University of Alberta

Robust Learning under Uncertain Test Distributions

by

Junfeng Wen

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

Fall 2013
Edmonton, Alberta

To my parents.

Abstract

Many learning situations involve learning the conditional distribution $p(y|x)$ when the training data is drawn from the training distribution $p_{tr}(x)$, even though it will later be used to predict for instances drawn from a different test distribution $p_{te}(x)$. Most current approaches focus on learning how to reweigh the training examples, to make them resemble the test distribution. However, reweighing does not always help, because (we show that) the test error also depends on the correctness of the underlying model class. This thesis analyses this situation by viewing the problem of learning under changing distributions as a game between a learner and an adversary. We characterize when such reweighing is needed, and also provide an algorithm, robust covariate shift adjustment (RCSA), that provides relevant weights. Our empirical studies, on UCI datasets and a real-world cancer prognostic prediction dataset, show that our analysis applies, and that our RCSA works effectively.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Prof. Russell Greiner, for his patience, guidance and support throughout my M.Sc. studies. I am deeply grateful to Chun-Nam Yu for many helpful discussions and insightful suggestions. Without them this work would not have been done.

I would like to express my appreciation to my committee members, Prof. Dale Schuurmans and Prof. Michael Bowling for their time, effort and precious feedback on this research.

Finally, I would like to thank all of my friends. Their cordial support have helped me overcome many difficulties and their pleasurable companionship have enriched my life.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Consider classifying images of cars and motorcycles as a machine learning task, where images of red cars and black motorcycles are given as the training dataset. A trained classifier may give a prediction rule that every red vehicle is a car while every black vehicle is a motorcycle. This rule is sufficient if we only care about red cars and black motorcycles, i.e., if the test images are "of the same kind" as the training images. What if we provide the classifier an image of a black car? Based on this prediction rule, it will be classified as a motorcycle, which is obviously wrong. However, if we detect in advance that the test images are "different" from what we have in the training set, we can adjust our classifier and learn a more robust prediction rule, such as every four-wheeled vehicle is a car while every two-wheeled vehicle is a motorcycle. Also consider predicting the survival time of a cancer patient, where patient data is collected from different cities, with different underlying distributions. If we train our predicting model on data from one city, is the model readily applicable to patients in another city? What are the consequences if the gender ratios are significantly different in different datasets? Do we need to re-train or adjust the model? All of these questions are crucial and require serious consideration in order to guarantee the effectiveness of a model. Addressing the problem of distribution shift has a wide range of applications, including but not limited to natural language processing [14, 9], bioinformatics [26] and sentiment classification [22].

Traditional machine learning often explicitly or implicitly assumes that

the data used for training a model come from the same distribution as that of the test data. However, this assumption is violated in many real-world applications: the training distribution $p_{tr}(x, y)$ may be different from the test distribution $p_{te}(x, y)$, where $x \in \mathcal{X}$ is the input variable and $y \in \mathcal{Y}$ is the output variable, $\mathcal{X}$ and $\mathcal{Y}$ are their respective domains.

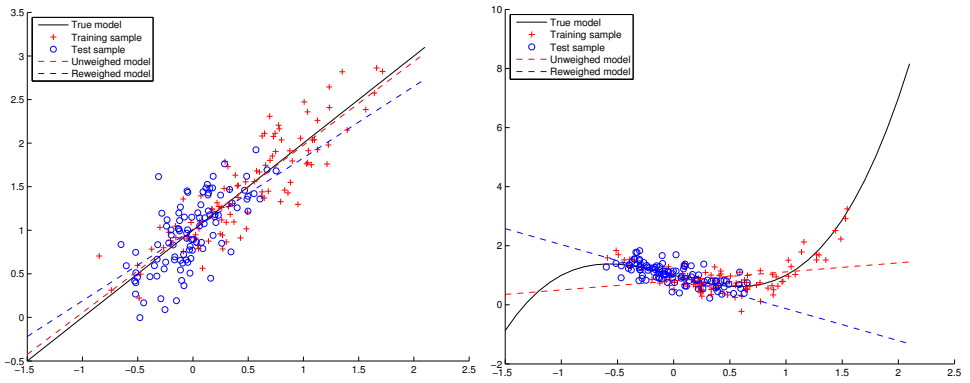
Learning will be effective and meaningful only when the training distribution and test distribution are somehow related. If the training distribution is completely irrelevant to the test distribution, learning itself will be hopeless, because we cannot extract useful information or knowledge from training data to resolve a problem in the test set. Therefore, assumption about how the distributions differ from each other is required. In this thesis, we investigate the problem of distribution change under *covariate shift* assumption [27], in which both training and test distributions share the same conditional distribution $p(y|x)$, while their marginal distributions, $p_{tr}(x)$ and $p_{te}(x)$, are different. To correct the shifted distribution, major efforts have been dedicated to importance reweighing approaches [32, 5, 11, 37, 16]. However, it is not clear how a learner will react to the weights.

In this thesis, we relate covariate shift to *model misspecification* [35]. We notice that importance reweighing helps most where the model misspecification is large. As illustrated in Figure 1.1, consider the regression task with additive Gaussian noise:

$$y = f(x) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, 0.3^2), \tag{1.1}$$

where we have 100 training instances x from $\mathcal{N}(0.5, 0.5^2)$ and 100 test instances from $\mathcal{N}(0, 0.3^2)$ [27]. Assume we are going to fit the data with a linear model $\boldsymbol{\theta} \in \Theta = \mathbb{R}^2$, i.e., our prediction will be $\hat{y} = \theta_1 \cdot x + \theta_0$. As shown in Figure 1.1a, if the true model is $f(x) = x + 1$, which is linear, then the unweighed model performs well on the test set even though the marginal test distribution is shifted. However, if the true model is $f(x) = x^3 - x + 1$ (as in Figure 1.1b), the model class Θ is noticeably misspecified. Then reweighing or model revision is required.

The goal of this thesis is to study when importance reweighing can help



(a) Linear model $f(x) = x + 1$. (b) Cubic model $f(x) = x^3 - x + 1$.

Figure 1.1: Fitting linear lines for data generated from different models.

a learner deal with covariate shift. We model this problem of learning under changing test distributions as a game between a learner and an adversary. The learner chooses a model $\boldsymbol{\theta}$ from a set Θ to *minimize* the loss, while the adversary chooses a reweighing function $\boldsymbol{\alpha}$ from a set $\mathcal{A}$ to create new marginal test distributions to *maximize* the loss. The set of strategies $\mathcal{A}$ is determined by our prior knowledge on how the test distributions might change. We show that the minimax solution of this game can be efficiently computed for many learning problems: e.g., when the loss function is convex in $\boldsymbol{\theta}$ and the set of reweighing functions are linear in $\boldsymbol{\alpha}$. Our key observation is that the question on whether importance reweighing is needed, can depend on whether there is a dominant strategy $\boldsymbol{\theta} \in \Theta$ of the learner against the adversarial set $\mathcal{A}$. By comparing the value of the minimax solution against the unweighed solution, we can test for the existence of such a dominant strategy and hence decide whether importance reweighing could be helpful.

1.1 Thesis Statement

We show that the problem of covariate shift is highly correlated to model specification. Specifically, when covariate shift occurs, if the underlying model class is highly misspecified, density ratio correction algorithms could produce better performance in test set; if the model class is relatively well-specified, density ratio correction would not give better performance in test set.

1.2 Thesis Contributions

This thesis focuses on three interrelated but distinct tasks:

1. Given a model class Θ and only labelled training set, how to learn a model that is robust to certain covariate shifts?

2. Given a model class Θ, both labelled training set and unlabelled test set, how to achieve good performance in this test set?

3. Given a model class Θ and only labelled training set, do we need to reweigh training instances to cope with certain covariate shifts?

There are three major contributions in this work:

- We introduce a robust learning formulation (Task 1) and a density ratio correction method (Task 2), robust covariate shift adjustment (RCSA), that ties density ratio correction to the learning problem.

- We provide a theoretical analysis for understanding why density ratio correction does not help in many covariate shift scenarios, which relates to whether the model class is misspecified. (Task 3)

- We provide a systematic method for checking the model against different covariate shift scenarios, to help the user decide if density ratio correction could be helpful, as opposed to considering a different model class. (Task 3)

1.3 Thesis Organization

Chapter 2 provides the background and related work. Chapter 3 describes our game-theoretic formulation of learning under uncertain test distributions, as well as how to correct distribution shift when unlabelled test points are available (Task 1 and Task 2). Chapter 4 characterizes the test for whether density ratio correction may be necessary and some associated theoretical results (Task 3). Chapter 5 provides experimental evaluation of this test on real datasets

against different classes of adversaries and compares our reweighing method with existing algorithms. Chapter 6 concludes the thesis by summarizing our contributions and outlining future work.

Chapter 2

Related Work

2.1 Covariate Shift

Machine learning techniques often impose specific assumptions on the relation between the training and test distributions (i.e., $p_{tr}(x, y)$ and $p_{te}(x, y)$) in order to guarantee learnability. For instance, most traditional machine learning methods assume that data on which a model is built comes from the same distribution (or source) as those for testing, i.e., $p_{tr}(x, y) = p_{te}(x, y)$. In some other *transfer learning* [19] scenarios, $p_{tr}(x, y)$ and $p_{te}(x, y)$ are more or less different in a non-trivial sense. Specifically, our *covariate shift* assumes the following:

$$\begin{aligned} p_{tr}(x, y) &= p_{tr}(x)\ p(y \mid x) \\ p_{te}(x, y) &= p_{te}(x)\ p(y \mid x) \\ p_{tr}(x) &\neq p_{te}(x). \end{aligned}$$

Note that we implicitly assume that the training input x_{tr} and test input x_{te} come from the same domain $\mathcal{X}$. They only differ in terms of the marginal distributions, $p_{tr}(x)$ and $p_{te}(x)$.

To address the issues caused by covariate shift, *importance reweighing* [23, 29, 33] approaches are relatively popular in the machine learning literature.

2.1.1 Importance Reweighing

Shimodaira [27] showed that given covariate shift and model misspecification,

reweighing each instance with

$$w(x) = \frac{p_{te}(x)}{p_{tr}(x)} \tag{2.1}$$

is asymptotically optimal for log-likelihood estimation. Because of this quantity, importance reweighing is sometimes referred to as density ratio correction in this thesis. As we do not always have enough data, they provided a theoretical analysis on the trade-off between bias and variance of the estimator when dataset is of moderate size. A practical information criterion was proposed to select appropriate weight form when the data is limited. However, $p_{te}(x)$ and $p_{tr}(x)$ were assumed to be known in their paper, which is not true in many real-world applications. Moreover, these distributions are difficult to estimate, especially in high-dimensional space. This work was extended later, where an (almost) unbiased estimator for L_2 generalization error was proposed [30]. In our work, we do not assume $p_{tr}(x)$ and $p_{te}(x)$ to be known in advance when correcting covariate shift.

Instead of estimating $p_{te}(x)$ and $p_{tr}(x)$ separately, it is more suitable to estimate $w(x)$ directly from training and test data. Sugiyama et al. [32] proposed *Kullback-Leibler importance estimation procedure* (KLIEP), which minimizes the Kullback-Leibler divergence (KL divergence) from $p_{te}(x)$ to $\widehat{p}_{te}(x) = \widehat{w}(x)p_{tr}(x)$. The proposed linearly parametric form was

$$\widehat{w}(x) = \sum_l \alpha_l \varphi_l(x), \tag{2.2}$$

where α_l are the parameters to be learned and $\varphi_l(x)$ are non-negative basis functions. In practice, Gaussian kernel was applied as $\varphi_l(x)$, and its parameters were tuned based on KL divergence score with cross validation. Yamada et al. [37] then proposed a similar approach, *relative unconstrained least-squares importance fitting* (RuLSIF), which minimizes Pearson divergence instead of KL divergence. This approach is also related to Kanamori et al. [15, 16]. Their works resemble ours in that we both estimate the reweighing function via a linearly parametrized form (Eq.(2.2)). However, notice that their approaches decouple the estimation of weights and the learning task, which is potentially inferior because it may includes weights that might be

irrelevant to the underlying task. For example, when we have a mislabelled training point that is close to test points, it is very likely to be up-weighed because its proximity to test points, but this will incur a large error rate since it is mislabelled. If we are aware of the underlying classification task, the mislabelled point is less likely to be up-weighed, as it will increase the loss.

Kernel mean matching (KMM) [13, 11] depicts a bijection between probability measure and marginal polytope. As a result, the shifted distribution can be corrected by a reweighing function to match means in a reproducing kernel Hilbert space (RKHS) [25] induced by a kernel. A recent analysis [38] showed that KMM is effective when the estimator in question (for example, generalization error estimator) is related to the underlying RKHS. However, this condition is difficult or impossible to verify and appears almost always violated in practice. Therefore, KMM is not always easy to tune [7, 32]. Our work and some other approaches [20, 21] also adapt the idea of matching means (first moments) of the dataset to correct shifted distribution, but we extend their approaches from a two-step optimization to a game framework that jointly learns a model and weights with covariate shift correction.

