Robust Prediction Methods for Covariate Shift and Active Learning

by

Anqi Liu
B.Eng. Software Engineering, Tianjin University of Finance and Economics, 2012
B.S. Finance, Tianjin University of Finance and Economics, 2012

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Chicago, 2018

Chicago, Illinois

Defense Committee:
Brian Ziebart, Chair and Advisor
Bing Liu
Philip Yu
Lev Reyzin, Mathematics, Statistics, and Computer Science
Miro Dudík, Microsoft Research New York City

Dedicated to Mom and Dad.

ACKNOWLEDGMENT

First and foremost, I want to express my sincere gratitude to my advisor, Prof. Brian Ziebart, for his great guidance and support along my Ph.D. years. Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Bing Liu, Prof. Philip Yu, Prof. Lev Reyzin and Dr. Miro Dudík, for their comments and help on the thesis and defense.

I also want to thank my collaborators without whom much of the work in this thesis could not have been done: Dr. Mathew Monfort, Dr. Xiangli Chen, Dr. Hong Wang, Rizal Fathony, Kaiser Asif, Wei Xing, Sima Behpour and Jia Li.

I thank my other fellow labmates: Andrea Tirinzoni, Ashkan Rezaei, Mohammad Bashiri and Sanket Gaurav, for the encouragement and feedback they gave me.

My sincere thanks also goes to Dr. Paul Vernaza in NEC Labs and Nick Rhinehard in CMU, for helping me and collaborating with me in the summer internship projects.

Last but not the least, I would like to thank my family and my boyfriend for the company during these years.

This thesis is based upon work partially supported by these NSF Grants: NSF IIS-1526379, NSF CAREER (RI)-1652530, and NSF III-1514126. I was also supported partially by Future of Life Institute for the thesis work.

AL

CONTRIBUTIONS OF AUTHORS

Chapter 1 is an introduction of the motivation of studying robust prediction methods for covariate shift and active learning. Chapter 2 covers related work about current prevalent methods for dealing with covariate shift in supervised learning and in active learning, problems with those methods, and their relation with our approach. Chapter 3 describes the methodology of our approach, in which the general framework, different loss functions, multi-view version, and the analysis of the general robust prediction method are mainly included in a preprint paper on axiv (Liu and Ziebart, 2017). I collaborated on this paper with my advisor, Prof. Brian Ziebart, and I was the primary author. The reduction of logarithmic loss is covered in a NIPS 2014 paper (Liu and Ziebart, 2014), which I collaborated on with my advisor Prof. Brian Ziebart, and I was the primary author. The kernelized version of the robust prediction framework is the main content in a preprint paper on axiv (Liu et al., 2017), on which I collaborated with my advisor, Prof. Brian Ziebart, and my labmate, Rizal Fathony, and I was the primary author. We first proposed the shift-pessimistic active learning method in a AAAI 2015 paper (Liu et al., 2015), on which I collaborated with my advisor, Prof. Brian Ziebart, and Prof. Lev Reyzin, and I was the primary author. Chapter 4 is the evaluation and application of the methods mentioned in Chapter 3. Chapter 5 concludes the thesis, discusses the limitations and presents future challenges.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

CHAPTER **PAGE**

TABLE OF CONTENTS (Continued)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

SUMMARY

In real world machine learning applications, it is often not very realistic to assume that the training data distribution aligns with the testing data distribution. In order to obtain a reliable predictor for new unseen test data, which could be drawn from a different distribution, a simplification is to assume the distribution shift only occurs on the input variables (covariates), while the conditional output distribution given the input variables (covariates) remains the same. This is called the covariate shift setting. Besides various examples of covariate shift in supervised learning tasks, one of the typical covariate shift scenarios is the sampling bias problem in pool-based active learning, in which the learner selects the labeled set, thus introducing a different input distribution from the unlabeled pool in each step of learning and prediction.

In this thesis, we propose a general framework for robust prediction under covariate shift. Rather than focusing on minimizing a reweighted empirical loss on training data, we manage to more directly optimize the expected test loss with a minimax approach. The resulting predictor provides more randomized predictions on test data when it lacks training data distribution support and therefore avoids possible loss induced by over-optimistic extrapolation of other predictors. This framework allows for facilitating different loss function minimization and incorporating different feature functions and feature generalization assumptions. We discuss how the framework reduces to specific forms and the corresponding approaches to estimate the parameters. Theoretical properties of the robust prediction methods about generalizability and consistency are also included in the study. Moreover, we investigate active learning using robust

prediction when the active learning step is constructed as a special case of robust covariate shift problem.

For the evaluation of the proposed methods, we cover both supervised learning and active learning cases. We provide two-dimensional toy examples for intuitive motivation of the method and its effect on low dimensional and small amounts of data. We conduct experiments on synthetic biased benchmark datasets and natural covariate shift datasets to show performance of the robust prediction on real data. Additionally, we evaluate pool-based active learning using robust prediction on benchmark real data sets. We demonstrate a number of benefits over existing methods.

CHAPTER 1

INTRODUCTION

1.1 Covariate Shift

The *independent and identically distributed* (IID) assumption employed widely across machine learning methods requires the testing data distribution to be the same as the training data distribution. This is quite restrictive in the sense that shift can occur between the training distribution and testing distribution in many settings. For example, survey response rates may vary by individuals' characteristics, medical results may only be available from a non-representative demographic sample, representative data in certain domains may just be very hard and costly to obtain, or dataset labels may have been solicited using active learning. Recent research in bias in machine learning also reminds us of the possible social bias that could exist in the way we collect our data (Crawford et al., 2016). These scenarios make models built on the IID assumption inappropriate and impose huge challenges for building reliable predictors that could work in the wild.

Though nothing can be learned when the shift between training and testing data is arbitrary, certain assumptions about how training and testing distributions differ allow reasonable adaptive learning methods to be derived (Blitzer et al., 2008). One of the common assumptions is that the bias only comes from the input variables. In this setting, also known as **covariate shift**, the distribution of inputs, $\mathsf{P}_{\text{train}}(\mathbf{x})$ and $\mathsf{P}_{\text{test}}(\mathbf{x})$, differ, while the conditional label distribution,

$\mathsf{P}(\mathsf{y}|\mathbf{x})$, is the same under both the training and the testing distributions. This assumption is much weaker than the IID assumption and covers a broad range of real application scenarios.

1.2 Motivating Example: Active Learning

Active learning (Settles, 2012; Tong and Koller, 2001; Sebastiani, 2002) aims to alleviate the burden of labeling an entire training dataset by allowing the predictor to determine what source data it has available to use for learning. In theory, this enables the active learner to solicit labels from the most informative datapoints. This has the potential to significantly improve data-efficiency beyond what is possible with randomly provided labels (Angluin, 1988). However, data produced from a pool-based active learner violates the *independent and identically distributed* (IID) data property broadly assumed by supervised machine learning techniques (Sugiyama and Kawanabe, 2012). Specifically, the labeled set input distribution would change gradually as the active learner picks unlabeled datapoints and adds them to the labeled set, converging only in the limit to the input distribution of the vast unlabeled set. In the process, the conditional label distribution is the same between them. These non-IID samples pose serious pitfalls for active learning methods both in theory and in practice that have not yet been resolved.

Existing active learning methods are optimistic about their own uncertainty. They generally employ an underlying supervised machine learning model and assume that all unlabeled datapoints' labels are distributed according to the model's (often strong) inductive biases. This approach has theoretical justification in IID settings where the inductive biases are shaped by increasing amounts of *representative* data. However, the potential of improved data-efficiency benefits from active learning versus passive supervised learning is only realized by biasing

label solicitation towards *non-representative* data that is more informative than random samples (Settles, 2012). Unfortunately, the combination of optimistic extrapolation based on IID assumptions and intentionally non-IID data collection often leads not only to inefficient learning, but to extreme inaccuracies. Even advocates of existing active learning methods suggest that "random sampling ... may be more advisable than taking one's chances on active learning with an inappropriate learning model" (Settles, 2012).

1.3 Debiasing via Importance Weighting

Under covariate shift, despite the training data distribution, $\mathsf{P}(\mathsf{y}|\mathbf{x})\mathsf{P}_{\text{train}}(\mathbf{x})$, and the testing data distribution, $\mathsf{P}(\mathsf{y}|\mathbf{x})\mathsf{P}_{\text{test}}(\mathbf{x})$, sharing a common conditional label probability distribution, $\mathsf{P}(\mathsf{y}|\mathsf{x})$, all probabilistic classifiers, $\hat{\mathsf{P}}(\mathsf{y}|\mathbf{x})$, are vulnerable to covariate shift when the test data and the inductive bias of the trained classifier do not match (Fan et al., 2005). Therefore, under the classical statistics perspective, a parametric model for the conditional label distribution, $\mathsf{P}(\mathsf{y}|\mathbf{x})$, (e.g., logistic regression) is first chosen and then model parameters are estimated in a manner that accounts for the covariate shift.

A preferred approach is to minimize the prediction loss under the test distribution, which is estimated by reweighting the training samples according to the test-train density ratio, $\mathsf{P}_{\text{test}}(\mathbf{x})/\mathsf{P}_{\text{train}}(\mathbf{x})$ (Shimodaira, 2000; Zadrozny, 2004). Machine learning research has primarily investigated covariate shift from this perspective, with various techniques for estimating the density ratio including kernel density estimation (Shimodaira, 2000), discriminative estimation (Bickel et al., 2009), Kullback-Leibler importance estimation (Sugiyama et al., 2008), kernel mean matching (Huang et al., 2006; Yu and Szepesvári, 2012), maximum entropy meth-

ods (Dudík et al., 2005), and minimax optimization (Wen et al., 2014). Unfortunately, despite asymptotic guarantees of minimizing Bayesian risk (Shimodaira, 2000), sample reweighting is often extremely inaccurate for finite datasets when covariate shift is large (Cortes et al., 2008). In other words, this approach tends to work well when the training and the testing distributions are fairly similar and large amounts of training samples are available. However, when these conditions are violated, i.e., there is only a limited amount of training data and/or significant differences between the training and testing distributions, some of the density ratios for training examples can be extremely large. This leads to high-variance estimates that extrapolate heavily from scant amounts of training data and a lack of generalization guarantees for the resulting predictor (Cortes et al., 2010; Cortes et al., 2008).

1.4 Robust Prediction Framework for Covariate Shift

In this thesis, we develop robust estimation approaches under covariate shift as an alternative to importance weighting. Instead of the classical machine learning perspective, where a parametric model for the conditional label distribution, $\mathsf{P}(\mathsf{y}|\mathbf{x})$, is first chosen and then model parameters are estimated like in importance weighting methods, we propose to robustly optimize the desired loss function directly, i.e., using the exact expected test loss, without a pre-chosen parametric form of the predictor. We enable this using a minimax estimation formulation.

In our general framework, we robustly minimize the expected test loss subject to known properties of data from the training data distribution. The resulting predictor needs to match statistical properties measured from the training distribution, but is otherwise the most uncertain on the test distribution. These statistics can be estimated without the inaccuracies introduced

by importance weighting from the training to the test distribution. Our formulation requires any assumptions of statistical properties generalizing beyond the training distribution to be explicitly incorporated into the classifier's construction. In other words, we can choose a broader distribution that training features generalize to explicitly to obtain adaptation. For example, certain regional demographic statistics may also apply to the national population. Therefore, our framework actually produces a family of predictors and even the importance weighting logarithmic loss minimization is a special case of our framework for a particularly strong assumption: that training statistics fully generalize to the test distribution. Additionally, the general framework accounts for different loss function minimization, among which logarithmic loss is the most convenient one since it results in an analytic parametric form and the optimization is convex. We show that the framework can also serve non-smooth losses like the zero-one loss and Hamming loss. We kernelize the general framework to obtain consistency guarantees for the predictor. We also develop an approach for incorporating partial feature generalization on multi-view data. We apply several predictors resulting from our robust prediction framework on synthetic and UCI binary classification datasets to compare its performance against importance weighting approaches for learning under covariate shift.

By explicitly considering a pool-based active learning setting as a special case of covariate shift supervised learning, we develop a pessimistic approach to active learning that avoids inefficiencies created by the combination of optimism and non-representative label solicitation. The active learning method leverages one of the most classical predictors from our framework for learning under covariate shift, which minimizes logarithmic loss and obtains the most conservative

predictions by assuming labeled set features do not generalize to the unlabeled set (Liu and Ziebart, 2014). We then use its resulting predictor to guide effective label solicitation strategies. Under this approach, we show that model uncertainty is closely calibrated to generalization loss. Thus, common label solicitation strategies guided by model uncertainty tend to directly improve the model's predictive performance. In addition to the theoretical properties, we evaluate and compare the effectiveness of our approach on a range of classification tasks.

1.5 Outline of the Thesis

In this thesis, we answer the question: How can we make robust predictions under covariate shift in supervised learning and active learning? By discussing the methodology, analysis and application of the proposed robust prediction methods for covariate shift and active learning, we provide a robust, flexible and accurate approach to solving the problem. Following is an outline of the thesis. We cover related work in Chapter 2 and start introducing the main contribution of the thesis after that. Chapter 3 is the methodology, which covers the major definition and theorems for deriving the general framework and its different forms when certain loss functions and feature generalization assumptions are incorporated. We focus on logarithmic loss and zero-one loss, as well as kernelizing features and multiview features. Chapter 3 also covers pool-based active learning approaches with robust predictions and analysis of the framework, which includes generalization bounds and consistency properties when equipped with kernels. Chapter 4 is the application section, consisting of toy examples and evaluations of several proposed methods on various datasets. We conclude the thesis in Chapter 5 and also discuss future research opportunities and challenges.

CHAPTER 2

BACKGROUND AND RELATED WORK

2.1 Transfer Learning and Domain Adaptation

Covariate shift is regarded as a special case of sample selection bias (Zadrozny, 2004), which is the systematic error brought by non-random samples of a population. It causes some portion of the population to be more likely to be sampled than others. We assume that a data point can be represented as a feature vector and a label, $(\mathbf{x}_i, y_i)$ and s_i is a variable that shows whether the data point is selected. If $s_i = 1$, the data point i is selected. Then there are four cases according to whether s_i is independent from feature vector $\mathbf{x}_i$ and data label y_i:

- a) s_i is independent from both $\mathbf{x}_i$ and y_i;
- b) s_i is only independent from y_i given $\mathbf{x}_i$ but not independent from $\mathbf{x}_i$;
- c) s_i is only independent from $\mathbf{x}_i$ given y_i but not independent from y_i;
- d) s_i is dependent on both $\mathbf{x}_i$ and y_i.

The first case matches the I.I.D. condition, in which case the sample is a random sample of the population. The remaining three cases each carry different types of sample selection bias. The third case means there is a change in the prior probabilities of the labels. This type of bias often leads to imbalanced classification, and has been studied using different methods like cost-sensitive learning in the machine learning literature (Elkan, 2001; Tang et al., 2009; Li et al., 2011; López et al., 2012; Huang et al., 2016). The fourth case means there is no independence

assumption that holds for $\mathbf{x}_i$, y_i and s_i, and we then cannot expect to learn a mapping from features to labels using the selected sample if just given the feature vectors. Finally, the second case where $P(s|\mathbf{x}, y) = P(s|\mathbf{x})$ is called covariate shift, which means the selected sample is biased but the bias only depends on the feature vector $\mathbf{x}$.

As a special case of sample selection bias (Heckman, 1977), covariate shift has ties to more general domain adaptation (Jiang, 2008) and transfer learning settings (Pan and Yang, 2010), where the assumption on how training and testing data distributions differ is not fully specified. For example, even though the importance weighting method is developed based on the covariate shift assumption, it is also applied to and studied for more general domain adaptation settings with more sophisticated methods of estimating weights for training data (Cortes and Mohri, 2014; Mansour et al., 2009a). Additionally, a wide range of approaches for learning under covariate shift and transfer learning leverage additional assumptions or knowledge to improve predictions (Pan and Yang, 2010). For example, a simple, but effective approach to domain adaptation (Daumé III, 2007) leverages some labeled test data to learn some relationships that generalize across source and target datasets. Another recent method assumes that training and testing data are generated from mixtures of "domains" and uses a learned mixture model to make predictions of test data based on more similar training data (Gong et al., 2013).

In this thesis, we focus on the covariate shift setting and investigate those related works that explicitly assume the covariate shift assumption is valid and omit many of the domain adaptation works that focus on more general distribution shift assumptions. The specific setting we focus on is then with only labeled training data and unlabeled testing data.

2.2 Empirical Risk Minimization, Surrogate Loss and Consistency

Empirical Risk Minimization (ERM) is an important principle that is assumed by many machine learning algorithms (Mohri et al., 2012). Supervised learning is a general setting that we have two spaces of objects $\mathbf{X}$ and Y and would like to learn a function h which maps from $\mathbf{X}$ to Y. Assuming L is the loss function that measures how we penalize the predictive difference between the prediction $h(\mathbf{x})$ of a hypothesis and the true outcome y. The risk of $h(\mathbf{x})$ is then defined as the expectation of the loss function:

$$R(h) = \mathbb{E}_{P(\mathbf{x},y)}[L(h(\mathbf{x}), y)], \tag{2.1}$$

where $P(\mathbf{x}, y)$ is the joint probability distribution over $\mathbf{X}$ and Y. Our goal of the learning tasks is to find a h^* among a class of functions H for which the expected risk $R(h)$ is minimal. However, $P(\mathbf{x}, y)$ is not available to learning algorithms. We then use an approximation evaluated on a finite number of training samples that are drawn independently and identically from the distribution $P(\mathbf{x}, y)$. The approximated risk on empirical training data is called the empirical risk:

$$\hat{R}(h) = \mathbb{E}_{\tilde{P}(\mathbf{x},y)}[L(h(\mathbf{x}), y)] = \frac{1}{m}\sum_{i} L(h(\mathbf{x}_i), y_i), \tag{2.2}$$

where m is the number of training samples. ERM then defines algorithms that find h to minimize the empirical risk.

The optimal predictor that minimizes the desired loss function $\mathfrak{l}$ is the Bayes optimal predictor. However, for the loss functions we care about in many tasks, finding the optimal predictor is not a convenient problem to optimize. For example, the 0-1 loss is the desired loss function for many predictive tasks that care about high accuracy. Unfortunately, it is discontinuous and therefore challenging to optimize directly. It is usual to use a proxy, which is the so-called surrogate loss function Φ. A convex function Φ is commonly chosen for computational reasons. For example, support vector machines use the hinge loss and logistic regression uses the logarithmic loss — each forms convex upper bound on the 0-1 loss. Thus, using convex surrogate loss provides a much more tractable optimization problem.

A natural question to ask is how much is lost in terms of prediction quality by this change of loss function. Whether minimizing Φ-risk leads to a function that minimize $\mathfrak{l}$-risk is referred to as the consistency or calibration property (Bartlett et al., 2006; Tewari and Bartlett, 2007). It depends on both the surrogate loss and the true desired loss. In binary classification, the most commonly used loss functions like hinge, exponential and logarithmic loss are all consistent with 0-1 loss. But in more sophisticated tasks, including multi-class classification and structured prediction, many surrogates are not consistent. There are also methods that work generally well in practice but are actually not consistent. One example is the famous Crammer-Singer multi-class hinge loss used by SVM methods (Crammer and Singer, 2001).

2.3 Minimax Robust Estimation

Minimax robust estimation (Topsøe, 1979; Grünwald and Dawid, 2004) is a powerful technique for constructing classifiers that assumes the worst case about unknown properties of probability

distributions. This formulation departs from the traditional statistical perspective by prescribing a parametric predictor (not necessarily with closed-form) that, apart from matching known distribution statistics, is the worst-case distribution possible for a given loss function. This provides a strong rationale for maximum entropy estimation methods (Jaynes, 1957) from which many familiar exponential family distributions (e.g., Gaussian, exponential, Laplacian, logistic regression, conditional random fields (Lafferty et al., 2001)) result by robustly minimizing logloss subject to constraints incorporating various known statistics.

In the robust estimation framework, the slack in the moment-matching constraints in the primal is closely related to the regularization weights in the dual formulation (Dudík and Schapire, 2006). In other words, the amount of regularization in logistic regression, for example, corresponds with how much relaxation we allow for the adversary player to have when matching the constraints.

2.4 Recent Advances in Adversarial Risk Minimization

Recent research about minimax robust estimation focuses on robustly minimizing non-smooth loss functions (Asif et al., 2015; Fathony et al., 2016; Farnia and Tse, 2016; Fathony et al., 2017), the relation of the resulting predictor with empirical risk minimization, and effectively applying this method to optimizing multivariate performance measures (Wang et al., 2015), structured prediction tasks based on sequences (Li et al., 2016), computer vision tasks (Behpour et al., 2017), and Inverse Optimal Control (Chen et al., 2016b). It has been shown that minimization of certain loss functions in the minimax robust estimation is equivalent with empirically minimizing convex surrogate loss functions. In those cases, there are more efficient ways or even analytical solution

forms for solving the minimax game. Also, it has been proved that this series of adversarial approaches is consistent with minimizing the desired loss when equipped with expressive feature constraints (Fathony et al., 2016; Li et al., 2016). In order to solve the potentially large minimax games, especially in the structured prediction case, it is necessary to apply a double oracle constraint generation method (McMahan et al., 2003). It provides an exact solution, which leads to good empirical predictive performance. But it could be slow in convergence sometimes.

In this thesis, we follow the same thread of work in minimax robust estimation and use it to deal with covariate shift problems, in which the loss functions evaluation distribution is different from the available data input distribution that forms the constraints for the adversary player. However, since the conditional label distribution is shared between training and testing under covariate shift, training features can still form the constraints set for the adversary effectively. This enables us to optimize using (sub-)gradient descent that only requires training data for evaluation.

2.5 Importance Weighting Methods

The most prevalent approach for addressing covariate shift attempts to remove the bias between the training and testing distributions (Shimodaira, 2000; Huang et al., 2006; Sugiyama et al., 2008) by making the training samples more representative of the test distribution. Under this perspective, minimizing the importance-weighted loss of (n) training examples,

$$\lim_{n\to\infty} \min_{\hat{f}} \mathbb{E}_{(\mathbf{X},Y)\sim\tilde{P}^{(n)}_{\text{train}}} \left[\frac{P_{\text{test}}(\mathbf{X})}{P_{\text{train}}(\mathbf{X})} \text{loss}(\hat{f}(\mathbf{X}), Y) \right] = \min_{\hat{f}} \mathbb{E}_{(\mathbf{X},Y)\sim P_{\text{test}}} \left[\text{loss}(\hat{f}(\mathbf{X}), Y) \right], \tag{2.3}$$

where $\hat{\mathsf{f}}$ is estimated predictor and $\tilde{\mathsf{p}}$ is the empirical distribution of data, asymptotically minimizes the testing distribution loss, so long as $\mathsf{P}_{\text{test}}(\mathbf{x}) > 0 \implies \mathsf{P}_{\text{train}}(\mathbf{x}) > 0$.

Despite this asymptotic guarantee, predictive performance can be poor when training from finite amounts of samples in both theory and practice. Conceptually, the density ratios of a small number of training examples can become disproportionately large, making the resulting predictor overly sensitive to a small number of training data points—or even one single datapoint. This will lead to predictive results with high variance. Related to this general difficulty of estimating test distribution loss, finite generalization bounds for importance-weighted methods require finite second moments:

$$\mathbb{E}_{\mathsf{P}_{\text{train}}(\mathbf{x})}[(\mathsf{P}_{\text{test}}(\mathbf{X})/\mathsf{P}_{\text{train}}(\mathbf{X}))^2] < \infty \tag{2.4}$$

(Cortes et al., 2010), which is often not satisfied in practice.

In Figure 1, there are two examples showing two Gaussian distributions overlapped in different ways. The data points with larger weights are annotated. There are only a few of them but they dominate the resulting importance weighting predictor due to this large importance weights, $\frac{\mathsf{P}_{\text{test}}(\mathbf{x})}{\mathsf{P}_{\text{train}}(\mathbf{x})}$.

In order to overcome these difficulties, there is a significant literature studying how to reasonably estimate the weights $\frac{\mathsf{P}_{\text{test}}(\mathbf{x})}{\mathsf{P}_{\text{train}}(\mathbf{x})}$ from training and testing sample data (Gretton et al., 2009). For example, methods based on minimizing certain types of divergences or loss functions on density (ratios) (Sugiyama et al., 2008; Kanamori et al., 2009) have been investigated. Other

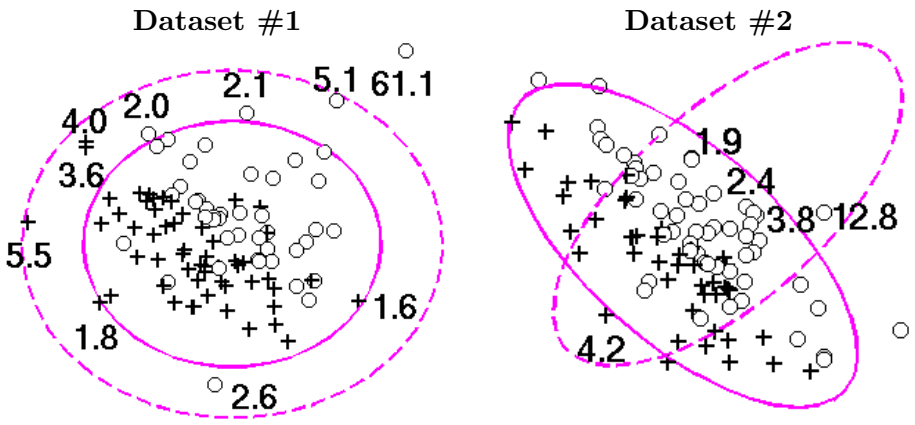


Figure 1: Datapoints (with '+' and 'o' labels) from two source distributions (Gaussians with solid 95% confidence ovals) and the largest data point importance weights, $\frac{\mathsf{P}_{\text{test}}(\mathbf{x})}{\mathsf{P}_{\text{train}}(\mathbf{x})}$, under the target distributions (Gaussian with dashed 95% confidence ovals).

methods (Cortes and Mohri, 2014) make implicit assumptions on the feature space that there exist weights $w(\mathbf{x})$ that makes the "distance" between training and testing features small enough so that reweighted training features can be used to learn classifiers on testing distribution. Other work focuses on utilizing two stages of regularization to reduce the variance of the resulting predictions (Reddi et al., 2014).

2.6 Other Minimax Approaches to Covariate Shift

Though developed using similar tools, previous minimax formulations of learning under covariate shift differ substantially from our approach. They consider the test distribution as being unknown and provide robustness to its worst-case assignment. The class of test distributions considered are those obtained by deleting a subset of measured statistics (Globerson et al., 2009) or all possible reweightings of the sample training data (Bagnell, 2005; Wen et al., 2014).

A recent study of the so-called Distributionally Robust Supervised Learning (DRSL) covers several types of minimax games between model parameters and either adversarial weights or adversarial samples. DRSL with Wasserstein distance (Ben-Tal et al., 2013; Esfahani and Kuhn, ; Blanchet et al., 2016; Sinha et al., 2017) tries to be robust to adversarial examples. DRSL with f-divergences (Duchi et al., 2016; Namkoong and Duchi, 2016; Hu et al., 2016) obtains robustness against adversarial weights of data points. Our approach, in contrast, obtains an estimate for each given test distribution that is robust to all the conditional label distributions matching training statistics.

2.7 Bayesian Methods for Robust Learning

Bayesian methods like Gaussian processes do not address the covariate shift directly. Instead, these methods are often regarded as means for controlling the predictive certainty easily when needed, which makes it popular in active learning settings (Kapoor et al., 2007; Krause and Guestrin, 2007; Hoang et al., 2014). If just viewed from the I.I.D. case, i.e., ignoring the covariate shift, Bayesian methods provide an alternative to maximum likelihood for estimating the parameters of a parametric predictive model using a probability distribution. However, in the covariate shift setting, given known differences between $\mathsf{P}_{\text{train}}(\mathbf{x})$ and $\mathsf{P}_{\text{test}}(\mathbf{x})$ and no labels from test data distribution, the likelihood function needed for evaluation is not available and it is not clear whether a Bayesian perspective is possible under our framework. Moreover, Bayesian Linear Regression is briefly mentioned in a previous work (Chen et al., 2016a) with key differences from the regression model derived from our robust framework (Chen et al., 2016a) because it is minimizing the empirical squared loss evaluated on training data. When

the concept for robust learning is not limited to covariate shift, more general robust Bayesian probabilistic models are proposed (Wang et al., 2017), which focus on Bayesian data reweighting and estimate the weights and model parameter simultaneously.

2.8 Methods for Covariate Shift in Active Learning

A pool-based active learner (Lewis and Gale, 1994) sequentially chooses datapoint labels to solicit from a set (pool) of unlabeled datapoints, $(\mathbf{x}_i) \in \mathcal{U}$. The learner often constructs an estimate of the conditional label distribution, $\hat{\mathsf{P}}(y|\mathbf{x})$, from its labeled dataset $(\mathbf{x}_j, y_j) \in \mathcal{L}$. It uses this estimate to select the next datapoint label to solicit. We denote the entire set of labeled and unlabeled datapoints as $\mathcal{D} = \mathcal{U} \cup \mathcal{L}$. Numerous metrics have been developed to assess the expected utility of a datapoint. The most common, uncertainty sampling (Lewis and Gale, 1994; Settles, 2012), solicits datapoint labels for which the active learner is least certain. The value-conditioned entropy,

$$\mathsf{H}(Y|\mathbf{X} = \mathbf{x}_i) \triangleq \mathsf{E}_{\hat{\mathsf{P}}(y|\mathbf{x})}[-\log \hat{\mathsf{P}}(Y|\mathbf{X})|\mathbf{x}_i] = -\sum_{y \in \mathcal{Y}} \hat{\mathsf{P}}(y|\mathbf{x}_i) \log \hat{\mathsf{P}}(y|\mathbf{x}_i), \tag{2.5}$$

is often used to measure this uncertainty. Other metrics assess how a datapoint label: (a) is expected to change the prediction model (Settles and Craven, 2008); (b) reduces an upper bound on the generalization error in expectation (Mackay, 1992); or (c) represents the input patterns of remaining unlabeled data (Settles, 2012). We show the general label solicitation process of pool-based active learning in Algorithm 1.

