Dataset Shift

Adaptive Computation and Machine Learning

Thomas Dietterich, Editor

Christopher Bishop, David Heckerman, Michael Jordan, and Michael Kearns, Associate Editors

Bioinformatics: The Machine Learning Approach, Pierre Baldi and Søren Brunak

Reinforcement Learning: An Introduction, Richard S. Sutton and Andrew G. Barto

Graphical Models for Machine Learning and Digital Communication, Brendan J. Frey

Learning in Graphical Models, Michael I. Jordan

Causation, Prediction, and Search, second edition, Peter Spirtes, Clark Glymour, and Richard Scheines

Principles of Data Mining, David Hand, Heikki Mannila, and Padhraic Smyth

Bioinformatics: The Machine Learning Approach, second edition, Pierre Baldi and Søren Brunak

Learning Kernel Classifiers: Theory and Algorithms, Ralf Herbrich

Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, Bernhard Schölkopf and Alexander J. Smola

Dataset Shift

Joaquin Quiñonero Candela
Masashi Sugiyama
Anton Schwaighofer
Neil D. Lawrence

The MIT Press
Cambridge, Massachusetts
London, England

Typeset by the authors using LaTeX 2$_\varepsilon$
Library of Congress Control No. xxx
Printed and bound in the United States of America

Library of Congress Cataloging-in-Publication Data

Contents

1 An Adversarial View of Covariate Shift and A Minimax Approach 9
1.1 Minimax Problem Formulation 11
1.2 Finding the Minimax Optimal Features 12
1.3 A Convex Dual for the Minimax Problem 17
1.4 An Alternate Setting: Uniform Feature Deletion 18
1.5 Related Frameworks 19
1.6 Experiments 21
1.7 Discussion & Conclusions 25

References 27

Index 29

Notation 29

Notation and Symbols 29

Preface

Joaquin Quiñonero Candela
Masashi Sugiyama
Anton Schwaighofer
Neil D. Lawrence

Overview

Dataset shift is a challenging situation where the joint distribution of inputs and outputs differs between the training and test stages. Covariate shift is a simpler particular case of dataset shift where only the input distribution changes (covariate denotes input), while the conditional distribution of the outputs given the inputs $p(y|x)$ remains unchanged. Dataset shift is present in most practical applications for reasons ranging from the bias introduced by experimental design, to the mere irreproducibility of the testing conditions at training time. For example, in an image classification task, training data might have been recorded under controlled laboratory conditions, whereas the test data may show different lighting conditions. In other applications, the process that generates data is in itself adaptive. Some of our authors consider the problem of spam email filtering: successful "spammers" will try to build spam in a form that differs from the spam the automatic filter has been built on.

Dataset shift seems to have raised relatively little interest in the machine learning community until very recently. Indeed, many machine learning algorithms are based on the assumption that the training data is drawn from exactly the same distribution as the test data on which the model will later be evaluated. Semi-supervised learning and active learning, two problems that seem very similar to covariate shift have received much more attention. How do they differ from covariate shift? Semi-supervised learning is designed to take advantage of unlabeled data present at training time, but is not conceived to be robust against changes in the input distribution. In fact, one can easily construct examples of covariate shift for which common SSL strategies such as the "cluster assumption" will lead to disaster. In active learning the algorithm is asked to select from the available unlabeled inputs those for which obtaining the label will be most beneficial for learning. This is very relevant in contexts where labeling data is very costly, but active learning strategies

are not specifically design for dealing with covariate shift.

This book attempts to give an overview of the different recent efforts that are being made in the machine learning community for dealing with dataset and covariate shift. The contributed chapters establish relations to transfer learning, transduction, local learning, active learning and to semi-supervised learning. Three recurrent themes are how the *capacity* or complexity of the model affects its behaviour in the face of dataset shift —are "true" conditional models and sufficiently rich models unaffected?— whether it is possible to find projections of the data that attenuate the differences in the training and test distributions while preserving predictability, and whether new forms of importance re-weighted likelihood and cross-validation can be devised which are robust to covariate shift.

The idea of compiling this book was born during the NIPS*06 Workshop on *Learning when test and training inputs have different distributions* we organized. The majority of the chapter authors either gave a talk or were present at the workshop; the few that weren't have made major contributions to dealing with dataset shift in machine learning. Thank you so much to all of you for making this volume happen!

Structure of the Book

The book is divided into four major parts:

1. **Introduction to Dataset shift**
   - Amos Storkey, *When training and test sets are different: characterising learning transfer*
   - David Corfield, *Projection and projectability*

Amos Storkey and David Corfield provide a mathematical and a philosophical introduction respectively to the problem of dataset shift. Storkey provides a unifying framework for different cases of dataset shift. Corfield starts from a philosophical perspective and ends comparing the frequentist to the Bayesian approach in machine learning. Which seems more promising to attack covariate shift?

2. **Theoretical Views on Dataset and Covariate Shift**
   - Matthias Hein, *Binary classification under sample selection bias*
   - Lars Kai Hansen, *On Bayesian transduction – Implications for the 'covariate shift' problem*
   - Shai Ben-David, *Data representation framework addressing the training/test distributions gap*

Matthias Hein discusses from a decision theoretic point of view the conditions under which dataset shift does not affect the performance of a Bayes classifier. To deal with the cases where these conditions are not met, he proposes a graph-based robust regularization method. Lars Kai Hansen shows that Bayesian transductive

learning is generalization optimal, and studies the generalizability of the conditional predictive distribution under covariate shift. Shai Ben-David proposes a taxonomy of methods for different sub-classes of goals within dataset shift. He proposes a framework based on appropriate feature representations of the data to alleviate the differences between distributions, and derives bounds on the performance relative to the best achievable.

3. **Algorithms for Covariate Shift**

   - Takafumi Kanamori and Hidetoshi Shimodaira, *Geometry of covariate shift with applications to active learning*
   - Masashi Sugiyama, Neil Rubens and Klaus-Robert Müller, *Model selection, active learning, and covariate shift*
   - Quoc Le, Alex Smola, Arthur Gretton, Jiayuan Huang, Karsten Borgwardt and Bernhard Schölkopf, *Covariate shift and local learning by distribution matching*
   - Steffen Bickel, Michael Brückner and Tobias Scheffer *Discriminative learning for differing training and test distributions*
   - Amir Globerson, Choon-Hui Teo, Alex Smola and Sam Roweis, *An adversarial view of covariate shift and a minimax approach*

Takafumi Kanamori and Hidetoshi Shimodaira analyze covariate shift from an information geometric point of view, and provide information criteria that can be used for model selection and active learning. Sugiyama and co-workers also discuss model selection and active learning in light of covariate shift, but in a data-dependent framework. They also provide a variant of cross-validation for covariate shift adaptation and show its usefulness for brain-computer interfaces. Quoc Le and co-workers provide a method of directly estimating importance weights without going through explicit density estimation, and discuss the relation to local learning paradigms. Steffen Bickel and co-workers provide a method for learning under covariate shift that is purely discriminative. It maximizes an integrated optimization criterion that is directly linked to the expected loss under the test distribution. They evaluate the method on spam filtering and other applications. Amir Globerson and co-workers address the situation where training and test data differ by adversarial feature corruption (for example deletion) and provide a robust learning method that can be computed efficiently. They demonstrate its usefulness for spam filtering.

4. **Discussion**

   - All editors and authors: **we want to organize a discussion among chapter authors and editors during NIPS2007! We will keep you posted**

In the discussion section the authors and the editors exchange their ideas about dataset shift: in what situations is it a problem, what theoretical perspectives are adquate, and what families of algorithms seem promising. The discussion should take the form of the transcript of a conversation.

Detailed Contents Overview

Part I of the book aims at providing a general introduction to the problem of learning when training and test distributions differ in some form.

Amos Storkey provides a general introduction in Chapter **??** from the view point of learning transfer. He introduces the general learning transfer problem, and formulates the problem in terms of a change of scenario. Standard regression and classification models can be characterized as conditional models. Assuming that the conditional model is true, covariate shift is not an issue. However, if this assumption does not hold, conditional modelling will fail. Storkey then characterizes a number of different cases of dataset shift, including simple covariate shift, prior probability shift, sample selection bias, imbalanced data, domain shift and source component shift. Each of these situations is cast within the framework of graphical models and a number of approaches to address each of these problems is reviewed. Storkey also introduces a framework for multiple dataset learning, that also prompts the possibility of using hierarchical dataset linkage.

Dataset shift has wider implications beyond machine learning, within philosophy of science. David Corfield, Chapter **??**, shows how the problem of dataset shift has been addressed by different philosophical schools under the concept of 'projectability'. When philosophers tried to formulate scientific reasoning with the resources of predicate logic and a Bayesian inductive logic, it became evident how vital background knowledge is to allow us to project confidently into the future, or to a different place, from previous experience. To transfer expectations from one domain to another, it is important to locate robust causal mechanisms. An important debate concerning these attempts to characterise background knowledge is over whether it can all be captured by probabilistic statements. Having placed the problem within the wider philosophical perspective, Corfield turns to machine learning, and addresses a number of questions: Have machine learning theorists been sufficiently creative in their efforts to encode background knowledge? Have the frequentists been more imaginative than the Bayesians, or vice-versa? Is the necessity of expressing background knowledge in a probabilistic framework too restrictive? Must relevant background knowledge be hand-crafted for each application, or can it be learned?

