tau_i v_i + f(\mathbf{v}) \right] \right) \right. \\
& \left. + \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^{n-1} \tau_i v_i + f(\mathbf{v}) \right] \right\} \\
& \mathbf{E}_{\tau_1^{n-1}} \left\{ 2p_2 \mathbf{E}_{\tau_n} \left\{ \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^n \tau_i v_i + f(\mathbf{v}) \right] \mid \tau_n \neq 0 \right\} \right. \\
& \left. + (1 - 2p_2) \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^{n-1} \tau_i v_i + f(\mathbf{v}) \right] \right\} \\
& \mathbf{E}_{\tau_1^n} \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^n \tau_i v_i + f(\mathbf{v}) \right].
\end{aligned} \tag{5.45}$$

The inequality (5.44) follow from the inductive hypothesis. The inequality (5.45) follows from (5.42) and the fact that  $p_1 < p_2$ .

## 5.8.2 Proof of Lemma 10

For technical convenience we use the following definition of pairwise Rademacher variables.

**Definition 11 (Pairwise Rademacher variables)** Let  $\mathbf{v} = (v(1), \dots, v(m+u)) \in \mathbb{R}^{m+u}$ . Let  $\mathcal{V}$  be a set of vectors from  $\mathbb{R}^{m+u}$ . Let  $\tilde{\boldsymbol{\sigma}} = \{\tilde{\sigma}_i\}_{i=1}^{m+u}$  be a vector of i.i.d. random variables defined as:

$$\tilde{\sigma}_i = (\tilde{\sigma}_{i,1}, \tilde{\sigma}_{i,2}) = \begin{cases} \left(-\frac{1}{m}, -\frac{1}{u}\right) & \text{with probability } \frac{mu}{(m+u)^2} ; \\ \left(-\frac{1}{m}, \frac{1}{m}\right) & \text{with probability } \frac{m^2}{(m+u)^2} ; \\ \left(\frac{1}{u}, \frac{1}{m}\right) & \text{with probability } \frac{mu}{(m+u)^2} ; \\ \left(\frac{1}{u}, -\frac{1}{u}\right) & \text{with probability } \frac{u^2}{(m+u)^2} . \end{cases} \quad (5.46)$$

We obtain Definition 11 from Definition 9 (with  $p = \frac{mu}{(m+u)^2}$ ) in the following way. If the Rademacher variable  $\sigma_i = 1$  then we split it to  $\tilde{\sigma}_i = \left(\frac{1}{u}, \frac{1}{m}\right)$ . If the Rademacher variable  $\sigma_i = -1$  then we split it to  $\tilde{\sigma}_i = \left(-\frac{1}{m}, -\frac{1}{u}\right)$ . If the Rademacher variable  $\sigma_i = 0$  then we split it randomly to  $\left(-\frac{1}{m}, \frac{1}{m}\right)$  or  $\left(\frac{1}{u}, -\frac{1}{u}\right)$ . The first component of  $\tilde{\sigma}_i$  indicates if the  $i$ th component of  $\mathbf{v}$  is in the first elements of  $\mathbf{v}(\mathbf{Z})$  or in the last  $u$  elements of  $\mathbf{v}(\mathbf{Z})$ . If the former case the value of  $\tilde{\sigma}_i$  is  $-\frac{1}{m}$  and in the latter case the value of  $\tilde{\sigma}_i$  is  $\frac{1}{u}$ . The second component of  $\tilde{\sigma}_i$  has the same meaning as the first one, but with  $\mathbf{Z}$  replaced by  $\mathbf{Z}'$ .

The values  $\pm\frac{1}{m}$  and  $\pm\frac{1}{u}$  are exactly the coefficients appearing inside  $\mathbf{T}\{\mathbf{v}(\mathbf{Z})\}$ ,  $\mathbf{T}\{\mathbf{v}(\mathbf{Z}')\}$ ,  $\mathbf{H}\{\mathbf{v}(\mathbf{Z}')\}$  and  $\mathbf{H}\{\mathbf{v}(\mathbf{Z})\}$  in (5.4). These coefficients are random and their distribution is induced by the uniform distribution over permutations. In the course of the proof we will establish the precise relation between the distribution of  $\pm\frac{1}{m}$  and  $\pm\frac{1}{u}$  coefficients and the distribution (5.46) of pairwise Rademacher variables.

It is easy to verify that

$$R_{m+u}(\mathcal{V}) = \mathbf{E}_{\tilde{\boldsymbol{\sigma}}} \left\{ \sup_{\mathbf{v} \in \mathcal{V}} \sum_{i=1}^{m+u} (\tilde{\sigma}_{i,1} + \tilde{\sigma}_{i,2}) v(i) \right\}. \quad (5.47)$$

Let  $n_1, n_2$  and  $n_3$  be the number of random variables  $\tilde{\sigma}_i$  realizing the value  $\left(-\frac{1}{m}, -\frac{1}{u}\right)$ ,  $\left(-\frac{1}{m}, \frac{1}{m}\right)$ ,  $\left(\frac{1}{u}, \frac{1}{m}\right)$ , respectively. Set  $N_1 \triangleq n_1 + n_2$  and  $N_2 \triangleq n_2 + n_3$ . Note that the  $n_i$ 's and  $N_i$ 's are random variables. Denote by  $\mathbf{Rad}$  the distribution of  $\tilde{\boldsymbol{\sigma}}$  defined by (5.46) and by  $\mathbf{Rad}(N_1, N_2)$ , the distribution  $\mathbf{Rad}$  conditioned on the events  $n_1 + n_2 = N_1$  and  $n_2 + n_3 = N_2$ . We define

$$s(N_1, N_2) \triangleq \mathbf{E}_{\tilde{\boldsymbol{\sigma}} \sim \mathbf{Rad}(N_1, N_2)} \left\{ \sup_{\mathbf{v} \in \mathcal{V}} \sum_{i=1}^{m+u} (\tilde{\sigma}_{i,1} + \tilde{\sigma}_{i,2}) v(i) \right\}. \quad (5.48)$$

The rest of the proof is based on the following three claims:

**Claim 1.**  $R_{m+u}(\mathcal{V}) = \mathbf{E}_{N_1, N_2} \{s(N_1, N_2)\}$ .

**Claim 2.**  $\mathbf{E}_{\mathbf{Z}} \{g(\mathbf{Z})\} = s(\mathbf{E}_{\tilde{\sigma}} N_1, \mathbf{E}_{\tilde{\sigma}} N_2)$ .

**Claim 3.**  $s(\mathbf{E}_{\tilde{\sigma}} N_1, \mathbf{E}_{\tilde{\sigma}} N_2) - \mathbf{E}_{N_1, N_2} \{s(N_1, N_2)\} \leq c_0 B_{\max} \left(\frac{1}{u} + \frac{1}{m}\right) \sqrt{m}$ .

Having established these three claims we immediately obtain

$$\mathbf{E}_{\mathbf{Z}} \{g(\mathbf{Z})\} \leq \tilde{R}_{m+u}(\mathcal{V}) + c_0 B_{\max} \left(\frac{1}{u} + \frac{1}{m}\right) \sqrt{m} . \quad (5.49)$$

The entire development is symmetric in  $m$  and  $u$  and, therefore, we also obtain the same result but with  $\sqrt{u}$  instead of  $\sqrt{m}$ . By taking the minimum of (5.49) and the symmetric bound (with  $\sqrt{u}$ ) we establish the theorem. It remains to prove the three claims.

**Proof of Claim 1.** Note that  $N_1$  and  $N_2$  are random variables whose distribution is induced by the distribution of  $\tilde{\sigma}$ . We have by (5.47) that

$$\tilde{R}_{m+u}(\mathcal{V}) = \mathbf{E}_{N_1, N_2} \mathbf{E}_{\tilde{\sigma} \sim \text{Rad}(N_1, N_2)} \sup_{\mathbf{v} \in \mathcal{V}} \sum_{i=1}^{m+u} (\tilde{\sigma}_{i,1} + \tilde{\sigma}_{i,2}) v(i) = \mathbf{E}_{N_1, N_2} s(N_1, N_2) .$$

**Proof of Claim 2.** By the definitions of  $\mathbf{H}_k$  and  $\mathbf{T}_k$  (appearing at the start of Section 5.3.1), for any  $N_1, N_2 \in I_1^{m+u}$  we have

$$\begin{aligned} & \mathbf{E}_{\mathbf{Z}, \mathbf{Z}'} \sup_{\mathbf{v} \in \mathcal{V}} \left[ \mathbf{T}_{N_1} \{\mathbf{v}(\mathbf{Z})\} - \mathbf{T}_{N_2} \{\mathbf{v}(\mathbf{Z}')\} + \mathbf{H}_{N_2} \{\mathbf{v}(\mathbf{Z}')\} - \mathbf{H}_{N_1} \{\mathbf{v}(\mathbf{Z})\} \right] = \\ & \mathbf{E}_{\mathbf{Z}, \mathbf{Z}'} \sup_{\mathbf{v} \in \mathcal{V}} \underbrace{\left[ \frac{1}{u} \sum_{i=N_1+1}^{m+u} v(Z_i) - \frac{1}{u} \sum_{i=N_2+1}^{m+u} v(Z'_i) + \frac{1}{m} \sum_{i=1}^{N_2} v(Z'_i) - \frac{1}{m} \sum_{i=1}^{N_1} v(Z_i) \right]}_{\triangleq r(\mathbf{v}, \mathbf{Z}, \mathbf{Z}', N_1, N_2)} . \end{aligned} \quad (5.50)$$

The values of  $N_1$  and  $N_2$ , and the distribution of  $\mathbf{Z}$  and  $\mathbf{Z}'$ , with respect to which we take the expectation in (5.50), induce a distribution of assignments of coefficients  $\left\{\frac{1}{m}, -\frac{1}{m}, \frac{1}{u}, -\frac{1}{u}\right\}$  to the components of  $\mathbf{v}$ . For any  $N_1, N_2$  and realizations of  $\mathbf{Z}$  and  $\mathbf{Z}'$ , each component  $v(i)$ ,  $i \in I_1^{m+u}$ , is assigned to exactly two coefficients, one for each of the two permutations ( $\mathbf{Z}$  and  $\mathbf{Z}'$ ). Let  $\mathbf{a} \triangleq (a_1, \dots, a_{m+u})$ , where  $a_i \triangleq (a_{i,1}, a_{i,2})$  is a pair of coefficients. For any  $i \in I_1^{m+u}$ , the pair  $(a_{i,1}, a_{i,2})$  takes the values of the coefficients of  $v(i)$ , where the first component is induced by the realization  $\mathbf{Z}$  (i.e.,  $a_{i,1}$  is either  $-\frac{1}{m}$  or  $\frac{1}{u}$ ) and the second component by the realization of  $\mathbf{Z}'$  (i.e.,  $a_{i,2}$  is either  $\frac{1}{m}$  or  $-\frac{1}{u}$ ).

Let  $\mathbf{A}(N_1, N_2)$  be the distribution of vectors  $\mathbf{a}$ , induced by the distribution of  $\mathbf{Z}$  and  $\mathbf{Z}'$ , for particular  $N_1, N_2$ . Using this definition we can write (5.50) as

$$\mathbf{E}_{\mathbf{a} \sim \mathbf{A}(N_1, N_2)} \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^{m+u} (a_{i,1} + a_{i,2}) v(i) \right] . \quad (5.51)$$

Let  $\text{Par}(k)$  be the uniform distribution over partitions of  $m + u$  elements into two subsets, of  $k$  and  $m + u - k$  elements, respectively. Clearly,  $\text{Par}(k)$  is a uniform distribution over  $\binom{m+u}{k}$  elements. The distribution of the random vector  $(a_{1,1}, a_{2,1}, \dots, a_{m+u,1})$  of the first elements of pairs in  $\mathbf{a}$  is equivalent to  $\text{Par}(N_1)$ . That is, this vector is obtained by taking the first  $N_1$  indices of the realization of  $\mathbf{Z}$  and assigning  $-\frac{1}{m}$  to the corresponding components. The other components are assigned to  $\frac{1}{u}$ . Similarly, the distribution of the random vector  $(a_{1,2}, a_{2,2}, \dots, a_{m+u,2})$  is equivalent to  $\text{Par}(N_2)$ . Therefore, the distribution  $\mathbf{A}(N_1, N_2)$  of the entire vector  $\mathbf{a}$  is equivalent to the product distribution of  $\text{Par}(N_1)$  and  $\text{Par}(N_2)$ , which is a uniform distribution over  $\binom{m+u}{N_1} \cdot \binom{m+u}{N_2}$  elements, where each element is a pair of independent permutations.

We show that the distributions  $\text{Rad}(N_1, N_2)$  and  $\mathbf{A}(N_1, N_2)$  are identical. Given  $N_1$  and  $N_2$  and setting  $\omega = (m + u)^2$ , the probability of drawing a specific realization of  $\tilde{\sigma}$  (satisfying  $n_1 + n_2 = N_1$  and  $n_2 + n_3 = N_2$ ) is

$$\left(\frac{m^2}{\omega}\right)^{n_2} \left(\frac{mu}{\omega}\right)^{N_1-n_2} \left(\frac{mu}{\omega}\right)^{N_2-n_2} \left(\frac{u^2}{\omega}\right)^{m+u-N_1-N_2+n_2} = \frac{m^{N_1+N_2} u^{2(m+u)-N_1-N_2}}{(m+u)^{2(m+u)}}. \quad (5.52)$$

Since (5.52) is independent of the  $n_i$ 's, the distribution  $\text{Rad}(N_1, N_2)$  is uniform over all possible Rademacher assignments satisfying the constraints  $N_1$  and  $N_2$ . It is easy to see that the support size of  $\text{Rad}(N_1, N_2)$  is the same as the support size of  $\mathbf{A}(N_1, N_2)$ . Moreover, the support sets of these distributions are identical; hence these distributions are identical. Therefore, it follows from (5.51) that (5.50) is equal to

$$\mathbf{E}_{\tilde{\sigma} \sim \text{Rad}(N_1, N_2)} \left\{ \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^{m+u} (\tilde{\sigma}_{i,1} + \tilde{\sigma}_{i,2}) v(i) \right] \right\} = s(N_1, N_2). \quad (5.53)$$

It is easy to see that  $\mathbf{E}_{\tilde{\sigma}} N_1 = \mathbf{E}_{\tilde{\sigma}} \{n_1 + n_2\} = m$  and that  $\mathbf{E}_{\tilde{\sigma}} N_2 = \mathbf{E}_{\tilde{\sigma}} \{n_2 + n_3\} = m$ . Since  $\mathbf{E}_{\mathbf{Z}} \{g(\mathbf{Z})\}$  is (5.50) with  $N_1 = m$  and  $N_2 = m$ , we have

$$\mathbf{E}_{\mathbf{Z}} \{g(\mathbf{Z})\} = \mathbf{E}_{\tilde{\sigma} \sim \text{Rad}(m, m)} \left\{ \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^{m+u} (\tilde{\sigma}_{i,1} + \tilde{\sigma}_{i,2}) v(i) \right] \right\} = s(\mathbf{E}_{\tilde{\sigma}} N_1, \mathbf{E}_{\tilde{\sigma}} N_2).$$

### Proof of Claim 3.

We bound the differences  $|s(N_1, N_2) - s(N'_1, N_2)|$  and  $|s(N_1, N_2) - s(N_1, N'_2)|$  for any  $1 \leq N_1, N_2, N'_1, N'_2 \leq m + u$ . Suppose w.l.o.g. that  $N'_1 \leq N_1$ . Recalling

the definition of  $r(\cdot)$  in (5.50) we have

$$s(N_1, N_2) = \mathbf{E}_{\mathbf{Z}, \mathbf{Z}'} \sup_{\mathbf{v} \in \mathcal{V}} \left[ r(\mathbf{v}, \mathbf{Z}, \mathbf{Z}', N_1, N_2) \right] \quad (5.54)$$

$$s(N'_1, N_2) = \mathbf{E}_{\mathbf{Z}, \mathbf{Z}'} \sup_{\mathbf{v} \in \mathcal{V}} \left[ r(\mathbf{v}, \mathbf{Z}, \mathbf{Z}', N_1, N_2) + \left( \frac{1}{u} + \frac{1}{m} \right) \sum_{i=N'_1+1}^{N_1} v(Z_i) \right] \quad (5.55)$$

The expressions under the supremums in  $s(N_1, N_2)$  and  $s(N'_1, N_2)$  differ only in the two terms in (5.55). Therefore, for any  $N_1$  and  $N'_1$ ,

$$|s(N_1, N_2) - s(N'_1, N_2)| \leq B_{\max} |N_1 - N'_1| \left( \frac{1}{u} + \frac{1}{m} \right). \quad (5.56)$$

Similarly we have that for any  $N_2$  and  $N'_2$ ,

$$|s(N_1, N_2) - s(N_1, N'_2)| \leq B_{\max} |N_2 - N'_2| \left( \frac{1}{u} + \frac{1}{m} \right). \quad (5.57)$$

We use the following Bernstein-type concentration inequality (see Devroye et al., 1996, Problem 8.3) for the binomial random variable  $X \sim \text{Bin}(p, n)$ :  $\mathbf{P}_X \{|X - \mathbf{E}X| > t\} < 2 \exp\left(-\frac{3t^2}{8np}\right)$ . Abbreviate  $Q \triangleq \frac{1}{m} + \frac{1}{u}$ . Noting that  $N_1, N_2 \sim \text{Bin}\left(\frac{m}{m+u}, m+u\right)$ , we use (5.56), (5.57) and the Bernstein-type inequality (applied with  $n \triangleq m+u$  and  $p \triangleq \frac{m}{m+u}$ ) to obtain

$$\begin{aligned} & \mathbf{P}_{N_1, N_2} \{|s(N_1, N_2) - s(\mathbf{E}_{\tilde{\sigma}}\{N_1\}, \mathbf{E}_{\tilde{\sigma}}\{N_2\})| \geq \epsilon\} \\ & \leq \mathbf{P}_{N_1, N_2} \{|s(N_1, N_2) - s(N_1, \mathbf{E}_{\tilde{\sigma}}N_2)| + |s(N_1, \mathbf{E}_{\tilde{\sigma}}N_2) - s(\mathbf{E}_{\tilde{\sigma}}N_1, \mathbf{E}_{\tilde{\sigma}}N_2)| \geq \epsilon\} \\ & \leq \mathbf{P}_{N_1, N_2} \left\{ |s(N_1, N_2) - s(N_1, \mathbf{E}_{\tilde{\sigma}}N_2)| \geq \frac{\epsilon}{2} \right\} \\ & \quad + \mathbf{P}_{N_1, N_2} \left\{ |s(N_1, \mathbf{E}_{\tilde{\sigma}}N_2) - s(\mathbf{E}_{\tilde{\sigma}}N_1, \mathbf{E}_{\tilde{\sigma}}N_2)| \geq \frac{\epsilon}{2} \right\} \\ & \leq \mathbf{P}_{N_2} \left\{ |N_2 - \mathbf{E}_{\tilde{\sigma}}N_2| B_{\max} Q \geq \frac{\epsilon}{2} \right\} + \mathbf{P}_{N_1} \left\{ |N_1 - \mathbf{E}_{\tilde{\sigma}}N_1| B_{\max} Q \geq \frac{\epsilon}{2} \right\} \\ & \leq 4 \exp\left(-\frac{3\epsilon^2}{32(m+u)\frac{m}{m+u}B_{\max}^2Q^2}\right) = 4 \exp\left(-\frac{3\epsilon^2}{32mB_{\max}^2Q^2}\right). \end{aligned}$$

