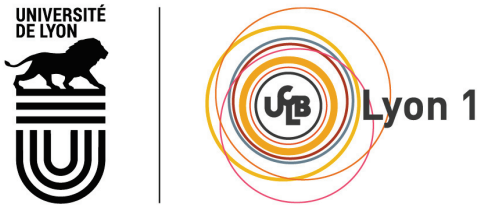


N° d'ordre NNT : 2017LYSE1118

THÈSE DE DOCTORAT DE L'UNIVERSITÉ DE LYON

opérée au sein de

l'Université Claude Bernard Lyon 1

école Doctorale ED512
Infomath

Spécialité de doctorat : Informatique

Soutenue publiquement le 11 Juillet 2017, par :

Van-Tinh Tran

Selection Bias Correction in Supervised Learning with Importance Weight

---

Devant le jury composé de :

Christophe Gonzales, Professeur, Université Paris 6	Présidente
Marc Sebban, Professor, Université de Saint-Etienne	Rapporteur
Elisa Fromont, Maître de Conférence, Université Jean Monnet	Examinatrice
Marianne Clausel, Maître de Conférence, Université Grenoble Alpes	Examinatrice
Céline Robardet, Professeur, INSA Lyon	Examinatrice
Alexandre Aussem, Professeur, Université Lyon 1	Directeur de thèse

UNIVERSITÉ CLAUDE BERNARD - LYON 1

Président de l'Université	**M. le Professeur Frédéric FLEURY**
Président du Conseil Académique	M. le Professeur Hamda BEN HADID
Vice-président du Conseil dAdministration	M. le Professeur Didier REVEL
Vice-président du Conseil Formation et Vie Universitaire	M. le Professeur Philippe CHEVALIER
Vice-président de la Commission Recherche	M. Fabrice VALLÉE
Directrice Générale des Services	Mme Dominique MARCHAND

COMPOSANTES SANTÉ

Faculté de Médecine Lyon Est - Claude Bernard	Directeur : M. le Professeur G.RODE
Faculté de Médecine et de Maïeutique Lyon Sud - Charles Mérieux	Directeur : Mme la Professeure C. BURILLON
Faculté dOdontologie	Directeur : M. le Professeur D. BOURGEOIS
Institut des Sciences Pharmaceutiques et Biologiques	Directeur : Mme la Professeure C. VINCIGUERRA
Institut des Sciences et Techniques de la Réadaptation	Directeur : M. X. PERROT
Département de formation et Centre de Recherche en Biologie Humaine	Directeur : Mme la Professeure A-M. SCHOTT

COMPOSANTES ET DÉPARTEMENTS DE SCIENCES ET TECHNOLOGIE

Faculté des Sciences et Technologies	Directeur : M. F. DE MARCHI
Département Biologie	Directeur : M. le Professeur F. THEVENARD
Département Chimie Biochimie	Directeur : Mme C. FELIX
Département GEP	Directeur : M. Hassan HAMMOURI
Département Informatique	Directeur : M. le Professeur S. AKKOUCHE
Département Mathématiques	Directeur : M. le Professeur G. TOMANOV
Département Mécanique	Directeur : M. le Professeur H. BEN HADID
Département Physique	Directeur : M. le Professeur J-C. PLENET
UFR Sciences et Techniques des Activités Physiques et Sportives	Directeur : M. Y.VANPOULLE
Observatoire des Sciences de lUniversité de Lyon	Directeur : M. B. GUIDERDONI
Polytech Lyon	Directeur : M. le Professeur E.PERRIN
Ecole Supérieure de Chimie Physique Electronique	Directeur : M. G. PIGNAULT
Institut Universitaire de Technologie de Lyon 1	Directeur : M. le Professeur C. VITON
Ecole Supérieure du Professorat et de lEducation	Directeur : M. le Professeur A. MOUGNIOTTE
Institut de Science Financière et d'Assurances	Directeur : M. N. LEBOISNE

Acknowledgements

First and foremost I wish to express my sincere gratitude to my advisor, Prof. Alexandre Aussem, for his tremendous academic support and constant faith in me. He has been giving me all the support and freedom I needed to study and write this thesis.

I would also like to thank the members of the jury for their time, interest, and insightful comments. In particular, I would like to thank my rapporteurs, Prof. Christophe Gonzales and Prof. Marc Sebban, for the considerable work they have devoted to the evaluation of my thesis.

My time in Lyon was made enjoyable in large part due to the many friends and colleagues in the LIRIS laboratory that became a part of my life. I am grateful for their support and the memorable moments that we spent together.

I gratefully acknowledge the funding sources that made my PhD. work possible including the European Union and the French government.

Finally, I would like to thank my family for their encouragement and the faith they put in me that help me go through the most difficult times.

Abstract

In the theory of supervised learning, the identical assumption, i.e. the training and the test samples are drawn from the same probability distribution, plays a crucial role. Unfortunately, this essential assumption is often violated in the presence of selection bias. Under such condition, the standard supervised learning frameworks may suffer a significant bias. In this thesis, we use the importance weighting method to address the selection bias problem in supervised learning.

We first introduce the supervised learning frameworks and discuss the importance of the identical assumption. We then study the importance weighting framework for the generative and the discriminative learning under a general selection scheme and investigate the potential of Bayesian Network to encode a priori assumptions about the relationships between the variables in study, including the selection variable, and to infer the independence and the conditional independence relationships that allow the selection bias to be corrected.

We pay special attention to covariate shift, i.e. a special class of selection bias where the conditional distribution, $P(y|x)$, of the training and of the test data are the same. We propose two methods to improve the importance weighting for covariate shift. We first show that the unweighted model is locally less biased than the weighted one on the low importance instances, and then propose a method that combines them in order to improve the predictive performance in the target domain. Finally, we investigate the relationship between the covariate shift and the missing data problem for data sets with small sample sizes and study a method that uses missing data imputation techniques to correct the covariate shift in some simple but realistic scenarios.

Résumé

Dans la théorie de l'apprentissage supervisé, l'hypothèse selon laquelle l'échantillon de d'apprentissage et de test proviennent de la même distribution de probabilité, joue un rôle crucial. Malheureusement, cette hypothèse essentielle est souvent violée en présence d'un biais de sélection. Dans ce contexte, les algorithmes d'apprentissage supervisés standards peuvent souffrir d'un biais significatif. Dans cette thèse, nous abordons le problème du biais de sélection en apprentissage supervisé en utilisant la méthode de pondération de l'importance ("importance weighting" en anglais).

Dans un premier temps, nous présentons le cadre formel de l'apprentissage supervisé et discutons des effets potentiellement néfastes du biais sur les performances prédictives. Nous étudions ensuite en détail comment les techniques de pondération de l'importance permettent, sous certaines hypothèses, de corriger le biais de sélection durant l'apprentissage de modèles génératifs et discriminants. Nous étudions enfin le potentiel des réseaux bayésiens comme outils de représentation graphique des relations d'indépendances conditionnelles entre les variables du problème et celles liées au mécanisme de sélection lui-même. Nous illustrons sur des exemples simples comment la graphe, construit avec de la connaissance experte, permet d'identifier a posteriori un sous-ensemble restreint de variables sur lesquelles agir pour réduire le biais.

Dans un second temps, nous accordons une attention particulière au covariate shift , i.e. un cas particulier de biais de sélection où la distribution conditionnelle $P(y|x)$ est invariante entre l'échantillon d'apprentissage et de test. Nous proposons deux méthodes pour améliorer la pondération de l'importance en présence de covariate shift. Nous montrons d'abord que le modèle non pondéré est localement moins biaisé que le modèle pondéré sur les échantillons faiblement pondérés, puis nous proposons une première méthode combinant les

modèles pondérés et non pondérés afin d'améliorer les performances prédictives dans le domaine cible. Enfin, nous étudions la relation entre le covariate shift et le problème des données manquantes dans les jeux de données de petite taille et proposons une seconde méthode qui utilise des techniques d'imputation de données manquantes pour corriger le covariate shift dans des scénarios simples mais réalistes. Ces méthodes sont validées expérimentalement sur de nombreux jeux de données.

Contents

Introduction

Selection bias, the problem when data are selected to training sets with an uneven probability across instances, occurs in a wide array of domains for a variety of reasons. This preferential sampling is pervasive in almost all empirical studies, including Machine Learning, Statistics, Social Sciences, Economics, Bioinformatics, Biostatistics, Epidemiology, Medicine, etc. Case-control studies in Epidemiology are particularly susceptible to selection bias, including bias resulting from inappropriate selection of controls in case-control studies, bias resulting from differential loss-to-follow up, incidence-prevalence bias, volunteer bias, healthy-worker bias, and nonresponse bias. In studies of occupational diseases, it is observed that: workers often exhibit lower overall death rates than the general population because the relatively healthy individuals are more likely to gain employment and to remain employed. Selection bias has also received a great deal of attention in econometrics. For instance, surveys are usually prone to contain volunteer bias since those who are willing to participate, thus included in training data, have a particular attitude or characteristic that is different from those who refuse to participate.

Selection bias causes the distribution of collected data used in the training phase to deviate from that of the general population. In supervised learning, selection bias usually causes a drop in the performance of predictive models because learning from one distribution then predicting on another distribution violates the basic independent and identical sampling assumption that almost every learning algorithm makes and invalidates any established performance guarantee.

Abstractly, we may consider an underlying random process that generates selection bias data. This generative process can be decomposed into an unbiased data process that generates independent and identically distributed

examples and a selection process that determines which examples of the unbiased data process will be included into the training set. The selection process or selection mechanism can be modeled by a binary variable, called selection variable, that takes the value of 1 when the examples is selected and the value of 0 otherwise. The use of the selection variable allows us to model the interaction between the selection mechanism and other variables in the study using conditional independence concepts and graphical models. For example, in survey data, the unbiased data process would generate a data set that contains every person of the whole population with the same probability. The selection mechanism then decides which person is more likely to be included into the study. We might presume that if a person have a certain interest or socioeconomic status, he or she might be more willing and more likely to participate than others. Covariate shift is a class of selection bias that received a lot of attention in machine learning and other research communities in recent years. Using graphical model we can characterize the selection mechanism in a meaningful way that could determine which additional data set allows correcting for selection bias.

Outline and Contributions

The focus of this thesis is on the algorithms for learning and predicting in the present of selection bias with the use of an additional data set beside the original training data and a graphical model that characterizes the selection mechanism. The first Chapter reviews the supervised learning frameworks. The second chapter reviews the selection bias problem and is followed by our first contribution in this thesis which is the method of using the importance weight to correct the selection bias. The last two chapters present two methods that we developed to improve the importance weighting techniques that are used to correct the bias caused by covariate shift, the most common types of selection bias.

The first Chapter introduces the supervised learning frameworks used in this thesis. We begin by reviewing its general formulation and the Bayes optimal prediction function. We briefly present the generative frameworks for approximating prediction functions including the Bayesian, the Maximum

a Posteriori and the Maximum Likelihood frameworks. We then review the learning theory that justifies why we can carry out the optimization on the training data and expect a certain level of generalization to the test data. In this theory, the assumption that the training and the test samples are drawn from the same probability distribution plays a crucial role. We discuss the difficulty of supervised learning in terms of the approximation-estimation error trade-off which leads to the inevitable model misspecification, an importance characteristic that affects the selection bias problem. Finally we review the holdout validation and the cross validation, which are two empirical procedures to estimate the prediction accuracy.

The crucial assumption that the training and the test samples are drawn from the same probability distribution is unfortunately often violated in the presence of selection bias. Under such condition, the learning frameworks presented in Chapter 1 need to be adjusted to remain valid. In chapter 2 we first define some useful terminologies and classes of selection bias. We then introduce the importance weighting framework for the generative and the discriminative learning. Given the importance weight, the adaptation of the generative learning to selection bias is very straight forward. We can approximate the generative distribution of the training data to a family of probability distributions using the training data and then adjust it by the importance weight to obtain an approximation of the test distribution before inferring the prediction function. On the other hand, the adaptation of the discriminative learning to the selection bias requires more complication. We introduce two methods that use the importance weight for correcting the selection bias in discriminative learning: one with sampling and the other with modification of loss function. We then show that the importance weighted cross validation gives an almost unbiased estimate of the generalization error. We review the covariate shift, which affects the prediction accuracy when coupling with the model misspecification, and common methods for learning the importance weight from the training data and a set of unlabeled examples. We also investigate the potential of Bayesian Networks to encode a priori assumptions of about the relationship between variables, including the selection variable, and to infer the independence and the conditional independence relationships that allow selection bias to be corrected. In the experimentation section, we as-

sess the ability of the importance weighting method in removing the complete selection bias based on the independence and the conditional independence relationships read from Bayesian Networks.

We observe that the bias in covariate shift is caused only by the model misspecification and not by the change of decision boundary. Therefore using the weighted model to predict every test instance may be excessive since the importance weighting usually reduces the effective sample size as a harmful side effect. In chapter 3, we show analytically that, while the unweighted model is globally more biased than the weighted one, it may locally be less biased on low importance instances. In view of this result, we then discuss a manner to optimally combine the weighted and the unweighted models in order to improve the predictive performance in the target domain. We conduct a series of experiments on the synthetic and the real-world data to demonstrate the efficiency of this approach.

Chapter 4 investigates the relationship between the covariate shift and the missing data problem and explores the possibility of using the missing data imputation to improve the covariate shift correction. The importance weighting even when being used partially as in previous chapter still reduces the effective sample size. In this chapter, we show that there exists a weighting scheme on the unlabeled data such that the combination of the weighted unlabeled data and the labeled training data mimics the test distribution. We further prove that the labels are missing at random in this combined data set and thus can be imputed safely in order to mitigate the undesirable sample-size-reduction effect of the importance weighting. A series of experiments on the synthetic and the real-world data are conducted to demonstrate the efficiency of our approach.

Chapter 1

Supervised Learning Framework

This chapter introduces the supervised learning frameworks used in this thesis. We begin by reviewing its general formulation and the Bayes optimal prediction function. We briefly present the generative frameworks for approximating prediction function including Bayesian, Maximum a Posteriori and Maximum Likelihood. We then review the learning theory that justifies why we can carry optimization on the training data and expect a certain level of generalization to the test data. We discuss the difficulty of supervised learning in terms of the approximation-estimation error trade-off that leads to the inevitable model misspecification, an importance characteristic affecting selection bias. Finally we review the holdout validation and the cross validation, two empirical procedures to estimate prediction accuracy. .

1 Formalization

The task of the supervised learning is to learn from a set of labeled examples, called the training data, a function to predict accurately unseen examples, called the test data. The training data set $\{x_i, y_i\}_{i=1}^n$ consists of n ordered pairs of $x_i \in \mathcal{X} \subset \mathbb{R}^{\mathrm{d}}$ and $y_i \in \mathcal{Y} \subset \mathbb{R}$, which are respectively a vector of measurements of a single example and its label. The test data is another set $\{x_j\}_{j=1}^m$ that need to be labeled with high accuracy based on certain measures.

The fundamental assumption of supervised learning is that the training and test data are independently and identically generated from an unknown but fixed probability distribution $P(x, y)$. This assumption implies that the

training and the test data are related and the observations in the training data carry the information about the targeted test data probability distribution.

Let $l(f(x), y)$ denote the function that measures the disagreement between the prediction $f(x)$ of an example and its real outcome y. We also call $l(f(x), y)$ the loss function since it represents the loss or the cost of predicting $f(x)$ when the true value is y. The choice of the loss function depends largely on the learning problem being solved.
For the regression problem, a typical choice is the squared loss

$$l(f(x), y) = (y - f(x))^2.$$

For the classification problem, one could choose the 0-1 loss

$$l(f(x), y) = 1 - \mathbb{E}_{f(X)=Y} = \begin{cases} 0 & \text{if } f(x) = y \\ 1 & \text{otherwise .} \end{cases}$$

The expected loss of a prediction function $f(x)$ over the generative distribution $P(x, y)$ is called the generalization error or the risk and defined as:

$$R(f) = E_p[l(f(X), Y)] \tag{1.1}$$

$$= \int_{\mathcal{X}} \int_{\mathcal{Y}} l(f(x), y) P(x, y) dy dx. \tag{1.2}$$

2 Optimal Prediction and Risk Minimization

The theoretical optimal prediction, or Bayes optimal prediction, is the function that minimizes $R(f)$ and is given by:

$$f^* = \operatorname{argmin}_f E_p[l(f(X), Y)] \tag{1.3}$$

$$= \operatorname{argmin}_f \int_{\mathcal{X}} \int_{\mathcal{Y}} l(f(x), y) P(x, y) dy dx. \tag{1.4}$$

The Bayes optimal risk achieved by the Bayes optimal prediction is then:

$$R(f^*) = \int_{\mathcal{X}} \int_{\mathcal{Y}} l(f^*(x), y) P(x, y) dy dx.$$

For example, if we use the square loss function, $R(f) = E_p[(Y - f(X))^2]$, then the Bayes optimal prediction and the Bayes optimal risk are:

$$\begin{aligned} f^* &= \operatorname*{argmin}_f E_p[(Y - f(X))^2] \\ &= \int_{\mathcal{Y}} yP(y|x)dy \\ &= E_p[Y|X], \end{aligned}$$

$$R(f^*) = \int_{\mathcal{X}} \int_{\mathcal{Y}} l(E_p[Y|x], y)P(x, y)dydx.$$

3 Generative Learning

Finding directly the Bayes optimal prediction of an examples $f^*(x)$ using its definition is unfeasible when the joint probability distribution $P(x, y)$ is unknown. Alternatively, supervised learning uses the principle of induction to infer $f^*(x)$ from a given set of labeled training data $\{x_i, y_i\}_{i=1}^n$. In generative-based approaches of supervised learning, the main idea is to approximate the generative distribution $P(x, y)$ to a family of probability distributions using the training data and then using its approximation to infer the prediction function.

3.1 Bayesian Inference

In Bayesian inference (Box and Tiao [2011]; MacKay [1992]), a family of probability distributions, $P_M(x, y|\theta)$ is specified to approximate the data distribution $P(x, y)$. Given the training data $\{x_i, y_i\}_{i=1}^n$ and a prior distribution $q(\theta)$, the posterior distribution of the parameter θ is estimated using Bayes Theorem as following:

$$P_M(\theta|\{x_i, y_i\}_{i=1}^n, q) = \frac{q(\theta)\prod_{i=1}^n P_M(x_i, y_i|\theta)}{\int q(\theta)\prod_{i=1}^n P_M(x_i, y_i|\theta)d\theta}.$$

The posterior generative distribution is then:

$$P_M(x, y|\{x_i, y_i\}_{i=1}^n, q) = \int P_M(x, y|\theta)P_M(\theta|\{x_i, y_i\}_{i=1}^n, q)d\theta. \tag{1.5}$$

By integrating over θ, we are integrating over all the probability density function in the model. The computation of the posterior distribution of parameter and posterior generative distribution relies on another layer of approximation using Markov chain Monte Carlo methods (Green [1995]).
Consequently, we obtain the Bayesian prediction function by substituting the posterior distribution above for the unknown conditional distribution $P(y|x)$ in Equation 1.4:

$$f_B(x) = \underset{\hat{y}}{\operatorname{argmin}} \int_{\mathcal{Y}} l(\hat{y}, y) P_M(y|x, \{x_i, y_i\}_{i=1}^n) dy \tag{1.6}$$

where

$$\begin{aligned} P_M(y|x, \{x_i, y_i\}_{i=1}^n, q) &= \frac{P_M(x, y|\{x_i, y_i\}_{i=1}^n, q)}{P_M(x|\{x_i, y_i\}_{i=1}^n, q)} \\ &= \frac{P_M(x, y|\{x_i, y_i\}_{i=1}^n, q)}{\int_y P_M(x, y|\{x_i, y_i\}_{i=1}^n, q) dy}. \end{aligned}$$

3.2 Maximum a Posteriori

The integral in Equation 1.5 is not easily estimated and usually relies on another layer of approximation. Alternatively Bayesian inference is often approximated by Maximum a Posteriori (MAP) (Sorenson [1980]). The premise of MAP is the same as Bayesian framework. We first specify a family of probability distributions, $P_M(x, y|\theta)$ with their prior probability $q(\theta)$, to approximate the data distribution $P(x, y)$. Given the training data $\{x_i, y_i\}_{i=1}^n$ and a prior distribution $q(\theta)$, the posterior distribution of parameter θ is also estimated using Bayes Theorem as following:

$$P_M(\theta|\{x_i, y_i\}_{i=1}^n, q) = \frac{q(\theta) \prod_{i=1}^n P_M(x_i, y_i|\theta)}{\int q(\theta) \prod_{i=1}^n P_M(x_i, y_i|\theta) d\theta}.$$

The MAP posterior generative distribution is then selected to be the single distribution with the highest posterior probability, $P_M(x, y|\theta_{MAP})$, where

$$\theta_{MAP} = \underset{\theta}{\operatorname{argmax}} P_M(\theta|\{x_i, y_i\}_{i=1}^n, q).$$

Finally, we obtain the MAP prediction function by substituting the posterior

distribution $P_M(y|x, \theta_{MAP})$ for the unknown conditional distribution $P(y|x)$ in Equation1.4:

$$f_{MAP}(x) = \underset{\hat{y}}{\operatorname{argmin}} \int_{y} l(\hat{y}, y) P_M(y|x, \theta_{MAP}) dy \tag{1.7}$$

where

$$\begin{aligned} P_M(y|x, \theta_{MAP}) &= \frac{P_M(x, y|\theta_{MAP})}{P_M(x|\theta_{MAP})} \\ &= \frac{P_M(x, y|\theta_{MAP})}{\int_y P_M(x, y|\theta_{MAP}) dy}. \end{aligned}$$

3.3 Maximum Likelihood

The maximum likelihood (ML) is based on selecting a distribution with the highest likelihood given the data. We first specify a family of probability distributions, $P_M(x, y|\theta)$ to approximate the data distribution $P(x, y)$. Given the training data $\{x_i, y_i\}_{i=1}^n$, the likelihood of parameter θ is:

$$P_M(\{x_i, y_i\}_{i=1}^n|\theta) = \prod_{i=1}^{n} P_M(x_i, y_i|\theta).$$

Since maximizing the logarithm of the likelihood above is easier to compute and results in the same maximizer, we write:

$$log(P_M(\{x_i, y_i\}_{i=1}^n|\theta)) = \sum_{i=1}^{n} log(P_M(x_i, y_i|\theta)).$$

We then select the generative distribution that maximizes the logarithm of the likelihood to approximate the data generating distribution.

$$\theta_{ML} = \underset{\theta}{\operatorname{argmax}}\, log(P_M(\{x_i, y_i\}_{i=1}^n|\theta)).$$

Finally, we obtain the ML prediction function by substituting the posterior distribution $P_M(y|x, \theta_{ML})$ for the unknown conditional distribution $P(y|x)$ in

Equation 1.4:

$$f_{ML}(x) = \underset{\hat{y}}{\operatorname{argmin}} \int_{y} l(\hat{y}, y) P_M(y|x, \theta_{ML}) dy \tag{1.8}$$

where

$$\begin{aligned} P_M(y|x, \theta_{ML}) &= \frac{P_M(x, y|\theta_{ML})}{P_M(x|\theta_{ML})} \\ &= \frac{P_M(x, y|\theta_{ML})}{\int_y P_M(x, y|\theta_{ML}) dy}. \end{aligned}$$

4 Discriminative Learning

Generative learning produces a probability distribution over all input and output variables and manipulates it to compute prediction functions. The disadvantage of generative learning is that searching for a probability density distribution is a hard problem particularly in high dimension while the objective of may learning problems is just to predict the output.