Part II of the book focuses on theoretical aspects of dataset and covariate shift, with contributions by Matthias Hein, Lars Kai Hansen, and Shai Ben-David.

In Chapter **??**, Matthias Hein studies the problem of binary classification under sample selection bias from a decision theoretic perspective. Starting from a derivation of the necessary and sufficient conditions for equivalence of the Bayes classifiers of training and test distributions, Hein provides the conditions under which –asymptotically– sample selection bias does not affect the performance of a classifier. From this viewpoint, there are fundamental differences between classifiers of low and high capacity, in particular the ones which are Bayes consistent. In the second part of his chapter, Hein provides means to modify existing learning algo-

rithms such that they are more robust to sample selection bias in the case where one has access to an unlabeled sample of the test data. This is achieved by constructing a graph-based regularization functional. The close connection of this approach to semi-supervised learning is also highlighted.

Lars Kai Hansen provides a Bayesian analysis of the problem of covariate shift in Chapter **??**. He approaches the problem starting with the hypothesis that it is possible to recover performance by tracking the non-stationary input distribution. Under the average log-probability loss, Bayesian transductive learning is generalization optimal (in terms of the conditional distribution $p(\text{label} \,|\, \text{input})$). For realizable supervised learning –where the 'true' model is at hand– all available data should be used in determining the posterior distribution, including unlabeled data. However, if the parameters of the input distribution are disjoint of those of the conditional predictive distribution, learning with unlabeled data has no effect on the supervised learning performance. For the case of unrealizable learning –the 'true' model is not contained in the prior– Hansen argues that 'learning with care' by discounting some of the data might improve performance. This is reminiscent of the importance weighting approaches of Kanamori *et al.* and Sugiyama *et al.*

In Chapter **??**, the third contribution of the theory part, Shai Ben-David provides a theoretical analysis based around 'domain adaptation': an embedding into a feature space under which training and test distribution appear similar, and where enough information is preserved for prediction. This relates back to the general view-point of Corfield in Chapter **??**, who argues that learning transfer is only possible once a robust (invariant) mechanism has been identified. Ben-David also introduces a taxonomy of formal models for different cases of dataset shift. For the analysis, he derives error bounds which are relative to the best possible performance in each of the different cases. In addition, he established a relation of his framework to inductive transfer.

Part III of the book focuses on algorithms to learn under the more specific setting of covariate shift, where the input distribution changes between training and test phases but the conditional distribution of outputs given inputs remains unchanged.

Chapter **??**, contributed by Takafumi Kanamori and Hidetoshi Shimodaira, starts with showing that the ordinary maximum likelihood estimator is heavily biased under covariate shift if the model is *misspecified*. By misspecified it is meant that the model is too simple to express the target function (see also Chapter **??** and Chapter **??** for the different behavior of misspecified and correct models). Kanamori and Shimodaira then show that the bias induced by covariate shift can be asymptotically cancelled by weighting the training samples according to the importance ratio between training and test input densities. However, the weighting is suboptimal in practical situations with finite samples since it tends to have larger variance than the unweighted counterpart. To cope with this problem, Kanamori and Shimodaira provide an information criterion that allows to optimally control the bias-variance trade-off. The latter half of their contribution focuses on the problem of active learning where the covariate distribution is designed by users for better prediction performances. Within the same information-criterion framework,

they develop an active learning algorithm that is guaranteed to be consistent.

In Chapter **??** Masashi Sugiyama and co-workers also discuss the problems of model selection and active learning in the covariate shift scenario, but in a slightly different framework; the conditional expectation of the generalization error given training inputs is evaluated here, while Kanamori and Shimodaira's analysis is in terms of the full expectation of the generalization error over training inputs and outputs. Sugiyama and co-workers argue that the conditional expectation framework is more data-dependent and thus more accurate than the methods based on the full expectation, and develop alternative methods of model selection and active learning for approximately linear regression. An algorithm that can effectively perform active learning and model selection at the same time is also provided. Sugiyama *et al.* also develop an importance-weighted cross-validation method, which can be applied to model selection under any loss functions, including classification. The effectiveness of the weighted cross-validation method is demonstrated in an application for brain-computer interfaces.

In Chapter **??**, third of the algorithms part, Quoc Le and co-workers address the problem of distribution matching between training and test stages, which is similar in spirit to the problem discussed in Chapter **??**. They propose a method called *kernel mean matching*, which allows to directly estimate the importance weight *without* going through density estimation. Le *et al.* then relate the re-weighted estimation approaches to *local learning*, where labels on test data are estimated given a subset of training data in a neighborhood of the test point. Examples are nearest neighbour estimators and Watson-Nadaraya type estimators. The authors further provide detailed proofs concerning the statistical properties of the kernel mean matching estimator and detailed experimental analyses for both covariate shift and local learning.

In Chapter **??** Steffen Bickel and co-workers derive a solution to covariate shift adaptation for arbitrarily different distributions that is purely discriminative: neither training nor test distribution are modeled explicitly. They formulate the general problem of learning under covariate shift as an integrated optimization problem and instantiate a kernel logistic regression and an exponential loss classifier for differing training and test distributions. They show under which condition the optimization problem is convex, and empirically study their method on problems of spam filtering, text classification, and landmine detection.

Amir Globerson and co-workers take an innovative view on covariate shift: in Chapter 1 they address the situation where training and test inputs differ by adversarial *feature corruption*. They formulate this problem as a two player game, where the action of one player (the one who builds the classifier) is to choose robust features, whereas the other player (the adversary) tries to corrupt the features which would harm the current classifier most at test time. Globerson *et al.* address this problem in a minimax setting, thus avoiding any modelling assumptions about the deletion mechanism. They use convex duality to show that it corresponds to a quadratic program and show how recently introduced methods for large scale online optimization can be used for fast optimization of this quadratic problem.

Finally, the authors apply their algorithm to handwritten-digit recognition and spam filtering tasks, and show that it outperforms a standard SVM when features are deleted from data samples.

The final Part IV of the book is a discussion. It is an open dialogue between the authors and editors, where personal opinions can be stateed, and research statements can be discussed. Is dataset shift a relevant problem, in which cases, which theoretical approaches seem promising, what algorithms seem powerful?

1 An Adversarial View of Covariate Shift and A Minimax Approach

Amir Globerson
Choon Hui Teo
Alex Smola
Sam Roweis

When constructing classifiers over high dimensional spaces such as texts or images, one is inherently faced with the problem of under-sampling of the true data distribution. Even so-called "discriminative" methods which focus on minimizing classification error (or a bound on it) are exposed to this difficulty since the training objective will be calculated over the observed input vectors only, and thus may not be a good approximation of the average objective on the test data. This is especially important in settings such as document classification where features may take on certain observed values (e.g. a zero count for a particular vocabulary item) due to small sample effects. A more serious difficulty may arise when dataset shift effects are present, namely when the training and testing distributions are different. For example, the distribution of words in spam email changes very rapidly and keywords which are highly predictive of class in the training set may not be indicative or even present in the test data. As another example, consider a digital camera whose output is fed to a face recognition system. Due to hardware or transmission failures, a few pixels may "die" over the course of time. In the image processing literature, this is referred to as *pepper* noise [Bovik et al., 2000] (*salt* noise refers to the case when pixels values are clipped to some fixed value). Any classifier which attached too much weight to any single pixel would suffer a substantial performance loss in this case. As a final example, consider a network of local processing elements in an artificial sensor network or a biological network such as the cortex. The hardware/wetware of such systems is known to be extremely unreliable (thousands of neurons die each day) and yet the overall architecture maintains its function, indicating a remarkable robustness to such non-stationarities in its input.

All the above examples describe a scenario where features that were present when constructing the classifier (i.e., in the training data), are potentially *deleted* at some future point in time. Such deletion may manifest itself differently depending on the particular domain: a deleted feature may be known to be unavailable or unmeasured; it may take on random values; or its value may be set to some constant. In our

formal treatment, we focus on the case where deletion corresponds to setting the feature's value to zero. Indeed, in the examples given above this is an appropriate description.

Of course, when constructing the classifier, we cannot anticipate in advance which features may be deleted in the future. One possible strategy is to analyze the performance under random deletion of features. However, this may not be a correct model of the deletion statistics. The approach we take here is to construct a classifier which is optimal in the worst case deletion scenario, thus avoiding any modeling assumptions about the deletion mechanism. This can be formulated as a two player game, where the action of one player (the classifier builder) is to choose robust classifier parameters, whereas the other player (the feature removal mechanism) tries to delete the features which would be most harmful given the current classifier. We note that the adversarial setting may not necessarily be an exact model of the problem (e.g., spam authors may not know the details of the spam filter, and are thus not as powerful as the adversary we model). However, considering the worst case scenario yields a classifier that is robust to any adversarial strategy, and avoids making statistical assumptions about the deletion process. Furthermore, even if there is no true underlying adversary, robustness to feature deletion yields robustness of the resulting classifier, in the sense that it will not attach too much weight to single features, even if those appear informative at training time.

Robust minimax approaches to learning classifiers have recently attracted interest in the machine learning community [Lanckriet et al., 2004, El Ghaoui et al., 2003, Kim et al., 2006]. Our approach is related to El Ghaoui et al. [2003] where the location of sample points is only known up to an ellipsoidal region, and a classifier that is optimal in the worst case is sought. However, in our case, the structure of uncertainty is inherently different and is related to the existence vs. non-existence of a feature. Adversarial models have also recently been studied in the context of spam filtering by Dalvi et al. [2004]. Their formalism addresses transformations that are more general than feature deletion, and also incorporates costs for different types of mistakes. However, finding the optimal strategy in their case is a computationally hard problem, and approximations are needed.