Next we use the following fact (see Devroye et al., 1996, Problem 12.1): if a nonnegative random variable  $X$  satisfies  $\mathbf{P}\{X > t\} \leq c \cdot \exp(-kt^2)$  for  $c \geq 1$ , then  $\mathbf{E}X \leq \sqrt{\ln(ce)/k}$ . Using this fact, along with  $c \triangleq 4$  and  $k \triangleq 3/(32mQ^2)$ , we have

$$\begin{aligned} |\mathbf{E}_{N_1, N_2} \{s(N_1, N_2)\} - s(\mathbf{E}_{\tilde{\sigma}}N_1, \mathbf{E}_{\tilde{\sigma}}N_2)| & \leq \mathbf{E}_{N_1, N_2} |s(N_1, N_2) - s(\mathbf{E}_{\tilde{\sigma}}N_1, \mathbf{E}_{\tilde{\sigma}}N_2)| \\ & \leq \sqrt{\frac{32 \ln(4e)}{3} m B_{\max}^2 \left( \frac{1}{u} + \frac{1}{m} \right)^2} \quad (5.58) \end{aligned}$$

### 5.8.3 Proof of Lemma 11

The proof is a straightforward extension of the proof of Lemma 5 from Meir and Zhang (2003) and is also similar to the proof of our Lemma 9 in Appendix 5.8.1. We prove a stronger claim: if for all  $i \in I_1^{m+u}$  and  $\mathbf{v}, \mathbf{v}' \in \mathcal{V}$ ,  $|f(v_i) - f(v'_i)| \leq |g(v_i) - g(v'_i)|$ , then for any function  $\tilde{c}: \mathbb{R}^{m+u} \rightarrow \mathbb{R}$ .

$$\mathbf{E}_\sigma \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^{m+u} \sigma_i f(v_i) + \tilde{c}(\mathbf{v}) \right] \leq \mathbf{E}_\sigma \sup_{\mathbf{v} \in \mathcal{V}} \left[ \sum_{i=1}^{m+u} \sigma_i g(v_i) + \tilde{c}(\mathbf{v}) \right]. \quad (5.59)$$

We use the abbreviation  $\sigma_1^n \triangleq \sigma_1, \dots, \sigma_n$ . The proof is by induction on  $n$ , such that  $0 \leq n \leq m+u$ . The lemma trivially holds for  $n = 0$ . Suppose the lemma holds for  $n-1$ . In other words, for any function  $\tilde{c}(\mathbf{v})$ ,

$$\mathbf{E}_{\sigma_1^{n-1}} \sup_{\mathbf{v} \in \mathcal{V}} \left[ \tilde{c}(\mathbf{v}) + \sum_{i=1}^{n-1} \sigma_i f(v_i) \right] \leq \mathbf{E}_{\sigma_1^{n-1}} \sup_{\mathbf{v} \in \mathcal{V}} \left[ \tilde{c}(\mathbf{v}) + \sum_{i=1}^{n-1} \sigma_i g(v_i) \right]. \quad (5.60)$$

Let  $p \triangleq \frac{mu}{(m+u)^2}$ . We have

$$A \triangleq \mathbf{E}_{\sigma_1^n} \sup_{\mathbf{v} \in \mathcal{V}} \left[ c(\mathbf{v}) + \sum_{i=1}^n \sigma_i f(v_i) \right] = \mathbf{E}_{\sigma_n} \mathbf{E}_{\sigma_1^{n-1}} \sup_{\mathbf{v} \in \mathcal{V}} \left[ c(\mathbf{v}) + \sum_{i=1}^n \sigma_i f(v_i) \right] \quad (5.61)$$

$$= p \mathbf{E}_{\sigma_1^{n-1}} \left\{ \sup_{\mathbf{v} \in \mathcal{V}} \left[ c(\mathbf{v}) + \sum_{i=1}^{n-1} \sigma_i f(v_i) + f(v_n) \right] + \sup_{\mathbf{v} \in \mathcal{V}} \left[ c(\mathbf{v}) + \sum_{i=1}^{n-1} \sigma_i f(v_i) - f(v_n) \right] \right\} \quad (5.62)$$

$$+ (1-2p) \mathbf{E}_{\sigma_1^{n-1}} \sup_{\mathbf{v} \in \mathcal{V}} \left[ c(\mathbf{v}) + \sum_{i=1}^{n-1} \sigma_i f(v_i) \right]. \quad (5.63)$$

We apply the inductive hypothesis three times: on the first and second summands in (5.62) with  $\tilde{c}(\mathbf{v}) \triangleq c(\mathbf{v}) + f(v_n)$  and  $\tilde{c}(\mathbf{v}) \triangleq c(\mathbf{v}) - f(v_n)$ , respectively, and on (5.63) with  $\tilde{c}(\mathbf{v}) \triangleq c(\mathbf{v})$ . We obtain

$$A \leq \underbrace{p \mathbf{E}_{\sigma_1^{n-1}} \left\{ \sup_{\mathbf{v} \in \mathcal{V}} \left[ c(\mathbf{v}) + \sum_{i=1}^{n-1} \sigma_i g(v_i) + f(v_n) \right] + \sup_{\mathbf{v} \in \mathcal{V}} \left[ c(\mathbf{v}) + \sum_{i=1}^{n-1} \sigma_i g(v_i) - f(v_n) \right] \right\}}_{\triangleq B} + \underbrace{(1-2p) \mathbf{E}_{\sigma_1^{n-1}} \sup_{\mathbf{v} \in \mathcal{V}} \left[ c(\mathbf{v}) + \sum_{i=1}^{n-1} \sigma_i g(v_i) \right]}_{\triangleq C}.$$

The expression  $B$  can be written as follows.

$$\begin{aligned}
B &= p\mathbf{E}_{\sigma_1^{n-1}} \left\{ \sup_{\mathbf{v} \in \mathcal{V}} \left[ c(\mathbf{v}) + \sum_{i=1}^{n-1} \sigma_i g(v_i) + f(v_n) \right] + \sup_{\mathbf{v}' \in \mathcal{V}} \left[ c(\mathbf{v}') + \sum_{i=1}^{n-1} \sigma_i g(v'_i) - f(v'_n) \right] \right\} \\
&= p\mathbf{E}_{\sigma_1^{n-1}} \sup_{\mathbf{v}, \mathbf{v}' \in \mathcal{V}} \left[ c(\mathbf{v}) + c(\mathbf{v}') + \sum_{i=1}^{n-1} \left[ \sigma_i (g(v_i) + g(v'_i)) \right] + (f(v_n) - f(v'_n)) \right] \\
&= p\mathbf{E}_{\sigma_1^{n-1}} \sup_{\mathbf{v}, \mathbf{v}' \in \mathcal{V}} \left[ c(\mathbf{v}) + c(\mathbf{v}') + \sum_{i=1}^{n-1} \left[ \sigma_i (g(v_i) + g(v'_i)) \right] + |f(v_n) - f(v'_n)| \right]. \quad (5.64)
\end{aligned}$$

The equality (5.64) holds since the expression  $c(\mathbf{v}) + c(\mathbf{v}') + \sum_{i=1}^{n-1} \sigma_i (g(v_i) + g(v'_i))$  is symmetric in  $\mathbf{v}$  and  $\mathbf{v}'$ . Thus, if  $f(\mathbf{v}) < f(\mathbf{v}')$  then we can exchange the values of  $\mathbf{v}$  and  $\mathbf{v}'$  and this will increase the value of the expression under the supremum. Since  $|f(v_n) - f(v'_n)| \leq |g(v_n) - g(v'_n)|$  we have

$$\begin{aligned}
B &\leq p\mathbf{E}_{\sigma_1^{n-1}} \sup_{\mathbf{v}, \mathbf{v}' \in \mathcal{V}} \left[ c(\mathbf{v}) + c(\mathbf{v}') + \sum_{i=1}^{n-1} \left[ \sigma_i (g(v_i) + g(v'_i)) \right] + |g(v_n) - g(v'_n)| \right] \\
&= p\mathbf{E}_{\sigma_1^{n-1}} \sup_{\mathbf{v}, \mathbf{v}' \in \mathcal{V}} \left[ c(\mathbf{v}) + c(\mathbf{v}') + \sum_{i=1}^{n-1} \left[ \sigma_i (g(v_i) + g(v'_i)) \right] + (g(v_n) - g(v'_n)) \right] \\
&= p\mathbf{E}_{\sigma_1^{n-1}} \left\{ \sup_{\mathbf{v} \in \mathcal{V}} \left[ c(\mathbf{v}) + \sum_{i=1}^{n-1} \sigma_i g(v_i) + g(v_n) \right] + \sup_{\mathbf{v} \in \mathcal{V}} \left[ c(\mathbf{v}) + \sum_{i=1}^{n-1} \sigma_i g(v_i) - g(v_n) \right] \right\} \triangleq D.
\end{aligned}$$

Therefore, using the reverse argument of (5.61)-(5.63),

$$A \leq C + D = \mathbf{E}_{\sigma_1^n} \sup_{\mathbf{v} \in \mathcal{V}} \left[ c(\mathbf{v}) + \sum_{i=1}^n \sigma_i g(v_i) \right].$$

#### 5.8.4 Proof of Lemma 12

Let  $c \in \mathbb{R}$ ,  $U \triangleq c \cdot I$ . If  $c = 0$ , then the soft classification generated by  $\mathcal{A}$  is a constant zero. In this case, for any  $\mathbf{h}$  generated by  $\mathcal{A}$ , we have  $\widehat{\mathcal{L}}_m^\gamma(\mathbf{h}) = 1$  and the lemma holds.

Suppose  $c \neq 0$ . Then

$$\boldsymbol{\alpha} = \frac{1}{c} \cdot \mathbf{h}. \quad (5.65)$$

Since the  $(m+u) \times (m+u)$  matrix  $U$  has  $m+u$  singular values, each one is precisely  $c$ , by (5.19) the Rademacher complexity of the trivial ULR is bounded by

$$\mu_1 \sqrt{\frac{2}{mu} (m+u)c^2} = c\mu_1 \sqrt{2 \left( \frac{1}{m} + \frac{1}{u} \right)}. \quad (5.66)$$

We assume w.l.o.g. that the training points have indices from 1 to  $m$ . Let  $A = \{i \in I_1^m \mid y_i h(i) > 0 \text{ and } |h(i)| > \gamma\}$  be a set of indices of training examples with zero margin loss. Let  $B = \{i \in I_1^m \mid |h(i)| \in [-\gamma, \gamma]\}$  and  $C = \{i \in I_1^m \mid y_i h(i) < 0 \text{ and } |h(i)| > \gamma\}$ . By (5.66) and the definition of the sets  $A$ ,  $B$  and  $C$  we obtain that the bound (5.66) is at least

$$c \sqrt{(|A| + |C|) \frac{\gamma^2}{c^2} + \sum_{i \in B} \frac{h(i)^2}{c^2}} \sqrt{\frac{1}{m}} , \quad (5.67)$$

where the left-hand square root is a lower bound on  $\mu_1$ . Therefore, the risk bound (5.12) is bounded from below by

$$\begin{aligned} & \widehat{\mathcal{L}}_m^\gamma(\mathbf{h}) + \frac{1}{\gamma} \sqrt{(|A| + |C|) \gamma^2 + \sum_{i \in B} h(i)^2} \cdot \sqrt{\frac{2}{m}} \geq \\ & \frac{\sum_{i \in B} (1 - |h(i)|/\gamma) + |C|}{m} + \sqrt{|A| + |C| + \sum_{i \in B} \frac{h(i)^2}{\gamma^2}} \cdot \sqrt{\frac{2}{m}} = \\ & \frac{|B| + |C| - \sum_{i \in B} r_i}{m} + \sqrt{|A| + |C| + \sum_{i \in B} r_i^2} \cdot \sqrt{\frac{2}{m}} = \\ & \frac{m - |A| - \sum_{i \in B} r_i}{m} + \sqrt{|A| + |C| + \sum_{i \in B} r_i^2} \cdot \sqrt{\frac{2}{m}} \triangleq D , \quad (5.68) \end{aligned}$$

where  $r_i = \frac{|h(i)|}{\gamma}$ . We prove that  $D \geq 1$ . Equivalently, it is sufficient to prove that for  $r_{i_1}, \dots, r_{i_{|B|}} \in [0, 1]^{|B|}$  it holds that

$$f(r_{i_1}, \dots, r_{i_{|B|}}) = \frac{(|A| + \sum_{i \in B} r_i)^2}{|A| + |C| + \sum_{i \in B} r_i^2} \leq m . \quad (5.69)$$

We claim that the stronger statement holds:

$$f(r_{i_1}, \dots, r_{i_{|B|}}) = \frac{(|A| + |C| + \sum_{i \in B} r_i)^2}{|A| + |C| + \sum_{i \in B} r_i^2} \leq m . \quad (5.70)$$

To prove (5.70) we use the Cauchy-Schwarz inequality, stating that for any two vectors  $\mathbf{a}, \mathbf{b} \in \mathbb{R}^m$ ,  $\langle \mathbf{a}, \mathbf{b} \rangle \leq \|\mathbf{a}\|_2 \cdot \|\mathbf{b}\|_2$ . We set  $b_i = 1$  for all  $i \in I_1^m$ . The vector  $\mathbf{a}$  is set as follows:  $a_i \triangleq r_i$  if  $i \in B$  and  $a_i = 1$  otherwise. By this definition of  $\mathbf{a}$  and  $\mathbf{b}$ , we have that  $\langle \mathbf{a}, \mathbf{b} \rangle \geq 0$  and thus  $(\langle \mathbf{a}, \mathbf{b} \rangle)^2 \leq \|\mathbf{a}\|_2^2 \cdot \|\mathbf{b}\|_2^2$ . The application of this inequality with the defined vectors  $\mathbf{a}$  and  $\mathbf{b}$  results in the inequality (5.70).

### 5.8.5 Proofs from Section 5.5.2

**Proof of Lemma 13:** Let  $\mathbf{e}_i$  be a  $1 \times (m + u)$  vector whose  $i$ th entry equals 1 and other entries are zero. According to the definition of RKHS, we need to show

that for any  $1 \leq i \leq m + u$ ,  $h(i) = \langle U(i, \cdot), \mathbf{h} \rangle_L$ . We have

$$\begin{aligned}
\langle U(i, \cdot), \mathbf{h} \rangle_L &= U(i, \cdot) L \mathbf{h} = \mathbf{e}_i U L \mathbf{h} \\
&= \mathbf{e}_i \left( \sum_{i=2}^{m+u} \frac{1}{\lambda_i} \mathbf{u}_i \mathbf{u}_i^T \right) \left( \sum_{i=1}^{m+u} \lambda_i \mathbf{u}_i \mathbf{u}_i^T \right) \mathbf{h} = \mathbf{e}_i \left( \sum_{i=2}^{m+u} \mathbf{u}_i \mathbf{u}_i^T \right) \mathbf{h} \\
&= \mathbf{e}_i (I - \mathbf{u}_1 \mathbf{u}_1^T) \mathbf{h} = \mathbf{e}_i \left( I - \frac{1}{m+u} \mathbf{1} \cdot \mathbf{1}^T \right) \mathbf{h} = h(i) . \quad (5.71)
\end{aligned}$$

□

**Lemma 16** For any  $1 \leq i \leq m + u$ ,  $U(i, \cdot) \in \mathcal{H}_L$ .

**Proof:** Since  $L$  is a Laplacian matrix,  $\mathbf{u}_1 = \mathbf{1}$ . Since the vectors  $\{\mathbf{u}_i\}_{i=1}^{m+u}$  are orthonormal and  $\mathbf{u}_1 = \mathbf{1}$ , we have  $U \cdot \mathbf{1} = \left( \sum_{i=2}^{m+u} \frac{1}{\lambda_i} \mathbf{u}_i \mathbf{u}_i^T \right) \mathbf{1} = 0$ . Therefore, for any  $1 \leq i \leq m + u$ ,  $U(i, \cdot) \cdot \mathbf{1} = 0$ . □

**Proof of Lemma 14:** Let  $\|\mathbf{h}\|_L = \sqrt{\langle \mathbf{h}, \mathbf{h} \rangle_L} \triangleq \sqrt{\mathbf{h}^T L \mathbf{h}}$  be a norm in  $\mathcal{G}_L$ . The optimization problem (5.28)-(5.29) can be stated in the following form:

$$\min_{\mathbf{h} \in \mathcal{H}_L} \|\mathbf{h}\|_L^2 + c(\mathbf{h} - \bar{\tau})^T C(\mathbf{h} - \bar{\tau}) . \quad (5.72)$$

Let  $\mathcal{U} \subseteq \mathcal{H}_L$  be a vector space spanned by the vectors  $\{U(i, \cdot)\}_{i=1}^{m+u}$ . Let  $\mathbf{h}_{\parallel} \triangleq \sum_{i=1}^{m+u} \alpha_i U(i, \cdot)$  be a projection of  $\mathbf{h}$  onto  $\mathcal{U}$ . For any  $1 \leq i \leq m + u$ ,  $\alpha_i = \frac{\langle \mathbf{h}, U(i, \cdot) \rangle_L}{\|U(i, \cdot)\|_L}$ . Let  $\mathbf{h}_{\perp} = \mathbf{h} - \mathbf{h}_{\parallel}$  be a part of  $\mathbf{h}$  that is perpendicular to  $\mathcal{U}$ . It can be verified that  $\mathbf{h}_{\perp} \in \mathcal{H}_L$  and for any  $1 \leq i \leq m + u$ ,  $\langle \mathbf{h}_{\perp}, U(i, \cdot) \rangle_L = 0$ . For any  $1 \leq i \leq m + u$  we have

$$\begin{aligned}
h(i) &= \langle \mathbf{h}, U(i, \cdot) \rangle_L = \left\langle \sum_{j=1}^{m+u} \alpha_j U(j, \cdot), U(i, \cdot) \right\rangle_L + \langle \mathbf{h}_{\perp}, U(i, \cdot) \rangle_L \\
&= \sum_{j=1}^{m+u} \alpha_j \langle U(j, \cdot), U(i, \cdot) \rangle_L = \sum_{j=1}^{m+u} \alpha_j U(i, j) = h_{\parallel}(i) . \quad (5.73)
\end{aligned}$$

The second equation in (5.73) holds by Lemma 16. As a consequence of (5.73), the empirical error (the second term in (5.72)) depends only on  $\mathbf{h}_{\parallel}$ . Furthermore,

$$\mathbf{h}^T L \mathbf{h} = \langle \mathbf{h}, \mathbf{h} \rangle_L = \|\mathbf{h}\|_L^2 = \left\| \sum_{i=1}^{m+u} \alpha_i U(i, \cdot) \right\|_L^2 + \|\mathbf{h}_{\perp}\|_L^2 \geq \left\| \sum_{i=1}^{m+u} \alpha_i U(i, \cdot) \right\|_L^2 . \quad (5.74)$$

Therefore, for an  $\mathbf{h}^* \in \mathcal{H}$  that minimizes (5.72),  $\mathbf{h}_{\perp}^* = 0$  and  $\mathbf{h}^* = \mathbf{h}_{\parallel}^* = \sum_{i=1}^{m+u} \alpha_i U(i, \cdot) = U \boldsymbol{\alpha}$ . □

### 5.8.6 Proof of Lemma 15

Let  $L_N \triangleq I - L = I - D^{-1/2}WD^{-1/2}$  be a normalized Laplacian of  $W$ . The eigenvalues  $\{\lambda'_i\}_{i=1}^{m+u}$  of  $L_N$  are non-negative and the smallest eigenvalue of  $L_N$ , denoted here by  $\lambda'_{\min}$ , is zero (Chung, 1997). The eigenvalues of the matrix  $I - \beta L = (1 - \beta)I + \beta L_N$  are  $\{1 - \beta + \beta\lambda'_i\}_{i=1}^{m+u}$ . Since  $0 < \beta < 1$ , all the eigenvalues of  $I - \beta L$  are strictly positive. Hence the matrix  $I - \beta L$  is invertible and its eigenvalues are  $\left\{\frac{1}{1 - \beta + \beta\lambda'_i}\right\}_{i=1}^{m+u}$ . Finally, the eigenvalues of the matrix  $U$  are  $\left\{\frac{1 - \beta}{1 - \beta + \beta\lambda'_i}\right\}_{i=1}^{m+u}$ . Since  $\lambda'_{\min} = 0$ , the largest eigenvalue of  $U$  is 1. Since all eigenvalues of  $L_N$  are non-negative, we have that  $\lambda_{\min} > 0$ .