2.1.2 Other Approaches and Theories

There are some approaches that explain covariate shift in a probabilistic point of view. Zadrozny [39] introduced the notion of binary selection variable, s, to characterize the selection mechanism of biased sample. An instance associated with $s = 1$ is selected into training sample. s is considered to be independent of y given x, that is, $p(s|x, y) = p(s|x)$. Given $p(s|x)$, reweigh each instance with $w(x) = \frac{p(s=1)}{p(s=1|x)}$, where $p(s = 1) = \sum_x p(s = 1, x)$. Such reweighing function is optimal if $p(s = 1|x)$ is positive for all x, i.e., the support of $p_{te}(x)$ should be a subset of the support of $p_{tr}(x)$. As pointed out by Bickel et al. [4, 5], this approach cannot resolve unseen instances with $p_{te}(x) \neq 0$ but $p_{tr}(x) = 0$. Therefore, a generalized approach was proposed [4, 5]. Figure 2.1 summarises their approach. A data pool was introduced and samples are selected based on a selection variable s. The binary variable s determines whether a instance goes to training set or test set. The selection is controlled by $\boldsymbol{v}$, while the

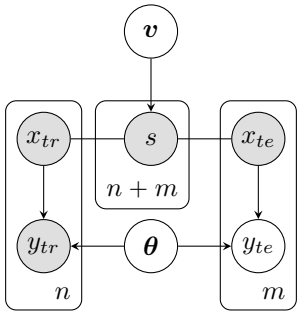


Figure 2.1: Graphical model of Bickel et al. [4, 5].

model is characterized by $\boldsymbol{\theta}$. To be specific,

$$p(y = 1|\boldsymbol{x}; \boldsymbol{\theta}) = \frac{1}{1 + \exp(-\boldsymbol{\theta}^T \boldsymbol{x})}, \tag{2.3}$$

$$p(s = 1|\boldsymbol{x}; \boldsymbol{v}) = \frac{1}{1 + \exp(-\boldsymbol{v}^T \boldsymbol{x})}. \tag{2.4}$$

Similar to our work, they perform a joint optimization to learn both weights (via $\boldsymbol{v}$) and model ($\boldsymbol{\theta}$) simultaneously. Their approach is capable of encoding instances that are either $p_{tr}(x) = 0$ or $p_{te}(x) = 0$ because of the data pool. However, differing from our approach, which is convex with weak conditions, their optimization is convex only with very specific conditions.

Storkey and Sugiyama [28] introduced a joint approach to distinguish data sources and learn a regressor. They assumed that training data comes from two possible sources, one of which is the source that generates test data. It is slightly different from our covariate shift scenario, which assumes both training and test models share the same discriminative model $p(y|x)$. They use Expectation-Maximization (EM) to find local solution for their work, while in our work, we are able to find the global solution because our task is convex.

Besides all these approaches, there are many other works focusing on the theoretical analysis of statistical learning bounds for covariate shift. Based on the $\mathcal{A}$-distance between distributions [17], Ben-David et al. [1] gave a bound on L_1 generalization error given the presence of mismatched distributions. Analyses on other forms of error were also introduced in the literature [27, 30, 8]. There are also some approaches detecting mismatched distributions [10]. However, most of their analyses neglect the effect of the model class. In this thesis, we consider both covariate shift and model misspecification.

2.2 Model Misspecification

Besides covariate shift, model misspecification is another factor that contributes to the performance on test set. Model misspecification is better recognized and studied in the field of econometrics/statistics than in the machine learning community. Our work is influenced by White [35], who proposed a model misspecification test based on the difference between model parameter $\boldsymbol{\theta}$ in two situations. Analysis on squared loss [35] and log likelihood [36] were provided when the model class was misspecified. They introduced a heuristic reweighing function, while comparatively, in our work, the reweighing function is shaped as an adversary that creates worst case scenario for the learner.

Machine learning literature often implicitly or explicitly assumes that the model class at hand is correctly specified for the learning problem [1, 34]. However, this assumption is not always true in real-world applications. It is possibly efficient to tell whether a linear model is well-specified, but there is no easy way to verify whether a non-linear model class is sufficient for the learning task. Many machine learning publications ignore the issue of model misspecification and simply attempt to learn the "best" model $\boldsymbol{\theta}^*$ in a given model class Θ.

Shimodaira [27] pointed out a connection between model misspecification and covariate shift, showing that covariate shift correction could be effective when the model class is misspecified. However, little quantitative evidence was provided for the claim. Gretton et al. [11] also suggested the similar conclusion: when the model class is "simpler" than the true model, reweighing methods are more likely to produce better results than unweighed learning. Theoretical analysis on the relationship between model misspecification and covariate shift is still missing. It is not clear how covariate shift will influence the learning task when the model class is correctly specified or misspecified. Our work partially resolves this issue in that we theoretically analyse the effect of importance reweighing in well-specified scenarios in terms of *dominant strategy* (discussed later), and empirically investigate the effect of model specification in covariate shift cases.

Chapter 3

Learning Under Uncertain Test Distributions as a Game

In this chapter, we introduce the problem formulation and our robust covariate shift adjustment (RCSA) algorithm.

Suppose we are given a training sample $(x_1, y_1), \cdots, (x_n, y_n)$ drawn independently and identically from a joint distribution $p_{tr}(x, y)$, and that the test distribution $p_{te}(x, y)$ is the same as $p_{tr}(x, y)$. The most common and well-established method to learn a prediction function $f : \mathcal{X} \mapsto \mathcal{Y}$ is through solving the following empirical risk minimization (ERM) problem:

$$\min_{\boldsymbol{\theta} \in \Theta} \ \frac{1}{n} \sum_{i=1}^{n} l(f_{\boldsymbol{\theta}}(x_i), y_i) + \lambda \Omega(\boldsymbol{\theta}), \tag{3.1}$$

where the prediction function $f_{\boldsymbol{\theta}}(\cdot)$ is parametrized by a vector $\boldsymbol{\theta}$, $l(\cdot, \cdot)$ is a loss function, $\Omega(\cdot)$ is a regularizer on $\boldsymbol{\theta}$ to control overfitting and $\lambda \in \mathbb{R}$ is regularization parameter.

When there is covariate shift, the feature distribution $p_{te}(x)$ is different from $p_{tr}(x)$ but the conditional distribution $p(y|x)$ representing the classification/regression rule remains the same across training and test sets. In this scenario, one of the most common approach to correct for the effect of covariate shift is to reweigh the training instances in the ERM problem to reflect their true proportions on the test set:

$$\min_{\boldsymbol{\theta} \in \Theta} \ \frac{1}{n} \sum_{i=1}^{n} w(x_i) \ l(f_{\boldsymbol{\theta}}(x_i), y_i) + \lambda \Omega(\boldsymbol{\theta}), \tag{3.2}$$

where $w(x_i)$ is a reweighing function that approximates the density ratio $p_{te}(x_i)/p_{tr}(x_i)$. There are many different methods for estimating the density ratio $w(x)$ using unlabelled test data [23, 29]. Consequently the learning problem becomes a two-step estimation problem, where the density ratio $w(x)$ is estimated first before the estimation of $\boldsymbol{\theta}$ in Eq. (3.2).

This two-step estimation procedure can improve the prediction accuracy on the test set if the density ratio $w(x)$ is accurate. However, the separation of density ratio estimation step and model learning step can lead to missing important interactions between these two steps. For example, $w(x)$ can reweigh instances based on features in x that are irrelevant to the prediction problem for learning $\boldsymbol{\theta}$, thus reducing the effective sample size in the second stage. Also, if there is little or no model misspecification, there is no need to do density ratio correction and reweighing merely increases the variance of the final learned predictor $\boldsymbol{\theta}$ [27]. In general, there is no easy way to tell whether density ratio correction helps or hurts in this two-step procedure, unless we have labelled data from the test distribution.

In this work we tie the two problems of density ratio estimation and learning a predictor together through the robust Bayes framework [12]. The learner tries to minimize the loss by selecting a model $\boldsymbol{\theta} \in \Theta$, while the adversary tries to *maximize* the loss by selecting a reweighing function $w(\cdot) \in \mathcal{W}$. Formally, we model the learning problem as a (regularized) minimax game:

$$\min_{\boldsymbol{\theta}\in\Theta} \max_{w\in\mathcal{W}} \frac{1}{n}\sum_{i=1}^{n} w(x_i)\; l(f_{\boldsymbol{\theta}}(x_i), y_i) + \lambda\Omega(\boldsymbol{\theta}). \tag{3.3}$$

The learner can be seen as minimizing the worst case loss over the set of test distributions $\mathcal{W}$ produced by the adversary. The definition of the strategy set $\mathcal{W}$ used by the adversary is important in our approach, as it determines the extent to which any model misspecification can be exploited by the adversary to increase the loss. Depending on the application scenario, it can be defined using our prior knowledge on how the test distributions could change, or with unlabelled test data if they are available.

To refine this formulation, we assume the reweighing functions $w(x)$ are

linearly parametrized:

$$w_{\boldsymbol{\alpha}}(x) = \sum_{j=1}^{k} \alpha_j k_j(x), \tag{3.4}$$

where $\boldsymbol{\alpha}$ contains the mixing coefficients and $k_j(x)$ are non-negative basis functions. For example, $k_j(x)$ could be non-negative kernel function, say, the Gaussian kernel

$$K(b_j, x) = \exp\left(-\frac{||b_j - x||^2}{2\sigma^2}\right) \tag{3.5}$$

with basis b_j, or it could be $I_j(x)$, the indicator function for the jth disjoint group of the data, representing groups from different genders, age ranges, or k-means clusters, etc. It could be seen as the conditional probability $p(x|j)$ of observing x given class j in a mixture model. As for $\boldsymbol{\alpha}$, it is generally constrained to lie in some compact subspace $\mathcal{A}$ of the non-negative quadrant of Euclidean space. This linear formulation is flexible enough to capture many different types of uncertainties in the test distributions, and yet simple enough to be solved efficiently as a convex optimization problem. Therefore, we consider uncertain test distributions and optimize the following minimax game:

$$\begin{aligned} &\min_{\boldsymbol{\theta}\in\Theta} \max_{\boldsymbol{\alpha}\in\mathbb{R}^k} \frac{1}{n}\sum_{i=1}^{n} w_{\boldsymbol{\alpha}}(x_i)\ l(f_{\boldsymbol{\theta}}(x_i), y_i) + \lambda\Omega(\boldsymbol{\theta}) \\ &\text{s.t. } \frac{1}{n}\sum_{i=1}^{n} w_{\boldsymbol{\alpha}}(x_i) = 1, \qquad 0 \leqslant \alpha_j \leqslant B. \end{aligned} \tag{3.6}$$

The sum-to-one normalization constraint ensures that $w_{\boldsymbol{\alpha}}(x)$ behaves like a Radon-Nikodym derivative [6] that properly reweighs the training distribution to a potential test distribution [27, 32]:

$$1 = \int_{\mathcal{X}} p_{te}(x)dx = \int_{\mathcal{X}} w(x)p_{tr}(x)dx \approx \frac{1}{n}\sum_{i=1}^{n} w_{\boldsymbol{\alpha}}(x_i).$$

The bounds $B \in \mathbb{R}$ on the parameters α_j ensure that the reweighing function $w_{\boldsymbol{\alpha}}(x)$ is bounded, which naturally controls the capacity of the adversary. In this formulation, the strategy set[1] $\mathcal{A}_n$ of the adversary is the intersection of a hypercube and an affine subspace:

$$\mathcal{A}_n = \left\{ \boldsymbol{\alpha} \middle| \frac{1}{n}\sum_{i=1}^{n} w_{\boldsymbol{\alpha}}(x_i) = 1, 0 \leqslant \alpha_j \leqslant B \right\}, \tag{3.7}$$

[1]We use the subscript n to denote its dependence on the sample $\{x_1, \cdots, x_n\}$.

which is closed and convex.

For the games defined above between the learner and the adversary, a minimax solution $(\boldsymbol{\theta}^*, \boldsymbol{\alpha}^*)$ exists. This claim is based on the well-known result on the existence of saddle points for functions $J(\boldsymbol{\theta}, \boldsymbol{\alpha})$ that are convex in $\boldsymbol{\theta}$ and concave in $\boldsymbol{\alpha}$.

Proposition 1 *Define*

$$J(\boldsymbol{\theta}, \boldsymbol{\alpha}) = \frac{1}{n} \sum_{i=1}^{n} w_{\boldsymbol{\alpha}}(x_i)\ l(f_{\boldsymbol{\theta}}(x_i), y_i). \tag{3.8}$$

If $l(f_{\boldsymbol{\theta}}(\cdot), \cdot)$ is convex in $\boldsymbol{\theta}$ and $w_{\boldsymbol{\alpha}}(\cdot)$ is concave in $\boldsymbol{\alpha}$, and Θ and $\mathcal{A}$ are both bounded closed convex sets, then a saddle point $(\boldsymbol{\theta}^*, \boldsymbol{\alpha}^*)$ exists for J, i.e.,

$$J(\boldsymbol{\theta}^*, \boldsymbol{\alpha}) \leqslant J(\boldsymbol{\theta}^*, \boldsymbol{\alpha}^*) \leqslant J(\boldsymbol{\theta}, \boldsymbol{\alpha}^*) \qquad \forall \boldsymbol{\theta} \in \Theta, \forall \boldsymbol{\alpha} \in \mathcal{A}, \tag{3.9}$$

and

$$J(\boldsymbol{\theta}^*, \boldsymbol{\alpha}^*) = \min_{\boldsymbol{\theta} \in \Theta} \max_{\boldsymbol{\alpha} \in \mathcal{A}} J(\boldsymbol{\theta}, \boldsymbol{\alpha}) = \max_{\boldsymbol{\alpha} \in \mathcal{A}} \min_{\boldsymbol{\theta} \in \Theta} J(\boldsymbol{\theta}, \boldsymbol{\alpha}) \tag{3.10}$$

Proof. Direct from Rockafellar [24, Corollary 37.3.2]. ■

3.1 Solving the Training Problem

We first define the *adversarial loss* as

$$L_{\mathcal{A}_n}(\boldsymbol{\theta}) = \max_{\boldsymbol{\alpha} \in \mathcal{A}_n}\ \frac{1}{n} \sum_{i=1}^{n} w_{\boldsymbol{\alpha}}(x_i)\ l(f_{\boldsymbol{\theta}}(x_i), y_i). \tag{3.11}$$

The training problem in Eq. (3.6) can be solved efficiently for loss functions $l(f_{\boldsymbol{\theta}}(\cdot), \cdot)$ that are convex in $\boldsymbol{\theta}$. Notice the adversarial loss in Eq. (3.11) is a *convex* function in $\boldsymbol{\theta}$ if $l(f_{\boldsymbol{\theta}}(\cdot), \cdot)$ is convex in $\boldsymbol{\theta}$, as we are taking the maximum over a set of convex functions. By Danskin's Theorem [3], a subgradient of $L_{\mathcal{A}_n}(\boldsymbol{\theta}')$ at a point $\boldsymbol{\theta}'$ is:

$$\frac{\partial}{\partial \boldsymbol{\theta}} L_{\mathcal{A}_n}(\boldsymbol{\theta}') = \frac{1}{n} \sum_{i=1}^{n} w_{\boldsymbol{\alpha}'}(x_i) \frac{\partial}{\partial \boldsymbol{\theta}} l(f_{\boldsymbol{\theta}'}(x_i), y_i), \tag{3.12}$$

where $\boldsymbol{\alpha}'$ is the solution of the maximization problem with $\boldsymbol{\theta}'$ fixed:

$$\boldsymbol{\alpha}' = \operatorname*{argmax}_{\boldsymbol{\alpha} \in \mathcal{A}_n}\ \frac{1}{n} \sum_{i=1}^{n} w_{\boldsymbol{\alpha}}(x_i)\ l(f_{\boldsymbol{\theta}'}(x_i), y_i). \tag{3.13}$$

Since the strategy set $\mathcal{A}_n$ is linearly constrained and the objective is also linear, $\boldsymbol{\alpha}'$ in Eq. (3.13) can be solved easily using linear programming. Knowing how to compute the subgradient, we can just treat the robust training problem as a convex empirical risk minimization problem with the adversarial loss. The optimization problem can be solved efficiently with subgradient methods [3] or bundle methods [18]; in the experiments below we employ the proximal bundle method for training.