Alternatively, discriminative learning, also called direct function approximation, directly attempts to estimate the input to output mappings without modeling the generative distributions. Given a loss function, discriminative learning tries to minimize the corresponding risk $R(f)$ with the optimal prediction function $f^*(x)$. Given n training data, $R_n(f)$, called training error or empirical risk and defined by

$$R_n(f) = \frac{1}{n} \sum_{i=1}^{n} l(f(x_i), y_i)$$

is an unbiased estimator of $R(f)$.

If the learning goal is only to find a prediction function that yields the smallest loss as possible, the prevailing consensus is that direct function approximation is always to be preferred to generative approach. The most compelling reason is that "one should solve the problem directly and never solve a more general problem as intermediate step" [Vapnik [1998]].

Given a learning problem with infinite input space and a finite number of training examples, if the probability distribution of the input is continuous,

there exists a prediction rule $\hat{f}$ among all possible functions that minimizes the training error to 0 but maximizes the generalization error to 1. This situation is called overfitting in literature. There are two principle methods to deal with this problem. The first one is to pre-define a model or a hypothesis space $\mathcal{H}$ of some possible functions, where the minimization of training error is performed

$$\hat{f} = \operatorname*{argmin}_{f \in \mathcal{H}} R_n(f).$$

This approach, called **Empirical Risk Minimization** (ERM), works best when the domain knowledge about a specific learning problem is sufficient to narrow down the searching range of the target function to a small set of possible functions $\mathcal{H}$.

However, in practical problems of machine learning, the family of the target function is usually unknown. In such case, we start with a small hypothesis space $\mathcal{H}_1$ and extend it gradually through an infinite increasing sequence $\{\mathcal{H}_d\}_{d=1}^{\infty}$, where $\mathcal{H}_d \subset \mathcal{H}_{d+1}$ for any d [1]. This second approach is called **Structural Risk Minimization**. The empirical risk minimization is performed on each $\mathcal{H}_d$ and we select the model in the sequence whose sum of empirical risk and penalty for its complexity is minimal

$$\hat{f} = \arg\min_{f \in \mathcal{H}, \in \mathbb{N}} R_n(f) + \lambda J(d, n),$$

where $J(d, n)$ denotes the complexity measure of $\mathcal{H}_d$ and λ is the regularization coefficient which allows choosing the trade-off between training error and complexity.

5 Learning Bounds

Given the frameworks we presented, this section presents the learning theory that justifies why we can carry optimization on the training data and expect a certain level of generalization to test data. A partial list of textbooks, surveys, and articles on statistical learning theory includes Devroye et al. [2013]; Kearns

[1] The choice of the sequence $\{\mathcal{H}_d\}_{d=1}^{\infty}$ comes from a domain knowledge of each specific problem under study and non of them is universally optimal. The necessity of domain knowledge is formally stated in what is called *No Free Lunch Theorem* (Wolpert [1996]).

and Vazirani [1994]; Mendelson [2003]; Vapnik [2013, 1998]

It worth mentioning that thanks to the law of large numbers, the training error almost surely converses, as the training sample size n approaches infinity, to the generalization error $R(f)$. However, in real application, n is a finite number. The analysis below quantifies how close the training and the generalization errors are in that situation.

5.1 Hoeffding's Inequality and Generalization Error Bound of a Single Function

Given a prediction function f, we rewrite the different between its generalization error $R(f)$, which need to be estimated, and the training error $R_n(f)$, which is accessible from the training data, as follows:

$$R(f) - R_n(f) = E_p[l(f(X), Y)] - \frac{1}{n}\sum_{i=1}^{n} l(f(x_i), y_i).$$

By the law of large number, convergence of the training error of a function f to its risk immediately yields:

$$P\left[\lim_{n\to\infty}\left(E_p[l(f(X), Y)] - \frac{1}{n}\sum_{i=1}^{n} l(f(x_i), y_i)\right) = 0\right] = 1.$$

When the training sample size is not infinite and the loss function is bounded, Hoeffding's Inequality quantifies how close the training error of a function approaches its risk.

Theorem 1 ***(Hoeffding).*** *Let $\{X_i, Y_i\}_{i=1}^n$ be n i.i.d. random variables with $l(Y_i, f(X_i) \in [a, b]$. Then for all $\epsilon > 0$, we have*

$$P\left[\left|E_p[l(f(X), Y)] - \frac{1}{n}\sum_{i=1}^{n} l(f(x_i), y_i)\right| > \epsilon\right] \leq 2\exp\left(-\frac{2n\epsilon^2}{(b-a)^2}\right).$$

Denote the right hand side of the above inequality by δ and only consider the binary classification problem with 0-1 loss function[1], we have $b - a = 1$,

[1]The result we obtain here generalizes well to other problems, including regression, multi-class classification, and binary classification with different loss function.

$\delta = 2\exp(-2n\epsilon^2)$, and $\epsilon = \sqrt{\frac{log\frac{2}{\delta}}{2n}}$.
The Hoeffding inequality becomes

$$P\left[|R(f) - R_n(f)| > \epsilon\right] \leq \delta.$$

Subtracting both sides of the inequality from 1, we find that for any function f and any $\delta > 0$, with probability at least $1 - \delta$,

$$|R(f) - R_n(f)| \leq \epsilon. \tag{1.9}$$

5.2 Uniform Convergence of Finite $\mathcal{H}$

The bound obtained in previous section is applied only to one specific function $f \in \mathcal{H}$ that is chosen before the training data is seen. However in supervised learning, we normally start will a set $\mathcal{H}$ of more than one functions and then choose one of them, which is $\hat{f}$ in ERM framework, after seeing the data. A useful bound should hold simultaneously for all $f \in \mathcal{H}$.

Given a finite hypothesis space $\mathcal{H} = \{f_i\}_{i=1}^N$. Given a function $f_i \in \mathcal{H}$, we define a corresponding set of examples

$$C_i = \{(x_j, y_j)_{j=1}^n : R(f_i) - R_n(f_i) > \epsilon\}$$

where the ϵ-bound fails. Hoeffding's inequality imposes that the probability measure of this set must be small, so

$$\forall i : P(C_i) \leq \delta.$$

Using the union bound we obtain

$$\bigcup_{i=1}^{N} P(C_i) \leq \sum_{i=1}^{N} P(C_i) \leq N\delta.$$

We can write

$$P[\exists f \in \mathcal{H} : R(f) - R_n(f) > \epsilon] = \bigcup_{i=1}^{N} P(C_i) \leq N2\exp(-2n\epsilon^2).$$

As a result, given a finite set of N function $\mathcal{H}$, for any $\delta \in (0, 1]$, with probability at least $1 - \delta$, the following bound holds

$$\sup_{f \in \mathcal{H}} |R(f) - R_n(f)| \leq \sqrt{\frac{log(\frac{2N}{\delta})}{2n}}. \tag{1.10}$$

5.3 Estimation Error

As stated earlier, we use the minimizer of training error $\hat{f} = \text{argmin}_{f \in \mathcal{H}} R_n(f)$ to perform prediction on the test data. Therefore it is more interesting to derive a generalization error bound of this function.

Denoting the best possible hypothesis in $\mathcal{H}$ as $f' = \text{argmin}_{f \in \mathcal{H}} R(f)$, the estimation error $R(\hat{f}) - R(f')$ is bounded by

$$\begin{aligned} R(\hat{f}) - R(f') &= [R(\hat{f}) - R_n(\hat{f})] + [R_n(\hat{f}) - R_n(f')] + [R_n(f') - R(f')] \\ &\leq \sup_{f \in \mathcal{H}} |R(f) - R_n(f)| + 0 + \sup_{f \in \mathcal{H}} |R(f) - R_n(f)| \\ &\leq 2 \sup_{f \in \mathcal{H}} |R(f) - R_n(f)|. \end{aligned}$$

This means that when the training error converges uniformly to the generalization error, the output $\hat{f}$ of the learning algorithm has a generalization error close to that of the best possible hypothesis in $\mathcal{H}$. The distance bounded by

$$2 \sup_{f \in \mathcal{H}} |R(f) - R_n(f)|.$$

We put this result together with 1.10 into a theorem.

Theorem 2 *Given a hypothesis space $\mathcal{H}$ with N elements, n training examples, and a fixed possitive δ, with probability at least $1 - \delta$, we have that*

$$R(\hat{f}) - \min_{f \in \mathcal{H}} R(f) \leq 2\sqrt{\frac{log(\frac{2N}{\delta})}{2n}}.$$

5.4 Uniform Convergence of Infinite $\mathcal{H}$

When $\mathcal{H}$ has infinite number of elements, the complexity of $\mathcal{H}$ cannot be measured by a simple counting. Vapnik [1998] extended the learning bound and convergence above to the case of infinite $\mathcal{H}$ by introducing the Vapnik-Chervonenkis (VC) dimension which measure complexity of infinite hypothesis spaces.

The VC dimension of a hypothesis space $\mathcal{H}$, denoted $VC(\mathcal{H})$ is the size d of the largest set $S = \{x_i \in \mathcal{X} : i = 1, ...d\}$ such that for all label set $L = \{y_i \in \mathcal{Y} : i = 1, ...d\}$, there exists some $f \in \mathcal{H}$ that classifies all examples in S correctly according to L, i.e. $f(x_i) = y_i$ for all $i = 1, ...d$. For example, consider the hypothesis space $\mathcal{H}$ of all haft-planes in two dimensions. $\mathcal{H}$ can shatter some set of three points like one in Figure 1.1a. All eight possible ways to label these points are listed in Figure 1.1b-i and each one can be perfectly classified by a haft-plane. On the other hand, for any set of four points, we can always find labeling for these points like in Figure 1.1j, for example, such that no haft-plan can classify them without error. Therefore, the size of the largest set that the hypothesis space $\mathcal{H}$ of all haft-planes in two dimensions can shatter is $VC(\mathcal{H}) = 3$.

It turns out that VC-dimension can be used to provide the uniform convergence of training error by following result due to Vapnik, which is seen by many to be the most important theorem in learning theory.

Theorem 3 *(Vapnik [1998]) Given an infinite hypothesis space $\mathcal{H}$ with a finite VC-dimension, for any $\delta \in (0, 1]$, with probability at least $1 - \delta$, we have the following bounds:*

$$\sup_{f \in \mathcal{H}} |R(f) - R_n(f)| \leq O\left(\sqrt{\frac{1}{n}\left(VC(\mathcal{H})log(\frac{n}{VC(\mathcal{H})}) + log(\frac{1}{\delta})\right)}\right) \tag{1.11}$$

and

$$R(\hat{f}) - \min_{f \in \mathcal{H}} R(f) \leq O\left(\sqrt{\frac{1}{n}\left(VC(\mathcal{H})log(\frac{n}{VC(\mathcal{H})}) + log(\frac{1}{\delta})\right)}\right).$$

With Theorem 2 and 3, we can estimate the minimum training samples size

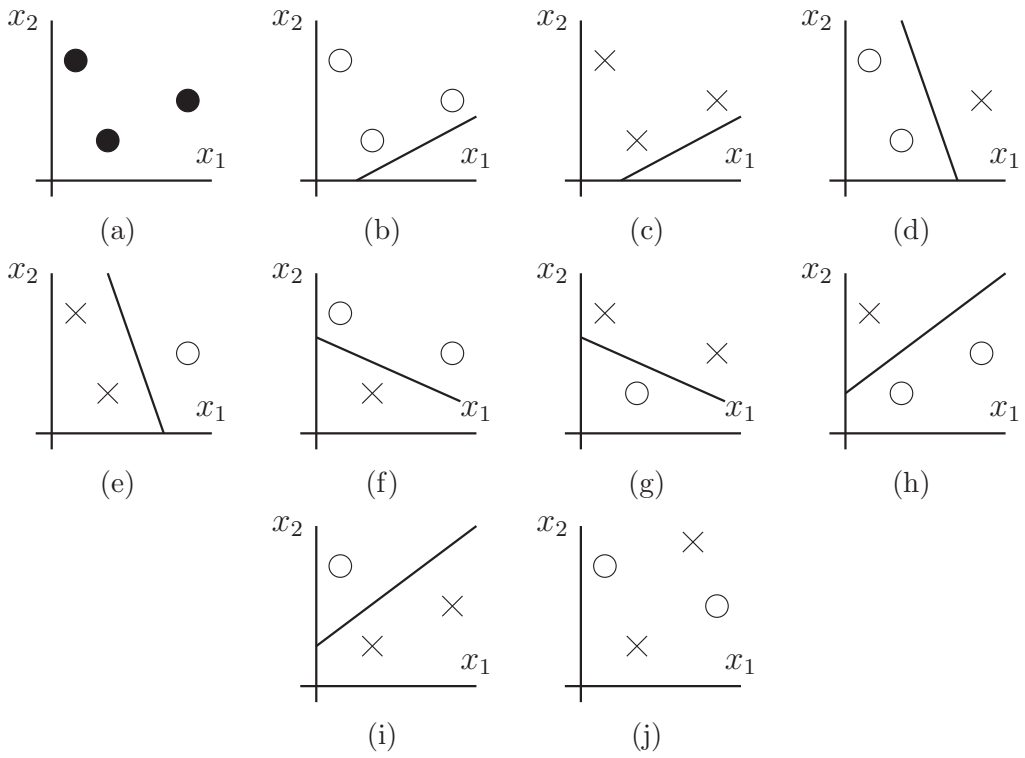


Figure 1.1: VC dimension of haft-planes in $\mathbb{R}^2$: a)Three original points; b, c, d, e, f; g; h; i : all eight possible labeling sets of the original three points can be shattered by a haft-plane; j) For any set of 4 points, there exits a labeling set that cannot be shattered by any haft-plane.

$M(\epsilon, \delta)$ that is necessary to bound the estimation error to be with a certain accuracy ϵ and with a certain confidence level $1 - \delta$. The quantity $M(\epsilon, \delta)$ is known as the sample complexity and defined formally as following.

Definition 1 *(Sample Complexity) For all $\epsilon, \delta \in (0, 1)$, a hypothesis space $\mathcal{H}$ is said to have a sample complexity $M(\epsilon, \delta)$ if it is the smallest sample size for which there exists an algorithm $\mathcal{A}$ that for all distribution P over $\mathcal{X} \times \mathcal{Y}$, $\mathcal{H}$ outputs a model $\hat{f} \in \mathbb{H}$, depending on training data, so that with probability $1 - \delta$:*

$$R(\hat{f}) - \min_{f \in \mathcal{H}} R(f) \leq \epsilon.$$

From Theorem 2 and 3, Blumer et al. [1986, 1989, 1990] derived an upper bound for the sample complexity of a hypothesis space $\mathcal{H}$ as following.

Corollary 4 *Given hypothesis space $\mathcal{H}$ and $0 < \delta, \epsilon < 1$, then*

- *The sample complexity of $\mathcal{H}$ is*

$$m(\epsilon, \delta) = \mathcal{O}\left(\frac{1}{\epsilon} ln \frac{1}{\delta} + \frac{VC(\mathcal{H})}{\epsilon} ln \frac{1}{\epsilon}\right).$$

- *If $\mathcal{H}$ is finite then the sample complexity of $\mathcal{H}$ is*

$$m(\epsilon, \delta) = \mathcal{O}\left(\frac{1}{\epsilon} ln \frac{1}{\delta} + \frac{|\mathcal{H}|}{\epsilon}\right).$$

6 Approximation-Estimation Error Trade-off

Before discussing model selection based on its complexity, we first revisit the approximation-estimation error trade-off. The bound of the difference between generation error of the output function $\hat{f}$ of an algorithm and the Bayes optimal prediction can be decomposed as

$$R(\hat{f}) - R(f^*) \leq \underbrace{\min_{f \in \mathcal{H}} R(f) - R(f^*)}_{\text{approximation error}} + \underbrace{R(\hat{f}) - \min_{f \in \mathcal{H}} R(f)}_{\text{Estimation error}}.$$

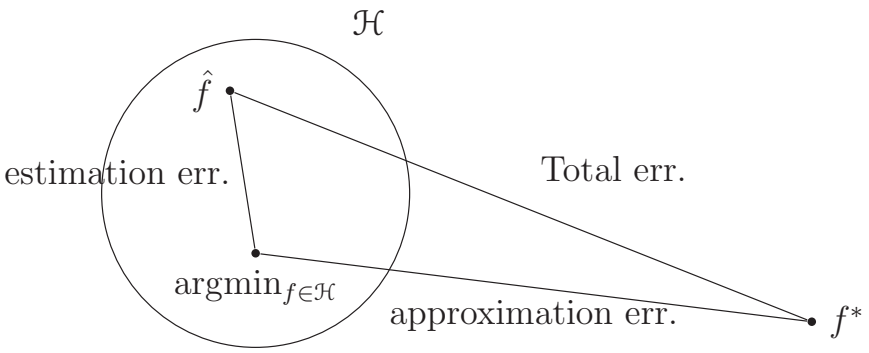


Figure 1.2: Decomposition of generalization error into approximation and estimation errors

The approximation error is normally unknown and depends only on the choice of $\mathcal{H}$ while the estimation error is quantifiable and and depend on the size of $\mathcal{H}$ as shown previously. The decomposition of generalization into approximation and estimation error is illustrated in Figure 1.2.

Suppose that we have two candidate hypothesis spaces $\mathcal{H}_1$ and $\mathcal{H}_2$ where $\mathcal{H}_1 \subset \mathcal{H}_2$. If we use $\mathcal{H}_2$, we can guarantee to achieve a better approximation (since $\min_{f\in\mathcal{H}} R(f) \leq \min_{f\in\mathcal{H}} R(f)$), at the expense of an increase of the sample complexity of the hypothesis space, which in turn increases the estimation error. Conversely, if we use $\mathcal{H}_1$, the estimation error is decreased while approximation error can only increase. This problem in commonly called bias-variance dilemma in literature: bias (or approximation error) and variance (estimation error) cannot be reduced at the same time.

7 Model Specification

In Empirical Risk Minimization framework, the approximation and estimation error are fixed because we specify a model $\mathcal{H}$ before seeing training data. This framework works well in practice if we have a decent domain knowledge to fix a model $\mathcal{H}$ that likely to contains the optimal model f^* or at least some model that closely approximate f^*. However, that's not always the case in practice where domain knowledge is not always enough to specify a useful model. An alternative is Structural Risk Minimization (SRM) method in which the learning algorithm is allowed to make the choice whether to move from one hy-

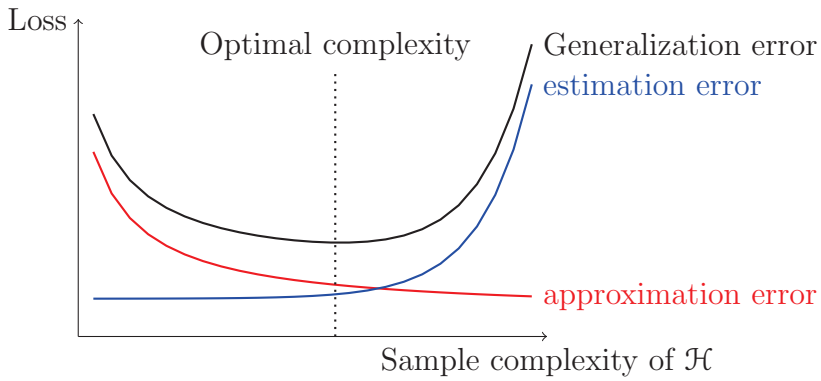


Figure 1.3: Illustration of approximation-estimation error trade-off. Increasing sample complex of the hypothesis space reduces approximation error but increase estimation error at the same time. Optimal generalization error is obtained at some complexity that usually neither optimizes estimation error nor approximation error.

pothesis space $\mathcal{H}_1$ to a more complex hypothesis space $\mathcal{H}_2$ depends on whether the reduction in approximation error is enough to justify for the increase in model complexity. The compromise of estimation error and approximation error is shown in Figure 1.3. At the optimal complexity, which minimizes the the generalization error, the approximation error is typically a strictly positive number. It means that in order to achieve optimal generalization error, we normally accept some approximation error and stop increasing sample complexity of the hypothesis space even when it has not included the universally optimal model f^*. This problem is called model misspecification and plays an importance role in certain types of selection bias. We define it formally as below.

Definition 2 *$\mathcal{H}$ is said to be well-specified if there exist some $\hat{f} \in \mathcal{H}$ such that $R(\hat{f}) - R(f^*) = 0$. Otherwise, $\mathcal{H}$ is said to be misspecified.*

An example of model misspecification is when we use linear regression while the underlying data generating function $P(y|x)$ is non-linear. Besides the reason of optimizing the approximation-estimation error trade-off as we discuss above, a simpler model is preferred to a more complicated one because the former is usually more transparent than the later. Model transparency, which facilitates interpretability, is a fundamentally desirable property in many research areas like biology, medical study, linguistics, or social science.

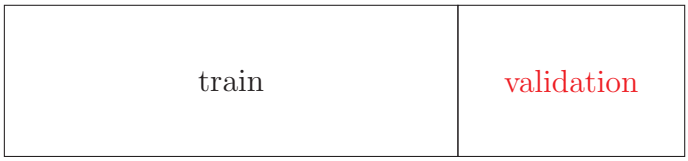


Figure 1.4: Illustration of data partition for holdout validation validation.

8 Empirical Accuracy Estimation

Even though model complexity provides a well-justified guidance to eliminate models that are either too complex or too simple, the model selection and parameter optimization processes still require estimating the accuracy of a prediction function induced by learning algorithms. Besides, accuracy estimation also predicts future performance of a prediction function. There are several possible empirical accuracy estimation methods including holdout validation and cross validation.

8.1 Holdout Validation

The available data set is partitioned, as illustrated in Figure 1.4, into a training set $D_T = \{x_i, y_i\}_{i=1}^{n_T}$ and a holdout validation set $D_V = \{x_i, y_i\}_{i=1}^{n_V}$, which is not to be used in training or parameter optimization process. The prediction function $\hat{f}$ is learned on the training data and evaluated on the validation set. The validation loss of $\hat{f}$ is defined as:

$$R_V(\hat{f}) = \frac{1}{n_V} \sum_{i=1}^{n_V} l(\hat{f}(x_i), y_i) \tag{1.12}$$

The holdout validation loss provides the most straightforward and unbiased estimator of the generalization error of $\hat{f}$ but it reduces the sample size of training data. If we have enough data we can assign a large holdout set to reduce the variance of validation loss while keeping a sufficient training data set. However, data are often scare, a more effective approach to make use the available data is desirable.

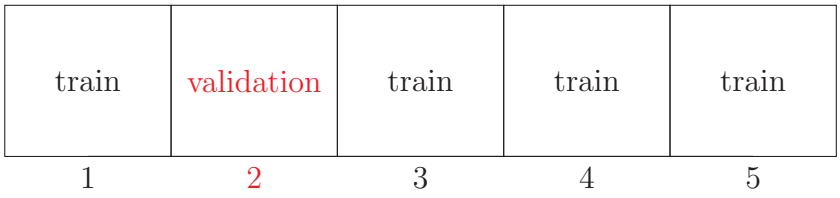


Figure 1.5: Illustrative example of data partition for cross validation when $K = 5$ and the second fold is used as validation set.