### 5.8.7 Proofs from Section 5.6

**Proof of Corollary 2:** Let  $\{A_i\}_{i=1}^{\infty}$  and  $\{p_i\}_{i=1}^{\infty}$  be a set of positive numbers such that  $\sum_{i=1}^{\infty} p_i \leq 1$ . By the weighted union bound argument we have from (5.38) that with probability of at least  $1 - \delta$  over the training/test set partitions, for all  $A_i$  and  $\mathbf{q} \in \Omega_{g, A_i}$ ,

$$\mathcal{L}_u(\tilde{\mathbf{h}}_{\mathbf{q}}) \leq \hat{\mathcal{L}}_m^{\gamma}(\tilde{\mathbf{h}}_{\mathbf{q}}) + \frac{R_{m+u}(\tilde{\mathcal{B}}_{g, A_i})}{\gamma} + c_0 Q \sqrt{\min(m, u)} + \sqrt{\frac{S}{2} Q \ln \frac{1}{p_i \delta}}. \quad (5.75)$$

We set  $A_i \triangleq g_0 s^i$  and  $p_i \triangleq \frac{1}{i(i+1)}$ . It can be verified that  $\sum_{i=1}^{\infty} p_i \leq 1$ . For each  $\mathbf{q}$  let  $i_{\mathbf{q}}$  be the smallest index for which  $A_{i_{\mathbf{q}}} \geq g(\mathbf{q})$ . We have two cases:

**Case 1**  $i_{\mathbf{q}} = 1$ . In this case  $i_{\mathbf{q}} = \log_s(\tilde{g}(\mathbf{q})/g_0) = 1$ .

**Case 2**  $i_{\mathbf{q}} \geq 2$ . In this case  $A_{i_{\mathbf{q}}-1} = g_0 s^{i_{\mathbf{q}}-1} < g(\mathbf{q}) \leq \tilde{g}(\mathbf{q}) s^{-1}$ , and therefore,  $i_{\mathbf{q}} \leq \log_s(\tilde{g}(\mathbf{q})/g_0)$ .

Thus we always have that  $i_{\mathbf{q}} \leq \log_s(\tilde{g}(\mathbf{q})/g_0)$ . It follows from the definition of  $A_{i_{\mathbf{q}}}$  and  $\tilde{g}(\mathbf{q})$  that  $A_{i_{\mathbf{q}}} \leq \tilde{g}(\mathbf{q})$ . We have that  $\ln(1/p_{i_{\mathbf{q}}}) \leq 2 \ln(i_{\mathbf{q}} + 1) \leq 2 \ln \log_s(s\tilde{g}(\mathbf{q})/g_0)$ . Substituting these bounds into (5.75) and taking into account the monotonicity of  $R_{m+u}(\tilde{\mathcal{B}}_{g, A_i})$  (in  $A_i$ ), we have that with probability of at least  $1 - \delta$ , for all  $\mathbf{q}$ , the bound (5.39) holds.  $\square$

**Proof of Theorem 8:** We require several definitions and facts from the convex analysis (Rockafellar, 1970). For any function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  the *conjugate function*  $f^* : \mathbb{R}^n \rightarrow \mathbb{R}$  is defined as  $f^*(\mathbf{z}) = \sup_{\mathbf{x} \in \mathbb{R}^n} (\langle \mathbf{z}, \mathbf{x} \rangle - f(\mathbf{x}))$ . The domain of  $f^*$  consists of all values of  $\mathbf{z}$  for which the value of the supremum is finite. A consequence of the definition of  $f^*$  is the so-called *Fenchel inequality*:

$$\langle \mathbf{x}, \mathbf{z} \rangle \leq f(\mathbf{x}) + f^*(\mathbf{z}). \quad (5.76)$$

It can be verified that the conjugate function of  $g(\mathbf{q}) = D(\mathbf{q}||\mathbf{p})$  is  $g^*(\mathbf{z}) = \ln \sum_{j=1}^{|\mathcal{B}|} p_j e^{z_j}$ . Let  $\tilde{\mathbf{h}}(i) \triangleq (h_1(i), \dots, h_{|\mathcal{B}|}(i))$ . In the derivation that follows we use the following inequality (Hoeffding, 1963): if  $X$  is a random variable such that  $a \leq X \leq b$  and  $c$  is a constant, then

$$\mathbf{E}_X \exp(cX) \leq \exp\left(\frac{c^2(b-a)^2}{8}\right). \quad (5.77)$$

For any  $\lambda > 0$  we have,

$$\begin{aligned} R_{m+u}(\tilde{\mathcal{B}}_{g,A}) &= Q \mathbf{E}_{\boldsymbol{\sigma}} \sup_{\mathbf{q} \in \Omega_{g,A}} \langle \boldsymbol{\sigma}, \tilde{\mathbf{h}}_{\mathbf{q}} \rangle = Q \mathbf{E}_{\boldsymbol{\sigma}} \sup_{\mathbf{q} \in \Omega_{g,A}} \left\langle \mathbf{q}, \sum_{i=1}^{m+u} \sigma_i \tilde{\mathbf{h}}(i) \right\rangle \\ &= \frac{Q}{\lambda} \mathbf{E}_{\boldsymbol{\sigma}} \sup_{\mathbf{q} \in \Omega_{g,A}} \left\langle \mathbf{q}, \lambda \sum_{i=1}^{m+u} \sigma_i \tilde{\mathbf{h}}(i) \right\rangle \\ &\leq \frac{Q}{\lambda} \left( \sup_{\mathbf{q} \in \Omega_{g,A}} g(\mathbf{q}) + \mathbf{E}_{\boldsymbol{\sigma}} g^* \left( \lambda \sum_{i=1}^{m+u} \sigma_i \tilde{\mathbf{h}}(i) \right) \right) \end{aligned} \quad (5.78)$$

$$\leq \frac{Q}{\lambda} \left( A + \mathbf{E}_{\boldsymbol{\sigma}} \ln \sum_{j=1}^{|\mathcal{B}|} p_j \exp \left[ \lambda \sum_{i=1}^{m+u} \sigma_i \mathbf{h}_j(i) \right] \right) \quad (5.79)$$

$$\begin{aligned} &\leq \frac{Q}{\lambda} \left( A + \sup_{\mathbf{h} \in \mathcal{B}} \mathbf{E}_{\boldsymbol{\sigma}} \ln \exp \left[ \lambda \sum_{i=1}^{m+u} \sigma_i \mathbf{h}(i) \right] \right) \\ &\leq \frac{Q}{\lambda} \left( A + \sup_{\mathbf{h} \in \mathcal{B}} \ln \mathbf{E}_{\boldsymbol{\sigma}} \exp \left[ \lambda \sum_{i=1}^{m+u} \sigma_i \mathbf{h}(i) \right] \right) \end{aligned} \quad (5.80)$$

$$\leq \frac{Q}{\lambda} \left( A + \sup_{\mathbf{h} \in \mathcal{B}} \ln \exp \left[ \frac{\lambda^2}{2} \sum_{i=1}^{m+u} \mathbf{h}(i)^2 \right] \right) \quad (5.81)$$

$$= Q \left( \frac{A}{\lambda} + \frac{\lambda}{2} \sup_{\mathbf{h} \in \mathcal{B}} \|\mathbf{h}\|_2^2 \right). \quad (5.82)$$

Inequality (5.78) is obtained by applying (5.76) with  $f \triangleq g$  and  $f^* \triangleq g^*$ . Inequality (5.79) follows from the definition of  $g$  and  $g^*$ . Inequality (5.80) is obtained by an application of the Jensen inequality and inequality (5.81) is obtained by applying  $m+u$  times (5.77). By minimizing (5.82) w.r.t.  $\lambda$  we obtain

$$R_{m+u}(\tilde{\mathcal{B}}_{g,A}) \leq Q \sqrt{2A \sup_{\mathbf{h} \in \mathcal{B}} \|\mathbf{h}\|_2^2}.$$

Substituting this bound into (5.38) we get that for any fixed  $A$ , with probability at least  $1 - \delta$ , for all  $\mathbf{q} \in \mathcal{B}_{g,A}$

$$\mathcal{L}_u(\tilde{\mathbf{h}}_{\mathbf{q}}) \leq \hat{\mathcal{L}}_m^{\gamma}(\tilde{\mathbf{h}}_{\mathbf{q}}) + \frac{Q}{\gamma} \sqrt{2A \sup_{\mathbf{h} \in \mathcal{B}} \|\mathbf{h}\|_2^2} + c_0 Q \sqrt{\min(m, u)} + \sqrt{\frac{S}{2}} Q \ln \frac{1}{\delta}.$$

Finally, by applying the weighted union bound technique, as in the proof of Corollary 2, we obtain the statement of the theorem.  $\square$

# Chapter 6

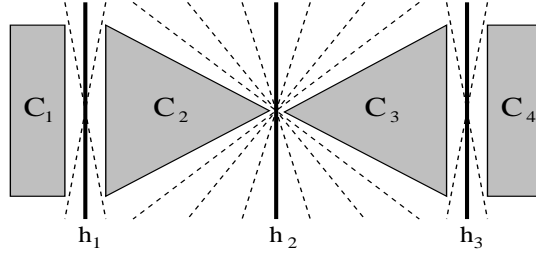
## Large Margin versus Large Volume in Transductive Learning

### 6.1 Introduction

In transductive binary classification, any hypothesis space (say, hyperplanes) is reduced to a finite collection of equivalence classes, given the unlabeled data. All hypotheses in the same class are identical in their binary classification of the data. For example, consider Fig. 6.1, in which  $C_1, \dots, C_4$  represent four “clouds” of unlabeled data. In this case all the hyperplanes passing in between  $C_3$  and  $C_4$  (and, in general, between  $C_i$  and  $C_{i+1}$ ) are in the same equivalence class (as well as infinitely many other hyperplanes that are not shown). The extra advantage in transduction is the possibility to prioritize these equivalence classes in accordance with our prior beliefs about the goodness of hypotheses given the current data. A classic principle for prioritizing equivalence classes is the large margin principle introduced in (Vapnik, 1982). According to this principle, the priority (or prior) of an equivalence class is proportional to the largest margin obtained by any of its members. In our example in Fig. 6.1 we should prefer the equivalence classes of  $\mathbf{h}_1$  and  $\mathbf{h}_3$  over the class of  $\mathbf{h}_2$  because they achieve a larger margin. This large margin consideration motivated the transductive support vector machine (TSVM) (Vapnik, 1998) approach for transduction.

In this chapter we consider a different, *large volume* principle, whereby the prior of an equivalence class is proportional to its “volume” in the hypothesis space. For example, in the case of hyperplanes, in Fig. 6.1 we should prefer the equivalence class of  $\mathbf{h}_2$  because it has a much larger volume in the hyperplane space. This is depicted in Fig. 6.1 by the number of dashed lines that pass between the clouds.

The large volume transductive principle was briefly treated in (Vapnik, 1982) for the case of hyperplanes. Here we further study this principle and instead of hyperplane spaces we consider general soft classification vectors whose set of



**Figure 6.1:** Large-margin vs. large-volume prior

equivalence classes with respect to all data points (ignoring labels) contains all possible dichotomies. Symmetry is broken by generating equivalence classes of non-uniform volume, defined via a non axis aligned data-dependent ellipsoid. Since exact or quantifiable volume approximation is computationally hard, we resort to a cruder approach whereby we measure the angles between hypotheses to the principal axes of the ellipsoid. This approach makes sense because long principal axes lie in regions of large volume. This construction leads to a general family of transductive algorithms and here we focus on one instantiation. Although the resulting algorithm is defined in terms of a non-convex optimization problem, we develop an efficient global optimum solution using a known technique. We also derive a transductive data-dependent error bound for this algorithm.

Our empirical evaluation of the new algorithm over a large number of datasets shows its overwhelming advantage over TSVM (and SVM) in text categorization and image classification problems. However, on a different set of UCI datasets, TSVM and SVM are significantly superior to the new algorithm. In our analysis of this finding, we identify some factors that influence the success and failure of our algorithm. In particular we show that our algorithm has significant advantage over TSVM when TSVM outperforms SVM.

## 6.2 The transductive setting

We consider the following distribution-free transductive model (Vapnik, 1998, page 341, Setting 1). A fixed set  $X_{m+u} = \{x_1, \dots, x_{m+u}\}$  of  $m + u$  points from some space  $\mathcal{X}$  is given. The binary labels  $y_i \in \{\pm 1\}$  of these points are also fixed but unknown to us. We uniformly at random pick a subset  $X_m \subseteq X_{m+u}$  of size  $m$  (among all subsets of size  $m$ ), and the labels of these points are provided. Rearranging indices, we denote by  $S_m = \{(x_i, y_i)\}_{i=1}^m$  the given labeled points, and by  $X_u = \{x_j\}_{j=m+1}^{m+u}$ , the remaining  $u$  unlabeled points. Using  $S_m$  and  $X_u$ , our goal is to guess the labels of points in  $X_u$  as accurately as possible.

Fixing  $m$  and  $u$ , we consider soft “hypotheses” that are *vectors*

$$\mathbf{h} = (h_1, \dots, h_{m+u}) \in \mathbb{R}^{m+u} \quad ,$$

where  $h_i$  is the soft, or confidence-rated label of example  $x_i$  given by “hypothesis”  $\mathbf{h}$ . The vector  $\mathbf{h}$  can be also interpreted as a *functional response vector* w.r.t. some underlying function  $f$  such that for any  $1 \leq i \leq m+u$ ,  $h_i = f(x_i)$ . Based on knowledge of the full-sample  $X_{m+u}$ , the learning algorithms we consider select an hypothesis space  $\mathcal{H} = \mathcal{H}(X_{m+u})$  of such soft classification vectors. Then, given the labels of training points the algorithm selects one hypothesis  $\mathbf{h} \in \mathcal{H}$ . For actual (binary) classification of  $x_i$  the algorithm outputs  $\text{sgn}(h_i)$ .

Two quantities of interest, for an hypothesis  $\mathbf{h}$ , are its transductive risk, or *test error*,  $R_u^\ell(\mathbf{h}) \triangleq \frac{1}{u} \sum_{i=m+1}^{m+u} \ell(h_i, y_i)$ , defined w.r.t. some loss function  $\ell$ , and the training or *empirical error* (w.r.t.  $\ell$ ),  $R_m^\ell(\mathbf{h}) \triangleq \frac{1}{m} \sum_{i=1}^m \ell(h_i, y_i)$ . In this paper  $\ell$  will be instantiated to the zero-one loss, the hinge loss, and linear loss functions. Whenever we omit  $\ell$  from  $R_u^\ell$  and  $R_m^\ell$ , we assume that the zero-one loss function is used.

### 6.3 Transductive maximum power inference

Let  $\mathcal{H}$  be any (soft) hypothesis space. A crucial observation, made by Vapnik (Vapnik, 1982) for a classification setting, is that after the introduction of the unlabeled data  $X_{m+u}$ , the set  $\mathcal{H}$  is partitioned into a finite number of *equivalence classes*  $H_1, \dots, H_N$ , such that all hypotheses in  $H_k, k = 1, \dots, N$ , generate the same dichotomy of  $X_{m+u}$ . Suppose that there exists some underlying distribution  $P(\mathbf{h})$  over  $\mathcal{H}$  such that one hypothesis is selected randomly according to  $P$  and the selected hypothesis determines the labels of points in  $X_{m+u}$ . Vapnik (Vapnik, 1998, page 708) defined the *power* of an equivalence class  $H_k$  as the probability mass (in terms of  $P$ ) of all the soft hypotheses in it,

$$\text{Power}(H_k) \triangleq \int_{H_k} dP(\mathbf{h}), \quad k = 1, \dots, N. \quad (6.1)$$

The power function provides a preference relation over all the dichotomies of  $X_{m+u}$  that can be generated by  $\mathcal{H}$ . So, for example, if we utilize an empirical error minimization framework, then, among all equivalence classes that classify the training set correctly, we should prefer one that has the largest power. We term this principle (that was already proposed in (Vapnik, 1998, pages 707-708)) ‘*transductive maximum power inference*’. The principle can also be extended to structural risk minimization.

In practice, of course, we do not know the underlying hypothesis distribution (and moreover, such a distribution may not exist) so in order to implement maximum power inference we must make a guess about some *prior* distribution  $P$

over  $\mathcal{H}$ , or directly infer  $\text{Power}(H_k)$  for  $k = 1, \dots, N$ . Obviously, a good prior distribution  $P$  should reflect auxiliary knowledge on the effectiveness of soft hypotheses.