3.2 Incorporating Unlabelled Test Data via Moment Matching Constraints

If unlabelled test data $\{x_{n+1}, \ldots, x_{n+m}\}$ are available, we would expect the reweighing functions $w_{\boldsymbol{\alpha}}(x)$ used by the adversary to produce test distributions that are close to the unlabelled data, especially when covariate shift occurs. In this case we can further restrict the strategy set $\mathcal{A}_n$ of the adversary via moment matching constraints:

$$\begin{aligned} \min_{\boldsymbol{\theta}\in\Theta} \max_{\boldsymbol{\alpha}\in\mathbb{R}^k} \; & \frac{1}{n}\sum_{i=1}^{n} w_{\boldsymbol{\alpha}}(x_i)\; l(f_{\boldsymbol{\theta}}(x_i), y_i) + \lambda\Omega(\boldsymbol{\theta}) \\ \text{s.t. } & \frac{1}{n}\sum_{i=1}^{n} w_{\boldsymbol{\alpha}}(x_i) = 1, \qquad 0 \leqslant \alpha_j \leqslant B \\ & \frac{1}{n}\sum_{i=1}^{n} w_{\boldsymbol{\alpha}}(x_i)\boldsymbol{\phi}(x_i) = \frac{1}{m}\sum_{i=n+1}^{n+m} \boldsymbol{\phi}(x_i), \end{aligned} \tag{3.14}$$

where $\boldsymbol{\phi}(\cdot)$ are feature functions similar to those used in maximum entropy models [2]. Let $K_n\boldsymbol{\alpha} = \bar{\boldsymbol{\phi}}_{te}$ represent the linear constraint of Eq. (3.14), then the strategy set $\mathcal{A}_n$ of the adversary becomes

$$\mathcal{A}_n = \left\{ \boldsymbol{\alpha} \;\middle|\; \frac{1}{n}\sum_{i=1}^{n} w_{\boldsymbol{\alpha}}(x_i) = 1,\; 0 \leqslant \alpha_j \leqslant B,\; K_n\boldsymbol{\alpha} = \bar{\boldsymbol{\phi}}_{te} \right\}, \tag{3.15}$$

which is closed and convex.

In practice, it might not be feasible to satisfy all the moment matching constraints. It is also unwise to enforce these as hard constraints, as the small test sample might not be representative of the true test distribution. We prefer

to solve the soft version of the optimization problem instead:

$$\begin{aligned}
&\min_{\boldsymbol{\theta}\in\Theta} \max_{\boldsymbol{\alpha},\boldsymbol{\xi}} \frac{1}{n}\sum_{i=1}^{n} w_{\boldsymbol{\alpha}}(x_i)\; l(f_{\boldsymbol{\theta}}(x_i), y_i) + \lambda\Omega(\boldsymbol{\theta}) - \mu\|\boldsymbol{\xi}\|_p^p \\
&\text{s.t. } \frac{1}{n}\sum_{i=1}^{n} w_{\boldsymbol{\alpha}}(x_i) = 1, \qquad 0 \leqslant \alpha_j \leqslant B \\
&\qquad \frac{1}{n}\sum_{i=1}^{n} w_{\boldsymbol{\alpha}}(x_i)\boldsymbol{\phi}(x_i) - \frac{1}{m}\sum_{i=n+1}^{n+m} \boldsymbol{\phi}(x_i) = \boldsymbol{\xi}. \qquad (3.16)
\end{aligned}$$

The parameter $\mu \in \mathbb{R}$ controls how hard we want the moment matching constraints to be. Note that the sign of $\mu\|\boldsymbol{\xi}\|_p^p$ is negative because we are penalizing a maximization problem (the adversary). If we use the L_1-norm, $\|\boldsymbol{\xi}\|_1$, on $\boldsymbol{\xi}$, then we are directly penalizing the absolute constraint violation, while using L_2-norm, $\|\boldsymbol{\xi}\|_2$, for $\boldsymbol{\xi}$ allows the matching features $\boldsymbol{\phi}$ to be kernelized, similar to the approach in kernel mean matching [13]. We refer to problem (3.16) as robust covariate shift adjustment (RCSA).

Chapter 4

Relating Covariate Shift to Model Misspecification

This chapter relates covariate shift to model misspecification and describes a procedure for testing whether correcting for covariate shift could be needed, assuming the test distribution comes from the strategy set $\mathcal{A}_n$ of the adversary. We will also state and discuss several theoretical results to justify our test. Their proofs are in the appendix.

Let $\widehat{\boldsymbol{\theta}}_n$ be a solution of the robust Bayes game:

$$\widehat{\boldsymbol{\theta}}_n = \operatorname*{argmin}_{\boldsymbol{\theta}\in\Theta} \max_{\boldsymbol{\alpha}\in\mathcal{A}_n} \frac{1}{n}\sum_{i=1}^{n} w_{\boldsymbol{\alpha}}(x_i)\; l(f_{\boldsymbol{\theta}}(x_i), y_i). \tag{4.1}$$

Let $\bar{\boldsymbol{\theta}}_n$ be a solution of the unweighed empirical risk minimization problem:

$$\bar{\boldsymbol{\theta}}_n = \operatorname*{argmin}_{\boldsymbol{\theta}\in\Theta} \frac{1}{n}\sum_{i=1}^{n} l(f_{\boldsymbol{\theta}}(x_i), y_i). \tag{4.2}$$

The main idea of the test is to compare the adversarial losses $L_{\mathcal{A}_n}(\widehat{\boldsymbol{\theta}}_n)$ and $L_{\mathcal{A}_n}(\bar{\boldsymbol{\theta}}_n)$. If $L_{\mathcal{A}_n}(\bar{\boldsymbol{\theta}}_n)$ is substantially larger than $L_{\mathcal{A}_n}(\widehat{\boldsymbol{\theta}}_n)$, then the adversary can find a strategy $\boldsymbol{\alpha}' \in \mathcal{A}_n$ that exploits the model $\bar{\boldsymbol{\theta}}_n$'s weaknesses much better than the minimax solution $\widehat{\boldsymbol{\theta}}_n$. In this case, density ratio correction *could* help, if the test distribution is characterized by $\boldsymbol{\alpha}'$ (the certificate produced by minimax formula) while the training distribution is not.

The first result is concerned with the convergence of the objective value of Eq. (3.6). Let $\mathcal{A}^S$ be the support of the strategy set $\mathcal{A}_n$ without the stochastic constraints, such as the normalization constraint $\frac{1}{n}\sum_{i=1}^{n} w_{\boldsymbol{\alpha}}(x_i) = 1$ or the

moment matching constraint $K_n\boldsymbol{\alpha} = \bar{\boldsymbol{\phi}}_{te}$. That is, $\mathcal{A}^S$ is the part that does not depend on the training sample, e.g., the hypercube $0 \leqslant \alpha_j \leqslant B$ in the previous chapter. Define

$$\begin{aligned} L_{\mathcal{A}_n}(\boldsymbol{\theta}) &= \max_{\boldsymbol{\alpha}\in\mathcal{A}_n} \frac{1}{n}\sum_{i=1}^{n} w_{\boldsymbol{\alpha}}(x_i)\ l(f_{\boldsymbol{\theta}}(x_i), y_i), \quad \text{where} \\ \mathcal{A}_n &= \left\{ \boldsymbol{\alpha} \in \mathcal{A}^S \middle| \frac{1}{n}\sum_{i=1}^{n} w_{\boldsymbol{\alpha}}(x_i) = 1 \right\}, \quad \text{and} \\ L_{\mathcal{A}_\infty}(\boldsymbol{\theta}) &= \max_{\boldsymbol{\alpha}\in\mathcal{A}_\infty} \int w_{\boldsymbol{\alpha}}(x)\ l(f_{\boldsymbol{\theta}}(x), y) dF(x,y), \quad \text{where} \\ \mathcal{A}_\infty &= \left\{ \boldsymbol{\alpha} \in \mathcal{A}^S \middle| \int w_{\boldsymbol{\alpha}}(x) dF(x,y) = 1 \right\}, \end{aligned} \tag{4.3}$$

where (x_i, y_i) is drawn according to the (Borel) probability measure $F(x, y)$ for $i = 1, \cdots, n$.

Theorem 2 *Suppose the support $\mathcal{A}^S$ for $\boldsymbol{\alpha}$ and Θ for $\boldsymbol{\theta}$ are each closed, convex, and bounded. Suppose also $w_{\boldsymbol{\alpha}}(x)$ and $l(f_{\boldsymbol{\theta}}(x), y)$ are bounded continuous functions in $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$ for each (x, y) pair. If the set satisfying the normalization constraint $\{\boldsymbol{\alpha} \in \mathcal{A}^S | \int w_{\boldsymbol{\alpha}}(x) dF(x,y) = 1\}$ is non-empty in the relative interior of $\mathcal{A}^S$, then we have, for all $\boldsymbol{\theta} \in \Theta$,*

$$L_{\mathcal{A}_n}(\boldsymbol{\theta}) \to L_{\mathcal{A}_\infty}(\boldsymbol{\theta}) \tag{4.4}$$

in probability, i.e., for all $\epsilon, \delta > 0$, we can find $m \in \mathbb{N}$ such that for all $n \geqslant m$, we have

$$|L_{\mathcal{A}_n}(\boldsymbol{\theta}) - L_{\mathcal{A}_\infty}(\boldsymbol{\theta})| < \epsilon \tag{4.5}$$

with probability at least $1 - \delta$.

Thm. 2 shows that the sample adversarial loss converges to a distribution limit for all $\boldsymbol{\theta} \in \Theta$. For simplicity, Thm. 2 does not consider the moment matching constraints $K_n\boldsymbol{\alpha} = \bar{\boldsymbol{\phi}}_{te}$ or $K_n\boldsymbol{\alpha} - \bar{\boldsymbol{\phi}}_{te} = \boldsymbol{\xi}$, but these can be handled in the proof with techniques similar to the normalization constraint.

Our second result is on using this limit as the payoff of the game between the learner and the adversary, to decide whether density ratio correction could be helpful.

Definition 3 (Dominant Strategy) *We say that* $\boldsymbol{\theta}^\dagger \in \Theta$ *is a* dominant strategy *for the learner if, for all* $\boldsymbol{\alpha} \in \mathcal{A}_\infty$*, for all* $\boldsymbol{\theta}' \in \Theta$*,*

$$\int w_{\boldsymbol{\alpha}}(x)\ l(f_{\boldsymbol{\theta}^\dagger}(x), y) dF(x, y) \leqslant \int w_{\boldsymbol{\alpha}}(x)\ l(f_{\boldsymbol{\theta}'}(x), y) dF(x, y). \tag{4.6}$$

The existence of a dominant strategy of the learner is the key criterion in deciding whether density ratio correction is necessary. If such a strategy $\boldsymbol{\theta}^\dagger$ exists, then it gives lower or equal loss compared to other models $\boldsymbol{\theta}'$, no matter which reweighing function $w_{\boldsymbol{\alpha}}(x)$ is used. Thus if one can find a $\boldsymbol{\theta}^\dagger$, no density ratio correction is needed, as long as the training and test distributions come from the given adversarial set. However, if no such strategy exists, then for any model $\boldsymbol{\theta}$, there exist another model $\boldsymbol{\theta}'$ and a reweighing function $w_{\boldsymbol{\alpha}'}(x)$ such that $\boldsymbol{\theta}'$ has strictly lower loss than $\boldsymbol{\theta}$ on $w_{\boldsymbol{\alpha}'}(x)$. This means that a reweighing $w_{\boldsymbol{\alpha}'}(x)$ and its corresponding model $\boldsymbol{\theta}'$ are preferable. As a result, density ratio correction could necessary if the test set is drawn from $w_{\boldsymbol{\alpha}'}(x)$ while the training set is not.