Algorithm 1 Label solicitation for pool-based active learner

Input: unlabeled pool dataset $\mathcal{U}$, labeled dataset $\mathcal{L}$

Output: example $\mathbf{x}_i \in \mathcal{U}$ to solicit label

Estimate $\hat{\mathsf{P}}(\mathsf{y}|\mathbf{x})$ from dataset $\mathcal{L}$

Compute $\text{value}_i \leftarrow \text{metric}(\mathbf{x}_i)$ for each $\mathbf{x}_i \in \mathcal{U}$

return $\mathbf{x}_{\text{argmax}_i \text{ value}_i}$ (example label to solicit)

Even when the entire pool of data can be fit to a particular model reasonably well, active learning from a small number of samples may fail badly. For example, a logistic regression model fits the synthetic dataset of Figure 2 *in its entirety* with fairly small average prediction loss. However, the optimistic active learner solicits a sequence of labels that does not uncover this appropriately fit model, as shown in Figure 2. This is primarily because it solicits labels for examples that would be the most useful *if* its current inductive biases were correct. After obtaining an initial '+'-class label and a noisy second '*'-class label from the bottom-left-most datapoint, the active learner forms an incorrect inductive bias—that a decision boundary for the dataset as a whole exists between those two datapoints—and exhaustively solicits labels to better define the belief contours of this incorrect decision boundary until eventually soliciting a more representative sample of labeled datapoints.

Inherent sample selection bias exists in pool-based active learning because examples for label solicitation are not chosen completely at random (Sugiyama and Kawanabe, 2012). Since the

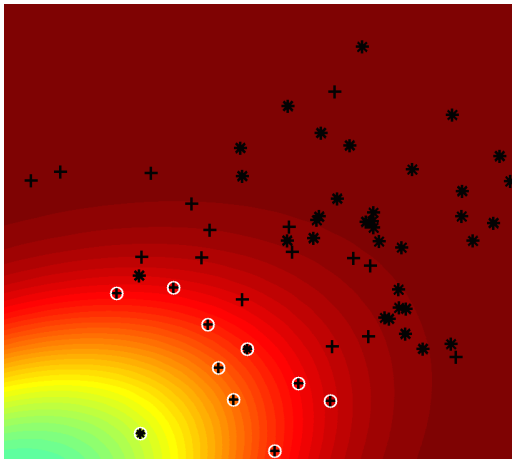


Figure 2: The predictions of active learning using an optimistic logistic regression active learner. Solicited labels ('+' and '*' classes, denoted as red and blue prediction beliefs, respectively), are selected using uncertainty sampling, and indicated with circles.

active learner can only select examples based on the input values, x_i, independently from the unknown label, y_i, this corresponds to *covariate shift* (Shimodaira, 2000).

As illustrated in Figure 2, the basic pool-based active learning algorithm often performs poorly in practice (Attenberg and Provost, 2011). Ad-hoc modifications to the algorithm dealing with this covariate shift that limit the power of the active learner—undermining the purported benefits of active learning—are often required for existing active learners to be competitive with random sampling. These modifications decrease the potential for bias in the labeled dataset by making the label solicitation strategy more similar to random sampling. One modification is to "seed" the learner with a set of randomly drawn datapoint labels (Schein and Ungar, 2007; Dligach and Palmer, 2011). In other words, the active learner is restricted to sampling uniformly for

its first $\mathfrak{n}$ datapoints. A second modification solicits labels from a very small random subset of the unlabeled dataset (e.g., a pool of 10 examples (Schein and Ungar, 2007)) rather than the entire unlabeled dataset, $\mathcal{U}$. These modifications treat the symptoms resulting from IID modeling and non-IID label solicitation rather than its fundamental cause. Unfortunately, there is no universally working strategy for overcoming the covariate shift. That is why in many cases, people use mixed strategies or heuristics to solicit both representative and informative data (Huang et al., 2010).

Active learning using importance weighting has been investigated in a handful of learning tasks (Kanamori and Shimodaira, 2003; Sugiyama, 2005; Bach, 2007). Common label solicitation strategies often produce labeled training data that is highly non-representative, as shown in Figure 2. For infinite pools of data, these strategies violate bounded second moments in (Equation 2.4) asymptotically and produce high-variance predictions from small amounts of data.

In stream-based active learning, unlabeled data points come in a stream, and the active learner decides whether the data point is "valuable" to be labeled, or just skips it (Micchelli et al., 2006; Loy et al., 2012). Importance Weighted Active Learning (IWAL) (Beygelzimer et al., 2009; Karampatziakis and Langford, 2010) is a streaming-based active learning method that generates its weights from the resulting loss from previous steps. The importance weights in their context refer to weighting examples according to the predictive errors, which is similar with algorithms in the boosting family (Schapire and Singer, 1999). Moreover, stream-based active learning has a worse convergence property in terms of the required number of samples

needed to learn a predictor in theory (Sabato and Hess, 2016), even though it is much cheaper to solicit labels.

2.9 Analysis for Covariate Shift

There is a series of research about learning bounds for domain adaptation, among which Ben-David *et al.* focus on the theoretical analysis of statistical learning bounds for covariate shift (Ben-David et al., 2007; Ben-David et al., 2010). That thread of work gives a bound on target generalization error given the presence of mismatched distributions. Advanced analysis follows up (Blitzer et al., 2008; Mansour et al., 2009a; Germain et al., 2013) and also extends the analysis to domain adaptation between multiple sources or a combination of source and target domains (Mansour et al., 2009b; Mansour et al., 2009c). These analyses follow the same philosophy as Ben-David *et al.* where the test error of the model is upper bounded by addition of three terms: the model error on the training distribution, the distance measurement between the training and testing marginal distributions, and a term related to the adaptive property, which is usually hard to estimate. Therefore, it motivates a set of algorithms that minimize a measure of divergence between the distributions to deal with domain adaptation. The focus of the analysis also varies later from classification to regression (Cortes and Mohri, 2014) and includes a new perspective that focuses on learning weighted majority votes (Germain et al., 2016).

Despite the breadth of this line of research, all of the analysis starts with the empirical performance on training data and uses divergence measurements to relate it to testing performance. The measure of divergence often takes the form of an upper bound of the difference of

certain functions evaluated on training and testing. Our formulation directly deals with the test expected loss on a pre-specified prediction function, which is different by design from those methods using empirical risk minimization. Thus, we develop new perspectives of theoretical analysis for the robust prediction framework in this thesis.

2.10 AI Safety

AI safety research has been brought to public attention recently. Even though the breakthrough to artificial general intelligence is still far, many problems have been identified as primary subjects to study in order to build safer artificial intelligence systems in the future (Amodei et al., 2016). Some of the goals (Hadfield-Menell et al., 2017) are related with our subject in this thesis, including:

- Robustness: "How can we make predictions robust to novel or adversarial examples? How can we handle corrupted training data?"
- Awareness: "How do we make a system aware of its environment and of its own limitations, so that it can recognize and signal when it is no longer able to make reliable predictions or decisions? Can it successfully identify strange inputs or situations and take appropriately conservative actions?"
- Adaptation: "How can machine learning systems detect and adapt to changes in their environment, especially large changes (e.g. low overlap between train and test distributions)? How should an autonomous agent act when confronting radically new contexts?"

Our robust prediction method provides a possible solution to predicting in new contexts, which is more narrowly defined as covariate shift in this thesis. The resulting predictor is robust to

large shifts and produces its most uncertain predictions when it detects the lack of training information by estimating the density. It predicts more conservatively or even gives totally uncertain predictions in the extreme case. Thus, it is able to admit when it does not know an answer. The adaption is then controlled by the feature generalization in our framework. When equipped with a generalization distribution for training features, we are able to set it explicitly and obtain adaptation.

CHAPTER 3

METHODOLOGY: ROBUST PREDICTION FRAMEWORK FOR COVARIATE SHIFT

3.1 General Form of the Robust Framework for Covariate Shift

Our goal is to construct a predictor that is robust to the worst case testing distribution implied by available training data. We now provide a general form of the robust prediction methods based on the minimax estimation for covariate shift that incorporates a set of loss functions and generalization assumptions between training and testing distributions. We then introduce variants and special cases that are applicable for different scenarios.

To allow flexibility in what the training data can imply, we assume there exists a generalization distribution, $\mathsf{P}_{\text{gen}}(\mathbf{x})$, where features of training data are assumed to generalize. We then apply the robust method for covariate shift to the generalization distribution instead of the original training distribution $\mathsf{P}_{\text{train}}(\mathbf{x})$, as shown in the following definition:

Definition 1. *The* ***generalized robust covariate shift classifier*** *results from the adversarial loss optimization game:*

$$\min_{\hat{\mathsf{P}}} \max_{\check{\mathsf{P}}} \mathbb{E}_{\mathbf{X}\sim\mathsf{P}_{test}} \left[Loss(\check{\mathsf{P}}_{\mathbf{X}}, \hat{\mathsf{P}}_{\mathbf{X}}) \right] \quad \textit{such that:} \tag{3.1}$$

$$\mathbb{E}_{\mathbf{X}\sim\tilde{\mathsf{P}}_{gen},\check{\mathsf{Y}}|\mathbf{X}\sim\check{\mathsf{P}}} \left[\phi(\mathbf{X}, \check{\mathsf{Y}}) \right] = \mathbb{E}_{(\mathbf{X},\mathsf{Y})\sim\tilde{\mathsf{P}}_{train}} \left[\frac{\mathsf{P}_{gen}(\mathbf{X})}{\mathsf{P}_{train}(\mathbf{X})} \phi(\mathbf{X}, \mathsf{Y}) \right], \tag{3.2}$$

with a loss function that we want to minimize and a distribution, $\mathsf{P}_{gen}(\mathbf{X})$, to which training data feature statistics are assumed to generalize.

Note that the statistics in the constraints are reweighted training sample statistics. This provides us with flexibility to impose different assumptions—in terms of $\mathsf{P}_{\text{gen}}(\mathbf{x})$ densities—for how training data should generalize.

Strong Lagrangian duality holds when $\text{Loss}(\cdot, \cdot)$ is a concave-convex function of $\check{\mathsf{P}}$ and $\hat{\mathsf{P}}$. This enables us to re-write the game in terms of Lagrangian multipliers θ:

$$\min_{\theta} \min_{\hat{\mathsf{P}}} \max_{\check{\mathsf{P}}} \mathbb{E}_{\mathbf{X} \sim \mathsf{P}_{\text{test}}} \left[\text{Loss}(\check{\mathsf{P}}_{\mathbf{X}}, \hat{\mathsf{P}}_{\mathbf{X}}) + \frac{\mathsf{P}_{\text{gen}}(\mathbf{X})}{\mathsf{P}_{\text{test}}(\mathbf{X})} \theta \cdot \phi(\mathbf{X}, \check{Y}) \right] - \theta \cdot \tilde{\phi} + \epsilon ||\theta||_2, \tag{3.3}$$

where $\tilde{\phi} \triangleq \mathbb{E}_{(\mathbf{X},Y) \sim \tilde{\mathsf{P}}_{\text{train}}} \left[\frac{\mathsf{P}_{\text{gen}}(\mathbf{X})}{\mathsf{P}_{\text{train}}(\mathbf{X})} \phi(\mathbf{X}, Y) \right]$ is the feature function evaluated on empirical training data, and we allow ϵ slack for matching the primal constraints, leading to regularization in the dual. The optimization of this objective function is then composed of two steps: first, solve the inner minimax game with respect to $\hat{\mathsf{P}}$ and $\check{\mathsf{P}}$; second, optimize for θ in the outer minimization to satisfy imposed constraints. We focus our attention on 0-1 loss and logarithmic loss, but many other loss functions can also be incorporated.

3.2 Logarithmic Loss Case

3.2.1 With No Feature Generalization: RBA predictor

When the loss function is logarithmic loss, probabilistic classification performance is measured by the conditional logloss (the negative conditional likelihood) of the estimator, $\hat{\mathsf{P}}(Y|\mathbf{X})$,

$$\text{logloss}_{\mathsf{P}_{\text{test}}(\mathbf{X})}\left(\mathsf{P}(Y|\mathbf{X}), \hat{\mathsf{P}}(Y|\mathbf{X})\right) \triangleq -\sum_{\mathbf{x}} \mathsf{P}_{\text{test}}(\mathbf{x}) \sum_{y} \mathsf{P}(y|\mathbf{x}) \log \hat{\mathsf{P}}(y|\mathbf{x}), \tag{3.4}$$

under an evaluation distribution (i.e., the testing distribution, $\mathsf{P}_{\text{test}}(\mathbf{X})\mathsf{P}(Y|\mathbf{X})$, for the covariate shift setting).

We assume that a set of empirically-measured statistics, $\tilde{\mathbf{c}} = \frac{1}{N}\sum_{(\mathbf{x}_i, y_i)\sim \mathsf{P}_{\text{train}}(\mathbf{x})\mathsf{P}(y|\mathbf{x})} \phi(\mathbf{x}_i, y_i)$, characterize the training distribution, $\mathsf{P}_{\text{train}}(\mathbf{x}, y)$. Additionally, we assume $\mathsf{P}_{\text{gen}}(\mathbf{x}) = \mathsf{P}_{\text{train}}(\mathbf{x})$, which means the features do not generalize to the testing distribution at all. Using the loss function (Equation 3.4) and these statistics as constraints, Definition 2 forms a robust minimax estimate (Topsøe, 1979; Grünwald and Dawid, 2004) of the conditional label distribution, $\hat{\mathsf{P}}(Y|\mathbf{X})$.

Definition 2. *The bias-adaptive robust probabilistic classifier is the fixed-point solution of:*

$$\min_{\hat{\mathsf{P}}(Y|\mathbf{X})\in\Delta} \max_{\mathsf{P}(Y|\mathbf{X})\in\Delta\,\cap\,\Xi} \textit{logloss}_{\mathsf{P}_{test}(\mathbf{X})}\left(\mathsf{P}(Y|\mathbf{X}), \hat{\mathsf{P}}(Y|\mathbf{X})\right), \tag{3.5}$$

where Δ *is the conditional probability simplex:* $\forall \mathbf{x} \in \mathcal{X}, y \in \mathcal{Y}, \mathsf{P}(y|\mathbf{x}) \geq 0$*;* $\forall \mathbf{x} \in \mathcal{X}, \sum_y \mathsf{P}(y|\mathbf{x}) = 1$ *and* Ξ *represents the set of distributions matching measured training statistics:* $\mathbb{E}_{\mathsf{P}_{train}(\mathbf{x})\mathsf{P}(y|\mathbf{x})}[\phi(\mathbf{X}, Y)] = \tilde{\mathbf{c}}$.

This formulation can be interpreted as a two-player game (Grünwald and Dawid, 2004) in which the *estimator player* first chooses $\hat{\mathsf{P}}(\mathsf{Y}|\mathbf{X})$ to minimize the conditional logloss and then the *evaluation player* chooses distribution $\mathsf{P}(\mathsf{Y}|\mathbf{X})$ from the set of statistic-matching conditional label distributions to maximize conditional logloss. This minimax game reduces to a maximum conditional entropy (Jaynes, 1957) problem:

Theorem 1. *The solution of the minimax logloss game* (Equation 3.5) *maximizes the testing distribution conditional entropy subject to matching statistics on the training distribution:*

$$\max_{\hat{\mathsf{P}}(\mathsf{Y}|\mathbf{X})\in\Delta} \mathsf{H}_{\mathsf{P}_{test}(\mathbf{x}),\hat{\mathsf{P}}(\mathsf{y}|\mathbf{x})}(\mathsf{Y}|\mathbf{X}) \triangleq -\sum_{\mathbf{x}\in\mathcal{X}} \mathsf{P}_{test}(\mathbf{x}) \overbrace{\sum_{\mathsf{y}\in\mathcal{Y}} \hat{\mathsf{P}}(\mathsf{y}|\mathbf{x}) \log \hat{\mathsf{P}}(\mathsf{y}|\mathbf{x})}^{\mathsf{H}_{\hat{\mathsf{P}}}(\mathsf{Y}|\mathbf{X}=\mathbf{x})} \tag{3.6}$$

$$\textit{such that: } \mathbb{E}_{\mathsf{P}_{train}(\mathbf{x})\hat{\mathsf{P}}(\mathsf{y}|\mathbf{x})}[\phi(\mathbf{X},\mathsf{Y})] = \tilde{\mathbf{c}},$$

in which the conditional entropy can be viewed as an affine function of value-dependent conditional entropies, $\mathsf{H}(\mathsf{Y}|\mathbf{X}=\mathbf{x})$.

Conceptually, the solution to this optimization has low certainty (i.e., large value-conditioned entropy, $\mathsf{H}(\mathsf{Y}|\mathbf{X}=\mathbf{x})$) where the testing density is high by matching the training distribution statistics (and often incurring small value-conditioned entropy) primarily where the testing density is low.

Proof. Our proof (and theorem) follows the classic maximum entropy paper (Grünwald and Dawid, 2004). The two-player game in Definition 1 can be written as:

$$\min_{\hat{\mathsf{P}}(Y|\mathbf{X})\in\Delta} \max_{\check{\mathsf{P}}(Y|\mathbf{X})\in\Delta\cap\Xi} \mathbb{E}_{\mathsf{P}_{\text{test}}(\mathbf{x})\check{\mathsf{P}}(y|\mathbf{x})}[-\log\hat{\mathsf{P}}(Y|\mathbf{X})].$$

Assuming the constraint set Ξ is convex and a solution exists on the relative interior of the set, strong duality holds and switching the order of the two players yields a solution with equivalent value:

$$\max_{\check{\mathsf{P}}(Y|\mathbf{X})\in\Delta\cap\Xi} \min_{\hat{\mathsf{P}}(Y|\mathbf{X})\in\Delta} \mathbb{E}_{\mathsf{P}_{\text{test}}(\mathbf{x})\check{\mathsf{P}}(y|\mathbf{x})}[-\log\hat{\mathsf{P}}(Y|\mathbf{X})].$$

Solving the inner minimization problem assuming that we know $\check{\mathsf{P}}(Y|\mathbf{X})$, we get the result that $\hat{\mathsf{P}}(Y|\mathbf{X}) = \check{\mathsf{P}}(Y|\mathbf{X})$. Plugging it into the maximizing problem, the whole problem reduces to:

$$\max_{\hat{\mathsf{P}}(Y|\mathbf{X})\in\Delta} \mathsf{H}_{\mathsf{P}_{\text{test}}(\mathbf{x}),\hat{\mathsf{P}}(y|\mathbf{x})}(Y|\mathbf{X}) \triangleq \mathbb{E}_{\mathsf{P}_{\text{test}}(\mathbf{x})\hat{\mathsf{P}}(y|\mathbf{x})}[-\log\hat{\mathsf{P}}(Y|\mathbf{X})]$$

$$\text{such that: } \mathbb{E}_{\mathsf{P}_{\text{train}}(\mathbf{x})\hat{\mathsf{P}}(y|\mathbf{x})}[\phi(\mathbf{X},Y)] = \mathbf{c}.$$

□

From Definition 2 and Theorem 1, our robust bias-aware classifier is built on the minimax estimation method where there is a discrepancy between the distribution defining the constraints for the adversarial player and the distribution evaluating the loss. It yields a test maximum

entropy maximization problem subject to a set of training constraints. We will see how to find the solution to this constrained convex optimization problem in the following sections.

3.2.1.1 Parametric form

The solution to the constrained optimization problem (Equation 3.6) that results from our formulation has a parametric form (Theorem 2) with Lagrange multiplier parameters, θ, weighing the feature functions, $\phi(\mathbf{x}, y)$, that constrain the conditional label distribution estimate (Equation 3.6). The density ratio, $P_{\text{train}}(\mathbf{x})/P_{\text{test}}(\mathbf{x})$, moderates the distribution's prediction certainty to increase when the ratio is large and decrease when it is small.

Theorem 2. *The robust bias-aware (RBA) classifier for test distribution* $P_{test}(\mathbf{x})$ *estimated from statistics of training distribution* $P_{train}(\mathbf{x})$ *has a form:*

$$\hat{P}_\theta(y|\mathbf{x}) = \frac{e^{\frac{P_{train}(\mathbf{x})}{P_{test}(\mathbf{x})}\theta\cdot\phi(\mathbf{x},y)}}{\sum_{y'\in\mathcal{Y}} e^{\frac{P_{train}(\mathbf{x})}{P_{test}(\mathbf{x})}\theta\cdot\phi(\mathbf{x},y')}}, \tag{3.7}$$

which is parameterized by Lagrange multipliers θ*. The Lagrangian dual optimization problem selects these parameters to maximize the test distribution log likelihood:* $\max_\theta \mathbb{E}_{P_{test}(\mathbf{x})P(y|\mathbf{x})}[\log \hat{P}_\theta(Y|\mathbf{X})]$.

Proof. The constrained optimization problem can be written as:

$$\max_{\hat{P}(Y|\mathbf{X})} \mathbb{E}_{P_{\text{test}}(\mathbf{x})\hat{P}(y|\mathbf{x})}[-\log \hat{P}(Y|\mathbf{X})]$$

$$\text{such that: } \mathbb{E}_{P_{\text{train}}(\mathbf{x})\hat{P}(y|\mathbf{x})}[\phi_k(\mathbf{X}, Y)] = c_k, \forall k \in \{1, ..., K\}$$

$$\forall \mathbf{x} \in \mathcal{X}\text{: } \mathbb{E}_{\hat{P}(y|\mathbf{x})}[1|\mathbf{X}] = 1$$

$$\forall \mathbf{x} \in \mathcal{X}, y \in \mathcal{Y}\text{: } \hat{P}(y|\mathbf{x}) \geq 0.$$

Note that the final constraint is superfluous since the domain of the objective function is the positive real numbers. The Lagrangian associated with this problem is:

$$\begin{aligned}\mathcal{L}(\hat{P}(y|\mathbf{x}), \theta, \lambda) =& \mathbb{E}_{P_{\text{test}}(\mathbf{x})\hat{P}(y|\mathbf{x})}[-\log \hat{P}(Y|\mathbf{X})] + \theta \cdot \left(\mathbb{E}_{P_{\text{train}}(\mathbf{x})\hat{P}(y|\mathbf{x})}[\phi(\mathbf{X}, Y)] - \mathbf{c}\right) \\ &+ \sum_{\mathbf{x} \in \mathcal{X}} \lambda(\mathbf{x}) \left[\mathbb{E}_{\hat{P}(y|\mathbf{x})}[1|\mathbf{X}] - 1\right],\end{aligned}$$

where θ and $\lambda(\mathbf{x})$ are the Lagrangian multipliers[1]. According to strong Lagrangian duality (assuming a solution on the relative interior of the constraint set),

$$\max_{\hat{P}(y|\mathbf{x}) \in \Delta} \min_{\theta, \lambda(\mathbf{x})} \mathcal{L}(\hat{P}(y|\mathbf{x}), \theta, \lambda(\mathbf{x})) = \min_{\theta, \lambda(\mathbf{x})} \max_{\hat{P}(y|\mathbf{x}) \in \Delta} \mathcal{L}(\hat{P}(y|\mathbf{x}), \theta, \lambda(\mathbf{x})).$$

---

[1] For continuous input spaces $\mathcal{X}$, the sum over $\mathbf{x} \in \mathcal{X}$ is replaced by an integral over $\mathbf{x} \in \mathcal{X}$, but the resulting distribution form is unchanged.

So, assuming a fixed θ and $\lambda(\mathbf{x})$, the internal maximization problem can be solved first. Taking the partial derivative with respect to the conditional probability of a specific y and $\mathbf{x}$, $\hat{P}(y|\mathbf{x})$,

$$\frac{\partial}{\partial \hat{P}(y|\mathbf{x})}\mathcal{L}(\hat{P}(y|\mathbf{x}),\theta,\lambda) = -P_{\text{test}}(\mathbf{x})\log\hat{P}(y|\mathbf{x}) - P_{\text{test}}(\mathbf{x}) + P_{\text{train}}(\mathbf{x})\theta\cdot\phi(\mathbf{x},y) + \lambda(\mathbf{x}),$$

setting it equal to zero, $\frac{\partial}{\partial \hat{P}(y|\mathbf{x})}\mathcal{L}(\hat{P}(y|\mathbf{x}),\theta,\lambda(\mathbf{x})) = 0$, and solving, we obtain:

$$\log\hat{P}(y|\mathbf{x}) = -1 + \frac{P_{\text{train}}(\mathbf{x})}{P_{\text{test}}(\mathbf{x})}\theta\cdot\phi(\mathbf{x},y) + \frac{\lambda(\mathbf{x})}{P_{\text{test}}(\mathbf{x})}.$$

Therefore, we conclude:

$$\hat{P}(y|\mathbf{x}) = e^{\frac{P_{\text{train}}(\mathbf{x})}{P_{\text{test}}(\mathbf{x})}\theta\cdot\phi(\mathbf{x},y) - 1 + \frac{\lambda(\mathbf{x})}{P_{\text{test}}(\mathbf{x})}}.$$

We analytically solve the normalization Lagrange multiplier terms,

$$\lambda(\mathbf{x}) = P_{\text{test}}(\mathbf{x})\left(-\log\sum_{y'\in\mathcal{Y}} e^{\frac{P_{\text{train}}(\mathbf{x})}{P_{\text{test}}(\mathbf{x})}\theta\cdot\phi(\mathbf{x},y')} + 1\right),$$

yielding the conditional probability distribution of labels (with $Z_\theta(\mathbf{x}) \triangleq \sum_{y'\in\mathcal{Y}} e^{\frac{P_{\text{train}}(\mathbf{x})}{P_{\text{test}}(\mathbf{x})}\theta\cdot\phi(\mathbf{x},y')}$):

$$\hat{P}(y|\mathbf{x}) = \frac{e^{\frac{P_{\text{train}}(\mathbf{x})}{P_{\text{test}}(\mathbf{x})}\theta\cdot\phi(\mathbf{x},y)}}{Z_\theta(\mathbf{x})} = \frac{e^{\frac{P_{\text{train}}(\mathbf{x})}{P_{\text{test}}(\mathbf{x})}\theta\cdot\phi(\mathbf{x},y)}}{\sum_{y'\in\mathcal{Y}} e^{\frac{P_{\text{train}}(\mathbf{x})}{P_{\text{test}}(\mathbf{x})}\theta\cdot\phi(\mathbf{x},y')}}.$$

We have now derived a closed-form expression for $\hat{P}(y|\mathbf{x})$. We will show that the parameter estimation for θ is equivalent with maximum log likelihood of $\hat{P}(y|\mathbf{x})$ on test distribution. Even

though maximum log likelihood on the test distribution is impossible since we assume we do not have test data with labels, this equivalence gives a nice justification for our method because we will get the same set of parameters when we maximize the expected log likelihood on the test distribution, if possible.

Plugging the expression back into the Lagrangian and solving the outer minimization problem, we obtain

$$\begin{aligned}\mathcal{L}(\theta) =& \mathbb{E}_{P_{test}(\mathbf{x})\hat{P}(y|\mathbf{x})}\left[\log Z_\theta(\mathbf{X}) - \frac{P_{train}(\mathbf{X})}{P_{test}(\mathbf{X})}\theta\cdot\phi(\mathbf{X},Y)\right] + \theta\cdot\left(\mathbb{E}_{P_{train}(\mathbf{x})\hat{P}(y|\mathbf{x})}[\phi(\mathbf{X},Y)] - \mathbf{c}\right)\\ =& \mathbb{E}_{P_{test}(\mathbf{x})}[\log Z_\theta(\mathbf{X})] - \theta\cdot\mathbf{c}.\end{aligned}$$

Thus, the optimal Lagrangian parameters from the dual optimization problem are:

$$\theta = \operatorname*{argmin}_\theta \mathbb{E}_{P_{test}(\mathbf{x})}[\log Z_\theta(\mathbf{X})] - \theta\cdot\mathbf{c}. \tag{3.8}$$

Given $c_k = \mathbb{E}_{P_{train}(\mathbf{x})P(y|\mathbf{x})}[\phi_k(\mathbf{X}, Y)]$, the parameter estimation can be regarded as maximizing the expectation of the log-likelihood over test distribution under $\hat{P}_\theta(y|\mathbf{x}) = \frac{e^{\frac{P_{train}(\mathbf{x})}{P_{test}(\mathbf{x})}\theta\cdot\phi(x,y)}}{Z_\theta(\mathbf{x})}$:

$$
\begin{aligned}
\min_\theta \mathcal{L}(\theta) &= \min_\theta \left(\mathbb{E}_{P_{test}(\mathbf{x})}[\log Z_\theta(\mathbf{X})] - \theta \cdot \mathbf{c}\right) \\
&= \min_\theta \left(\mathbb{E}_{P_{test}(\mathbf{x})}[\log Z_\theta(\mathbf{X})] - \theta \cdot \mathbb{E}_{P_{train}(\mathbf{x})P(y|\mathbf{x})}[\phi(\mathbf{X}, Y)]\right) \\
&= \min_\theta \left(\mathbb{E}_{P_{test}(\mathbf{x})}[\log Z_\theta(\mathbf{X})] - \theta \cdot \mathbb{E}_{P_{test}(\mathbf{x})P(y|\mathbf{x})}\left[\frac{P_{train}(\mathbf{X})}{P_{test}(\mathbf{X})}\phi(\mathbf{X}, Y)\right]\right) \\
&= \max_\theta \mathbb{E}_{P_{test}(\mathbf{x})P(y|\mathbf{x})}\left[\log\left(\frac{e^{\frac{P_{train}(\mathbf{X})}{P_{test}(\mathbf{X})}\theta\cdot\phi(\mathbf{X},Y)}}{Z_\theta(\mathbf{X})}\right)\right] \\
&= \max_\theta \mathbb{E}_{P_{test}(\mathbf{x})P(y|\mathbf{x})}[\log \hat{P}_\theta(Y|\mathbf{X})].
\end{aligned}
$$

□

We show in Figure 3 a synthetic example comparing our method to logistic regression and importance weighting methods to demonstrate the key difference in the resulting conditional probability distribution of the three methods.