8.2 Cross Validation

An alternative to holdout validation when the training data are not massively available is cross validation (Stone [1974]; Wahba [1990]). CV has been shown to give a nearly unbiased estimator of the generalization error with finite sample (Smola and Schölkopf [1998]). In K-fold cross-validation, the training sample D is partitioned into K mutually exclusive and roughly equal-sized subsets $D_1, D_2, ..., D_K$, as illustrated in Figure 1.5 for the case $K = 5$. For each $k \in 1..K$, prediction accuracy of the function $\hat{f}_k$ that is constructed based on the training set $\cup_{i \neq k} D_i$ is evaluated on the corresponding validation set D_k. Let $\kappa : 1, ..., N \rightarrow 1, ..., K$ be an indexing function that maps an example to its randomly allocated partition, the cross validation estimate of the generalization error is

$$CV(\hat{f}, K) = \frac{1}{n} \sum_{i=1}^{n} l(\hat{f}_{\kappa(i)}(x_i), y_i) \tag{1.13}$$

Typical choices of K are 5, 10 and n. In leave-one-out cross validation (LOOCV), i.e. $K = n$, the CV gives an approximately unbiased estimator of the generalization error but can have a high variance because any two training set are only different by one examples. On the other hand, when k is small, CV has a lower variance since each training set is quite different from the other but it may overestimate the generalization error. The extend of the overestimation depends on where the how the performance of the learning method varies with the training sample size. Overall, Breiman and Spector [1992]; Kohavi et al. [1995] recommend $K = 5$ or $K = 10$ as good compromise for the bias and variance of the estimation.

Chapter 2

Correcting Selection Bias with Importance Weighting Framework

As discussed in Section 1 of Chapter 1, the assumption that the training and the test samples are drawn from the same probability distribution plays a crucial role in the theory of supervised learning. Unfortunately, this essential assumption is often violated in the presence of selection bias. Under such condition, the learning frameworks presented in Chapter 1 need to be adjusted to remain valid.

In this chapter we first define some useful terminologies and the classification of selection bias. We then introduce the importance weighting framework for the generative and discriminative learning. Given the importance weight, the adaptation of generative learning methods to the importance weighting is very straight forward. We can approximate the generative distribution of the training data to a family of probability distributions using the training data and then adjust it by the importance weight to obtain an approximation of the test distribution before inferring the prediction function. On the other hand, the adaptation of discriminative learning to the selection bias requires more complication. We introduce two methods of using the importance weight to correct selection bias in discriminative learning: one with sampling and the other with modification of the loss function. We then show that the importance weighted cross validation gives an almost unbiased estimate of

the generalization error. We review covariate shift problem and two common methods for learning the importance weight from the training data and a set of unlabeled examples. We also investigate the potential of Bayesian Network to encode researcher's a priori assumption about the relationship between variables, including selection variable, and to infer independence and conditional independence relationships that allow selection bias to be corrected. In the experimentation section, we assess the ability of the importance weighting to remove the complete selection bias based on the independence and conditional independence relationships read from Bayesian Network. A part of this chapter has been presented at ICONIP2015 conference (Tran and Aussem [2015b]).

1 Terminology and Categorization

Selection bias, also termed dataset shift or domain adaptation in the literature Candela et al. [2009]; Moreno-Torres et al. [2012b], occurs when the training distribution $P_{tr}(x, y)$ and the test distribution $P_{te}(x, y)$ are different. It is pervasive in almost all empirical studies, including Machine Learning, Statistics, Social Sciences, Economics, Bioinformatics, Biostatistics, Epidemiology, Medicine, etc. Selection bias is prevalent in many real-world machine learning problems because the common assumption in machine learning is that the training and the test data are drawn independently and identically from the same distribution. The term "domain adaptation" is used when one builds a model from some fixed source domain, but wishes to deploy it across one or more different target domains. The term "selection bias" is slightly more specific as it assumes implicitly that there exists a binary variable S that controls the selection of examples in the training set, in other words we only have access to the examples that have $S = 1$. The use of selection variable S to represent structural assumptions about how the sampling mechanism is related to other variables appears frequently in many selection bias researches, e.g. Cooper [1995]; Cox and Wermuth [1996]; Didelez et al. [2010]; Geneletti et al. [2009]. To be specific, we assume that there exist a probability distribution $P(x, y, s)$, where training data are sampled from

$$P_{tr}(x, y) = P(x, y|s = 1)$$

while test data are sampled from

$$P_{te}(x, y) = \sum_{s} P(x, y, s) = P(x, y).$$

The existence of the selection variable S also distinguishes selection bias from other sub-fields of domain adaptation. In selection bias, we can see that the support of the test data distribution always contains that of the training data distribution and if $P(s = 1|x, y) > 0$ for all x and y, then the two supports overlap each other. On the contrary, in most of other domain adaptation problems, the two supports can freely have non-overlapping region or even be completely disjointed in extreme cases. In place of the selection variable, other domain adaption methods usually assume the existence of some domain-invariant representations which allows the training distribution to be able to match with the test distribution after some transformations or alignments (Courty et al. [2016]; Fernando et al. [2014]; Sun et al. [2016]). Other domain adaptation methods assume the access to some labeled data with full feature vector from test distribution (Ben-David et al. [2010]; Daumé III [2009]).

The existence of this variable S allows modeling expert knowledge about selection process in a causal sense using graphical model as will be shown in later section. There are several cases worth considering regarding the dependence structure between X, Y, and S (Fan and Davidson [2007]; Moreno-Torres et al. [2012a]; Zadrozny [2004]):

1. If $S \perp\!\!\!\perp X$ and $S \perp\!\!\!\perp Y$, the selected sample is not biased, that is, the examples (x, y, s) which have $S = 1$ constitute a random sample from the general distribution $P(x, y)$. In this case, the i.i.d assumption is satisfied, all theoretical results presented in previous section holds true without any adjustment needed.

2. Covariate shift: $S \perp\!\!\!\perp Y|X$, the selected sample is biased but the biasedness only depends on the feature vector X. This case is also termed *sample bias* and corresponds to a change in the prior probabilities of the features. This type of bias has been extensively studied in machine learning literature and there are methods for correcting it Ben-david et al. [2007]; Bickel et al. [2009]; Blitzer et al. [2008]; Cortes et al. [2010];

Dudík et al. [2005]; Huang et al. [2006]; Kanamori et al. [2009, 2012]; Shimodaira [2000]; Sugiyama and Kawanabe [2012]; Sugiyama et al. [2007b]; Yu and Szepesvári [2012]; Zadrozny [2004].

3. Prior probability shift: $S \perp\!\!\!\perp X|Y$, the selected sample is biased but the biasedness depends only on the label Y. This case is also termed *label bias* and corresponds to a change in the prior probabilities of the labels. This type of bias has been studied in machine learning literature and there are methods for correcting it Elkan [2001]; Ting [2002].

4. If no independence assumption holds between X, Y, and S. This is termed *complete selection bias* in the literature. The selected sample is biased and we cannot hope to learn a mapping from features to labels using the selected sample, unless we have some additional information on the mechanism by which the samples were preferentially selected to the data set as will see.

2 Learning under Selection Bias with Importance Weighting

In this section, we assume that we know the selection probability distribution $P(s = 1|x, y)$, which fully quantifies the selection mechanism. We first relate the the selection probability to the change of distribution from training to test data by the so-called importance weight. We then show that this importance weight can be used effectively to correct selection bias of all three classed discussed above.

Definition 3 *(**Importance weight**) Given the support of $P_{tr}(x, y)$ contains the support of $P_{te}(x, y)$, i.e. for all $(x, y) \in \mathcal{X} \times \mathcal{Y} : (P_{te}(x, y) > 0 \implies P_{tr}(x, y) > 0)$, the ratio*

$$\beta(x, y) = \frac{P_{te}(x, y)}{P_{tr}(x, y)}$$

is defined over the support of $P_{te}(x, y)$. It quantifies the change of distribution from training to test data and is called the importance weight.

Given selection probability distribution $P(s = 1|x, y)$, if it is positive for all (x, y) in the support of $P(x, y)$, i.e. there is no deterministic exclusion of example, using Bayes' rule

$$\begin{aligned} P(x, y, s) &= P(x, y|s = 1)P(s = 1) \\ &= P(s = 1|x, y)P(x, y), \end{aligned}$$

we can relate the importance weight to the selection distribution as following:

$$\beta(x, y) = \frac{P_{te}(x, y)}{P_{tr}(x, y)} = \frac{P(x, y)}{P(x, y|s = 1)} = \frac{\int_{\mathcal{X}} \int_{\mathcal{Y}} P(s = 1|x', y')dy'dx'}{P(s = 1|x, y)}.$$

The non-deterministic exclusion of example is important for selection bias to be corrected. If there are some instances (x, y) that are always excluded from the training data, i.e. $P(s = 1|x, y) = 0$, learning from training data with selection bias becomes an extrapolation problem, where prediction on excluded examples requires further assumptions or becomes unreliable. In general dataset shift, there may be cases where test data that are never seen in training set but are instead associated with training data by some assumed relationships depends on each specific problem. For example in image processing domain, training images might be taken under certain lighting or equipment conditions, whereas prediction is performed on images taken under different conditions. In these cases, changes from training to test data are usually modeled by some transformations e.g. translation or rotation of the feature vector rather than by the change of data distribution. This is another kind of non-stationary problem where focus is placed on the transformation of data instead of learning model adaptation.

2.1 Importance Weighting for Generative Learning

Given the selection distribution, or equivalently the importance weight, the adaptation of generative learning methods is very straightforward. We can approximate the generative distribution of training data $P_{tr}(x, y)$ to a family of probability distributions using the training data and then adjust it by the importance weight to obtain test distribution, $P_{te}(x, y) = P_{tr}(x, y)\beta(x, y)$, before inferring the prediction function.

Importance Weighting for Bayesian Inference

The training data distribution $P_{tr}(x, y)$ is approximated by a family of probability distributions, $P_M(x, y|\theta)$, specified by θ with prior probability $q(\theta)$. Given the training data, the posterior distribution of parameter θ is estimated using Bayes Theorem as following:

$$P_M(\theta|\{x_i, y_i\}_{i=1}^n) = \frac{q(\theta)\prod_{i=1}^n P_M(x_i, y_i|\theta)}{\int q(\theta)\prod_{i=1}^n P_M(x_i, y_i|\theta)d\theta}.$$

The posterior training distribution is then:

$$P_M(x, y|\{x_i, y_i\}_{i=1}^n) = \int P_M(x, y|\theta)P_M(\theta|\{x_i, y_i\}_{i=1}^n)d\theta.$$

The estimated test distribution is obtained by adjusting the posterior training distribution by the importance weight

$$P_{te,M}(x, y|\{x_i, y_i\}_{i=1}^n) = P_M(x, y|\{x_i, y_i\}_{i=1}^n)\beta(x, y).$$

Consequently, we obtain the Bayesian prediction function by substituting the estimated distribution above for the unknown conditional distribution ($p(y|x)$ in Equation 1.4:

$$f_B(x) = \underset{\hat{y}}{\operatorname{argmin}} \int_{\mathcal{Y}} l(\hat{y}, y)P_{te,M}(y|x, \{x_i, y_i\}_{i=1}^n)dy \tag{2.1}$$

where

$$\begin{aligned} P_{te,M}(y|x, \{x_i, y_i\}_{i=1}^n) &= \frac{P_{te,M}(x, y|\{x_i, y_i\}_{i=1}^n)}{P_{te,M}(x|\{x_i, y_i\}_{i=1}^n)} \\ &= \frac{P_{te,M}(x, y|\{x_i, y_i\}_{i=1}^n)}{\int_{\mathcal{Y}} P_{te,M}(x, y|\{x_i, y_i\}_{i=1}^n)dy} \\ &= \frac{P_M(x, y|\{x_i, y_i\}_{i=1}^n)\beta(x, y)}{\int_{\mathcal{Y}} P_M(x, y|\{x_i, y_i\}_{i=1}^n)\beta(x, y)dy}. \end{aligned}$$

Importance Weighting for Maximum a Posteriori

Again, the training data distribution $P_{tr}(x, y)$ is approximated by a family of probability distributions, $P_M(x, y|\theta)$, specified by θ with prior probability $q(\theta)$.

Given the training data, the posterior distribution of parameter θ is estimated using Bayes Theorem as following:

$$P_M(\theta|\{x_i, y_i\}_{i=1}^n) = \frac{q(\theta)\prod_{i=1}^n P_M(x_i, y_i|\theta)}{\int q(\theta)\prod_{i=1}^n P_M(x_i, y_i|\theta)d\theta}.$$

The posterior training distribution in MAP framework is selected to be the single distribution with highest posterior probability:

$$\theta_{MAP} = \operatorname*{argmax}_{\theta} P_M(\theta|\{x_i, y_i\}_{i=1}^n).$$

The estimated test distribution is obtained by adjusting the posterior training distribution by the importance weight

$$P_{te,M}(x, y|\theta_{MAP}) = P_M(x, y|\theta_{MAP})\beta(x, y).$$

Finally, we obtain the MAP prediction function by substituting the posterior distribution $P_M(y|x, \theta_{MAP})$ for the unknown conditional distribution ($p(y|x)$ in Equation 1.4:

$$f_{MAP}(x) = \operatorname*{argmin}_{\hat{y}} \int_y l(\hat{y}, y)P_{te,M}(y|x, \theta_{MAP})\beta(x, y)dy \tag{2.2}$$

where

$$\begin{aligned} P_{te,M}(y|x, \theta_{MAP}) &= \frac{P_{te,M}(x, y|\theta_{MAP})}{P_{te,M}(x|\theta_{MAP})} \\ &= \frac{P_{te,M}(x, y|\theta_{MAP})}{\int_y P_{te,M}(x, y|\theta_{MAP})dy} \\ &= \frac{P_M(x, y|\theta_{MAP})\beta(x, y)}{\int_y P_M(x, y|\theta_{MAP})\beta(x, y)dy}. \end{aligned}$$

Importance Weighting for Maximum Likelihood

Under selection bias, we first specify a family of probability distributions, $P_M(x, y|\theta)$ to approximate the training data distribution $P_{tr}(x, y)$. Given the

training data $\{x_i, y_i\}_{i=1}^n$, the likelihood of parameter θ and its logarithm is:

$$P_M(\{x_i, y_i\}_{i=1}^n|\theta) = \prod_{i=1}^{n} P_M(x_i, y_i|\theta).$$

$$log(P_M(\{x_i, y_i\}_{i=1}^n|\theta)) = \sum_{i=1}^{n} log(P_M(x_i, y_i|\theta)).$$

We then select the generative distribution that maximizes the logarithm of the likelihood to approximate the training data distribution.

$$\theta_{ML} = \underset{\theta}{\text{argmax}}\, log(P_M(\{x_i, y_i\}_{i=1}^n|\theta)).$$

The estimated test distribution is obtained by adjusting the posterior training distribution by the importance weight

$$P_{te,M}(x, y|\theta ML) = P_M(x, y|\theta_{ML})\beta(x, y).$$

Finally, we obtain the MAP prediction function by substituting the posterior distribution $P_M(y|x, \theta_{ML})$ for the unknown conditional distribution ($p(y|x)$ in Equation1.4:

$$f_{ML}(x) = \underset{\hat{y}}{\text{argmin}} \int_{y} l(\hat{y}, y) P_{te,M}(y|x, \theta_{ML})\beta(x, y)dy \tag{2.3}$$

where

$$\begin{aligned} P_{te,M}(y|x, \theta_{ML}) &= \frac{P_{te,M}(x, y|\theta_{ML})}{P_{te,M}(x|\theta_{ML})} \\ &= \frac{P_{te,M}(x, y|\theta_{ML})}{\int_y P_{te,M}(x, y|\theta_{ML})dy} \\ &= \frac{P_M(x, y|\theta_{ML})\beta(x, y)}{\int_y P_M(x, y|\theta_{ML})\beta(x, y)dy}. \end{aligned}$$

2.2 Importance Weighting for Discriminative Learning

The adaptation of generative learning methods to selection bias problem is fairly simple. One only needs to adjust the approximation of the training data generative distribution by the importance weight before inferring the prediction

function. However, the application of generative learning in practice is very limited. Discriminative learning is often preferred to generative learning when the learning goal is to find a prediction rule with lowest loss as possible. In this section, we introduce two methods of using the importance weight to correct selection bias in discriminative learning: one with sampling and the other with modification of loss function.

2.3 Importance Weighting Using Sub-sampling Methods

Given a training data with selection bias, if we can construct a new data set that follows the general distribution (test data) then we can expect to correct selection bias without having to modify the algorithm. This method, therefore, can work with any algorithms just by changing the training data. In fact, sub-sampling the training data with the importance weight can recover the original unbiased distribution.

Lemma 5 *Given a selection distribution $P(s = 1|x, y)$, and its corresponding importance weight $\beta(x, y)$ if we define a reweighted distribution*

$$\hat{P}(x, y, s) = \beta(x, y)P(x, y, s)$$

then

$$\hat{P}(x, y|s = 1) = P(x, y).$$

Proof

$$\begin{aligned}
\hat{P}(x, y, s = 1) &= P(x, y, s = 1)\beta(x, y) \\
&= P(x, y, s = 1)\frac{P(x, y)}{P(x, y|s = 1)} \\
&= P(s = 1)P(x, y|s = 1)\frac{P(x, y)}{P(x, y|s = 1)} \\
&= P(s = 1)P(x, y).
\end{aligned}$$

Thus, $\hat{P}(x, y, s = 1) = P(x, y)P(s = 1)$. If we sum this expression over x, y we obtain $\hat{P}(s = 1) = P(s = 1)$. Therefore,

$$\begin{aligned}\hat{P}(x, y|s = 1) &= \frac{\hat{P}(x, y, s = 1)}{\hat{P}(s = 1)} \\ &= \frac{P(x, y)P(s = 1)}{P(s = 1)} \\ &= P(x, y). \quad \blacksquare\end{aligned}$$

Lemma 5 states that an unbiased training sample can be obtained by sampling each training example by $\beta(x, y) = \frac{P(x,y)}{P(x,y|s=1)}$. Note however that the support of $P(x, y)$ should be contained in the support of $P(x, y|s = 1)$ for the $\beta(x, y)$ to be always defined. A similar technique applied to covariate shift was discussed in Zadrozny [2004].
The unbiased expected loss of the model follows:

Theorem 6 *Given the weighted distribution $\hat{P}(x, y, s) = \beta(x, y)P(x, y, s)$, for any loss function $l(f(x), y)$, we have:*

$$E_{x,y\sim\hat{P}}[l(f(x), y)|s = 1] = E_{x,y\sim P}[l(f(x), y)].$$

Proof From Lemma 5, we have $\hat{P}(x, y|s = 1) = P(x, y)$. Therefore,

$$\begin{aligned}E_{x,y\sim\hat{P}}[l(f(x), y)|s = 1] &= \int_x \int_y l(f(x), y).\hat{P}(x, y|s = 1)dydx \\ &= \int_x \int_y l(f(x), y).P(x, y)dydx \\ &= E_{x,y\sim P}[l(f(x), y)]. \quad \blacksquare\end{aligned}$$

Using Theorem 6, we can learn or evaluate a model based on examples drawn from weighted distribution $\hat{P}$ without suffering from selection bias. There are two basic sampling methods that allow us represent samples draw from $\hat{P}$: sampling with replacement and acceptance sampling (Von Neumann [1951]). It has been shown in Zadrozny et al. [2003] that the former cre-

ates duplicate examples which may causes severe overfitting while the later achieves a much better performance. In rejection sampling, given the training data set $D = \{x_i, y_i\}_{i=1}^n$ which includes n i.i.d. examples from the distribution $P(x, y|s = 1)$ we include each original training example to a new sub-sampled data set D' with a probability proportional to the importance weight $\beta(x, y) = \frac{P(x,y)}{P(x,y|s=1)}$ to obtain examples follow test distribution $P(x, y)$.
Noting that examples in D are independent and that their acceptance to D' are also independent of each other, we can deduce that D' is an i.i.d sample from $P(x, y)$ thanks to Lemma 5.

To maximize the size of the sub-sampled data set we set the probability of acceptance to

$$P(a = 1|x, y) = \frac{\beta(x, y)}{max_{x,y}(\beta(x, y))}$$

Where a is the acceptance indicator, $a = 1$ when the examples is accepted and $a = 0$ otherwise. We have

$$\begin{aligned}\mathbb{E}_{P_{tr}}[\beta(x, y)] &= \int_x \int_y \frac{P_{tr}(x, y)}{P_{te}(x, y)} P_{tr}(x, y) dy dx \\ &= \int_x \int_y P_{te}(x, y) dy dx = 1.\end{aligned}$$

Hence the expected size of sub-sampled data set is

$$n' = n\frac{\mathbb{E}[\beta(x, y)]}{\max_{x,y}(\beta)} = \frac{n}{\max_{x,y}(\beta)}.$$

This reduction of training sample size only depends on the maximum value of $\beta(x, y)$ and can be significant in some selection bias scheme. The most impacted data are low importance training examples which are expected to be rejected with high probability. To remedy this waste of training data, we can repeatedly sub-sample the original training data, then train many models on the sub-sampled data sets and aggregate the prediction of the models in a manner similar to ensemble learning methods as in Zadrozny et al. [2003]. However, this approach requires an increase of computation cost. On the other hand, most learning algorithms allow us to modify the loss function to compensate for the change of distribution and leave the the original training data unchanged.

2.4 Direct Importance Weighting on Loss Function

Given that most popular learning algorithms can be formulated as empirical risk minimization of a certain loss functions, in this section we present a direct approach of using the importance weight to correct selection bias by modifying the loss functions. We observe that the generalization error of a function can be written as

$$\begin{aligned} R(f) &= E_{P_{te}}[l(f(X), Y)] \\ &= \int_{\mathcal{X}} \int_{\mathcal{Y}} l(f(x), y) P_{te}(x, y) dy dx \\ &= \int_{\mathcal{X}} \int_{\mathcal{Y}} l(f(x), y) \frac{P_{te}(x, y)}{P_{tr}(x, y)} P_{tr}(x, y) dy dx \\ &= E_{P_{tr}}[\beta(x, y) l(f(X), Y)]. \end{aligned}$$

As a result, minimizing the expectation of importance weighted loss over training distribution is equivalent to over test distribution. Each learning algorithm requires a different modification to implement the importance weight but in general we can assume a class of parameterized function class $\mathcal{H} = \{f(x, \theta) : \theta \in \Theta\}$ for the learning task, where

$$\theta = (\theta_1, \theta_2, ..., \theta_m)^T \in \Theta \subset \mathbb{R}^m.$$

In risk minimization frame work, we need to solve

$$\min_{\theta \in \Theta} \sum_{i=1}^{n} l(f(x_i, \theta), y_i) + \lambda J(f) \tag{2.4}$$

where $J(f)$ is a penalty functional and is defined on $\mathcal{H}$.

Under sample reweighting scheme β, it becomes

$$\min_{\theta \in \Theta} \sum_{i=1}^{n} \beta_i l(f(x_i, \theta), y_i) + \lambda J(f). \tag{2.5}$$

Below, we discuss how to minimized this regularized empirical risk in some common settings.