In the context of dataset shift, our minimax approach assumes that the difference between training and testing scenarios is defined via a class of possible transformations (here we consider feature deletions), and that learning should be robust with respect to this class.

In Section 1.1 we formalize the feature dropping minimax game for classifiers such as the support vector machine [Schölkopf and Smola, 2002] in which the training objective is measured using a regularized hinge loss. We denote this optimization problem by the name FDROP. We next show that this problem can be exactly solved in polynomial time, and provide several optimization algorithms for solving it. Finally, we illustrate the method's performance on handwritten digit recognition and spam filtering tasks.

1.1 Minimax Problem Formulation

Given a labeled sample $(\mathbf{x}_i, y_i)$ $(i = 1, \ldots, n)$, with input feature vectors $\mathbf{x}_i \in \mathbb{R}^d$ and class labels[1] $y_i \in \{\pm 1\}$, we would like to construct classifiers which are robust to *deletion* of features. We focus on the case where a feature is assigned the value of zero if it is deleted, and denote by K the number of features the adversary can delete for any given sample point $\mathbf{x}$. The number K is assumed to be given and fixed in what follows, although in practice we set it using cross-validation.

In standard support vector machines (e.g., see Schölkopf and Smola [2002]), the goal of the learning algorithm is to find a weight vector $\mathbf{w} \in \mathbb{R}^d$ that minimizes a regularized hinge loss:

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i [1 - y_i \mathbf{w} \cdot \mathbf{x}_i]_+ \ , \tag{1.1}$$

where we use the notation $[x]_+ = \max\{x, 0\}$. However, in the feature deletion case, the adversary may change the input $\mathbf{x}_i$ by deleting features from it. We would like our classifier to be robust to such deletions. Thus, we seek a classifier which minimizes the worst case hinge loss when K features may be deleted from each data vector. In this setting, the worst case hinge loss for example i is given by

$$\begin{array}{rlll} h^{wc}(\mathbf{w}, y_i\mathbf{x}_i) = & \max & [1 - y_i \mathbf{w} \cdot (\mathbf{x}_i \circ (1 - \boldsymbol{\alpha}_i))]_+ \\ & s.t. & \boldsymbol{\alpha}_i \in \{0,1\}^d \\ & & \sum_j \alpha_{ij} = K \end{array} \tag{1.2}$$

where α_{ij} denotes the j^{th} element of $\boldsymbol{\alpha}_i$, and is equal to 1 if the j^{th} feature of $\mathbf{x}_i$ is deleted (we use $\circ$ to denote the element-wise multiplication operation).

The worst case hinge loss over the entire training set is $\sum_i h^{wc}(\mathbf{w}, y_i\mathbf{x}_i)$. The overall optimization problem, which we denote by FDROP, is then

$$\underline{\text{FDROP:}} \qquad \mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i h^{wc}(\mathbf{w}, y_i\mathbf{x}_i) \ . \tag{1.3}$$

The above can be explicitly written as a minimax optimization problem:

$$\begin{array}{lll} \min\limits_{\mathbf{w}} & \max\limits_{\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_n} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i [1 - y_i \mathbf{w} \cdot (\mathbf{x}_i \circ (1 - \boldsymbol{\alpha}_i))]_+ \\ \text{s.}t. & \boldsymbol{\alpha}_i \in \{0,1\}^d \\ & \sum_j \alpha_{ij} = K \end{array} \tag{1.4}$$

Denote the objective of the above by $f(\mathbf{w}, \boldsymbol{\alpha})$. Then Equation 1.4 may be interpreted as finding an optimal strategy for a zero-sum game where the learning algorithm is payed $-f(\mathbf{w}, \boldsymbol{\alpha})$ and the adversary is payed $f(\mathbf{w}, \boldsymbol{\alpha})$ when the joint action $\mathbf{w}, \boldsymbol{\alpha}$ is

1. We focus on the binary case here. All results can be easily generalized to the multi class case.

taken.

In the next section we present two approaches to solving the optimization problem in Equation 1.3.

1.2 Finding the Minimax Optimal Features

The minimization problem in Equation 1.3 is closely related to the SVM optimization problem. However, in our case we have a worst case hinge loss instead of the standard hinge loss. Since this worst case requires maximization over $\binom{n}{k}$ possibilities per sample, it is not immediately clear how to design an efficient method for solving the overall optimization. In the following section we describe two methods for solving FDROP. The first is to use convex duality transformations to turn it into a quadratic program with $O(nd)$ variables. The second is to solve it directly in the $\mathbf{w}$ variable using the recently introduced BMRM method [Teo et al., 2007b].

1.2.1 An Equivalent Quadratic Program

In this section we show that the problem in Equation 1.3 is equivalent to a certain convex quadratic program. We begin by analyzing the worst case hinge loss $h^{wc}(\mathbf{w}, y_i\mathbf{x}_i)$. For a given $\mathbf{w}$, this loss can be seen to be minimized when $\boldsymbol{\alpha}_i$ is chosen to delete the K features x_{ij} with highest values $y_i w_j x_{ij}$, since these will have the strongest decreasing effect on the loss. Thus we can rewrite $h^{wc}(\mathbf{w}, y_i\mathbf{x}_i)$ as

$$h^{wc}(\mathbf{w}, y_i\mathbf{x}_i) \quad = \quad [1 - y_i\mathbf{w}\cdot\mathbf{x}_i + s_i]_+ \ ,$$

where we have defined

$$s_i = \max_{\boldsymbol{\alpha}_i\in\{0,1\}^d, \sum_j \alpha_{ij}=K} y_i\mathbf{w}\cdot(\mathbf{x}_i\circ\boldsymbol{\alpha}_i) \tag{1.5}$$

as the maximum contribution of K features to the margin of sample $\mathbf{x}_i$.

To simplify the expression for s_i, we note that the integer constraint on the variables $\boldsymbol{\alpha}_i$ may be relaxed to $0 \le \boldsymbol{\alpha}_i \le 1$ without changing the optimum. This is true since the vertices of the resulting $2d+1$ linear constraints are integral. Since the maximization (with respect to $\boldsymbol{\alpha}_i$) is over a linear function, the optimum will be at the vertices, and is therefore integral. We rewrite s_i using this relaxation, and also changing the order of multiplication

$$\begin{aligned} s_i = \quad & \max \quad y_i\,(\mathbf{w}\circ\mathbf{x}_i)\cdot\boldsymbol{\alpha}_i \\ & s.t. \quad 0 \le \boldsymbol{\alpha}_i \le 1 \\ & \qquad \textstyle\sum_j \alpha_{ij} = K \ . \end{aligned} \tag{1.6}$$

The above expression is bilinear in $\boldsymbol{\alpha}_i$ and $\mathbf{w}$. Since this may potentially contribute a non-convex factor into the optimization, we use a duality transformation with

respect to the $\boldsymbol{\alpha}_i$ variables to avoid bilinearity. An important outcome of using a duality transformation is that a minimization problem is obtained so that the original minimax problem is turned into a minimization problem in the new variables. Note that the above problem is linear in $\boldsymbol{\alpha}_i$ so that the value of the dual will exactly equal that of s_i.[2]

Denoting the dual variables by $\mathbf{v}_i \in \mathbb{R}^d, z_i \in \mathbb{R}$, we obtain the dual of the maximization in Equation 1.6

$$\begin{aligned} s_i = \quad & \min \quad K z_i + \textstyle\sum_j v_{ij} \\ & s.t. \quad z_i + \mathbf{v}_i \geq y_i \mathbf{x}_i \circ \mathbf{w} \\ & \qquad \mathbf{v}_i \geq 0 \ . \end{aligned} \tag{1.7}$$

To use this in the FDROP minimization problem (Equation 1.3), we introduce an auxiliary variable t_i, which at the optimum will obtain the minimum of (1.7). The resulting problem is a reformulation of the FDROP problem

$$\begin{aligned} \min \quad & \tfrac{1}{2}\|\mathbf{w}\|^2 + C \textstyle\sum_i [1 - y_i \mathbf{w} \cdot \mathbf{x}_i + t_i]_+ \\ \text{s.t.} \quad & t_i \geq K z_i + \textstyle\sum_j v_{ij} \\ & \mathbf{v}_i \geq 0 \\ & z_i + \mathbf{v}_i \geq y_i \mathbf{x}_i \circ \mathbf{w} \ . \end{aligned} \tag{1.8}$$

The above problem can be easily converted into a standard quadratic program, by introducing extra variables $\xi_i \geq 0$ (for $i = 1, \ldots, n$) to represent the hinge function via linear equalities:

$$\begin{aligned} \min \quad & \tfrac{1}{2}\|\mathbf{w}\|^2 + C \textstyle\sum_i \xi_i \\ \text{s.t.} \quad & \xi_i \geq 1 - y_i \mathbf{w} \cdot \mathbf{x}_i + t_i \\ & \xi_i \geq 0 \\ & t_i \geq K z_i + \textstyle\sum_j v_{ij} \\ & \mathbf{v}_i \geq 0 \\ & z_i + \mathbf{v}_i \geq y_i \mathbf{x}_i \circ \mathbf{w} \ . \end{aligned} \tag{1.9}$$

We thus have the result that the FDROP problem in (1.3) is equivalent to the convex quadratic program (QP) in (1.9). The latter has $O(nd)$ variables and constraints, and can be solved using standard QP solvers. However, such solvers may not scale well with nd, and thus may not be usable for datasets with hundreds of thousands of variables and samples. For example, each iteration of an interior point method will require memory that is quadratic in nd and running time that is cubic in nd [Fine and Scheinberg, 2002]. In the next section we describe a method which is more suitable for these cases, and scales linearly with nd for both memory and running

2. Strong duality requires Slater's condition to hold (see Boyd and Vandenberghe [2004]), which is the case for the current problem.

time.