Let $\bar{\boldsymbol{\theta}}$ be the solution of the unweighed loss minimization problem

$$\bar{\boldsymbol{\theta}} = \operatorname*{argmin}_{\boldsymbol{\theta} \in \Theta} \int l(f_{\boldsymbol{\theta}}(x), y) dF(x, y), \tag{4.7}$$

and $\widehat{\boldsymbol{\theta}}$ be the solution of the reweighed adversarial loss minimization problem

$$\widehat{\boldsymbol{\theta}} = \operatorname*{argmin}_{\boldsymbol{\theta} \in \Theta} \max_{\boldsymbol{\alpha} \in \mathcal{A}_\infty} \int w_{\boldsymbol{\alpha}}(x)\ l(f_{\boldsymbol{\theta}}(x), y) dF(x, y). \tag{4.8}$$

Our second result states that, if a dominant strategy $\boldsymbol{\theta}^\dagger$ exists, then under suitable assumption on the adversary, the unweighed solution $\bar{\boldsymbol{\theta}}$ is also a dominant strategy.

Theorem 4 *Suppose the reweighing function* $w_{\boldsymbol{\alpha}}(x)$ *is linear in* $\boldsymbol{\alpha}$*, and the constant reweighing* $\boldsymbol{\alpha}_0$ *with* $w_{\boldsymbol{\alpha}_0}(x) = 1$ *is in the relative interior of* $\mathcal{A}_\infty$*. If a dominant strategy* $\boldsymbol{\theta}^\dagger$ *of the learner exists, then the unweighed solution* $\bar{\boldsymbol{\theta}}$ *is also a dominant strategy for the learner.*

As any dominant strategy $\boldsymbol{\theta}^\dagger$ minimizes the adversarial loss in Eq. (3.11), Thm. 4 implies that the unweighed solution $\bar{\boldsymbol{\theta}}$ will also minimize the adversarial

loss. Therefore by comparing the value of the minimax solution $L_{\mathcal{A}_\infty}(\widehat{\boldsymbol{\theta}})$ (which by definition minimizes the adversarial loss) against $L_{\mathcal{A}_\infty}(\bar{\boldsymbol{\theta}})$, we can tell if a dominant strategy exists. If they are not equal, then we are certain that no such dominant strategy exists, and density ratio correction could be helpful, depending on the distributions from which the training and test sets are drawn. On the other hand, if they are equal, we cannot conclude that a dominant strategy exists, as it is possible that the reweighed adversarial distribution matches the uniform unweighed distribution arbitrarily closely. However, such examples are rather contrived and we never encountered such a situation in any of our experiments. As Thm. 2 shows $L_{\mathcal{A}_n}(\boldsymbol{\theta})$ converges to $L_{\mathcal{A}_\infty}(\boldsymbol{\theta})$ for all $\boldsymbol{\theta} \in \Theta$, our experiments will compare the empirical adversarial loss $L_{\mathcal{A}_n}(\widehat{\boldsymbol{\theta}}_n)$ against $L_{\mathcal{A}_n}(\bar{\boldsymbol{\theta}}_n)$ for the test set, with samples drawn via cross validation.

Now, we can relate our game formulation for learning under uncertain test distributions to model misspecification.

Definition 5 (Pointwise Domination) *A parameter $\boldsymbol{\theta}^\star$ is said to* pointwisely dominate *all $\boldsymbol{\theta}' \in \Theta$ over the loss function $l(\cdot,\cdot)$ if, for all $x \in \mathcal{X}$ and for all $\boldsymbol{\theta}' \in \Theta$,*

$$\int l(f_{\boldsymbol{\theta}^\star}(x), y)\ p(y|x)\ dy \leqslant \int l(f_{\boldsymbol{\theta}'}(x), y)\ p(y|x)\ dy, \tag{4.9}$$

That is to say, there is a single $\boldsymbol{\theta}^\star$ that pointwisely minimizes the loss l for all $x \in \mathcal{X}$.

It is easy to see that this pointwise domination condition is implied by the traditional definition of model misspecification when $l(\cdot,\cdot)$ is the log loss:

$$l(f_{\boldsymbol{\theta}}(x), y) = -\log p_{\boldsymbol{\theta}}(y|x). \tag{4.10}$$

If $p(y|x)$ is the true conditional distribution, then we say that the model class is correctly specified if there exists $\boldsymbol{\theta}^\star \in \Theta$ such that $p_{\boldsymbol{\theta}^\star}(y|x) = p(y|x)$. The pointwise domination condition then becomes:

$$-\int p(y|x)\ \log p_{\boldsymbol{\theta}^\star}(y|x)\ dy \leqslant -\int p(y|x)\ \log p_{\boldsymbol{\theta}'}(y|x)\ dy. \tag{4.11}$$

This inequality always holds because $p_{\boldsymbol{\theta}^\star}(y|x) = p(y|x)$ minimizes the entropy on the left hand side. Therefore, a correctly specified model always implies the

existence of a pointwise dominator $\boldsymbol{\theta}^\star$. However, the converse is not always true, as the underlying model class Θ might be too weak (e.g., Θ contains only a single model $\boldsymbol{\theta}$).

It is easy to show that the pointwise domination condition implies the existence of a dominant strategy $\boldsymbol{\theta}^\dagger$ (Def. 3), against *any* class of adversary $\mathcal{A}$:

Theorem 6 *Suppose a pointwise dominator $\boldsymbol{\theta}^\star$ exists, then $\boldsymbol{\theta}^\star$ is also a dominant strategy for the learner, against any bounded adversarial set $\mathcal{A}$.*

The chain of implications can be summarized as:

No model misspecification for Θ

$\Rightarrow$ Pointwise dominator exists for Θ

$\Rightarrow$ Dominant strategy against any bounded adversary $\mathcal{A}$ exists

$\Rightarrow$ Regular unweighed solution $\bar{\boldsymbol{\theta}}$ is a dominant strategy against some adversary $\mathcal{A}_\infty$

$\Rightarrow$ $\bar{\boldsymbol{\theta}}$ should have no worse performance than robust reweighed solution $\widehat{\boldsymbol{\theta}}$, i.e., $L_{\mathcal{A}_\infty}(\bar{\boldsymbol{\theta}})$ and $L_{\mathcal{A}_\infty}(\widehat{\boldsymbol{\theta}})$ should be equal.

The first implication is a result of the definition of pointwise dominator (Def. 5). Thm. 6 states the second implication. The third implication is Thm. 4, while the forth one is a result of the definition of dominant strategy (Def. 3) and the definition of robust learner $\widehat{\boldsymbol{\theta}}$ (Eq. 4.8). To compare $L_{\mathcal{A}_\infty}(\bar{\boldsymbol{\theta}})$ and $L_{\mathcal{A}_\infty}(\widehat{\boldsymbol{\theta}})$ in practice, we use the convergence theorem (Thm. 2) and compare their empirical estimations $L_{\mathcal{A}_n}(\bar{\boldsymbol{\theta}}_n)$ and $L_{\mathcal{A}_n}(\widehat{\boldsymbol{\theta}}_n)$. If $L_{\mathcal{A}_n}(\bar{\boldsymbol{\theta}}_n)$ is substantially larger than $L_{\mathcal{A}_n}(\widehat{\boldsymbol{\theta}}_n)$, then the adversary can find a strategy $\boldsymbol{\alpha}' \in \mathcal{A}_n$ that exploits the model $\bar{\boldsymbol{\theta}}_n$'s weaknesses much better than the minimax solution $\widehat{\boldsymbol{\theta}}_n$. In this case, density ratio correction *could* help, if the test distribution is characterized by $\boldsymbol{\alpha}'$ (the certificate produced by minimax formula) while the training distribution is not. If $L_{\mathcal{A}_n}(\bar{\boldsymbol{\theta}}_n)$ is very close to $L_{\mathcal{A}_n}(\widehat{\boldsymbol{\theta}}_n)$, then it is unlikely that density ratio correction will improve the learning performance, as long as the training and test distributions come from the given adversarial set.

We can see that "no model misspecification" is a very strong condition, as it requires a dominant strategy against any bounded adversary $\mathcal{A}$, including

pathologically spiky test distributions with tall spikes and small support, or distributions with arbitrary points of discontinuities. Also, there is an *implicit* assumption in using density ratio correction that covariate shifts on the test set are not represented by arbitrarily complex functions. Otherwise estimation of density ratio cannot take place and covariate shift correction is not possible. We believe it is better to test the model class Θ against a restricted set of potential changes in the test distributions represented by our adversarial set $\mathcal{A}_\infty$, than to assume the learner is going to face arbitrary changes in the test distribution as required by model misspecification.

Chapter 5

Empirical Studies

This chapter presents experimental results on toy examples as well as real-world datasets to empirically demonstrate our dominant strategy detection procedure and to show the effectiveness of our robust covariate shift adjustment (RCSA) algorithm.

5.1 Experiment on Toy Datasets

We first present two toy examples to show the performance of our RCSA algorithm. We construct a linear model, $f_1(x) = x + 1 + \epsilon$, and a non-linear (cubic) model, $f_2(x) = x^3 - x + 1 + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 0.1^2)$ is additive Gaussian noise.[1] A linear regressor $f_{\boldsymbol{\theta}}(x) = \theta_1 \cdot x + \theta_0$ is learned from data with squared loss: $l(f_{\boldsymbol{\theta}}(\mathbf{x}_i), y_i) = ||\boldsymbol{\theta}^T \mathbf{x}_i - y_i||^2$. For the regularizer, we use the L_2 norm of $\boldsymbol{\theta}$: $\Omega(\boldsymbol{\theta}) = \frac{1}{2}||\boldsymbol{\theta}||_2^2$.

First we show how to detect whether a dominant strategy exists with various adversarial sets $\mathcal{A}$. We generate 500 data points uniformly in the interval $[-1.5, 2]$ which we partition into training and test sets via 10-fold cross validation. To construct reasonable adversaries, we use Eq.(3.4) with Gaussian kernel as our reweighing function. As we mentioned earlier, the adversarial set is determined by prior knowledge of how the test distribution might change. In this toy example, we use a large range of σ, based on the average distance from an instance to its $\frac{n}{c}$-nearest neighbours, where n is the number of training points and $c \in \{2, 4, 8, 16, \cdots\}$. The smaller σ is, the more powerful the

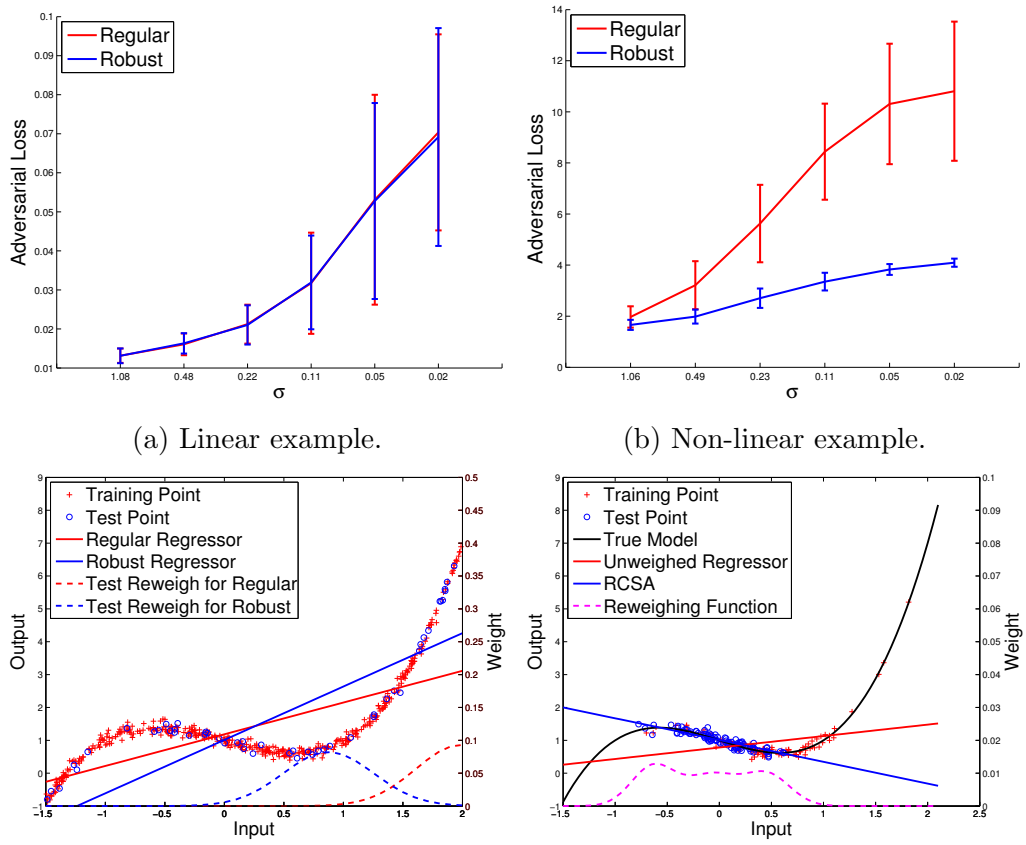
[1] This toy example is adapted from Shimodaira [27].

adversary can be, i.e., the more possible test distributions it can generate. The bases, b_j, are chosen to be the training points. B is set to be 5, a bound that is rarely reached in practice due to the normalization constraint. Therefore, this bound does not significantly limit the adversary's power, as it allows the adversary to put as much importance on a single kernel as it wants. We tune the parameter λ via 10-fold cross validation.[2] Figure 5.1a shows that $L_{\mathcal{A}_n}(\widehat{\boldsymbol{\theta}}_n)$ and $L_{\mathcal{A}_n}(\bar{\boldsymbol{\theta}}_n)$ (mean and one standard deviation as error bar) are very close for all σ in the linear example, indicating that the adversary cannot exploit the weakness of linear learner. Figure 5.1b shows that, for the non-linear example, even with moderate σ, there is a noticeable difference between $L_{\mathcal{A}_n}(\widehat{\boldsymbol{\theta}}_n)$ against $L_{\mathcal{A}_n}(\bar{\boldsymbol{\theta}}_n)$, strongly suggesting that no dominant strategy exists in this case, which suggests that covariate shift correction may be necessary if test distribution is shifted in the non-linear example.