If the power function is only dependent on the unlabeled data (and not on the train/test partition and the labels), the following error bound, which is an immediate consequence of Theorem 22 in (Derbeko et al., 2004), provides a compelling motivation for maximum power inference: for any  $0 < \delta < 1$ , with probability of at least  $1 - \delta$  over choices of  $S_m$ , for all  $k = 1, \dots, N$ ,

$$R_u(H_k) \leq R_m(H_k) + \sqrt{\left(\ln \frac{1}{\text{Power}(H_k)} + \ln \frac{1}{\delta}\right) \cdot \left(\frac{1}{m} + \frac{1}{u}\right)}, \quad (6.2)$$

where  $R_m(H_k)$  (respectively  $R_u(H_k)$ ) is the training (respectively test) error obtained by any instance of  $H_k$ . The error bound (6.2) implies that if an equivalence class  $H_k$  with a large power is empirically successful (over the training set), its test error over  $X_u$  is guaranteed to be small, with high probability.

## 6.4 On priors and powers

The bound (6.2), which essentially provides a sufficient condition for transductive learning, tells us that the power of the equivalence classes is a crucial quantity that can directly affect the learning speed and accuracy. Power assignment can be based on ‘low-level’ considerations, via prior assignment for hypotheses, as in (6.1). However, this assignment can also be done directly on complete equivalence classes, without defining a prior distribution  $P$  over soft hypotheses. In the latter case, the power is simply a prior over equivalence classes.

Various approaches of defining prior directly over equivalence classes have been considered in the past. The most well known approach is the *maximum margin* principle given by (Vapnik, 1982). The margin of a hyperplane is its minimal distance to any example in the full sample. By the maximum margin principle, the prior of the equivalence class  $H_k$  is proportional to the maximal margin obtained by any hyperplane  $h \in H_k$ . The maximal-margin principle motivated the well known transductive SVM (TSVM) approach. Other prior assignment approaches using compression, clustering and graph cuts are discussed in (Derbeko et al., 2004) and (Hanneke, 2006).

Effective power assignments must rely on some specialized knowledge that requires insight into the learning problem at hand. For some problems, priors on soft hypotheses (or power of equivalence classes) can be difficult to identify or quantify. Vapnik proposed an alternative prior encoding scheme through the *universum* (Vapnik, 1998, page 707). The universum is a set of unlabeled examples belonging to the same domain  $\mathcal{X}$ , but are known not to belong to any one of the two classes. The power of an equivalence class should be taken as the

number of dichotomies it obtains over the universum examples.<sup>1</sup> Since counting the number of dichotomies is computationally hard, it was proposed to approximate it with the number of contradictions. A universum example  $x$  contradicts an equivalence class  $H_k$  if there exists a pair of soft hypotheses  $\mathbf{h}, \mathbf{h}' \in H_k$ , such that  $h(x) \neq h'(x)$ . We term this approximation as the (universum) *maximum contradiction* principle.

Although the universum approach as presented above is transductive, it can also be motivated for induction. In fact, the first empirical study and validation of the universum idea is within an inductive setting (Weston, Collobert, Sinz, Bottou, & Vapnik, 2006), where an hypothesis class of hyperplanes is considered and it is suggested to approximate the number of contradictions of an equivalence class  $H_k$  by the minimum, over all  $\mathbf{h} \in H_k$ , of the sum of  $\ell_1$ -distances of  $\mathbf{h}$  from all universum examples. The intuition behind this approximation is that a very close proximity of  $h$  to a universum example  $x$  implies that a slight perturbation in the direction of  $\mathbf{h}$  will generate a new  $\mathbf{h}'$  that classifies  $x$  differently. The success of this approximated maximum contradictions principle depends on the choice of universum examples and it was shown in (Weston et al., 2006) that a combination of both the maximum margin and the maximum contradictions principles can outperform the maximum margin principle alone, if the universum is effectively selected.

In some domains universum examples arise naturally. For example, in a binary recognition problem where we want to separate the digits '1' and '2', examples of other digits can form an effective universum (Weston et al., 2006). But in general, universum examples may be hard to generate, especially in problems where we cannot easily perceive the membership of the universum examples to the domain.

## 6.5 A large volume principle

Consider a transductive classification setting and assume for now that  $\mathcal{H}$  (which may depend on  $X_{m+u}$ ) is finite. We consider the assignment of a prior measure  $P$  over  $\mathcal{H}$ . In the absence of any other knowledge, by the principle of insufficient reason, the prior of *any* two soft hypotheses (not necessarily from the same equivalence class) should be the same. This, of course, does not imply that the powers of two equivalence classes are identical.<sup>2</sup> According to (6.1), if  $P$  is uniform and  $\mathcal{H}$  is finite then the power of any equivalence class is proportional to its size. A straightforward extension of this argument to a continuously infinite (soft)  $\mathcal{H}$

---

<sup>1</sup>In philosophical terms, the universum is used to measure *falsifiability* (or *specificity*) – the more powerful equivalence classes are those that are more *falsifiable* by the universum points (Vapnik, 1998).

<sup>2</sup>Note that whenever the number of equivalence classes is  $\Omega(2^{m+u})$ , if the power is uniform over classes, we cannot bound the transductive test error.

results in a power function that assigns to each equivalence class the geometric volume of soft hypotheses contained in it. We term this application of the maximal power inference principle with a uniform prior (over the soft hypotheses) the *large volume* principle.

There are a few previous works that explicitly or implicitly utilized a large volume principle for an hypothesis space of (kernelized) hyperplanes. Vapnik (Vapnik, 1982, Section 10.5) proposed to approximate the volume of an equivalence class (of hyperplanes) by the distance between convex hulls of positive and negative examples. As shown by (Bennett & Bredensteiner, 2000), this distance is precisely the margin.

Tong and Koller (Tong & Koller, 2001) exploited a duality between hyperplanes and instance points, where hyperplanes are viewed as points on a sphere and examples are viewed as hyperplanes passing through the sphere. They approximated the volume of an equivalence class (corresponding to the version space) by the radius of the maximally inscribed ball within a conic section. This radius is precisely the margin and the approximation can be arbitrarily poor whenever the equivalence class is an elongated section.

Again for hyperplanes, Graepel et al. (Graepel, Herbrich, & Obermayer, 1999) approximated the volume of hypothesis equivalence classes using a kernel billiard algorithm. Their algorithm operates in a transductive setting, but considers equivalence classes defined by training points and a single test point. In contrast, we consider here equivalence classes defined by all training and test points.

Finally, we observe that one can approximate the volume using uniformly drawn universum examples. In this case one can show that, asymptotically, the equivalence classes with larger volume will have a larger number of contradictions.<sup>3</sup>

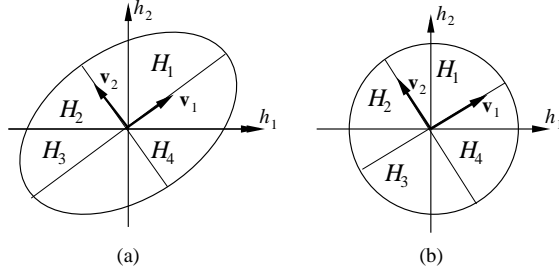
The main difference between our contribution and the previous work described above is that instead of hyperplane spaces we consider a much richer space of general soft classification vectors. This space, unlike hyperplanes, generates all possible  $2^{m+u}$  dichotomies.

## 6.6 Transductive learning using the large volume principle

We describe a transductive learning scheme that approximates the large volume principle. This scheme motivates a family of new transductive algorithms. In

---

<sup>3</sup>Proof outline: consider a dual space of hyperplanes with the offset  $b = 0$  (w.l.o.g.). This space is a sphere and full sample points are hyperplanes passing through the origin and cutting the sphere. Each equivalence class is a conical section of this sphere. In the dual space, uniformly drawn universum examples are equivalent to uniformly drawn hyperplanes. Thus, a universum example generates a contradiction in a conical section iff its hyperplane cuts the section. If the conical section is large then it will be cut by many hyperplanes.



**Figure 6.2:** Visualization of hypothesis space: (a) equivalence classes have different volumes. (b) equivalence classes have the same volume.

this section we develop and analyze one instantiation of this scheme.

### 6.6.1 Volume approximation

Our approach for approximating the volume of the equivalence classes relies on hypothesis spaces that can be represented as ellipsoids in  $\mathbb{R}^{m+u}$ . Each soft hypothesis in the hypothesis space is a point in the ellipsoid. We approximate the volume of an equivalence class  $H_k$  by the angles between an (arbitrary) hypothesis in  $H_k$  and the principal axes of the ellipsoid.

Let the full sample  $X_{m+u}$  be given and fixed. Let  $\mathbf{h} \in \mathbb{R}^{m+u}$  be a soft transductive hypothesis, and  $Q$ , a positive definite  $(m+u) \times (m+u)$  matrix that may depend on  $X_{m+u}$ . The matrix  $Q$  is considered as a hyperparameter and in Section 6.9 we give an example for its instantiation. Consider a hypothesis space  $\mathcal{H}_Q = \{\mathbf{h} \mid \mathbf{h}^T Q \mathbf{h} \leq 1\}$ . Geometrically,  $\mathcal{H}_Q$  is an ellipsoid in  $\mathbb{R}^{m+u}$ , centered at zero. We denote it by  $\mathcal{E}(\mathcal{H}_Q)$ . Since  $Q$  is positive definite, the set  $\{\text{sign}(\mathbf{h}) : \mathbf{h} \in \mathcal{H}_Q\}$  contains all  $2^{m+u}$  possible dichotomies of  $X_{m+u}$ .

The Cartesian coordinate system divides the space  $\mathbb{R}^{m+u}$  into  $2^{m+u}$  quadrants. Each quadrant corresponds to one equivalence class in terms of hard classification. For any  $1 \leq k \leq 2^{m+u}$ , the volume of the equivalence class  $H_k$  is the volume of the intersection of  $\mathcal{E}(\mathcal{H}_Q)$  with the  $k$ th quadrant. For example, Fig. 6.2(a) shows four equivalence classes,  $H_k, k = 1, \dots, 4$ , for the case  $m+u = 2$ . Ultimately, we would like to compute the exact volume of these quadrant intersections. However, currently known algorithms for approximate volume computation of general convex bodies seem to be too slow for practical purposes (Lovász & Vempala, 2006).

We resort to the following heuristic approximation. Let  $\{(\lambda_i, \mathbf{v}_i)\}_{i=1}^{m+u}$  be the eigenvalues of  $Q$  along with their eigenvectors, such that for all  $2 \leq i \leq m+u$ ,  $\lambda_{i-1} \leq \lambda_i$ . We assume w.l.o.g. that for any  $1 \leq i \leq m+u$ ,  $\|\mathbf{v}_i\|_2 = 1$ . The direction and length of the  $i$ th principal axis of  $\mathcal{E}(\mathcal{H}_Q)$  are, respectively,  $\mathbf{v}_i$  and  $\sqrt{1/\lambda_i}$ . As shown in Fig. 6.2(a), the volume of  $H_k$  is proportional to the length of the principal axes of the ellipsoid, which falls in its quadrant. In

the extreme case of a perfect sphere (Fig. 6.2(b)), all equivalence classes are of the same volume and cannot be differentiated. Therefore, we should prefer skewed ellipsoids that ultimately reflect useful priors on hypothesis effectiveness. In Section 6.9.1 we give example of such skewed ellipsoids that yield preference for “smoother” hypotheses.

The number of principal axes is always  $m + u$  whereas the number of quadrants (and equivalent classes) is  $2^{m+u}$ . Hence, the vast majority of quadrants do not contain any principal axis and, unlike the 2-dimensional case, we cannot estimate the volume of an equivalence class using a corresponding principal axis. We propose to estimate the volume using a weighted sum of axes’ lengths such that the weights are determined by the polar proximity of an hypothesis to the principal axes.

Fix  $i$  and  $j$  such that  $1 \leq i < j \leq m + u$  and  $\lambda_i < \lambda_j$ . Then, the length of the  $i$ th principal axis is larger than the length of the  $j$ th one. Hence, the local neighborhood of  $\mathbf{v}_i$  has a larger volume than that of  $\mathbf{v}_j$ . A small angle between an hypothesis  $\mathbf{h}$  and some long principal axis is taken as an evidence that its equivalence class has large volume. Conversely, a small angle to a short principal axis is taken as an evidence of a small volume. Note that these two opposing conditions cannot be satisfied simultaneously since the principal axes are orthogonal. In Section 6.9 we briefly discuss the meaning of the eigenvectors for a particular  $Q$  of interest.

Let  $0 \leq a_1 \leq a_2 \leq \dots \leq a_{m+u}$  be an increasing sequence of weights. For any soft hypothesis  $\mathbf{h}$  let

$$V(\mathbf{h}) = \sum_{i=1}^{m+u} a_i \frac{(\mathbf{h}^T \mathbf{v}_i)^2}{\|\mathbf{h}\|_2^2} . \quad (6.3)$$

The expression  $(\mathbf{h}^T \mathbf{v}_i)^2 / \|\mathbf{h}\|_2^2$  is the square of the cosine of the angle between  $\mathbf{h}$  and the unit-length vector  $\mathbf{v}_i$ . The monotone increasing sequence of  $a_i$ ’s corresponds to a monotone decreasing sequence of the lengths of  $\mathbf{v}_i$ ’s. Thus, the weighted sum (6.3) gives larger weight to the angular closeness to short principal axes than to the long ones. Consequently, we expect (6.3) to be large when  $\mathbf{h}$  lies in the equivalence class of *low* volume and be small when  $\mathbf{h}$  lies in the equivalence class of *high* volume.

### 6.6.2 Approximate Volume Regularization (AVR) algorithm

We propose the following natural instantiation of the  $\{a_i\}_{i=1}^{m+u}$  such that they are inversely proportional to the lengths of their corresponding principal axes. Let  $d_i = \sqrt{1/\lambda_i}$  be the length of the  $i$ th largest principal axis of the ellipsoid and for

any  $\mathbf{h} \in \mathbb{R}^{m+u}$ , set  $a_i = 1/d_i^2 = \lambda_i$ . Then,

$$V(\mathbf{h}) = \sum_{i=1}^{m+u} \lambda_i \frac{(\mathbf{h}^T \mathbf{v}_i)^2}{\|\mathbf{h}\|_2^2} = \frac{\mathbf{h}^T Q \mathbf{h}}{\|\mathbf{h}\|_2^2} .$$

This volume approximation motivates the following family of transductive algorithms, which implements the large volume principle:

$$\min_{\mathbf{h} \in \mathcal{H}_Q} R_m(\mathbf{h}) + \gamma \cdot \frac{\mathbf{h}^T Q \mathbf{h}}{\|\mathbf{h}\|_2^2} , \quad (6.4)$$

where  $\gamma > 0$  is a regularization parameter.<sup>4</sup>

Instead of the 0/1 loss empirical error in (6.4), due to computational considerations (see Remark 11 in Section 6.9), we instantiate the loss function to the linear loss,  $\ell(h_i, y_i) \triangleq -h_i y_i$ . Fixing  $t > 0$  and constraining  $\mathbf{h}$  to be of length  $t$  we eliminate the denominator in (6.4). Also, we replace the constraint  $\mathbf{h} \in \mathcal{H}_Q$  with  $\mathbf{h} \in \mathbb{R}^{m+u}$  (see below). The resulting optimization problem is

$$\min_{\mathbf{h} \in \mathbb{R}^{m+u}} -\frac{1}{m} \sum_{i=1}^m h_i y_i + \gamma \cdot \mathbf{h}^T Q \mathbf{h} \quad (6.5)$$

$$\text{s.t.} \quad \|\mathbf{h}\|_2^2 = t^2. \quad (6.6)$$

We refer to the optimization problem (6.5)-(6.6) as the *Approximate Volume Regularization (AVR) algorithm*. Due to constraint (6.6) the loss  $-h_i y_i$  of each training example is lower bounded by  $-t$ . Notice that while the optimization in (6.5) is done in  $\mathbb{R}^{m+u}$ , the regularization is done relative to  $\mathcal{H}_Q$ . The reason is that under the constraint (6.6) the complexity term  $\mathbf{h}^T Q \mathbf{h}$  is a weighted sum of the squares of cosines between  $\mathbf{h}$  and the principal axes of  $\mathcal{E}(\mathcal{H}_Q)$ . Thus, the optimization problem (6.5)-(6.6) is directly implied by (6.4) under the above instantiations of the free parameters.

While the optimization problem (6.5)-(6.6) is not convex, it can be solved efficiently and exactly (to obtain a global optimum) using the method of (Gander, Golub, & von Matt, 1989). This solution is developed in Section 6.7.

## 6.7 Global optimum AVR optimization

Following (Gander et al., 1989), we solve (6.5)-(6.6) using Lagrange multipliers. Set

$$\Phi(\mathbf{h}, \rho) = -\frac{1}{m} \sum_{i=1}^m h_i y_i + \gamma \cdot \mathbf{h}^T Q \mathbf{h} - \rho(\|\mathbf{h}\|_2^2 - t^2) ,$$

---

<sup>4</sup>Note that one deficiency of the above approximation is that for two hypotheses  $\mathbf{h}$  and  $\mathbf{h}'$  from the same equivalence class,  $V(\mathbf{h})$  is not in general identical to  $V(\mathbf{h}')$ .

where  $\rho$  is a Lagrange multiplier. Then,  $\mathbf{h}^* = \min_{\mathbf{h} \in \mathbb{R}^{m+u}, \rho \in \mathbb{R}} \Phi(\mathbf{h}, \rho)$  is a solution of (6.5)-(6.6). The minimum of  $\Phi(\mathbf{h}, \rho)$  is achieved when its partial derivatives are zero. Let  $\mathbf{y} \in \mathbb{R}^{m+u}$  be a vector of labels, with the first  $m$  entries being the training labels and the last  $u$  entries being zeros. Differentiating  $\Phi(\mathbf{h}, \rho)$  w.r.t.  $\mathbf{h}$  and  $\rho$ , and equating both these derivatives to zero we get,

$$-\mathbf{y}/m + 2\gamma Q\mathbf{h} - 2\rho\mathbf{h} = 0; \quad (6.7)$$

$$\|\mathbf{h}\|_2^2 - t^2 = 0. \quad (6.8)$$

It follows from (6.7) that<sup>5</sup>

$$\mathbf{h} = \frac{1}{2m} (\gamma Q - \rho I)^{-1} \mathbf{y} . \quad (6.9)$$

is a solution of (6.5)-(6.6).