1.2.2 Efficient Optimization Using Bundle Methods

The FDROP optimization problem in Equation 1.3 involves minimization of a non-differentiable (piecewise linear) function of the variable $\mathbf{w}$. Although such minimization problems cannot be solved using standard gradient methods (e.g., L-BFGS), there is a large class of sub-gradient methods which can be applied in this case [e.g., see Shalev-Shwartz et al., 2007, Nedic and Bertsekas, 2001]

In this section, we show how the recently introduced Bundle Method for regularized Risk Minimization, or BMRM [Teo et al., 2007b], may be applied to solving FDROP. BMRM is a generic method for solving convex regularized risk minimization problems, and does not have any tunable parameters, making it simple to implement. Furthermore, the cost of each BMRM iteration in terms of memory and running time scales linearly with the size of the problem. In what follows, we briefly review BMRM, and show how it can be applied to solve the FDROP problem.

Consider the following minimization problem:

$$\min_{\mathbf{w}} \; J(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + CR_{\text{emp}}(\mathbf{w}) \tag{1.10}$$

where $R_{\text{emp}}(\mathbf{w}) = \sum_{i=1}^{n} l(\mathbf{x}_i, y_i, \mathbf{w})$ and $l(\mathbf{x}_i, y_i, \mathbf{w})$ is a convex non-negative loss function. The FDROP problem in Equation 1.3 has this form with[3]

$$l(\mathbf{x}_i, y_i, \mathbf{w}) = h^{wc}(\mathbf{w}, y_i\mathbf{x}_i) \; . \tag{1.11}$$

The BMRM method solves the minimization in Equation 1.10 by forming a piecewise linear lower bound on $R_{\text{emp}}(\mathbf{w})$, which is made tighter at each iteration. The bound relies on the fact that because of the convexity of $R_{\text{emp}}(\mathbf{w})$, the first order Taylor expansion of $R_{\text{emp}}(\mathbf{w})$ at any point $\mathbf{w}_i$ is a (linear) lower bound on $R_{\text{emp}}(\mathbf{w})$:

$$R_{\text{emp}}(\mathbf{w}) \geq f(\mathbf{w}; \mathbf{w}_i) \tag{1.12}$$

where

$$f(\mathbf{w}; \mathbf{w}_i) = R_{\text{emp}}(\mathbf{w}_i) + (\mathbf{w} - \mathbf{w}_i)\partial_{\mathbf{w}} R_{\text{emp}}(\mathbf{w}_i) \; . \tag{1.13}$$

and $\partial_{\mathbf{w}} R_{\text{emp}}(\mathbf{w}_i)$ is the subgradient of the function $R_{\text{emp}}(\mathbf{w})$ at the point $\mathbf{w}_i$. Taking the maximum of a set of such lower bounds for $\mathbf{w}_1, \ldots, \mathbf{w}_t$ also yields a lower bound on $R_{\text{emp}}(\mathbf{w})$:

$$R_{\text{emp}}(\mathbf{w}) \geq \max_{i=1,\ldots,t} f(\mathbf{w}; \mathbf{w}_i) \; , \tag{1.14}$$

3. The function $h^{wc}(\mathbf{w}, y_i\mathbf{x}_i)$ is convex in $\mathbf{w}$ since it is a maximum of functions that are linear in $\mathbf{w}$.

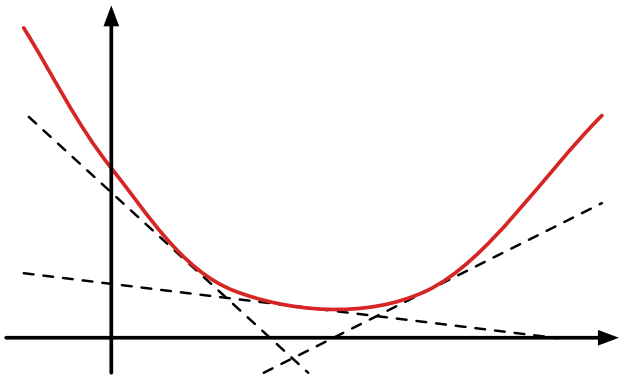

Figure 1.1 A convex function (solid red line) is bounded from below by Taylor approximations of first order (dashed black line). Adding more terms improves the bound.

and this bound becomes tighter as t grows. See Figure 1.1 for an illustration. Since $R_{\text{emp}}(\mathbf{w})$ is non-negative we may further tighten the lower bound by requiring it to be non-negative:

$$R_{\text{emp}}(\mathbf{w}) \geq \max\left[0, \max_{i=1,\ldots,t} f(\mathbf{w}; \mathbf{w}_i)\right] , \tag{1.15}$$

The sequence of points $\mathbf{w}_1, \ldots, \mathbf{w}_t$ is chosen as follows: at iteration t we construct a function $J_t(\mathbf{w})$ that is a lower bound on $J(\mathbf{w})$

$$J_t(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C \max\left[0, \max_{i=1,\ldots,t} f(\mathbf{w}; \mathbf{w}_i)\right] . \tag{1.16}$$

The next point $\mathbf{w}_{t+1}$ is chosen to be the minimizer of $J_t(\mathbf{w})$, i.e.,

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}} J_t(\mathbf{w}) . \tag{1.17}$$

The minimization problem above can be expressed as a QP with t constraints by introducing an auxiliary variable ξ as follows

$$\begin{array}{lll} \min & \frac{1}{2}\|\mathbf{w}\|^2 + C\xi & \\ \text{s.t.} & \xi \geq f(\mathbf{w}; \mathbf{w}_i) & i = 1, \ldots, t \\ & \xi \geq 0 . & \end{array} \tag{1.18}$$

The above QP can be solved efficiently, as long as t is not too large. Teo et al. [2007b] prove that the BMRM method converges, and show that $O(\frac{1}{\epsilon})$ iterations are required to achieve a duality gap of ϵ. In practice, we have found that convergence is achieved after a few hundred iterations at most.

To apply BMRM to the FDROP problem, we need the subgradient of $R_{\text{emp}}(\mathbf{w}) = \sum_i h^{wc}(\mathbf{w}, y_i\mathbf{x}_i)$. Denote the $\boldsymbol{\alpha}_i$ that achieves the worst case loss for example i by

Inputs: A training set $\{\mathbf{x}_i, y_i\}_{i=1}^n$. Number of features to delete K. Desired precision $\epsilon > 0$.

Initialization: Set $\mathbf{w}_1 = \mathbf{0}$ and $t = 1$.

Algorithm:

- Repeat:
  - Define

$$f(\mathbf{w}; \mathbf{w}_t) = \sum_{i=1}^n h^{wc}(\mathbf{w}_t, y_i \mathbf{x}_i) + (\mathbf{w} - \mathbf{w}_t) \sum_{i=1}^n \partial_{\mathbf{w}} h^{wc}(\mathbf{w}_t, y_i \mathbf{x}_i)$$

  where the subgradient is given in Equation 1.20.
  - Solve the quadratic program:

$$\begin{array}{ll} \min & \frac{1}{2}\|\mathbf{w}\|^2 + C\xi \\ \text{s.t.} & \xi \geq f(\mathbf{w}; \mathbf{w}_i) \quad i = 1, \ldots, t \\ & \xi \geq 0 \end{array}$$

  Denote the minimizer by $\mathbf{w}_{t+1}$, and the objective value by l_t.
  - Calculate upper bound $u_t = \frac{1}{2}\|\mathbf{w}_t\|^2 + C \sum_{i=1}^n h^{wc}(\mathbf{w}_t, y_i \mathbf{x}_i)$.
  - Halt if $u_t - l_t < \epsilon$.
  - Set $t = t + 1$.

Output: Final weight vector $\mathbf{w}_t$.

Figure 1.2 The BMRM algorithm applied to the FDROP problem.

$\boldsymbol{\alpha}_i^{\max}(\mathbf{w}, y_i \mathbf{x}_i)$ so that

$$\begin{array}{rll} \boldsymbol{\alpha}_i^{\max}(\mathbf{w}, y_i \mathbf{x}_i) = & \arg\max & [1 - y_i \mathbf{w} \cdot (\mathbf{x}_i \circ (1 - \boldsymbol{\alpha}_i))]_+ \\ & s.t. & \boldsymbol{\alpha}_i \in \{0, 1\}^d \\ & & \sum_j \alpha_{ij} = K \ . \end{array} \tag{1.19}$$

In Section 1.2.1 we showed that this $\boldsymbol{\alpha}_i^{\max}$ is obtained by finding the K features with maximal $y_i w_j x_{ij}$. The subgradient is then[4]

$$\partial_{\mathbf{w}} h^{wc}(\mathbf{w}, y_i \mathbf{x}_i) = \begin{cases} \mathbf{0} & \text{if } h^{wc}(\mathbf{w}, y_i \mathbf{x}_i) = 0 \\ -y_i \mathbf{x}_i \circ (1 - \boldsymbol{\alpha}_i^{\max}(\mathbf{w}, y_i \mathbf{x}_i)) & \text{if } h^{wc}(\mathbf{w}, y_i \mathbf{x}_i) > 0 \ . \end{cases} \tag{1.20}$$

The subgradient of R_{emp} is then given by

$$\partial_{\mathbf{w}} R_{\text{emp}}(\mathbf{w}) = \sum_i \partial_{\mathbf{w}} h^{wc}(\mathbf{w}, y_i \mathbf{x}_i)) \tag{1.21}$$

4. Note that the subgradient is very similar to a perceptron update where the original point $\mathbf{x}_i$ has been replaced by its "feature deleted" version $\mathbf{x}_i \circ (1 - \boldsymbol{\alpha}_i^{\max}(\mathbf{w}, y_i \mathbf{x}_i))$.