To see how the adversary creates different adversarial losses in a non-linear example, we fix the σ to the average distance from an instance to its $\frac{n}{5}$-nearest neighbour and illustrate a concrete example in Figure 5.1c. It is obvious that the adversary tends to put more weights at the test points where the loss of the classifier learned from training data is large. Our robust formulation takes the adversary into consideration and prevents any point from having too large a loss. As a result, the adversary cannot undermine the robust learner severely, which leads to the gap of the adversarial losses of robust and regular learners in Figure 5.1b.

Now we consider the performance of RCSA in the non-linear example with covariate shift. We generated 100 training points from $\mathcal{N}(0.5, 0.5^2)$ and 100 test points from $\mathcal{N}(0, 0.3^2)$ [27]. Here we set σ as the average distance from an instance to its $\frac{n}{5}$-nearest neighbour. We correct covariate shift using Eq. (3.16), where p is set to 2 for kernelization and μ is set to be the ratio of empirical test loss to empirical moment difference between training and test sets. This particular choice of μ balances our effort on minimizing the loss and enforcing

[2]Here, as there is no covariate shift, we just use simple cross validation. Whenever test distribution is shifted in the experiment, parameters are tuned via importance weighted cross validation [31].



(a) Linear example.

(b) Non-linear example.

(c) Adversarial reweighing.

(d) Reweighing for shift.

Figure 5.1: Toy examples. Adversarial test losses are shown in Figures 5.1a and 5.1b, where the x-axis shows the value of σ. Figure 5.1c provides a non-linear example to show how the adversary attacks the regressors by reweighing the test points, with output on the left y-axis and weight on the right y-axis. Figure 5.1d provides a concrete instantiation of RCSA reweighing for covariate shift in non-linear example.

moments to match. As shown in Figure 5.1d, the RCSA regressor has much better performance compared with regular unweighed regressor on test set. The reweigh function decays quickly as we move away from test set.

5.2 Experiment on Real-world Datasets

This section presents the experimental results of RCSA algorithm on real world datasets to demonstrate how our formulation determines whether there is a dominant strategy against some adversaries and if so, how to correct such covariate shifts. We investigate both regression problems using squared loss,

Table 5.1: Dataset Summary

Dataset	Size	Dim	Type
Australian	690	14	Classification
Breast_cancer	683	10	Classification
German_numer	1000	24	Classification
Heart	270	13	Classification
Ionosphere	351	34	Classification
Liver_disorder	345	6	Classification
Sonar	208	60	Classification
Splice	1000	60	Classification
Auto-mpg	392	6	Regression
Cancer	1523	40	Regression

and classification problems using hinge loss: $l(f_{\boldsymbol{\theta}}(\mathbf{x}_i), y_i) = \max(0, 1 - y_i \boldsymbol{\theta}^T \mathbf{x}_i)$. A linear model is learned from the dataset unless otherwise specified.

5.2.1 Datasets

We obtain some classification datasets from UCI repository[3]. All are binary classification problems. For regression task, we use `Auto-mpg` dataset, which is considered to be natural covariate shift scenario, as it contains data collected from 3 different cities. We also have a set of cancer patient survival time data provided by our medical collaborators, containing 1523 uncensored patients with 40 features, including gender, site and stage of cancer, and various blood work measurements obtained at the time of diagnosis. Table 5.1 shows the summary of the datasets we used in the experiments.

5.2.2 Dominant Strategy Detection

To construct reasonable adversaries, Gaussian kernel is applied to Eq.(3.4). We set σ to be the average distance from an instance to its $\frac{n}{5}$-nearest neighbour. We set the bases b_j to be the training points and set B to be 5. It is possible to set up another adversarial set, as it depends on user's belief about how the test distribution may change. However, this experiment focuses on test distributions that come from this particular adversarial set.

[3] `http://archive.ics.uci.edu/ml/index.html`

Table 5.2: Average Robust and Non-robust Adversarial Test Losses of Linear Model: over 10 Runs

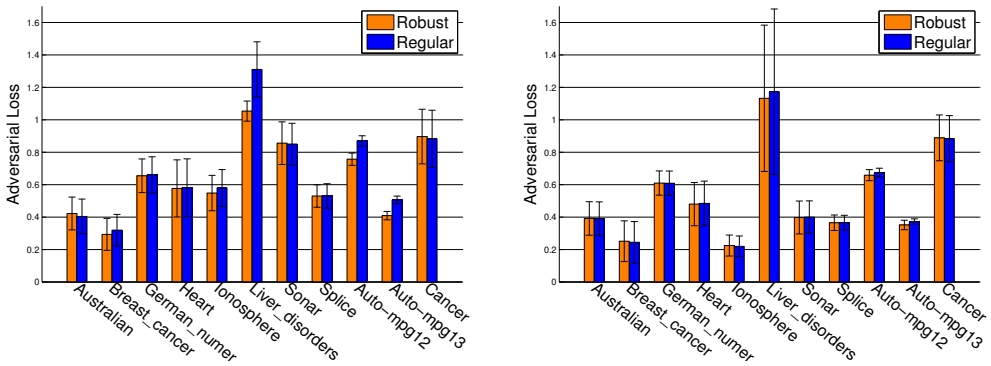
Dataset	Robust	Non-robust	t-test
Australian	0.4222 ± 0.1011	0.4041 ± 0.107	×
Breast_cancer	0.2936 ± 0.09871	0.3197 ± 0.09698	×
German_numer	0.6548 ± 0.1038	0.6597 ± 0.1122	×
Heart	0.5769 ± 0.176	0.5813 ± 0.1781	×
Ionosphere	0.5483 ± 0.1092	0.5795 ± 0.1137	×
Liver_disorders	1.054 ± 0.06216	1.31 ± 0.1706	√
Sonar	0.8559 ± 0.1313	0.8505 ± 0.1278	×
Splice	0.5299 ± 0.06923	0.5304 ± 0.07541	×
Auto-mpg12	0.7572 ± 0.0377	0.8698 ± 0.03188	√
Auto-mpg13	0.4092 ± 0.02557	0.5058 ± 0.02356	√
Cancer	0.8968 ± 0.1684	0.8827 ± 0.176	×

Experimental results are shown in Table 5.2 and Figure 5.2a. `Auto-mpg12` explores when the training data comes from city 1 and test data is from city 2, while `Auto-mpg13` explores when training data comes from city 1 and test data comes from city 3. Here we focus on the adversarial losses of robust versus regular models. A significant difference indicates that there is no dominant strategy and thus, the linear model is vulnerable to our reweighing adversary. For classification datasets and the cancer dataset, we apply 10-fold cross validation to obtain training and test sets. For `Auto-mpg`, we fix the test set and apply 10-fold cross validation to obtain training set. Figure 5.2a presents these losses over the datasets (mean and one standard deviation as error bar). Table 5.2 presents these losses (mean ± one standard deviation) over the datasets, and includes a tick mark to indicate when two losses are significantly different (t-test with significance level 0.05). Our result indicates that the linear model is vulnerable for the `Liver_disorders` and `Auto-mpg` datasets.

To further substantiate the incapability of the linear model, we attempted to detect dominant strategy for a Gaussian model set Θ (i.e., changing from linear kernel to Gaussian kernel with kernel width chosen with cross validation by learner). Results are shown in Table 5.3 and Figure 5.2b. The gap of adversarial losses between robust and regular models shrinks significantly as

Table 5.3: Average Robust and Non-robust Adversarial Test Losses of Gaussian Model: over 10 Runs

Dataset	Robust	Non-robust	t-test
Australian	0.3917 ± 0.1035	0.3899 ± 0.1043	×
Breast_cancer	0.2512 ± 0.1255	0.245 ± 0.1278	×
German_numer	0.6098 ± 0.07473	0.609 ± 0.07513	×
Heart	0.4804 ± 0.1333	0.4846 ± 0.1371	×
Ionosphere	0.2248 ± 0.06474	0.2196 ± 0.06396	×
Liver_disorders	1.132 ± 0.4512	1.173 ± 0.5103	×
Sonar	0.3977 ± 0.1014	0.4 ± 0.1001	×
Splice	0.3653 ± 0.04783	0.3656 ± 0.04541	×
Auto-mpg12	0.6585 ± 0.03384	0.6738 ± 0.02765	×
Auto-mpg13	0.3514 ± 0.02925	0.373 ± 0.01601	×
Cancer	0.8888 ± 0.1413	0.8837 ± 0.142	×



(a) Linear learner (b) Gaussian learner

Figure 5.2: Experimental results for dominant strategy detection and covariate shift correction. Figure 5.2a and Figure 5.2b show the adversarial test losses of robust and regular learners.

in Figure 5.2b. Our result indicates that t-test no longer claims a significant difference between these losses and the adversary cannot severely undermine the performance of regular learning. Therefore, model revision can be a good alternative to performing covariate shift correction.

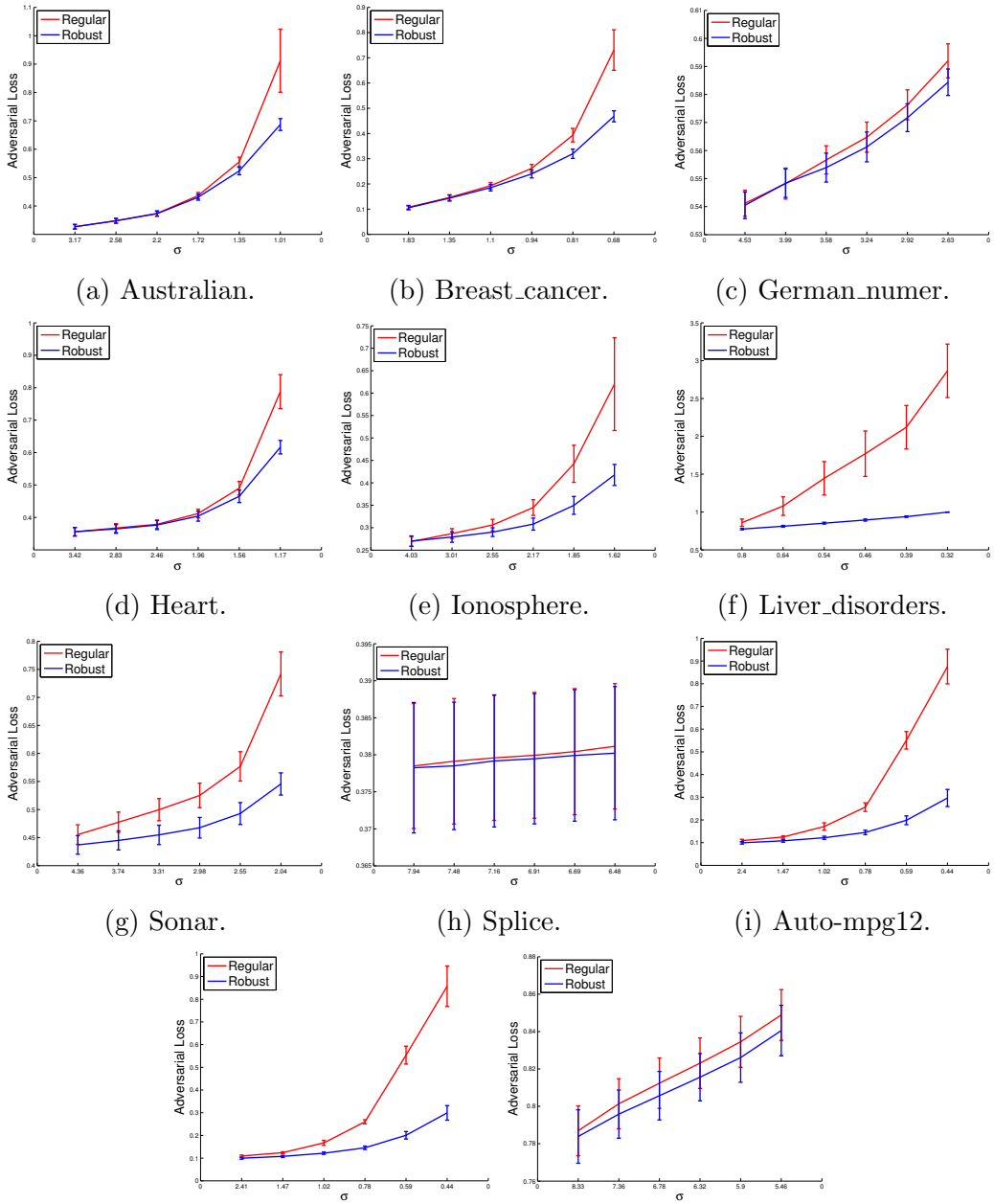
We also investigate the effect of different adversarial sets. Specifically, we vary the kernel width σ in the reweighing basis function Eq. (3.4) as in the toy example of Section 5.1. The adversarial *training* losses are reported in Figure 5.3. We choose the adversarial training loss here because the adversarial

loss on the test set under 10-fold cross validation is sometimes less representative of the true generalization adversarial loss, especially when σ is very small. This is due to the small test set size and the presence of unseen noisy data. Different from the toy example above, we cannot guarantee that the support of training sample includes that of test sample. Unseen noisy data can incur very large loss for both robust and regular models because of its unpredictable nature, which a powerful adversary can exploit. Therefore, we focus on adversarial training losses in Figure 5.3. Note that for `Liver_disorders` and `Auto-mpg` datasets, there are noticeable difference between robust and regular adversarial losses for moderate σ, which resembles our non-linear example in Figure 5.1b. When it comes to other datasets, the difference between adversarial losses is not obvious until σ becomes small enough. This is reasonable because every real-world dataset has some noise and as the adversary becomes more and more powerful, it will concentrate more and more mass on such noisy data. Such a baleful and powerful adversary will eventually exploit the noisy points and undermine the performance of linear model for real-world datasets even if the underlying true model is linear. It is also worth mentioning that if test distribution $p_{te}(x)$ is highly shifted, which can happen when σ are small, density ratio correction will be necessary for not only `Liver_disorders` and `Auto-mpg` but also some other datasets.