The expression (6.9) contains the unknown  $\rho$ , which we now determine. Let  $\mathbf{V}\mathbf{D}\mathbf{V}^T$  be the spectral decomposition of  $Q$ , such that  $\mathbf{V}\mathbf{D}\mathbf{V}^T = Q$ ,  $\mathbf{V}^T\mathbf{V} = \mathbf{V}\mathbf{V}^T = I$  and  $\mathbf{D}$  is a diagonal matrix with its diagonal elements  $\lambda_i$  being the eigenvalues of  $Q$ . Then (6.7)-(6.8) can be rewritten as

$$-\mathbf{y}/m + 2\gamma\mathbf{V}\mathbf{D}\mathbf{V}^T\mathbf{h} - 2\rho\mathbf{V}\mathbf{V}^T\mathbf{h} = 0 \quad (6.10)$$

$$\mathbf{h}^T\mathbf{V}\mathbf{V}^T\mathbf{h} - t^2 = 0 . \quad (6.11)$$

Letting  $\mathbf{u} = \mathbf{V}^T\mathbf{h}$  and  $\mathbf{d} = \mathbf{V}^T\mathbf{y}$ , (6.10)-(6.11) becomes

$$-\mathbf{d}/m + 2\gamma\mathbf{D}\mathbf{u} - 2\rho\mathbf{u} = 0 \quad (6.12)$$

$$\mathbf{u}^T\mathbf{u} - t^2 = 0 . \quad (6.13)$$

Isolating  $\mathbf{u}$  at (6.12) and substituting it in (6.13) we get

$$\frac{1}{(2m)^2} \mathbf{d}^T (\gamma\mathbf{D} - \rho I)^{-2} \mathbf{d} - t^2 = 0 . \quad (6.14)$$

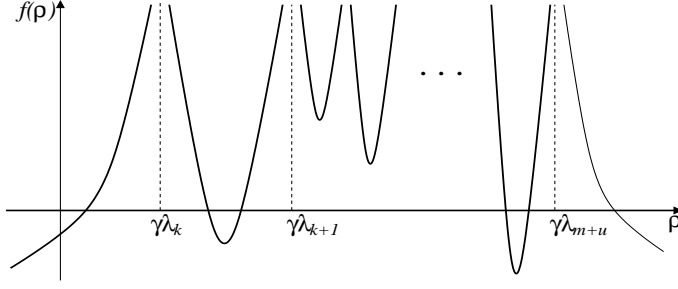
Let  $d_i$  be the  $i$ th component of  $\mathbf{d}$ . Since the matrix  $\mathbf{D}$  is diagonal, equation (6.14) is equivalent to

$$\frac{1}{(2m)^2} \sum_{i=1}^{m+u} \frac{d_i^2}{(\gamma\lambda_i - \rho)^2} - t^2 = 0 . \quad (6.15)$$

Equation (6.15) has multiple  $\rho$  solutions. As shown by (Forsythe & Golub, 1965, Theorem 2.7), the smallest possible  $\rho$  that satisfies (6.15) also minimizes  $\Phi(\mathbf{h}, \rho)$ . Thus, our goal is to find the smallest  $\rho$  satisfying (6.15). Since the matrix  $Q$  is

---

<sup>5</sup>Here we assume that the value of  $\rho$  satisfying (6.7)-(6.8) is not an eigenvalue of  $\gamma Q$  and the inverse  $(\gamma Q - \rho I)^{-1}$  exists. If this assumption does not hold (this can be easily verified by checking for each eigenvalue of  $\gamma Q$  if it satisfies (6.7)-(6.8)), then we can slightly perturb the hyperparameter  $\gamma$  to satisfy the assumption.



**Figure 6.3:** Structure of the function  $f(\rho)$ .  $k$  is the index of the smallest eigenvalue  $\lambda_i$  such that  $d_i \neq 0$ .

positive definite, all  $\lambda_i$ 's are strictly positive. Therefore, the function

$$f(\rho) = \frac{1}{(2m)^2} \sum_{i=1}^{m+u} \frac{d_i^2}{(\gamma\lambda_i - \rho)^2} - t^2 \quad (6.16)$$

has the structure depicted in Fig. 6.3. Considering this structure, our algorithm for finding the smallest  $\rho$  such that  $f(\rho) = 0$  is as follows: Let  $\tilde{\lambda}$  be the smallest  $\lambda_i$  such that  $d_i \neq 0$ . We consider the interval  $[\tilde{\lambda} - t_1, \tilde{\lambda} - t_2]$  such that  $t_1 > 0$ ,  $t_2 > 0$ ,  $f(\tilde{\lambda} - t_1) < 0$ ,  $f(\tilde{\lambda} - t_2) > 0$  and find the root of  $f$  in this domain using a binary search.

## 6.8 A risk bound

In this section we derive a transductive risk bound for the AVR algorithm (6.5)-(6.6). Our derivation relies on a known general transductive risk bound for ‘unlabeled/labeled (UL) decompositions’ of transductive algorithms as discussed in (El-Yaniv & Pechyony, 2007).

The soft classification output  $\mathbf{h}^*$  of any transductive algorithm can always be represented as  $\mathbf{h}^* = K \cdot \boldsymbol{\alpha}$ , where  $K$  is an  $(m+u) \times (m+u)$  matrix depending only on the unlabeled full sample  $X_{m+u}$ , and  $\boldsymbol{\alpha}$  is an  $(m+u) \times 1$  vector that can depend on both  $S_m$  and  $X_u$ . Such a decomposition is termed a UL (unlabeled-labeled) decomposition (El-Yaniv & Pechyony, 2007). Let  $K_{ij}$  be the  $(i, j)$ th entry of  $K$  and  $\|K\|_{\text{Fro}}^2 = \sum_{i,j=1}^{m+u} K_{ij}^2$ , be the squared Frobenius norm of  $K$ . For UL decompositions, the following holds.

**Theorem 1 ((El-Yaniv & Pechyony, 2007))** *Let  $\mathcal{A}$  be a transductive algorithm and  $\mathbf{h}^* = K \cdot \boldsymbol{\alpha}$  be its UL decomposition. Suppose that  $\|\boldsymbol{\alpha}\|_2 \leq \mu_1$ . Let  $c_0 = \sqrt{(32 \ln(4e))/3} < 5.05$  and  $W \triangleq 1/m + 1/u$ . Let  $\tilde{\mathcal{H}}$  be the set of soft*

hypotheses that can be generated by the algorithm when operated on any training/test partition. Then<sup>6</sup>, for any  $\nu > 0$  and  $\delta > 0$ , with probability of at least  $1 - \delta$  over the choice of the training set of size  $m$  from  $X_{m+u}$ , for all  $\mathbf{h} \in \tilde{\mathcal{H}}$ ,

$$R_u(\mathbf{h}) \leq R_m^{\ell_\nu}(\mathbf{h}) + \frac{\mu_1}{\nu} \sqrt{\frac{2}{mu} \|K\|_{\text{Fro}}^2} + c_0 W \sqrt{\min(m, u)} + \sqrt{2W \ln(1/\delta)} . \quad (6.17)$$

Since the  $(m + u) \times (m + u)$  matrix  $\mathbf{V}$  (defined in Section 6.7) consists of orthonormal columns, the solution  $\mathbf{h}^*$  of (6.5)-(6.6) can be represented as  $\mathbf{h}^* = \mathbf{V}\boldsymbol{\alpha}$ . The matrix  $\mathbf{V}$  depends only on the unlabeled examples. Hence the last equation is a UL decomposition of the AVR algorithm, with  $K \triangleq \mathbf{V}$ . By (6.6) we have that  $t^2 = \|\mathbf{h}^*\|_2^2 = \boldsymbol{\alpha}^T \mathbf{V}^T \mathbf{V} \boldsymbol{\alpha} = \boldsymbol{\alpha}^T \boldsymbol{\alpha}$ . Since each column of  $\mathbf{V}$  has unit length,  $\|K\|_{\text{Fro}}^2 = m + u$ . Substituting  $\mu_1 = t$  and  $\|K\|_{\text{Fro}}^2 = m + u$  in (6.17), we obtain<sup>7</sup>

$$R_u(\mathbf{h}) \leq R_m^{\ell_\nu}(\mathbf{h}) + (t/\nu) \sqrt{2W} + c_0 W \sqrt{\min(m, u)} + \sqrt{2W \ln(1/\delta)} . \quad (6.18)$$

Notice that the matrix  $Q$  influences the bound (6.18) indirectly, through the empirical error term. If  $t/\nu$  is a constant then the bound (6.18) converges at rate  $1/\sqrt{\min(m, u)}$ . In general, there is a trade-off between the values of  $t$  and  $\nu$ . If  $t$  is very small then, due to the constraint (6.6), all entries of the hypothesis  $\mathbf{h}$  generated by AVR are very close to zero. In this case, to achieve a small empirical hinge-loss, the value of  $\nu$  should also be small.

## 6.9 Experimental results

We tested the AVR algorithm over 31 binary problems including all 7 datasets from (Chapelle et al., 2006) and all 8 datasets from (Blum & Chawla, 2001). We also generated 6 datasets of image classification problems from the COIL-100 dataset (Nene, Nayar, & Murase, 1996), and took all 10 possible binary problems from the `comp.*` “super-category” in the 20-newsgroups dataset. We randomly subsampled large datasets to contain exactly 1500 examples and in all experiments we used a training set of size 100. We represented text datasets using word-based TF-IDF scores and normalized other datasets using a linear transformation such that the dynamic range of their attributes is  $[0, 1]$ .

We compared AVR with SVM and TSVM (Collobert et al., 2006).<sup>8</sup> In all problems, with the exception of the text datasets, SVM and TSVM were applied

<sup>6</sup>The loss function  $\ell_\nu$  used in the empirical error term is the hinge loss. For a positive real  $\nu$ ,  $\ell_\nu(y_1, y_2) = 0$  if  $y_1 y_2 \geq \nu$  and  $\ell_\nu(y_1, y_2) = \min\{1, 1 - y_1 y_2 / \nu\}$ .

<sup>7</sup>Using the standard technique of (Bousquet & Elisseeff, 2002) (see Theorem 18 there) it is possible to extend (6.18) to be uniform in  $\nu$ .

<sup>8</sup>We applied both SVM and TSVM using the *UniverSVM* package of F. Sinz, R. Collobert, J. Weston and L. Bottou, available at <http://www.kyb.tuebingen.mpg.de/bs/people/fabee/universvm.html>.

with an RBF kernel. For the text problems, slightly better performance was obtained with a linear kernel. All relevant hyperparameters of the SVM, TSVM and the AVR algorithm were selected using 5-fold cross validation (over the training set), from “reasonable” grids of possible values. For SVM/TSVM, the grid contained 80 possible hyperparameter assignments<sup>9</sup> and for AVR, 72 assignments (as described below).

### 6.9.1 On the AVR hyperparameters

The AVR algorithm has a number of parameters. The main parameter, which is left unspecified, is the matrix  $Q$ . One natural choice for  $Q$  is graph-based using the unnormalized Laplacian.<sup>10</sup> We constructed  $Q$  using the unlabeled data  $X_{m+u}$  as follows. We computed an  $(m+u) \times (m+u)$  similarity matrix  $S$ , whose  $(i, j)$ th entry is the cosine of the angle between  $x_i$  and  $x_j$ . Then we built an undirected  $k$ -nearest neighbors weighted graph,  $G = G(X_{m+u})$ , where there is an edge between  $x_i$  and  $x_j$  iff  $x_i$  is among the  $k$  most similar (according to  $S$ ) neighbors of  $x_j$ , or vice versa. Edge weights were taken to be the corresponding entries from  $S$ .<sup>11</sup> The value of  $k$  was selected using 5-fold cross validation from the set  $\{5, 10, 100\}$ . Let  $M$  be the adjacency matrix of  $G$ , and let  $D$  be the diagonal matrix whose  $(i, i)$ th entry is the sum of the  $i$ th row of  $M$ . Let  $L = D - M$  be the unnormalized Laplacian of  $G$ .

**Remark 9 (on the meaning of the eigenvectors of  $L$ )** *Let*

$$\mathcal{G} = \{\mathbf{g} \mid \mathbf{g} \in \mathbb{R}^{m+u}, \|\mathbf{g}\|_2 = 1\} .$$

*For any  $\mathbf{g} \in \mathcal{G}$ , let  $\mathbf{g}^T L \mathbf{g} = \sum_{i,j=1}^{m+u} (g_i - g_j)^2 m_{ij}$  be its “soft smoothness” w.r.t. the graph  $G(X_{m+u})$ , where  $m_{ij}$  is the  $(i, j)$ th entry of  $M$ . By the Rayleigh-Ritz theorem (Horn & Johnson, 1990), the smallest eigenvalue of  $L$  is  $\lambda_1 = \min_{\mathbf{g} \in \mathcal{G}} \mathbf{g}^T L \mathbf{g}$  and its eigenvector is  $\mathbf{v}_1 = \arg \min_{\mathbf{g} \in \mathcal{G}} \mathbf{g}^T L \mathbf{g}$ . Thus, the first eigenvector  $\mathbf{v}_1$  has the maximal smoothness (of value  $\lambda_1$ ). A generalization of this theorem in (Horn & Johnson, 1990) implies that for any  $1 \leq r \leq m+u$ ,  $\lambda_r = \min_{\mathbf{g} \in \mathcal{G}, \mathbf{g}^T \mathbf{v}_1 = 0, \dots, \mathbf{g}^T \mathbf{v}_{r-1} = 0} \mathbf{g}^T L \mathbf{g}$ , and the minimum is achieved by  $\mathbf{v}_r$ . Thus,*

<sup>9</sup>The SVM hyperparameters are  $C$  (weight of errors of labeled examples) and  $\sigma$  (the “width” of RBF kernel) and we considered all pairs  $(C, 1/\sigma^2) \in \{2^{-5}, 2^{-3}, 2^{-1}, 2^1, 2^3, 2^5, 2^7, 2^9\} \times \{2^{-15}, 2^{-13}, 2^{-11}, 2^{-9}, 2^{-7}, 2^{-5}, 2^{-3}, 2^{-1}, 2^1, 2^3\}$ . TSVM has additional hyperparameter  $C^*$  (weight of errors of unlabeled examples) and we considered all triples  $(C, C^*, 1/\sigma^2) = \{2^{-3}, 2^1, 2^5, 2^9\} \times \{2^{-3}, 2^1, 2^5, 2^9\} \times \{2^{-13}, 2^{-9}, 2^{-5}, 2^{-1}, 2^3\}$ . For text datasets the hyperparameter  $\sigma$  was not used.

<sup>10</sup>There are, of course, other possibilities for  $Q$ , such as a normalized Laplacian, linear models (Wu & Schölkopf, 2007) and RBF kernels. We have done preliminary experiments with each one of these possibilities for  $Q$  and found that they result in roughly the same performance as the choice of  $Q$  based on the unnormalized Laplacian.

<sup>11</sup>Our dataset normalization procedure (described above) implies that the entries of  $S$  are non-negative and thus, edge weights in  $G$  are non-negative as well.

the  $r$ -th largest eigenvector is the maximally smooth vector (with smoothness  $\lambda_r$ ) among all the vectors that are orthogonal to the  $r - 1$  maximally smooth vectors. By taking  $Q \triangleq L$ , we therefore prioritize highly smooth equivalent classes.

Although we would like to assign  $Q \triangleq L$ , it is well known that  $L$  is positive semi-definite. We need  $Q$  to be positive-definite to ensure a finite volume.<sup>12</sup> To this end, we truncate the larger eigenvectors of  $L$  using the following simple heuristic. We fix two parameters: a threshold  $\tau > 0$  and  $0 < c < 1$ , and truncate the  $l = \lceil c \cdot (m + u) \rceil$  largest eigenvectors to length  $\tau$ . Let  $\mu$  be the “stretch factor” of the  $l$ th largest vector (new length/old length). To preserve length proportions among the  $m + u - l$  smallest eigenvectors we stretch (or shrink) them by a factor  $\mu$ . In our experiments we selected the values of the hyperparameters  $\tau$  and  $c$  from the sets  $\{1, 10\}$  and  $\{0.05, 0.1, 0.9, 0.95\}$ , respectively, and set  $t = 1$ . Finally, the hyperparameter  $\gamma$  was selected from  $\{0.01/m, 1/m, 100/m\}$ .

**Remark 10 (On the computational complexity of AVR)** *By our construction of  $Q$  the computational complexity of AVR is dominated by the eigendecomposition of the graph Laplacian<sup>13</sup>  $L$ . In general the complexity of this decomposition is  $O((m + u)^3)$ . For small  $k \ll m + u$ , the matrix  $L$  is very sparse and the eigendecomposition can be computed faster. In Section 6.10 we discuss a fast method for constructing  $Q$  without performing costly eigendecompositions.*

**Remark 11 (On the choice of the loss function)** *The solution (6.9) of the AVR optimization problem involves the inversion of the  $(m + u) \times (m + u)$  matrix  $\gamma Q - \rho I$ . This operation is computationally expensive and has time complexity of  $O((m + u)^{2.376})$  (Coppersmith & Winograd, 1990). Let  $\{(\lambda_i, \mathbf{v}_i)\}_{i=1}^{m+u}$  be the eigendecomposition of  $Q$ . Recall that by our method of constructing  $Q$ , we know its eigendecomposition before computing  $(\gamma Q - \rho I)^{-1}$ . Since*

$$(\gamma Q - \rho I)^{-1} = \sum_{i=1}^{m+u} \frac{1}{\gamma \lambda_i - \rho} \mathbf{v}_i \mathbf{v}_i^T, \quad (6.19)$$

*given the eigendecomposition of  $Q$ , the inverse  $(\gamma Q - \rho I)^{-1}$  can be computed fast. Note that the eigendecomposition of  $Q$  is independent of the training/test partition and the choice of the hyperparameters  $\gamma$ ,  $\tau$  and  $c$ . Thus, we can reuse the eigendecomposition of  $Q$  for different values of  $\gamma$ ,  $\tau$ ,  $c$  and different training/test partitions and speed-up our experiments. This reuse would be impossible if instead of the linear loss  $-y_i h_i$  we took the commonly used squared loss<sup>14</sup>  $(y_i - h_i)^2$ ,*

<sup>12</sup>If  $Q$  is semi-definite then its smallest eigenvalue  $\lambda_1$  is zero. In this case the length  $(1/\sqrt{\lambda_1})$  of the principal vector  $\mathbf{v}_1$  (corresponding to  $\lambda_1$ ) of the ellipse  $\mathcal{E}(\mathcal{H}_Q)$  (see Fig. 6.2) is infinite.

<sup>13</sup>Recall that this eigendecomposition is required in order to make  $L$  positive definite.

<sup>14</sup>If we use the squared loss, then instead of (6.9) we would obtain  $\mathbf{h} = \frac{1}{2m} (\gamma Q - \rho I + I^*)^{-1} \mathbf{y}$ , where  $I^*$  is a diagonal matrix whose  $(i, i)$ th entry equals 1, if the  $i$ th example is in the training set, and zero otherwise. The inverse in the last expression cannot be computed using the eigendecomposition of  $Q$ .