Finally, it is also possible to define a simple stopping criterion for BMRM. Note that the minimum value in (1.17) is a lower bound on the minimum of the FDROP problem. An upper bound may also be obtained by evaluating the FDROP objective at $\mathbf{w}_t$. Thus, the difference between these two bounds yields a measure of the accuracy of the current solution, and can be used as a stopping criterion. Pseudo-code for the BMRM procedure is given in Figure 1.2.

1.3 A Convex Dual for the Minimax Problem

The standard support vector machine problem is a convex quadratic problem, and has a dual convex which reveals some interesting properties and allows the use of kernel classifiers. Since our robust problem is also quadratic and convex, it is interesting to consider its dual problem. A standard duality transformation (e.g., see Boyd and Vandenberghe [2004]) can be used to show that the dual of our robust classifier construction problem is

$$\begin{array}{ll} \min & \frac{1}{2}\|\sum_i y_i\alpha_i\mathbf{x}_i \circ (1-\boldsymbol{\lambda}_i)\|^2 - \sum_i \alpha_i \\ \text{s.t.} & 0 \leq \boldsymbol{\alpha} \leq C \\ & 0 \leq \boldsymbol{\lambda}_i \leq 1 \\ & \sum_j \lambda_{ij} = K \end{array} \tag{1.22}$$

where the variables are: $\boldsymbol{\alpha} \in \mathbb{R}^n$ where n is the number of samples, and $\boldsymbol{\lambda}_i \in \mathbb{R}^d$ for $i = 1, \ldots, n$ where d is the dimension of the input. Furthermore, the optimal set of weights $\mathbf{w}$ can be expressed as:

$$\mathbf{w} = \sum_i y_i\alpha_i\mathbf{x}_i \circ (1-\boldsymbol{\lambda}_i) \ . \tag{1.23}$$

The above problem can be written in an alternative form, where it is more clearly convex

$$\begin{array}{ll} \min & \frac{1}{2}\|\sum_i y_i\mathbf{x}_i \circ (\alpha_i-\boldsymbol{\lambda}_i)\|^2 - \sum_i \alpha_i \\ \text{s.t.} & 0 \leq \boldsymbol{\alpha} \leq C \\ & 0 \leq \boldsymbol{\lambda}_i \leq \alpha_i \\ & \sum_j \lambda_{ij} = K\alpha_i \ . \end{array} \tag{1.24}$$

Here the expression in the norm is an affine function of the variables, and thus the problem is convex.

Recall that the SVM dual is

$$\begin{array}{ll} \min & \frac{1}{2}\|\sum_i \alpha_i y_i\mathbf{x}_i\|^2 - \sum_i \alpha_i \\ \text{s.t.} & 0 \leq \boldsymbol{\alpha} \leq C \end{array} \tag{1.25}$$

where $\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$.

Thus, in our case the weight vector is not a combination of input vectors, but

rather a combination of vectors weighted by elements of weight *up to* α_i where the maximal number of elements that may be set to zero is K. Interestingly, the $\boldsymbol{\lambda}_i$ values can be fractional, so that none of the features has to be completely deleted.

Note that, as opposed to the standard SVM, our dual objective will not involve dot products between $\mathbf{x}_i$, but rather between vectors $\mathbf{x}_i \circ (1 - \boldsymbol{\lambda}_i)$. Thus it is not immediately clear if and how kernel methods may be put to use in this case. This is not surprising, since the algorithm is strongly linked to the structure of the sample space $\mathbb{R}^d$, where features are dropped. Dropping such features alters the kernel function. For a given kernel function, one may consider the relevant minimax problem and try to solve for the $\mathbf{w}$ and $\boldsymbol{\alpha}$ variables, in a similar fashion to Weston et al. [2000]. However, for non-linear kernels this would typically result in a non-convex optimization problem, and would depend on the specific kernel used. It thus remains an interesting challenge to obtain globally optimal algorithms for this case.

1.4 An Alternate Setting: Uniform Feature Deletion

In Section 1.1, we assumed that different features may be deleted for different data points. We can also consider an alternative formulation where once a feature is chosen to be deleted it is deleted uniformly from all data points simultaneously. Clearly, this scenario is subsumed by the one described in the previous section, and is thus less pessimistic.

The worst case hinge loss is defined as in the non-uniform case in Equation 1.2. However, now there is a single $\boldsymbol{\alpha}$ vector for all examples, whereas in the previous scenario, each sample had its own vector. The optimization thus becomes

$$\begin{aligned} \mathbf{w}^* = \min_{\mathbf{w}} \max_{\boldsymbol{\alpha}} \quad & \|\mathbf{w}\|^2 + C \textstyle\sum_i \left[1 - y_i \mathbf{w} \cdot (\mathbf{x}_i \circ (1 - \boldsymbol{\alpha}))\right]_+ \\ \text{s.t.} \quad & \boldsymbol{\alpha} \in \{0,1\}^d \\ & \textstyle\sum_j \alpha_j = K \ . \end{aligned}$$

We first note that the above optimization problem is still convex in $\mathbf{w}$. To see why, denote by $f(\mathbf{w})$ the maximum value over all legal $\boldsymbol{\alpha}$ assignments for a given value of $\mathbf{w}$. Then $f(\mathbf{w})$ is a pointwise maximum over a set of convex functions and is thus convex [Boyd and Vandenberghe, 2004] . The problem of minimizing over $\mathbf{w}$ is therefore convex.

However, although it is convex, the current optimization problem appears more difficult than the one in the previous section, due to the presence of the $\boldsymbol{\alpha}$ in all the sum elements. As before, the integral constraints on $\boldsymbol{\alpha}$ can be relaxed, since the maximum of the inner optimization is attained at the vertices (because the target is convex). However, since the target is non-linear (a hinge function) this maximization is not itself a convex problem, and does not seem to be efficiently solvable.

The problem *can* be solved efficiently as long as $\binom{d}{K}$ is sufficiently small so that all the feasible values of $\boldsymbol{\alpha}$ can be enumerated over. However, our experiments show

that in many cases K needs to be at least 10, so that the uniform method is often not applicable.

1.5 Related Frameworks

The FDROP problem was motivated from a minimax perspective where the goal is to minimize the loss incurred by an adversary. In this section we discuss alternative interpretations of our framework, in the context of feature selection and learning with invariances.

1.5.1 Feature Selection

The adversary in the FDROP minimax problem identifies those input features whose contribution to the margin is maximal. In this sense, the adversary can be thought of as being related to feature selection algorithms which try to find the set of features which, when taken alone, would yield optimal generalization (e.g., see Yang and Pedersen [1997]). A clear illustration of this effect can be seen in Figure 1.4 (section 1.6.2).

However, the current minimax setup differs from the standard feature selection approach in two important aspects. The first is that here we focus on feature *elimination*, i.e., finding the set of features whose elimination would maximally decrease performance. Intuitively, these features should also convey high information when taken on their own, but this is not guaranteed to be the case.

The other aspect which distinguishes the current approach from feature selection is that here features are selected (or eliminated to be precise), for every sample individually. The uniform feature deletion approach described in Section 1.4 is more in line with the standard feature selection framework.

We can provide a somewhat more formal treatment of feature selection optimization algorithms which highlights their relation to the current approach. The standard feature selection goal is to find a set of K features which minimize generalization error. A reasonable approximation is the empirical error, or the hinge loss in our case. Thus the feature selection problem can be posed as (we omit the regularization term here)

$$\begin{array}{ll} \min & \sum_i \left[1 - y_i \mathbf{w} \cdot (\mathbf{x}_i \circ \boldsymbol{\alpha})\right]_+ \\ s.t. & \boldsymbol{\alpha} \in \{0,1\}^d \\ & \sum_j \alpha_j = K \end{array} \tag{1.26}$$

such that minimization is over both $\boldsymbol{\alpha}$ and $\mathbf{w}$. Denote by $f(\mathbf{w})$ the minimum over $\boldsymbol{\alpha}$ assignments for a given value of $\mathbf{w}$. Then $f(\mathbf{w})$ is a pointwise minimum of convex functions and is thus generally non-convex. Thus the optimization problem in (1.26) is not convex, and is generally hard to solve. Furthermore for a large number of features, calculating $f(\mathbf{w})$ requires enumeration over possible $\boldsymbol{\alpha}$ assignments. The

problem may be approximated via different relaxations as in Gilad-Bachrach et al. [2004] or Weston et al. [2000].