5.2.3 Reweighing Algorithm for Covariate Shift Scenario

As previously mentioned, the reweighing mechanism could improve the performance if the model is vulnerable to the reweighing adversary. For the covariate shift correction task, we set the test points as the reference bases b_j of the weight function (Eq. (3.4)), because they are more informative than training points about the test distribution, as suggested by Sugiyama et al. [32]. We use the same choices of p and μ as in the toy example. The reweighing set (σ and B) is chosen as in Section 5.2.2.

To create covariate shift scenarios in the classification datasets, we apply the following mechanism to obtain shifted test set: we first randomly pick 75%



(a) Australian. (b) Breast_cancer. (c) German_numer.

(d) Heart. (e) Ionosphere. (f) Liver_disorders.

(g) Sonar. (h) Splice. (i) Auto-mpg12.

(j) Auto-mpg13. (k) Cancer.

Figure 5.3: Adversarial training losses for different σs.

of the set for robust training Eq. (3.6); from adversarial test loss Eq. (3.11), every test instance has a weight; the probability that a test instance x remains in the test set is $\min\left(1, \frac{w_{\boldsymbol{\alpha}}(x)}{1/m}\right)$, where m is the number of test points at the moment (25% of the set). About 10% of the whole dataset remain as the test set after filtering. Then we run reweighing algorithms on this split.

The intuition of this filtering is that instances that are up-weighed by the adversary are more favourable to be in test set. The procedure is performed 10 times, leading to the average test losses reported in Table 5.4 and Figure 5.4. `Auto-mpg` is a natural covariate shift scenario so we do not need to artificially partition the dataset. We applied 10-fold cross validation to obtain the training set. We consider two covariate shift scenarios in the cancer survival time prediction:

1. `Gender` split. The dataset contains about 60% male and 40% female patients. In gender split, we randomly take 20% of the male and 80% of the female patients into training set, while the rest goes to test set. That is, the training set is dominated by male patients while the test set is dominated by female patients.

2. Cancer `stage` split. Approximately 70% of the dataset are of stage-4. In cancer stage split, we randomly take 20% of stage-1-to-3 and 80% of stage-4 patients to training set, while the rest goes to test set. That is, the training set is dominated by stage-4 patients while the test set is dominated by stage-1-to-3 patients.

Table 5.4 and Figure 5.4 compares the test losses of RCSA with the regular unweighed learning algorithm, the clustering-based reweighing algorithm [7], KLIEP [32] and RuLSIF [37]. Recall that the linear model is insufficient for the `Liver_disorders` and `Auto-mpg` datasets. As a result, by putting more weights on the training instances that are similar to test instances, the reweighing algorithms are able to produce models with smaller test losses. Although our robust game formulation is mainly designed to detect dominant strategy, our RCSA algorithm can correct shifted distribution using the moment matching constraint. As shown in Table 5.4 and Figure 5.4, our method performs on par with state-of-the-art algorithms when covariate shift correction is required. For the datasets that appear linear (i.e., where the linear model performs relatively well), we found that the reweighing algorithms did not significantly reduce the test losses. In some cases, reweighing actually increased the test losses due to the presence of noise.

Table 5.4: Average Test Losses of Different Reweighing Algorithms with Linear Model: over 10 Runs

Dataset	Unweighed	Clust	KLIEP	RuLSIF	RCSA
Australian	0.3185	0.3188	0.3187	0.317	0.3209
Breast_cancer	0.07606	0.07611	0.07971	0.08801	0.08036
German_numer	0.5879	0.5842	0.5841	0.5848	0.5809
Heart	0.4846	0.4824	0.4908	0.4807	0.4832
Ionosphere	0.2965	0.2712	0.2918	0.2948	0.2848
Liver_disorders	**0.7875**	**0.7446**	**0.7702**	**0.7222**	**0.7213**
Sonar	0.5781	0.5604	0.5667	0.56	0.5642
Splice	0.462	0.4637	0.4612	0.4603	0.4563
Auto-mpg12	**0.4503**	**0.3329**	**0.3547**	**0.3249**	**0.3385**
Auto-mpg13	**0.4053**	**0.2057**	**0.2497**	**0.2071**	**0.2063**
Cancer-gender	0.7766	0.7766	0.7762	0.7915	0.7762
Cancer-stage	0.9306	0.9304	0.9271	0.9221	0.9252

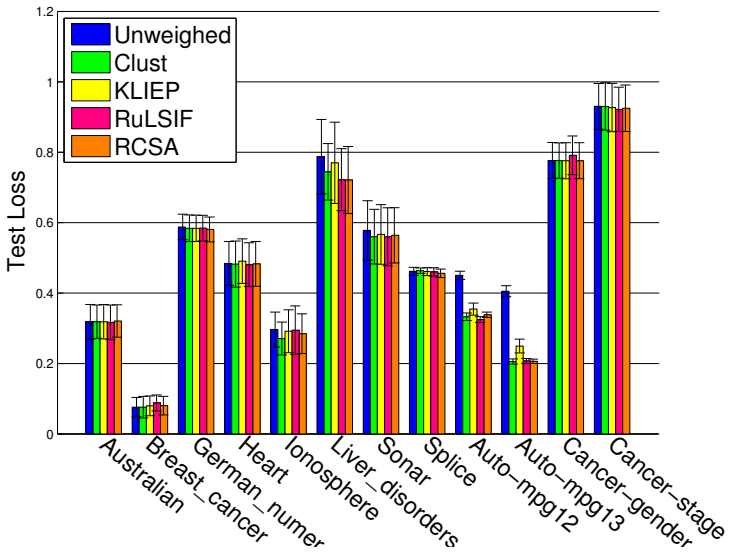


Figure 5.4: Performance of reweighing algorithms with linear model.

Chapter 6

Conclusion

6.1 Future Work

There are several research directions that we can further explore.

- We plan to convert our current asymptotic analysis of the game between learner and adversary into a finite sample analysis. This can lead to improved understanding of the variance in this robust game formulation and improved detection tests.

- We also plan to extend our game interpretation to consider ϵ-dominant strategies instead of dominant strategies. As every real-world dataset has some noise, allowing ϵ tolerance would be helpful to understand the extent to which our detection procedure remains effective.

- One major limitation of our detection procedure is that we cannot certainly claim the existence of dominant strategy when robust and regular learners perform equally well against a pre-defined adversarial set (see the discussion on Theorem 4). In such cases, reweighing algorithms may not improve the performance on test set but revising the model class may help. Detecting such situations would be beneficial for future studies.

- In this thesis, we used moment matching constraints to reweigh training instances such that the reweighed training sample resembles the test sample. However, moment matching constraint is not the only way to enforce sample similarity. It is possible to encode sample similarity based

on some other divergences (for example, Sugiyama et al. [32], Yamada et al. [37]), which may lead to better performance in test set.

6.2 Conclusions and Contributions

We have provided a method for determining if covariate shift correction is needed under a pre-defined set of potential changes in the test distribution. This is useful for ensuring the learned predictor will still perform well when there are uncertainties about the test example distribution in the deployment environment. It can also be used to decide if a model class revision of Θ is necessary.

Experimental results show that our detection test is effective on UCI datasets and a real-world cancer patient dataset. This analysis shows the importance of studying the interaction of covariate shift and model misspecification, because the final test set error depends on both factors.

Bibliography

[1] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems*, volume 19, page 137, 2007.

[2] A. Berger, V. Pietra, and S. Pietra. A maximum entropy approach to natural language processing. *Computational linguistics*, 22(1):39–71, 1996.

[3] D. P. Bertsekas. *Nonlinear programming*. Athena Scientific, 2nd edition, 1999.

[4] S. Bickel, M. Brückner, and T. Scheffer. Discriminative learning for differing training and test distributions. In *International Conference on Machine Learning*, pages 81–88, 2007.

[5] S. Bickel, M. Brückner, and T. Scheffer. Discriminative learning under covariate shift. *The Journal of Machine Learning Research*, 10:2137–2155, 2009.

[6] P. Billingsley. *Probability and measure*. Wiley Series in Probability and Statistics. Wiley, 2012.

[7] C. Cortes, M. Mohri, M. Riley, and A. Rostamizadeh. Sample selection bias correction theory. *Algorithmic Learning Theory*, 5254:38–53, 2008.

[8] C. Cortes, Y. Mansour, and M. Mohri. Learning bounds for importance weighting. *Advances in Neural Information Processing Systems*, 23:442–450, 2010.

[9] H. Daumé III and D. Marcu. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26(1):101–126, 2006.

[10] A. Gretton, K. Borgwardt, M. Rasch, B. Schöelkopf, and A. Smola. A kernel method for the two-sample-problem. In *Advances in Neural Information Processing Systems*, volume 19, pages 513–520, 2007.

[11] A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf. Covariate shift by kernel mean matching. *Dataset Shift in Machine Learning*, pages 131–160, 2009.

[12] P. D. Grünwald and A. P. Dawid. Game theory, maximum entropy, minimum discrepancy and robust bayesian decision theory. *The Annals of Statistics*, 32(4):1367–1433, 2004.

[13] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems*, volume 19, page 601, 2007.

[14] J. Jiang and C. Zhai. Instance weighting for domain adaptation in NLP. In *Annual Meeting-Association For Computational Linguistics*, volume 45, page 264, 2007.

[15] T. Kanamori, S. Hido, and M. Sugiyama. Efficient direct density ratio estimation for non-stationarity adaptation and outlier detection. In *Advances in Neural Information Processing Systems*, 2008.

[16] T. Kanamori, T. Suzuki, and M. Sugiyama. Statistical analysis of kernel-based least-squares density-ratio estimation. *Machine Learning*, 86(3): 335–367, 2012.

[17] D. Kifer, S. Ben-David, and J. Gehrke. Detecting change in data streams. In *International Conference on Very Large Data Bases*, volume 30, pages 180–191, 2004.

[18] K. C. Kiwiel. Proximity control in bundle methods for convex nondifferentiable minimization. *Mathematical Programming*, 46(1):105–122, 1990.

[19] S. J. Pan and Q. Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359, 2010.

[20] S. J. Pan, J. T. Kwok, and Q. Yang. Transfer learning via dimensionality reduction. In *National Conference on Artificial Intelligence*, volume 2, pages 677–682, 2008.

[21] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. In *International Joint Conference on Artifical Intelligence*, pages 1187–1192, 2009.

[22] S. J. Pan, X. Ni, J. Sun, Q. Yang, and Z. Chen. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th international conference on World wide web*, pages 751–760. ACM, 2010.

[23] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009.

[24] R. T. Rockafellar. *Convex analysis*. Princeton University Press, 1996.

[25] B. Schölkopf and A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization and beyond*. The MIT Press, 2002.

[26] G. Schweikert, C. Widmer, B. Schölkopf, and G. Rätsch. An empirical analysis of domain adaptation algorithms for genomic sequence analysis. In *Advances in Neural Information Processing Systems*, pages 1433–1440, 2008.

[27] H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.

[28] A. J. Storkey and M. Sugiyama. Mixture regression for covariate shift. In *Advances in Neural Information Processing Systems*, 2007.

[29] M. Sugiyama and M. Kawanabe. *Machine learning in non-stationary environments: introduction to covariate shift adaptation*. The MIT Press, 2012.

[30] M. Sugiyama and K. Müller. Model selection under covariate shift. In *Artificial Neural Networks: Formal Models and Their Applications*, pages 235–240. Springer, 2005.

[31] M. Sugiyama, M. Krauledat, and K. Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Maching Learning Research*, 8:985–1005, 2007.

[32] M. Sugiyama, S. Nakajima, H. Kashima, P. Von Buenau, and M. Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in Neural Information Processing Systems*, 2008.

[33] M. Sugiyama, T. Suzuki, and T. Kanamori. *Density ratio estimation in machine learning*. Cambridge University Press, 2012.

[34] V. Vandewalle, C. Biernacki, G. Celeux, and G. Govaert. A predictive deviance criterion for selecting a generative model in semi-supervised classification. *Computational Statistics and Data Analysis*, 2013.

[35] H. White. Consequences and detection of misspecified nonlinear regression models. *Journal of the American Statistical Association*, 76(374):419–433, 1981.

[36] H. White. Maximum likelihood estimation of misspecified models. *Econometrica: Journal of the Econometric Society*, pages 1–25, 1982.

[37] M. Yamada, T. Suzuki, T. Kanamori, H. Hachiya, and M. Sugiyama. Relative density-ratio estimation for robust distribution comparison. In *Advances in Neural Information Processing Systems*, pages 594–602, 2011.

[38] Y. Yu and C. Szepesvári. Analysis of kernel mean matching under covariate shift. In *International Conference on Machine Learning*, 2012.

[39] B. Zadrozny. Learning and evaluating classifiers under sample selection bias. In *International Conference on Machine Learning*, page 114. ACM, 2004.