Logistic regression and importance weighted loss minimization (Equation 2.3) extrapolate in the face of uncertainty to make strong predictions without sufficient supporting evidence, while the RBA approach is robust to uncertainty that is inherent when learning from finite shifted data samples. In this example, prediction uncertainty is large at all tail fringes of the training distribution for the robust approach. In contrast, there is a high degree of certainty for both the logistic regression and importance weighting approaches in portions of those regions (e.g., the bottom left and top right). This is due to the strong inductive biases of those approaches being

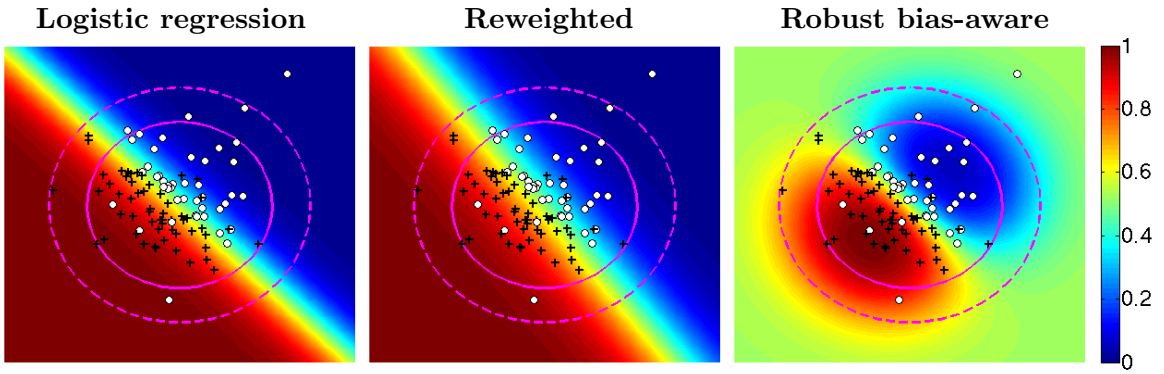


Figure 3: Probabilistic predictions from logistic regression, importance weighting logloss minimization, and robust bias-aware models given labeled data ('+' and 'o' classes) sampled from the training distribution (solid oval indicating Gaussian covariance) and a testing distribution (dashed oval Gaussian covariance) for first-order moment statistics (i.e., $\phi(\mathbf{x}, y) = [y \; yx_1 \; yx_2]^T$).

applied to portions of the input space where there is sparse evidence to support them. The conceptual argument against this strong inductive generalization is that the labels of datapoints in these tail fringe regions could take either value and negligibly affect the training distribution statistics. Given this ambiguity, the robust approach suggests much more agnostic predictions.

Moreover, unlike the importance weighting approach, our approach does not require that test distribution support implies training distribution support (i.e., $P_{test}(\mathbf{x}) > 0 \implies P_{train}(\mathbf{x}) > 0$ is not required). Where testing support vanishes (i.e., $P_{test}(\mathbf{x}) \to 0$), the classifier's prediction is extremely certain, and where training support vanishes (i.e., $P_{train}(\mathbf{x}) = 0$), the classifier's prediction is a uniform distribution.

3.2.1.2 Regularization and Parameter Estimation

In practice, the characteristics of the training distribution, Ξ, are not precisely known. Instead, empirical estimates for moment-matching constraints, $\tilde{\mathbf{c}} \triangleq \mathbb{E}_{\tilde{\mathsf{P}}_{\text{train}}(\mathbf{x})\tilde{\mathsf{P}}(\mathsf{y}|\mathbf{x})}[\phi(\mathbf{X}, \mathsf{Y})]$, are available, but are prone to sampling error. When the constraints of Equation 3.6 are relaxed using various convex norms, $||\tilde{\mathbf{c}} - \mathbb{E}_{\tilde{\mathsf{P}}_{\text{train}}(\mathbf{x})\hat{\mathsf{P}}(\mathsf{y}|\mathbf{x})}[\phi(\mathbf{X}, \mathsf{Y})]|| \leq \epsilon$, the RBA classifier is obtained by ℓ_1- or ℓ_2-regularized maximum conditional likelihood estimation (Theorem 2) of the dual optimization problem (Dudík and Schapire, 2006; Altun and Smola, 2006),

$$\theta = \underset{\theta}{\operatorname{argmax}} \; \mathbb{E}_{\mathsf{P}_{\text{test}}(\mathbf{x})\mathsf{P}(\mathsf{y}|\mathbf{x})} \left[\log \hat{\mathsf{P}}_\theta(\mathsf{Y}|\mathbf{X})\right] - \epsilon \, ||\theta|| \,. \tag{3.9}$$

The regularization parameters in this approach can be chosen using straight-forward bounds on finite sampling error (Altun and Smola, 2006). In contrast, the importance weighting approach to learning under sample selection bias (Shimodaira, 2000; Zadrozny, 2004) also makes use of regularization (Sugiyama et al., 2008), but appropriate regularization parameters for it must be haphazardly chosen based on how well the training samples represent the testing data.

Maximizing this regularized test conditional likelihood (Equation 3.9) appears difficult because test data from $\mathsf{P}_{\text{trg}}(\mathbf{x})\mathsf{P}(\mathsf{y}|\mathbf{x})$ is unavailable. We avoid the importance weighting approach (Equation 2.3) (Shimodaira, 2000; Zadrozny, 2004), due to its inaccuracies when facing distributions with large differences in bias given finite samples. Instead, we use the gradient

of the regularized test conditional likelihood and only rely on training samples adequately approximating the training distribution statistics:

$$\tilde{\mathbf{c}} - \mathbb{E}_{\tilde{P}_{\text{train}}(\mathbf{x})\hat{P}(y|\mathbf{x})}[\phi(\mathbf{X}, Y)]. \tag{3.10}$$

The derivation of the gradient is as below. Taking the derivative with respect to θ, we obtain

$$\begin{aligned}
\frac{\partial}{\partial\theta_k}\mathbb{E}_{P_{\text{test}}(\mathbf{x})P(y|\mathbf{x})}[\log \hat{P}_\theta(Y|\mathbf{X})] &= \frac{\partial}{\partial\theta_k}\left(\theta\cdot\mathbf{c} - \mathbb{E}_{P_{\text{test}}(\mathbf{x})}[\log Z_\theta(\mathbf{X})]\right) \\
&= c_k - \mathbb{E}_{P_{\text{test}}(\mathbf{x})}\left[\sum_{y\in\mathcal{Y}} \frac{e^{\frac{P_{\text{train}}(\mathbf{X})}{P_{\text{test}}(\mathbf{X})}\theta\cdot\phi(\mathbf{X},y)}}{Z_\theta(\mathbf{X})}\frac{P_{\text{train}}(\mathbf{X})}{P_{\text{test}}(\mathbf{X})}\phi_k(\mathbf{X}, y)\right] \\
&= c_k - \mathbb{E}_{P_{\text{test}}(\mathbf{x})\hat{P}(y|\mathbf{x})}\left[\frac{P_{\text{train}}(\mathbf{X})}{P_{\text{test}}(\mathbf{X})}\phi_k(\mathbf{X}, Y)\right] \\
&= c_k - \mathbb{E}_{P_{\text{train}}(\mathbf{x})\hat{P}(y|\mathbf{x})}\left[\phi_k(\mathbf{X}, Y)\right].
\end{aligned}$$

If we use empirical estimates for both terms, we get the gradient in Equation 3.33.

Algorithm 2 is a batch gradient algorithm for parameter estimation under our model. It does not require objective function calculations and converges to a global optimum due to convexity (Boyd and Vandenberghe, 2004).

It is obvious that the better our features can characterize the training distribution, the better our estimation of the parameter will be. Figure 4 shows the comparison on two overlapping Gaussian distribution. Meanwhile, we can observe a difference when using first order features and second order features. When using second order features, the certainty in the overlapping

Algorithm 2 Batch gradient for robust bias-aware classifier learning.

Input: Dataset $\{(\mathbf{x}_i, y_i)\}$, training density $P_{\text{train}}(\mathbf{x})$, testing density $P_{\text{test}}(\mathbf{x})$, feature function $\phi(\mathbf{x}, y)$, measured statistics $\tilde{\mathbf{c}}$, (decaying) learning rate $\{\gamma_t\}$, regularizer ϵ, convergence threshold τ

Output: Model parameters θ

$\theta \leftarrow \mathbf{0}$

repeat

$\psi(\mathbf{x}_i, y) \leftarrow \frac{P_{\text{train}}(\mathbf{x})}{P_{\text{test}}(\mathbf{x})} \theta \cdot \phi(\mathbf{x}_i, y)$ for all: dataset examples i, labels y

$\hat{P}(Y_i = y|\mathbf{x}_i) \leftarrow \frac{e^{\psi(\mathbf{x}_i, y)}}{\sum_{y'} e^{\psi(\mathbf{x}_i, y')}}$ for all: dataset examples i, labels y

$\nabla\mathcal{L} \leftarrow \tilde{\mathbf{c}} - \frac{1}{N} \sum_{i=1}^{N} \sum_{y \in \mathcal{Y}} \hat{P}(Y_i = y|\mathbf{x}_i)\ \phi(\mathbf{x}_i, y)$

$\theta \leftarrow \theta + \gamma_t(\nabla\mathcal{L} + \epsilon\nabla_\theta\|\theta\|)$

until $\|\epsilon\nabla_\theta\|\theta\| + \nabla\mathcal{L}\| \leq \tau$

return θ

area, where data have relatively high density in both training and testing distribution, is much higher. This reflects that the conditional probability is better estimated when better features that constrain the adversarial player are provided by training data.

3.2.2 With Some Feature Generalization

In many settings, expert knowledge may be available to construct the constraint set Ξ instead of, or in addition to, statistics $\tilde{\mathbf{c}} \triangleq \mathbb{E}_{\tilde{P}_{\text{train}}(\mathbf{x})\tilde{P}(y|\mathbf{x})}[\phi(\mathbf{X}, Y)]$ estimated from training data. Expert-provided training distributions, feature functions, and constraint statistic values, respectfully

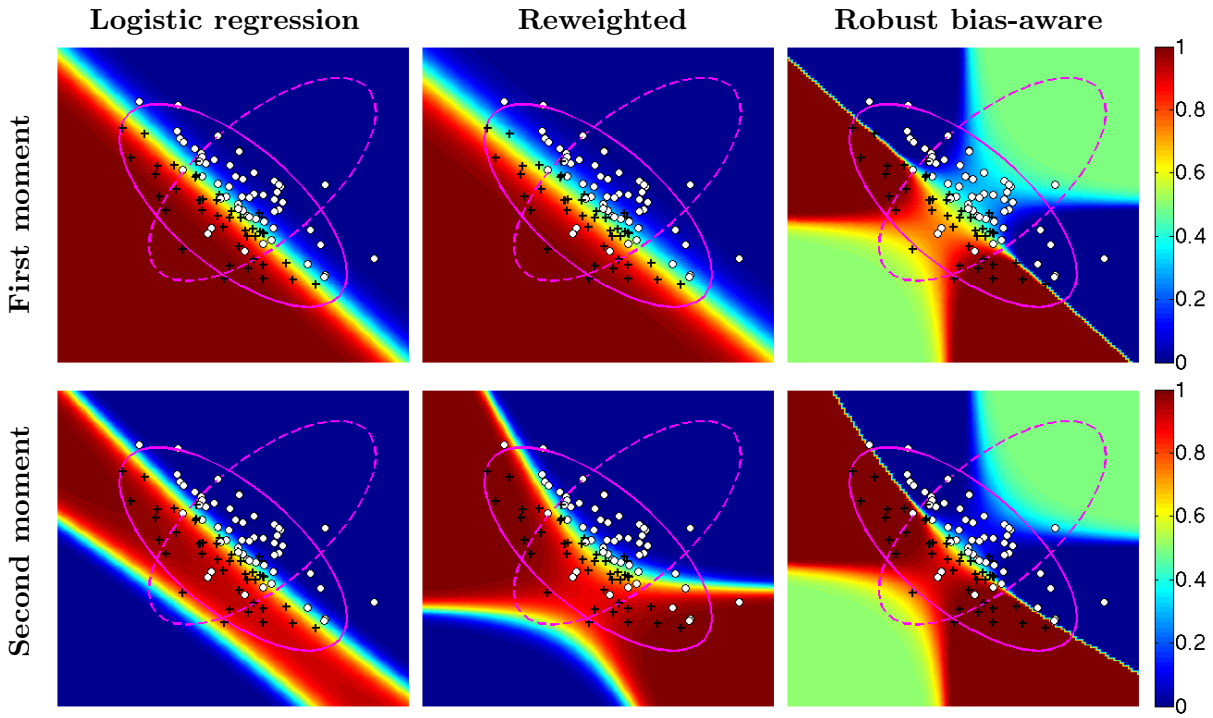


Figure 4: The prediction setting of partially overlapping training and testing densities for first-order (top) and second-order (bottom) mixed-moments statistics (i.e., $\phi(\mathbf{x}, y) = [y \; yx_1 \; yx_2 \; yx_1^2 \; yx_1x_2 \; yx_2^2]^\mathsf{T}$). Logistic regression and the importance weighting approach make high-certainty predictions in portions of the input space that have high testing density. These predictions are made despite the sparseness of sampled training data in those regions (e.g., the upper-right portion of the testing distribution). In contrast, the robust approach "pushes" its more certain predictions to areas where the testing density is less.

denoted $\mathrm{P}'_{\text{train}}(\mathbf{x})$, $\phi'(\mathbf{x}, y)$, and $\mathbf{c}'$, can be specified to express a range of assumptions about the conditional label distribution and how it generalizes.

Weaker expert knowledge can also be incorporated. Figure 5 shows various assumptions of how widely sample reweighted statistics are representative across the input space. As the

generalization assumptions are made to align more closely with the testing distribution (Figure 5), the regions of uncertainty shrink substantially.

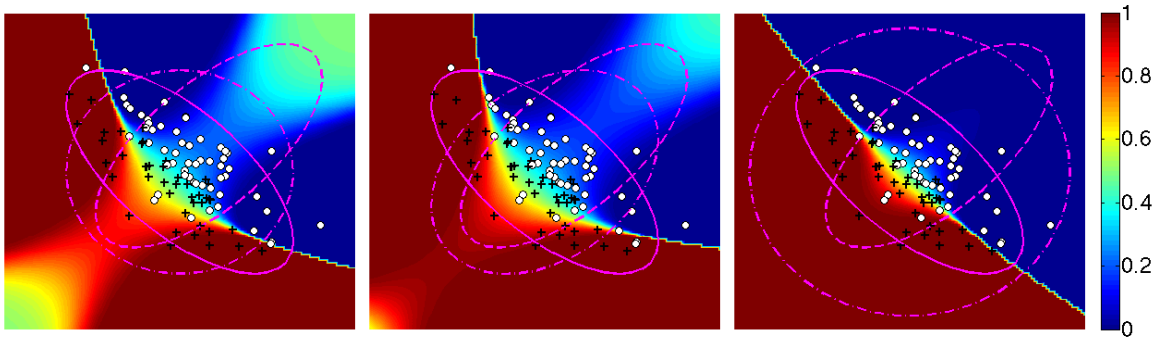


Figure 5: The robust estimation setting of Figure 4 (bottom, right) with assumed Gaussian feature distribution generalization (dashed-dotted oval) incorporated into the density ratio. Three increasingly broad generalization distributions lead to reduced testing prediction uncertainty.

This property reflects the flexibility of our method. Even though we assume the worst case for the unknown label distribution that is to be estimated, we can still control the extent to which we think our features should generalize in the testing distribution by applying a different $\mathsf{P}_{\text{train}}(\mathbf{x})$. More aggressively, if we are equipped with side information that shows how training features generalize, more accurate $\mathsf{P}_{\text{gen}}(\mathbf{x})$ could be incorporated into our formulation.

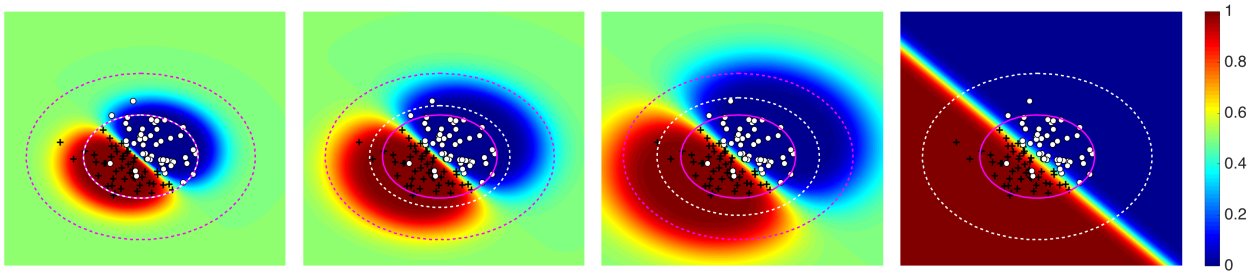
Feature generalization makes it possible to utilize information shared by both training and testing distributions and is essential to improve performance in predicting testing data. In order to illustrate the effect of the generalization distribution, in Figure 6 we consider a synthetic

example with data sampled from two Gaussian distributions in 2-dimensional input space, with the training distribution totally contained in the testing distribution (two magenta ellipses). We compare the performance of logloss-based classifier on them. 100 data points are sampled and only the training datapoints are shown in Figure 6, with 5% of noise in both training and testing data. We assume there exists a generalization distribution that training features generalize to (white ellipse). After training with larger and larger generalization distribution, predictive performance is evaluated on testing data and logloss is shown under the figures.

We can see from the figures that the generalization from training features gets broader and broader with larger and larger generalization distributions. In the first case, $\mathsf{P}_{\text{gen}}(\mathbf{x})$ equals $\mathsf{P}_{\text{train}}(\mathbf{x})$, the method is equivalent with robust covariate shift method and the prediction is limited only to the space around where there is enough training support. In the second and third cases, the certain portion in the whole space increases with logloss on testing data getting better. Finally, in the last case, $\mathsf{P}_{\text{gen}}(\mathbf{x})$ equals to $\mathsf{P}_{\text{test}}(\mathbf{x})$, the method is equivalent to importance weighting. We can see the prediction is quite certain across the whole space in this setting. The logloss, however, gets worse in this case due to the noise in the data. So the takeaway from this example is that it is important to get a balance between feature generalization and robustness.

3.2.2.1 Effect of Density (Ratio) Estimation

In practice, the feature generalization is partially determined by the densities like $\mathsf{P}_{\text{train}}(\mathbf{x})$, which need to be estimated beforehand. Especially, if we want to apply robust predictor with some generalization, it is crucial to choose a suitable distribution $\mathsf{P}_{\text{gen}}(\mathbf{x})$ other than $\mathsf{P}_{\text{train}}(\mathbf{x})$ and estimate the density accurately. The difficulties in the nature of density (ratio)



logloss = 0.56 logoss = 0.48 logloss = 0.46 logloss = 0.75

Figure 6: Comparison of incorporating different generalization distribution (white ellipses) in robust covariate shift classifier. Logloss evaluated on testing data points (not shown) is shown below each figure. Colormap represents the predicted probability of $\mathsf{P}(\text{`+ '}|\mathbf{x})$.

estimation could lead to inaccurate predictive result from our robust predictor. Similar with the important weighting method, we may benefit from different techniques that improve the quality of density (ratio) estimation. This includes methods based on trimmed Maximum Likelihood Estimation (Hadi and Luceño, 1997; Liu et al., 2017), thresholded density ratio (Smola et al., 2009) and a biased version of density ratio (Yamada et al., 2011). Moreover, the robust method is also less sensitive to the datapoints that has dominating weights in importance weighting method. Because the inverse of the ratio would be small and the resulting predictive certainty from our robust predictor is then smaller.

3.2.3 With Full Feature Generalization: Reduction to IW

Theorem 3 establishes that for empirically-based constraints provided by the expert,

$$\mathbb{E}_{\mathsf{P}_{\text{test}}(\mathbf{x})\hat{\mathsf{P}}(y|\mathbf{x})}[\phi(\mathbf{X},Y)] = \tilde{\mathbf{c}}' \triangleq \mathbb{E}_{\tilde{\mathsf{P}}_{\text{train}}(\mathbf{x})\tilde{\mathsf{P}}(y|\mathbf{x})}[(\mathsf{P}_{\text{test}}(\mathbf{X})/\mathsf{P}_{\text{train}}(\mathbf{X}))\phi(\mathbf{X},Y)].$$

corresponding to strong train-to-test feature generalization assumptions, $\mathsf{P}'_{\text{train}}(\mathbf{x}) \triangleq \mathsf{P}_{\text{test}}(\mathbf{x})$, reweighted logloss minimization is a special case of our robust bias-aware approach.

Theorem 3. *When direct feature generalization of reweighting training samples to the target distribution is assumed, the constraints become*

$$\mathbb{E}_{\mathsf{P}_{test}(\mathbf{x})\hat{\mathsf{P}}(\mathsf{y}|\mathbf{x})}[\phi(\mathbf{X},\mathsf{Y})] = \tilde{\mathbf{c}}' \triangleq \mathbb{E}_{\tilde{\mathsf{P}}_{train}(\mathbf{x})\tilde{\mathsf{P}}(\mathsf{y}|\mathbf{x})}\left[\frac{\mathsf{P}_{test}(\mathbf{X})}{\mathsf{P}_{train}(\mathbf{X})}\phi(\mathbf{X},\mathsf{Y})\right] \tag{3.11}$$

and the robust classifier minimizes importance weighting logloss (Equation 2.3).

This equivalence suggests that if there is expert knowledge that reweighted training statistics are representative of the target distribution, then these strong generalization assumptions should be included as constraints in the RBA predictor and results in the importance weighting approach[1].

Proof. Assuming $\mathsf{P}'_{\text{train}}(\mathbf{x}) = \mathsf{P}_{\text{test}}(\mathbf{x})$ for feature expectations in the constraints of Equation 3.6, and following the same approach as the proof of Theorem 2, we obtain the form of the RBA classifier in this case, which is the same as logistic regression, $\hat{\mathsf{P}}(\mathsf{y}|\mathbf{x}) = \frac{e^{\theta\cdot\phi(\mathbf{x},\mathsf{y})}}{\mathsf{Z}'(\mathbf{x})}$, with $\mathsf{Z}'_\theta(\mathbf{x}) = \sum_{\mathsf{y}'\in\mathcal{Y}} e^{\theta\cdot\phi(\mathbf{x},\mathsf{y}')}$.

---

[1]Relaxed constraints $\|\tilde{\mathbf{c}}' - \mathbb{E}_{\tilde{\mathsf{P}}_{\text{train}}(\mathbf{x})\hat{\mathsf{P}}(\mathsf{y}|\mathbf{x})}[\phi(\mathbf{X},\mathsf{Y})]\| \leq \epsilon$, are employed in practice and parameters are obtained by maximizing the regularized conditional likelihood.

Plugging this parametric form into the Lagrangian and solving the minimization problem, applying that $\tilde{c}'_k = \mathbb{E}_{\tilde{P}_{train}(x)\tilde{P}(y|x)}\left[\frac{P_{test}(X)}{P_{train}(X)}\phi_k(X,Y)\right]$, the problem becomes minimizing the following:

$$\begin{aligned}\mathcal{L}(\theta) =& \mathbb{E}_{P_{test}(x)\hat{P}(y|x)}\left[Z'_\theta(X) - \theta\cdot\phi(X,Y)\right] \\ &+ \theta\cdot\left(\mathbb{E}_{P_{test}(x)\hat{P}(y|x)}[\phi(X,Y)] - \mathbb{E}_{\tilde{P}_{train}(x)\tilde{P}(y|x)}\left[\frac{P_{test}(X)}{P_{train}(X)}\phi(X,Y)\right]\right) \\ =& \mathbb{E}_{P_{test}(x)}[\log Z'_\theta(X)] - \theta\cdot\tilde{c}'. \end{aligned} \tag{3.12}$$

The gradient of Equation 3.12 is $\mathbb{E}_{P_{test}(x)\hat{P}(y|x)}[\phi(X,Y)] - \tilde{c}' = \mathbb{E}_{P_{train}(x)\hat{P}(y|x)}\left[\frac{P_{test}(X)}{P_{train}(X)}\phi(X,Y)\right] - \tilde{c}' \approx \mathbb{E}_{\tilde{P}_{train}(x)\hat{P}(y|x)}\left[\frac{P_{test}(X)}{P_{train}(X)}\phi(X,Y)\right] - \tilde{c}'$. Constraint slack and dual regularization can be applied to allow for the noise from finite sample approximation, as described in §3.2.1.2. We omit these in the interest of clarity and brevity. The importance weighting logloss minimization problem assumes $\hat{P}_\theta(y|x) = \frac{e^{\theta\cdot\phi(x,y)}}{\sum_{y'\in\mathcal{Y}} e^{\theta\cdot\phi(x,y')}}$ and minimizes the following:

$$\begin{aligned}\ell(\theta) &= \mathbb{E}_{P_{test}(x)P(y|x)}[-\log\hat{P}_\theta(Y|X)] \\ &\approx \mathbb{E}_{\tilde{P}_{train}(x)\tilde{P}(y|x)}\left[-\frac{P_{test}(X)}{P_{train}(X)}\log\hat{P}_\theta(Y|X)\right] \\ &= \mathbb{E}_{\tilde{P}_{train}(x)}\left[\frac{P_{test}(X)}{P_{train}(X)}\log Z'_\theta(X)\right] - \mathbb{E}_{\tilde{P}_{train}(x)\tilde{P}(y|x)}\left[\frac{P_{test}(X)}{P_{train}(X)}\theta\cdot\phi(X,Y)\right] \\ &= \mathbb{E}_{\tilde{P}_{train}(x)}\left[\frac{P_{test}(X)}{P_{train}(X)}\log Z'_\theta(X)\right] - \theta\cdot\tilde{c}'. \end{aligned} \tag{3.13}$$

The approximation in the second step is based on the importance weighting assumption (Equation 2.3). The gradient of Equation 3.13 is $\mathbb{E}_{\tilde{P}_{train}(x)\hat{P}(y|x)}\left[\frac{P_{test}(X)}{P_{train}(X)}\phi(X,Y)\right] - \tilde{c}'$, which is

the same with the gradient of Equation 3.12. Therefore, the two approaches are equivalent in this special case. □

So in summary, there are two extremes in the logarithmic loss case.

- Reduction to Robust Bias-Aware Prediction: When $\mathsf{P}_{\text{gen}}(\mathbf{x}) = \mathsf{P}_{\text{train}}(\mathbf{x})$, this gives us a model that is representationally equivalent to the robust bias aware prediction method (Liu and Ziebart, 2014). The solution has a parametric form with the density ratio appearing as: $\mathsf{P}(\mathsf{y}|\mathbf{x}) \propto e^{\frac{\mathsf{P}_{\text{train}}(\mathbf{x})}{\mathsf{P}_{\text{test}}(\mathbf{x})}\theta\cdot\phi(\mathbf{x},\mathsf{y})}$ and moderates the uncertainty of the predictor to be larger for inputs that are relatively less likely in the training data. Thus, the density ratio $\frac{\mathsf{P}_{\text{train}}(\mathbf{x})}{\mathsf{P}_{\text{test}}(\mathbf{x})}$ in this parametric form controls the uncertainty of predictions. This method is the most conservative model this framework generates, which limits the adversarial player to match training sample statistics strictly.

- Reduction to Importance Weighting: If all the features are assumed to generalize fully to the testing distribution, i.e., $\mathsf{P}_{\text{gen}}(\mathbf{x}) = \mathsf{P}_{\text{test}}(\mathbf{x})$, the generalized robust covariate shift classifier is equivalent to the importance weighting method (Equation 2.3). It produces the most aggressive model when the $\check{\mathsf{P}}$ could match reweighted features using $\frac{\mathsf{P}_{\text{test}}(\mathbf{x})}{\mathsf{P}_{\text{train}}(\mathbf{x})}$. Note that this has been proposed and proven before (Liu and Ziebart, 2014). The good generalization distribution should help avoid errors brought by over optimistic estimates but also achieve better performance than no generalization.

3.3 Zero-One Loss Case

Letting $\text{Loss}(\check{P}_{\mathbf{X}}, \hat{P}_{\mathbf{X}}) = \hat{P}^{\mathsf{T}} C \check{P}$—a bilinear and therefore concave-convex function of $\check{P}$ and $\hat{P}$—allows many classification losses to be represented in the cost matrix C. We can reformulate the inner minimax game as $\min_{\hat{P}} \max_{\check{P}} \mathbb{E}_{\mathbf{X}}[\hat{P}^{\mathsf{T}} C' \check{P}]$, where $C' = C + \frac{P_{\text{gen}}(\mathbf{X})}{P_{\text{test}}(\mathbf{X})} \theta \phi(\mathbf{X}, \check{Y})$. The inner minimax game, which is a two player zero sum game, can be solved by linear programming. Another way to find the equilibrium of the inner minimax game for the special case of 0-1 loss is by seeking an analytical form of the game value (Fathony et al., 2016), which brings more computational efficiency. For the outer minimization, we take the subgradient with respect to θ, which we approximate by reweighting training samples to the generalization distribution,

$$\mathbb{E}_{\mathbf{X} \sim P_{\text{gen}}, \check{Y} | \mathbf{X} \sim \check{P}} \left[\phi(\mathbf{X}, \check{Y}) \right] - \tilde{\phi} + 2\epsilon\theta, \tag{3.14}$$

and perform subgradient descent. Even though the objective function is defined under test distribution, we are able to approximate the subgradient using training data by reweighting training samples to the generalization distribution:

$$\mathbb{E}_{P_{\text{gen}}(\mathbf{X})} \left[\phi(\mathbf{X}, \check{Y}) \right] - \sum \tilde{\phi} + 2\epsilon\theta \tag{3.15}$$

We show two illustrative examples in Figure 7, where training distribution (solid line) and testing distribution(dashed line) is overlapping in different ways. The prediction color map shows a similar uncertain prediction with logloss-based classifier where there is not enough training data support, like the top right corner in the first figure. Moreover, the 0-1 loss provides more certain

prediction in the overlapped region while logloss-based classifier's prediction changes gradually in certainty from the most supported region to the least.