**Table 6.1:** Results for three dataset collections.

DATASET	SVM	TSVM	AVR
DATASETS FROM (CHAPELLE ET AL., 2006)			
G241C	23.07±0.44	<b>18.62±1.09</b>	<b>20.21±1.40</b>
G241N	25.26±0.40	23.29±0.87	<b>12.1±0.87</b>
DIGIT	6.03±0.53	<b>5.18±0.73</b>	<b>4.21±0.56</b>
USPS	9.07±0.45	8.11±0.95	<b>6.23±0.46</b>
COIL	17.41±1.31	17.15±1.39	<b>5.89±0.42</b>
BCI	<b>31.75±1.21</b>	<b>32.92±0.47</b>	48.94±0.99
TEXT	<b>25.47±0.74</b>	<b>24.05±0.88</b>	<b>24.73±0.47</b>
IMAGE DATASETS			
COIL1	<b>12.35±0.45</b>	<b>12.21±0.39</b>	<b>11.63±0.37</b>
COIL2	9.37±0.23	8.25±0.34	<b>2.07±0.52</b>
COIL3	20.04±0.56	18.44±0.61	<b>12.17±0.71</b>
COIL4	12.35±0.67	9.73±0.35	<b>5.46±0.56</b>
COIL5	25.75±1.46	24.69±1.89	<b>15.62±0.87</b>
COIL6	24.46±1.07	23.13±0.90	<b>8.50±1.39</b>
TEXT (20 NEWSGROUPS)			
GRAPHICS/MISC	19.79±1.46	17.54±1.09	<b>14.76±0.34</b>
GRAPHICS/PC	16.86±1.79	13.96±1.39	<b>9.55±0.30</b>
GRAPHICS/MAC	12.85±1.68	10.39±1.28	<b>7.64±0.54</b>
GRAPHICS/X	20.99±2.12	16.42±1.17	<b>14.36±0.76</b>
MISC/PC	21.25±1.79	19.40±1.30	<b>16.12±0.68</b>
MISC/MAC	13.74±1.70	<b>11.83±1.24</b>	<b>10.90±0.35</b>
MISC/X	16.91±2.13	<b>13.63±1.44</b>	<b>12.85±0.49</b>
PC/MAC	<b>23.41±2.00</b>	<b>20.40±1.21</b>	<b>20.42±0.78</b>
PC/X	9.79±2.27	8.76±1.38	<b>5.74±0.21</b>
MAC/X	10.73±2.55	8.27±1.35	<b>4.28±0.28</b>

resulting in an order of magnitude slow-down.

## 6.9.2 Results

Our results for the 31 datasets appear in Tables 6.1 and 6.2. Each experiment was performed 12 times with different random train/test partitions. In Tables 6.1 and 6.2, each entry is an average ( $\pm$  standard error of the mean) of these 12 experiments. It is evident that AVR overwhelmingly outperforms SVM and TSVM on the dataset collection of Table 6.1. In particular, AVR exhibits excellent performance in text categorization and image classification. However, AVR is significantly inferior to SVM/TSVM on the UCI datasets of Table 6.2.

**Table 6.2:** UCI datasets taken from (Blum & Chawla, 2001).

DATASET	SVM	TSVM	AVR
PIMA	<b>27.96±1.06</b>	<b>27.97±1.07</b>	36.78±0.83
BUPA	<b>34.52±0.86</b>	<b>32.99±1.09</b>	36.63±1.20
MUSH	<b>3.26±0.41</b>	<b>2.88±0.47</b>	<b>2.85±0.49</b>
MUSK	<b>12.44±0.70</b>	<b>11.75±0.59</b>	15.62±0.78
MONK	<b>0.58±0.36</b>	1.93±0.82	20.16±1.26
IONOSPHERE	<b>7.93±0.66</b>	<b>7.44±0.75</b>	16.50±0.61
TAE	<b>26.96±1.73</b>	<b>29.08±1.67</b>	37.75±2.05
VOTING	<b>5.26±0.40</b>	<b>4.64±0.26</b>	7.06±0.43

### 6.9.3 Analysis of results

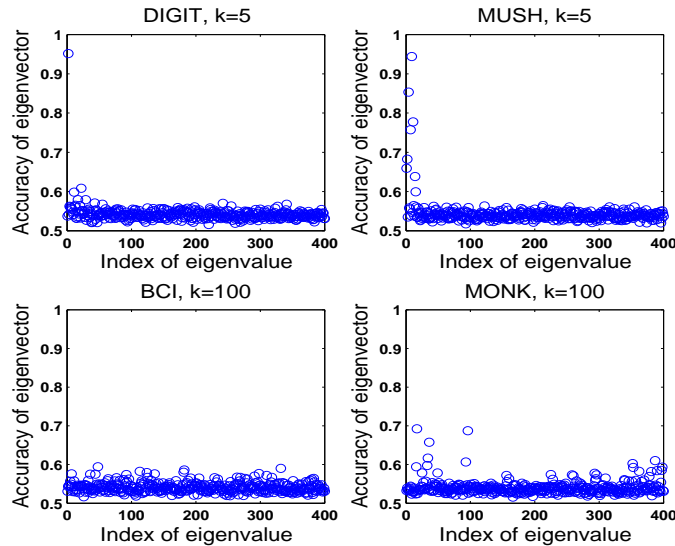
We investigated further cases where the AVR succeeded and failed and found two empirical characterizations of its performance.

#### Accuracy of eigenvectors

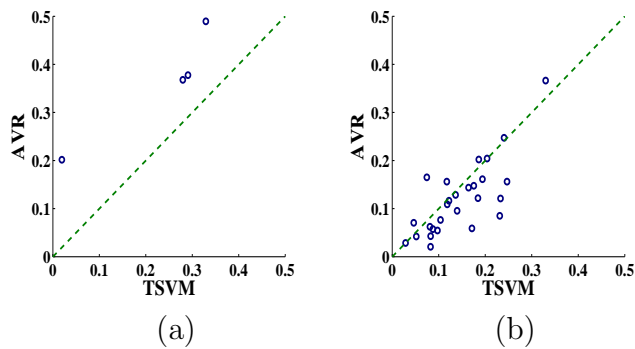
Let  $\{\lambda_i, \mathbf{v}_i\}_{i=1}^{m+u}$  be the eigenvectors and eigenvalues of  $Q$ , such that  $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_{m+u}$ . Since  $\mathbf{v}_i \in \mathbb{R}^{m+u}$ , we consider it as a vector of soft classifications of the full sample examples from  $X_{m+u}$ . In particular, we consider the  $j$ th entry of  $\mathbf{v}_i$  as the soft classification of  $x_j$ . For each  $1 \leq i \leq m+u$  we computed the training accuracy of  $\mathbf{v}_i$  and  $-\mathbf{v}_i$  and took the best accuracy among these two numbers. The resulting graphs of eigenvector training accuracies, averaged over 12 training/test partitions, are shown in Fig. 6.4 for 4 datasets. For each dataset we chose the value of  $k$  yielding the best performance in hindsight. We truncated the graphs to include the 400 smallest eigenvalues, since the accuracies of the eigenvectors corresponding to larger eigenvalues are almost always as those obtained by the eigenvectors corresponding to the eigenvalues with indices 200 – 400.

Figure 6.4 shows that in two datasets where AVR succeeds (DIGIT and MUSH) there are a few very accurate eigenvectors, which correspond to small eigenvalues of  $Q$ . Moreover, in these datasets there is a large gap in the accuracy of these eigenvectors and the others and there are no accurate eigenvectors corresponding to large eigenvalues. In contrast, in datasets where AVR failed (BCI and MONK) the accuracy of the eigenvectors corresponding to small eigenvalues is quite low. Qualitatively similar effects were observed in all the other datasets.

The above characterization in terms of the accuracies of the eigenvectors of  $Q$  suggests the following heuristic to quickly assess whether we should use AVR or large margin methods (SVM or TSVM). If the accuracy of the leading eigenvectors of  $Q$  is high relative to the accuracy of the large-margin methods, then run AVR



**Figure 6.4:** Accuracy of the eigenvectors of  $Q$ .



**Figure 6.5:** Comparison of AVR versus TSVM: (a) TSVM loses to SVM; (b) TSVM wins over SVM

with cross validation to determine its best parameters. Otherwise, use a large-margin method.

### Magnification of TSVM success and failure

Using the results of Tables 6.1 and 6.2 we divide all 31 datasets into two categories. The first category consists of the datasets where SVM outperformed TSVM. The second category consists of those where TSVM outperformed SVM. There are 4 datasets in the former category and 27 in the latter. Note that in this partition we measure performance by considering only average errors and ignore standards

error of the means.

A comparison of AVR and TSVM over datasets of these two categories is depicted in Fig. 6.5 using scatter plots. In these plots each point represents a comparison on a single dataset. If the point falls below the dashed line then AVR outperformed TSVM, and vice versa. It is evident that if SVM outperformed TSVM then TSVM also outperformed AVR. Conversely, if TSVM outperformed SVM, then in the significant majority of the datasets, AVR also outperformed TSVM. Thus, in cases where transductive learning was effective (in the sense that TSVM outperformed SVM), the AVR algorithm magnified the success of TSVM, and vice versa,

## 6.10 Concluding remarks

We developed a new transductive technique based on a large volume principle. The new technique is well motivated using the transductive maximal power inference. The resulting AVR algorithm that approximates this scheme is extremely successful in three (out of the four) sets of problems we examined (in particular, in text categorization and image classification problems) and fails in a set of UCI problems. The main questions are: why does AVR fail in the last set? How can we make a better data-dependent selection of the ellipsoid matrix  $Q$ ?

One possible direction could be to explicitly design  $Q$  matrices by encoding in eigenvectors and eigenvalues useful prior knowledge and information about the given data. We note that such constructions can also be beneficial from a computational complexity viewpoint since they would save the need for the spectral decompositions we currently perform.

It would be very interesting to identify datasets' characteristics that give an advantage to either the large margin or the large volume principle. In this regard, we note that in our comparison here these two principles were applied on different hypothesis spaces, namely, (kernelized) hyperplanes in the case of large margin and arbitrary soft response vectors in the case of large volume (which has much larger capacity). It would be of interest to compare these two principles w.r.t. the same space. We observed that the AVR algorithm magnified the success and failure of TSVM. We plan to further investigate this interesting effect. Further. Finally, a technical interesting question is how to better approximate or provide approximation guarantees for volume assessments in the context of our elliptic hypothesis classes.

# References

- Agarwal, A., & Chakrabarti, S. (2007). Learning random walks to rank nodes on graphs. In *Proceedings of the 26th International Conference on Machine Learning*, pp. 9–16.
- Agarwal, S. (2008). Transductive ranking on graphs. Tech. rep. MIT-CSAIL-TR-2008-51, Massachusetts Institute of Technology.
- Agarwal, S., Branson, K., & Belongie, S. (2006). Higher order learning with graphs. In *Proceedings of the 23rd International Conference on Machine Learning*, pp. 17–24.
- Ambroladze, A., Parrado-Hernandez, E., & Shawe-Taylor, J. (2007). Complexity of pattern classes and the lipschitz property. *Theoretical Computer Science*, 382(3), 232–246.
- Angluin, D. (1988). Queries and concept learning. *Machine Learning*, 2, 319–342.
- Argyriou, A., Herbster, M., & Pontil, M. (2006). Combining graph Laplacians for semi-supervised learning. In *Advances in Neural Information Processing Systems 18*, pp. 67–74.
- Audibert, J.-Y. (2004). A better variance control for PAC-Bayesian classification. Tech. rep. 905, Laboratoire de Probabilites et Modeles Aleatoires, Universites Paris 6 and Paris 7.
- Azran, A. (2007). The rendezvous algorithm: Multiclass semi-supervised learning with Markov random walks. In *Proceedings of the 24th International Conference on Machine Learning*, pp. 49–56.
- Azuma, K. (1967). Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal*, 19, 357–367.
- Bair, E., & Tibshirani, R. (2004). Semi-supervised methods to predict patient survival from gene expression data. *PLoS Biology*, 2(4), 511–522.
- Balcan, M.-F., & Blum, A. (2006). An augmented PAC model for semi-supervised learning. In Chapelle, O., Scholkopf, B., & Zien, A. (Eds.), *Semi-supervised learning*, pp. 383–404. MIT Press.
- Balcan, M.-F., Blum, A., Choi, P., Lafferty, J., Pantano, B., Rwebangira, M., & Zhu, X. (2005). Person identification in webcam images: An application

- of semi-supervised learning. In *ICML 2005 Workshop on Learning with Partially Classified Training Data*, pp. 1–9.
- Banerjee, A., & Langford, J. (2004). An objective evaluation criterion for clustering. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 515–520.
- Bartlett, P., Bousquet, O., & Mendelson, S. (2005). Local Rademacher complexities. *Annals of Probability*, *33*(4), 1497–1537.
- Bartlett, P., & Mendelson, S. (2002). Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, *3*, 463–482.
- Bax, E., & Callejas, A. (2008). An error bound based on a worst likely assignment. *Journal of Machine Learning Research*, *9*, 859–891.
- Belkin, M., Matveeva, I., & Niyogi, P. (2004). Regularization and semi-supervised learning on large graphs. In *Proceedings of the 17th Annual Conference on Learning Theory*, pp. 624–638.
- Belkin, M., & Niyogi, P. (2004). Semi-supervised learning on Riemannian manifolds. *Machine Learning Journal*, *56*, 209–239.
- Belkin, M., Niyogi, P., & Sindhvani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, *7*, 2399–2434.
- Ben-David, S., Kushilevitz, E., & Mansour, Y. (1997). Online learning versus offline learning. *Machine Learning*, *29*, 45–63.
- Ben-David, S., Lu, T., & Pál, D. (2008). Does unlabeled data provably help? Worst-case analysis of the sample complexity of semi-supervised learning. In *Proceedings of the 21th Annual Conference on Learning Theory*, pp. 33–44.
- Bennett, C., & Bredensteiner, E. (2000). *Geometry at Work*, chap. Geometry in Learning, pp. 132–145. Mathematical Association of America.
- Bennett, K., & Demiriz, A. (1999). Semi-supervised support vector machines. In *Advances in Neural Information Processing Systems 11*, pp. 368–374.
- Bennett, K., Demiriz, A., & Maclin, R. (2002). Exploiting unlabeled data in ensemble methods. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 289–296.
- Bie, T. D., & Cristianini, N. (2006). Fast SDP relaxations of graph cut clustering, transduction, and other combinatorial problem. *Journal of Machine Learning Research*, *7*, 1409–1436.
- Blake, A., Rother, C., Brown, M., Perez, P., & Torr, P. (2004). Interactive image segmentation using an adaptive GMMRF model. In *Proceedings of the 8th European Conference on Computer Vision*, pp. 428–441.

- Blum, A., & Chawla, S. (2001). Learning from labeled and unlabeled data using graph mincuts. In *Proceedings of the 18th International Conference on Machine Learning*, pp. 19–26.
- Blum, A., Lafferty, J., Rwebangira, M., & Reddy, R. (2004). Semi-supervised learning using randomized mincuts. In *Proceedings of the 21st International Conference on Machine Learning*.
- Blum, A., & Langford, J. (2003). PAC-MDL bounds. In *Proceedings of the 16th Annual Conference on Learning Theory*, pp. 344–357.
- Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pp. 92–100.
- Bottou, L., Cortes, C., & Vapnik, V. (1994). On the effective VC dimension.. Tech. rep., Neuroprose (<ftp://archive.cis.ohio-state.edu/pub/neuroprose>). Also available on <http://leon.bottou.org/papers>.
- Boucheron, S., Bousquet, O., & Lugosi, G. (2005). Theory of classification: A survey of some recent advances. *ESAIM: Probability and Statistics*, 9, 323–375.
- Bousquet, O., & Elisseeff, A. (2002). Stability and generalization. *Journal of Machine Learning Research*, 2, 499–526.
- Brefeld, U., Gärtner, T., Scheffer, T., & Wrobel, S. (2006). Efficient co-regularized least squares regression. In *Proceedings of the 23rd International Conference on Machine Learning*, pp. 137–144.
- Callut, J., Francoise, K., Saerens, M., & Dupont, P. (2008). Semi-supervised classification from discriminative random walks. In *Proceedings of the 19th European Conference on Machine Learning*, pp. 162–177.
- Carreira-Perpiñán, M., & Zemel, R. (2005). Proximity graphs for clustering and manifold learning. In *Advances in Neural Information Processing Systems 17*, pp. 225–232.
- Catoni, O. (2004). Improved vapnik-cervonenkis bounds. Tech. rep. 942, Laboratoire de Probabilites et Modeles Aleatoires, Universites Paris 6 and Paris 7.
- Catoni, O. (2007). *PAC-Bayesian supervised classification*, Vol. 56 of *IMS Lecture Notes - Monograph Series*. Institute of Mathematical Statistics.
- Chapelle, O., Schölkopf, B., & Zien, A. (2006). *Semi-supervised learning*. MIT Press.
- Chapelle, O., Sindhwani, V., & Keerthi, S. (2007). Brand and bound for semi-supervised support vector machines. In *Advances in Neural Information Processing Systems 19*, pp. 217–224.

- Chapelle, O., Sindhwani, V., & Keerthi, S. (2008). Optimization techniques for semi-supervised support vector machines. *Journal of Machine Learning Research*, 9, 203–233.
- Chapelle, O., Weston, J., & Schölkopf, B. (2003). Cluster kernels for semi-supervised learning. In *Advances in Neural Information Processing Systems 15*, pp. 585–592.
- Chapelle, O., & Zien, A. (2005). Semi-supervised classification by low-density separation. In *Proceedings of the 10th Workshop on Artificial Intelligence and Statistics*.
- Cheng, L., & Vishvanathan, S. (2007). Learning to compress images and video. In *Proceedings of the 24th International Conference on Machine Learning*, pp. 161–168.
- Chung, F. R. (1997). *Spectral graph theory*, Vol. 92 of *CBMS Regional Conference Series in Mathematics*. American Mathematical Society.
- Collobert, R., Sinz, F., Weston, J., & Bottou, L. (2006). Large scale transductive SVM. *Journal of Machine Learning Research*, 7, 1687–1712.
- Coppersmith, D., & Winograd, S. (1990). Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 9, 251–280.
- Cortes, C., & Mohri, M. (2007). On transductive regression. In *Advances in Neural Information Processing Systems 19*, pp. 305–312.
- Cortes, C., Mohri, M., Pechyony, D., & Rastogi, A. (2008). Stability of transductive regression algorithms. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 176–183.
- Culp, M., & Michailidis, G. (2008). Graph-based semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(1), 174–179.
- Dai, G., & Yeung, D.-Y. (2007). Kernel selection for semi-supervised kernel machines. In *Proceedings of the 24th International Conference on Machine Learning*, pp. 185–192.
- Dasgupta, S. (2005). Analysis of a greedy active learning strategy. In *Advances in Neural Information Processing Systems 17*, pp. 337–344.
- Dasgupta, S., Littman, M., & McAllester, D. (2002). PAC generalization bounds for co-training. In *Advances in Neural Information Processing Systems 14*, pp. 375–382.
- de Sa, V. (1994). Learning classification with unlabeled data. In *Advances in Neural Information Processing Systems 6*, pp. 112–119.
- Delalleau, O., Bengio, Y., & Roux, N. L. (2005). Efficient non-parametric function induction in semi-supervised learning. In *Proceedings of the 10th Workshop on Artificial Intelligence and Statistics*.