Appendix

Proof of Theorem 2

Notice the reweighed loss is linear in $\boldsymbol{\alpha}$ for fixed $\boldsymbol{\theta}$:

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n} w_{\boldsymbol{\alpha}}(x_i) l(f_{\boldsymbol{\theta}}(x_i), y_i) &= \sum_{j=1}^{k} \alpha_j \frac{1}{n}\sum_{i=1}^{n} k_j(x_i) l(f_{\boldsymbol{\theta}}(x_i), y_i) \\ &= \boldsymbol{h}_n^T \boldsymbol{\alpha},\end{aligned}$$

where

$$(\boldsymbol{h}_n)_j = \frac{1}{n}\sum_{i=1}^{n} k_j(x_i) l(f_{\boldsymbol{\theta}}(x_i), y_i).$$

Therefore we can write $L_{\mathcal{A}_n}(\boldsymbol{\theta})$ as:

$$L_{\mathcal{A}_n}(\boldsymbol{\theta}) = \max_{\boldsymbol{\alpha}\in\mathcal{A}_n} \boldsymbol{h}_n^T \boldsymbol{\alpha}$$

Similarly, define the corresponding cost vector $\boldsymbol{h}$ for the expected adversarial loss such that

$$(\boldsymbol{h})_j = \int k_j(x)\ l_{\boldsymbol{\theta}}(f(x), y) dF(x, y), \tag{6.1}$$

and we have

$$L_{\mathcal{A}_\infty} = \max_{\boldsymbol{\alpha}\in\mathcal{A}_\infty} \boldsymbol{h}^T \boldsymbol{\alpha}$$

Similarly, define for the normalization constraint:

$$\frac{1}{n}\sum_{i=1}^{n} w_{\boldsymbol{\alpha}}(x_i) = \sum_{j=1}^{k} \alpha_j \frac{1}{n}\sum_{i=1}^{n} k_j(x_i) = \boldsymbol{g}_n^T \boldsymbol{\alpha},$$

where

$$(\boldsymbol{g}_n)_j = \frac{1}{n}\sum_{i=1}^{n} k_j(x_i).$$

Define the corresponding constraint vector $\boldsymbol{g}$ such that

$$(\boldsymbol{g})_j = \int k_j(x)\ dF(x, y).$$

Translating into the new notations, we want to prove

$$\left| \max_{\boldsymbol{\alpha} \in \mathcal{A}^S : \boldsymbol{g}^T \boldsymbol{\alpha} = 1} \boldsymbol{h}^T \boldsymbol{\alpha} - \max_{\boldsymbol{\alpha} \in \mathcal{A}^S : \boldsymbol{g}_n^T \boldsymbol{\alpha} = 1} \boldsymbol{h}_n^T \boldsymbol{\alpha} \right| < \epsilon$$

with probability at least $1 - \delta$, for all sufficiently large n.

To prove the result we need two lemmas, whose proofs appear after the main proof. The first lemma states that the sample cost vector $\boldsymbol{h}_n$ converges in the infinite limit to $\boldsymbol{h}$. The second lemma states that near the feasible solutions of $\boldsymbol{g}^T \boldsymbol{\alpha} = 1$, there are feasible solutions of finite sample constraint $\boldsymbol{g}_n^T \boldsymbol{\alpha} = 1$ for large n, and also vice versa.

Lemma 7 *Assume the basis $k_j(x)$ for the reweighing function $w(x)$ are bounded above by B_k, and the loss function l bounded above by B_l. We then have*

$$Pr(\|\boldsymbol{h}_n - \boldsymbol{h}\|_2 \geqslant \epsilon) \leqslant 2k \exp\left(-\frac{2n\epsilon^2}{B_k^2 B_l^2 k^2} \right).$$

Lemma 8 *Suppose $\epsilon, \delta > 0$ are given and let $\boldsymbol{\alpha}^* \in \mathcal{A}_\infty$. Then there exists $m \in \mathbb{N}$ such that, for all $n \geqslant m$, with probability at least $1 - \delta$, we can find $\boldsymbol{\alpha}_n \in \mathcal{A}_n$ such that*

$$\|\boldsymbol{\alpha}^* - \boldsymbol{\alpha}_n\| \leqslant \epsilon$$

Similarly, suppose $\epsilon, \delta > 0$ are given. Then there exists $m \in \mathbb{N}$ such that for all $n \geqslant m$, for any $\boldsymbol{\alpha}_n \in \mathcal{A}_n$, with probability at least $1 - \delta$, we can find $\boldsymbol{\alpha}^* \in \mathcal{A}_\infty$ such that

$$\|\boldsymbol{\alpha}_n - \boldsymbol{\alpha}^*\| \leqslant \epsilon.$$

Proof of Main Theorem

By Lemma 7, there exists $n_1 \in \mathbb{N}$ such that for all $n \geqslant n_1$, $\|\boldsymbol{h} - \boldsymbol{h}_n\| \leqslant \epsilon/(2B_\alpha)$ with probability $1 - \delta/3$. [condition 1]

Let $\boldsymbol{h}^T \boldsymbol{\alpha}^* = \max_{\boldsymbol{\alpha} \in \mathcal{A}_\infty} \boldsymbol{h}^T \boldsymbol{\alpha}$. By Lemma 8, there exists $n_2 \in \mathbb{N}$ such that for all $n \geqslant n_2$, we can find $\boldsymbol{\alpha}'_n \in A_n$ with $\|\boldsymbol{\alpha}^* - \boldsymbol{\alpha}'_n\| \leqslant \epsilon/(2\|\boldsymbol{h}\|)$ with probability $1 - \delta/3$ [condition 2]. Condition 1 and 2 give

$$\begin{aligned}
\max_{\boldsymbol{\alpha}\in\mathcal{A}_\infty} \boldsymbol{h}^T\boldsymbol{\alpha} - \max_{\boldsymbol{\alpha}\in\mathcal{A}_n} \boldsymbol{h}_n^T\boldsymbol{\alpha} &\leqslant \boldsymbol{h}^T\boldsymbol{\alpha}^* - \boldsymbol{h}_n^T\boldsymbol{\alpha}'_n \\
&= \boldsymbol{h}^T\boldsymbol{\alpha}^* - \boldsymbol{h}^T\boldsymbol{\alpha}'_n + \boldsymbol{h}^T\boldsymbol{\alpha}'_n - \boldsymbol{h}_n^T\boldsymbol{\alpha}'_n \\
&= \boldsymbol{h}^T(\boldsymbol{\alpha}^* - \boldsymbol{\alpha}'_n) + (\boldsymbol{h} - \boldsymbol{h}_n)^T\boldsymbol{\alpha}'_n \\
&\leqslant \|\boldsymbol{h}\|\|\boldsymbol{\alpha}^* - \boldsymbol{\alpha}'_n\| + \|\boldsymbol{h} - \boldsymbol{h}_n\|\|\boldsymbol{\alpha}'_n\| \\
&\leqslant \|\boldsymbol{h}\|\frac{\epsilon}{2\|\boldsymbol{h}\|} + \frac{\epsilon}{2B_\alpha}B_\alpha \\
&= \epsilon
\end{aligned}$$

Similarly, let $\boldsymbol{h}_n^T\boldsymbol{\alpha}_n^* = \max_{\boldsymbol{\alpha}\in\mathcal{A}_n} \boldsymbol{h}_n^T\boldsymbol{\alpha}$. By Lemma 8, there exists $n_3 \in \mathbb{N}$ such that for each $n \geqslant n_3$, we can find $\boldsymbol{\alpha}' \in A_\infty$ with $\|\boldsymbol{\alpha}_n^* - \boldsymbol{\alpha}'\| \leqslant \epsilon/2\|\boldsymbol{h}\|$ with probability $1 - \delta/3$ [condition 3]. Condition 1 and 3 give

$$\begin{aligned}
\max_{\boldsymbol{\alpha}\in\mathcal{A}_n} \boldsymbol{h}_n^T\boldsymbol{\alpha} - \max_{\boldsymbol{\alpha}\in\mathcal{A}_\infty} \boldsymbol{h}^T\boldsymbol{\alpha} &\leqslant \boldsymbol{h}_n^T\boldsymbol{\alpha}_n^* - \boldsymbol{h}^T\boldsymbol{\alpha}' \\
&\leqslant \boldsymbol{h}_n^T\boldsymbol{\alpha}_n^* - \boldsymbol{h}^T\boldsymbol{\alpha}_n^* + \boldsymbol{h}^T\boldsymbol{\alpha}_n^* - \boldsymbol{h}^T\boldsymbol{\alpha}' \\
&= (\boldsymbol{h}_n - \boldsymbol{h})^T\boldsymbol{\alpha}_n^* + \boldsymbol{h}^T(\boldsymbol{\alpha}_n^* - \boldsymbol{\alpha}') \\
&\leqslant \|\boldsymbol{h}_n - \boldsymbol{h}\|\|\boldsymbol{\alpha}_n^*\| + \|\boldsymbol{h}\|\|\boldsymbol{\alpha}_n^* - \boldsymbol{\alpha}'\| \\
&\leqslant \frac{\epsilon}{2B_\alpha}B_\alpha + \|\boldsymbol{h}\|\frac{\epsilon}{2\|\boldsymbol{h}\|} \\
&= \epsilon
\end{aligned}$$

Therefore when $n \geqslant \max\{n_1, n_2, n_3\}$, with probability at least $1 - \delta$ (by union bound), we have

$$|\max_{\boldsymbol{\alpha}\in\mathcal{A}_n} \boldsymbol{h}_n^T\boldsymbol{\alpha} - \max_{\boldsymbol{\alpha}\in\mathcal{A}_\infty} \boldsymbol{h}^T\boldsymbol{\alpha}| \leqslant \epsilon \qquad \square$$

Proof of Lemma 7

By Hoeffding's inequality, we have

$$Pr(|(\boldsymbol{h}_n)_j - (\boldsymbol{h})_j| > \frac{\epsilon}{k}) \leqslant 2\exp\left(-\frac{2n\epsilon^2}{B_k^2 B_l^2 k^2}\right).$$

By union bound, we have

$$Pr(\|\boldsymbol{h}_n - \boldsymbol{h}\|_1 \geqslant \epsilon) \leqslant 2k\exp\left(-\frac{2n\epsilon^2}{B_k^2 B_l^2 k^2}\right).$$

As $\|\boldsymbol{h}_n - \boldsymbol{h}\|_2 \leqslant \|\boldsymbol{h}_n - \boldsymbol{h}\|_1$, we have

$$Pr(\|\boldsymbol{h}_n - \boldsymbol{h}\|_2 \geqslant \epsilon) \leqslant 2k\exp\left(-\frac{2n\epsilon^2}{B_k^2 B_l^2 k^2}\right). \qquad \square$$

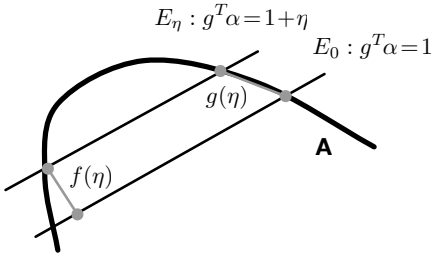


Figure 6.1: Definition of $f(\eta)$ and $g(\eta)$

Proof of Lemma 8

Using Hoeffding's inequality and union bound (similar to the proof of Lemma 7), we have

$$Pr(\|\boldsymbol{g}_n - \boldsymbol{g}\|_2 \geqslant \epsilon) \leqslant 2k \exp\left(-\frac{2n\epsilon^2}{B_k^2 k^2}\right). \tag{6.2}$$

Define

$$E_\eta = \{\boldsymbol{\alpha} \in \mathcal{A}^S | \boldsymbol{g}^T\boldsymbol{\alpha} = 1 + \eta\}$$

for $\eta \in \mathbb{R}$. This is the set of subspace parallel to $\boldsymbol{g}^T\boldsymbol{\alpha} = 1$ (E_0). Define also $f(\eta)$ the maximum distance of any points in E_η to E_0, and $g(\eta)$ the maximum distance of any points in E_0 to E_η (see Fig. 6.1), i.e.,

$$f(\eta) = \max_{\boldsymbol{\alpha}\in E_\eta} \min_{\boldsymbol{\alpha}'\in E_0} \|\boldsymbol{\alpha} - \boldsymbol{\alpha}'\|,$$
$$g(\eta) = \max_{\boldsymbol{\alpha}\in E_0} \min_{\boldsymbol{\alpha}'\in E_\eta} \|\boldsymbol{\alpha} - \boldsymbol{\alpha}'\|.$$

Suppose $\epsilon, \delta > 0$ are given. Using Lemma 9 below, $f(\eta) \to 0$ as $\eta \to 0$, so we can find $\eta_0 > 0$ such that $f(\eta) < \epsilon$ whenever $|\eta| < \eta_0$. From Eq. (6.2), we can find $m \in \mathbb{N}$ such that for all $n \geqslant m$, $\|\boldsymbol{g}_n - \boldsymbol{g}\| < \eta_0/B_\alpha$ with probability at least $1 - \delta$.

Let $\boldsymbol{\alpha}_n \in \mathcal{A}_n$ for $n \geqslant m$, we have

$$|\boldsymbol{g}^T\boldsymbol{\alpha}_n - 1| = |\boldsymbol{g}^T\boldsymbol{\alpha}_n - \boldsymbol{g}_n^T\boldsymbol{\alpha}_n| \leqslant \|\boldsymbol{g} - \boldsymbol{g}_n\|\|\boldsymbol{\alpha}_n\| \leqslant \frac{\eta_0}{B_\alpha} B_\alpha = \eta_0 \tag{6.3}$$

with probability at least $1 - \delta$. Hence the subspace $\boldsymbol{g}_n^T\boldsymbol{\alpha} = 1$, i.e. $\mathcal{A}_n$, lies between E_{η_0} and $E_{-\eta_0}$ with probability $1 - \delta$. Specifically for a fixed $\boldsymbol{\alpha}_n \in \mathcal{A}_n$, it lies on E_η for some η with $|\eta| < \eta_0$.