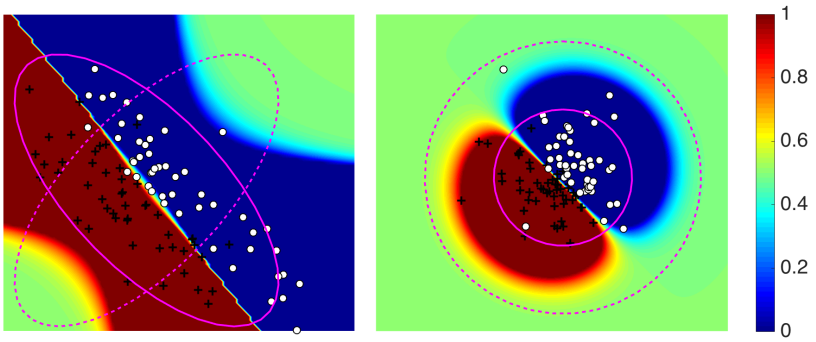


Figure 7: Prediction colormap with robust classifier using 0-1 loss when $\mathsf{P}_{\text{gen}}(\mathbf{x}) = \mathsf{P}_{\text{train}}(\mathbf{x})$. The colormap shows the $\mathsf{P}(\text{`+ '}|\mathbf{x})$. Training data with 5% noise is also shown.

3.4 Applying Kernel Methods

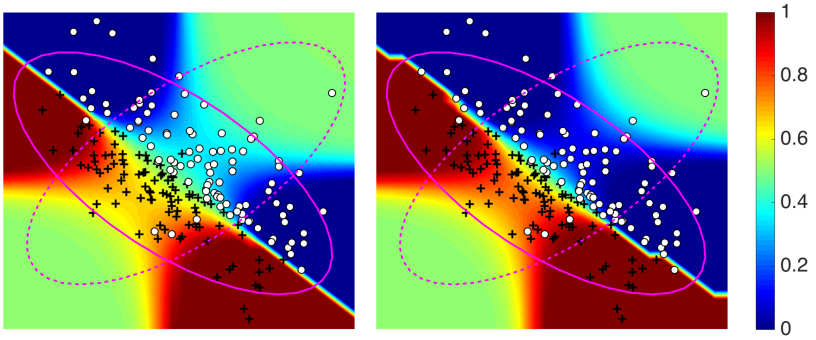
Applying kernel methods is not straightforward in our robust prediction framework because we directly minimize the expected test loss, while the standard representer theorem starts from a regularized empirical loss. Here we use the RBA predictor as an example to show the way to apply kernel methods and its benefit.

Our framework is based on a minimax robust estimation formulation (Grünwald and Dawid, 2004) that assumes the worst case conditional label distribution and requires only training feature expectation matching as constraints. The approach provides conservative test predictions

when the testing distribution does not have sufficient statistical support from the training data. This statistical support is defined by the choice of training statistics or features. The classifier tries to make the prediction certainty under the testing distribution as small as possible, but feature matching constraints prevent it from doing so fully. As a result, less restrictive feature constraints produce less certain predictions on testing data from the resulting classifier. As shown in Figure 8(a), with limited features, the classifier may allocate most of the certainty under portions of the training distribution (solid line) where the testing distribution (dashed line) density is small to satisfy the training feature expectation matching constraints, leaving too much uncertainty in portions of the testing distribution. On the other hand, when there are more restrictive features constraining the conditional label distribution, the classifier produces a better model of the data and gives more informative predictions with less test entropy and logloss, as in Figure 8(b). This relation inspires our contribution: leveraging kernel methods to provide higher dimensional features to the RBA classifier without introducing a proportionate computational burden.

According to the representer theorem (Kimeldorf and Wahba, 1971), the minimizer of regularized empirical loss in reproducing kernel Hilbert space can be represented by a linear combination of kernel products evaluated on training data. Model parameters are then obtained by estimating the coefficients of this linear combination. However, in the robust bias-aware classification framework, the objective function of the dual problem is the regularized expected logarithmic loss under the testing data distribution. It cannot be computed explicitly using data because labeled test samples are unavailable. Meanwhile, the distribution discrepancy

(a) First moment features (b) Third moment features



logloss: 0.74
entropy: 0.93

logloss: 0.53
entropy: 0.73

Figure 8: Performance comparison with the robust bias aware classifier using first-order features (a) and using first-order through third-order features (b). Labeled training data samples ('o' and '+' classes), training (solid line) and testing (dashed line) distribution that data are drawn from are shown. Colormap represents the predicted probability $\mathsf{P}(\mathsf{y} =$'+'$|\mathbf{x})$. The intersection of training distribution and testing distribution is better predicted with third-order features and is much more uncertain when only using first moment features. The corresponding test logloss and entropy are shown under the figures.

when evaluating the risk function and sampling training data prevents us from applying the representer theorem directly.

A quantitative form of the representer theorem has been proposed that holds for the continuous case (De Vito et al., 2004) in which a minimizer over a distribution—rather than discrete samples—is sought. The minimizer of regularized expected risk is represented as the expectation under the same probability distribution instead of a linear combination of the training data. We utilize this result to extend the representer theorem for RBA prediction in the covariate shift setting. We show that the minimizer of the regularized expected test risk can

be represented as a reweighted kernel expectation under the training distribution. This enables us to apply kernel methods to the robust bias aware classifier.

3.4.1 Kernel Methods in Other Framework under Covariate Shift

Kernel methods have been mainly employed for estimating the density ratio in importance weighting methods in the covariate shift setting, for example, as kernel mean matching (Huang et al., 2006; Yu and Szepesvári, 2012). The core idea is that the kernel mean in a reproducing kernel Hilbert space (RKHS) of the training data should be close to that of the reweighted test data and the optimal density ratio is obtained by minimizing this difference. Kernel methods have also served as a bridge between the training and the testing domains in broader transfer learning or domain adaptation problems. In these approaches, kernel methods are used to project training data and testing data into a latent space where the distance between the two distributions is small or can be minimized (Pan and Yang, 2010).

These existing applications of kernel methods for covariate shift are orthogonal to our approach because they are based on empirical risk minimization formulations with the assumption that training data could somehow be transformed to match testing distributions. This differs substantially from our robust approach.

3.4.2 Extended Representer Theorem for RBA

We know that minimizing the test logarithmic loss,

$$\theta = \underset{\theta}{\operatorname{argmin}}\, \mathbb{E}_{P_{test}(\mathbf{x})P(y|\mathbf{x})}[-\log P_\theta(Y|\mathbf{X})] + \lambda||\theta||_2^2, \tag{3.16}$$

provides parameter vector estimates θ. This can be accomplished by approximating the gradient using training samples rather than approximating the objective function (Equation 3.16). Kernel methods are motivated for the RBA approach to provide a more sufficiently restrictive set of constraints that forces generalization from training data samples to testing data. However, the inability to directly apply empirical risk minimization in the RBA approach complicates their incorporation since kernel method applications often use empirical risk minimization as a starting point.

We extend the representer theorem in the RBA approach by first investigating the minimizer of the regularized expected test loss. Theorem 4 shows that the minimizer of a regularized expected test loss can instead be represented by a reweighted expectation under the training distribution.

Theorem 4. *Let $\mathcal{X}$ be the input space and $\mathcal{Y}$ be the output space, K is a positive definite real valued kernel on $\mathcal{X} \times \mathcal{X}$ with corresponding reproducing kernel Hilbert space H_k, if the training samples $(x_1^s, y_1^s), \ldots, (x_n^s, y_n^s) \in \mathcal{X} \times \mathcal{Y}$ are drawn from a training distribution $P_{train}(x)P(y|x)$ and the testing samples $(x_1^t, y_1^t), \ldots, (x_m^t, y_m^t) \in \mathcal{X} \times \mathcal{Y}$ are drawn from a testing distribution $P_{test}(x)P(y|x)$, any minimizer f^* of Equation 3.16 in H_k, defining the conditional label distribution,*

$$\hat{P}(y|x) = e^{f^*(x,y)} \Big/ \sum_{y'} e^{f^*(x,y')}, \tag{3.17}$$

admits a representation with a form such that each $f^*(x_i^t, y_i^t) =$

$$\frac{P_{train}(x_i^t)}{P_{test}(x_i^t)} \mathbb{E}_{P_{train}(x)P(y|x)} \left[\alpha(X, Y) K((x_i^t, y_i^t), (X, Y)) \right], \tag{3.18}$$

where $\alpha(x_i, y_i) \in \mathbb{R}$, *for* $1 \leq i \leq m$, *with*

$$\theta = \mathbb{E}_{P_{train}(x)P(y|x)} \left[\alpha(X, Y) \Phi(X, Y) \right]. \tag{3.19}$$

This theorem indicates that it is possible to represent the minimizer of the expected test objective function using reweighted training samples. Note that it is essentially different from the kernel version of importance weighting method where the objective is first approximated with training samples and then the kernel method is applied.

Proof. Defining $\Phi'(x, y) \triangleq \frac{P_{\text{train}}(x)}{P_{\text{test}}(x)} \Phi(x, y)$, the robust bias-aware label distribution can be rewritten as $\hat{P}(y|x) = e^{\theta \cdot \Phi'(x,y)} / Z(x)$, with $Z(x) = \sum_{y' \in \mathcal{Y}} e^{\theta \cdot \Phi'(x,y')}$. The objective function is then:

$$\begin{aligned} &\mathbb{E}_{P_{\text{test}}(x)P(y|x)}[-\log P_\theta(Y|X)] + \lambda ||\theta||_2^2 \\ &= \mathbb{E}_{P_{\text{test}}(x)P(y|x)}[-f(X, Y) + \log Z(X)] + \lambda ||\theta||_2^2, \end{aligned} \tag{3.20}$$

where $f(x, y) = \langle \Phi'(x, y), \theta \rangle$ is the function that we aim to find that minimizes this regularized expected loss. Let K' be a positive definite real valued kernel on H'_k, according to the generalized

representer theorem (De Vito et al., 2004) in this expected risk case, the minimizer f^* takes the form:

$$f^*(x_i^t, y_i^t) = \mathbb{E}_{P_{test}(x)P(y|x)}[\alpha(X, Y)K'((x_i^t, y_i^t), (X, Y))],$$

where $K'((x_i^t, y_i^t), (x, y)) = \langle \Phi'(x_i^t, y_i^t), \Phi'(x, y) \rangle$. Since the test label is not available in training, the minimizer cannot be represented directly by testing data. Instead it can be represented by training data, which, for each $1 \leq i \leq m$, is:

$$\begin{aligned} f^*(x_i^t, y_i^t) &= \mathbb{E}_{P_{test}(x)P(y|x)}\left[\frac{P_{train}(x_i^t)}{P_{test}(x_i^t)}\frac{P_{train}(X)}{P_{test}(X)}\alpha(X, Y)K((x_i^t, y_i^t), (X, Y))\right] \\ &= \frac{P_{train}(x_i^t)}{P_{test}(x_i^t)}\mathbb{E}_{P_{train}(x)P(y|x)}\left[\alpha(X, Y)K((x_i^t, y_i^t), (X, Y))\right]. \end{aligned}$$

Given $f(x, y) = \langle \Phi'(x, y), \theta \rangle = \frac{P_{train}(x)}{P_{test}(y)}\theta \cdot \Phi(x, y)$, we obtain $\theta = \mathbb{E}_{P_{train}(x)P(y|x)}[\alpha(X, Y)\Phi(X, Y)]$.

□

3.4.3 Kernel RBA Parameter Estimation

As in the non-kernelized RBA model, the objective function (Equation 3.16) is defined in terms of the labeled testing distribution data, which is unavailable. However, the parametric model's form (Equation 3.17) enables this difficulty to be bypassed when employing the kernelized minimizer (Equation 3.18). In order to estimate the parameters $\{\alpha(x, y)\}$, we derive the gradient of the kernel RBA predictor.

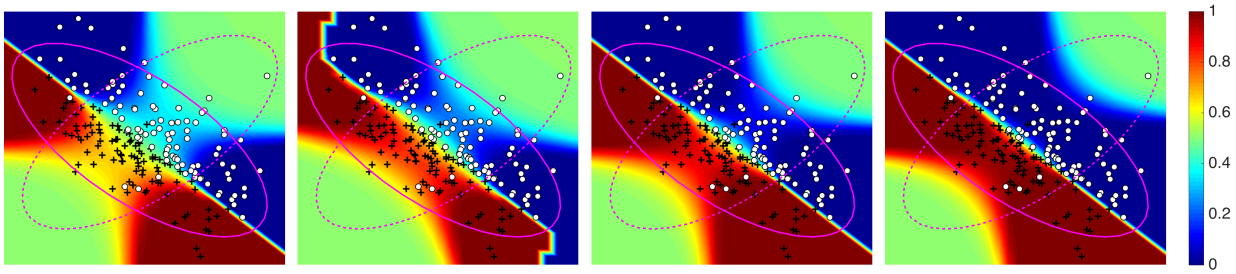
Corollary 1 (of Theorem 4)**.** *The gradient (with respect to kernelized parameters α) of the regularized expected loss is obtained by approximating kernel evaluations under the training distribution with training sample kernel evaluations.*

Corollary 1 indicates that the computation of the gradient only requires training samples. This requires an approximation of the training distribution's expected kernel evaluations with the empirical evaluations of the sample mean, whose error can be controlled using standard finite sample bounds, similar to kernel logistic regression.

Proof. Plugging Equation 3.19 into Equation 3.16, we obtain the form of objective function represented by kernels and take derivatives with respect to $\alpha(\mathbf{x}', y')$:

$$
\begin{aligned}
&\frac{\partial}{\partial \alpha(\mathbf{x}', y')} \left(\mathbb{E}_{P_{\text{test}}(\mathbf{x}) P(y|\mathbf{x})}[-\log P_\theta(Y|\mathbf{X})] + \lambda ||\theta||_2^2 \right) \\
&= -\mathbb{E}_{P_{\text{train}}(\mathbf{x}) P(y|\mathbf{x})}[K((\mathbf{x}', y'), (\mathbf{X}, Y))] + \mathbb{E}_{P_{\text{train}}(\mathbf{x}) \hat{P}(y|\mathbf{x})}[K((\mathbf{x}', y'), (\mathbf{X}, Y))] \\
&+ \lambda \mathbb{E}_{P_{\text{train}}(\mathbf{x}'') P(y''|\mathbf{x}'')}[\alpha(\mathbf{X}'', Y'') K((\mathbf{x}', y'), (\mathbf{X}'', Y''))] \\
&\approx -\mathbb{E}_{\tilde{P}_{\text{train}}(\mathbf{x}) \tilde{P}(y|\mathbf{x})} \left[K((\mathbf{x}', y'), (\mathbf{X}, Y)) \right] + \mathbb{E}_{\tilde{P}_{\text{train}}(\mathbf{x}) \hat{P}(y|\mathbf{x})} \left[K((\mathbf{x}', y'), (\mathbf{X}, Y)) \right] \\
&+ \lambda \mathbb{E}_{\tilde{P}_{\text{train}}(\mathbf{x}'') \tilde{P}(y''|\mathbf{x}'')}[\alpha(\mathbf{X}'', Y'') K(\mathbf{x}', y'), (\mathbf{X}'', Y'')].
\end{aligned}
$$

□



logloss: 0.74	logloss: 0.65	logloss: 0.48	logloss: 0.41
entropy: 0.93	entropy: 0.86	entropy: 0.67	entropy: 0.45

Figure 9: Performance comparison with robust bias aware classifier using linear features (a), using Gaussian kernels with bandwidth 0.5 (b), using polynomial kernels with order 2 (c) and using polynomial kernels with order 3 (d). Ellipses show the same training and testing data distribution as in Figure 8. The intersection of training distribution and testing distribution is better predicted with kernel methods applied. The corresponding logloss and entropy evaluated on the testing distribution shows that more certain and informative predictions are produced by kernel RBA.

3.4.4 Understanding Kernel RBA

In order to illustrate the effectiveness of kernel RBA, we consider the same datasets from Figure 8 and compare linear RBA and kernel RBA with different kernel types and parameters in Figure 9. Even though kernel methods are usually regarded as a way to introduce non-linearity, its main effect in kernel RBA is the expansion of the constraint space for the adversarial player $\check{\mathsf{P}}(Y|\mathbf{X})$ in the two player game. As shown in Figure 9, kernel RBA achieves better (smaller test logarithmic loss) and more informative (smaller test prediction entropy) predictions in the intersection of training and testing distribution, while the true decision boundary is a linear one.

Moreover, the difference between test entropy and logarithmic loss gradually gets smaller in the last three figures. This corresponds with the property of RBA that test logarithmic loss is always upper bounded by the test entropy (with high probability), as proven for a general case in previous literature (Liu et al., 2015). Therefore, when a larger number of constraints are imposed, i.e., kernel methods are applied, it forms a more restrictive constraint set for $\check{P}(Y|\mathbf{X})$ so that test entropy will bound test loss more and more tightly.

Note that the choice of kernel method and kernel parameters depends on the specific learning problem because we also need to account for overfitting issues in practice. The amount of bias also plays a role in how more training constraints brought by kernel methods help improve over RBA method. Specifically, the larger the bias is, the more RBA will suffer from insufficient constraints from training sample data, which results in larger entropy in test predictions.

3.4.5 Consistency of Kernel RBA

We now analyze some theoretical properties of the kernel RBA method. As stated before, kernel RBA directly minimizes the regularized expected test loss. We start with defining this expected test loss explicitly, parameterized by learned θ, at a specific data point $(\mathbf{x}, y)$ as: $L_{RBA}(\mathbf{x}, y) = \gamma(\theta, \mathbf{x}, y) - \log Z$, where $\gamma(\theta, \mathbf{x}, y) = \frac{P_{train}(\mathbf{x})}{P_{test}(\mathbf{x})}\theta\Phi(\mathbf{x}, y)$ and $\log Z$ is the normalization term. It is easy to check that it follows the basic form of logistic loss except $\gamma(\theta, \mathbf{x}, y)$ consists of one more component: the density ratio. Therefore, L_{RBA} is a Lipschitz loss. Given Theorem 4, the minimizer of expected test L_{RBA} can be represented using training samples, we can conclude that kernel RBA is consistent w.r.t $\mathbb{E}_{P_{test}(\mathbf{x},y)}[L_{RBA}(\theta, \mathbf{x}, y)]$ when equipped with a universal

kernel (Micchelli et al., 2006) in training data, assuming $\frac{P_{\text{train}}(\mathbf{x})}{P_{\text{test}}(\mathbf{x})}$ is accurate, according to consistency properties for Lipschitz loss (Steinwart, 2005).

Theorem 5. *Let* k *be an bounded universal kernel, and regularization* λ *tending to zero slower than* $1/m$ *for the kernel RBA method, with* $\hat{\theta}$ *as the parameter in the resulting predictor, then* $\mathbb{E}_{P_{test(\mathbf{x},y)}}[L_{RBA}(\hat{\theta}, \mathbf{x}, y)] - \mathbb{E}_{P_{test}(\mathbf{x},y)}[L_{RBA}(\theta^*, \mathbf{x}, y)] \xrightarrow{a.s.} 0$.

Next, we explore whether the optimal expected L_{RBA}, $\mathbb{E}_{P_{\text{test}}(\mathbf{x},y)}[L_{RBA}(\theta^*, \mathbf{x}, y)]$, indicates the optimal 0-1 loss on the testing distribution[1]. This is similar to the universal Bayes consistency argument w.r.t 0-1 loss (Bartlett et al., 2006), except that we are making the statement in a specific pair of training and testing distributions.

Corollary 2 (of Theorem 5)**.** *For any pair of distributions that* $P_{train}(\mathbf{x}) > 0$, $P_{test}(\mathbf{x}) > 0$ *and* $P_{train}(y|\mathbf{x}) = P_{test}(y|\mathbf{x})$*, if* $\hat{\eta}(\mathbf{x})$ *is the kernel RBA predictor satisfying all the conditions in Theorem 5, then* $\mathbb{E}_{P_{test}(\mathbf{x},y)}[L_{0-1}(\hat{\eta}(\mathbf{x}), y)] - \mathbb{E}_{P_{test}(\mathbf{x},y)}[L_{0-1}(\eta^*(\mathbf{x}), y)] \xrightarrow{a.s.} 0$.

Note that employing a universal kernel is a sufficient condition for the consistency to hold. Therefore, kernel methods not only provide larger number of features without increasing computational burdens, but also facilitate the theoretical property to hold for kernel RBA. Even though the consistency property is hard to verify empirically, this analysis provide reassurance for producing Bayes optimal predictor when employed with infinite amounts of data. We now prove this Corollary.

---

[1] We assume the density ratio $P_{\text{train}}(\mathbf{x})/P_{\text{test}}(\mathbf{x})$ is accurately estimated in this case and leave the analysis for the case when it is approximate to future work.

Proof. L_{RBA} is a strongly proper composite loss in both the binary (Reid and Williamson, 2010) and multi-class cases (Vernet et al., 2011), which means it satisfies $L_{RBA}(\eta, \hat{\eta}) - L_{RBA}(\eta, \eta) \geq \frac{C}{2}(\hat{\eta} - \eta)^2$ for any $\eta, \hat{\eta} \in [0, 1]$, where η is the Bayes conditional label probability, $\hat{\eta}$ is the estimated label probability function $\eta(\hat{\theta}, x)$ from RBA and $C > 0$ is a constant (Agarwal, 2013; Agarwal, 2014). We then have test expected 0-1 regret be bounded by the expected L_{RBA} regret:

$$\begin{aligned}
&\mathbb{E}_{P_{test}(\mathbf{x},y)}[L_{0-1}(\hat{h}(\mathbf{x}), y)] - \mathbb{E}_{P_{test}(\mathbf{x},y)}[L_{0-1}(h^*(\mathbf{x}), y)] \\
&\leq 2\sqrt{\mathbb{E}_{P_{test}(\mathbf{x},y)}[\hat{\eta}(\mathbf{x}) - \eta^*(\mathbf{x})]^2} \\
&\leq 2\sqrt{\frac{2}{C}\mathbb{E}_{P_{test}(\mathbf{x},y)}[L_{RBA}(\hat{\eta}(\mathbf{x})) - L_{RBA}(\eta^*(\mathbf{x}))]},
\end{aligned}$$

where h is a predictor function that map conditional label probability $\eta(x)$ to label. Here the first inequality is due to property of plug-in classifiers and Jensen's inequality and the second inequality directly comes from the definition of strongly proper losses (Agarwal, 2013; Agarwal, 2014). Therefore, according to Theorem 5, kernel RBA is consistent w.r.t L_{RBA}, and we then conclude that $\mathbb{E}_{P_{test}(\mathbf{x},y)}[L_{0-1}(\hat{\eta}(\mathbf{x}), y)] - \mathbb{E}_{P_{test}(\mathbf{x},y)}[L_{0-1}(\eta^*(\mathbf{x}), y)] \xrightarrow{a.s.} 0$. □

3.4.6 Comparison of the Convergence Behavior between IW and RBA

We demonstrate how the true decision boundary in the testing distribution is recovered with an increasing number of samples when training and testing distribution are fairly close in Figure 10. As shown in the first figure, the decision boundary in the linear case is tilted due to the noise. Equipped with more samples and a universal kernel (Gaussian kernel), the decision

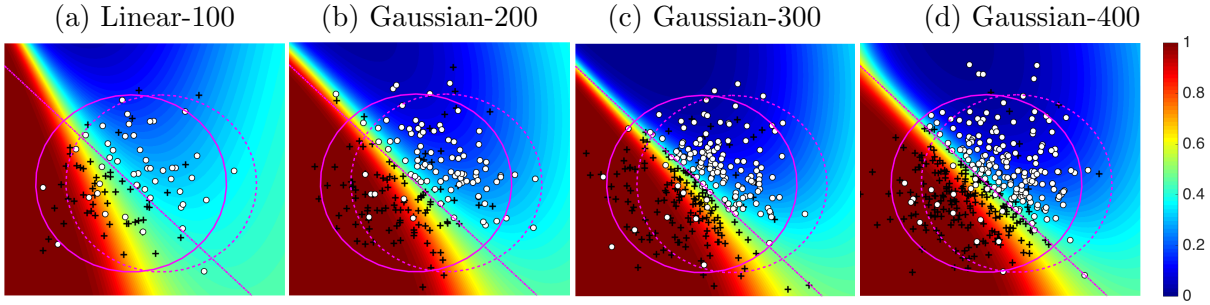


Figure 10: Convergence of decision boundary in RBA classifier using linear features on 100 samples (a), using Gaussian kernels on 200 samples (b), on 300 samples (c) and on 400 samples (d), with 20% noise in each example. Ellipses show training and testing data distribution that closely overlap. The tiled line shows the true decision boundary. With an increasing number of samples and universal kernels, the true decision boundary is recovered with accuracy gradually converging to optimal.

boundary shifts to align with the true one. At the same time, the accuracy on testing data gets better and better, roughly converging to the optimal. This property of kernel RBA corresponds to Corollary 5 that the 0-1 loss of kernel RBA should converge to the optimal 0-1 loss in the limit.

Note that we need kernel features to obtain consistent 0-1 loss minimization in RBA, while importance weighting method does not, which only requires the pre-defined predictor to be consistent on the training data distribution given accurate weights $\mathsf{P}_{\text{test}}(\mathbf{x})/\mathsf{P}_{\text{train}}(\mathbf{x})$. However, that only means the importance weighting method is asymptotically unbiased in the limit. The convergence behavior could have very high variance in practice. We next show an example to

demonstrate that kernel RBA has better convergence tendency than kernel IW with a finite number of data.

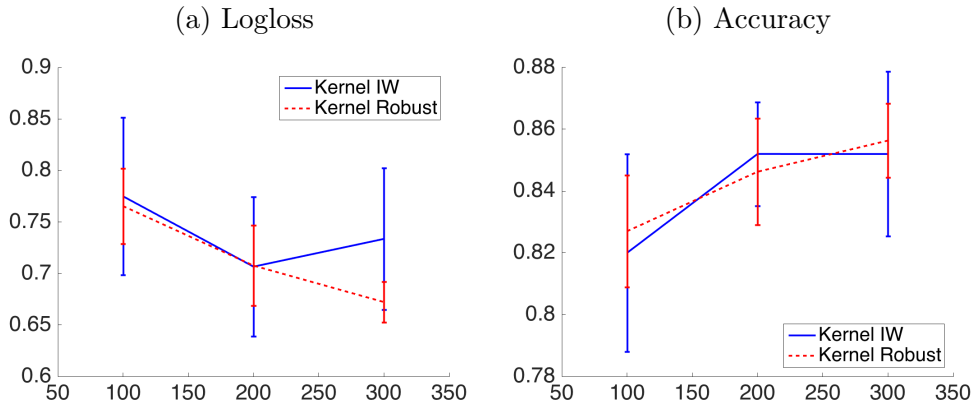


Figure 11: Logloss and accuracy plots as sample size increases from 100 to 300 in kernel IW and kernel Robust methods, with Gaussian kernel, for datasets similar in Figure 10. The error bar shows the 95% confidence interval of the sampling distribution after 20 repeated experiments. IW methods suffer from large variance as robust methods gradually reduce variance and improves on logloss and accuracy more consistently.

As a comparison, we show the plots of logloss and accuracy of Kernel IW (solid line) and Kernel Robust (dashed line) methods after 20 repeated experiments using increasing number of samples in Figure 11. The dataset is similar with the example in Figure 10 with 10% noise and training and testing distribution closely overlapped. The kernel used here is Gaussian kernel. As shown in the error bars, even though the importance weighted loss converges to the test loss in the limit in theory, it suffers from larger variance and sensitivity to noise in reality when

there is only limited number of samples. The reason is that it can be dominated by data with large $\mathsf{P}_{\text{test}}(\mathbf{x})/\mathsf{P}_{\text{train}}(\mathbf{x})$ weights, like points with '+' labels in the right-upper corner in Figure 10. Those noise points will push the decision boundary to the left-bottom direction in order to suffer less logloss. On the other hand, Kernel Robust is more robust to noise and keeps reducing the variance and improving the mean logloss and accuracy. This is not only due to the inherently more modest predictions that robust methods produce on biased testing distribution, but also due to the consistency property it enjoys as stated in Theorem 5 and its Corollary. Even though the number of samples is still small and limited here, the training and testing distribution is close enough to reflect the convergence tendency with the increasing of training samples.

3.5 Robust Multi-view Reformulation

3.5.1 View-based Feature Generalization

The choice of the generalization distribution contributes heavily to the resulting prediction on testing data. We now propose a possible set of assumptions for the generalized formulation in the case when there are multiple views of features. We then explicitly apply assumptions about how each individual view of feature will generalize to the testing distribution. We denote the variables outside of view ν as $\mathbf{x}_{-\nu}$. If we assume that certain view-based features partially generalize from the training distribution to the testing distribution by assuming only the input

variables outside of view ν generalize to testing distribution, this corresponds with the following relationships between inputs:

$$\mathsf{P}_{\text{gen}_\nu}(\mathbf{x}_{-\nu}|\mathbf{x}_\nu) = \mathsf{P}_{\text{test}}(\mathbf{x}_{-\nu}|\mathbf{x}_\nu) \tag{3.21}$$

$$\mathsf{P}_{\text{gen}_\nu}(\mathbf{x}_\nu) = \mathsf{P}_{\text{train}}(\mathbf{x}_\nu). \tag{3.22}$$

Applying the above assumptions in the generalized formulation, the right hand side of the constraints for those generalized views take the form of an importance weighting of view ν's feature vector based on the non-view input variables, $\mathbf{x}_{-\nu}$:

$$\mathbb{E}_{(\mathbf{X},Y)\sim\tilde{\mathsf{P}}_{\text{train}}}\left[\frac{\mathsf{P}_{\text{test}}(\mathbf{X}_{-\nu}|\mathbf{X}_\nu)}{\mathsf{P}_{\text{train}}(\mathbf{X}_{-\nu}|\mathbf{X}_\nu)}\phi_\nu(\mathbf{X}_\nu, Y)\right]. \tag{3.23}$$

We use these partially reweighted features for those generalize features to formulate a new predictor for classification under covariate shift. This view-based robust classifier leverages partial generalization of features, which is possible in many applications when there exists noise or covariate shift in only certain feature views. This provides a solution that is robust to shift in a subset of feature views but also utilizes the information from the ones that are not shifted.