- Derbeko, P., El-Yaniv, R., & Meir, R. (2004). Explicit learning curves for transduction and application to clustering and compression algorithms. *Journal of Artificial Intelligence Research*, 22, 117–142.
- Devroye, L., L.Györfi, & Lugosi, G. (1996). *A Probabilistic Theory of Pattern Recognition*. Springer.
- Duh, K., & Kirchhoff, K. (2006). Lexicon acquisition for dialectical Arabic using transductive learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pp. 399–407.
- El-Yaniv, R., & Gerzon, L. (2005). Effective transductive learning via objective model selection. *Pattern Recognition Letters*, 26(13), 2104–2115.
- El-Yaniv, R., & Pechyony, D. (2006). Stable transductive learning. In *Proceedings of the 19th Annual Conference on Learning Theory*, pp. 35–49.
- El-Yaniv, R., & Pechyony, D. (2007). Transductive Rademacher complexity and its applications. In *Proceedings of the 20th Annual Conference on Learning Theory*, pp. 157–171.
- El-Yaniv, R., Pechyony, D., & Vapnik, V. (2008). Large margin vs. large volume in transductive learning. *Machine Learning Journal*, 72(3), 173–188.
- Forsythe, G., & Golub, G. (1965). On the stationary values of a second-degree polynomial on the unit sphere. *Journal of the Society for Industrial and Applied Mathematics*, 13(4), 1050–1068.
- Freund, Y., & Schapire, R. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1), 119–139.
- Galeano, S., & Herbster, M. (2007). A fast method to predict the labeling of a tree. In *ECML workshop*.
- Galstyan, A., & Cohen, P. (2008). Comparing diffusion models for graph-based semi-supervised learning. In *6th International Workshop on Mining and Learning with Graphs*.
- Gander, W., Golub, G., & von Matt, U. (1989). A constrained eigenvalue problem. *Linear Algebra and its Applications*, 114/115, 815–839.
- Gärtner, T., Le, Q., Burton, S., Smola, A., & Vishwanathan, V. (2006). Large-scale multiclass transduction. In *Advances in Neural Information Processing Systems 18*, pp. 411–418.
- Getz, G., Shental, N., & Domany, E. (2005). Semi-supervised learning - a statistical physics approach. In *Proceedings of ICML 2005 Workshop on Learning with Partially Classified Training Data*, pp. 37–44.
- Goldman, S., & Zhou, Y. (2000). Enhancing supervised learning with unlabeled data. In *Proceedings of the 17th International Conference on Machine Learning*, pp. 327–334.

- Graepel, T., Herbrich, R., & Obermayer, K. (1999). Bayesian transduction. In *NIPS*, pp. 456–462.
- Graepel, T., Herbrich, R., & Obermayer, K. (2000). Bayesian transduction. In *Advances in Neural Information Processing Systems 12*, pp. 456–462.
- Grimmett, G., & Stirzaker, D. (1995). *Probability and Random Processes*. Oxford Science Publications. Second edition.
- Haffari, G. (2006). A survey on inductive semi-supervised learning..
- Haffari, G., & Sarkar, A. (2007). Analysis of semi-supervised learning with the Yarowsky algorithm. In *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence*.
- Hanneke, S. (2006). An analysis of graph cut size for transductive learning. In *Proceedings of the 23rd International Conference on Machine Learning*, pp. 393–399.
- Hein, M., & Maier, M. (2007). Manifold denoising for finding natural representations of data. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, pp. 1646–1649.
- Herbster, M. (2008). Exploiting cluster-structure to predict the labeling of a graph. In *Proceedings of the 19th International Conference on Algorithmic Learning Theory*.
- Herbster, M., & Pontil, M. (2007). Prediction on a graph with a perceptron. In *Advances in Neural Information Processing Systems 19*, pp. 577–584.
- Herbster, M., Pontil, M., & Wainer, L. (2005). Online learning over graphs. In *Proceedings of the 22nd International Conference on Machine Learning*, pp. 305–312.
- Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58, 13–30.
- Horn, R., & Johnson, C. (1990). *Matrix Analysis*. Cambridge University Press.
- Huang, T., & Kecman, V. (2005). Performance comparisons of semi-supervised learning algorithms. In *ICML Workshop “Learning with Partially Classified Training Data”*, pp. 45–49.
- Hughes, N., Roberts, S., & Tarassenko, L. (2004). Semi-supervised learning of probabilistic models for ECG segmentation. In *Proceedings of the 26th Annual IEEE Conference on Engineering in Medicine and Biology*, pp. 434–437.
- Ifrim, G., & Weikum, G. (2006). Transductive learning for text classification using explicit knowledge models. In *Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pp. 223–234.

- Joachims, T. (1999). Transductive inference for text classification using support vector machines. In *Proceedings of the 19th International Conference on Machine Learning*, pp. 200–209.
- Joachims, T. (2003). Transductive learning via spectral graph partitioning. In *Proceedings of the 20th International Conference on Machine Learning*, pp. 290–297.
- Johnson, R., & Zhang, T. (2007). On the effectiveness of Laplacian normalization for graph semi-supervised learning. *Journal of Machine Learning Research*, 8, 1489–1517.
- Johnson, R., & Zhang, T. (2008). Graph-based semi-supervised learning and spectral kernel design. *IEEE Transactions on Information Theory*, 54(1), 275–288.
- Kakade, S., & Kalai, A. (2006). From batch to transductive online learning. In *Advances in Neural Information Processing Systems 18*, pp. 611–618.
- Karlen, M., Weston, J., Erkan, A., & Collobert, R. (2008). Large scale manifold transduction. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 448–455.
- Kearns, M., & Ron, D. (1999). Algorithmic stability and sanity-check bounds for leave-one-out cross-validation. *Neural Computation*, 11(6), 1427–1453.
- Kim, K.-H., & Choi, S. (2007). Neighbor search with global geometry: A min-max message passing algorithm. In *Proceedings of the 24th International Conference on Machine Learning*, pp. 401–408.
- Kindermann, R., & Snell, J. (1980). *Markov Random Fields and Their Applications*. American Mathematical Society.
- Koltchinskii, V. (2001). Rademacher penalties and structural risk minimization. *IEEE Transactions on Information Theory*, 47(5), 1902–1915.
- Koltchinskii, V., & Panchenko, D. (2002). Empirical margin distributions and bounding the generalization error of combined classifiers. *Annals of Statistics*, 30(1), 1–50.
- Kutin, S. (2002). Extensions to McDiarmid’s inequality when differences are bounded with high probability. Tech. rep. TR-2002-04, University of Chicago.
- Kutin, S., & Niyogi, P. (2002). Almost-everywhere algorithmic stability and generalization error. In *Proceedings of the 18th Conference in Uncertainty in Artificial Intelligence*, pp. 275–282.
- Lanckriet, G., Cristianini, N., Bartlett, P., Ghaoui, L. E., & Jordan, M. (2004). Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5, 27–72.

- Lane, T. (2006). A decision-theoretic, semi-supervised model for intrusion detection. In Maloof, M. (Ed.), *Machine learning and data mining for computer security: Methods and applications*, pp. 157–178. Springer-Verlag.
- Lawrence, N., & Jordan, M. (2006). Gaussian processes and the null-category noise model. In Chapelle, O., Scholkopf, B., & Zien, A. (Eds.), *Semi-supervised learning*, pp. 131–144. MIT Press.
- Le, Q., Smola, A., Gärtner, T., & Altun, Y. (2006). Transductive Gaussian process regression with automatic model selection. In *Proceedings of the 17th European Conference on Machine Learning*, pp. 306–317.
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.
- Ledoux, M. (2001). *The concentration of measure phenomenon*. Mathematical Surveys and Monographs 98. American Mathematical Society.
- Ledoux, M., & Talagrand, M. (1991). *Probability in Banach spaces*. Springer-Verlag.
- Leskes, B., & Torenvliet, L. (2008). The value of agreement a new boosting algorithm. *Journal of Computer and System Sciences*, 74(4), 557–586.
- Levin, A., Lischinskii, D., & Weiss, Y. (2004). Colorization using optimization. In *ACM Transactions on Graphics*, Vol. 23, pp. 689–694.
- Loeff, N., Forsyth, D., & Ramachandran, D. (2008). ManifoldBoost: Stagewise function approximation for fully-, semi- and un-supervised learning. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 600–607.
- Lovász, L., & Vempala, S. (2006). Simulated annealing in convex bodies and an  $O^*(n^4)$  volume algorithm. *Journal of Computer and System Sciences*, 72(2), 392–417.
- Macskassy, S., & Provost, F. (2007). Classification in networked data: A toolkit and a univariate case study. *Journal of Machine Learning Research*, 8, 935–983.
- Mahadevan, S. (2007). Adaptive mesh compression in 3D computer graphics using multiscale manifold learning. In *Proceedings of the 24th International Conference on Machine Learning*, pp. 585–592.
- Manku, G., Rajagopalan, S., & Lindsay, B. (1998). Approximate medians and other quantiles in one pass and with limited memory. In *SIGMOD*, Vol. 28, pp. 426–435.
- Mason, L., Baxter, J., Bartlett, P., & Frean, M. (2000). Boosting algorithms as gradient descent. In *Advances in Neural Information Processing Systems 12*, pp. 512–518.

- Maurey, B. (1979). Construction de suites symétriques. *Comptes Rendus Acad. Sci. Paris*, 288, 679–681.
- McAllester, D. (2003). Simplified PAC-Bayesian margin bounds. In *Proceedings of the 16th Annual Conference on Learning Theory*, pp. 203–215.
- McDiarmid, C. (1989). *Surveys in Combinatorics*, chap. “On the method of bounded differences”, pp. 148–188. Cambridge University Press.
- McDiarmid, C. (1998). Concentration. In Habib, M., McDiarmid, C., Ramirez, J., & Reed, B. (Eds.), *Probabilistic methods for algorithmic discrete mathematics*, pp. 195–248. Springer-Verlag.
- Meir, R., & Zhang, T. (2003). Generalization error bounds for Bayesian mixture algorithms. *Journal of Machine Learning Research*, 4, 839–860.
- Mukherjee, S., Niyogi, P., Poggio, T., & Rifkin, R. (2004). Statistical learning: Stability is sufficient for generalization and necessary and sufficient for consistency of empirical risk minimization. Tech. rep. AI Memo 2002–024, MIT.
- Nene, S., Nayar, S., & Murase, H. (1996). Columbia object image library (coil-100). Tech. rep. CUCS-006-96, Columbia University.
- Ng, A., Jordan, M., & Weiss, Y. (2002). On spectral clustering: Analysis and algorithms. In *Advances in Neural Information Processing Systems 14*.
- Nigam, K., McCallum, A., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning Journal*, 39(2/3), 103–134.
- Niyogi, P. (2008). Manifold regularization and semi-supervised learning: Some theoretical analyses. Tech. rep. TR-2008-01, University of Chicago.
- Pelckmans, K., Shawe-Taylor, J., Suykens, J., & Moor, B. D. (2007). Margin based transductive graph cuts using linear programming. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, pp. 363–370.
- Pelckmans, K., Suykens, J., & Moor, B. D. (2006). The kingdom-capacity of a graph: On the difficulty of learning a graph labeling. In *Proceedings of the International Workshop on Mining and Learning with Graphs*, pp. 189–196.
- Rakhlin, A., Mukherjee, S., & Poggio, T. (2005). Stability results in learning theory. *Analysis and Applications*, 3(4), 397–419.
- Rasmussen, C., & Williams, C. (2006). *Gaussian Processes for Machine Learning*. MIT Press.
- Rockafellar, R. (1970). *Convex Analysis*. Princeton University Press, Princeton, N.J.

- Schölkopf, B., Herbrich, R., & Smola, A. (2001). A generalized representer theorem. In *Proceedings of the 14th Annual Conference on Computational Learning Theory and 5th European Conference on Computational Learning Theory*, pp. 416–426.
- Schölkopf, B., & Smola, A. (2002). *Learning with kernels*. MIT Press.
- Schwaighofer, A., & Tresp, V. (2003). Transductive and inductive methods for approximate Gaussian process regression. In *Advances in Neural Information Processing Systems 15*, pp. 953–960.
- Seeger, M. (2000). Learning with labeled and unlabeled data. Tech. rep., University of Edinburgh.
- Serfling, R. (1974). Probability inequalities for the sum in sampling without replacement. *The Annals of Statistics*, 2(1), 39–48.
- Shin, H., Hill, N., & Rätsch, G. (2006). Graph based semi-supervised learning with sharper edges. In *Proceedings of the 17th European Conference on Machine Learning*, pp. 402–413.
- Sindhwani, V., Chu, W., & Keerthi, S. (2007). Semi-supervised Gaussian process classifiers. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pp. 1059–1064.
- Sindhwani, V., & Keerthi, S. (2006). Large scale semi-supervised linear SVMs. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 477–484.
- Sindhwani, V., & Rosenberg, D. (2008). An RKHS for multi-view learning and manifold co-regularization. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 976–983.
- Sinha, K., & Belkin, M. (2008). The value of labeled and unlabeled examples. In *Advances in Neural Information Processing Systems 20*, pp. 1361–1368.
- Szlam, A., Maggioni, M., & Coifman, R. (2008). Regularization on graphs with function-adapted diffusion processes. *Journal of Machine Learning Research*, 9, 1711–1739.
- Szummer, M., & Jaakkola, T. (2002). Partially labeled classification with Markov random walks. In *Advances in Neural Information Processing Systems 14*, pp. 945–952.
- Taira, H., & Haruno, M. (2001). Text categorization using transductive boosting. In *Proceedings of the 12th European Conference on Machine Learning*, pp. 454–465.
- Talagrand, M. (1995). Concentration of measure and isoperimetric inequalities in product spaces. *Publications Mathématiques de l’I.H.E.S.*, 81, 73–203.

- Talagrand, M. (2005). *Majorizing measures: The generic chaining*. Springer Verlag.
- Tong, S., & Koller, D. (2001). Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2, 45–66.
- Tong, W., & Jin, R. (2007). Semi-supervised learning by mixed label propagation. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, pp. 651–656.
- Tsang, I., & Kwok, J. (2007). Large-scale sparsified manifold regularization. In *Advances in Neural Information Processing Systems 19*, pp. 1401–1408.
- Ueffing, N., Haffari, G., & Sarkar, A. (2007). Transductive learning for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 25–32.
- Vapnik, V., & Chervonenkis, A. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16, 264–280.
- Vapnik, V., & Chervonenkis, A. (1991). The necessary and sufficient conditions for consistency of the method of empirical risk minimization. *Pattern Recognition and Image Analysis*, 1(3), 284–305. Translated from 1989 paper published in Russian.
- Vapnik, V. N. (1982). *Estimation of Dependences Based on Empirical Data*. Springer Verlag, New York. Translated from 1979 edition in Russian.
- Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley Interscience, New York.
- Vert, J.-P., & Yamanishi, Y. (2005). Supervised graph inference. In *Advances in Neural Information Processing Systems 17*, pp. 1433–1440.
- Wang, F., Wang, S., Zhang, C., & Winther, O. (2007). Semi-supervised mean fields. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, pp. 596–603.
- Wang, F., & Zhang, C. (2008). Label propagation through linear neighborhoods. *IEEE Transactions on Knowledge and Data Engineering*, 20(1), 55–67.
- Wang, H., Yan, S., Huang, T., Liu, J., & Tang, X. (2007). Transductive regression piloted by inter-manifold relations. In *Proceedings of the 24th International Conference on Machine Learning*, pp. 967–974.
- Wang, J. (2007). Efficient large margin semisupervised learning. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, pp. 588–595.
- Wang, J., Jebara, T., & Chang, S.-F. (2008). Graph transduction via alternating minimization. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 1144–1151.

- Wang, J., & Shen, X. (2007). Large margin semi-supervised learning. *Journal of Machine Learning Research*, 8, 1867–1891.
- Weston, J., Collobert, R., Sinz, F., Bottou, L., & Vapnik, V. (2006). On the inference with universum. In *Proceedings of the 23th International Conference on Machine Learning*, pp. 1009–1016.
- Weston, J., Leslie, C., Ie, E., Zhou, D., Elisseeff, A., & Nobble, W. (2005). Semi-supervised protein classification using kernel design. *Bioinformatics*, 21(15), 3241–3247.
- Weston, J., Perez-Cruz, F., Bousquet, O., Chapelle, O., Elisseeff, A., & Schölkopf, B. (2003). Feature selection and transduction for prediction of molecular bioactivity for drug design. *Bioinformatics*, 19(6), 764–771.
- Wu, M., & Schölkopf, B. (2007). Transductive classification via local learning regularization. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, pp. 628–635.
- Xu, Z., Jin, R., Zhu, J., King, I., & Lyu, M. (2008). Efficient convex relaxation for transductive support vector machine. In *Advances in Neural Information Processing Systems 20*, pp. 1641–1648.
- Yarovsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pp. 189–196.
- Yin, Y. (1995). Reducing the number of queries in self-directed learning. In *Proceedings of the 8th Annual Conference on Learning Theory*, pp. 128–135.
- Yu, S., Krishnapuram, B., Rosales, R., Steck, H., & Rao, R. (2008). Bayesian co-training. In *Advances in Neural Information Processing Systems 20*, pp. 1665–1672.
- Zhang, T., & Oles, F. (2000). A probability analysis on the value of unlabeled data for classification problems. In *Proceedings of the 20th International Conference on Machine Learning*, pp. 1191–1198.
- Zhao, B., Wang, F., & Zhang, C. (2008). CutS3SVM: A fast semi-supervised SVM algorithm. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 830–838.
- Zhou, D., Bousquet, O., Lal, T., Weston, J., & Schölkopf, B. (2004). Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, pp. 321–328.
- Zhou, D., & Burges, C. (2007). Spectral clustering and transductive learning with multiple views. In *Proceedings of the 24th International Conference on Machine Learning*, pp. 1159–1166.