Regularized Least-squares Regression with Kernel Model

Regularized least-squares regression (RLS) is one of the most importance regression methods in machine learning. The Kernel model is defined by

$$f(\cdot, \theta) = \langle \Phi(\cdot), \theta \rangle$$

where $\Phi(\cdot)$ is a feature map from $\mathcal{X}$ to a feature space $\mathcal{F}$ and $\theta \in \mathcal{F}$. The inner product in $\mathcal{F}$ is defined by a kernel function $k(x, x') = \langle \Phi(x), \Phi(x') \rangle$. Some common choices of kernel are:

- Linear kernel
$$k(x_i, x_j) = x_i^T x_j.$$
- Polynomial kernel
$$k(x_i, x_j) = (x_i^T x_j + 1)^d.$$
- Gaussian Kernel
$$k(x_i, x_j) = exp\left(-\frac{||x_i - x_j||^2}{\sigma^2}\right).$$

Using square norm for regularization and square loss function, 2.5 becomes

$$\min_{\theta \in \Theta} \sum_{i=1}^{n} \beta_i (\langle \Phi(x_i), \theta \rangle - y_i)^2 + ||f||^2. \tag{2.6}$$

It can be shown that the solution to 2.6 to be written as (Girosi et al. [1995])

$$f(\cdot, \theta) = \sum_{i=1}^{n} \alpha_i k(\cdot, x_i)$$

We rewrite the regularization term as

$$\begin{aligned}||f||^2 &= \langle f, f\rangle \\ &= \langle \sum_{i=1}^{n} \alpha_i k(\cdot, x_i), \sum_{j=1}^{n} \alpha_j k(\cdot, x_j)\rangle \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j \langle k(\cdot, x_i), k(\cdot, x_j)\rangle \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j k(x_i, y_j)\end{aligned}$$

Let K is kernel matrix, i.e. K is the $n \times n$ matrix with the (i,j)-th element $K_{ij} = k(x_i, y_j)$. Then the regularization term becomes

$$J(f) = \alpha^T K\alpha.$$

For weighted square loss function:

$$\begin{aligned}\sum_{i=1}^{n} \beta_i l(f(x_i, \theta), y_i) &= \sum_{i=1}^{n} (f(x_i, \theta) - y_i)^2 \\ &= \sum_{i=1}^{n} \beta_i \left(\sum_{j=1}^{n} \alpha_j k(x_i, y_j) - y_i\right)^2\end{aligned}$$

Denote by B the diagonal matrix with diagonal $(\beta_1, \beta_2, ..., \beta_n)$, the matrix form of the weighted square loss is then

$$(K\alpha - y)^T B(K\alpha - y).$$

Therefore, 2.5 becomes

$$\min_{\alpha\in\alpha}(K\alpha - y)^T B(K\alpha - y) + \lambda\alpha^T K\alpha. \tag{2.7}$$

The optimization is convex in α, so we can set its gradient with respected to

α to 0 and obtain the solution for α as:

$$\alpha = ((\lambda B^{-1} + K)^{-1} y.$$

As a result we just need to solve a single linear system for α :

$$((\lambda B^{-1} + K)\alpha = y,$$

which can be handled by any available linear system solver.

Support Vector Machine for Classification

A support vector machine (SVM) constructs a separate hyperplane that maximizes the margin between the training data points and the decision boundary Boser et al. [1992]; Scholkopf and Smola [2001]. Although SVM was originally dirived using maximum margin principle, it can be reconstructed under empirical risk minimization framework (Evgeniou et al. [2000]) with hinge loss, which is defined as

$$l(f(x,\theta), y) = [1 - yf(x,\theta)]_+ = max(0, 1 - yf(x,\theta))$$

The weighted empirical risk minimization problem 2.5 becomes

$$\min_{\theta\in\Theta} \sum_{i=1}^{n} \beta_i [1 - y_i f(x_i,\theta)]_+ + \frac{\lambda}{2}||\theta||^2. \tag{2.8}$$

Since $y_i \in \{-1, 1\}$ for classification problem, the formulation above is equivalent to:

$$\min_{\theta,\xi} \frac{1}{2}||\theta||^2 + C\sum_{i=1}^{n} \beta_i \xi_i \tag{2.9}$$

subject to the constraints:

$$y_i f(x_i) \geq 1 - \xi_i,$$

$$\xi \geq 0.$$

which is a quadratic programming problem that is different than the original problem presented in Cortes and Vapnik [1995] only by the importance weight added into the total empirical loss. If we use kernel method and let $k(x_i, y_j)$ denote the kernel that defines the inner product between the feature maps. The dual of 2.9 is

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j k(x_i, y_j) \tag{2.10}$$

subject to

$$0 \leq \alpha_i \leq \beta_i C,$$
$$\sum_{i=1}^{n} \alpha_i y_i = 0.$$

This quadratic programming can be handled by many existing solver such as SVMlight (Joachims [1998]).

Support Vector Machine for Regression

Support vector regression (SVR) (Vapnik [2013]) can be reconstructed under empirical risk minimization framework with ϵinsensitive loss function $|f(x, \theta), y|_\epsilon$ described by

$$|f(x, \theta), y|_\epsilon = \begin{cases} 0 & \text{if } |f(x) - y| \leq \epsilon \\ |f(x) - y| - \epsilon & \text{otherwise.} \end{cases}$$

The weighted empirical risk minimization problem 2.5 becomes

$$\min_{\theta \in \Theta} \sum_{i=1}^{n} \beta_i |f(x, \theta), y|_\epsilon + \frac{\lambda}{2} ||\theta||^2, \tag{2.11}$$

which is equivalent to:

$$\min_{\theta, \xi} \frac{1}{2} ||\theta||^2 + C \sum_{i=1}^{n} \beta_i (\xi_i + \xi_i^*) \tag{2.12}$$

subject to the constraints:

$$y_i - f(x_i) \leq \epsilon + \xi_i,$$

$$-y_i + f(x_i) \leq \epsilon + \xi_i^*,$$

$$\xi \geq 0.$$

If we use kernel method and let $k(x_i, y_j)$ denote the kernel that defines the inner product between the feature maps. The dual of 2.12 is

$$\max_{\alpha} -\frac{1}{2}\sum_{i,j=1}^{n}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)k(x_i, y_j) - \epsilon\sum_{i=1}^{n}(\alpha_i + \alpha_i^*) + \epsilon\sum_{i=1}^{n}(\alpha_i - \alpha_i^*)y_i \quad (2.13)$$

subject to constraints

$$0 \leq \alpha_i, \alpha_i^* \leq \beta_i C,$$

$$\sum_{i=1}^{n} \alpha_i - \alpha_i^* = 0.$$

This again becomes quadratic programming and can be solve by any available solver.

Adaptive Boosting for Classification

Freund and Schapire [1995] Adaptive Boosting (AdaBoost), formulated by Yoav Freund and Robert Schapire (Freund and Schapire [1995]), can be used in conjunction with many weak learner to improve their performance. The outputs of the weak learners are combined into a weighted additive model that represents the final output. The final output of AdaBoost can be written as a linear combination of all the weak learners trained at every stage of the algorithm

$$f(x, \theta) = \sum_{m=1}^{M} \alpha_m a_m(x, \gamma_m)$$

where

- $\theta = \{\alpha_m, \gamma_m\}_{m=1}^{M}$ is the set of model parameters,
- $a_m(x, \gamma_m)$ is output of the weak learner trained at staged m.

Given exponential loss function

$$l(f(x,\theta),y) = exp(yf(x,\theta)),$$

the weighted empirical risk minimization problem is

$$\min_{\{\alpha_m,\gamma_m\}_{m=1}^M} \sum_{i=1}^{n} \beta_i exp(y_i f(x_i, \{\alpha_m,\gamma_m\}_{m=1}^M)). \tag{2.14}$$

This optimization problem is equivalent to the original AdaBoost with initial weighting for example (x_i, y_i) set to β_i.

2.5 Importance Weighted Cross Validation

When selection bias is covariate shift, Sugiyama et al. [2007a] demonstrated that the importance weighted cross validation (IWCV) gives an almost unbiased estimate of the generalization error. In this section, we show that this is true even in the general selection bias setting.

Recall that in K-fold cross-validation, the training sample D is partitioned into K mutually exclusive subsets $D_1, D_2, ..., D_K$, which we assume to be equal size for simplicity. For each $k \in 1..K$, prediction accuracy of the function $\hat{f}_k$ that is constructed based on the training set $D\backslash D_k$ is evaluated on the corresponding validation set D_k. Also let $\kappa : 1, ..., N \to 1, ..., K$ be an indexing function that maps an example to its randomly allocated training partition.

To compensate for the effect of selection bias in cross validation procedure, we modify CV in equation 1.13 so that importance weight is taken into account:

$$IWCV(\hat{f}, K) = \frac{1}{n} \sum_{i=1}^{n} \beta(x_i, y_i) l(\hat{f}_{\kappa(i)}(x_i), y_i) \tag{2.15}$$

The property of IWCV under selection bias is exactly the same with CV in standard learning condition as can be seen below.

Lemma 7 *Given training data D with n examples that can be partitioned into K subset of equal size n/K, $IWCV(\hat{f}, K)$ on biased training data gives an unbiased estimate of the generalization error of the algorithm when it is*

given $n - n/K$ training data, i.e.

$$E_{D\sim P_{tr}}[IWCV(\hat{f}, K)] = R_{n-n/K}(\hat{f}). \tag{2.16}$$

Proof For any example (x_i, y_i) in the training data D, we have

$$\begin{aligned}
&E_{D\sim P_{tr}}[\beta(x_i, y_i) l(\hat{f}_{\kappa(i)}(x_i), y_i)] \\
&\quad = E_{D_{\kappa(i)}\sim P_{tr}} E_{D\backslash D_{\kappa(i)}\sim P_{tr}}[\beta(x_i, y_i) l(\hat{f}_{\kappa(i)}(x_i), y_i)] \\
&\quad = E_{D\backslash D_{\kappa(i)}\sim P_{tr}} \int_{\mathcal{X}\times\mathcal{Y}} [\frac{P_{te}(x_i, y_i)}{P_{tr}(x_i, y_i)} l(\hat{f}_{\kappa(i)}(x_i), y_i) P_{tr}(x_i, y_i) dx_i dy_i \\
&\quad = E_{D\backslash D_{\kappa(i)}\sim P_{tr}} \int_{\mathcal{X}\times\mathcal{Y}} [l(\hat{f}_{\kappa(i)}(x_i), y_i) P_{te}(x_i, y_i) dx_i dy_i \\
&\quad = E_{D_{\kappa(i)}\sim P_{te}} E_{D\backslash D_{\kappa(i)}\sim P_{tr}}[l(\hat{f}_{\kappa(i)}(x_i), y_i)] \\
&\quad = R_{n-n/K}(\hat{f})
\end{aligned}$$

(because $D\backslash D_{\kappa(i)}$ is a set of $n - n/K$ training examples).

Therefore,

$$\begin{aligned}
E_{D\sim P_{tr}}[IWCV(\hat{f}, K)] &= E_{D\sim P_{tr}}[\frac{1}{n}\sum_{i=1}^{n} \beta(x_i, y_i) l(\hat{f}_{\kappa(i)}(x_i), y_i)] \\
&= \frac{1}{n}\sum_{i=1}^{n} E_{D\sim P_{tr}}[\beta(x_i, y_i) l(\hat{f}_{\kappa(i)}(x_i), y_i)] \\
&= \frac{1}{n}\sum_{i=1}^{n} R_{n-n/K}(\hat{f}) \\
&= R_{n-n/K}(\hat{f}). \quad \blacksquare
\end{aligned}$$

Lemma 7 implies that if we choose K large enough for importance weighted cross validation (IWCV), e.g. $K = n$ (Leave One Out CV) or $K = \frac{n}{2}$, $IWCV(\hat{f}, K)$ provides an almost unbiased estimate of the generalization error of the algorithm given training data with selection bias. This property is valid for any loss function with or without smoothness. Therefore, we can use IWCV to evaluate performance of any algorithm in the presence of selection bias just like we can use standard CV when there is no selection bias.

2.6 Covariate Shift

At first glance, it may appear that covariate shift is not a problem because it assumes that $P(y|x)$ which determine the predicting function remains unchanged. In fact, Shimodaira [2000] showed that there are circumstances under which the predictive performance is jeopardized by covariate shift. This happens typically when the parametric model family $\{P(y|x,\theta)\}_{\theta\in\Theta}$ is misspecified, that is, there does not exist any $\theta \in \Theta$ such that $P(y|x = x,\theta) = P(y|x = x)$ for all $x \in \mathcal{X}$, so none of the models in the model family can exactly match the true relation between x and y.

The intuitive explanation of why covariate shift under model misspecification causes learning bias is that the optimal (misspecified) model performs better in dense regions of the input space than in sparse regions, because the dense regions dominate the average prediction error. If the dense regions of input are different in the training and test sets, the optimal model on the former will no longer be optimal on the latter. In other words, under model misspecification the optimal model depends on the input distribution $P(x)$, which is changed from training to test data by covariate shift.

Illustrative Example of Regression under Covariate Shift

Consider a regression problem under covariate shift where the training data are sampled from a normal distribution with mean 0 and standard deviation 0.5, i.e. $P_{tr}(x) = N(0, 0.5)$ while the test data are sampled from $P_{te}(x) = N(0, 1)$. The two distributions are depicted in Fig. 2.1a. We can see that data centered around 0 are sampled into the training set more frequently than those far away from the origin. Therefore, any model trained on this data will put more effort in minimizing prediction error in the central. This may or may not be a problem depends on whether the model is well-specified or miss specified.

Given that we use Least squares (LS) regression to learn a linear model $y = \theta x$, the empirical risk minimization becomes:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \left[\frac{1}{n} \sum_{i=1}^{n} (\theta x_i - y_i)^2 \right]$$

If the underlying data generating function is also a linear function, for

example $y = \theta^* x + noise = 2x + N(0, 0.3)$, then the model is well-specified. In this case, $\theta^* = 2$ is locally optimal for every input point. As a result, it also globally optimal given any input distribution. Given sufficient training data, $\hat{\theta} \to \theta^*$ even under covariate shift. We can see in Figure 2.1 that the linear model learned from the biased training data almost perfectly matches the optimal model to predict test data when training data set is large enough (500).

On the other hand, when the generating function is non-linear, for example $y = f(x) + noise = -x + x^3 + \sim N(0, 0.3)$, then the model is misspecified. There is no linear function in the form $y = \theta x$ that is uniformly optimal over every input point. The optimal linear function is one that minimizes prediction error in denser region of input distribution while compromising in sparser region. Therefore, when the input distribution changes, the optimal function changes accordingly. As training sample size increases, the empirical risk minimization estimator $\hat{\theta}$ still converges but not to the optimal parameter θ' for test data, i.e. $\hat{\theta} \to \theta'_{tr} \neq \theta'$.

2.7 Importance weighting for Covariate shift

Covariate shift is the simplest case of selection bias. Given the assumption $S \perp\!\!\!\perp Y|X$, which implies that $P(y|x, s = 1) = P(y|x)$, we can decompose the training distribution as following:

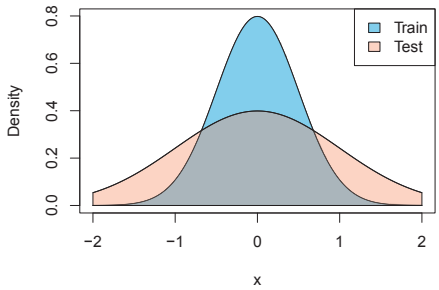
$$\begin{aligned} P_{tr}(x, y) &= P(x, y|s = 1) \\ &= P(y|x, s = 1)P(x|s = 1) \\ &= P(y|x)P(x|s = 1) \end{aligned}$$

while the test distribution is

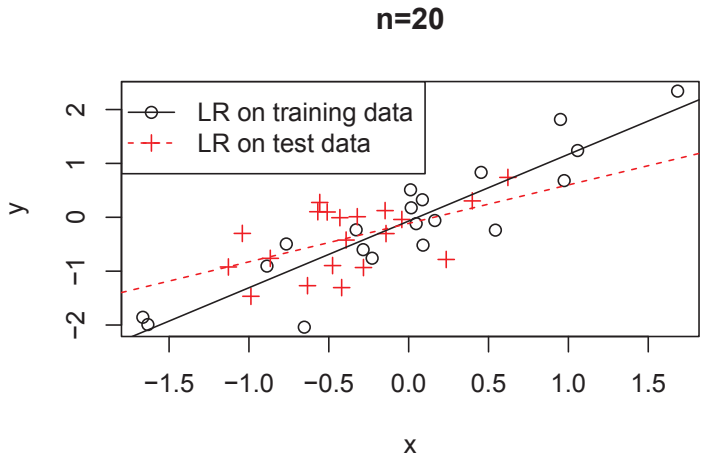
$$P_{te}(x, y) = P(x, y) = P(y|x)P(x).$$

Therefore we can use importance weighting approach presented previously to correct selection bias with the importance weight only depends on the input

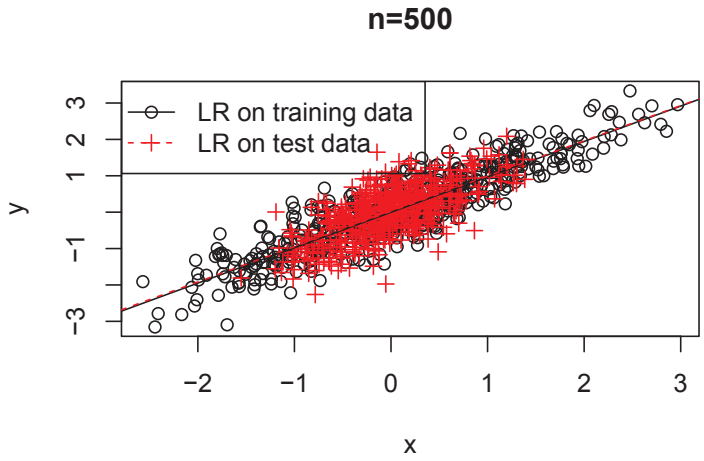
$$\beta(x, y) = \frac{P_{te}(x, y)}{P_{tr}(x, y)} = \frac{P(x)}{P(x|s = 1)} = \beta(x).$$



(a) Input density distribution in training and test data

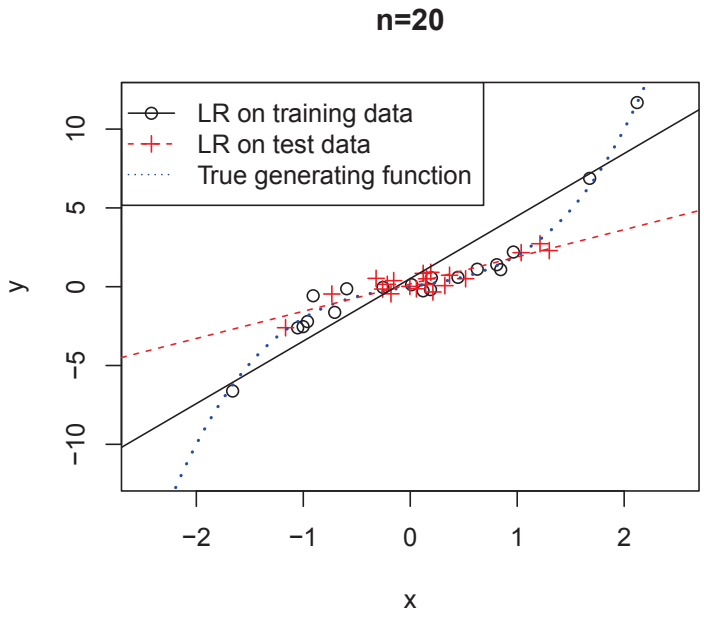


(b) Linear regression on train vs. test data with 20 examples

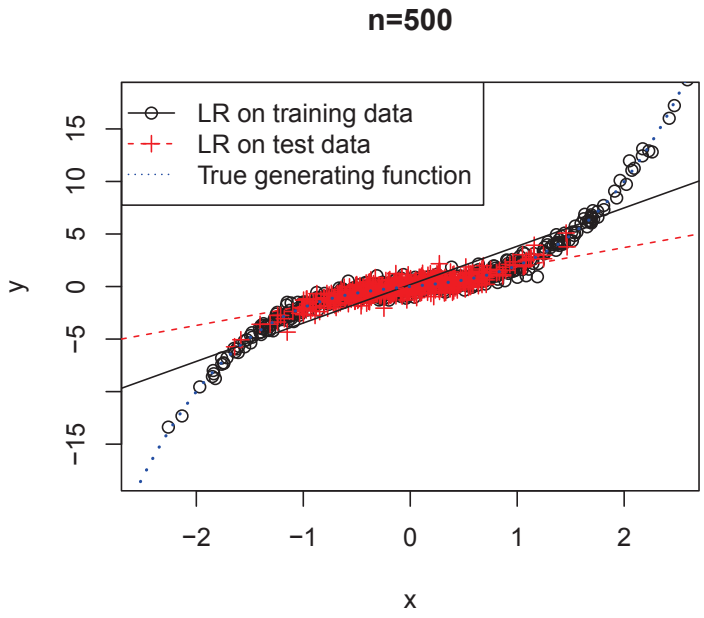


(c) Linear regression on train vs. test data with 500 examples

Figure 2.1: An illustrative example of learning under sample bias while model is well-specified



(a) Linear regression on train vs. test data with 20 examples



(b) Linear regression on train vs. test data with 500 examples

Figure 2.2: An illustrative example of learning under sample bias while model is miss-specified

This formulation allows importance weight to be estimated from the biased training data, which gives $P(x|s=1)$ and an unlabeled data set that is sampled in general population, which gives $P(x)$.

A naive approach for estimating $\beta(x)$ is to estimate the two marginal measures from the training sample and the unbiased external sample respectively. For instance, we can employ standard density estimators like Kernel Density Estimation (Parzen [1962]; Rosenblatt et al. [1956]) to estimate them. However, this naive approach is known to be inferior since density estimation in high dimensions is hard, and moreover, small estimation error could worsen the performance significantly, especially when these two probabilities $P(x|s=1)$ and $P(x)$ are small. It seems more appealing to directly estimate $\beta(x)$. Indeed, a large body of work has been devoted to this line of research Bickel et al. [2009]; Cortes et al. [2010]; Huang et al. [2006]; Kanamori et al. [2009]; Nguyen et al. [2010]; Sugiyama et al. [2007b]; Zadrozny [2004]. The function of density ratio can be estimated directly by matching the two distributions in terms of the Kullback-Leibler divergence as in Nguyen et al. [2010]; Sugiyama et al. [2007b], in terms of a least-squares function fitting Kanamori et al. [2009] or in term of kernel mean in reproducing kernel Hilbert spaces Huang et al. [2006]. In this study, we consider two of them that where shown to be successful with Covariate shift: 1) the Kernel Mean Matching (KMM) algorithm proposed by Huang et al. Huang et al. [2006] and 2) the Unconstrained Least-Square Importance Fitting (uLSIF), proposed by Kanamori et al. Kanamori et al. [2009]. We briefly present below these two methods and discuss how they adapt naturally to complete selection bias.

We assume a similar setup to semi-supervised learning where we have a labeled set of training data and an external unlabeled set of data. However unlike semi-supervised learning, where training and test data are assumed to come from the same distribution, in covariate shift framework we assume the availability of n i.i.d. training samples

$$\{(x^i, y^i)\}_{i=1}^{n} \sim P_{tr}(x) = P(x, y|s=1)$$

and n' i.i.d. test samples

$$\{(x'^i, y'^i)\}_{i=1}^{n'} \sim P_{te}(x) = P(x, y)$$

on domain $\mathcal{X} \times \mathcal{Y}$.