The above problem may be slightly altered to resemble our current formulation by allowing the best K features to be chosen on a *per sample* basis (a single set of features might then be selected, for example, by taking the features chosen most often across samples). The resulting optimization problem is

$$\begin{array}{ll} \min & \sum_i \left[1 - y_i \mathbf{w} \cdot (\mathbf{x}_i \circ \boldsymbol{\alpha}_i))\right]_+ \\ s.t. & \boldsymbol{\alpha}_i \in \{0,1\}^d \\ & \sum_j \alpha_{ij} = K \ . \end{array} \tag{1.27}$$

This problem is easier than that in Equation 1.26 in that the minimization over $\boldsymbol{\alpha}_i$ is always tractable: the minimizing $\boldsymbol{\alpha}_i$ is the one which has the minimum contribution to the margin. However, the function $f(\mathbf{w})$ is again non-convex, and thus it seems that the problem remains hard.

It is interesting that these two feature selection variants, while similar in spirit to our minimax problems, seem to have considerably higher complexity, in terms of optimization efficiency. This suggests the FDROP approach may also prove useful for feature selection by finding the set of features it tends to *delete*.

1.5.2 Learning With Invariances

In some learning scenarios, it is reasonable to assume that an input point may be perturbed in certain ways without changing its class. For example, digits may undergo translations or rotations by small angles. Several recent works have addressed learning in this setting [Teo et al., 2007a, Graepel and Herbrich, 2004, Decoste and Schölkopf, 2002]. They share the common approach of assuming that the set of possible perturbations of a data point $\mathbf{x}$ generate a *cloud* of virtual data points, and that the margin of the point $\mathbf{x}$ should be measured with respect to this cloud.

Our adversarial view of feature deletion may also be interpreted in the above framework. The cloud of points in this case would be the point $\mathbf{x}$ and all points that correspond to K feature deletions on $\mathbf{x}$. Our worst case margin in Equation 1.2 may then be interpreted as the worst case margin of any point within this cloud of virtual points. Note, however, that the FDROP problem can be solved without explicitly generating the virtual points, using the methods in Section 1.2. In Teo et al. [2007a] we provide a general formalism of such invariance learning, and show how algorithms such as BMRM [Teo et al., 2007b] may be applied to solving it. Under this formalism, any invariance may be used, as long as an efficient algorithm exists for finding the point with worst case margin. One extension of FDROP which we present in Teo et al. [2007a] is to the case where features are not necessarily deleted, but scaled by some minimum and maximum factor. This new invariance is shown to improve generalization performance on a spam filtering task, when

compared to both FDROP and standard SVM.[5]

1.6 Experiments

In this section we apply FDROP to synthetic and real data. We shall especially be interested in evaluating performance when features are deleted from the test set. Thus, for example we test handwritten digit recognition when pixels are removed from the image. We first focus on relatively small training sets, such that the inherent sparseness of the problem is high, and most classification algorithms are likely to overfit. In Section 1.6.3 we report results on a large scale spam filtering experiment, with hundreds of thousands of features. In all experiments, we compare our method with a linear support vector machine algorithm.[6]

For the small scale experiments (Sections 1.6.1-1.6.2) we used the QP approach in Section 1.2.1 (the ILOG-CPLEX package was used to solve the QPs). For the large scale experiment the BMRM method was used (see Section 1.2.2).

1.6.1 A Synthetic Example

To illustrate the advantages of the current method, we apply it to a setting where the test data indeed differs from the training data by deleting features. We consider a feature vector in $\mathbf{x} \in \mathbb{R}^{20}$ where training examples are drawn uniformly in that space. The label is assigned according to a logistic regression rule

$$p(y = 1|\mathbf{x}) \propto e^{\mathbf{w}\cdot\mathbf{x}+b} \ . \tag{1.28}$$

In our experiments, $w_1 = 5$ and all the other $w_i = -2$. The bias b was set to the mean of $\mathbf{w}$. Thus the feature x_1 is likely to be assigned a high weight by learning algorithms which do not expect feature deletion. In the test data, we delete the feature x_1, i.e. set it to zero, with a given probability $p(delete)$. We compare the performance of our FDROP minimax algorithm (with $K = 1$) to that of a standard SVM. For both methods, we choose the weight of the regularization parameter C via cross validation.

Figure 1.3 shows the resulting error rates. It can be seen that as the probability of deletion increases, the performance of SVM decreases, while that of the minimax algorithm stays roughly constant. This constant behavior is due to the fact that the FDROP classifier is optimized for the worst case when this feature is deleted. To

5. We present results for the same spam dataset in Section 1.6.3, but since different preprocessing is used, the results differ from those in Teo et al. [2007a].
6. In Sections 1.6.1 and 1.6.2, both FDROP and SVM use a bias term, by adding a constant feature $x_{d+1} = 1$. The FDROP algorithm was not *allowed* to delete the bias feature $x_{d+1} = 1$. In Section 1.6.3 we did not use a bias term, since this degraded the results for both algorithms.

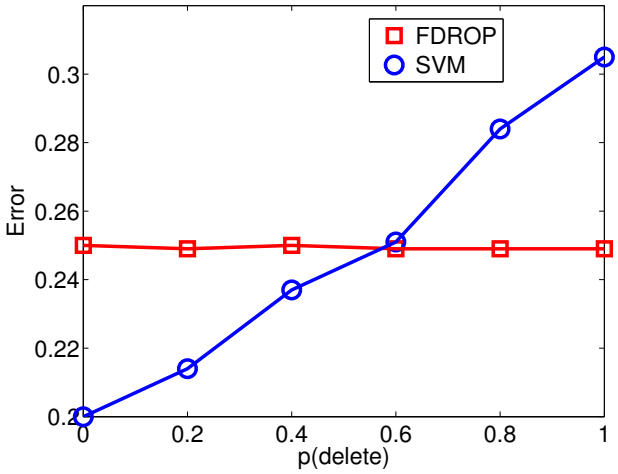

Figure 1.3 Evaluation of FDROP and SVM on a toy logistic regression example, where a highly informative feature is randomly dropped from the test sample. The value of K was set to 1. The figure shows classification error as a function of the deletion probability $p(delete)$.

understand this behavior further, we checked which feature was deleted by FDROP for every one of the samples. Indeed, on *all* the cases where $x_1 = 1$ and $y = 1$, it was x_1 that was deleted in the optimization.

1.6.2 Handwritten Digit Classification

Image classification into categories should in principle be robust to pixel deletion, or in other words deletion of parts of the image. Our game theoretic framework captures this intuition by modeling the worst case pixel deletion scenario.

We investigated the application of FDROP to classifying handwritten digits, and focused on robustness to pixel deletion in these images. We applied FDROP to the MNIST dataset [LeCun et al., 1995] of handwritten digits, and focused on binary problems with small training sets of 50 samples per digit. Furthermore, we only considered binary problems created by label pairs which had more than 5% error when learned using an SVM (the chosen pairs were $(4,9)$,$(3,5)$,$(7,9)$,$(5,8)$,$(3,8)$,$(2,8)$,$(2,3)$,$(8,9)$,$(5,6)$,$(2,7)$,$(4,7)$ and $(2,6)$). The size in pixels of each digit was (28×28). A holdout sample of size 200 was used to optimize the algorithm parameters, and a set of 300 samples was used for testing. The holdout set underwent the same pixel deletion as the test set, in order to achieve a fair comparison between SVM and FDROP. Experiments were repeated with 20 random subsets of the above sizes.

To evaluate the robustness of the algorithm to feature deletion, we trained it on the raw data (i.e., without deleted features), and then tested it on data from which K features were deleted. The values of K were $(0, 25, 50, 75, 100, 125, 150)$.

Figure 1.4 gives a visual representation of the feature deletion process. The

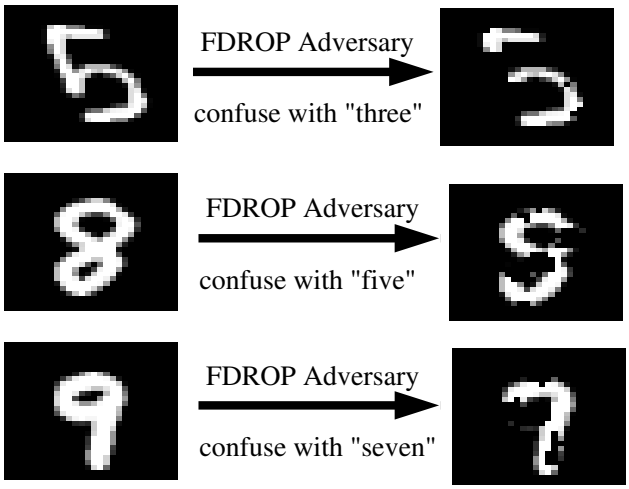

Figure 1.4 Illustration of adversarial feature (pixel) deletion for handwritten digits. Three binary classification problems were created from the MNIST digit database by discriminating the classes "five" vs. "three" (top), "eight" vs. "five" (middle) and "seven" vs. "nine". The training data consisted of 50 samples per class. The number of deleted features was $K = 50$. The images show three corresponding examples of features deleted by the FDROP adversary. The left column shows the original digit, and the right column shows the digit with the 50 pixels dropped by the FDROP algorithm. It can be seen that the worst case against which our algorithm attempts to be robust corresponds to the deletion of extremely discriminative features for each example: the top right digit has been made to look as much as possible like a "three", the middle right digit very much like a "five" and the bottom right digit has been distorted to look very much like a "seven".