Leveraging the view-based feature generalizations of Equation 3.23, we re-formulate the adversarial game in Definition 3 with set $\mathcal{V}_{\text{g}}$ of generalized views and set $\mathcal{V}_{\text{o}}$ of non-generalized views of features.

Definition 3. *The **robust multi-view covariate shift classifier** is the solution to the adversarial loss optimization game:*

$$\min_{\hat{\mathsf{P}}} \max_{\check{\mathsf{P}}} \mathbb{E}_{\mathbf{X}\sim \mathsf{P}_{test}} \left[Loss(\hat{\mathsf{P}}_{\mathbf{X}}, \check{\mathsf{P}}_{\mathbf{X}}) \right]. \tag{3.24}$$

such that: $\forall \mathcal{v} \in \mathcal{V}_{\mathsf{g}}$,

$$\mathbb{E}_{\mathbf{X}\sim\tilde{\mathsf{P}}_{train},\check{\mathsf{Y}}|\mathbf{X}\sim\check{\mathsf{P}}} \left[\frac{\mathsf{P}_{test}(\mathbf{X}_{-\mathcal{v}}|\mathbf{X}_{\mathcal{v}})}{\mathsf{P}_{train}(\mathbf{X}_{-\mathcal{v}}|\mathbf{X}_{\mathcal{v}})} \phi_{\mathcal{v}}(\mathbf{X}_{\mathcal{v}}, \check{\mathsf{Y}}) \right] = \mathbb{E}_{(\mathbf{X},\mathsf{Y})\sim\tilde{\mathsf{P}}_{train}} \left[\frac{\mathsf{P}_{test}(\mathbf{X}_{-\mathcal{v}}|\mathbf{X}_{\mathcal{v}})}{\mathsf{P}_{train}(\mathbf{X}_{-\mathcal{v}}|\mathbf{X}_{\mathcal{v}})} \phi_{\mathcal{v}}(\mathbf{X}_{\mathcal{v}}, \mathsf{Y}) \right],$$

and for: $\forall \mathcal{v}' \in \mathcal{V}_{\mathsf{o}}$,

$$\mathbb{E}_{\mathbf{X}\sim\tilde{\mathsf{P}}_{train},\check{\mathsf{Y}}|\mathbf{X}\sim\check{\mathsf{P}}} \left[\phi_{\mathcal{v}'}(\mathbf{X}_{\mathcal{v}'}, \check{\mathsf{Y}}) \right] = \mathbb{E}_{(\mathbf{X},\mathsf{Y})\sim\tilde{\mathsf{P}}_{train}} \left[\phi_{\mathcal{v}'}(\mathbf{X}_{\mathcal{v}'}, \mathsf{Y}) \right].$$

This definition implies that there are two different sets of constraints: one set for features that we believe could be generalized, and another set for features that we believe could not. Solving this constrained game formulation based on minimax duality and the method of Lagrangian multipliers for solving convex optimization problems, we have the parametric form of Theorem 6 for conditional label probability distribution when $\text{Loss}(\check{\mathsf{P}}_{\mathbf{X}}, \hat{\mathsf{P}}_{\mathbf{X}}) = \mathbb{E}_{\check{\mathsf{Y}}|\mathbf{X}\sim\check{\mathsf{P}}}[-\log \hat{\mathsf{P}}(\check{\mathsf{Y}}|\mathbf{X})]$.

Theorem 6. *The robust multi-view covariate shift classifier when minimizing expected logloss has the following parametric form:*

$$\hat{\mathsf{P}}_{\theta}(\mathsf{y}|\mathbf{x}) \propto e^{\sum_{\mathcal{v}} \frac{\mathsf{P}_{train}(\mathbf{x}_{\mathcal{v}})}{\mathsf{P}_{test}(\mathbf{x}_{\mathcal{v}})} \theta_{\mathcal{v}}\cdot\phi_{\mathcal{v}}(\mathbf{x}_{\mathcal{v}},\mathsf{y}) + \frac{\mathsf{P}_{train}(\mathbf{x})}{\mathsf{P}_{test}(\mathbf{x})} \sum_{\mathcal{v}'} \theta_{\mathcal{v}'}\cdot\phi_{\mathcal{v}'}(\mathbf{x}_{\mathcal{v}'},\mathsf{y})}, \tag{3.25}$$

where view-specific density ratios, $\mathsf{P}_{train}(\mathbf{x}_{\mathcal{v}})/\mathsf{P}_{test}(\mathbf{x}_{\mathcal{v}})$ *are applied on the generalized views* $\mathcal{V}_{\mathsf{g}}$ *and joint density ratios* $\mathsf{P}_{train}(\mathbf{x})/\mathsf{P}_{test}(\mathbf{x})$ *are applied on non-generalized views* $\mathcal{V}_{\mathsf{o}}$.

We show in the next theorem that in the logloss case, the parameter estimation for θ is equivalent with maximizing the conditional likelihood of testing data with $\hat{P}(Y|X)$ defined as Equation 3.25. Therefore, the parameter can be estimated by using a gradient descent algorithm outlined in Theorem 7 by using reweighted training samples.

Proof. Solving the constrained minimax game (3), the minimax game reduces to a constrained maximum entropy problem:

$$\min_{\hat{P}} \max_{\hat{P}} \mathbb{E}_{\mathbf{X}\sim P_{\text{test}},\hat{Y}|\mathbf{X}\sim\hat{P}} \left[-\log \hat{P}(\hat{Y}|\mathbf{X})\right]. \tag{3.26}$$

$$\text{such that: } \forall \nu \in \mathcal{V}_g, \mathbb{E}_{\mathbf{X}\sim\tilde{P}_{\text{train}},\hat{Y}|\mathbf{X}\sim\hat{P}} \left[\frac{P_{\text{test}}(\mathbf{X}_{-\nu}|\mathbf{X}_\nu)}{P_{\text{train}}(\mathbf{X}_{-\nu}|\mathbf{X}_\nu)}\phi_\nu(\mathbf{X}_\nu,\hat{Y})\right] = \mathbb{E}_{(\mathbf{X},Y)\sim\tilde{P}_{\text{train}}} \left[\frac{P_{\text{test}}(\mathbf{X}_{-\nu}|\mathbf{X}_\nu)}{P_{\text{train}}(\mathbf{X}_{-\nu}|\mathbf{X}_\nu)}\phi_\nu(\mathbf{X}_\nu,Y)\right],$$

$$\text{and for: } \forall \nu' \in \mathcal{V}_o, \mathbb{E}_{\mathbf{X}\sim\tilde{P}_{\text{train}},\hat{Y}|\mathbf{X}\sim\hat{P}} \left[\phi_{\nu'}(\mathbf{X}_{\nu'},\hat{Y})\right] = \mathbb{E}_{(\mathbf{X},Y)\sim\tilde{P}_{\text{train}}} \left[\phi_{\nu'}(\mathbf{X}_{\nu'},Y)\right]$$

$$\forall x \in \mathcal{X} \mathbb{E}_{\hat{Y}|\mathbf{X}\sim\hat{P}}[1|X] = 1 \qquad \forall x \in \mathcal{X}, y \in \mathcal{Y} : \hat{P}(y|x) \geq 0. \tag{3.27}$$

Solving this constrained optimization problem using Lagrangian multiplier method, the Lagrangian is:

$$\begin{aligned}\mathcal{L}(\hat{P}(y|x),\theta,\lambda(\mathbf{x})) = {} & \mathbb{E}_{\mathbf{X}\sim P_{\text{test}},\hat{Y}|\mathbf{X}\sim\hat{P}} \left[-\log \hat{P}(\hat{Y}|\mathbf{X})\right] + \theta_\nu \cdot (\mathbb{E}_{\mathbf{X}\sim\tilde{P}_{\text{train}},\hat{Y}|\mathbf{X}\sim\hat{P}} \left[\frac{P_{\text{test}}(X_\nu|X_{-\nu})}{P_{\text{train}}(X_\nu|X_{-\nu})}\phi_\nu(\mathbf{X}_\nu,\hat{Y})\right] \\ & - \mathbb{E}_{(\mathbf{X},Y)\sim\tilde{P}_{\text{train}}} \left[\frac{P_{\text{test}}(X_\nu|X_{-\nu})}{P_{\text{train}}(X_\nu|X_{-\nu})}\phi_\nu(\mathbf{X}_\nu,Y)\right]) + \theta_{\nu'} \cdot (\mathbb{E}_{\mathbf{X}\sim\tilde{P}_{\text{train}},\hat{Y}|\mathbf{X}\sim\hat{P}} \left[\phi_{\nu'}(\mathbf{X}_{\nu'},\hat{Y})\right] \\ & - \mathbb{E}_{(\mathbf{X},Y)\sim\tilde{P}_{\text{train}}} \left[\phi_{\nu'}(\mathbf{X}_{\nu'},Y)\right]) + \sum_{x\in\mathcal{X}} \lambda(\mathbf{x})[\mathbb{E}_{\hat{Y}|\mathbf{X}\sim\hat{P}}[1|X] - 1].\end{aligned} \tag{3.28}$$

Taking the partial derivative with respect to the conditional probability of a specific y and x, $\hat{P}(y|\mathbf{x})$,

$$\frac{\partial}{\partial \hat{P}(y|\mathbf{x})}\mathcal{L}(\hat{P}(y|\mathbf{x}), \theta, \lambda(\mathbf{x})) = -P_{\text{test}}(\mathbf{x}) \log \hat{P}(y|\mathbf{x}) - P_{\text{test}}(\mathbf{x}) + \sum_{\nu} P_{\text{train}}(\mathbf{x}_\nu) P_{\text{test}}(\mathbf{x}_{-\nu}|x_\nu)\theta_\nu \cdot \phi_\nu(\mathbf{x}_\nu, y) + \sum_{\nu'} P_{\text{train}}(\mathbf{x})\theta_{\nu'} \cdot \phi_{\nu'}(\mathbf{x}_{\nu'}, y) + \lambda(\mathbf{x}), \tag{3.29}$$

setting it equal to zero, and solving it, we obtain:

$$\log \hat{P}(y|x) = -1 + \sum_{\nu} \frac{P_{\text{train}}(\mathbf{x}_\nu)}{P_{\text{test}}(\mathbf{x}_\nu)}\theta_\nu \cdot \phi_\nu(\mathbf{x}_\nu, y) + \sum_{\nu'} \frac{P_{\text{train}}(\mathbf{x})}{P_{\text{test}}(\mathbf{x})}\theta_{\nu'} \cdot \phi_{\nu'}(\mathbf{x}_{\nu'}, y) + \frac{\lambda(\mathbf{x})}{P_{\text{test}}(\mathbf{x})}. \tag{3.30}$$

Therefore,

$$\hat{P}(y|\mathbf{x}) = e^{-1+\sum_\nu \frac{P_{\text{train}}(\mathbf{x}_\nu)}{P_{\text{test}}(\mathbf{x}_\nu)}\theta_\nu \cdot \phi_\nu(\mathbf{x}_\nu, y) + \sum_{\nu'} \frac{P_{\text{train}}(\mathbf{x})}{P_{\text{test}}(\mathbf{x})}\theta_{\nu'} \cdot \phi_{\nu'}(\mathbf{x}_{\nu'}, y) + \frac{\lambda(\mathbf{x})}{P_{\text{test}}(\mathbf{x})}}. \tag{3.31}$$

We analytically solve the normalization terms, yielding the conditional probability distribution:

$$\hat{P}(y|\mathbf{x}) = \frac{e^{\sum_\nu \frac{P_{\text{train}}(\mathbf{x}_\nu)}{P_{\text{test}}(\mathbf{x}_\nu)}\theta_\nu \cdot \phi_\nu(\mathbf{x}_\nu, y) + \sum_{\nu'} \frac{P_{\text{train}}(\mathbf{x})}{P_{\text{test}}(\mathbf{x})}\theta_{\nu'} \cdot \phi_{\nu'}(\mathbf{x}_{\nu'}, y)}}{Z_\theta(\mathbf{x})}, \tag{3.32}$$

where $Z_\theta(\mathbf{x}) = \sum_{y' \in \mathcal{Y}} e^{\sum_\nu \frac{P_{\text{train}}(\mathbf{x}_\nu)}{P_{\text{test}}(\mathbf{x}_\nu)}\theta_\nu \cdot \phi_\nu(\mathbf{x}_\nu, y') + \sum_{\nu'} \frac{P_{\text{train}}(\mathbf{x})}{P_{\text{test}}(\mathbf{x})}\theta_{\nu'} \cdot \phi_{\nu'}(\mathbf{x}_{\nu'}, y')}$. □

Theorem 7. *The parameters of the robust multi-view covariate shift classifier are obtained through implicitly maximizing the conditional likelihood of testing data by taking gradient steps as:*

$$\mathbb{E}_{\mathbf{X}\sim\tilde{\mathsf{P}}_{train},\check{\mathsf{Y}}|\mathbf{X}\sim\check{\mathsf{P}}}\left[\sum_{\nu}\frac{\mathsf{P}_{test}(\mathbf{X}_{-\nu}|\mathbf{X}_{\nu})}{\mathsf{P}_{train}(\mathbf{X}_{-\nu}|\mathbf{X}_{\nu})}\phi_{\nu}(\mathbf{X}_{\nu},\check{\mathsf{Y}})+\sum_{\nu'}\phi_{\nu'}(\mathbf{X}_{\nu'},\check{\mathsf{Y}})\right]$$
$$-\sum_{\nu}\mathbb{E}_{(\mathbf{X},\mathsf{Y})\sim\tilde{\mathsf{P}}_{train}}\left[\frac{\mathsf{P}_{test}(\mathbf{X}_{-\nu}|\mathbf{X}_{\nu})}{\mathsf{P}_{train}(\mathbf{X}_{-\nu}|\mathbf{X}_{\nu})}\phi_{\nu}(\mathbf{X}_{\nu},\mathsf{Y})+\sum_{\nu'}\phi_{\nu'}(\mathbf{X}_{\nu'},\check{\mathsf{Y}})\right]. \tag{3.33}$$

Note that even though we discuss the logloss case here in more detail, the same generalization assumption could also be applied to other loss functions. For those losses, there may not exist analytic forms for $\hat{\mathsf{P}}$, but the sub-gradient should follow the same form as in (Equation 3.33). Therefore, as long as we are able to find an equilibrium of the inner minimax game, we can solve the optimization by sub-gradient descent.

Proof. Plugging in the parametric form of $\hat{P}(y|x)$ into the Lagrangian objective function, we have:

$$\theta^* = \underset{\theta}{\operatorname{argmax}}\, \mathbb{E}_{(\mathbf{X},Y)\sim P_{\text{train}}}[\sum_{\nu} \frac{P_{\text{test}}(\mathbf{X}_{-\nu}|\mathbf{X}_\nu)}{P_{\text{train}}(\mathbf{X}_{-\nu}|\mathbf{X}_\nu)}\theta_\nu \cdot \phi_\nu(\mathbf{X}_\nu, Y) \tag{3.34}$$

$$+ \sum_{\nu'} \theta_{\nu'} \cdot \phi_{\nu'}(\mathbf{X}_{\nu'}, Y)]$$

$$- \mathbb{E}_{\mathbf{X}\sim P_{\text{test}}}\left[\log \sum_{y'\in\mathcal{Y}} e^{\sum_\nu \frac{P_{\text{train}}(\mathbf{X}_\nu)}{P_{\text{test}}(\mathbf{X}_\nu)}\theta_\nu\cdot\phi_\nu(\mathbf{X}_\nu,y')+\sum_{\nu'}\frac{P_{\text{train}}(\mathbf{X})}{P_{\text{test}}(\mathbf{X})}\theta_{\nu'}\cdot\phi_{\nu'}(\mathbf{X}_{\nu'},y')}\right] \tag{3.35}$$

$$= \underset{\theta}{\operatorname{argmax}}\, \mathbb{E}_{(\mathbf{X},Y)\sim P_{\text{test}}}\left[\sum_{\nu} \frac{P_{\text{train}}(\mathbf{X}_\nu)}{P_{\text{test}}(\mathbf{X}_\nu)}\theta_\nu \cdot \phi_\nu(\mathbf{X}_\nu, Y)\right.$$

$$+ \sum_{\nu'} \frac{P_{\text{train}}(\mathbf{X})}{P_{\text{test}}(\mathbf{X})}\theta_{\nu'}\phi_{\nu'}(\mathbf{X}_{\nu'}, Y)$$

$$\left. - \log \sum_{y'\in\mathcal{Y}} e^{\sum_\nu \frac{P_{\text{train}}(\mathbf{X}_\nu)}{P_{\text{test}}(\mathbf{X}_\nu)}\theta_\nu\cdot\phi_\nu(\mathbf{X}_\nu,y')+\sum_{\nu'}\frac{P_{\text{train}}(\mathbf{X})}{P_{\text{test}}(\mathbf{X})}\theta_{\nu'}\cdot\phi_{\nu'}(\mathbf{X}_{\nu'},y')}\right]$$

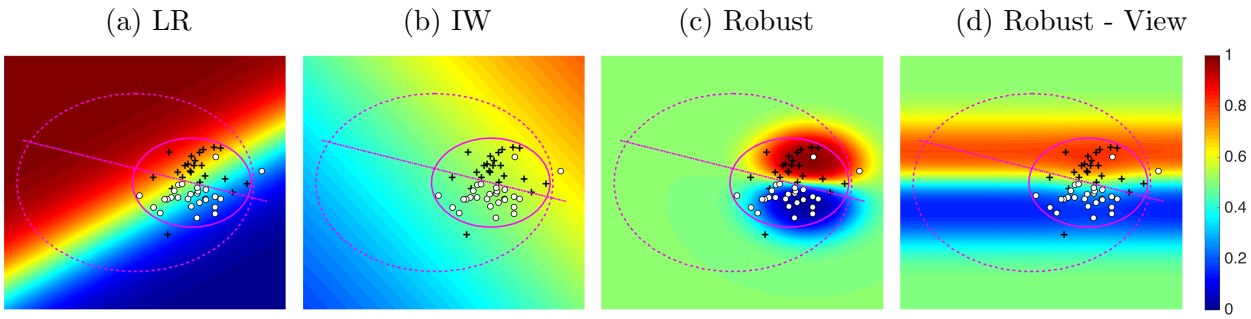
$$= \underset{\theta}{\operatorname{argmax}}\, \mathbb{E}_{(\mathbf{X},Y)\sim P_{\text{test}}}\left[\log \hat{P}_\theta(Y|\mathbf{X})\right] \tag{3.36}$$

□

3.5.2 Understanding the Multi-view Classifier

We consider an illustrative synthetic example with data sampled from two overlapping Gaussian distributions (X) and identical true decision boundary (Y) in Figure 12. In 50 training and 100 testing data points, 10% of the example are chosen uniformly at random to be noise (label flipped). We train four methods, all of which are logloss-based, using training data points (shown in the figures, roughly within the smaller ellipses) and evaluate them on testing data (not shown in the figures, roughly within larger ellipse). The colormap represents the testing

conditional label distribution in the whole space. Logloss evaluated on the testing data is listed below each figure.



logloss = 1.18 logoss = 0.916 logloss = 0.859 logloss = 0.729

Figure 12: Comparison of Logistic Regression (a), Importanct Weighting Logistic Regression (b), Robust Bias-Aware Prediction (c) and View-based Robust Bias-Aware Prediction (d). Logloss evaluated on testing data points (not shown) is shown below each figure. Colormap represents the predicted probability of $\mathsf{P}(\text{`+'} \,|\mathbf{x})$.

We see from the figures that the true decision boundary (the tilted line) could not be recovered by any of the methods using the limited data points. In fact, this is why covariate shift problems are so challenging, even though the assumption $\mathsf{P}_{\text{train}}(\mathsf{y}|\mathsf{x}) = \mathsf{P}_{\text{test}}(\mathsf{y}|\mathsf{x})$ holds. LR makes very certain predictions based on the training data, but produces an incorrect decision boundary and a worse logloss. IW, with reweighted training data, provides a less abrupt decision boundary but remains very certain towards the corners of the input space. The robust method, on the other hand, restricts the certain prediction regions only to areas with enough training

data to support the prediction. The rest of the testing distribution space is covered with uniform predictions. It achieves better testing logloss by being more conservative. However, the question remains: could we leverage more information from the training data? We obtain our answer from the last model: our robust view-based model, which leverages the fact that the view-based training and testing feature distribution is much closer in the vertical dimension ($\mathbf{x}_2$) than in the horizontal dimension ($\mathbf{x}_1$). Thus, the assumption that the training vertical feature dimension can generalize to the testing distribution in our generalized robust covariate shift classifier. This corresponds with a parametric form of $\hat{P}_\theta(y|\mathbf{x}) \propto e^{\frac{P_{train}(\mathbf{x})}{P_{test}(\mathbf{x})}\theta_1 \cdot \phi_v(\mathbf{x}_1,y)+\frac{P_{train}(\mathbf{x}_2)}{P_{test}(\mathbf{x}_2)}\theta_2 \cdot \phi_v(\mathbf{x}_2,y)}$. The partial generalization robust method makes it possible to produce a solution that leverages the benefits of both the conservative robust method and the IW method. It maintains uncertainty in areas that have too little data to make predictions with any certainty (the top and bottom area in the input space), but gives more meaningful predictions in areas where the method expects the data could provide reasonable extrapolations.

3.6 Bounding Expected Worst Case Test Loss

One significant difference between the robust covariate shift methods and empirical risk minimization based methods is that we directly minimize the worst case expected loss under the testing distribution. The reason why this works is that the (sub-)gradient in our formulation only depends on the training distribution, so we are able to use training data to approximate it. On the contrary, the ERM based methods directly approximate the expected loss function using limited data as in Equation 2.3. Despite this difference, we can easily control the error in our (sub-)gradients and therefore bound the error in the optimized worst case expected test loss.

We first define the notation WCLoss(θ) as the (regularized) worst case loss under the testing distribution, which is equivalent with the Lagrangian form of the optimization game of robust covariate shift classifier. Note that WCLoss(θ) differs in meaning from the Loss($\hat{P}, \check{P}$) we used to optimize in the original framework in (Equation 3.1). For example, in logloss case, the worst case testing loss is obtained from the solved parametric form of $\hat{P}$ and $\check{P}$, which is the worst case predictor $P_\theta(\hat{Y}|\mathbf{X})$, to the Loss($\hat{P}, \check{P}$): $\mathbb{E}_{P_{test(X,\hat{Y})}}[-\log P_\theta(\hat{Y}|\mathbf{X})]$.

Theorem 8. *Assuming we have m training samples and n dimensional features, the Lagrangian form of the robust covariate shift classifier (Equation 3.3) is strongly convex in terms of θ with strong convexity constant M, all density estimation is accurate, and the inner minimax game in (Equation 3.3) is solved exactly, the expected loss on testing distribution of the robust covariate shift classifier is bounded, with probability $1-\delta$:*

$$\mathbb{E}_{P_{test}(\mathbf{X})}[WCLoss(\hat{\theta})] \leq \mathbb{E}_{P_{test}(\mathbf{X})}[WCLoss(\theta^*)] + \frac{n \log \frac{2n}{\delta}}{4Mm}.$$

Proof. We first investigate the empirical approximation of (sub-)gradient $\tilde{G}$ and see how far it could deviate from the true (sub-)gradient G.

$$\begin{aligned}
|||G||^2 - ||\tilde{G}||^2| &= |||\mathbb{E}_{(\mathbf{X},Y)\sim P_{train}}[\phi(\mathbf{X},Y)] - \tilde{\phi}||^2 - ||\mathbb{E}_{(\mathbf{X},Y)\sim \tilde{P}_{train}}[\phi(\mathbf{X},Y)] - \tilde{\phi}||^2|, \\
&\leq ||\mathbb{E}_{(\mathbf{X},Y)\sim P_{train}}[\phi(\mathbf{X},Y)] - \mathbb{E}_{(\mathbf{X},Y)\sim \tilde{P}_{train}}[\phi(\mathbf{X},Y)]||^2 \\
&= \sum_{i=1}^{n} |\mathbb{E}_{(\mathbf{X},Y)\sim P_{train}}[\phi_i(\mathbf{X},Y)] - \mathbb{E}_{(\mathbf{X},Y)\sim \tilde{P}_{train}}[\phi_i(\mathbf{X},Y)]|^2, \qquad (3.37)
\end{aligned}$$

where n is the total dimension of features in $\phi(x, y)$. According to Hoeffding bound:

$$P(|\mathbb{E}_{(\mathbf{X},Y)\sim P_{train}}[\phi_i(x,y)] - \mathbb{E}_{(\mathbf{X},Y)\sim \tilde{P}_{train}}[\phi_i(\mathbf{X},Y)]| > \epsilon) < 2e^{-2n\epsilon^2} \tag{3.38}$$

Then we have the following, with probability $1-\delta$,

$$|\|G\|^2 - \|\tilde{G}\|^2| \leq \frac{n\log\frac{2n}{\delta}}{2m}. \tag{3.39}$$

The reason we are interested in the error in norm-2 of (sub-)gradient is we want to utilize the property that for a strongly convex objective function the following is true:

$$f(t) - \min_{s\in S} f(s) \leq \frac{1}{2M}\|\nabla f(t)\|^2, \tag{3.40}$$

where M is the constant for strong convexity, i.e. $f(s) \geq f(t) + \nabla f(t)^T(s-t) + \frac{M}{2}\|s-t\|^2$. This is also true when the objective function is not smooth, when $\nabla f(t)$ can be replaced by subgradient $g \in \partial f(x)$. Therefore, if we assume $\min_{s\in S} f(s)$, which in our case is the true worse case expected test loss, is reached at $\|G\|^2 = 0$, then the objective function is bounded by

$$\mathbb{E}_{(\mathbf{X},Y)\sim P_{test}}[\text{Loss}(\hat{\theta})] \leq \mathbb{E}_{(\mathbf{X},Y)\sim P_{test}}[\text{Loss}(\theta^*)] + \frac{n\log\frac{2n}{\delta}}{4Mm}, \tag{3.41}$$

with probability $1-\delta$, where $\text{Loss}(\theta)$ is the worst case loss function in the general game formulation. □

This bound indicates the distance between the expected test loss induced by our learned model from $\mathfrak{m}$ training data and the optimal test loss is decreasing with speed $\mathcal{O}(\frac{1}{\mathfrak{m}})$. Note that the strong convexity condition is easy to satisfy even with non-smooth loss functions with L_2 regularization.

3.7 Shift-Pessimistic Active Learning

3.7.1 Algorithm

We develop shift-pessimistic active learning method using RBA predictor. The active learner estimates $\hat{\mathsf{P}}(\mathsf{y}|\mathbf{x})$ using our RBA predictor. Here the labeled set is our training data and the pool of unlabeled data is our test. Since we add one data point to the labeled set at every iteration, the training and testing distribution changes over time, which we need to estimate before training using the robust model. We denote the training and testing distribution using $\mathsf{P}_{\mathcal{L}}(\mathbf{x})$ and $\mathsf{P}_{\mathcal{D}}(\mathbf{x})$ and give the basic formulation in the active learning setting. We show the label solicitation for pool-based active learner with covariate shift correction in Algorithm 3

3.7.2 Toy Examples

We first give a comparison for a previous toy example dataset. The key difference from previous methods—RBA's limited, more *pessimistic* extrapolation from available labeled data—is shown in Figure 13b.

3.7.3 Uncertainty Sampling Strategy

There are many possible label solicitation strategies for active learning. We look at the most commonly used one: uncertain sampling. And we found that the uncertainty of our robust bias-aware distribution closely matches to its generalization error (Theorem 9). Note that there

Algorithm 3 Label solicitation for pool-based active learner with covariate shift correction

Input: unlabeled pool dataset $\mathcal{U}$, labeled dataset $\mathcal{L}$

Output: example $\mathbf{x}_{\mathsf{i}} \in \mathcal{U}$ to solicit label

Estimate labeled distribution density $\mathsf{P}_{\mathcal{L}}(\mathbf{x})$

Estimate full data distribution density $\mathsf{P}_{\mathcal{D}}(\mathbf{x})$ ($\mathcal{D} = \mathcal{U} \cup \mathcal{L}$)

Estimate $\hat{\mathsf{P}}(\mathsf{y}|\mathbf{x})$ from dataset $\mathcal{L}$, $\mathsf{P}_{\mathcal{L}}(\mathbf{x})$, and $\mathsf{P}_{\mathcal{D}}(\mathbf{x})$.

Compute $\text{value}_{\mathsf{i}} \leftarrow \text{metric}(\hat{\mathsf{P}}, \mathbf{x}_{\mathsf{i}}, \mathcal{D}, \mathcal{U})$ for each $\mathbf{x}_{\mathsf{i}} \in \mathcal{U}$

return $\mathbf{x}_{\text{argmax}_{\mathsf{i}} \text{value}_{\mathsf{i}}}$ (example label to solicit)

are also disadvantages of the uncertainty sampling strategy under certain circumstances. Our active learner could also benefit from other sampling strategies or mixed sampling strategies.