2.7.1 Kernel Mean Matching

Let $\Phi : \mathcal{X} \to \mathcal{F}$ denote the canonical feature map into a feature space $\mathcal{F}$ is the reproducing kernel Hilbert space (RKHS)(Aronszajn [1950]; Hofmann et al. [2008]) induced by a kernel k. B is the upper bound of the density ratio and $\mu : \mathcal{P} \to \mathcal{F}$ the expectation operator: $\mu(P) := E_{x \sim P(x)}[\Phi(x)]$. The relation between kernel mean matching and importance weighting is justified by following theorem

Theorem 8 *(Gretton et al. [2009]) The operator mu is a bijection between the space of all probability measures and the marginal polytope induced by the feature map $\Phi(x)$ if $\mathcal{F}$ is an RKHS with universal kernel $k(x, x') = \langle \Phi(x), \Phi(x') \rangle$ in the sense of Steinwart [2002].*

KMM tries to match the means in the feature space of training sample $\mu(P(x_s|s = 1))$ and test sample $\mu(P(x_s))$ by minimizing discrepancy between their empirical value,

$$\min_{\beta} \| \frac{1}{n} \sum_{i=1}^{n} \beta^i \Phi(x_s^i) - \frac{1}{n'} \sum_{i=1}^{n'} \Phi(x_s'^i) \|^2$$

subject to the constraints $\beta^i \in [0, B]$ and $|\frac{1}{n} \sum_{i=1}^{n} \beta^i - 1| \leq \epsilon$,
where $\{x_s^i\}_{i=1}^n$ are the training samples and $\{x_s'^i\}_{i=1}^n$ are the samples obtained from external sources. In the subsequence experiments, $\epsilon = (\sqrt{n} - 1)/\sqrt{n}$ and $B = 1000$ as suggested in Gretton et al. [2009].

2.7.2 Unconstrained Least-Squares Importance Fitting

This method is based on linear density-ratio models. Formally, it assumes that the density ratio $\beta(x)$ can be approximated by a linear model

$$\hat{\beta}(x) = \sum_{i=1}^{M} \alpha_i h_i(x)$$

where the basis functions h_i, $i = 1, ..., M$ are chosen so that $h_i(x) \geq 0$ for all x. The coefficients $\alpha_1, ..., \alpha_M$ are parameters of the linear model and are

determined by minimizing the discrepancy between the true and the estimated importance weights:

$$\begin{aligned} L(\alpha) &= \frac{1}{2} E_{P_{tr}}[(\hat{\beta}(x) - \beta(x))^2] \\ &= \frac{1}{2} E_{P_{tr}}[\hat{\beta}^2(x)] - E_{P_{tr}}[\hat{\beta}(x)\beta(x)] + \frac{1}{2} E_{P_{tr}}[\beta^2(x)] \\ &= \frac{1}{2} E_{P_{tr}}[\hat{\beta}^2(x)] - E_{P_{te}}[\hat{\beta}(x)] + \frac{1}{2} E_{P_{tr}}[\beta^2(x)] \end{aligned}$$

We have the last equality since

$$\begin{aligned} E_{P_{tr}}[\hat{\beta}(x)\beta(x)] &= \int \hat{\beta}(x) \frac{P_{te}(x)}{Ptr(x)} Ptr(x) dx \\ &= \int \hat{\beta}(x) P_{te}(x) dx \\ &= E_{P_{te}}[\hat{\beta}(x)] \end{aligned}$$

Approximating the expectations in L by their empirical averages and drop the last term, which is a constant, the importance weight fitting becomes a minimization problem

$$\min_{\alpha} \frac{1}{2n} \sum_{i=1}^{n} (\hat{\beta}(x_s^i))^2 - \frac{1}{n'} \sum_{j=1}^{n'} \hat{\beta}(x'^i_s) + \lambda . Reg(\alpha)$$

where the regularization term $Reg(\alpha)$ is introduced to avoid overfitting. A heuristic choice of $h_i(x_s)$ proposed in Kanamori et al. [2009] is a Gaussian kernel centered at the test points $\{x^j\}_{j=1}^{n_{te}}$ when number of test points is small (less than 100) or at *template* points $\{x^j\}_{j=1}^{100}$, which is a random subset of test set when the number of test points is large for computation advantage. The kernel width and the regularization term $Reg(\alpha)$ are optimized by cross-validation with grid search.

3 Importance Weight Estimation with Bayesian Network

The categorization of selection bias presented in previous section ignores all possible conditional independence between feature variables $X_1, X_2, ... \in X$. Therefore when there is no conditional independence holds between X as a whole, Y, and S, we cannot hope for a bias correction method. However, in practice, there are many cases where independence or conditional independence relationships between some but not all feature variables, output variable, and selection variables can help identifying formula to correct selection bias. One of the tools that have been found to be particularly useful in inferring these independence relationships is Bayesian networks (BNs). In this section we investigate the potential of BNs to encode researcher's a priori assumption about the relationship between variables, including selection variable, and to infer independence and conditional independence relationships that allow selection bias to be corrected. Besides selection bias, BN is a useful tool to diagnose the bias in estimating causal effect between variables in many biomedical and epidemiologic researches (Glymour [2006]; Greenland et al. [1999]; Hernán et al. [2002]).

Formally, a BN is a tuple $< \mathbb{G}, P >$, where $\mathbb{G} =< \mathbf{V}, \mathbf{E} >$ is a directed acyclic graph (DAG) with a set of nodes V representing the variables in the study, and a set of edges $\mathbf{E}$ representing direct probabilistic dependencies between them. P denotes the joint probability distribution on $\mathbf{V}$ whose dependencies are induced by $\mathbb{G}$. In $\mathbb{G}$, one node can be linked to another by an *directed edge*, for examples $X \rightarrow Y$, without forming any directed closed loops. If there exists a directed edge from X to Y then X and Y are said to be *adjacent* while X is called a *parent* of Y and Y is called a *child* of X. A *path* is an unbroken route traced along or against directed edges connecting adjacent nodes. A *directed path* is a path that can be traced through a sequence of directed edges in the direction indicated by the arrows of the directed edges, such as the path from X to S in $X \rightarrow Y \rightarrow S$. A node S is said to be a *collider* on a specific path if it is a common child of two variables on that path, such as S in $X \rightarrow S \leftarrow Y$, which is said to *collide* at S. If a path does not collide at S than S is said to be *non-collider* on that specific path. A

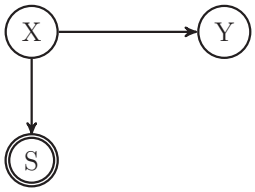
path is *unconditionally blocked* if it has one or more colliders. A path from a node Y to a node S is said to be *blocked* conditionally on X if either there is a variable in X that is a non-collider on the path. Otherwise the path is said to be *unblocked*. Two nodes X and S are are said to be *d-separated* conditional on Y if all paths from X to S are blocked conditional on Y. The BN structure encodes a set of conditional independence assumptions: that each node V_i is conditionally independent of all of its non-descendants in $\mathbb{G}$ given its parents. These independence assumptions, in turn, imply many other conditional independence statements, which can be extracted from the DAG using called d-separation criterion Pearl [1988]. If X and S are d-separated conditional on Y, X and S are conditionally independent given Y in distribution P.

The construction BN to diagnose selection biased problem can be based on the investigators understanding of the relationships and dependencies among variables which usually bear a causal effect interpretation. A direct edge from $X \to Y$ implies X is a cause of Y and Y is the effect of X. A missing link between them implies that they have no direct causal effect. The causal effect interpretation of the BN helps domain expert easily encode their assumption into a DAG from which useful independence relationships can be inferred. However, that is not the only way to construct a BN. In many practical settings the BN is unknown and one needs to learn it from the data de Morais and Aussem [2010]; Kojima et al. [2010]; Peña [2011]; Scutari and Brogini [2012]; Villanueva and Maciel [2012]. In our study, we always assume that a BN is always be given.

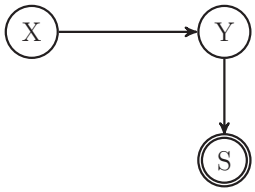
The BNs in Figure 2.3 represent three types of selection bias discussed in previous section. In Figure 2.3a, d-separation of S and Y given X implies that $S \perp\!\!\!\perp Y|X$, which is covariate shift assumption. Similarly, d-separation of S and X given Y in 2.3b implies prior probability shift assumption. In Figure 2.3c, all variables are connected, thus it falls into complete selection bias category.

3.1 Examples

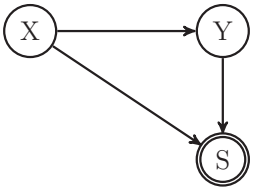
To illuminate the nature of complete selection bias that arises in the complete selection bias case, consider the examples depicted in Figure 2.4 and Figure 2.5. The Bayesian network structures should be regarded as graphical structures



(a) Covariate shift: $Y \perp\!\!\!\perp S|X$



(b) Prior probability shift: $X \perp\!\!\!\perp S|Y$



(c) Complete selection bias

Figure 2.3: Three types of selection bias represented by DAGs

encoding conditional independencies between X, Y, and S which may involve other variables as well. We provide two concrete examples in Epidemiology and Medicine for purposes of illustration.

Example 1 *A medical example of selection bias shown in Figure 2.4 (where X is a two dimensional vector (X_1, X_2)) was reported in Geneletti et al. [2009]; Horwitz and Feinstein [1978], and subsequently studied in Pearl [2012], in which it was noticed that the effect of Estrogen, X_2 (i.e., $X \setminus X_1$), on Endometrial Cancer, Y, was overestimated in the data studied. One of the symptoms of the use of Estrogen is vaginal bleeding X_1 and the hypothesis was that women noticing bleeding are more likely to visit their doctors, causing women using Estrogen to be overrepresented in the study. The exposure X_2 and the disease Y may be associated. However, this association is distorted because the selection criteria favor women who have vaginal bleeding.*

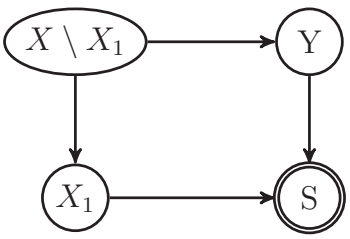


Figure 2.4: Example of selection bias in Endometrial Cancer study where $X_s = \{X_1, Y\}$.

Example 2 *Figure 2.5 represents a case-control study reported in Hernán et al. [2004] of the effect of postmenopausal estrogens, X, on the risk of myocardial infarction, Y. The variable S indicates whether a woman in the population study is selected for the case control study. The edge from disease status to selection S indicates that cases in the cohort are more likely to be selected than non case, which is the key feature of a case-control study. As women with a low bone mass density, denoted by M, were preferentially selected as controls, M is connected to S. The edge from X to M represents the protective effects of estrogens on the bone mass density. Note that Figure 2.5 is essentially the same as Figure 2.4, except that we have now M is missing in the test set. This situation typically arises in various clinical studies or epidemiological scenarios, where M is too difficult or costly to measure in the target population.*

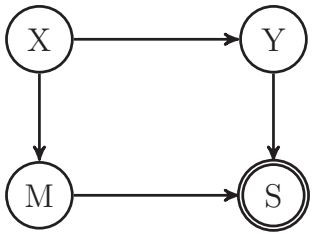
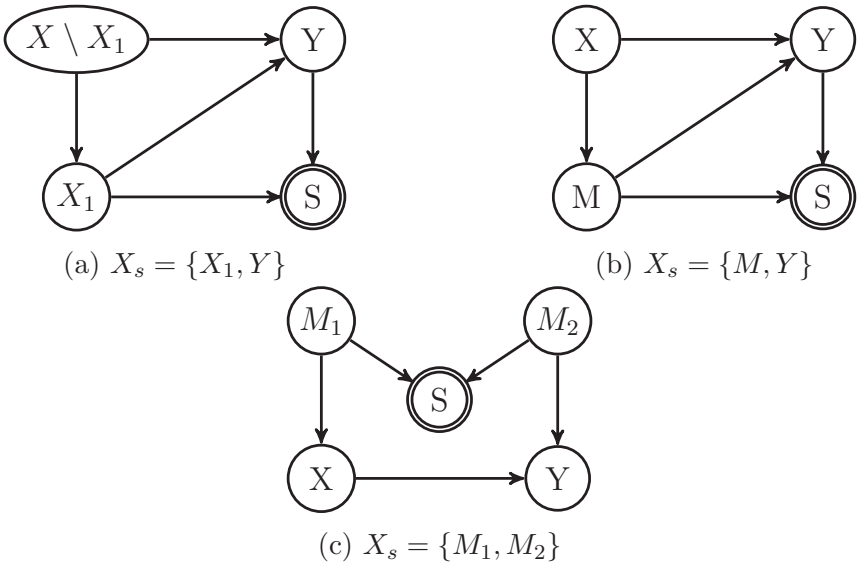


Figure 2.5: Example selection bias in the study of the effect of postmenopausal estrogens where $X_s = \{M, Y\}$.

The selection bias mechanisms shown in Figure 2.6a and 2.6b are simple variations thereof. Example in selection bias shown in Figure 2.6c is another example known as a M-structure.



(a) $X_s = \{X_1, Y\}$ (b) $X_s = \{M, Y\}$

(c) $X_s = \{M_1, M_2\}$

Figure 2.6: Top figures: Covariate shift and prior probability shift. From (a) to (e): Examples of complete selection bias mechanisms depicted graphically. The S-control vector is shown along each plot.

3.2 Recoverability of Selection Bias in Graphical Model

Recent research by Bareinboim and Pearl [2012]; Bareinboim et al. [2014] provide probabilistic and graphical conditions for recovering probability distribution from selection biased data with and without unbiased data over a subset of the variables.

3.2.1 Recoverability without External Data

Definition 4 *Given a BN graph G_s augmented with a node S encoding the selection mechanism, the distribution or conditional distribution Q is said to be s-recoverable from selection biased data in G_s if the assumptions embedded in G_s renders Q expressible in terms of the distribution under selection bias $P(v|S = 1)$. Formally, for every two probability distributions P_1 and P_2 compatible with G_s, $P_1(v|S = 1) = P_2(v|S = 1) > 0$ implies $P_1(v) = P_2(v)$*

Theorem 9 *The distribution $P(y|x)$ is s-recoverable from G_s if and only if $(S \perp\!\!\!\perp Y|X)$.*

Among three cases of selection bias, only covariate shift allows the conditional distribution of y given x to be recoverable without external data. However

recoverability of $P(y|x)$ is not sufficient to correct the bias when the model is misspecified, i.e. when the hypothesis space does not contain the true data generating mechanism $P(y|x)$ as discussed in the previous chapter. In such case, the asymptotic optimal hypothesis on the training data may be different than the asymptotic optimal hypothesis on the test data and both $P(y|x)$ and $P(x)$ (or equivalently $P(x, y)$) are required to correct the bias.

3.2.2 Recoverability with External Data

Definition 5 *Given a BN graph G_s augmented with a node S encoding the selection mechanism, the distribution query Q is said to be s-recoverable from selection biased data in G_s with external information over $T \subseteq V$ and selection biased data over $M \subseteq V$ if the assumptions embedded in G_s renders Q expressible in terms of the distribution under selection bias $P(m|S = 1)$ and $P(t)$, both positive. Formally, for every two probability distributions P_1 and P_2 compatible with G_s, if they agree on the available distributions, $P_1(m|S = 1) = P_2(m|S = 1) > 0$, $P_1(t) = P_2(t) > 0$, they must agree on the query distribution, $Q_{P_1} = Q_{P_2}$.*

Theorem 10 *The bias-free distribution $P(x, y)$ is recoverable from a S-bias training samples if there exists a set of variables X_s that satisfies:*

- *S-bias training sample contains X_s*
- *The biased free distribution of X_s is estimable.*
- *X_s controls S over (X, Y), i.e. $S \perp\!\!\!\perp (X, Y)|X_s$*
- *The support of $P(x_s|s = 1)$ contains the support of $P(x_s)$.*

Under these conditions:

$$P(x, y) = \sum_{x_s \setminus \{x,y\}} P(x, y, x_s|s = 1)\beta(x_s) \tag{2.17}$$

Where $\beta(x_s) = \frac{P(s=1)}{P(s=1|x_s)}$.

In our notation, X_s may include X, a partial of X, Y, or some variables M that is measure in training data but not in test data, e.g., bone mass density in Example 2.

Proof Bayes' rule, we have

$$P(x, y, x_s) = \frac{P(x, y, x_s, s = 1)}{P(s = 1|x, y, x_s)}.$$

In addition $P(s = 1|x, y, x_s) = P(s = 1|x_s)$ since $S \perp\!\!\!\perp (X, Y)|X_s$. Therefore,

$$P(x, y, x_s) = P(x, y, x_s|s = 1)\frac{P(s = 1)}{P(s = 1|x_s)}.$$

Finally,

$$\begin{aligned} P(x, y) &= \sum_{x_s \setminus \{x,y\}} P(x, y, x_s) \\ &= \sum_{x_s \setminus \{x,y\}} P(x, y, x_s|s = 1)\beta(x_s) \end{aligned}$$ ∎

Theorem 10 relies on a combination of data assumptions ($P(x_s)$ can be estimated) and qualitative assumptions (X_s controls S over (X, Y)) that may appear difficult to satisfy in practice. However, in certain domains like epidemiology, information about the selection process can sometimes be expressed and modeled in a communicable scientific language (e.g., graphs or structural equations) by the domain experts. Examples of common selection bias in epidemiology can be found in Hernán et al. [2004].

Theorem 10 reduces the importance weight to only depends on x_s, which is measured in both training and external data set. It is also worth noting that $\beta(x_s)$ can be reformulated as,

$$\beta(x_s) = \frac{P(x_s)}{P(x_s|s = 1)} \tag{2.18}$$

So $\beta(x_s)$ may be estimated from a combination of biased and external data. Covariate shift and prior probability shift can be seen as special cases this selection bias scheme where $X_s = X$ for covariate shift and $X_s = Y$ for prior probability shift. Replacing $\beta(x, y)$ by $\beta(x_s)$, the following results are drawn directly from Lemma 5 and Theorem 6.

Corollary 1 *Given that condition of Theorem 10 is satisfied, if $\hat{P}$ is a new distribution such that*

$$\hat{P}(x, y, x_s, s) = P(x, y, x_s, s)\beta(x_s)$$

then

$$\hat{P}(x, y|s = 1) \equiv P(x, y).$$

Corollary 2 *Given that the condition of Theorem 10 is satisfied, and $\hat{P}$ in Corollary 1, for all classifier h, all loss function $l = l(h(x), y)$,*

$$E_{x,y\sim P}(l) = E_{x,y\sim\hat{P}}(l|s = 1).$$

$E_{x,y\sim P}(l)$ is the loss that we would like to minimize and $E_{x,y\sim\hat{P}}(l|s = 1))$ is the loss that may be estimated from the new biased sample drawn from the weighted distribution $\hat{P}$.

Similarly, directly weighting loss function of a learning algorithm with $\beta(x_s)$ will correct selection bias.

Corollary 3 *The expectation of importance weighted loss with $\beta(x_s)$ over the training distribution is equal to the expectation of loss over test distribution.*

$$\begin{aligned} R(f) &= E_{P_{te}}[l(f(x), y)] \\ &= E_{P_{tr}}[\beta(x_s)l(f(x), y)]. \end{aligned}$$

As a result, we can either use subsampling or modify the leaning algorithm with importance weighted loss function to correct for selection bias.

4 Experimentation and Results

In this section, we assess the ability of importance weighting to remove complete selection bias based on Theorem 10. In the first three toy experiments (two regression problems and one classification problem), we investigate whether covariate shift and prior probability shift corrections may help reduce complete selection bias despite our assumptions between the training and test distributions difference being violated (through an invalid choice for

X_s). With this in mind, KMM (uLSIF will be used later on real data only) is applied under three assumptions:

- Covariate shift (i.e., $\beta(x) = \frac{P(x)}{P(x|s=1)}$ or $x_s = x$),
- Prior probability shift (i.e., $\beta(y) = \frac{P(y)}{P(y|s=1)}$ or $x_s = y$, the importance weight is estimated using the bias training data set and an unbiased data set that contains only labels),
- Complete selection bias (i.e., $\beta(x_s) = \frac{P(x_s)}{P(x_s|s=1)}$, x_s is correctly specified).

They are denoted KMM(X), KMM(Y), KMM(X_s) in the sequel. The test error will plotted as a function of the number of training points. All experiments on synthetic data are repeated 30 times for each number of training points. The reported errors are average values. We examine: 1) the case where the learning model is well-specified or misspecified and 2) when X_s is not completely observed. The toy experiments are intended mainly to provide a comparison between the above three estimators and the plug-in estimator that estimates $\beta(x)$ from the true (known) distribution, against the optimal solution that consist of fitting the model directly on the test data. It should be emphasized that neither KMM nor uLSIF requires any prior knowledge of the true sampling probabilities. We then test our approach on real world benchmark data sets, from which the training examples are selected according to various biased sampling schemes as suggested in Huang et al. [2006]. Finally, we consider a plausible biased sampling schemes on a prospective cohort study which included more than 7500 elderly osteoporotic women followed-up during 4 years.

4.1 Regression Problem with a Well-specified Model

Consider the S-bias mechanism displayed in Figure 2.7, where the feature X has a uniform distribution in $[0, 1]$: $P(X) \sim \mathcal{U}(0, 1)$. Note that the influence of M on Y is mediated by $\{X, S\}$.

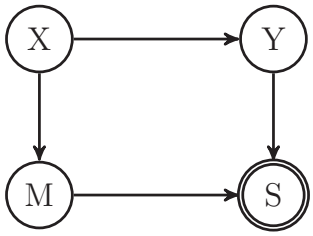


Figure 2.7: Selection mechanism in regression problem with a well-specified model and $X_s = \{M, Y\}$.