- Zhou, D., Huang, J., & Schölkopf, B. (2005). Learning from labeled and unlabeled data on a directed graph. In *Proceedings of the 22nd International Conference on Machine Learning*, pp. 1036–1043.
- Zhou, D., Huang, J., & Schölkopf, B. (2007). Learning with hypergraphs: Clustering, classification, and embedding. In *Advances in Neural Information Processing Systems 19*, pp. 1601–1608.
- Zhou, D., Schölkopf, B., & Hofmann, T. (2005). Semi-supervised learning on directed graphs. In *Advances in Neural Information Processing Systems 17*, pp. 1633–1640.
- Zhou, Z.-H., & Li, M. (2007). Semi-supervised regression with co-training style algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 19(11), 1479–1493.
- Zhou, Z.-H., Zhan, D.-C., & Yang, Q. (2007). Semi-supervised learning with very few labeled training examples. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, pp. 675–680.
- Zhu, X. (2008). Semi-supervised learning literature survey. Tech. rep. 1530, University of Wisconsin - Madison.
- Zhu, X., Ghahramani, Z., & Lafferty, J. (2003). Semi-supervised learning using Gaussian fields and harmonic functions. In *Proceedings of the 20th International Conference on Machine Learning*, pp. 912–919.
- Zhu, X., Kandola, J., Lafferty, J., & Ghahramani, Z. (2006). Graph kernels by spectral transforms. In Chapelle, O., Schölkopf, B., & Zien, A. (Eds.), *Semi-supervised learning*, pp. 265–278. MIT Press.
- Zhu, X., & Lafferty, J. (2005). Harmonic mixtures: Combining mixture models and graph-based methods for inductive and scalable semi-supervised learning. In *Proceedings of the 22th International Conference on Machine Learning*, pp. 1052–1059.
- Zhu, X., Lafferty, J., & Ghahramani, Z. (2003). Semi-supervised learning: From Gaussian fields to Gaussian processes. Tech. rep. CMU-CS-03-175, Carnegie-Mellon University.



תיאוריה ומעשה בלמידה טרנסדוקטיבית

דמיטרי פצ'יוני



# תיאוריה ומעשה בלמידה טרנסדוקטיבית

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת התואר

דוקטור לפילוסופיה

דמיטרי פצ'יוני

הוגש לסנט הטכניון — מכון טכנולוגי לישראל

אוקטובר 2008

חיפה

תשרי תשס"ט



# תוכן עניינים

1	תקציר	
5	מבוא	1
5	רקע	1.1
7	יישומים של למידה טרנסדוקטיבית	1.2
7	סיווג טקסטים	1.2.1
7	עיבוד תמונות	1.2.2
8	דחיסה עם עיבוד מידע	1.2.3
9	שחזור גרפים	1.2.4
9	עיבוד שפות טבעיות	1.2.5
10	מבנה של התזה	1.3
11	סקירה של למידה טרנסדוקטיבית	2
11	תיאוריה של למידה טרנסדוקטיבית	2.1
11	שני מודלים של למידה טרנסדוקטיבית	2.1.1
13	חסמים עליונים	2.1.2
27	עקביות	2.1.3
28	חסמים תחתונים	2.1.4
30	קשר למודלים אחרים של למידה	2.1.5
31	מתי למידה טרנסדוקטיבית יותר טובה מלמידה אינדוקטיבית	2.1.6
32	מודלים של למידה שקשורים ללמידה טרנסדוקטיבית	2.1.7
36	סיכום	2.1.8
36	אלגוריתמים ללמידה טרנסדוקטיבית	2.2
37	שיטות של שוליים רחבות	2.2.1
39	שיטות מבוססות על גרפים	2.2.2
46	שיטות שמשלבות שוליים רחבות וגרפים	2.2.3
46	רגולריזציה לפי נפח	2.2.4
47	תהליכים גאוסיאניים	2.2.5
48	בוסטינג	2.2.6
49	שיטות מבוססות על מינימיזציה של חסמים	2.2.7
50	שיטות ממכניקה סטטיסטית	2.2.8
51	שיטות של למידה עצמית	2.2.9
53	שיטות מבוססות על הסכמה	2.2.10

54	בעיות סיבוכיות	2.2.11	
55	השוואה אמפירית של אלגוריתמים	2.2.12	
55	סיכום	2.2.13	
55	הוכחה של למה 1		2.3
57	אי שוויונות ריכוז לפונקציות מעל החלוקות		3
59	אי שוויון מבוסס על יציבות פרמוטציות חזקה		3.1
60	אי שוויון מבוסס על יציבות פרמוטציות חלשה		3.2
61	הערות סיכום		3.3
62	הוכחות		3.4
62	הוכחה של משפט 1	3.4.1	
63	הוכחה של למה 4	3.4.2	
65	הוכחה של משפט 2	3.4.3	
68	יציבות טרנסדוקטיבית		4
68	מבוא		4.1
68	עבודה קודמת		4.2
69	הגדרות		4.3
70	חסם מבוסס על יציבות אחידה		4.4
73	חסם מבוסס על יציבות חלשה		4.5
75	חסם מבוסס על יציבות חלשה		4.6
75	שערוך של קוונטייל	4.6.1	
76	אלגוריתם לשערוך של יציבות	4.6.2	
77	דוגמאות של שערוך של יציבות	4.6.3	
79	הערות סיכום		4.7
81	סיבוכיות ראדמכר טרנסדוקטיבית ויישומיה		5
81	מבוא		5.1
82	מחקר קודם	5.1.1	
83	הגדרות		5.2
83	מודל למידה	5.2.1	
84	סיבוכיות ראדמכר טרנסדוקטיבית	5.2.2	
86	חסם שגיאה אחיד מבוסס על סיבוכיות ראדמכר		5.3
86	אי שוויון ריכוז לקבוצה של וקטורים	5.3.1	
89	התכווצות של סיבוכיות ראדמכר	5.3.2	
90	חסם שגיאה והשוואה לתוצאות הקודמות	5.3.3	
93	ייצוג מטריציוני של אלגוריתמים טרנסדוקטיביים		5.4
94	חסם כללי על סיבוכיות ראדמכר טרנסדוקטיבית	5.4.1	
95	ייצוג מטריציוני בעזרת הגרעין	5.4.2	
97	חסמי מונטה-קרלו על סיבוכיות ראדמכר	5.4.3	
98	יישום: חסמים מפורשים לאלגוריתמים מסוימים		5.5
98	אלגוריתם של יואכימס	5.5.1	
99	אלגוריתם של בלקין ואחרים	5.5.2	

102	אלגוריתם של ז'ו ואחרים	5.5.3	
104	חסם לתערובת טרנסדוקטיבית	5.6	
105	הערות סיכום	5.7	
106	הוכחות	5.8	
106	הוכחה של למה 9	5.8.1	
108	הוכחה של למה 10	5.8.2	
112	הוכחה של למה	5.8.3	
113	הוכחה של למה 12	5.8.4	
114	הוכחות מפרק 5.5.2	5.8.5	
116	הוכחה של למה 15	5.8.6	
116	הוכחות מפרק	5.8.7	
119	שוליים רחבות מול נפח גדול בלמידה טרנסדוקטיבית	6	
119	מבוא	6.1	
120	מודל למידה טרנסדוקטיבי	6.2	
121	הסקה טרנסדוקטיבית לפי עקרון העוצמה הגדולה	6.3	
122	על ידע מוקדם ועוצמה	6.4	
123	עקרון הנפח הגדול	6.5	
124	למידה טרנסדוקטיבית לפי עקרון הנפח הגדול	6.6	
125	קירוב של נפח	6.6.1	
126	אלגוריתם של רגולריזציה נפח מקורבת	6.6.2	
127	פתרון אופטימלי של בעיית האופטימיזציה	6.7	
129	חסם שגיאה	6.8	
130	תוצאות ניסויים	6.9	
131	על הפרמטרים של האלגוריתם	6.9.1	
133	תוצאות	6.9.2	
134	ניתוח של התוצאות	6.9.3	
136	הערות סיכום	6.10	



# רשימת איורים

49	משפחת אלגוריתמים לבוסטינג	2.1
80	שערוכי יציבות ושגיאות אמיתיות	4.1
103	השוואה של חסמי ראדמכר טרנדסדוקטיביים	5.1
	ידע מוקדם מבוסס על שוליים רחבות מול ידע מוקדם מבוסס על נפח גדול	6.1
120	ויזואליזציה של מרחב ההיפוטזות	6.2
125	מבנה של הפונקציה	6.3
129	דיוק של וקטורים עצמיים	6.4
135	השוואה של אלגוריתמים	6.5

# רשימת טבלאות

133	תוצאות חיוביות	6.1
134	תוצאות שליליות	6.2

# תקציר

החיים המודרניים מאופיינים על ידי כמויות עצומות של נתונים. הנתונים מיוצרים על ידי מקורות שונים ובצורות שונות, למשל נתונים ביולוגיים, נתונים רפואיים, נתונים פיננסיים, נתונים על לקוחות ונתונים מתצפיות. כמו כן, כולנו צרכני נתונים, שנדרשים על בסיס יומי לסנן כמויות עצומות של נתונים כדי למצוא את המידע שנחוץ לנו. התחום של למידה ממוחשבת עוסק בניתוח חכם של נתונים. ניתוח כזה יכול לגלות מידע מוסתר וחוקיות, שיכולים להיות מאוד נחוצים לצרכני הנתונים. כמו כן ניתוח כזה יכול למכן תהליכים חוזרים. לדוגמה, הודות למערכת שמשתמשת בשיטות של למידה ממוחשבת כדי לזהות כתב יד, מיליוני צ'קים, שמופקדים לכספומטים בארצות הברית, מעובדים בצורה אוטומטית לחלוטין בלי שום התערבות של בני אדם.

תזה זאת עוסקת בלמידה ממוחשבת, בפרט בגישה של למידה מדוגמאות. בגישה הזאת הנתונים מיוצגים כאוסף של דוגמאות. האוסף הזה יכול להיות אינסופי. ישנם מספר מודלים של למידה מדוגמאות. המודל הכי נפוץ הינו מודל למידה מפוקחת. במודל הזה הלומד מקבל קבוצת אימון של דוגמאות מתויגות. הדוגמאות נדגמו מתוך ההתפלגות שלא ידועה ללומד. מטרת הלומד היא למצוא, על סמך קבוצת האי-מון, היפוטזה שתתייג בצורה מדויקת את הדוגמאות שיידגמו מתוך אותה התפלגות. כיוון שאוספים גדולים של דוגמאות מתויגות מאוד יקרים להשגה, בשנים האחרונות המודלים של למידה מפוקחת למחיצה ולמידה טרנסדוקטיבית נהיו מאוד נפוצים. בשני המודלים הללו הלומד לומד מתוך אוסף של דוגמאות מתויגות ואוסף של דוגמאות לא מתויגות. בדרך כלל בבעיות מעשיות מספר הדוגמאות המתויגות הרבה יותר קטן

ממספר הדוגמאות הלא מתויגות.

מודלים של למידה מפוקחת ולמידה מפוקחת למחיצה הינם סוגים שונים של למידה אינדוקטיבית. בלמידה אינדוקטיבית הלומד נדרש לייצר היפוטיזה כללית מתוך אוסף של מקרים פרטיים (דוגמאות). בלמידה טרנסדוקטיבית הלומד נדרש להעביר ידע מאוסף אחד של מקרים פרטיים לאוסף שני. באופן פורמלי, בלמידה טרנסדו-קטיבית הלומד מקבל כקלט מדגם מלא, המורכב מקבוצת אימון של נקודות מתויגות וקבוצת מבחן של נקודות לא מתויגות. המטרה של הלומד היא למצוא תיוגים מדוי-קים של נקודות המבחן. בניגוד למודל של למידה אינדוקטיבית, במודל הטרנסדו-קטיבי הלומד יודע מראש את קבוצת המבחן שלו ויכול להתאים את תהליך הלמידה שלו בשביל להשיג ביצועים יותר טובים ספציפית על הקבוצה הזו. מודל למידה טר-נסדוקטיבית הוצג לראשונה בשנת 1971 על ידי לבהיסיע בנימיכ ולרוש טועצ'נומכיץ. המוטיבציה של המודל הטרנסדוקטיבי היתה תיאורטית לחלוטין. אך לאחר מכן התב-רר שבמציאות קיים מספר רב של בעיות למידה נושאות אופי טרנסדוקטיבי. דהיינו בבעיות האלה קבוצת המבחן ידועה ללומד כבר בשלב הלמידה.

בתזה הזאת אנחנו מניחים שהלומד פועל בצורה הבאה. בשלב ראשון הלומד יוצר היפוטיזה טרנסדוקטיבית רכה, שנותנת לכל נקודה במדגם המלא תיוג רציף. בשלב שני הלומד מפעיל פונקציית ייחמ על התיוגים הרציפים ומקבל תיוגים בינאריים של הנקודות. ההנחה הזאת על צורת העבודה של אלגוריתם טרנסדוקטיבי היא כמעט ולא מגבילה, כי כמעט כל האלגוריתמים הטרנסדוקטיביים שקיימים כיום פועלים בצורה הזאת. אנחנו נקרא לאוסף של כל ההיפוטיזות הרכות שהלומד מייצר, כאשר הוא מורץ על כל החלוקות האפשריות של מדגם המלא לקבוצת האימון וקבוצת המבחן, בשם 'מרחב ההיפוטיזות הרכות של האלגוריתם'. מרחב ההיפוטיזות הקשות מוגדר בצורה דומה.

כל לומד אינדוקטיבי מסוגל לבצע למידה טרנסדוקטיבית. זאת כי בעזרת היפוטיזה כללית שמיוצרת ל ידי הלומד האינדוקטיבי ניתן לתייג גם את הנקודות של קבוצת המבחן. בצורה הזאת הלמידה נעשית בשני שלבים שהם לימוד של היפוטיזה כללית

והפעלתה בשביל לתייג נקודות המבחן. המטרה של מודל הלמידה הטרנסדוקטיבית היא לתייג את נקודות המבחן בשלב אחד על ידי העברת ידע על תיוגים מנקודות האימון לנקודות המבחן.

בתזה הזאת אנו מציגים מספר תוצאות ללמידה טרנסדוקטיבית, כאשר ההתמ-קדות היא גם בפן התיאורטי וגם פן האמפירי. המטרה של התזה היא לחבר בין תוצאות תיאורטיות בתורת למידה לבין האלגוריתמים המעשיים שנפוצים כיום. התוצאות של התזה כוללות פיתוח של אבטחות ביצועים לאלגוריתמים קיימים וגם פיתוח אלגור-יתמים חדשים על סמך אבטחות ביצועים כלליות.

בחלק הראשון והתיאורטי של התזה אנחנו מפתחים חסמי שגיאה כלליים לאל-גוריתמים טרנסדוקטיביים. כדי לפתח את החסמים הללו השתמשנו בשיטת המר-טינגלים ופיתחנו מספר אי שוויונות ריכוז לפונקציות מעל חלוקות של מספר סופי של איברים. אי שוויונות ריכוז האלה מבוססים על אי שוויון תצ"ב.

בהתבסס על אי שוויונות ריכוז אנחנו מפתחים חסמי שגיאה כלליים לאלגוריתמ-ים טרנסדוקטיביים. החסמים האלה חוסמים מלמעלה את השגיאה הממוצעת של אלגוריתם על נקודות המבחן בעזרת סכום משוקלל של שגיאה ממוצעת של אלגור-יתם על נקודות האימון וגורם היתרה. גורם היתרה תלוי בגודל של קבוצת האימון, בגודל של קבוצת המבחן ובגודל של מרחב ההיפוטאות הרכות של האלגוריתם. בחסמ-ים שלנו הגודל של מרחב ההיפוטאות הרכות נמדד על ידי הגרסאות הטרנסדוקטיביות של יציבות אחידה ויציבות חלשה, וכמו כן גם על ידי סיבוכיות בהוסבדטוע טרנסדו-קטיבית. יציבות של אלגוריתם מודדת את הרגישות של הפלט שלו לשינויים קטנים בקלט. יציבות אחידה מודדת את הרגישות במקרה הכי גרוע. יציבות חלשה מודדת את היציבות ברוב המקרים, אבל לא במקרה הכי גרוע. סיבוכיות בהוסבדטוע מודדת את הקורלציה של מחלקת ההיפוטאות עם רעש אקראי.

אנחנו מציגים מספר שיטות כלליות לחסימת היציבות וסיבוכיות בהוסבדטוע. שי-טות חסימה האלה ישימות למספר של משפחות של אלגוריתמים טרנסדוקטיביים. שיטות שונות משלימות אחת את השנייה. לכל משפחה של אלגוריתמים ישנה שיטת

חסימה אחת שנוחה לשימוש ושאר השיטות פחות נוחות. אנחנו מציגים מספר דוגמאות של פיתוח חסמים עבור אלגוריתמים ספציפיים. החסמים שמתקבלים הם מפורשים ותלויי נתונים. למיטב ידיעתנו, החסמים שלנו במונחים יציבות חלשה ובמונחים של סיבוכיות בהוסבדטוע טרנסדוקטיבית הם חסמים ראשוניים מסוגם.

בחלק השני והמעשי של התזה אנחנו חוקרים את עיקרון חדש בלמידה הטרנסדוקטיבית, שנקרא עקרון הנפח הגדול. בהינתן אוסף של היפותזות רכות, המדגם המלא מנפץ מחלקת היפוטזות לאוסף סופי של מחלקות שקילות. כל ההיפוטזות הרכות באותה מחלקת שקילות נותנות אותנו תיוג קשיח לכל הדוגמאות במדגם המלא. העי-קרון של הנפח הגדול נותן עדיפות למחלקות שקילות טרנסדוקטיביות לפי הנפח שהן תופסות במרחב ההיפוטזות. החישוב המקורב של הנפח נעשה בעזרת פירוש גיאומטרי של מרחב ההיפוטזות הרכות. אלגוריתם למידה טרנסדוקטיבית שמתקבל הינו בעיית אופטימיזציה לא קמורה. למרות האי-קמירות לבעיה הזאת קיים פתרון יעיל שמוצא מינימום גלובלי בזמן קצר. אנחנו משווים את האלגוריתם שלנו מול האלגוריתם שמממש את עיקרון הלמידה של שוליים רחבות. העיקרון של שוליים רחבות הוא עיקרון הלמידה הכי נפוץ כיום. ההשוואה נעשית על פני מספר גדול של אוספי נתונים שכוללים בפרט טקסטים ותמונות. תוצאות ההשוואה מראות יתרון משמעותי של האלגוריתם שלנו בתחומים מסוימים (כמו למשל טקסט ותמונות) וחסרון משמעותי בתחומים אחרים (כמו למשל אוספים מתוך המאגר של ).

התזה משאירה מספר שאלות מחקר פתוחות. השאלה המרכזית שנשארת עדיין פתוחה היא מתי ובאיזה תנאים למידה טרנסדוקטיבית יותר יעילה מאשר למידה אינדוקטיבית. כדי לוודא עד כמה החסמים העליונים שלנו הדוקים נדרש פיתוח של חסמי שגיאה תחתונים. לבסוף, שאלת מחקר מעניינת היא לנסות לאפיין מקרים בהם הלמידה לפי עיקרון הנפח הגדול היא יותר יעילה מאשר הלמידה לפי העיקרון של השוליים הרחבות.