Theorem 9. *Assuming that the actual label distribution* $\mathsf{P}(\mathsf{y}|\mathbf{x})$ *is within the set* $\tilde{\Xi}$*, the full data entropy of our robust predictor upper bounds its generalization loss:*

$$\begin{aligned} \mathsf{H}_{\mathcal{D}}(\mathsf{Y}|\mathbf{X}) &\triangleq \mathbb{E}_{\mathsf{P}_{test}(\mathbf{x})\hat{\mathsf{P}}(\mathsf{y}|\mathbf{x})}\left[-\log \hat{\mathsf{P}}(\mathsf{Y}|\mathbf{X})\right] \\ &\geq \mathbb{E}_{\mathsf{P}_{test}(\mathbf{x})\mathsf{P}(\mathsf{y}|\mathbf{x})}[-\log \hat{\mathsf{P}}(\mathsf{Y}|\mathbf{X})]. \end{aligned} \tag{3.42}$$

(a) Optimistic active learner (b) Pessimistic active learner

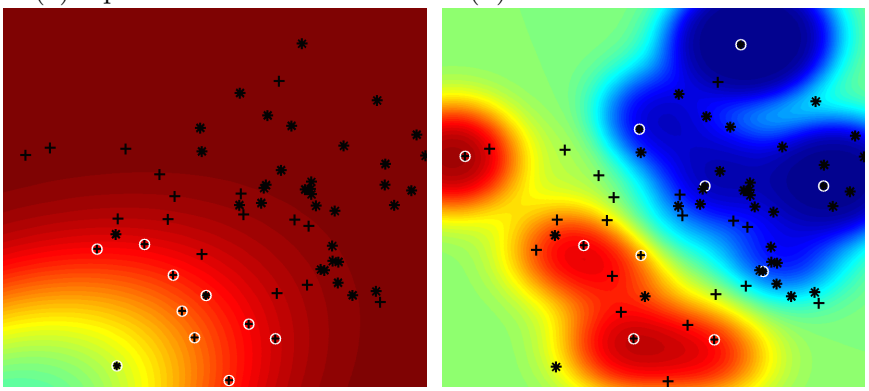


Figure 13: Probabilistic predictions ranging from dark red (+ class) to dark blue (* class) are shown after 10 examples solicited (white circles) from active learning using: (a) a standard optimistic approach—uncertainty sampling (Lewis and Gale, 1994) with logistic regression; and (b) uncertainty sampling using our more pessimistic robust bias-aware active learner.

Proof. The proof follows from two classic papers (Grünwald and Dawid, 2004; Topsøe, 1979) using: (a) strong duality; (b) the equivalence of the logloss minimizer to its evaluation distribution when given; and (c) the assumption that $\mathsf{P}(\mathsf{y}|\mathbf{x})$ is in set $\tilde{\Xi}$:

$$\begin{aligned}
&\min_{\hat{\mathsf{P}}(\mathsf{y}|\mathbf{x})} \max_{\check{\mathsf{P}}(\mathsf{y}|\mathbf{x})\in\tilde{\Xi}} \mathbb{E}_{\mathsf{P}_{\text{test}}(\mathbf{x})\check{\mathsf{P}}(\mathsf{y}|\mathbf{x})}[-\log \hat{\mathsf{P}}(Y|\mathbf{X})] \\
&\overset{(a)}{=} \max_{\check{\mathsf{P}}(\mathsf{y}|\mathbf{x})\in\tilde{\Xi}} \min_{\hat{\mathsf{P}}(\mathsf{y}|\mathbf{x})} \mathbb{E}_{\mathsf{P}_{\text{test}}(\mathbf{x})\check{\mathsf{P}}(\mathsf{y}|\mathbf{x})}[-\log \hat{\mathsf{P}}(Y|\mathbf{X})] \\
&\overset{(b)}{=} \max_{\hat{\mathsf{P}}(\mathsf{y}|\mathbf{x})\in\tilde{\Xi}} \mathsf{H}_{\mathcal{D}}(Y|\mathbf{X}) \overset{(c)}{\geq} \mathbb{E}_{\mathsf{P}_{\text{test}}(\mathbf{x})\mathsf{P}(\mathsf{y}|\mathbf{x})}[-\log \hat{\mathsf{P}}(Y|\mathbf{X})].
\end{aligned}$$

□

Constructing a constraint set $\tilde{\Xi}$ from finite sample data to satisfy the premise of Theorem 9 is overly restrictive. Instead, we can relax the guarantee to be probabilistic based on finite sample error bounds in Corollary 3.

Corollary 3. *When δ defining the ℓ_1-norm or ℓ_2-norm constraint set,*

$$\left|\left|\mathbb{E}_{\tilde{P}_{train}(\mathbf{x})\hat{P}(y|\mathbf{x})}[\phi(\mathbf{X},Y)] - \mathbb{E}_{\tilde{P}_{train}(\mathbf{x})\tilde{P}(y|\mathbf{x})}[\phi(\mathbf{X},Y)]\right|\right| \leq \delta,$$

is chosen using sample error bounds between the labeled data distribution's sample statistics and expected statistics,

$$P\left(\left|\left|\mathbb{E}_{\tilde{P}_{train}(\mathbf{x})\hat{P}(y|\mathbf{x})}[\phi(\mathbf{X},Y)] - \mathbb{E}_{\tilde{P}_{train}(\mathbf{x})\tilde{P}(y|\mathbf{x})}[\phi(\mathbf{X},Y)]\right|\right| \geq \delta\right) \leq \alpha,$$

then the bound (Equation 3.42) *of Theorem 9 holds with probability at least* $(1-\alpha)$.

The constraint set slack δ corresponds to ℓ_1 or ℓ_2 regularization weight λ in the dual optimization problem (Dudík and Schapire, 2006).

3.7.4 Optimizing Different Loss Functions

There is a natural difference in the beliefs we obtain by minimizing different loss functions. We give a synthetic example in Figure 14 to compare the robust logloss minimizer (RBA) and the robust 0-1 loss minimizer. The similarity between these two figures is that we can see clearly how the density ratio affects the prediction. High certainty is mainly located around the region where data points are labeled, which means the data has high density in our training distribution. However, because of the 0-1 loss minimizer's tendency to give a margin-like classifier, the density

(a) Log-loss learner (b) Zero-one loss learner

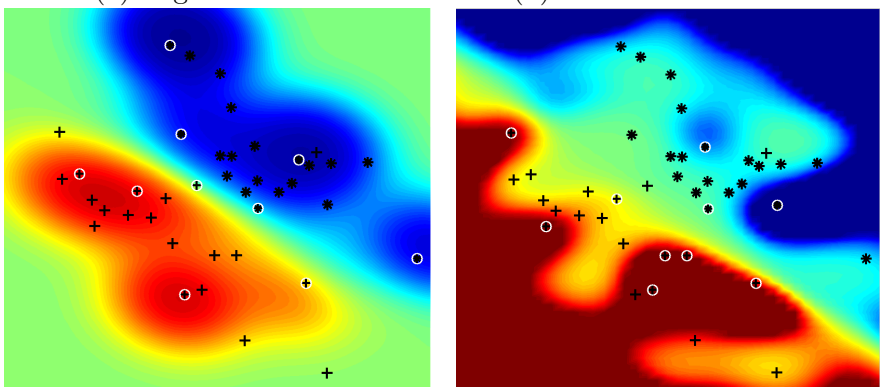


Figure 14: Differences in beliefs of the adversarial log-loss active learner and the adversarial zero-one loss active learner on a synthetic dataset.

ratio will dramatically skew the "decision boundary" according to different densities, while within the region that is far from the "decision boundary," we observe high certainty.

CHAPTER 4

APPLICATIONS

4.1 Robust Bias-Aware Predictor (RBA)

(This section is partially published in the Proceedings of the Neural Information Processing Systems Conference as Robust Classification Under Sample Selection Bias (Liu and Ziebart, 2014).)

We demonstrate the benefit of the robust bias-aware predictor (RBA) using UCI benchmark datasets (Bache and Lichman, 2013).

4.1.1 Comparative Approaches and Implementation Details

We compare three approaches for learning classifiers from biased sample training data:

- **Logistic regression** maximizes conditional likelihood on the training data,

$$\max_{\theta} \mathbb{E}_{\tilde{P}_{\text{train}}(\mathbf{x})\tilde{P}(y|\mathbf{x})}[\log P_\theta(Y|\mathbf{X}) - \epsilon||\theta||]; \tag{4.1}$$

- **Importance weighting logistic regression** minimizes the conditional likelihood of training data reweighted to the testing distribution (Equation 2.3),

$$\max_{\theta} \mathbb{E}_{\tilde{P}_{\text{train}}(\mathbf{x})\tilde{P}(y|\mathbf{x})}[(P_{\text{test}}(\mathbf{x})/P_{\text{train}}(\mathbf{x}))\log P_\theta(Y|\mathbf{X}) - \epsilon||\theta||]; \tag{4.2}$$

- **Robust bias-aware classification** robustly minimizes testing distribution logloss (Equation 3.7) trained using direct gradient calculations (Equation 3.33).

As statistics/features for these approaches, we consider n^{th} order uni-input moments, e.g., $yx_1, yx_2^2, yx_3^n, \ldots$, and mixed moments, e.g., $yx_1, yx_1x_2, yx_3^2x_5x_6, \ldots$. We employ the CVX package (Grant and Boyd, 2014) to estimate parameters of the first two approaches and batch gradient ascent (Algorithm 2) for our robust approach.

4.1.2 Empirical Performance Evaluations and Comparisons

We empirically compare the predictive performance of the three approaches. We consider four classification datasets, selected from the UCI repository (Bache and Lichman, 2013) based on the criteria that each contains roughly 1,000 or more examples, has discretely-valued inputs, and has minimal missing values. We reduce multi-class prediction tasks into binary prediction tasks by combining labels into two groups, as described in Table I.

TABLE I: DATASETS FOR RBA EVALUATION

Dataset	**Features**	**Examples**	**Negative labels**	**Positive labels**
Mushroom	22	8,124	*Edible*	*Poisonous*
Car	6	1,728	*Not acceptable*	all others
Tic-tac-toe	9	958	*'X' does not win*	*'X' wins*
Nursery	8	12,960	*Not recommended*	all others

We generate randomized subsets of these classification datasets to use as our training and testing samples. We accomplish this by sampling a random likelihood function for each from a Dirichlet distribution and then sample training and testing data without replacement in proportion to each datapoint's likelihood. We stress the inherent difficulties of the corresponding prediction task; by design, the training and testing samples are often very different with less overlap that our synthetic experiments. Label imbalance in the samples is also common, despite sampling independently from the example label (given input values) due to samples being drawn from focused portions of the input space. We combine the likelihood function and statistics from the sample to form naïve training and testing distribution estimates.

We sub-sample each original dataset to create biased training and testing datasets using the following procedure:

1. Randomly split half of the dataset into a training set and half into a testing set.
2. For each input dimension, independently sample $P_{\text{train}}(x_k)$ and $P_{\text{test}}(x_k)$ uniformly from the $(|\mathcal{X}_k| - 1)$-simplex (i.e., a Dirichlet$(1, \ldots, 1)$ distribution).
3. Compute $P_{\text{train}}(x_i)$ or $P_{\text{test}}(x_i)$ for each example in the training and testing datasets.
4. Sub-sample N examples from the training and testing distributions in proportion to $P_{\text{train}}(x_i)$ or $P_{\text{test}}(x_i)$ to form $\tilde{P}_{\text{src}}(\mathbf{x})$ and $\tilde{P}_{\text{trg}}(\mathbf{x})$.
5. Set $P_{\text{train}}(x_k) \leftarrow \alpha P_{\text{train}}(x_k) + (1-\alpha)\tilde{P}_{\text{src}}(x_k)$ and set $P_{\text{test}}(x_k) \leftarrow \alpha P_{\text{test}}(x_k) + (1-\alpha)\tilde{P}_{\text{trg}}(x_k)$ for each input dimension k.

We incorporate the fifth step to address datasets with very low likelihood around the sampled training and testing probability distributions. In these datasets, the empirical distribution would deviate substantially from the initial training and testing distributions otherwise. We use $N = 100$ and $\alpha = 0.5$ in our experiments.

Our training and testing distributions make a strong independence assumption, $P(\mathbf{x}) = \prod_{k=1}^{K} P(x_k)$. We limit the negative influence of this naïve independence assumption by bounding the training-testing probability ratio to $[0.0001, 1]$. Without these bounds, we encounter a large number of testing samples that should occur less than "one in a billion" times in our samples of 100 examples.

For the *Mushroom* dataset, we omitted the *stalk-root* feature in our experiments due to it having missing values for some instances.

We considered a range of regularization weights and report the best one for each dataset. Table II lists the best regularization weights employed for each dataset and approach.

TABLE II: REGULARIZATION WEIGHT FOR DIFFERENT DATASETS

Dataset	**Logistic Regression**	**Reweighted**	**Robust**
Mushroom	5	10	0.02
Car	0.5	0.2	0
Nursery	0.2	0.1	0.02
Tic-tac-toe	0.5	0.5	0

Using the best regularization weights is generous to the logistic regression and importance weighting approaches, as their regularization weights are based on how well their inductive biases hold in the testing distribution. This is unknown in the sample selection bias setting. In contrast, approximation error rates for the training distribution statistics guide appropriate regularization parameters for the robust approach. As noted in Section 4.1.2, extremely large regularization can be employed to reduce (or increase) the logloss to 1, but then nothing is learned.

We evaluate the logistic regression model, the importance weighted maximum likelihood model, and our bias-adaptive robust approach. For each, we use first-order and second-order non-mixed statistics: $x_1^2 y, x_2^2 y, \ldots, x_K^2 y, x_1 y, x_2 y, \ldots, x_K y$. For each dataset, we evaluate testing distribution logloss, $\mathbb{E}_{\tilde{P}_{\text{test}}(\mathbf{x})\tilde{P}(y|\mathbf{x})}[-\log \hat{P}(Y|\mathbf{X})]$, averaged over 50 random training and testing samples. We employ $\log_2$ for our loss, which conveniently provides a baseline logloss of 1 for a uniform distribution. We note that with exceedingly large regularization, all parameters will be driven to zero, enabling each approach to achieve this baseline level of logloss. Unfortunately, since testing labels are assumed not to be available in this problem, obtaining optimal regularization via cross-validation is not possible. After trying a range of ℓ_2-regularization weights, we find that heavy ℓ_2-regularization for the logistic regression model and the importance weighting model is needed in our experiments. Without this heavy regularization, the logloss is often extremely high. In contrast, heavy regularization for the robust approach is not necessary; we employ only a mild amount of ℓ_2-regularization corresponding to training statistic estimation error.

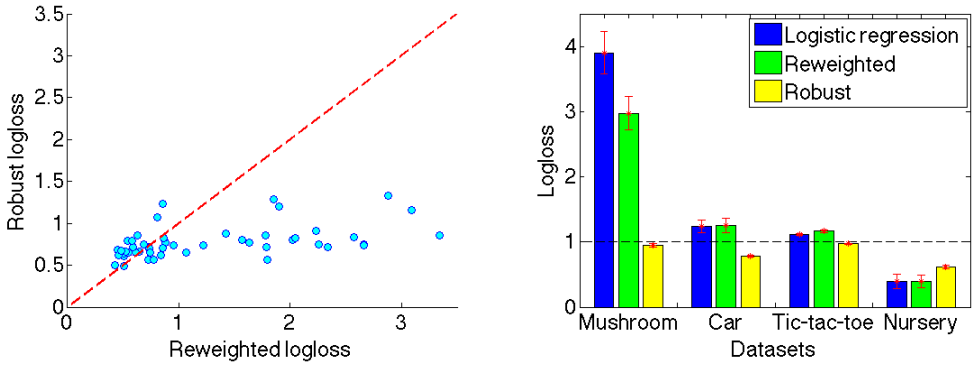


Figure 15: *Left:* Log-loss comparison for 50 training and testing distribution samples between the robust and reweighted approaches for the *Car* classification task. *Right:* Average logloss with 95% confidence intervals for logistic regression, reweighted logistic regression, and bias-adaptive robust testing classifier on four UCI classification tasks.

We show a comparison of individual predictions from the importance weighting approach and the robust approach for the *Car* dataset on the left of Figure 15. The pairs of logloss measures for each of the 50 sampled training and testing datasets are shown in the scatter plot. For some of the samples, the inductive biases of the importance weighting approach provide better predictions (left of the dotted line). However, for many of the samples, the inductive biases do not fit the testing distribution well and this leads to much higher logloss.

The average logloss for each approach and dataset is shown on the right of Figure 15. The robust approach provides better performance than the baseline uniform distribution (logloss of 1) with statistical significance for all datasets. For the first three datasets, the other two approaches are significantly worse than this baseline. The confidence intervals for logistic regression and the importance weighting model tend to be significantly larger than the robust approach because of

the variability in how well their inductive biases generalize to the testing distribution for each sample. However, the robust approach is not a panacea for all covariate shift problems; the *No Free Lunch* theorem (Wolpert, 1996) still applies. We see this with the *Nursery* dataset, in which the inductive biases of the logistic regression and importance weighting approaches do tend to hold across training and testing distributions, providing better predictions.

4.2 Robust Zero-One Loss Minimization

We conduct experiments on real datasets and investigate the performance of the robust 0-1loss minimization from the general framework. In the method, we just consider all features as one view and assumes $\mathsf{P}_{\text{gen}}(\mathbf{x}) = \mathsf{P}_{\text{train}}(\mathbf{x})$. We chose four datasets from the UCI repository (Bache and Lichman, 2013) for this set of experiments. We show the detailed information about each dataset in Table III. In order to create covariate shift, we synthetically generate 30 separate experiments in each dataset by drawing 100 training samples and 100 testing data samples from it, following similar sampling procedure in previous literature (Huang et al., 2006). We show the details below.

1. Separate the data into training and testing portions according to a feature;
2. Perform Principal Component Analysis (PCA) on both portions of data respectively;
3. Generate a random value $\mathtt{a}$ and $\mathtt{b}$ from different intervals;
4. Randomly choose a principal component $\mathtt{i}$, calculate the weight vector as $\mathtt{normpdf}(\mathtt{m_i}, \mu_\mathtt{i}, \sigma_\mathtt{i})$, where $\mu_\mathtt{i} = \mathtt{min}(\mathtt{m_i}) + (\mathtt{max}(\mathtt{m_i}) - \mathtt{min}(\mathtt{m_i}))/\mathtt{a}$, $\sigma_\mathtt{i} = \mathtt{std}(\mathtt{m_i})/\mathtt{b}$;

5. Sample examples for the testing data samples from the testing portion of data in proportion to the weight vector values;

6. Follow the same procedure to sample examples for the training data.

Note that we normalize the data to $[0, 1]$ beforehand. For each method, the regularization parameter λ is chosen using 5-fold cross validation, or importance weighted cross validation (IWCV) on a parameter range $\lambda \in [2^{-16}, 2^{-12}, 2^{-8}, 2^{-4}, 1]$. Here the traditional cross validation is applied on LR, while IWCV is applied on all the other methods. Note that the traditional cross validation process is not correct anymore in the covariate shift setting where the training marginal data distribution of $P(\mathbf{x})$ is different from the testing distribution (Sugiyama et al., 2007). Therefore, standard cross validation only matches the logistic regression method which ignores the bias. Though IWCV was originally designed for the importance weighting methods, it is proven to be unbiased for any loss function. We apply it to perform model tuning for our robust methods, even though the error estimate variance could be large.

TABLE III: DATASETS FOR ROBUST 0-1 EVALUATION

Dataset	**Features**	**Examples**	**Classes**
`Seed`	7	210	3
`Vertebral`	6	310	3
`Vehicle`	18	946	4
`Spam`	57	4601	2

4.2.1 Logistic Regression as Density Estimator

We use a discriminative density estimation method that leverages the logistic regression classifier for estimating the density ratios. According to Bayes rule:

$$\frac{\mathsf{P}_{\text{train}}(\mathbf{x})}{\mathsf{P}_{\text{test}}(\mathbf{x})} = \frac{\mathsf{P}(\mathbf{x}|\text{train})}{\mathsf{P}(\mathbf{x}|\text{test})} = \frac{\mathsf{P}(\text{train}|\mathbf{x})\mathsf{P}(\mathbf{x})/\mathsf{P}(\text{train})}{\mathsf{P}(\text{train}|\mathbf{x})\mathsf{P}(\mathbf{x})/\mathsf{P}(\text{test})} = \frac{\mathsf{P}(\text{train}|\mathbf{x})}{\mathsf{P}(\text{test}|\mathbf{x})}\frac{\mathsf{P}(\text{test})}{\mathsf{P}(\text{train})},$$

where the second ratio $\mathsf{P}(\text{test})/\mathsf{P}(\text{train})$ is computed as the ratio of the number of test and train examples, and the first one is obtained by training a classifier with training data labeled as one class and test data as another class. Similar ideas also appears in recent literature (Lopez-Paz and Oquab, 2016). The resulting density ratio of this method is also closely controlled by the amount of regularization. We also choose the regularization weight by cross validation on $\lambda \in [2^{-16}, 2^{-12}, 2^{-8}, 2^{-4}, 1]$.

4.2.2 Comparative Approaches

We evaluate three methods:

- **Robust bias aware 0-1 classifier (Robust 0-1)** utilizes the general robust covariate shift classification framework (Equation 3.1) with $\text{Loss}(\check{\mathsf{P}}_{\mathbf{X}}, \hat{\mathsf{P}}_{\mathbf{X}}) = \hat{\mathsf{P}}^{\mathsf{T}}\mathsf{C}\check{\mathsf{P}}$ with C as the 0-1 loss matrix and $\mathsf{P}_{\text{gen}_\nu}(\mathbf{x}) = \mathsf{P}_{\text{train}}(\mathbf{x})$.
- **Adversarial 0-1 classifier (Adv 0-1)** minimizes expected 0-1 loss on the training distribution and has an optimization objective of: $\min_\theta \min_{\hat{\mathsf{P}}} \max_{\check{\mathsf{P}}} \mathbb{E}_{\mathbf{X}\sim\mathsf{P}_{\text{train}}}[\hat{\mathsf{P}}^{\mathsf{T}}\mathsf{C}\check{\mathsf{P}} + \sum_\nu \theta_\nu \phi_\nu(X_\nu, \check{Y})] - \sum_\nu \theta_\nu \tilde{\phi_\nu} + \epsilon\|\theta\|_2$, where $\tilde{\phi} = \mathbb{E}_{(\mathbf{X},Y)\sim\tilde{\mathsf{P}}_{\text{train}}}[\phi_\nu(\mathbf{X}_\nu, Y)]$ here.

- **Multiclass SVM (SVM)** follows the popular Crammer-Singer method for multiclass (Crammer and Singer, 2001) by minimizing hinge loss on training data.

4.2.3 Empirical Performance Evaluations and Comparisons

We show the comparison of accuracy in Table IV and highlight methods that are either the best under paired t-test or not statistically distinguishable with significance level 0.1 in bold. We can see that Robust 0-1 performs better than other methods except in `Seed`, where it is statistically no worse than others. And Robust 0-1 can improve from Adv 0-1 at most times. That means minimizing worst case test loss using the adversarial game formulation (Equation 3.1) under covariate shfit is better than minimizing training loss and ignoring the bias using the same formulation.

TABLE IV: AVERAGE ACCURACY COMPARISON FOR ROBUST 0-1 EVALUATION

Datasets	Robust 0-1	Adv 0-1	SVM
`Seed`	**0.834**	**0.820**	**0.820**
`Vertebral`	**0.823**	0.805	0.748
`Vehicle`	**0.547**	0.535	0.497
`Spam`	**0.757**	0.711	0.724

4.3 Kernel RBA

We demonstrate the advantages of our kernel RBA approach on datasets that are either synthetically biased via sampling or naturally biased by a differing characteristic or noise. We

chose three datasets from the UCI repository (Bache and Lichman, 2013) for synthetically biased experiments, based on the criteria that each contains approximately 1,000 or more examples and has minimal missing values. They are `Vehicle`, `Segment` and `Sat`. For each dataset, we synthetically generate 20 separate experiments by taking 200 training samples and 200 testing data samples from it, generally following the sampling procedure described in previous literature (Huang et al., 2006), which we summarize as:

1. Separate the data into training and testing portion according to mean of a variable;
2. Randomly sample the testing portion as the testing dataset;
3. In the training portion, calculate the sample mean μ and sample covariance σ, then sample in proportion to weights generated from a multivariate Gaussian with $\mu' = \mu/5$ and $\sigma' = \sigma/5$ as the training dataset. If the dimension is too large to sample any points, perform PCA first and use the first several principle components to obtain the weights.

We follow the same procedure for density estimation and model selection as in the robust 0-1 loss minimization experiments. Therefore, we omit some duplicated details here.

We also investigate three naturally biased covariate shift datasets. One of them is `Abalone`, in which we use the sex variable (male, female, and infant) to create bias. Specifically, we use infant as training samples and the rest as test samples. Note that we use the simplified 3-category classification problem of the `Abalone` dataset (Clark et al., 1996) and also sample 200 data points respectively for the training and testing datasets. We chose this data because the sex variable makes train-test separation easier and reasonable, and allows the covariate shift assumption to generally hold. In addition, we evaluate our methods on the MNIST dataset (LeCun et al.,

1998), which we reduce to binary predictive tasks of differentiating '3' versus '8' and '7' versus '9'. We add a biased Gaussian noise with mean 0.2 and standard deviation 0.5 to the testing data to form the covariate shift, i.e. noise $z \sim \mathsf{N}(0.2, 0.5)$. We randomly sample 2000 training and testing samples and repeat the experiments 20 times. Shown in Figure 16 is the comparison between one batch of training samples and testing samples.

(a) Training Samples (b) Testing Samples

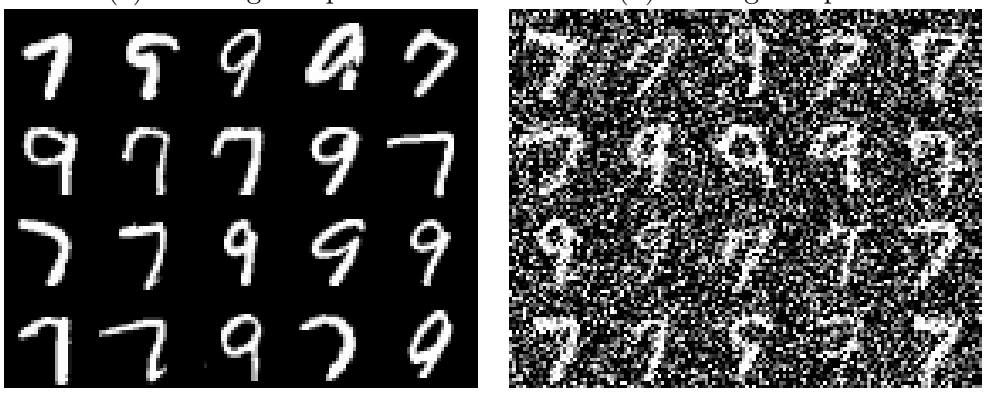


Figure 16: Binarized MNIST data with noise added to the testing set to form covariate shift.

We show more detailed information of the datasets we used in Table V We expect the method to also work for higher dimensional dataset when equipped with accurate density ratio estimation. Since the development and analysis of this paper focus more on the Kernel RBA method itself and not on density estimation, we believe smaller datasets are more suitable for the evaluation.

TABLE V: BIASED DATASETS FOR KERNEL RBA EVALUATION

Dataset	Features	Examples	Classes
Vehicle	18	846	4
Segment	19	2310	7
Sat	36	6435	7
Abalone	7	4177	3
MNIST-3v8	784	5885	2
MNIST-7v9	784	5959	2

4.3.1 Methods

We evaluate our approach and five other methods:

- **Kernel robust bias aware classifier (Kernel Robust)** adversarially minimizes the test distribution logloss using kernel methods, trained using direct gradient calculations as in Corollary 1.
- **Kernel logistic regression (Kernel LR)** ignores the covariate shift and maximizes the training data conditional likelihood, $\max_\theta \mathbb{E}_{P_{\text{train}}(\mathbf{x})P(y|\mathbf{x})}[\log P_\theta(Y|\mathbf{X})] - \lambda\|\theta\|_2^2$, where $\hat{P}_\theta(y|\mathbf{x}) = \frac{\exp(\theta\cdot\Phi(\mathbf{x},y))}{\sum_{y'\in\mathcal{Y}}\exp(\theta\cdot\Phi(\mathbf{x},y'))}$ and λ is the regularization constant.
- **Kernel importance weighting method (Kernel IW)** maximizes the conditional test data likelihood as estimated using importance weighting with the density ratio,

$$\max_\theta \mathbb{E}_{P_{\text{train}}(\mathbf{x})P(y|\mathbf{x})}\left[\frac{P_{\text{test}}(\mathbf{x})}{P_{\text{train}}(\mathbf{x})}\left(\log P_\theta(Y|\mathbf{X})\right)\right] - \lambda\|\theta\|_2^2 \tag{4.3}$$

- **Linear robust bias aware prediction (Robust)** adversarially minimizes the test distribution logloss without utilizing kernelization , i.e. only first order features are used, trained using direct gradient calculations (Equation 3.33).
- **Linear logistic regression (LR)** utilizes only first order features in the training conditional log likelihood maximization.
- **Linear importance weighting method (IW)** uses first order features only to maximize importance weighted training likelihood.

4.3.2 Performance Evaluation

We compare average logloss, $\mathbb{E}_{\tilde{\mathsf{P}}_{\mathrm{trg}}(\mathbf{x})\tilde{\mathsf{P}}(\mathsf{y}|\mathbf{x})}[-\log_2 \widehat{\mathsf{P}}(\mathsf{Y}|\mathbf{X})]$, for each method in Table VI. We perform a paired t-test among each pair of methods. We indicate the methods that have the best performance in bold, along with methods that are statistically indistinguishable from the best (paired t-test with 0.05 significance level). As shown from the table, the average logloss of the Kernel Robust method is significantly better or not significantly worse than all of the alternatives in all of the datasets. Moreover, we make three main observations.