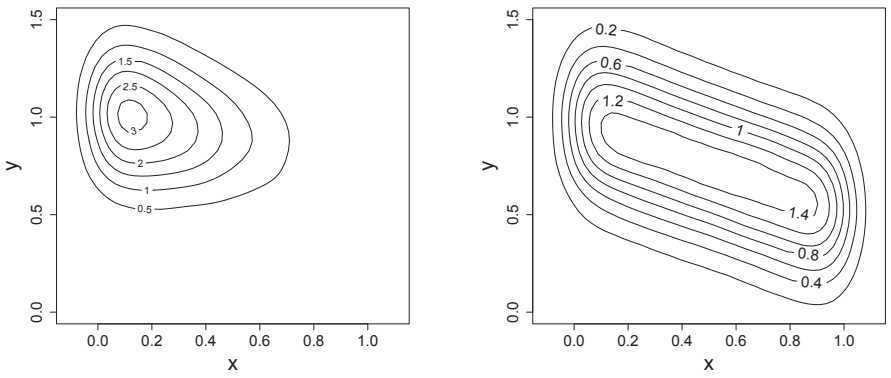
The observations are generated according to $y = 1 - 0.5x$ and are observed in Gaussian noise with standard deviation 0.5 (see Figure 2.8c); the black solid line is the noise-free signal). The intermediate variable M, between X and S, is generated according to $M = X + \mathcal{N}(0, 0.3^2)$. As M is only measured in the training set, it is not used an input feature in our regression model. Therefore, we are investigating a case where X_s is partially missing in the test set. The probability of a given example being included in the training set depends on Y and M and is given by

$$P(S = 1|m, y) \sim \begin{cases} y - m, & \text{if } 0.1 \leq (y - m) \leq 1 \\ 0.1, & \text{if } (y - m) \leq 0.1 \\ 1, & \text{otherwise} \end{cases}$$

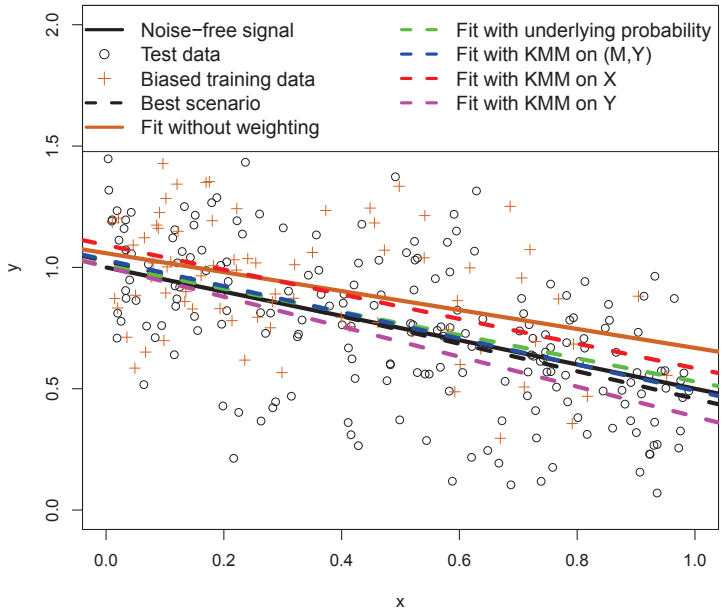
Note that the minimum value of $P(S = 1|m, y)$ needs to be greater than 0 so that the support of $P(m, y)$ is contained in the support of $P(m, y|s = 1)$, as required by Theorem 10. The choice of $P(m, y)$ is intended to induce a noticeable discrepancy between $P(y|x, s = 1)$ and $P(y|x)$. We sampled 200 training (red crosses in Figure 2.8c) and testing (grey circles) points from P_{tr} and P_{te} respectively. The bias is clearly noticeable from the X-Y contour plots in Figure 2.8*a* and *b*. The bias-free distribution $P(x, y)$ is recoverable from the S-bias training samples since $\{M, Y\}$ satisfies Theorem 10. Thus we use Corollary 3, to remove selection bias by weighting the squares loss on each example of the linear model by the importance ratio:

$$\beta(x_s) = \beta(m, y) = \frac{P(m, y)}{P(m, y|s = 1)} = \frac{P(s = 1|m, y)}{P(s = 1)}$$

where $P(s = 1|m, y)$ and $P(s = 1)$ may be obtained from the known



(a) Train distribution (b) Test distribution



(c) Linear model fit

Figure 2.8: Toy regression problem 1. (a) and (b) Contour plots X-Y on training and test sets; (c) Polynomial models of degree 1 fit with OLS and WOLS.

selection mechanism shown above or directly estimated by KMM using training and unlabeled data.

We attempted to model the observations with a linear model, which is a well-specified model considering that the true generating function is also

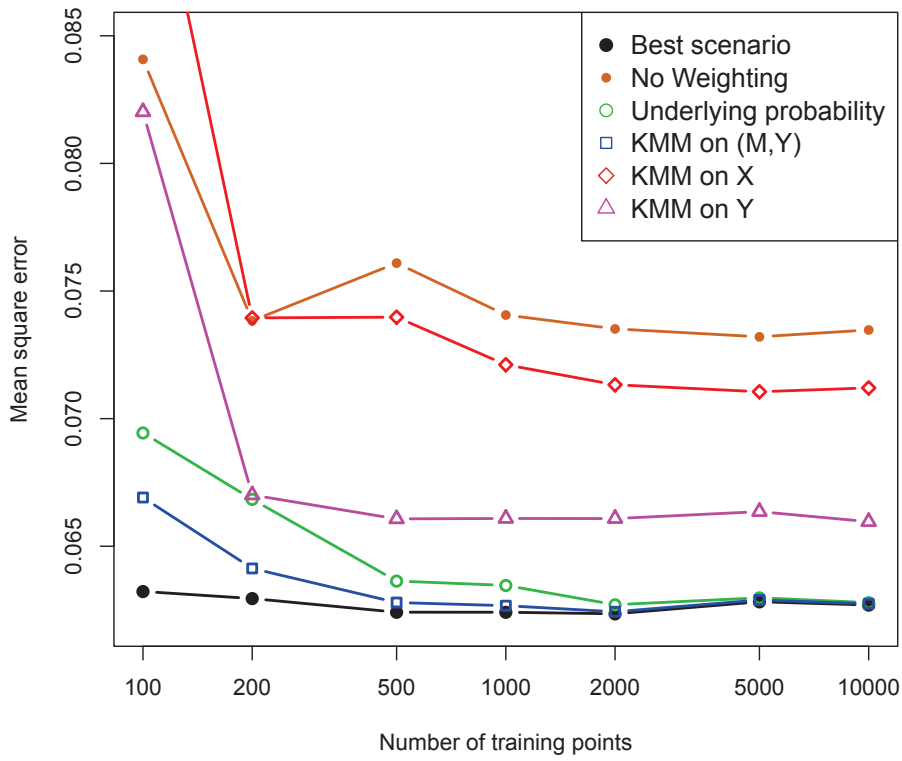


Figure 2.9: Average performances of four WOLS methods and OLS on the test data as a function of the number of training points.

linear. The black dashed line in Figure 2.8c is a best-case scenario given our test points, which is shown for reference purposes: it represents the model fit using ordinary least squared (OLS) on the test set. The brown line is a second reference result, derived only from the training data via OLS, and predicts the test data very poorly. Note that unlike covariate shift where well-specified model can perform well without importance weighting, in this case, selection bias strongly affects the prediction performance even the learning model is well-specified.

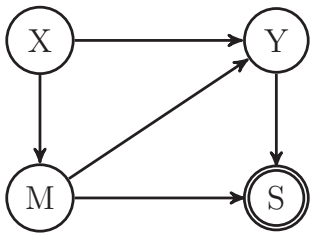
The green dashed line is a third reference result, fit with weighted ordinary least square (WOLS), using the true $\beta(x_s)$ values calculated from the true data generating mechanism, and predicts the test data quite well. The other three dashed lines are fit with WOLS using the KMM weighting schemes under the

three assumptions. Note that the true generating model between X and Y is included in the hypothesis space.

We estimated the effect of the number of training points on the estimation of the reweighting factors by examining the average mean square error (MSE) on the test set as a function of the number of training points. As may be observed in Figure 2.9, the error goes down as the sample size increases, until it reaches an asymptotic value. KMM(X_s) performs well even with relatively moderate amounts of data and achieves almost optimal error quite quickly, handily outperforming the reweighting method based on KMM(X) and KMM(Y) by a noticeable margin. More interestingly, KMM(X_s) also outperforms the reweighting method based on the true data generating mechanism, especially when the sample size is small. This result may seem counter-intuitive at first sight: the reason is that the exact importance-sampler weights are not always optimal unless we have an infinite sample size. See Shimodaira [2000] for a thorough discussion. Remarkably, despite our assumption regarding the difference between the training and test distributions being violated, KMM(Y) and KMM(X) improve the test performance. However, this improvement is not sufficient to correct totally the selection regardless of the training sample size.

4.2 Regression Problem with a Misspecified Model

In this second toy experiment, our data are generated according to the non-linear function. In addition, we assume that Y is directly dependent on the missing variable M and not mediated by X and S as depicted in the S-bias mechanism in Figure 2.10.



.

Figure 2.10: Selection mechanism in regression problem with a misspecified model and $X_s = \{M, Y\}$

The input samples are generated according to $X \sim \mathcal{N}(0, 0.3)$. The intermediate variable M is generated according to $M = X + \mathcal{N}(0, 0.3^2)$. The observations are generated according to $y = sinc(x) + 0.5m$ and are observed in Gaussian noise with standard deviation 0.5 (see Figure 2.11c; the black curve is the noise-free signal). Here again, we attempted to model the observations with a linear model which is misspecified, i.e. the true generating model between X and Y is not included in the hypothesis space. The S variable indicating actual selection to the training set is generated according to,

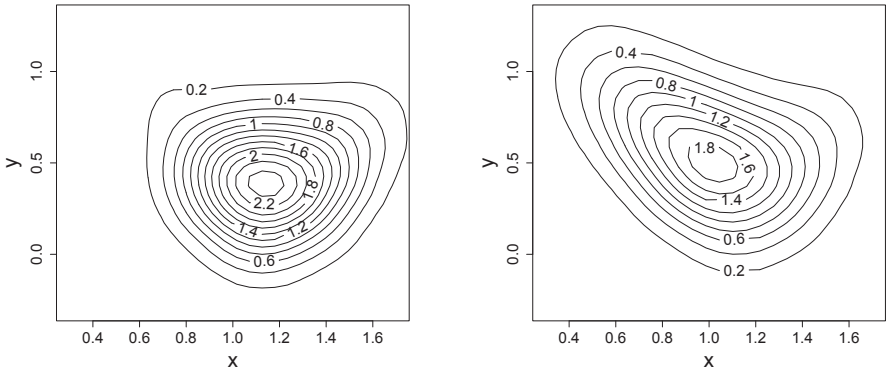
$$P(S = 1|m, y) \sim \begin{cases} m - y & \text{if } 0.1 \leq m - y \leq 1 \\ 0.1 & \text{if } m - y \leq 0.1 \\ 1 & \text{otherwise} \end{cases}$$

The distribution shift due to selection bias above is clearly noticeable from the X-Y contour plots in Figure 2.11a and 2.11b. Here again, the bias-free distribution $P(x, y)$ is recoverable from the S-bias training samples since $\{M, Y\}$ satisfies Theorem 10 (i.e., $\{X, Y\} \perp\!\!\!\perp S|\{M, Y\}$). Thus we use Corollary 3 to remove selection bias.

As expected, KMM(X_s) compares more favorably to the other methods and does exceptionally well even with moderate amounts of data. Note that, contrary to the previous experiment, this is pretty much a dead heat between KMM(X) and KMM(Y) in terms of performance. Still, both approaches were able to reduce the bias by a noticeable margin compared to the baseline unweighted approach ("no weighting") although not being able to match the best scenario where there is no selection bias. Note that because KMM(X) and KMM(Y) relies on the wrong assumptions about selection mechanism, we can always hand-pick a selection scenario so that importance weighting that solely relies on X or Y becomes less effective or even worse than the baseline unweighted approach as we will see in the next experiment.

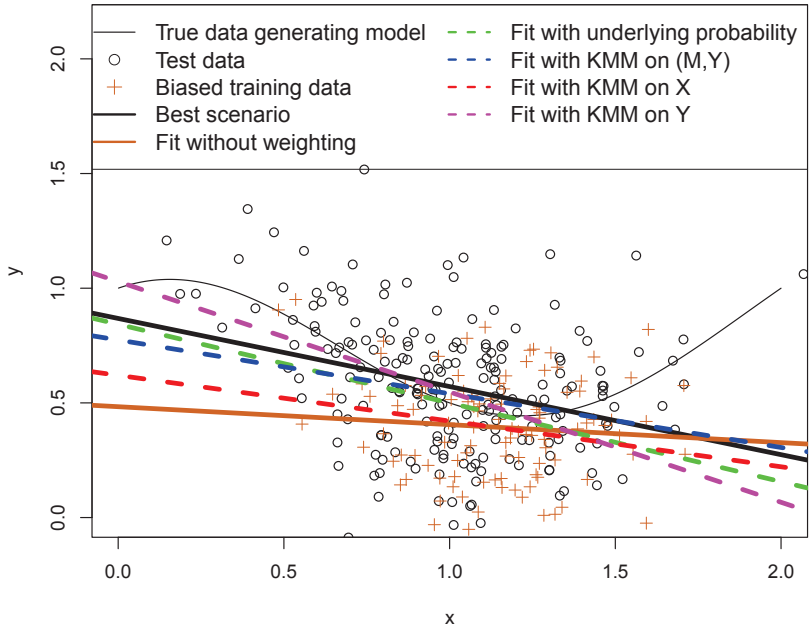
4.3 Toy Classification Problem

We now turn our attention to a synthetic classification problem. Consider the S-bias mechanism depicted in Figure 2.13, where X consists of two variable (X_1, X_2).



(a) Train distribution

(b) Test distribution



(c) Linear model fit

Figure 2.11: Toy regression problem 2. (a) and (b) Contour plots X-Y on training and test sets; (c) Polynomial models of degree 1 fit with OLS and WOLS.

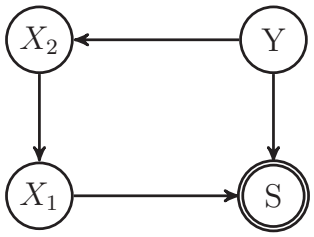


Figure 2.13: Selection mechanism in classification experiment with $X_s = \{X_1, Y\}$.

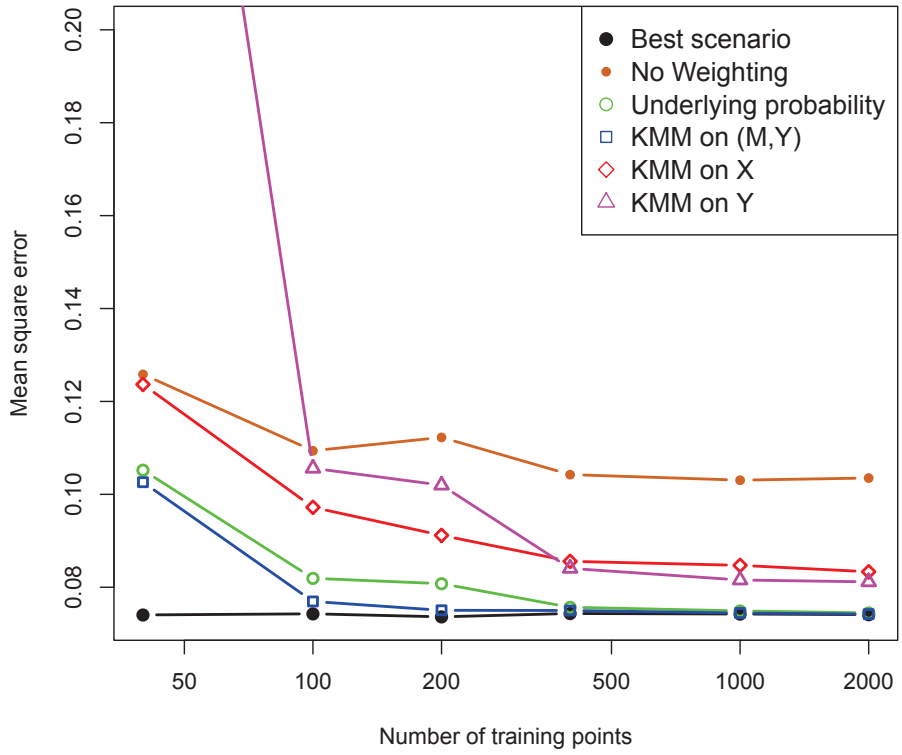


Figure 2.12: Average performances of four WOLS methods and OLS on the test data as a function the number of training points of toy regression problem 2.

Two class of data present with the same probability
$p(y = 1) = p(y = -1) = 0.5$.
X_2 depends on Y as $P(X_2|Y = 1) \sim \mathcal{N}(0, 0.5)$ and $P(X_2|Y = -1) \sim \mathcal{N}(2, 0.5)$.
Finally, X_1 is generated according to $X_1 = X_2/2 + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 0.5^2)$.

It follows that the optimal decision boundary in terms of mean square error between positive-labeled and negative-labeled examples is the line $x_2 = 1$. While the labels are solely determined by the feature X_2, labels are dependent on X_1 in the biased training set because conditioning on S opens a path between X_1 and Y. Positive samples are preferentially selected to the training set when they are close to the true decision boundary,

$$P(S=1|x_1,y) \sim \begin{cases} 0.2, & \text{if } 0 \leq x_1 \leq 1 \text{ and } y = 1 \\ 1, & \text{otherwise} \end{cases}$$

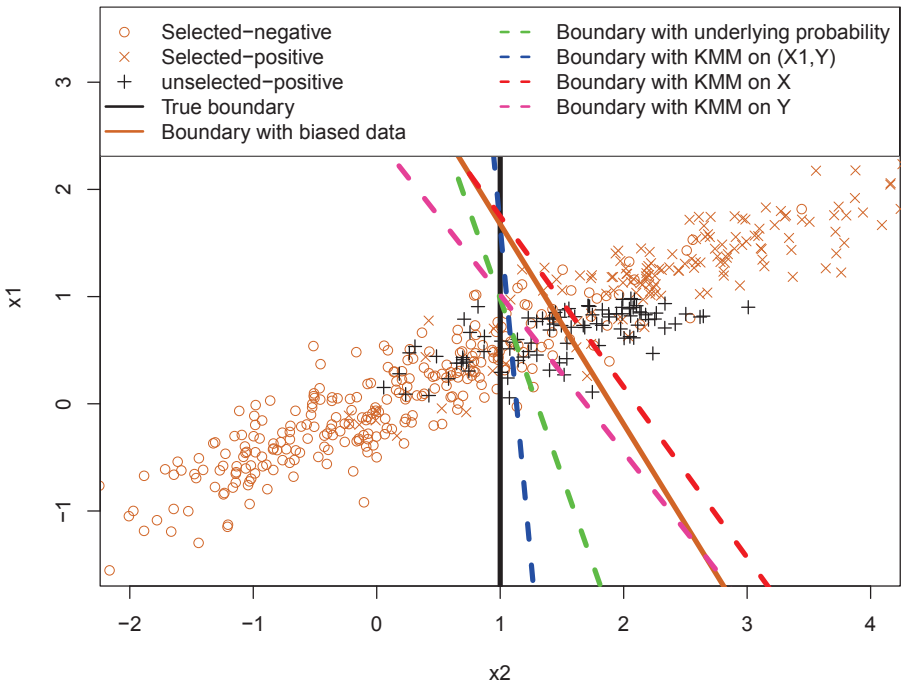


Figure 2.14: Polynomial models of degree 1 fit with OLS and WOLS of toy classification problem

500 training data points are plotted in Figure 2.14. As may be seen, the selection causes some positive examples (black cross sign) to be excluded from the training set while all the negative examples (brown circle) are included. A linear function $f(x_1, x_2)$ is trained to minimize the Mean Square Error (MSE) on the training set. Due to selection bias the boundary learned on biased training set (brown solid line) is shifted and rotated. The set of variable that controls the selection mechanism is $X_s = \{X_1, Y\}$ since $S \perp\!\!\!\perp X, Y | \{Y, X_1\}$. Importance weighting using the underlying probability (green dashed line) and KMM on X_s (blue dashed line) achieve a MSE almost as low as the best possible model when training sample size is large enough as can be seen

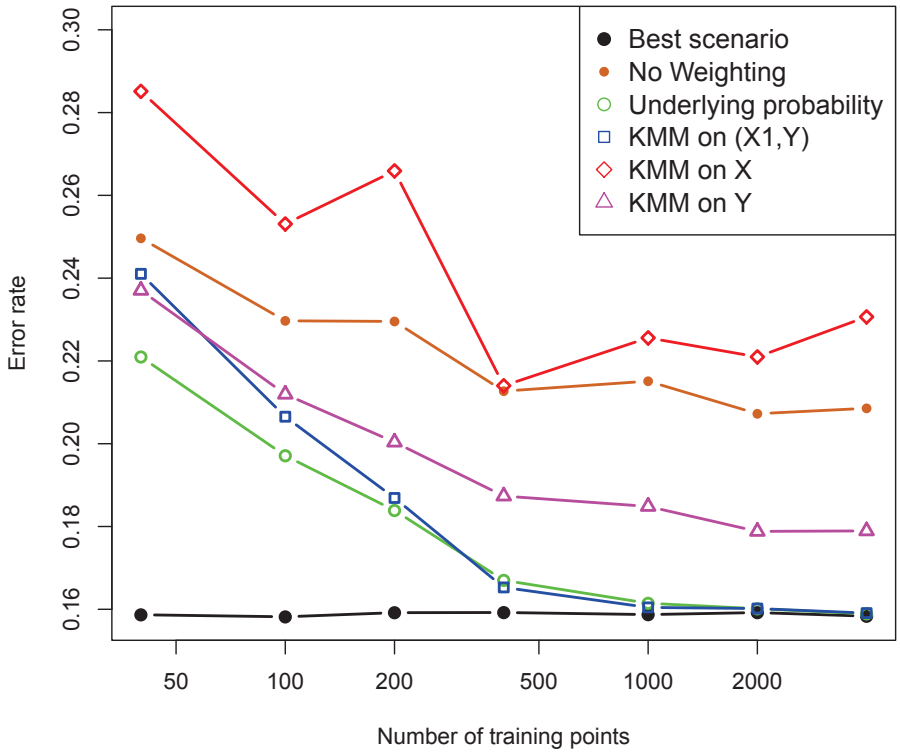


Figure 2.15: Average performances of four WOLS methods and OLS on the test data as a function the number of training points of toy classification problem.

in Figure 2.15. KMM(X) (red dashed line) amplifies the current selection bias, causing a higher classification error rate with respect to the unweighted baseline method. This can be seen as an example of a bias amplification caused by an inappropriate choice of variates to control.

In contrast, KMM(Y) (purple dashed line) adjusts the proportion of positive-labeled and negative-labeled in the training set and reduces the bias by 75% as shown in Fig.2.15. However this improvement can easily be reversed if we choose a different selection mechanism as can be seen in example below.

Example 3 *Consider a learning problem where the training and test distribution are shown in Table 2.1 and Table 2.2.*

$p(x, y\|s = 1)$	$y = 0$	$y = 1$
$x = 0$	0.375	0.25
$x = 1$	0.25	0.125

Table 2.1: Train distribution

$p(x, y)$	$y = 0$	$y = 1$
$x = 0$	0.4	0.1
$x = 1$	0.1	0.4

Table 2.2: Test distribution

The optimal prediction for this learning problem is $y = f_0(x) = x$*, which achieves a prediction error of* 20%*. However, under the given selection bias, the prediction function learned from the training data (assumed to be large enough) will be* $y = f_1(x) = 0$*, which predicts correctly only* 50% *of the test data. This selection bias is controlled by both* x *and* y*. If we make a wrong assumption about the selection mechanism, e.g.,the prior probability shift, we will apply the importance weight* $\beta(y)$ *as shown in Table 2.3 and get the training data set that follows a weighted distribution as shown in Table 2.4. Consequently, we learn the prediction function* $y = f_2(x) = 1 - x$*, which predicts incorrectly* 80% *of test data, worse than the unweighted model. Therefore, using importance weight with prior probability shift assumption is harmful in this case.*

$\beta(y)$	$y = 0$	$y = 1$
$x = 0$	0.8	1.3333
$x = 1$	0.8	1.3333

Table 2.3: Importance weight assuming prior probability shift.

$p_w(x, y)$	$y = 0$	$y = 1$
$y = 0$	0.3	0.3333
$y = 1$	0.2	0.1667

Table 2.4: Weighted distribution.