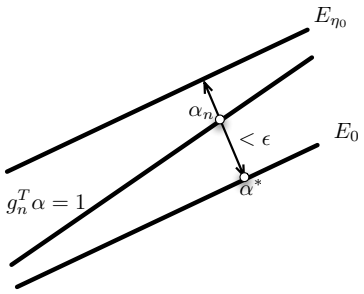


Figure 6.2: Illustration for proof of Lemma 8

Therefore

$$\begin{aligned}\min_{\boldsymbol{\alpha}' \in E_0} \|\boldsymbol{\alpha}_n - \boldsymbol{\alpha}'\| &\leqslant \max_{\boldsymbol{\alpha} \in E_\eta} \min_{\boldsymbol{\alpha}' \in E_0} \|\boldsymbol{\alpha} - \boldsymbol{\alpha}'\| \\ &= f(\eta) \leqslant \epsilon\end{aligned}$$

with probability $1 - \delta$.

For the second part, let $\boldsymbol{\alpha}^* \in \mathcal{A}_\infty (= E_0)$, and $\epsilon, \delta > 0$ be given. Using Lemma 9 below, $g(\eta) \to 0$ as $\eta \to 0$, so we can find $\eta_0 > 0$ such that $g(\eta) < \epsilon$ whenever $|\eta| < \eta_0$. By Eq. (6.3) above we can find $m \in \mathbb{N}$ such that $\mathcal{A}_n$ lies entirely between $E_{-\eta_0}$ and E_{η_0} with probability at least $1 - \delta$. By definition

$$\min_{\boldsymbol{\alpha}' \in E_{\eta_0}} \|\boldsymbol{\alpha}^* - \boldsymbol{\alpha}'\| \leqslant g(\eta_0) \leqslant \epsilon.$$

Let $\boldsymbol{\alpha}_{\eta_0}$ be a point on E_{η_0} minimizing the distance to $\boldsymbol{\alpha}^*$, then the line joining $\boldsymbol{\alpha}_{\eta_0}$ and $\boldsymbol{\alpha}^*$ has to intersect with the subspace $\boldsymbol{g}_n^T \boldsymbol{\alpha} = 1$ at some $\boldsymbol{\alpha}_n$ (see Fig. 6.2). This holds for all $n \geqslant m$ and we have $\|\boldsymbol{\alpha}_n - \boldsymbol{\alpha}^*\| \leqslant \epsilon$. The same argument applies to the case when $\boldsymbol{g}_n^T \boldsymbol{\alpha} = 1$ lies between E_0 and $E_{-\eta_0}$. Thus

$$\min_{\boldsymbol{\alpha}' \in \mathcal{A}_n} \|\boldsymbol{\alpha}' - \boldsymbol{\alpha}^*\| \leqslant \epsilon$$

for all $n \geqslant m$, with probability at least $1 - \delta$.

Lemma 9

$$\begin{aligned}f(\eta) &= \max_{\boldsymbol{\alpha} \in E_\eta} \min_{\boldsymbol{\alpha}' \in E_0} \|\boldsymbol{\alpha} - \boldsymbol{\alpha}'\| \\ g(\eta) &= \max_{\boldsymbol{\alpha} \in E_0} \min_{\boldsymbol{\alpha}' \in E_\eta} \|\boldsymbol{\alpha} - \boldsymbol{\alpha}'\|\end{aligned}$$

converge to 0 as $\eta \to 0$.

Proof. We want to show $f(\eta) \to 0$ as $\eta \to 0$. If not, then there exists $f_0 > 0$ and a sequence $\{\eta_t\}_{t=1}^{\infty}$ with $\eta_t \to 0$, such that $f(\eta_t) \geqslant f_0$ infinitely often. We collect all those indices t_n such that $f(\eta_{t_n}) \geqslant f_0$, and form a new sequence $\mu_n = \eta_{t_n}$. Let

$$\boldsymbol{\alpha}_n = \operatorname*{argmax}_{\boldsymbol{\alpha} \in E_{\mu_n}} \min_{\boldsymbol{\alpha}' \in E_0} \|\boldsymbol{\alpha} - \boldsymbol{\alpha}'\|.$$

As $\boldsymbol{\alpha}_n$ lies in a compact set $\mathcal{A}^S$, there exist a convergent subsequence, say $\boldsymbol{\beta}_n$. Let the subsequence $\boldsymbol{\beta}_n$ converge to some $\boldsymbol{\beta}$, and by continuity we know $\boldsymbol{g}^T\boldsymbol{\beta} = 1$, so $\boldsymbol{\beta} \in E_0$.

The function

$$s(\boldsymbol{\alpha}) = \min_{\boldsymbol{\alpha}' \in E_0} \|\boldsymbol{\alpha} - \boldsymbol{\alpha}'\|$$

is a continuous function in $\boldsymbol{\alpha}$ (minimum of a bivariate continuous function over a compact set).

We have $s(\boldsymbol{\beta}_n) \geqslant f_0$ and $\boldsymbol{\beta}_n \to \boldsymbol{\beta}$, so $s(\boldsymbol{\beta}_n)$ converges to some $f_0' \geqslant f_0$ as s is continuous. However, since $\boldsymbol{\beta} \in E_0$, we have $s(\boldsymbol{\beta}) = 0$. This creates a contradiction and therefore $f(\eta) \to 0$.

Next we want to show $g(\eta) \to 0$ as $\eta \to 0$. Given $\gamma > 0$, as E_0 is compact, we can cover E_0 with at most k balls of radius $\gamma/2$ for some finite k. We label the centres of these balls as $\boldsymbol{\alpha}_j$, $1 \leqslant j \leqslant k$.

We consider the case where $\eta > 0$. The case for $\eta < 0$ is symmetric. By the assumption of the theorem the set $\{\boldsymbol{\alpha} \in \mathcal{A}^S \mid \boldsymbol{g}^T\boldsymbol{\alpha} = 1\}$ is non-empty in the relative interior of $\mathcal{A}^S$. So there exists $\eta > 0$ such that E_η is non-empty. Without loss of generality assume E_1 non-empty (can rescale with any positive constant other than 1), define

$$d_j = \min_{\boldsymbol{\alpha}' \in E_1} \|\boldsymbol{\alpha}_j - \boldsymbol{\alpha}'\|.$$

By convexity (see Fig. 6.3), for $0 < \eta \leqslant 1$,

$$\min_{\boldsymbol{\alpha}' \in E_\eta} \|\boldsymbol{\alpha}_j - \boldsymbol{\alpha}'\| \leqslant \eta \min_{\boldsymbol{\alpha}' \in E_1} \|\boldsymbol{\alpha}_j - \boldsymbol{\alpha}'\| = \eta d_j$$

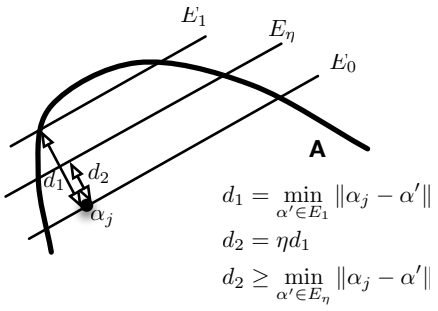


Figure 6.3: Illustration for the proof of Lemma 9

For any $\boldsymbol{\alpha} \in E_0$, it lies within one of the k balls, say $\boldsymbol{\alpha}_j$. We have

$$\begin{aligned}\min_{\boldsymbol{\alpha}' \in E_\eta} \|\boldsymbol{\alpha} - \boldsymbol{\alpha}'\| &\leqslant \min_{\boldsymbol{\alpha}' \in E_\eta} [\|\boldsymbol{\alpha} - \boldsymbol{\alpha}_j\| + \|\boldsymbol{\alpha}_j - \boldsymbol{\alpha}'\|] \\ &= \|\boldsymbol{\alpha} - \boldsymbol{\alpha}_j\| + \min_{\boldsymbol{\alpha}' \in E_\eta} \|\boldsymbol{\alpha}_j - \boldsymbol{\alpha}'\| \\ &\leqslant \frac{\gamma}{2} + \eta d_j\end{aligned}$$

Since the k balls altogether cover E_0, for all $\boldsymbol{\alpha} \in E_0$, when $\eta \leqslant \frac{\gamma}{2 \max_{1 \leqslant j \leqslant k} d_j}$,

$$\begin{aligned}\min_{\boldsymbol{\alpha}' \in E_\eta} \|\boldsymbol{\alpha} - \boldsymbol{\alpha}'\| &\leqslant \frac{\gamma}{2} + \eta \max_{1 \leqslant j \leqslant k} d_j \\ &\leqslant \frac{\gamma}{2} + \frac{\gamma}{2} = \gamma\end{aligned}$$

Hence

$$\max_{\boldsymbol{\alpha} \in E_0} \min_{\boldsymbol{\alpha}' \in E_\eta} \|\boldsymbol{\alpha} - \boldsymbol{\alpha}'\| \leqslant \gamma$$

whenever $\eta \leqslant \min(1, \gamma/(2 \max_{1 \leqslant j \leqslant k} d_j))$. The argument for $\eta < 0$ is symmetric. Therefore $g(\eta) \to 0$ as $\eta \to 0$. ■

Proof of Theorem 4

Proof. We use $\boldsymbol{h}(\boldsymbol{\theta})$ from Eq. (6.1) to denote the cost vector for expected adversarial loss, with the extra argument $\boldsymbol{\theta}$ to emphasize its dependence on $\boldsymbol{\theta}$. As $\boldsymbol{\theta}^\dagger$ is a dominant strategy, we have

$$\begin{aligned}&\boldsymbol{h}(\boldsymbol{\theta}^\dagger)^T \boldsymbol{\alpha} \leqslant \boldsymbol{h}(\bar{\boldsymbol{\theta}})^T \boldsymbol{\alpha} \\ \Rightarrow &(\boldsymbol{h}(\boldsymbol{\theta}^\dagger) - \boldsymbol{h}(\bar{\boldsymbol{\theta}}))^T \boldsymbol{\alpha} \leqslant 0 \qquad (6.4)\end{aligned}$$

for all $\boldsymbol{\alpha} \in \mathcal{A}_\infty$. By definition $\bar{\boldsymbol{\theta}}$ minimizes the adversarial loss for the constant unweighed strategy $\boldsymbol{\alpha}_0$ of the adversary, so we have

$$(\boldsymbol{h}(\boldsymbol{\theta}^\dagger) - \boldsymbol{h}(\bar{\boldsymbol{\theta}}))^T \boldsymbol{\alpha}_0 = 0. \qquad (6.5)$$

Let $\boldsymbol{\alpha}' \in \mathcal{A}_\infty$. As $\boldsymbol{\alpha}_0$ is in the relative interior of $\mathcal{A}_\infty$ and $\mathcal{A}_\infty$ is convex, there exists $\epsilon > 0$ such that

$$\boldsymbol{\alpha}'' = \boldsymbol{\alpha}' + (1+\epsilon)(\boldsymbol{\alpha}_0 - \boldsymbol{\alpha}')$$

is in $\mathcal{A}_\infty$. Now by Eq. (6.4) and (6.5), we have three colinear points such that

$$\begin{aligned}
(\boldsymbol{h}(\boldsymbol{\theta}^\dagger) - \boldsymbol{h}(\bar{\boldsymbol{\theta}}))^T \boldsymbol{\alpha}' &\leqslant 0 \\
(\boldsymbol{h}(\boldsymbol{\theta}^\dagger) - \boldsymbol{h}(\bar{\boldsymbol{\theta}}))^T \boldsymbol{\alpha}_0 &= 0 \\
(\boldsymbol{h}(\boldsymbol{\theta}^\dagger) - \boldsymbol{h}(\bar{\boldsymbol{\theta}}))^T \boldsymbol{\alpha}'' &\leqslant 0.
\end{aligned}$$

So $(\boldsymbol{h}(\boldsymbol{\theta}^\dagger) - \boldsymbol{h}(\bar{\boldsymbol{\theta}}))^T \boldsymbol{\alpha}$ must be identically 0 on the interval $[\boldsymbol{\alpha}', \boldsymbol{\alpha}'']$, as it is a linear function in $\boldsymbol{\alpha}$.

This shows $\boldsymbol{h}(\bar{\boldsymbol{\theta}})^T \boldsymbol{\alpha}' = \boldsymbol{h}(\boldsymbol{\theta}^\dagger)^T \boldsymbol{\alpha}'$. As $\boldsymbol{\alpha}'$ is arbitrary, the unweighed solution $\bar{\boldsymbol{\theta}}$ is also a dominant strategy for the learner Θ. ■

Proof of Theorem 6

Proof. By definition of pointwise dominator

$$\int l(f_{\boldsymbol{\theta}^\star}(x), y) dF(y \mid x) - \int l(f_{\boldsymbol{\theta}'}(x), y) dF(y \mid x) \leqslant 0$$

for all $\boldsymbol{\theta}' \in \Theta$. Given any bounded adversarial set $\mathcal{A}$, any $\boldsymbol{\alpha} \in \mathcal{A}$, $w_{\boldsymbol{\alpha}}(x)$ is a non-negative function of x. Therefore integrating with respect to $dF(x)$ gives

$$\begin{aligned}
&\int w_{\boldsymbol{\alpha}}(x) \left[\int l(f_{\boldsymbol{\theta}^\star}(x), y) dF(y \mid x) - \int l(f_{\boldsymbol{\theta}'}(x), y) dF(y \mid x) \right] dF(x) \leqslant 0 \\
&\int w_{\boldsymbol{\alpha}}(x) l(f_{\boldsymbol{\theta}^\star}(x), y) dF(x, y) \leqslant \int w_{\boldsymbol{\alpha}}(x) l(f_{\boldsymbol{\theta}'}(x), y) dF(x, y).
\end{aligned}$$

Thus $\boldsymbol{\theta}^\star$ is also a dominant strategy against the adversarial set $\mathcal{A}$. ■