First, logloss of Kernel Robust and Robust is bounded by the uniform distribution baselines, while LR and IW methods can be arbitrary worse when the bias is large, like in `Vehicle`. This aligns with the properties of robust methods because when the bias is large, the density ratio becomes small and results in uniform predictions. This indicates that robust methods should be preferred if robustness or safety is a concern when the amount of covariate shift is large.

Secondly, Kernel Robust consistently improves the performance from Robust while kernelization may harm LR and IW methods, like in `Sat`. The reason is when the implicit assumption

TABLE VI: AVERAGE LOGLOSS COMPARISON FOR KERNEL RBA EVALUATION

Dataset	Kernel Robust	Kernel LR	Kernel IW	Robust	LR	IW
Vehicle	**1.92**	16.41	87.69	1.94	8.15	4.94
Segment	**2.53**	9.62	83.75	2.55	4.37	4.01
Sat	**2.44**	205.27	111.57	2.57	13.27	8.95
Abalone	**1.58**	8.52	6.91	**1.59**	8.73	2.09
MNIST-7v9	**0.42**	0.44	0.49	0.55	0.80	0.59
MNIST-3v8	**0.39**	0.46	0.41	0.48	0.84	0.60

that (reweighted) training features can be generalize to test distribution in LR and IW does not hold anymore, incorporating larger dimensions of features could make predictions worse. For Kernel Robust and Robust, even though overfitting could still be a concern, the density ratio could adjust the certainty of the prediction and function like a regularizer based on the data's density in training and testing distribution, so that they suffer less from overfitting.

Finally, we find that Kernel Robust improvement over Robust is related to how far the training input distributions is from the test input distribution. The natural bias in Abalone comes from one feature variable and could be smaller than the bias in synthetic data. This could be why the improvement of logloss in Abalone is smaller than other datasets.

4.3.3 Accuracy Analysis

We investigate the accuracy (the complement of the misclassification error) of the predictions provided by each of the six approaches on both synthetically biased datasets and naturally biased datasets (in Table VII), where the significant best performance in paired t-test are

TABLE VII: AVERAGE ACCURACY COMPARISON FOR KERNEL RBA EVALUATION

Dataset	Kernel Robust	Kernel LR	Kernel IW	Robust	LR	IW
Vehicle	**38%**	37%	33%	36%	36%	28%
Segment	**71%**	70%	37%	67%	68%	36%
Sat	**33%**	30%	28%	10%	10%	16%
Abalone	46%	43%	42%	**48%**	47%	39%
MNIST-3v8	**88%**	86%	86%	87%	75%	85%
MNIST-7v9	**87%**	85%	86%	86%	71%	83%

demonstrated in bold numbers. The significance level here is 0.05. Despite the discrepancy between the logarithmic loss and the misclassification error, the Kernel Robust approach provides statistically better performance than other alternative methods, except on the Abalone dataset. The logarithmic loss is an upper bound of the 0-1 loss. However, the bound can be somewhat loose, so a lower log loss does not necessarily indicate a smaller classification error rate. This is a natural outcome of using logarithmic loss for convenience of optimization. Since logloss is the natural loss measure for probabilistic prediction and is being optimized by all methods (and not accuracy), we validate our method by comparing to other methods using it. Accuracy and logloss do not correlated perfectly, so it is unsurprising that this small difference exists on a measure not being directly optimized.

4.4 Robust Multi-view Predictor

In our evaluation of the robust multi-view predictor, we regard each feature dimension as a specific view to simplify our experimental setup for UCI datasets. We use the same UCI datasets as in experiments of robust 0-1 minimization. We use KL-divergence as the criterion

to determine the features that are generalizable or not. Besides the UCI datasets, we also evaluate our method on the multi-view dataset `Language` (Amini et al., 2009), which consist of text features of documents in five different languages (English-EN, French-FR, Germany-GR, Italian-IT, and Spanish-SP). This dataset is generated by translating documents originally in one language to the other four languages using machine translation. We regard different language features as different views for this task. In our experiment, we use the document originally in English. We use two languages in training and testing, with one view the same and the other view different between training and testing.

There are six categories as labels, more than ten thousand features for each language and around twenty thousand samples for English documents. To better estimate the densities we use PCA to reduce the dimension of features to 100 for each view and randomly sample 500 data points as training and 500 data points as testing. Therefore, we construct different settings for this dataset. For example, we can train using English and French views and test on Germany and French views (`EN FR - GR FR`). We evaluate the multi-view robust covariate shift approach and three other methods:

- **Multi-view robust bias aware classifier (Robust-View)** utilize the general robust covariate shift classification framework applying multi-view feature generalization assumptions as in Definition 3.
- **Robust bias aware classifier (Robust)** adversarially minimizes the testing distribution logloss Equation 3.9, using the parametric form as $\mathsf{P}(\mathsf{y}|\mathbf{x}) \propto e^{\frac{\mathsf{P}_{\text{train}}(\mathbf{x})}{\mathsf{P}_{\text{test}}(\mathbf{x})} \sum_{\nu} \theta_{\nu} \cdot \phi_{\nu}(\mathbf{x}_{\nu}, \mathsf{y})}$.

- **Logistic regression (LR)** maximizes the conditional log likelihood on training data,

$$\max_{\theta} \mathbb{E}_{P_{\text{train}}(x)P(y|x)} \left[\log P_\theta(Y|X)\right] - \lambda\|\theta\|_2,$$

  where $\hat{P}_\theta(y|x) = \frac{\exp(\theta\cdot\Phi(x,y))}{\sum_{y'\in\mathcal{Y}}\exp(\theta\cdot\Phi(x,y'))}$ and λ is the regularization constant. This approach ignores the covariate shift of the problem setting entirely.

- **Importance weighting method (IW)** maximizes the conditional testing data likelihood as estimated using importance weighting with the density ratio,

$$\max_{\theta} \mathbb{E}_{P_{\text{train}}(x)P(y|x)} \left[\frac{P_{\text{test}}(x)}{P_{\text{train}}(x)} \left(\log P_\theta(Y|X)\right)\right] - \lambda\|\theta\|_2.$$

We follow the same procedure for bias sampling, density estimation and model selection as in the robust 0-1 loss minimization experiments. Therefore, we omit some duplicated details here.

4.4.1 Logistic Regression as Density Estimator

We use a discriminative density estimation method that leverages the logistic regression classifier for estimating the density ratios. For $\frac{\mathsf{P}_{\text{train}}(\mathbf{X})}{\mathsf{P}_{\text{test}}(\mathbf{X})}$ and $\frac{\mathsf{P}_{\text{train}}(\mathbf{X}_v)}{\mathsf{P}_{\text{test}}(\mathbf{X}_v)}$, we can follow the principle as in Section 4.2.1. For the view-related densities, we follow the principle below:

$$\begin{aligned}
\frac{\mathsf{P}_{\text{test}}(\mathbf{X}_{-v}|\mathbf{X}_v)}{\mathsf{P}_{\text{train}}(\mathbf{X}_{-v}|\mathbf{X}_v)} &= \frac{\mathsf{P}_{\text{test}}(\mathbf{X})}{\mathsf{P}_{\text{train}}(\mathbf{X})} \cdot \frac{\mathsf{P}_{\text{train}}(\mathbf{X}_v)}{\mathsf{P}_{\text{test}}(\mathbf{X}_v)} \\
&= \frac{\mathsf{P}(\mathbf{X}|\text{test})}{\mathsf{P}(\mathbf{X}|\text{train})} \cdot \frac{\mathsf{P}(\mathbf{X}_v|\text{train})}{\mathsf{P}(\mathbf{X}_v|\text{test})} \\
&= \frac{\mathsf{P}(\text{test}|\mathbf{X})\mathsf{P}(\mathbf{X})/\mathsf{P}(\text{test})}{\mathsf{P}(\text{train}|\mathbf{X})\mathsf{P}(\mathbf{X})/\mathsf{P}(\text{train})} \cdot \frac{\mathsf{P}(\text{train}|\mathbf{X}_v)\mathsf{P}(\mathbf{X}_v)/\mathsf{P}(\text{train})}{\mathsf{P}(\text{test}|\mathbf{X}_v)\mathsf{P}(\mathbf{X}_v)/\mathsf{P}(\text{test})} \\
&= \frac{\mathsf{P}(\text{test}|\mathbf{X})}{\mathsf{P}(\text{train}|\mathbf{X})} \cdot \frac{\mathsf{P}(\text{train}|\mathbf{X}_v)}{\mathsf{P}(\text{test}|\mathbf{X}_v)}.
\end{aligned} \tag{4.4}$$

The resulting density ratio of this method is also closely controlled by the amount of regularization. We also choose the regularization weight by cross validation.

4.4.2 Generalization Criterion

For UCI datasets, we regard each feature dimension as a view. We evaluate the KL-divergence of the training distribution $\mathsf{P}_{\text{train}}(\mathbf{x}_v)$ and the testing distribution $\mathsf{P}_{\text{test}}(\mathbf{x}_v)$ after density estimation to determine whether we should assume the generalization of each view, i.e., $v \in \mathcal{V}_{\text{o}}$ or $v \in \mathcal{V}_{\text{g}}$. We use the threshold of 0.1, that if $\mathsf{K} < 0.1$, we consider $\mathsf{P}_{\text{train}}(\mathbf{x}_v)$ to

be similar enough with $\mathsf{P}_{\text{test}}(\mathbf{x}_v)$ and $v \in \mathcal{V}_g$, otherwise, $v \in \mathcal{V}_o$. We include both training and testing inputs in the computation of KL-divergence.

$$\begin{aligned}
&\mathsf{K}(\mathsf{P}_{\text{train}}(\mathbf{x}_v), \mathsf{P}_{\text{test}}(\mathbf{x}_v)) \\
&= \sum_{\mathbf{x}_v \in \mathbf{x}_{\text{train}}} \mathsf{P}_{\text{train}}(\mathbf{x}_v) \mathsf{log}(\mathsf{P}_{\text{train}}(\mathbf{x}_v)/\mathsf{P}_{\text{test}}(\mathbf{x}_v)) \\
&+ \sum_{\mathbf{x}_v \in \mathbf{x}_{\text{test}}} \mathsf{P}_{\text{test}}(\mathbf{x}_v) \mathsf{log}(\mathsf{P}_{\text{test}}(\mathbf{x}_v)/\mathsf{P}_{\text{train}}(\mathbf{x}_v))
\end{aligned}$$

For `Language` datasets, we assume we do not have prior knowledge for which view of features should be generalized. We conduct density estimation on both views and detect the one which is similar between training and testing. In practice, we could rely on both data observation and expert knowledge to choose the generalization criterion.

TABLE VIII: ROBUST MULTI-VIEW EVALUATION: AVERAGE LOGLOSS COMPARISON FOR UCI DATASETS

Dataset	**Robust-View**	**Robust**	**LR**	**IW**
`Seed`	**1.039**	**1.105**	1.385	1.299
`Vertebral`	**0.577**	0.830	0.811	0.810
`Vehicle`	**1.68**	1.82	2.82	2.59
`Spam`	**0.853**	1.804	1.981	0.969

We compare logloss of each method in Table VIII. We denote the significantly best result under paired t-test with significance level 0.05 in bold numbers for UCI datasets. We can see from the Table VIII that Robust-View outperforms all other methods in most datasets for UCI experiments by having the lowest logloss. Moreover, it always improve from Robust, except being comparable with Robust in `Seed` for logloss and in `Vehicle` for accuracy. On the other hand, the performance of the other methods are mixed, with Robust achieves slightly better logloss and comparable accuracy with LR and IW. In practice, LR and IW are actually even worse than the Random baseline in terms of logloss due to the possibly large shift between different languages. Robust -View and Robust are usually better than the baseline due to their robustness property. Robust -View can improve from Robust in both logloss by utilizing the generalization property of certain features, especially when Robust is even worse than IR in accuracy because it is overly uncertain with logloss close to random.

TABLE IX: ROBUST MULTI-VIEW EVALUATION: AVERAGE LOGLOSS COMPARISON FOR LANGUAGE DATASETS

Dataset	**Robust-View**	**Robust**	**LR**	**IW**
`EN FR - GR FR`	1.88	2.44	11.04	10.39
`EN GR - FR GR`	1.69	2.38	6.53	6.15
`IT GR - FR GR`	1.96	2.48	8.40	7.59
`EN IT - GR IT`	1.94	2.54	12.72	8.31

For `Language` datasets, whose performance comparison is in Table IX, LR and IW are actually even worse than the Random baseline in terms of logloss due to the possibly large shift between different languages. In contrast, Robust -View and Robust are usually better than the baseline due to their robustness property. And Robust -View can improve from Robust in both logloss by utilizing the generalization property of certain features, especially when Robust is even worse than IR in accuracy because it is overly uncertain with logloss close to random. The reason why Robust is so close to uniform is that it regards all features as a whole and differentiate training and testing data. It disregards the fact that there are useful information that could be used to improve the predictive performance, which is exactly what motivates this work.

4.5 Shift-Pessimistic Active Learning

(This section is partially published in the Association for the Advancement of Artificial Intelligence Conference as Shift-Pessimistic Active Learning Using Robust Bias-Aware Prediction (Liu et al., 2015).)

We have covered the evaluation of several robust predictors in supervised learning tasks with covariate shift. Next, we investigate the performance of different active learning approaches, including our shift-pessimistic active learning using four datasets from the UCI repository (Bache and Lichman, 2013). We consider datasets with real-valued features to simplify density estimation for methods that address covariate shift. We reduce multi-class datasets to binary classification tasks by merging classes (typically plurality class versus other) as detailed in Table X.

TABLE X: DATASETS FOR SHIFT-PESSIMISTIC ACTIVE LEARNING EVALUATION

Dataset	**Features**	**Examples**	**Positive labels**	**Negative labels**
Iris	4	150	*Setosa*	all others
Seed	7	210	*Type "1"*	all others
Banknote	4	1372	*Class "0"*	*Class"1"*
E. coli	8	336	*Cytoplasm*	all others

In each of our experiments, we divide the dataset into a training set (80% of data) and a testing set (the remaining 20%).

4.5.1 Learning Methods

We apply three different models for estimating the conditional label distribution:

- **Standard logistic regression** (abbreviated as **standard** in this section) uses the Boltzmann distribution $\mathsf{P}(\mathsf{y}|\mathbf{x}) = e^{\theta\cdot\phi(\mathbf{x},\mathsf{y})}/(\sum_{\mathsf{y}'\in\mathcal{Y}} e^{\theta\cdot\phi(\mathbf{x},\mathsf{y})})$ and minimizes the logloss of the labeled distribution samples, $\min_\theta \mathbb{E}_{\tilde{\mathsf{P}}_{\text{train}}(\mathbf{x})\tilde{\mathsf{P}}(\mathsf{y}|\mathbf{x})}[-\log \hat{\mathsf{P}}_\theta(Y|\mathbf{X})] + \lambda||\theta||$;
- **importance weighting logistic regression** (abbreviated as **reweighted**) uses the same logistic regression model, but with parameters estimated to minimize the importance weighted estimate of the target loss, which is $\min_\theta \mathbb{E}_{\tilde{\mathsf{P}}_{\text{train}}(\mathbf{x})\tilde{\mathsf{P}}(\mathsf{y}|\mathbf{x})}\left[-\frac{\mathsf{P}_{\text{test}}(\mathbf{X})}{\mathsf{P}_{\text{train}}(\mathbf{X})}\log \hat{\mathsf{P}}_\theta(Y|\mathbf{X})\right] + \lambda||\theta||$;
- **Robust bias-aware prediction** (abbreviated as **robust**) uses the conditional label distribution of Equation 3.7 trained by maximizing target likelihood (Equation 3.9) (approximating the gradient with labeled datapoints).

We employ two label solicitation strategies for each model:

- **Uncertainty sampling** (abbreviated as **active**) selects the example with the largest value-conditioned entropy from the unlabeled dataset. The first datapoint label solicited is selected uniformly at random (the same first datapoint as passive learners); and
- **Random sampling** (abbreviated as **passive**) selects each datapoint uniformly at random from the unlabeled dataset.

In addition, we apply a density-ratio-based strategy with our robust approach:

- **Density-ratio sampling** (abbreviated as **active density**) selects the example with the highest $\mathsf{P}_{\mathcal{D}}(\mathbf{x})/\mathsf{P}_{\mathcal{L}}(\mathbf{x})$ under the estimated distribution.

We conduct 30 experiments with each learner on randomized training/testing splits of each dataset and report the mean and the 95% confidence interval of the predictive performance after every data point solicited in the first 20 steps, corresponding to 0.05 significance level in student t-test. We focus on the first 20 examples because real applications require good predictive performance with limited labeled data.

4.5.2 KDE as Density Estimation

The degree that features from labeled data generalizes to other portions of the input space in the robust approach is controlled by the density estimates. If the labeled data distribution estimate provides minimal support beyond the labeled data samples, density predictions outside of the labeled samples will tend to be overly conservative and maximally uncertain. If the labeled data distribution estimate provides too broad of support for the full data distribution, the guarantees of Corollary 3 will be improbable (i.e., a large α will be required). If the full data

distribution is misestimated, the prediction guarantees (Corollary 3) will not apply to actual full data distribution samples.

When the dimension of the data is not too high, we apply Gaussian kernel density estimation (KDE) on the labeled examples to estimate the labeled data density,

$$\mathsf{P}_{\mathcal{L}}(\mathbf{x}) = \frac{1}{|\mathcal{L}|} \sum_{\mathbf{x}_i \in \mathcal{L}} \mathsf{K}_{\mathbf{H}}(\mathbf{x} - \mathbf{x}_i)$$

with a bandwidth that minimizes the logloss on the whole dataset, $\mathbf{H}$=argmin $\mathbb{E}_{\mathsf{P}_{\mathcal{D}}(\mathbf{x})}[-\log \hat{\mathsf{P}}(\mathbf{X})]$, from a restricted set of bandwidths proportionate to a covariance estimate of the entire data, $\mathbf{H} = \alpha\hat{\Sigma}(\mathcal{D})$. For higher dimensional data (*Seed* and *E. coli*), we first apply principal component analysis to reduce the dimensionality to a space that covers at least 95% of the input variance, before applying Gaussian KDE. We use the uniform distribution over training and testing datapoints for the full data distribution density.

As for the pool, which is the our target in active learning, we leverage specific properties of the active learning setting to help alleviate some of the potentially negative consequences of inaccurate density estimation. We narrow our focus to minimizing the loss on a specific set of

full dataset distribution samples (i.e. all labeled and unlabeled datapoints). Thus, we employ the uniform distribution of datapoints,

$$P_{\mathcal{D}}(\mathbf{x}) = \begin{cases} \frac{1}{|\mathcal{D}|} & \text{if } \mathbf{x} \in \mathcal{D} \\ 0 & \text{otherwise,} \end{cases}$$

to represent the full data distribution density. This would be ill-advised for general covariate shift prediction, because it would make the density ratio $P_{\mathcal{L}}(\mathbf{x})/P_{\mathcal{D}}(\mathbf{x})$ infinite (i.e, no "penalty" for overly certain predictions) at many labeled sample datapoints in $\mathcal{L}$. However, for the active learning setting, all labeled data samples will have support in the full distribution, since $\mathcal{L} \subseteq \mathcal{D}$, so this situation does not occur.

4.5.3 Features and Regularization

For all methods, we use first-order and second-order statistics of the inputs as features: $x_1^2 y$, $x_2^2 y, \ldots, x_K^2 y$, $x_1 x_2 y, x_1 x_3 y, \ldots, x_{K-1} x_K y$, $x_1 y, x_2 y, \ldots, x_K y, y$. Since the regularization weight λ corresponds to slack in the constraints (Corollary 3) and feature scales differ, we use a different regularization weight for each feature corresponding with the 95% confidence interval of the feature's mean, $2\sigma(\phi(\mathbf{x}, y))/\sqrt{|\mathcal{L}|}$. However when the scale of the density ratio is overwhelmingly large (*E.coli*) or small (*Banknote*), we reweight each feature's mean using the learning model's density ratio before taking the standard deviation in the reweighted and robust algorithms.

4.5.4 Optimistic Active Learning versus IID Learning

We first investigate how the optimistic active learning methods (active standard and active reweighted) compare to IID logistic regression (passive standard). In Figure 17a and Figure 17b,

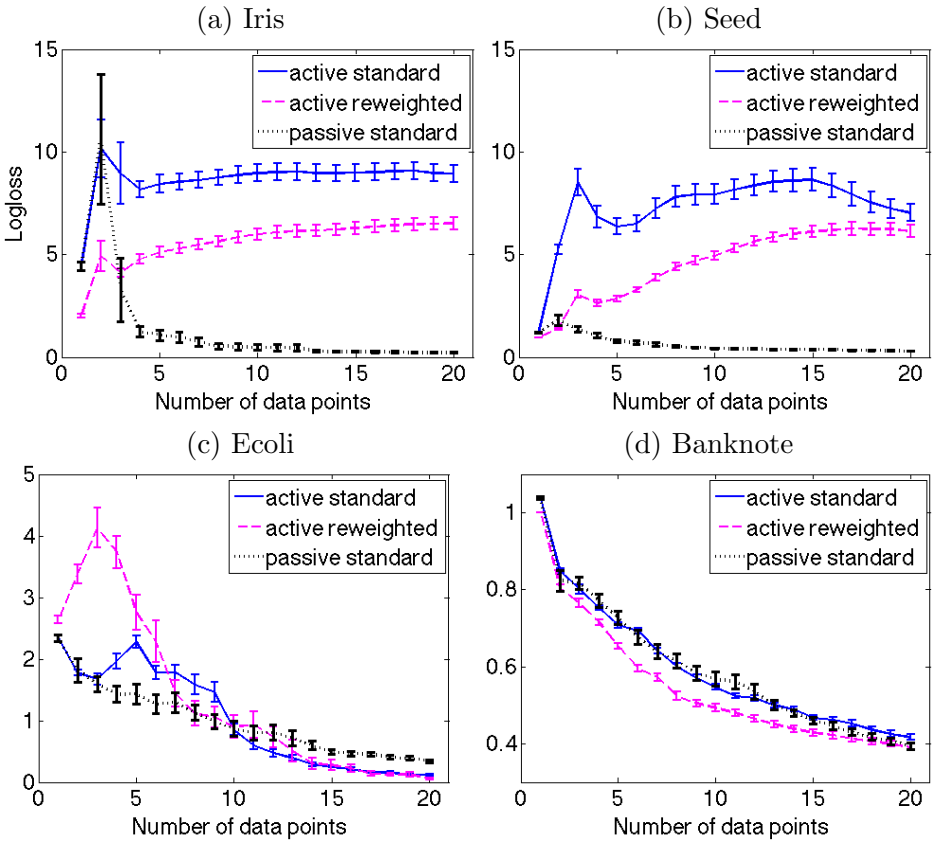


Figure 17: Logloss of optimistic active learning versus passive (IID) learning for the first 20 datapoints of learning averaged over 30 randomized withheld evaluation dataset splits with 95% confidence intervals.

the logloss of active standard and active reweighted are worse than passive learners for the entire 20 steps of learning with statistical significance. This is similarly the case for the active standard and active reweighted algorithms in the first 10 steps of learning in *Banknote* (Figure 17d). This frequent poor performance results from the active learners getting "stuck" soliciting labels suggested by its optimistic biases to be useful rather than labels that would correct its incorrect

beliefs. Further prediction improvements often require first exhausting from the pool of examples that conform to the learner's incorrect beliefs. Only when the inductive biases of labeled data match those of the unlabeled data, as in active methods for the *E. coli* dataset, will the optimistic active learner not provide high logloss in the initial steps of active learning.

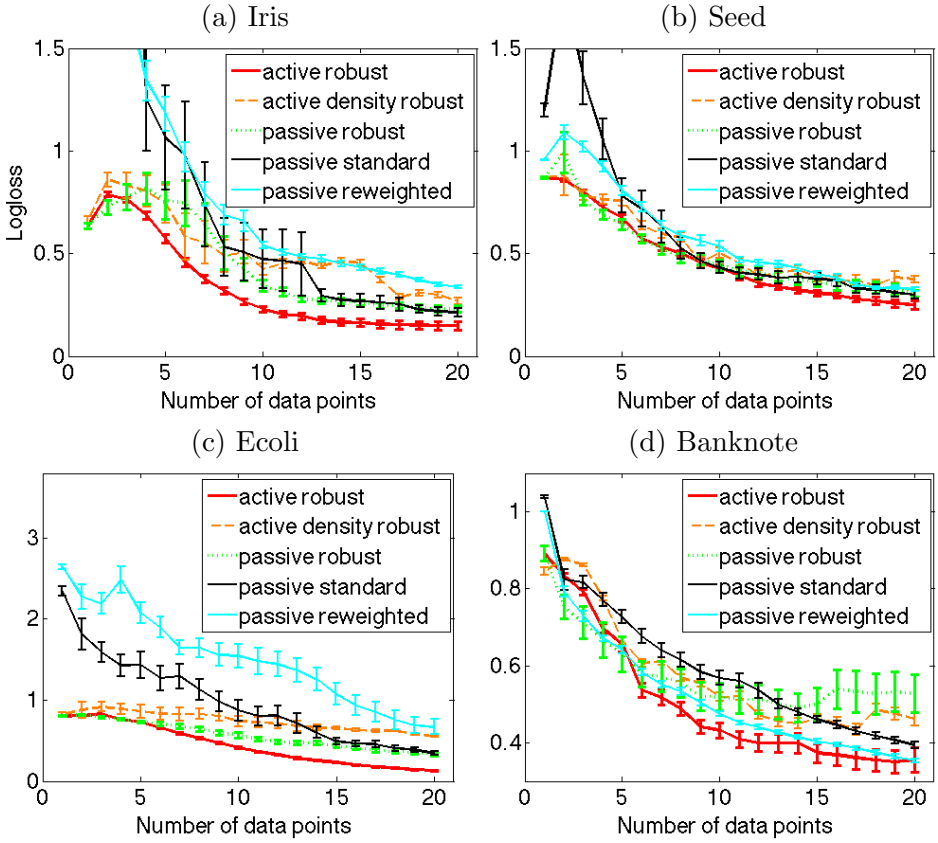


Figure 18: Logloss of shift-pessimistic active learning versus passive (IID) learning for the first 20 datapoints of learning averaged over 30 randomized withheld evaluation dataset splits with 95% confidence intervals.

4.5.5 Pessimistic Active Learning versus IID Learning

We next compare the performance of our shift-pessimistic active learning method (active robust and active density robust) to several passive learning methods. As shown in Figure 18, active robust and active density robust perform better from the very beginning than agnostic baseline, which would provide a logloss of *1*, and are better than, or at least comparable to any other methods for all amounts of available data. Small error bars reflect high stability compared to other methods. In contrast, IID learning methods are quite unstable especially at the beginning due to the bias of a small, randomly chosen sample. Active density robust cannot significantly compete with passive robust because it only considers densities when soliciting labels. Passive robust outperforms passive standard and reweighted, which shows that robust bias-aware prediction effectively controls the extent to which the prediction should generalize. However, since the inductive biases from labeled data tend to generalize accurately using the passive standard and reweighted methods on the *E.coli* dataset, they exceed the passive robust method given 20 labeled examples.

4.5.6 Comparing Classification Accuracy

Though all the algorithms do not minimize classification error directly, the log loss upper bounds the non-convex classification error (0-1 loss). Thus, one might expect that efficiently reducing logarithmic loss in the active learning setting will lead to low classification error. We investigate this in Figure 19, comparing the classification error rate of all seven methods on each dataset. The active robust approach provides the highest prediction accuracy for almost

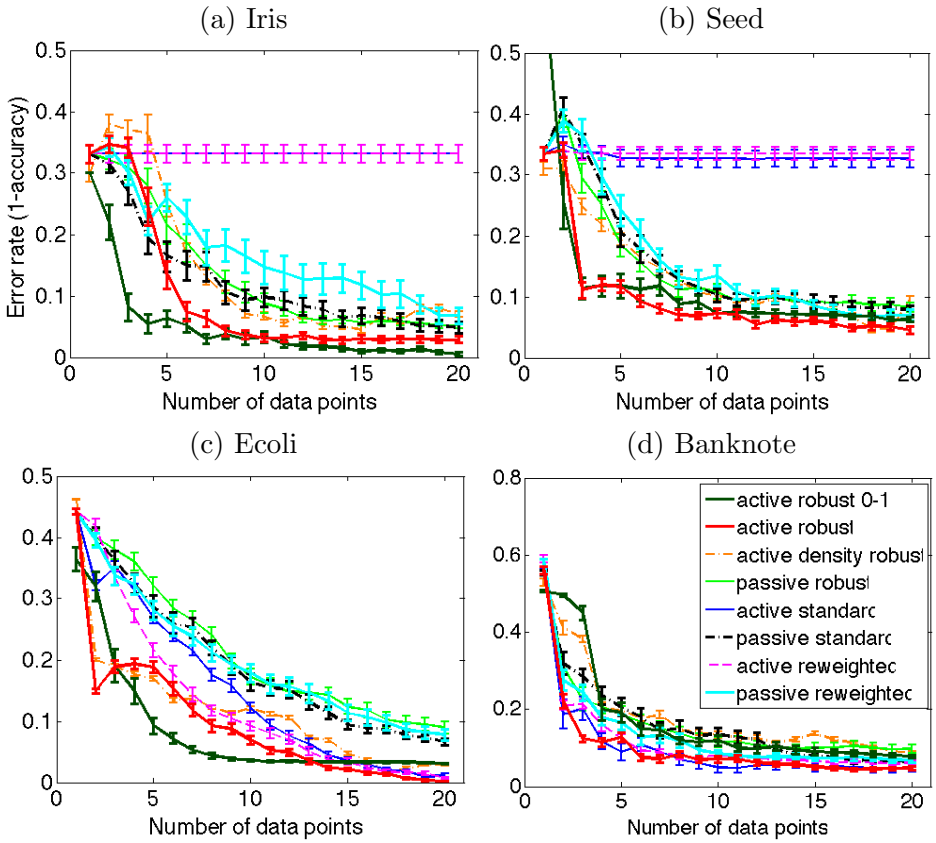


Figure 19: Classification error rate of all learning methods for the first 20 datapoints of learning averaged over 30 randomized withheld evaluation dataset splits. The legend is shared for all datasets. Active standard and active reweighted overlap in (a).

all amounts of available labeled data. In contrast, the high logarithmic loss predictions of active standard and active reweighted in *Iris* and *Seed* translate to poor classification error rates.