The conclusion we can draw from these toy experiments is that complete selection bias could be corrected if we are able to estimate $\beta(x_s)$ with sufficient confidence. As the number of training samples increases, the method's prediction error converges to the unbiased error. KMM(X_s) results are comparable and sometime better with respect to the underlying probability method. While KMM(X_s) is far superior to both KMM(Y) and KMM(X), it is worth mentioning that KMM(Y) and KMM(X) could improve test performance significantly in some cases and amplify selection bias in some other cases. Therefore, selecting a correct set of variables to be controlled is critical in correcting for selection bias. If we make a wrong assumption about the selection mechanism,

the result could be very random. A similar problem of bias amplification when wrong variables are adjusted can be found the causal inference domain (Pearl [2010]). Finally, it should also be noted that the external data used to estimate the importance weight in KMM(X_s) are labeled data with partial feature vector (not all features have to be included). This requirement seems to severely restrict the application of our approach in practice. However, labeled data with few features at population level are a lot easier to find than the labeled data with full feature vector in many cases. For instance, the distribution of a certain diseases at different age is quite easy to find from the public domain or from combining data of many researches. As a result, if the variables that control the selection of patients into the study are patients' age and the disease itself, we can use those available data to correct for the selection bias.

4.4 Real-world Data sets

In the following experiments, we examine whether using importance weighting can reduce selection bias in 10 UCI data sets with 5 classification tasks and 5 regression tasks. We employ three methods to estimate importance weighting: ratio of underlying probability, KMM, and uLSIF and compare their performance against the baseline unweighted method.

For each data set, X_s is chosen to be the label Y and the most correlated input variable to Y (denoted as X_1 for simplicity). The choice of X_1 is to induce a clear effect of selection bias. In fact, X_1 can be a set of variables and in the extreme case, it could include all the parent nodes of Y, making the external data needed for the selection bias correction be sufficient for predicting the label Y. In that case, we can just ignore the biased training data and use the external data set, if it is available, to build the prediction model. However, we argue that in practice unless in adversarial cases, the selection mechanism is normally controlled by very few variables whose prediction power can hardly dominate that of other variables in the study. Therefore, using X_1 to predict Y yields a much worse prediction accuracy compared to using the full feature vector even under selection bias.

The selection bias mechanism is illustrated in Figure 2.16.

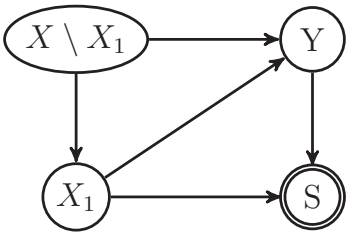


Figure 2.16: Selection mechanism in real world data set experiment with $X_s = \{X_1, Y\}$

The selection variable S for each training example is determined according to two scenarios depending on whether it is regression or classification problem. For regression problem, we use

$$P(s = 1|x_1, y) = \frac{exp(ax_1 + by + c)}{1 + exp(ax_1 + by + c)},$$

where a, b, c, are parameters that determine the bias and showed in Table 2.5, along with information of each data set.

Data set	dim	# train	# test	σ	a	b	c
India diabetes	8	400	208	1e-4	0.5	-	-
Ionosphere	34	250	128	0.1	0.5	-	-
BreastCancer	10	300	158	0.01	0.5	-	-
Haberman	3	200	105	0.05	0.5	-	-
GermanCredit	24	700	375	1e-5	0.5	-	-
Airfoil self noise	5	1000	492	0.1	1	1	0
Abalone	8	2000	1360	0.1	1.5	-1	1
Computer Hardware	9	100	53	1e-4	1	-2	0
Auto MGP	8	300	173	0.1	1	-1	0
Boston Housing	13	300	92	0.01	1	1	0

Table 2.5: UCI data sets characteristics, Gaussian kernel width, and bias parameters. Parameters b and c are not used for classification tasks.

For binary classification problem, we use:

$$P(s=1|x_1,y) = \begin{cases} a & \text{if } x_1 > mean(x_1) \text{ and } y = 1 \\ 1, & otherwise \end{cases}$$

.

For each data set, we then train 4 predictive models learned under the four weighting schemes discussed above and a model learned from the unbiased data (baseline) using SVM-light (Joachims [1998]) which allows importance weighting to be fed directly to SVM. All classifiers are trained with the common Radial Basis Function (RBF), with a kernel size σ chosen through a 5-fold cross validation. This procedure is repeated 100 times for each data set. The performance metrics we use are averaged test errors for classification problems and normalized mean square errors (NMSE) given by

$$\frac{1}{n_{te}} \sum_{i=1}^{n_{te}} \frac{(y_i^{te} - \hat{y}_i)^2}{var(y^{te})}$$

for regression problem.

The numerical results are reported in Table 2.6 and visualized in Figured 2.18. As may be seen, all importance weighting schemes achieve lower prediction error with respect to the baseline unweighted scheme. The underlying probability weighting scheme performs pretty good.

Curiously, on the Boston Housing data set, all the three weighting schemes perform worse than the baseline unweighted method. This is an example showing that the increase of variance due to importance weighting may exceed its bias correction effect, worsening the overall performance. Figure 2.17 shows that the prediction error on the Boston Housing data set is much more sensitive to training sample size than other data set, e.g. the Airfoil Self Noise data set, on which importance weighted models perform well. Therefore the reduction of the effective training sample size due to importance weighting have a much deeper impact on the Boston Housing than on other data set.

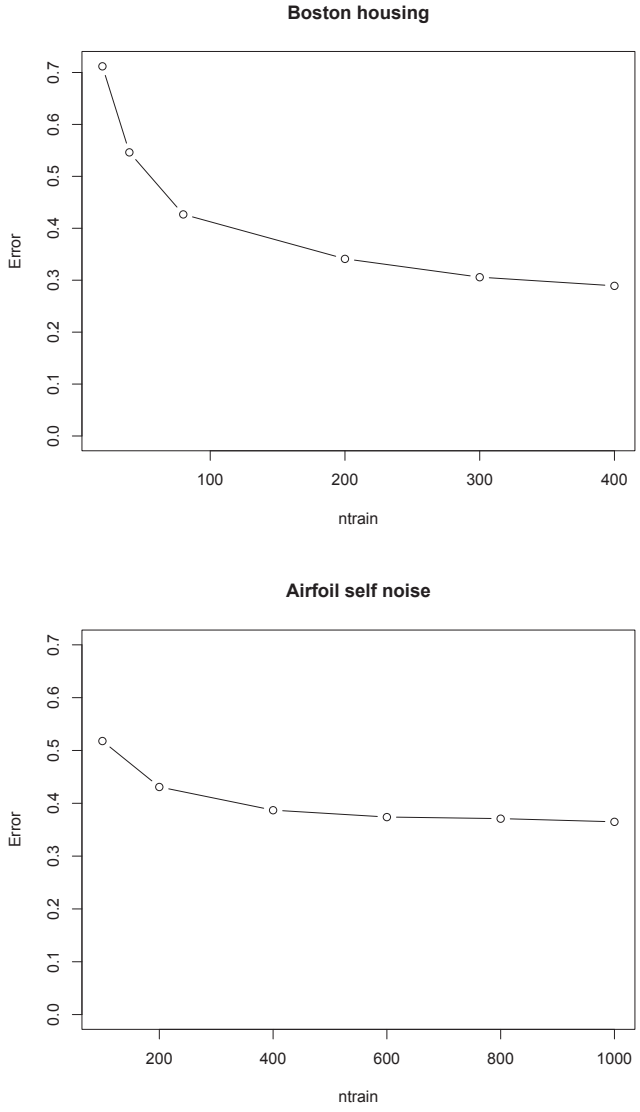


Figure 2.17: MSE vs. training sample size on Boston Housing and Airfoil Self Noise data sets.

Data set	No weighting	KMM	uLSIF	Underlying P	Unbiased model
India diabetes	0.338 ± 0.049	**0.266 ± 0.040**	0.332 ± 0.053	0.287 ± 0.055	0.258 ± 0.035
Ionosphere	0.069 ± 0.039	**0.066 ± 0.039**	0.067 ± 0.040	0.067 ± 0.039	0.065 ± 0.036
BreastCancer	0.044 ± 0.016	**0.039 ± 0.015**	0.043 ± 0.017	0.040 ± 0.016	0.038 ± 0.015
Haberman	0.264 ± 0.069	**0.262 ± 0.071**	0.263 ± 0.070	0.262 ± 0.071	0.262 ± 0.071
GermanCredit	0.300 ± 0.044	0.298 ± 0.046	0.298 ± 0.045	**0.298 ± 0.046**	0.295 ± 0.046
Airfoil self noise*	0.534 ± 0.104	0.470 ± 0.122	0.475 ± 0.082	**0.445 ± 0.081**	0.403 ± 0.059
Abalone*	0.526 ± 0.048	0.484 ± 0.054	0.521 ± 0.057	**0.466 ± 0.041**	0.456 ± 0.036
Computer Hardware*	0.326 ± 0.308	0.321 ± 0.304	0.321 ± 0.299	**0.319 ± 0.307**	0.305 ± 0.201
Auto MGP*	0.268 ± 0.148	0.298 ± 0.192	0.212 ± 0.129	**0.203 ± 0.128**	0.129 ± 0.063
Boston Housing*	**0.323 ± 0.110**	0.327 ± 0.127	0.349 ± 0.133	0.332 ± 0.127	0.298 ± 0.112

Table 2.6: Mean test error averaged over 100 trials of different weighting schemes on UCI data set. For each data set, the method that yields the lower error is described in bold face. Data sets marked with * are for regression problems

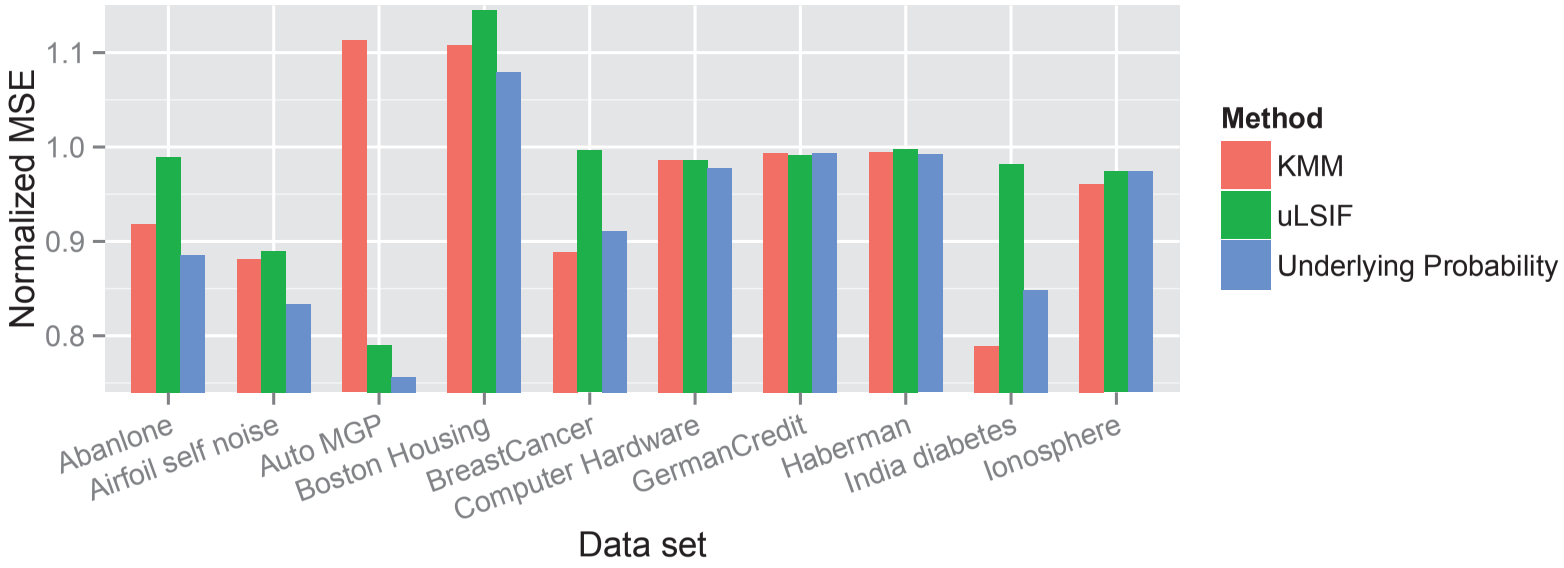


Figure 2.18: MSE gain of the weighted model (over the unweighted model) on each real-world data set when the importance weights are estimated using KMM, ulSIF and underlying probability.

Importance Weighted Cross Validation for Selecting Weighted or Unweighted Model

The importance weighting has two effect on the learning algorithms: the bias correction and the effective training sample size reduction. The former improves while the later worsens the algorithms' overall performance. Which effect dominates the other is data-dependent and cannot be determined theoretically. Therefore, we employ IWCV, which was proven to be almost unbiased, to avoid using importance weighted model when its training sample size reduction effect dominates its bias correction effect. We repeat the experiments with UCI data sets, adding a 10-fold IWCV for each weighting scheme. For each of the 100 trials on each data set and each weighting scheme, we only use a weighted model to predict test data when it outperforms unweighted model; otherwise we use unweighted model.

The numerical results are visualized in Figured 2.19. With IWCV, all the three weighting scheme still perform worse than unweighted model on Boston Housing data set but with a very small margin because IWCV is able to detect that the weighted model is worse than unweighted model most of the time (83/100 trials with KMM, 93/100 trials with uLSIF, 85/100 trials with underlying probability weighting). On the other hand, IWCV can only detect 47/100 trials where weighted model with KMM performs worse than unweighted model on Auto MGP data set. This and the fact that the other two weighting scheme perform pretty well on the same data set (reduce over 20% of MSE) suggest that the importance weight on Auto MGP data set estimated by KMM is not a good approximation of the true importance weight. This is an example showing that the overall performance of the importance weighting methods depend not only on the trade-off between bias reduction effect and effective training sample size reduction effect but also on how good the importance weight is estimated.

Non-parametric Test of Experiment Result

In order to better assess the overall results obtained for each of 4 weighting schemes, a non-parametric Friedman test was firstly used to evaluate the rejection of the hypothesis that all the models perform equally well (except the

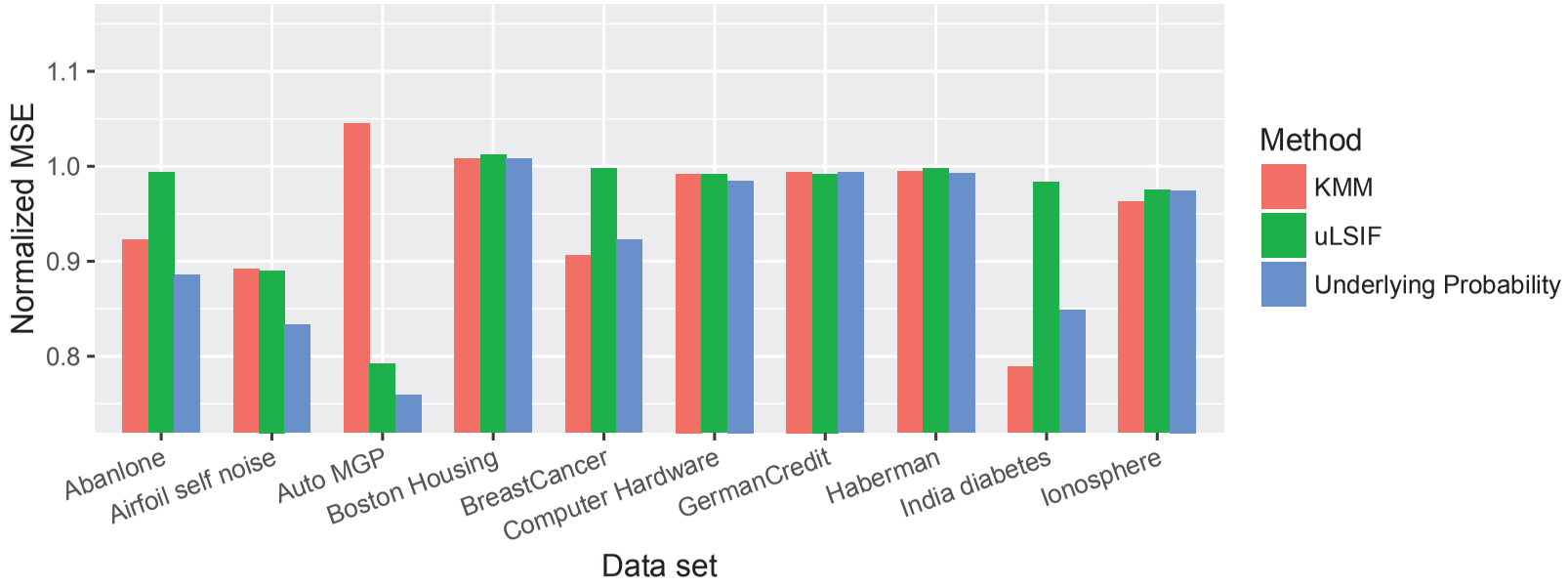


Figure 2.19: MSE gain of the weighted model (over the unweighted model) on each real-world data set when the importance weights are estimated using KMM, ulSIF and underlying probability with IWCV to decide whether to use weighted or unweighted model.

unbiased model of course) at significant level 5%. Statistically significant differences were observed. So we proceeded with the Nemenyi post hoc test. The results along with the average rank diagrams are shown in Fig. 2.20. The ranks are depicted on the axis, in such a manner that the best ranking algorithms are at the rightmost side of the diagram. The algorithms that do not differ significantly (at p = 0.05) are connected with a line. As may be observed in Figure 1, contrary to uLSIF, KMM is significantly better than no weighting.

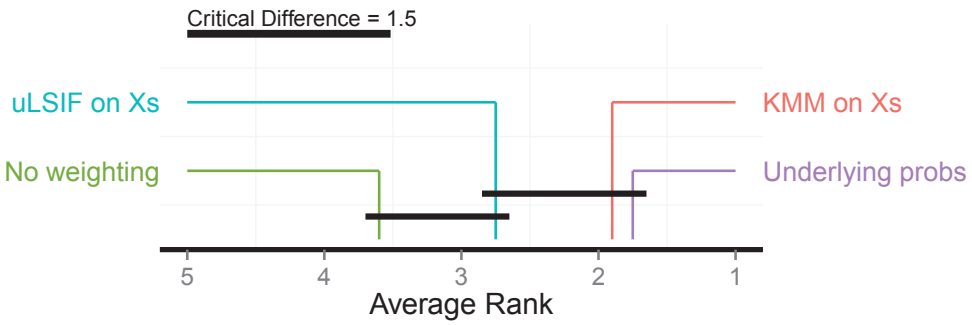


Figure 2.20: post hoc analysis

4.5 Hip Fracture Data

We next test our selection bias correction method on EPIDOS samples Caillet et al. [2015] which included more than 7598 elderly osteoporotic women followed-up during 4 years. Women who were not able to walk independently and those who had a hip fracture or bilateral hip replacement were excluded

The goal is to discriminate between individuals of the positive class (having hip fracture) from negative class based on 14 available features described in Table 2.7. The data set is highly imbalanced with only 293 positive example, so we sample to keep only 4% negative examples and also remove examples with missing value to simplify the problem.

We simulate a synthetic selection bias scenario that the chronicle diseases (X_1) and the hip fracture Y have a negative effect on the inclusion of an example into the training sample, following the Table 2.8. In this scenario, women who have many ($\geq$ 2 chronicle diseases, or hip fracture have some

probability $\frac{b}{2} \geq 0$, where b represents the bias level, to fail to follow up with the experiment, thus be excluded from the training set. This probability is increased to b if they have both many chronicle diseases and hip fracture. This is a typical survival bias in epidemiology.

Name	**Description**	**Values**
Fracture	Hip-fracture during the 4-years follow-up	binary
Age	Age at study inclusion	< 80, $80 \leq 85$, $85 \leq 90$, > 90
Chron_disease	Number of chronic diseases	binary: < 2, ≥ 2
Psycho	Use of sedatives or anxiolytics at inclusion	binary
Vit_D	Use of vitamin D at inclusion or history of vitamin D one year before inclusion	binary
Gluco	Use of glucocorticoids at inclusion or history of glucocorticoids one year before inclusion	binary
Alcohol	Daily intake of alcohol in g	binary: $1 \leq 20$, > 20
Tobacco	Tobacco smoking	none, former, actual
Gait_speed	Gait speed at inclusion in m/s	< 0.60, $0.6 \leq 0.85$, $0.85 \leq 1$,> 1
Test_5	Five chair test (time to sit down and stand up five times) in s	$1 \leq 9$, $9 \leq 16$, $16 \leq 23$,> 23, incapable
BMI	Body mass index at inclusion	low, normal, obesity
BMD	T-score of BMD of the neck at inclusion	binary: normal or ≤ 1, ≤ 2.5
Falls	Number of falls during 6 months before inclusion	binary: ≤ 2, > 2
Earl_Frac	History of fracture from age of 55 to inclusion	binary
Par_Frac	History of hip-fracture in the parents	binary

Table 2.7: Variables included in the study.

X_1	Y	S=0	S=1
$X_1 = 1$	0	b/2	1-b/2
$X_1 = 1$	1	a	1-b
$X_1 = 0$	0	0	1
$X_1 = 0$	1	b/2	1-b/2

Table 2.8: Probability of S given X_1,Y.

We then train a SVM classifier on biased data under the four weighting schemes discussed in previous experiments and a classifier on unbiased data for reference. All classifiers are trained with Radial Basis Function (RBF), $C = 1$ and kernel size $\sigma = 0.1$, which is chosen through a 5-fold cross validation. We vary the bias level, b, from 0.2 to 0.95 and repeat this experiment 100 times. The result plotted in Figure 2.21 shows that all three importance weighting schemes perform well and help reduce some bias compared to unweighting model. We observe that the improvement of importance weighting is high when bias level is up to 0.9. Above this range, the improvement is reduced. This implies that when selection bias is too strong, importance weighting may increase the variance significantly, reducing the overall performance. However, overall, the importance weighting is effective for this specific data set for selection bias in the experimented range. We also notice that the underlying probability weighting method works slightly better then the other two at all bias level. Finally, IWCV has very little impact on this data set since weighted models outperform unweighted model with a clear margin.