FDROP minimax optimization deletes K features from every sample point. We can find which features were deleted from each sample by finding the K features with maximum margin contribution at the optimal $\mathbf{w}$. Figure 1.4 illustrates these features for three sample points. Each row displays the original raw input image and the same input image with the K most *destructive* features deleted (here $K = 50$). It can be seen that FDROP chooses to delete the features which maximize the resemblance between the given digit and digits in the other class. These results suggest that FDROP may indeed be useful as a feature selection mechanism.

Classification error rate should intuitively decrease as more features are deleted. The goal of FDROP is to minimize the damage incurred by such deletion. Figure 1.5 shows the dependence of classification error on the number of deleted features for both FDROP and SVM. The parameter K is taken as an unknown and is chosen to minimize error on the holdout sample for each digit pair and deletion level separately. It can be seen that FDROP suffers less degradation in error when compared to SVM. Furthermore, the optimal K grows monotonically with the number of deleted features, as is intuitively expected. The dependence on K for a specific digit pair (4 and 7) and deletion level (50 deletions) is shown in Figure 1.6. It can be seen that performance is improved up to a value of $K = 25$ which supposedly matches the deletion level in the data set (recall that FDROP considers a

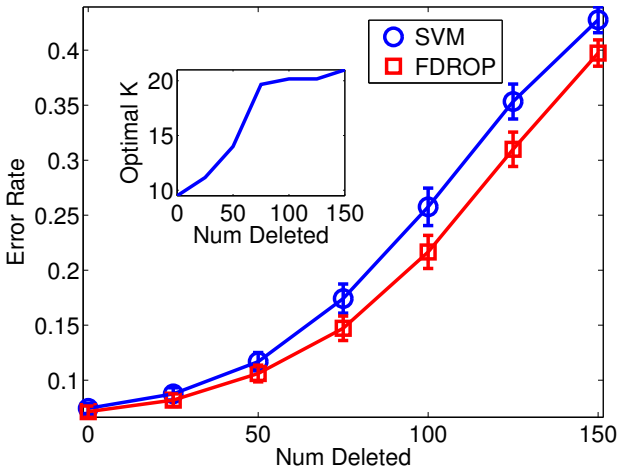

Figure 1.5 Classification error rate for the MNIST dataset, as a function of the number of features deleted from the test set. Standard errors over 20 repetitions are shown on the curve. The optimal K parameter for the FDROP algorithm was chosen per classification problem and per number of deleted features. The inset shows the optimal K for each deletion scheme.

worst case scenario, whereas here features are dropped randomly, so that K and the actual number of deleted features should not be expected to be close numerically).

1.6.3 Spam Filtering

One of the difficulties in filtering spam email from legitimate email is that the problem is dynamic in nature, in the sense that spam authors react to spam filters by changing content. In this sense, it is indeed a game where the two players are the spam filter and spam authors. Our formalism captures this competition, and it is therefore interesting to apply it to this case. Clearly spam authors may change their email in ways other than removing words. For example they may add *good* words, or change the spelling of words [Lowd and Meek, 2005, Wittel and Wu, 2004, Dalvi et al., 2004]. Here we limit the adversarial strategy to word deletion, but our method may be extended to handling other strategies, using its extension in Teo et al. [2007a].

In the experiments described in previous sections, we used relatively small sample-sizes and data dimensionality. In these cases, the FDROP problem could be solved using the QP in Equation 1.8. The current section focuses on a much larger problem, where the QP in Equation 1.8 becomes too big to solve using standard solvers. However, the problem can still be solved using the BMRM method described in Section 1.2.2.

We used the ECML'06 Discovery Challenge (Task A) evaluation dataset [Bickel, 2006]. The training set consists of 4000 emails from a single inbox whereas the testing set consists of 7500 emails from 3 different inboxes. The vocabulary size was

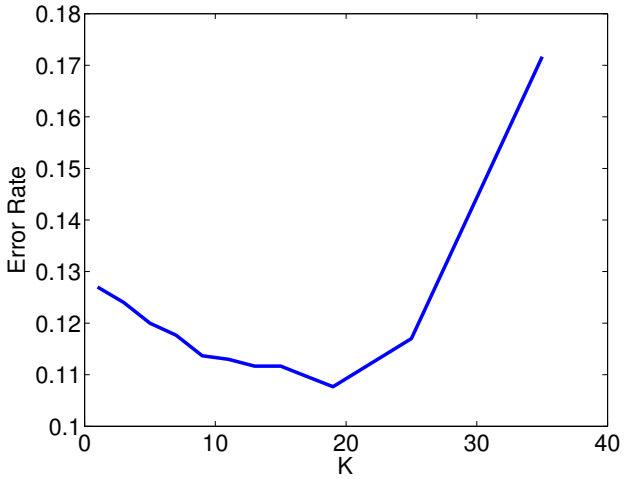

Figure 1.6 Classification error as a function of the parameter K for the digit pair $(4, 7)$ with 50 deleted features.

Method	Accuracy %	AUC %	Parameters (K, C)
SVM	77.20	90.02	(0, 1.25)
FDROP	86.63	94.03	(14, 1.25)

Figure 1.7 Results on the ECML'06 spam detection task for the SVM and FDROP algorithms. The table reports classification accuracy and area under the ROC curve (AUC). The values of C and K were obtained by optimizing over a separate tuning dataset.

$d = 206,908$. We followed the approach of Drucker et al. [1999] by pre-processing the bag-of-word feature vectors into binary vectors and then normalizing them to unit norm. The values of C and K were chosen to optimize performance on a separate tuning data set.

Performance was compared to a linear SVM, and measured in terms of classification accuracy and the area under the ROC curve (AUC). Results are reported in Table 1.7. It can be seen that FDROP significantly outperforms SVM on this task, for both performance measures. We emphasize that the test data was not changed, and no features were artificially deleted, so that FDROP indeed results in robustness and improved generalization performance.

1.7 Discussion & Conclusions

We have introduced a novel method for learning classifiers which are minimax optimal under a worst case scenario of feature deletion at test time. This is an important step towards extending statistical learning paradigms beyond the restrictive assumption that the training and testing data must come from the same distribution. An alternative view of our algorithm is as a feature selection method

which seeks the features which are most crucial for performance. A key assumption of our approach is that small sets of features should not be relied upon at test time to faithfully represent the class structure. Thus, in some sense, the features available to the algorithm at training time are viewed as being subject to random, or even deliberate removal at test time. Interestingly, a recent paper by Krupka and Tishby [2006] presents a related view of features, where one considers a learning scheme where features are selected randomly from a large set, and generalization is studied with respect to unseen features.

Clearly, in some cases the adversarial model may be too strong, and thus result in decreased performance when compared to a standard SVM. For example, the data may not undergo any feature deletion, or we may have a large enough training set so that there is no need to introduce robustness via feature deletion. In these cases it may be preferable to use our model with $K = 0$. One way of addressing this issue is to use cross-validation in choosing the parameter K, so that if $K = 0$ yields lower errors on a validation set, it will be used in the final classifier. This is the approach we used in our experiments, and we indeed found that it results in lower K values in problems where less features are deleted.

A different game theoretic approach to feature selection was previously suggested in Cohen et al. [2005]. Their approach is related to Shapley values in cooperative games. The Shapley value is a measure of the performance drop incurred by dropping a feature from a given set of features, where this performance is averaged over all subsets in which this feature participates. It is thus close in spirit to our feature elimination approach. However, our approach searches for multiple features simultaneously and is furthermore tractable, as opposed to exact calculation of Shapley values.

The notion of robustness to feature deletion is not limited to the classification setting. One may consider a similar setting in the context of regression or dimensionality reduction. It would be interesting to extend the method described here to these settings.

Finally, while here we focus on an adversary that deletes features, the formalism can be easily extended to other perturbations of the feature vector. In Teo et al. [2007a] we outline such a general approach, and provide algorithms for solving the resulting optimization problem.

References

S. Bickel. ECML-PKDD Discovery Challenge 2006 Overview. In *Proceedings of the ECML-PKDD Discovery Challenge Workshop*, 2006.

A. C. Bovik, J. D. Gibson, and A. Bovik, editors. *Handbook of Image and Video Processing*. Academic Press, Inc., Orlando, FL, USA, 2000.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

S. Cohen, E. Ruppin, and G. Dror. Feature selection based on the Shapley value. In L. P. Kaelbling and A. Saffiotti, editors, *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, pages 665–670. Professional Book Center, 2005.

N. Dalvi, P. Domingos, Mausam, S. Sanghai, and D. Verma. Adversarial classification. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 99–108, New York, NY, 2004. ACM Press.

D. Decoste and B. Schölkopf. Training invariant support vector machines. *Machine Learning*, 46 (1-3):161–190, 2002.

H. Drucker, D. Wu, and V. Vapnik. Support vector machines for spam categorization. *IEEE Transactions on Neural Networks*, 10(5):1048–1054, 1999.

L. El Ghaoui, G. R. G. Lanckriet, and G. Natsoulis. Robust classification with interval data. Technical Report UCB/CSD-03-1279, EECS Department, University of California, Berkeley, 2003.

S. Fine and K. Scheinberg. Efficient SVM training using low-rank kernel representations. *Journal of Machine Learning Research*, 2:243–264, 2002.

R. Gilad-Bachrach, A. Navot, and N. Tishby. Margin based feature selection - theory and algorithms. In *Proceedings of the Twenty-First International Conference on Machine Learning*, pages 43–50. ACM Press, New York, NY, 2004.