CHAPTER 5

CONCLUSION AND DISCUSSION

5.1 Conclusion

In conclusion, we propose a general framework for robust prediction for covariate shift and active learning. The framework takes advantage of robust minimax estimation and is flexible for different loss function minimization. The resulting predictor is the most uncertain under constraints from (generalized) training data. We can compute the (sub)-gradient using only labeled training data for parameter estimation. We show several models generated by our framework work better than state-of-art importance weighting methods for covariate shift. We also develop a pool-based active learning methods based on robust logarithmic loss minimizing and robust Hamming loss minimizing with structured features.

It is a difficult learning task when the covariates shift between training and testing and we lack test labels. We want to be robust to the potentially large shift, so our robust predictor not only produce labels with less certainty when needed, it could also abstain and produce a completely uncertain probability. This reflects a very important rule that is ignored in many machine learning areas, which is we should not make overly optimistic estimates or predictions when we are not sure, especially in areas that is safety critical like self-driving cars and health care. Otherwise, we could make wrong decisions and suffer huge loss. In many sequential predictive tasks like active learning, the mistakes made in early stages harm the learning process

later severely, which is also an area that more conservative predictions may contribute to improve.

5.2 Things Learned from the Study

The study of covariate shift dates back years ago, as well as the study of transfer learning and domain adaptation. Literature in these areas mostly emphasize approaches to leverage information from one domain or distribution to help learning in another domain or distribution. The general assumption that is always implicitly valid there is as long as there are certain similarities or connections between those two or multiple domains, it is going to help. However, it is not true in many experimental results we demonstrated when the importance weights are ill-estimated or models are misspecified. Importance weighting can be regarded as possibly the simplest adaptation approach. This infers that it is actually very important to either tune parameters or engineer features in other adaptation methods, in order to obtain the right degree of generalization. Therefore, it is necessary to face the challenge and investigate the robust way to transfer or adapt that is developed under more constrained assumptions like covariate shift and generates bounded worst-case predictor.

However, it is still tricky to balance the robustness and model informativeness of our approach. Adaptation is all we want when equipped with adequate information. How to detect whether we should adapt or not? We managed to develop the multiview perspective and manipulate the density estimation for different feature views. We also need prior information or some empirical test from the training and testing input variables to set the feature generalization explicitly. In other words, our predictor has the feature generalization distribution explicitly in the model and

there are pros and cons. The pros are obvious that it is clear how the adaptation is accomplished while the cons could be the difficulties in constructing feature generalization distributions.

The logarithmic loss and the zero-one loss is our main focus of loss function in this thesis. One thing comes to our attention is how they penalize probabilistic prediction performance differently, and how this affects the effectiveness of our methods compared with others. In the logarithmic loss case, it penalizes predictions that are confident but wrong severely and provides the guarantee of RBA that it is always bounded by the random baseline which provides benefits useful in practice. Our advantage is not as significant in the zero-one loss cases since it penalizes wrong predictions, either certain or uncertain, by the same. The fact that we would produce more randomized probability could still be very useful in certain areas. However, it does not stand out as much in plain supervised learning settings.

5.3 Challenges in the Future

There are several possible directions for future research opportunities following the thesis. Firstly, we have not explored much in the area of structured prediction. In the future it is promising to explore covariate shift in structured prediction, where we need a better density estimator for high-dimensional data. Secondly, we could benefit from an approach that learns the densities and model parameters at the same time, since they work together to determine the model generalization. Deep networks could also be an interesting structure to explore in this direction.

Theoretical analysis about covariate shift so far is still limited, especially in that the analysis does not often guide the development of better algorithms. We have some analysis in our thesis

that is specially investigated about our own methods, which is a very special case. In the future, the more general covariate shift method analysis that could lead to better algorithms and better condition to check before applying methods in applications is an interesting area.

Last but not least, as mentioned in the thesis, the more conservative predictions could benefit some interactive machine learning methods besides pool-based active learning. For example, in many cases, we need to balance exploiting by prediction given the current, possibly incomplete knowledge of the model parameters and exploring sample space to gain more information about the model. Whether there is potential to utilize our robust prediction framework there could be the good question to answer to widen the range of our possible applications.

CITED LITERATURE

Agarwal, S.: Surrogate regret bounds for the area under the roc curve via strongly proper losses. In Conference on Learning Theory, pages 338–353, 2013.

Agarwal, S.: Surrogate regret bounds for bipartite ranking via strongly proper losses. The Journal of Machine Learning Research, 15(1):1653–1674, 2014.

Altun, Y. and Smola, A.: Unifying divergence minimization and statistical inference via convex duality. In Learning Theory, pages 139–153. Springer Berlin Heidelberg, 2006.

Amini, M., Usunier, N., and Goutte, C.: Learning from multiple partially observed views-an application to multilingual text categorization. In Advances in neural information processing systems, pages 28–36, 2009.

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D.: Concrete problems in ai safety. arXiv preprint arXiv:1606.06565, 2016.

Angluin, D.: Queries and concept learning. Machine learning, 2(4):319–342, 1988.

Asif, K., Xing, W., Behpour, S., and Ziebart, B. D.: Adversarial cost-sensitive classification. In Proceedings of the Conference on Uncertainty in Artificial Intelligence, 2015.

Attenberg, J. and Provost, F.: Inactive learning? Difficulties employing active learning in practice. ACM SIGKDD Explorations Newsletter, 12(2):36–41, 2011.

Bach, F. R.: Active learning for misspecified generalized linear models. In Advances in Neural Information Processing Systems, pages 65–72. MIT Press, 2007.

Bache, K. and Lichman, M.: UCI machine learning repository, 2013.

Bagnell, J. A.: Robust supervised learning. In Proceedings of the 20th national conference on Artificial intelligence-Volume 2, pages 714–719. AAAI Press, 2005.

Bartlett, P. L., Jordan, M. I., and McAuliffe, J. D.: Convexity, classification, and risk bounds. Journal of the American Statistical Association, 101(473):138–156, 2006.

Behpour, S., Kitani, K. M., and Ziebart, B. D.: Ada: A game-theoretic perspective on data augmentation for object detection. 2017.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W.: A theory of learning from different domains. Machine learning, 79(1-2):151–175, 2010.

Ben-David, S., Blitzer, J., Crammer, K., Pereira, F., et al.: Analysis of representations for domain adaptation. Advances in neural information processing systems, 19:137, 2007.

Ben-Tal, A., Den Hertog, D., De Waegenaere, A., Melenberg, B., and Rennen, G.: Robust solutions of optimization problems affected by uncertain probabilities. Management Science, 59(2):341–357, 2013.

Beygelzimer, A., Dasgupta, S., and Langford, J.: Importance weighted active learning. In Proceedings of the International Conference on Machine Learning, pages 49–56. ACM, 2009.

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

Blanchet, J., Kang, Y., and Murthy, K.: Robust wasserstein profile inference and applications to machine learning. arXiv preprint arXiv:1610.05627, 2016.

Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Wortman, J.: Learning bounds for domain adaptation. In Advances in neural information processing systems, pages 129–136, 2008.

Boyd, S. and Vandenberghe, L.: Convex Optimization. Cambridge University Press, 2004.

Chen, X., Monfort, M., Liu, A., and Ziebart, B. D.: Robust covariate shift regression. In Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, pages 1270–1279, 2016.

Chen, X., Monfort, M., Ziebart, B. D., and Carr, P.: Adversarial inverse optimal control for general imitation learning losses and embodiment transfer. In Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence, pages 102–111. AUAI Press, 2016.

Clark, D., Schreter, Z., and Adams, A.: A quantitative comparison of dystal and backpropagation. In Australian Conference on Neural Networks, 1996.

Cortes, C., Mansour, Y., and Mohri, M.: Learning bounds for importance weighting. In Advances in Neural Information Processing Systems, pages 442–450, 2010.

Cortes, C. and Mohri, M.: Domain adaptation and sample bias correction theory and algorithm for regression. Theoretical Computer Science, 519:103–126, 2014.

Cortes, C., Mohri, M., Riley, M., and Rostamizadeh, A.: Sample selection bias correction theory. In Proceedings of the International Conference on Algorithmic Learning Theory, pages 38–53, 2008.

Crammer, K. and Singer, Y.: On the algorithmic implementation of multiclass kernel-based vector machines. Journal of machine learning research, 2(Dec):265–292, 2001.

Crawford, K., Whittaker, M., Elish, M., Barocas, S., Plasek, A., and Ferryman, K.: The ai now report: The social and economic implications of artificial intelligence technologies in the near-term. In AI Now public symposium, hosted by the White House and New York Universitys Information Law Institute, July 7th, 2016.

Daumé III, H.: Frustratingly easy domain adaptation. In Conference of the Association for Computational Linguistics, pages 256–263, 2007.

De Vito, E., Rosasco, L., Caponnetto, A., Piana, M., and Verri, A.: Some properties of regularized kernel methods. The Journal of Machine Learning Research, 5:1363–1390, 2004.

Dligach, D. and Palmer, M.: Good seed makes a good crop: accelerating active learning using language modeling. In Proceedings Annual Meeting of the Association for Computational Linguistics, pages 6–10. Association for Computational Linguistics, 2011.

Duchi, J., Glynn, P., and Namkoong, H.: Statistics of robust optimization: A generalized empirical likelihood approach. arXiv preprint arXiv:1610.03425, 2016.

Dudík, M. and Schapire, R. E.: Maximum entropy distribution estimation with generalized regularization. In Proc. Computational Learning Theory, pages 123–138, 2006.

Dudík, M., Schapire, R. E., and Phillips, S. J.: Correcting sample selection bias in maximum entropy density estimation. In Advances in Neural Information Processing Systems, pages 323–330, 2005.

Elkan, C.: The foundations of cost-sensitive learning. In IJCAI, pages 973–978, 2001.

Esfahani, P. M. and Kuhn, D.: Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations. Mathematical Programming, pages 1–52.

Fan, W., Davidson, I., Zadrozny, B., and Yu, P. S.: An improved categorization of classifier's sensitivity on sample selection bias. In Proc. of the IEEE International Conference on Data Mining, pages 605–608, 2005.

Farnia, F. and Tse, D.: A minimax approach to supervised learning. In Advances in Neural Information Processing Systems, pages 4240–4248, 2016.

Fathony, R., Bashiri, M. A., and Ziebart, B.: Adversarial surrogate losses for ordinal regression. In Advances in Neural Information Processing Systems, pages 563–573, 2017.

Fathony, R., Liu, A., Asif, K., and Ziebart, B.: Adversarial multiclass classification: A risk minimization perspective. In Advances in Neural Information Processing Systems 29, eds. D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, pages 559–567. Curran Associates, Inc., 2016.

Germain, P., Habrard, A., Laviolette, F., and Morvant, E.: A pac-bayesian approach for domain adaptation with specialization to linear classifiers. In International Conference on Machine Learning 2013, pages 738–746, 2013.

Germain, P., Habrard, A., Laviolette, F., and Morvant, E.: A new pac-bayesian perspective on domain adaptation. In International Conference on Machine Learning, pages 859–868, 2016.

Globerson, A., Teo, C. H., Smola, A., and Roweis, S.: An Adversarial View of Covariate Shift and A Minimax Approach. Cambridge, MA, USA, MIT Press, 2009.

Gong, B., Grauman, K., and Sha, F.: Reshaping visual datasets for domain adaptation. In Advances in Neural Information Processing Systems, pages 1286–1294, 2013.

Grant, M. and Boyd, S.: CVX: Matlab software for disciplined convex programming, version 2.1. http://cvxr.com/cvx, March 2014.

Gretton, A., Smola, A., Huang, J., Schmittfull, M., Borgwardt, K., and Schölkopf, B.: Covariate shift by kernel mean matching. Dataset shift in machine learning, 3(4):5, 2009.

Grünwald, P. D. and Dawid, A. P.: Game theory, maximum entropy, minimum discrepancy, and robust Bayesian decision theory. Annals of Statistics, 32:1367–1433, 2004.

Hadfield-Menell, D., Weller, A., Steinhardt, J., and Milli, S.: Reliable machine learning in the wild–icml 2017 workshop, 2017. `https://sites.google.com/site/wildml2017icml/`, Last access: 2018-07-23.

Hadi, A. S. and Luceño, A.: Maximum trimmed likelihood estimators: a unified approach, examples, and algorithms. Computational Statistics & Data Analysis, 25(3):251–272, 1997.

Heckman, J. J.: Sample selection bias as a specification error (with an application to the estimation of labor supply functions), 1977.

Hoang, T. N., Low, K. H., Jaillet, P., and Kankanhalli, M.: Nonmyopic ϵ-bayes-optimal active learning of gaussian processes. In Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32, ICML'14, pages II–739–II–747. JMLR.org, 2014.

Hu, W., Sato, I., and Sugiyama, M.: Robust supervised learning under distribution shift uncertainty. arXiv preprint arXiv:1611.02041, 2016.

Huang, C., Li, Y., Change Loy, C., and Tang, X.: Learning deep representation for imbalanced classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5375–5384, 2016.

Huang, J., Smola, A. J., Gretton, A., Borgwardt, K. M., and Schlkopf, B.: Correcting sample selection bias by unlabeled data. In Advances in Neural Information Processing Systems, pages 601–608, 2006.

Huang, S.-J., Jin, R., and Zhou, Z.-H.: Active learning by querying informative and representative examples. In Advances in neural information processing systems, pages 892–900, 2010.

Jaynes, E. T.: Information theory and statistical mechanics. Physical Review, 106:620–630, 1957.

Jiang, J.: A literature survey on domain adaptation of statistical classifiers. URL: http://sifaka. cs. uiuc. edu/jiang4/domainadaptation/survey, 3, 2008.

Kanamori, T., Hido, S., and Sugiyama, M.: Efficient direct density ratio estimation for non-stationarity adaptation and outlier detection. In Advances in neural information processing systems, pages 809–816, 2009.

Kanamori, T. and Shimodaira, H.: Active learning algorithm using the maximum weighted log-likelihood estimator. Journal of Statistical Planning and Inference, 116(1):149–162, 2003.

Kapoor, A., Grauman, K., Urtasun, R., and Darrell, T.: Active learning with gaussian processes for object categorization. In Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, pages 1–8. IEEE, 2007.

Karampatziakis, N. and Langford, J.: Online importance weight aware updates. arXiv preprint arXiv:1011.1576, 2010.

Kimeldorf, G. and Wahba, G.: Some results on tchebycheffian spline functions. Journal of Mathematical Analysis and Applications, 33(1):82 – 95, 1971.

Krause, A. and Guestrin, C.: Nonmyopic active learning of gaussian processes: an exploration-exploitation approach. In Proceedings of the 24th international conference on Machine learning, pages 449–456. ACM, 2007.

Lafferty, J., McCallum, A., and Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proc. of the International Conference on Machine Learning, pages 282–289, 2001.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

Lewis, D. D. and Gale, W. A.: A sequential algorithm for training text classifiers. In Proc. of the International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 3–12. Springer-Verlag New York, Inc., 1994.

Li, J., Asif, K., Wang, H., Ziebart, B. D., and Berger-Wolf, T.: Adversarial sequence tagging. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, pages 1690–1696. AAAI Press, 2016.

Li, S., Wang, Z., Zhou, G., and Lee, S. Y. M.: Semi-supervised learning for imbalanced sentiment classification. In IJCAI proceedings-international joint conference on artificial intelligence, volume 22, page 1826, 2011.

Liu, A., Fathony, R., and Ziebart, B. D.: Kernel robust bias-aware prediction under covariate shift. arXiv preprint arXiv:1712.10050, 2017.

Liu, A., Reyzin, L., and Ziebart, B. D.: Shift-pessimistic active learning using robust bias-aware prediction. In AAAI, pages 2764–2770, 2015.

Liu, A. and Ziebart, B. D.: Robust classification under sample selection bias. In Advances in Neural Information Processing Systems, pages 37–45, 2014.

Liu, A. and Ziebart, B. D.: Robust covariate shift prediction with general losses and feature views. arXiv preprint arXiv:1712.10043, 2017.

Liu, S., Takeda, A., Suzuki, T., and Fukumizu, K.: Trimmed density ratio estimation. In Advances in Neural Information Processing Systems, pages 4518–4528, 2017.

López, V., Fernández, A., Moreno-Torres, J. G., and Herrera, F.: Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. open problems on intrinsic data characteristics. Expert Systems with Applications, 39(7):6585–6608, 2012.

Lopez-Paz, D. and Oquab, M.: Revisiting classifier two-sample tests. arXiv preprint arXiv:1610.06545, 2016.

Loy, C. C., Hospedales, T. M., Xiang, T., and Gong, S.: Stream-based joint exploration-exploitation active learning. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pages 1560–1567. IEEE, 2012.

Mackay, D.: The evidence framework applied to classification networks. Neural Computation, 4(5):720–736, 1992.

Mansour, Y., Mohri, M., and Rostamizadeh, A.: Domain adaptation: Learning bounds and algorithms. arXiv preprint arXiv:0902.3430, 2009.

Mansour, Y., Mohri, M., and Rostamizadeh, A.: Domain adaptation with multiple sources. In Advances in neural information processing systems, pages 1041–1048, 2009.

Mansour, Y., Mohri, M., and Rostamizadeh, A.: Multiple source adaptation and the rényi divergence. In Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, pages 367–374. AUAI Press, 2009.

McMahan, H. B., Gordon, G. J., and Blum, A.: Planning in the presence of cost functions controlled by an adversary. In Proceedings of the International Conference on Machine Learning, pages 536–543, 2003.

Micchelli, C. A., Xu, Y., and Zhang, H.: Universal kernels. Journal of Machine Learning Research, 7(Dec):2651–2667, 2006.

Mohri, M., Rostamizadeh, A., and Talwalkar, A.: Foundations of Machine Learning. The MIT Press, 2012.

Namkoong, H. and Duchi, J. C.: Stochastic gradient methods for distributionally robust optimization with f-divergences. In Advances in Neural Information Processing Systems, pages 2208–2216, 2016.

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

Reddi, S. J., Póczos, B., and Smola, A.: Doubly robust covariate shift correction. 2014.

Reid, M. D. and Williamson, R. C.: Composite binary losses. Journal of Machine Learning Research, 11(Sep):2387–2422, 2010.

Sabato, S. and Hess, T.: Interactive algorithms: from pool to stream. In Conference on Learning Theory, pages 1419–1439, 2016.

Schapire, R. E. and Singer, Y.: Improved boosting algorithms using confidence-rated predictions. Machine learning, 37(3):297–336, 1999.

Schein, A. I. and Ungar, L. H.: Active learning for logistic regression: an evaluation. Machine Learning, 68(3):235–265, 2007.

Sebastiani, F.: Machine learning in automated text categorization. ACM computing surveys (CSUR), 34(1):1–47, 2002.

Settles, B.: Active learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, 6(1):1–114, 2012.

Settles, B. and Craven, M.: An analysis of active learning strategies for sequence labeling tasks. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 1070–1079. Association for Computational Linguistics, 2008.

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

Sinha, A., Namkoong, H., and Duchi, J.: Certifiable distributional robustness with principled adversarial training. arXiv preprint arXiv:1710.10571, 2017.

Smola, A., Song, L., and Teo, C. H.: Relative novelty detection. In Artificial Intelligence and Statistics, pages 536–543, 2009.

Steinwart, I.: Consistency of support vector machines and other regularized kernel classifiers. IEEE Transactions on Information Theory, 51(1):128–142, 2005.

Sugiyama, M.: Active learning for misspecified models. In Advances in Neural Information Processing Systems, pages 1305–1312, 2005.

Sugiyama, M. and Kawanabe, M.: Machine Learning in Non-stationary Environments: Introduction to Covariate Shift Adaptation. MIT Press, 2012.

Sugiyama, M., Krauledat, M., and Müller, K.-R.: Covariate shift adaptation by importance weighted cross validation. The Journal of Machine Learning Research, 8:985–1005, 2007.

Sugiyama, M., Nakajima, S., Kashima, H., Buenau, P. V., and Kawanabe, M.: Direct importance estimation with model selection and its application to covariate shift adaptation. In Advances in Neural Information Processing Systems, pages 1433–1440, 2008.

Tang, Y., Zhang, Y.-Q., Chawla, N. V., and Krasser, S.: Svms modeling for highly imbalanced classification. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 39(1):281–288, 2009.

Tewari, A. and Bartlett, P. L.: On the consistency of multiclass classification methods. Journal of Machine Learning Research, 8(May):1007–1025, 2007.

Tong, S. and Koller, D.: Support vector machine active learning with applications to text classification. Journal of machine learning research, 2(Nov):45–66, 2001.

Topsøe, F.: Information theoretical optimization techniques. Kybernetika, 15(1):8–27, 1979.

Vernet, E., Reid, M. D., and Williamson, R. C.: Composite multiclass losses. In Advances in Neural Information Processing Systems, pages 1224–1232, 2011.

Wang, H., Xing, W., Asif, K., and Ziebart, B.: Adversarial prediction games for multivariate losses. In Advances in Neural Information Processing Systems, pages 2728–2736, 2015.

Wang, Y., Kucukelbir, A., and Blei, D. M.: Robust probabilistic modeling with bayesian data reweighting. In International Conference on Machine Learning, pages 3646–3655, 2017.

Wen, J., Yu, C.-N., and Greiner, R.: Robust learning under uncertain test distributions: Relating covariate shift to model misspecification. In Proc. of the International Conference on Machine Learning, pages 631–639, 2014.

Wolpert, D. H.: The lack of a priori distinctions between learning algorithms. Neural Comput., 8(7):1341–1390, 1996.

Yamada, M., Suzuki, T., Kanamori, T., Hachiya, H., and Sugiyama, M.: Relative density-ratio estimation for robust distribution comparison. In Advances in neural information processing systems, pages 594–602, 2011.

Yu, Y. and Szepesvári, C.: Analysis of kernel mean matching under covariate shift. In Proc. of the International Conference on Machine Learning, pages 607–614, 2012.

Zadrozny, B.: Learning and evaluating classifiers under sample selection bias. In Proceedings of the International Conference on Machine Learning, pages 903–910. ACM, 2004.

APPENDIX

COPYRIGHT POLICIES

A.1 Copyright Policy of Neural Information Proccessing Systems (NIPS)

All NIPS authors retain copyright of their work. You will need to sign a nonexclusive license giving the NIPS foundation permission to publish the work. Ultimately, however, you can do whatever you like with the content, including having the paper as a chapter of your thesis.

A.2 Copyright Policy of Association for the Advancement of Artificial Intelligence (AAAI)

1. Author(s) agree to transfer their copyrights in their article/paper to the Association for the Advancement of Artificial Intelligence (AAAI), in order to deal with future requests for reprints, translations, anthologies, reproductions, excerpts, and other publications. This grant will include, without limitation, the entire copyright in the article/paper in all countries of the world, including all renewals, extensions, and reversions thereof, whether such rights current exist or hereafter come into effect, and also the exclusive right to create electronic versions of the article/paper, to the extent that such right is not subsumed under copyright.

2. The author(s) warrants that they are the sole author and owner of the copyright in the above article/paper, except for those portions shown to be in quotations; that the article/paper is original throughout; and that the undersigned right to make the grants set forth above is complete and unencumbered.

APPENDIX (Continued)

3. The author(s) agree that if anyone brings any claim or action alleging facts that, if true, constitute a breach of any of the foregoing warranties, the author(s) will hold harmless and indemnify AAAI, their grantees, their licensees, and their distributors against any liability, whether under judgment, decree, or compromise, and any legal fees and expenses arising out of that claim or actions, and the undersigned will cooperate fully in any defense AAAI may make to such claim or action. Moreover, the undersigned agrees to cooperate in any claim or other action seeking to protect or enforce any right the undersigned has granted to AAAI in the article/paper. If any such claim or action fails because of facts that constitute a breach of any of the foregoing warranties, the undersigned agrees to reimburse whomever brings such claim or action for expenses and attorneys fees incurred therein.

4. Author(s) retain all proprietary rights other than copyright (such as patent rights).

5. Author(s) may make personal reuse of all or portions of the above article/paper in other works of their own authorship.

6. Author(s) may reproduce, or have reproduced, their article/paper for the authors personal use, or for company use provided that AAAI copyright and the source are indicated, and that the copies are not used in a way that implies AAAI endorsement of a product or service of an employer, and that the copies per se are not offered for sale. The foregoing right shall not permit the posting of the article/paper in electronic or digital form on any computer network, except by the author or the authors employer, and then only on the authors or the employers own web page or ftp site. Such web page or ftp site, in addition to the aforementioned requirements of this Paragraph, must provide an electronic reference or link back to the AAAI electronic server, and

APPENDIX (Continued)

shall not post other AAAI copyrighted materials not of the authors or the employers creation (including tables of contents with links to other papers) without AAAIs written permission.

7. Author(s) may make limited distribution of all or portions of their article/paper prior to publication.

8. In the case of work performed under U.S. Government contract, AAAI grants the U.S. Government royalty-free permission to reproduce all or portions of the above article/paper, and to authorize others to do so, for U.S. Government purposes.

9. In the event the above article/paper is not accepted and published by AAAI, or is withdrawn by the author(s) before acceptance by AAAI, this agreement becomes null and void.

VITA

NAME Anqi Liu

EDUCATION Ph. D. Computer Science, University of Illinois at Chicago, IL, Expected 2018
B. E. Software Engineering, Tianjin University of Finance and Economics, Tianjin, 2012
B. S. Finance, Tianjin University of Finance and Economics, Tianjin, 2012

EXPERIENCE Research Assistant at UIC, 2013.6-present

Summer Research Assistant Intern at NEC Labs, 2017.5-2017.8

Summer Data Scientist Intern at Microsoft, 2015.5-2015.8

Teaching Assistant, CS107, at UIC, 2012-2013

PUBLICATIONS Nicholas Rhinehart, Anqi Liu, Kihyuk Sohn and Paul Vernaza. "Learning Gibbs-Regularized Pushforward Density Estimators with a Symmetric KL Objective." In submission to NIPS, 2018.

Anqi Liu, Brian D. Ziebart. "Robust Covariate Shift Prediction with General Losses and Feature Views." In arXiv preprint arXiv:1712.10043, 2017.

Anqi Liu, Rizal Fathony and Brian D. Ziebart. "Kernel Robust Bias-Aware Prediction." In arXiv preprint arXiv:1712.10050, 2017.

Rizal Fathony, Anqi Liu, Kaiser Asif and Brian D. Ziebart. "Adversarial Multiclass Classification: A Risk Minimization Perspective." In proceedings of Neural Information Processing Systems (NIPS, 2016).

Xiangli Chen, Mathew Monfort, Anqi Liu and Brian D. Ziebart. Robust Covariate Shift Regression. In Proceedings of International Conference on Artificial Intelligence and Statistics (AISTAT, 2016).

Hong Wang, Anqi Liu, Jing Wang, Brian D. Ziebart, Clement T. Yu, Warren Shen. "Context Retrieval for Web Tables." In ACM Inter-

national Conference on the Theory of Information Retrieval (ICTIR, 2015).

Mathew Monfort, Anqi Liu and Brian D. Ziebart. "Trajectory Forecasting and Intent Recognition via Predictive Inverse Linear-Quadratic Regulation." In proceedings of AAAI conference on Artificial Intelligence (AAAI, 2015).

Anqi Liu, Lev Reyzin and Brian D. Ziebart. Shift-Pessimistic Active Learning using Robust Bias-Aware Prediction. In proceedings of AAAI conference on Artificial Intelligence (AAAI, 2015).

Anqi Liu and Brian D. Ziebart. "Robust Classification under Sample Selection Bias." In proceedings of Neural Information Processing Systems (NIPS, 2014).

WORKSHOPS

Anqi Liu and Brian D. Ziebart. "Robust Covariate Shift Classification with Exact Loss Functions." In NIPS workshop: Aligned Artificial Intelligence, 2017. Contributed Talk.

Anqi Liu, Hong Wang and Brian D. Ziebart. "Robust Covariate Shift Classification using Multiple Feature Views." In NIPS workshop: Reliable Machine Learning in the Wild, 2016.

Anqi Liu, Kaiser Asif, Wei Xing, Sima Behpour, Brian Ziebart, Lev Reyzin. "Addressing Covariate Shift in Active Learning with Adversarial Prediction." In ICML Active Learning Workshop 2015. Contributed Talk

Mathew Monfort, Anqi Liu and Brian D. Ziebart. "Trajectory Forecasting and Intent Recognition via Predictive Inverse Linear-Quadratic Regulation." In IROS Workshop on Assistance and Service Robotics in a Human Environment, 2014.

OTHER TALKS

Anqi Liu. "Avoiding the Pitfalls of Active Learning with Robust Predictors for Covariate Shift." In Booth School of Business, University of Chicago, 2018.

Anqi Liu. "Avoiding the Pitfalls of Active Learning with Robust Predictors for Covariate Shift." In MSR NYC, 2018.

Anqi Liu. "Robust Classification under Covariate Shift with Application to Active Learning." In AAAI Doctoral Consortium, 2016.

HONORS/AWARDS Doctoral Consortium Scholarship, AAAI 2016

Travel Award of AAAI 2015

Travel Award of NIPS, WIML 2014

Travel Award of CRA-W Grad Cohort 2013-2015

Tianjin Municipal Merit Student, 2012

TUFE Best Graduation Project and Thesis Award, 2012

TUFE Merit Student and First Prize Scholarship, 2009-2011

The Wang Kechang Scholarship, 2009

SERVICE Reviewer, NIPS 2018

Reviewer, ICML 2018

Reviewer, NIPS 2017

Reviewer, S.I. ECML PKDD Springer Machine Learning Journal 2017

Reviewer, Women in Machine Learning Workshop 2017

Reviewer, IEEE Transactions on Cybernetics 2016