T. Graepel and R. Herbrich. Invariant pattern recognition by semidefinite programming machines. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 33–40. MIT Press, Cambridge, MA, 2004.

S. Kim, A. Magnani, and S. Boyd. Robust Fisher discriminant analysis. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 659–666. MIT Press, Cambridge, MA, 2006.

E. Krupka and N. Tishby. Generalization in clustering with unobserved features. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 683–690. MIT Press, Cambridge, MA, 2006.

G. R. G. Lanckriet, N. Cristianini, P. L. Bartlett, L. El Ghaoui, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

Y. LeCun, L. Jackel, L. Bottou, A. Brunot, C. Cortes, J. Denker, H. Drucker, I. Guyon, U. Müller, E. Sackinger, P. Simard, and V. N. Vapnik. Comparison of learning algorithms for handwritten digit recognition. In F. Fogelman and P. Gallinari, editors, *International Conference on Artificial Neural Networks,*, pages 53–60. North-Holland, Amsterdam, 1995.

D. Lowd and C. Meek. Good word attacks on statistical spam filters. In *Proceedings of The Second Conference on Email and Anti-Spam (CEAS)*, 2005.

A. Nedic and D. P. Bertsekas. Incremental subgradient methods for nondifferentiable optimization. *SIAM Journal on Optimization*, 12(1):109–138, 2001.

B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for SVM. In *Proceedings of the Twenty-Fourth International Conference on Machine Learning*. ACM, New York, NY, 2007.

C. H. Teo, A. Globerson, S. Roweis, and A. Smola. Convex learning with invariances. In Advances in Neural Information Processing Systems 20, editor, *Advances in Neural Information Processing Systems 20*. MIT Press, Cambridge, MA, 2007a.

C. H. Teo, Q. Le, A. Smola, and S. V. N. Vishwanathan. A scalable modular convex solver for regularized risk minimization. In *Proc. of the Thirteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 727–736. ACM Press, New York, NY, 2007b.

J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. N. Vapnik. Feature selection for SVMs. In *Advances in Neural Information Processing Systems 13*, pages 668–674. MIT Press, Cambridge, MA, 2000.

G. Wittel and S. Wu. On attacking statistical spam filters. In *Proceedings of the First Conference on Email and Anti-Spam (CEAS)*, 2004.

Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 412–420, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

Notation and Symbols

Sets of Numbers

$\mathbb{N}$	the set of natural numbers, $\mathbb{N} = \{1, 2, \dots\}$
$\mathbb{R}$	the set of reals
$[n]$	compact notation for $\{1, \dots, n\}$
$x \in [a, b]$	interval $a \leq x \leq b$
$x \in (a, b]$	interval $a < x \leq b$
$x \in (a, b)$	interval $a < x < b$
$\lvert C \rvert$	cardinality of a set C (for finite sets, the number of elements)

Data

$\mathcal{X}$	the input domain
d	(used if $\mathcal{X}$ is a vector space) dimension of $\mathcal{X}$
M	number of classes (for classification)
n	a number of data examples.
n_{tr}	number of training examples.
n_{te}	number of test examples.
i, j	indices, often running over $[n_{\text{te}}]$ or $[n_{\text{tr}}]$.
x_i	input patterns $x_i \in \mathcal{X}$
x_i^{tr}	input training patterns $x_i^{\text{tr}} \in \mathcal{X}$
x_i^{te}	input test patterns $x_i^{\text{te}} \in \mathcal{X}$
y_i	classes $y_i \in [M]$ (for regression: target values $y_i \in \mathbb{R}$)
y_i^{tr}	training data classes $y_i^{\text{tr}} \in [M]$ (for regression: target values $y_i^{\text{tr}} \in \mathbb{R}$)
y_i^{te}	test data classes $y_i^{\text{tr}} \in [M]$ (for regression: target values $y_i^{\text{te}} \in \mathbb{R}$)
X	a sample of input patterns, $X = (x_1, \dots, x_n)$
X^{tr}	a sample of training input patterns, $X^{\text{tr}} = (x_1^{\text{tr}}, \dots, x_n^{\text{tr}})$
X^{te}	a sample of test input patterns, $X^{\text{te}} = (x_1^{\text{te}}, \dots, x_n^{\text{te}})$
Y	a sample of output targets, $Y = (y_1, \dots, y_n)$
Y^{tr}	a sample of training output targets, $Y^{\text{tr}} = (y_1^{\text{tr}}, \dots, y_n^{\text{tr}})$
Y^{te}	a sample of training output targets, $Y^{\text{te}} = (y_1^{\text{te}}, \dots, y_n^{\text{te}})$

Kernels

$\mathcal{H}$	feature space induced by a kernel
Φ	feature map, $\Phi : \mathcal{X} \to \mathcal{H}$
k	(positive definite) kernel
K	kernel matrix or Gram matrix, $K_{ij} = k(x_i, x_j)$

Vectors, Matrices and Norms

$\mathbf{1}$	vector with all entries equal to one
$\mathbf{I}$	identity matrix
$A^\top$	transposed matrix (or vector)
A^{-1}	inverse matrix (in some cases, pseudo-inverse)
$\mathbf{tr}\,(A)$	trace of a matrix
$\mathbf{det}\,(A)$	determinant of a matrix
$\langle \mathbf{x}, \mathbf{x}' \rangle$	dot product between $\mathbf{x}$ and $\mathbf{x}'$
$\mathbf{x} \circ \mathbf{x}'$	Elementwise multiplication of vectors $\mathbf{x}$ and $\mathbf{x}'$
$\|\cdot\|$	2-norm, $\|\mathbf{x}\| := \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$
$\|\cdot\|_p$	p-norm , $\|\mathbf{x}\|_p := \left(\sum_{i=1}^N \|x_i\|^p \right)^{1/p}$, $N \in \mathbb{N} \cup \{\infty\}$
$\|\cdot\|_\infty$	∞-norm , $\|\mathbf{x}\|_\infty := \sup_{i=1}^N \|x_i\|$, $N \in \mathbb{N} \cup \{\infty\}$

Functions

ln	logarithm to base e
$\log_2$	logarithm to base 2
f	a function, often from $\mathcal{X}$ or $[n]$ to $\mathbb{R}$, $\mathbb{R}^M$ or $[M]$
$\mathcal{F}$	a family of functions
$L_p(\mathcal{X})$	function spaces, $1 \leq p \leq \infty$

Probability

$\mathrm{P}\{\cdot\}$	probability of a logical formula
$\mathrm{P}_{\mathrm{tr}}\{\cdot\}$	probability of a logical formula associated with training data distribution.
$\mathrm{P}_{\mathrm{te}}\{\cdot\}$	probability of a logical formula associated with test data distribution.
$\mathrm{P}(C)$	probability of a set (event) C
$p(x)$	density evaluated at $x \in \mathcal{X}$
$p_{\mathrm{tr}}(x)$	density associated with training data distribution evaluated at $x \in \mathcal{X}$
$p_{\mathrm{te}}(x)$	density associated with test data distribution evaluated at $x \in \mathcal{X}$
$\mathbf{E}\left[\cdot\right]$	expectation of a random variable
$\mathbf{Var}\left[\cdot\right]$	variance of a random variable
$\mathcal{N}(\mu, \sigma^2)$	normal distribution with mean μ and variance σ^2

Graphs

g	graph $\mathsf{g} = (V, E)$ with nodes V and edges E
$\mathcal{G}$	set of graphs
W	weighted adjacency matrix of a graph ($W_{ij} \neq 0 \Leftrightarrow (i,j) \in E$)
D	(diagonal) degree matrix of a graph, $D_{ii} = \sum_j W_{ij}$
$\mathcal{L}$	normalized graph Laplacian, $\mathcal{L} = D^{-1/2} W D^{-1/2}$
L	un-normalized graph Laplacian, $L = D - W$

SVM-related

$\rho_f(x, y)$	margin of function f on the example (x, y), i.e., $y \cdot f(x)$
ρ_f	margin of f on the training set, i.e., $\min_{i=1}^m \rho_f(x_i, y_i)$
h	VC dimension
C	regularization parameter in front of the empirical risk term
λ	regularization parameter in front of the regularizer
$\mathbf{w}$	weight vector
b	constant offset (or threshold)
α_i	Lagrange multiplier or expansion coefficient
β_i	Lagrange multiplier
$\boldsymbol{\alpha}, \boldsymbol{\beta}$	vectors of Lagrange multipliers
ξ_i	slack variables
$\boldsymbol{\xi}$	vector of all slack variables
Q	Hessian of a quadratic program

Miscellaneous

I_A	characteristic (or indicator) function on a set A i.e., $I_A(x) = 1$ if $x \in A$ and 0 otherwise
δ_{ij}	Kronecker δ ($\delta_{ij} = 1$ if $i = j, 0$ otherwise)
δ_x	Dirac δ, satisfying $\int \delta_x(y) f(y) dy = f(x)$
$O(g(n))$	a function $f(n)$ is said to be $O(g(n))$ if there exist constants $C > 0$ and $n_0 \in \mathbb{N}$ such that $\|f(n)\| \leq Cg(n)$ for all $n \geq n_0$
$o(g(n))$	a function $f(n)$ is said to be $o(g(n))$ if there exist constants $c > 0$ and $n_0 \in \mathbb{N}$ such that $\|f(n)\| \geq cg(n)$ for all $n \geq n_0$
rhs/lhs	shorthand for "right/left hand side"
■	the end of a proof