5 Discussion & Conclusion

The results presented in this chapter show that importance weighting method that exploits the assumptions deemed plausible about the sampling mechanism is able to correct or reduce selection bias. The method hinges on the existence of a bias control feature vector, X_s, and an additional (biased-free) sample that allows us to estimate the distribution of X_s. We showed that direct weighting estimation is able to achieve significant improvements in regression and classification accuracy over the unweighted method, using toy problems, benchmarks from UCI, and a prospective cohort study. The correctness of the

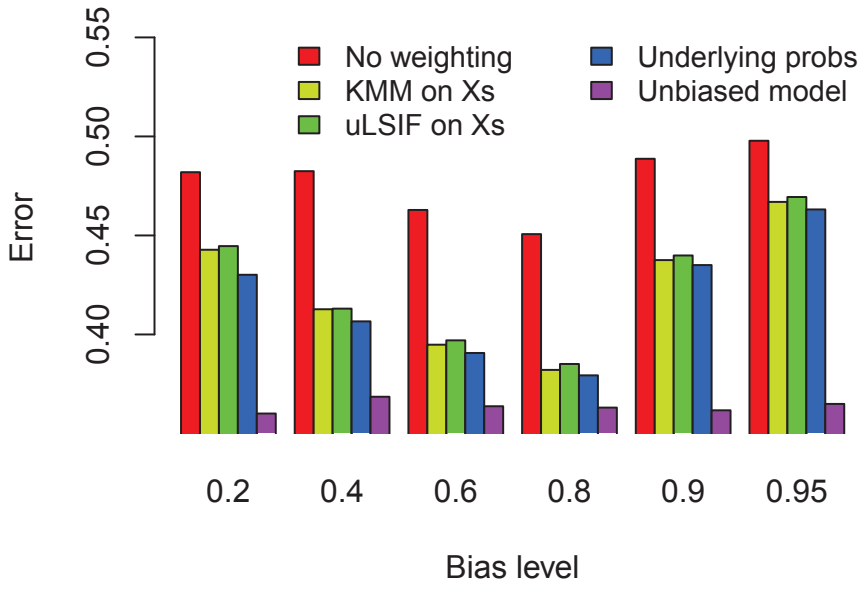


Figure 2.21: Classification performance vs. bias level, b, on hip fracture data set

bias-control feature vector plays a critical role in this improvement. Although we observe in our experiment that the weighting schemes could perform well in many situations where our key assumption is not valid (assuming covariate shift and prior probability shift instead of complete selection bias), there still exist situations where the weighting schemes worsen the performance of learning algorithms when the assumption about the selection mechanism is invalid. So we caution the hurried researcher against correcting for selection bias with invalid assumptions, as it may have harmful consequences on the model performance. In fact, all conclusions are extremely sensitive to which variables we choose for X_s. As the choice of X_s usually reflects the investigator's subjective and qualitative knowledge of statistical influences in the domain, the data analyst must weight the benefit of reducing selection bias against the risk of introducing new bias carried by unmeasured covariates even where none existed before. Nevertheless, we hope this study will convince others about the importance of selection bias correction methods in practical studies and

suggest relevant tools which can be used to achieve that goal.

Another difficulty posed by the selection bias problem is that given the key assumptions about the selection mechanism is valid, the gain in accuracy of the weighting scheme is still data dependent. Fortunately, IWCV can detect when the training sample size reduction effect of the importance weighting dominates its bias correction effect. Therefore, using IWCV, we can reliably decide when to use weighted model to correct selection bias and when to use unweighted model and accept that the training sample size is not sufficient for the importance weighting.

Chapter 3

Improving Importance Weighting in Covariate Shift Correction

Importance weighting has been shown in previous chapter to be an effective technique to deal with selection bias. We has also demonstrated that bias in covariate shift is caused only by the model mispecification and not by the change of decision boundary. Therefore using weighting model to predict every test instance may be excessive since importance weighting usually produces a side effect of effective sample size reduction, which is harmful in many cases. In this chapter, we show analytically that, while the unweighted model is globally more biased than the weighted one, it may locally be less biased on low importance instances. In view of this result, we then discuss a manner to optimally combine the weighted and the unweighted models in order to improve the predictive performance in the target domain. We conduct a series of experiments on synthetic and real-world data to demonstrate the efficiency of this approach. A version of this chapter has been presented at ECML2015 conference (Tran and Aussem [2015a]).

1 Expectation and Local Expectation of Loss

We first define some key concepts used along the chapter and state some results that will support our analysis. We are interested in predicting the output value

y at an input point x using a model $h_\theta(x) = h(x, \theta)$ parameterized by $\theta \in \Theta \subset \mathcal{R}^m$. Under covariate shift assumption, the test inputs follow a different probability distribution $p_{te}(x)$ while the conditional probability distribution of test output $p(y|x)$ remains unchanged. The ratio $\beta(x) = \frac{p_{te}(x)}{p_{tr}(x)}$ is called the *importance* of x. Given a loss function $l(h(x, \theta), y) : \mathcal{Y} \times \mathcal{Y} \to [0, \infty)$, we shall consider throughout this chapter, the following loss functions:

- **EL-Tr**: Expectation of loss over training distribution $p(x, y) = p_{tr}(x)p(y|x)$

$$Loss_0(h_\theta) = E_{x,y\sim p}[l(h(x, \theta), y)] = \int p_{tr}(x) \int p(y|x)l(h(x, \theta), y)dydx$$

- **EL-Te**: Expectation of loss over test distribution $p'(x, y) = p_{te}(x)p(y|x)$

$$Loss_1(h_\theta) = E_{x,y\sim p'}[l(h(x, \theta), y)] = \int p_{te}(x) \int p(y|x)l(h(x, \theta), y)dydx$$

- **EL-IWTr**: Expectation of Importance-weighted loss over training distribution

$$Loss_\beta(h_\theta) = E_{x,y\sim p}[\beta(x)l(h(x, \theta), y)]$$

- **B-LEL-Te**: We then define Local Expectation of loss over test distribution given $\beta(x) \leq B$ of any given hypothesis h_θ:

$$loss(h_\theta, \beta(x) \leq B) = \int_{\beta(x)\leq B} p_{te}(x) \int_{\mathcal{Y}} p(y|x)l(h(x, \theta), y)dydx$$

B-LEL-Te can be seen as a generalization of EL-Te since

$$loss(h_\theta, \beta(x) \leq \infty) = Loss_1(h_\theta)$$

We also define the optimal parameters of EL-Tr, EL-Te and EL-IWTr:

$$\begin{cases} \theta_0 & = \text{argmin}_\theta \, Loss_0(h_\theta) \\ \theta_1 & = \text{argmin}_\theta \, Loss_1(h_\theta) \\ \theta_\beta & = \text{argmin}_\theta \, Loss_\beta(h_\theta). \end{cases}$$

It may easily be shown that EL-IWTr is equal to EL-Te,

$$E_{x,y\sim p}[\beta(x)l(h(x,\theta),y)] = \int p_{tr}(x) \int p(y|x)\frac{p_{te}(x)}{p_{tr}(x)}l(h(x,\theta),y)dydx$$
$$= \int p_{te}(x) \int p(y|x)l(h(x,\theta),y)dydx$$

Therefore, minimizing EL-IWTr is equivalent to minimizing EL-Te. Nonetheless, while h_{θ_β} is globally less biased than h_{θ_0}, we will show next that it is more biased than h_{θ_0} on low-importance instances. Note that B-LEL-Te can be rewritten as:

$$loss(h_\theta, \beta(x) \leq B) = \int_{\beta(x)\leq B} \beta(x) \int_{\mathcal{Y}} p_{tr}(x)p(y|x)l(h(x,\theta),y)dydx$$

Suppose $\beta(x)$ takes on continuous value in $[b_0, b_M]$ where $b_0 > 0$, we may rewrite B-LEL-Te as following:

$$loss(h_\theta, \beta(x) \leq B) = \int_{b_0}^{B} b \int_{\beta(x)=b} \int_{\mathcal{Y}} p_{tr}(x)p(y|x)l(h(x,\theta),y)dydxdb$$

Let $\mathcal{L}(h_\theta, \beta(x) = b) = \int_{\beta(x)=b} \int_{\mathcal{Y}} p_{tr}(x)p(y|x)l(h(x,\theta),y)dydx$, then:

$$loss(h_\theta, \beta(x) \leq B) = \int_{b_0}^{B} b\mathcal{L}(h_\theta, \beta(x) = b)db$$

Similarly, if $\beta(x)$ takes on discrete values in $\{b_i\}_{i=0}^{M}$ such that $b_0 < b_1 < ... < b_M$, we rewrite B-LEL-IWTr as:

$$loss(h_\theta, \beta(x) \leq B) = \sum_{i=0}^{k(B)} b_i\mathcal{L}(h_\theta, \beta(x) = b_i)$$

where $k(B)$ is the largest integer such that $b_{k(B)} \leq B$. From the definitions above, we may write

$$\begin{cases} Loss_1(h_\theta) & = loss(h_\theta, \beta(x) \leq b_M), \\ Loss_0(h_\theta) & = \int_{b_0}^{\infty} \mathcal{L}(h_\theta, \beta(x) = b)db, \quad \text{for continuous } \beta(x), \\ Loss_0(h_\theta) & = \sum_{i=0}^{M} \mathcal{L}(h_\theta, \beta(x) = b_i), \quad \text{for discrete } \beta(x). \end{cases}$$

As aforementioned, a model $h(x, \theta)$ is said to be *correctly specified* if there exist parameter $\theta^* \in \Theta$ such that $h(x, \theta^*) = f(x)$, otherwise it is said to be *misspecified*. It is obvious that if a model is correctly specified, the optimal parameter θ of EL-Tr, EL-Te, and any B-LEL-Te coincide. Therefore, the model that minimizes EL-Tr will perform well on the test data globally (i.e., minimizing EL-Te) as well as locally (i.e., B-LEL-Te) in any region of the form $\beta(x) < B$. Yet, in practice, almost all models are more or less misspecified. So minimizing EL-Tr θ_0 is not necessarily equivalent minimizing EL-Te. Since EL-Te is equal to EL-IWTr, the parameter minimizing of EL-IWTr θ_β, which can be estimated from data, will also minimize EL-Te as shown in Shimodaira [2000], Zadrozny [2004]. However, due to the model misspecification, θ_β does not necessarily minimize B-LEL-Te. In fact, we will prove that there exist some $B^*(h_{\theta_\beta}) \in [b_0, b_M]$ such that B-LEL-Te of θ_β exceeds that of θ_0 by proving a stronger conclusion that for all model h_θ, with $\theta \in \Theta$, there exist some $B^*(h_\theta) \in [b_0, b_M]$ such that B*-LEL-Te of h_θ exceeds that of h_{θ_0}, in other words any h_θ is **locally more biased** than h_{θ_0} when predicting the instances with $\beta(x) \leq B^*$.

In addition, the estimation of θ_β may subject to high variance since it involves instance weighting. Hence the idea to use h_{θ_0} of instead of h_{θ_β} to predict the test instances with $\beta(x) \leq B^*$.

2 Problem Analysis

In this section, we conduct theoretical analyses for a simple and then a more general selection bias mechanism. Those analyses will be used to derive a practical procedure aiming at reducing the bias due to covariate shift with misspecified regression or classification learning models.

We first show how EL-Tr is related to B-LEL-Te,

Lemma 11 *Suppose $\beta(x)$ takes on continuous value in $[b_0, b_M]$ with $b_M > b_0 > 0$, then:*

$$Loss_0(h_\theta) = \frac{1}{b_M} loss(h_\theta, \beta(x) \leq b_M) + \int_{b_0}^{b_M} \frac{1}{B^2} loss(h_\theta, \beta(x) \leq B) dB$$

Proof For continuous $\beta(x)$:

$$\begin{aligned}\int_{b_0}^{b_M} \frac{1}{B^2} loss(h_\theta, \beta(x) \leq B) dB &= \int_{b_0}^{b_M} loss(h_\theta, \beta(x) \leq B) d\left(\frac{-1}{B}\right) \\ &= loss(h_\theta, \beta(x) \leq B)\left(\frac{-1}{B}\right)|_{b_0}^{b_M} - \int_{b_0}^{b_M} \frac{-1}{B} d(loss(h_\theta, \beta(x) \leq B))\end{aligned}$$

Recall that $loss(h_\theta, \beta(x) \leq B) = \int_{b_0}^{B} b\mathcal{L}(h_\theta, b) db$, we have $loss(h_\theta, \beta(x) \leq b_0) = 0$ and $d(loss(h_\theta, \beta(x) \leq B)) = B\mathcal{L}(h_\theta, \beta(x) = B) dB$. Thus:

$$\begin{aligned}\int_{b_0}^{b_M} \frac{1}{B^2} loss(h_\theta, \beta(x) \leq B) dB = &\frac{-1}{b_M} loss(h_\theta, \beta(x) \leq b_M) \\ &+ \int_{b_0}^{b_M} \frac{1}{B}(B\mathcal{L}(h_\theta, \beta(x) = B) dB)\end{aligned}$$

By definition, we have $Loss_0(h_\theta) = \int_{b_0}^{b_M} \mathcal{L}(h_\theta, \beta(x) = B) dB$, so:

$$\int_{b_0}^{b_M} \frac{1}{B^2} loss(h_\theta, \beta(x) \leq B) dB = -\frac{1}{b_M} loss(h_\theta, b_M) + Loss_0(h_\theta)$$

which concludes the proof ∎

A similar results holds in the discrete case.

Corollary 4 *Suppose $\beta(x)$ takes on discrete values $\{b_i\}_{i=0}^{M}$ such that $b_0 < b_1 < ... < b_M$, then:*

$$Loss_0(h_\theta) = \frac{1}{b_M} loss(h_\theta, \beta(x) \leq b_M) + \sum_{k=0}^{M-1} \left(\frac{1}{b_k} - \frac{1}{b_{k+1}}\right) loss(h_\theta, \beta(x) \leq b_k).$$

Proof

$$\sum_{k=0}^{M-1}\left(\frac{1}{b_k}-\frac{1}{b_k+1}\right)loss(h_\theta,\beta(x)\leq b_k)+\frac{1}{b_M}loss(h_\theta,\beta(x)\leq b_M)$$
$$=\left(\frac{1}{b_0}-\frac{1}{b_1}\right)[b_0\mathcal{L}(h_\theta,\beta(x)=b_0)]$$
$$+\left(\frac{1}{b_1}-\frac{1}{b_2}\right)[b_0\mathcal{L}(h_\theta,\beta(x)=b_0)+b_1\mathcal{L}(h_\theta,\beta(x)=b_1)]$$
$$+...$$
$$+\left(\frac{1}{b_{M-1}}-\frac{1}{b_M}\right)[b_0\mathcal{L}(h_\theta,\beta(x)=b_0)+...+b_{M-1}\mathcal{L}(h_\theta,\beta(x)=b_{M-1})]$$
$$+\frac{1}{b_M}[b_0\mathcal{L}(h_\theta,\beta(x)=b_0)+b_1\mathcal{L}(h_\theta,\beta(x)=b_1)+..+b_M\mathcal{L}(h_\theta,\beta(x)=b_M)]$$

$$=b_0\mathcal{L}(h_\theta,\beta(x)=b_0)\left[\left(\frac{1}{b_0}-\frac{1}{b_1}\right)+\left(\frac{1}{b_1}-\frac{1}{b_2}\right)+...+\left(\frac{1}{b_{M-1}}-\frac{1}{b_M}\right)+\frac{1}{b_M}\right]$$
$$+...$$
$$+b_{M-1}\mathcal{L}(h_\theta,\beta(x)=b_{M-1})\left[\left(\frac{1}{b_{M-1}}-\frac{1}{b_M}\right)+\frac{1}{b_M}\right]$$
$$+b_M\mathcal{L}(h_\theta,\beta(x)=b_M)\left[\frac{1}{b_M}\right]$$
$$=\sum_{i=0}^{M}\mathcal{L}(h_\theta,\beta(x)=b_i)$$
$$=Loss_0(h_\theta)\quad\blacksquare$$

In view of Corollary 4, we may now state the following theorem,

Theorem 12 *Suppose there exists two real values, b_0 and b_1, such that $b_0\leq 1\leq b_1$ and a subset $X_0\subset\mathcal{X}$ such that*

$$\beta(x)=\begin{cases}b_0 & \text{if } x\in X_0\\ b_1 & \text{if } x\notin X_0,\end{cases}$$

then there exists a threshold B^* such that:

$$loss(h_{\theta_1},\beta(x)\leq B^*)\geq loss(h_{\theta_0},\beta(x)\leq B^*).$$

In fact, B^* can take any value in $[b_0, b_1)$.

Proof By definition, $Loss_0(h_{\theta_0}) \leq Loss_0(h_{\theta_1})$, using Lemma 11, we may write:

$$\begin{aligned} Loss_0(h_{\theta_0}) &= \frac{1}{b_1} loss(h_{\theta_0}, \beta(x) \leq b_1) + \left(\frac{1}{b_0} - \frac{1}{b_1}\right) loss(h_{\theta_0}, \beta(x) \leq b_0) \\ &= \frac{1}{b_1} Loss_1(h_{\theta_0}) + \left(\frac{1}{b_0} - \frac{1}{b_1}\right) loss(h_{\theta_0}, \beta(x) \leq b_0) \end{aligned}$$

Similarly,

$$Loss_0(h_{\theta_1}) = \frac{1}{b_1} Loss_1(h_{\theta_1}) + \left(\frac{1}{b_0} - \frac{1}{b_1}\right) loss(h_{\theta_1}, \beta(x) \leq b_0)$$

Thus,

$$\begin{aligned} \frac{1}{b_1} Loss_1(h_{\theta_0}) + \left(\frac{1}{b_0} - \frac{1}{b_1}\right) loss(h_{\theta_0}, \beta(x) \leq b_0) \leq \frac{1}{b_1} Loss_1(h_{\theta_1}) \\ + \left(\frac{1}{b_0} - \frac{1}{b_1}\right) loss(h_{\theta_1}, \beta(x) \leq b_0) \end{aligned}$$

Finally,

$$loss(h_{\theta_1}, \beta(x) \leq b_0) - loss(h_{\theta_0}, \beta(x) \leq b_0) \geq \frac{b_0}{b_1 - b_0} \left[Loss_1(h_{\theta_0}) - Loss_1(h_{\theta_1})\right]$$

It is easily shown that the right hand side of the inequality above is non-negative due to the definition of θ_1. It follows that

$$loss(h_{\theta_1}, \beta(x) \leq b_0) - loss(h_{\theta_0}, \beta(x) \leq b_0) \geq 0$$

which, given the assumption about $\beta(x)$, is equivalent to,

$$loss(h_{\theta_1}, \beta(x) = b_0) - loss(h_{\theta_0}, \beta(x) = b_0) \geq 0$$

Thus the Theorem is true when $B^* = b_0$. It is also true for any other $B^* \in [b_0, b_1)$ as a consequence. ∎

When the assumptions of Theorem 12 holds, we say that the covariate shift scheme follows a simple step distribution. The equality in Theorem 12 only

occurs when θ_0 minimizes EL-Te and θ_1 minimizes EL-Tr. Such condition indicates that covariate shift does not have an effect on searching for optimal θ, which is a rare case as shown by other studies. Theorem 12 shows that for *simple step distribution* where inclusion in the training sample is either proportional to b_0^{-1} (over-sampled instances), or to b_1^{-1} (under-sampled instances), h_{θ_0} exhibits a lower bias compared to h_{θ_1} on the low importance test instances. This type of selection bias mechanism is actually quite common. For instance, prospective cohort studies in epidemiology are by design prone to covariate shift because selection criteria are associated with the exposure to potential risk factors.

Theorem 13 *For all $\theta \in \Theta$, there exists a threshold $B^*(h_\theta)$ such that*

$$loss(h_\theta, \beta(x) \leq B^*(h_\theta)) \geq loss(h_{\theta_0}, \beta(x) \leq B^*(h_\theta)) \tag{3.1}$$

$B^*(h_\theta)$ could take any value in the set below:

$$\mathcal{B}^*(h_\theta) = \underset{B}{\operatorname{argmax}}(loss(h_\theta, \beta(x) \leq B) - loss(h_{\theta_0}, \beta(x) \leq B))$$

The equality occurs whenever θ_1 is also a minimum for EL-Tr.

Proof We prove by contradiction that Theorem 13 holds. Assume that inequality 3.1 does not hold for $B^*(h_\theta)$ defined above:

$$loss(h_\theta, \beta(x) \leq B^*(h_\theta)) - loss(h_{\theta_0}, \beta(x) \leq B^*(h_\theta)) < 0 \tag{3.2}$$

By definition of $B^*(h_\theta)$, we may show that, for all $B \in [b_0, b_M]$,

$$loss(h_\theta, \beta(x) \leq B) - loss(h_{\theta_0}, \beta(x) \leq B) < 0$$

Thus, for all $B \in [b_0, b_M]$

$$loss(h_{\theta_0}, \beta(x) \leq B) > loss(h_\theta, \beta(x) \leq B)$$

Now, using Lemma 11 for continuous $\beta(x)$, we have:

$$
\begin{aligned}
Loss_0(h_{\theta_0}) &= \frac{1}{b_M} loss(h_{\theta_0}, \beta(x) \leq b_M) + \int_{b_0}^{b_M} \frac{1}{B^2} loss(h_{\theta_0}, \beta(x) \leq B) dB \\
&> \frac{1}{b_M} loss(h_{\theta}, \beta(x) \leq b_M) + \int_{b_0}^{b_M} \frac{1}{B^2} loss(h_{\theta}, \beta(x) \leq B) dB \\
&= Loss_0(h_{\theta})
\end{aligned}
$$

Hence, $Loss_0(h_{\theta_0}) > Loss_0(h_{\theta})$, contradicts the fact that $\theta_0 = \text{argmin}_\theta \, Loss_0(h_\theta)$ is the optimal hypothesis under the unweighting scheme and $\theta \neq \text{argmin}_\theta \, Loss_0(h_\theta)$.

If the two terms in inequality3.1 are equal, then we can prove similarly that $Loss_0(h_{\theta_0}) = Loss_0(h_{\theta})$, which implies that θ_1 is also a minimal solution of EL-Tr. The demonstration for discrete $\beta(x)$ values follows similarly. ∎

Theorem 13 states that any model h_θ with $\theta \in \Theta$ is outperformed by h_{θ_0} learned from the unweighted training samples in terms of bias when predicting examples with $\beta(x) \leq B^*(h_\theta)$. This is also applied to model h_{θ_β} which minimizes EL-IWTr. In addition, the estimation of θ_β may exhibit a higher variance due to the effective sample size reduction as discussed in Cortes et al. [2010]; Gretton et al. [2009]. These results altogether suggest that h_{θ_0} should be preferred to h_{θ_β} for predicting the instance's outputs in the region $\beta(x) \leq B^*(h_\theta)$, termed **low-importance region**. Therefore, for any learning task with covariate shift, we shall train two distinct models, one with and the other without the importance weighting scheme. Then, we shall use the latter to predict instances satisfying $\beta(x) \leq B^*(h_\theta)$ and use the former to predict the remaining instances. The optimal value for $B^*(h_\theta)$ may be estimated from the training data. The set of all possible empirical threshold $\hat{B}^*(h_{\theta_\beta})$ can be obtained empirically by solving the following problem :

$$
\hat{\mathcal{B}}^*(h_\theta) = \underset{B}{\text{argmax}} \frac{1}{n} \sum_{\substack{i \in \{1,..,n\} \\ \beta(x_i) \leq B}} \beta(x_i)[l(y_i, h(x_i, \theta_\beta)) - l(y_i, h(y_i, \theta_0))] \tag{3.3}
$$

As n grows to infinity, it follows from the law of large numbers that,

$$\hat{\mathcal{B}}^*(h_\theta) \to \mathcal{B}^*(h_\theta)$$

Therefore, $B^*(h_{\theta_\beta})$ could be estimated empirically either from training data or by cross validation. In this study, we use a 5-fold importance weighted cross validation to estimate $B^*(h_{\theta_\beta})$ as suggested in Sugiyama et al. [2007b]. It should be emphasized that $B^*(h_{\theta_\beta})$ is not necessarily unique. For instance, any value between b_0 and b_1 in Theorem 12 is admissible as mentioned earlier. For a fixed h_θ equation 3.3 can be solved by first calculate:

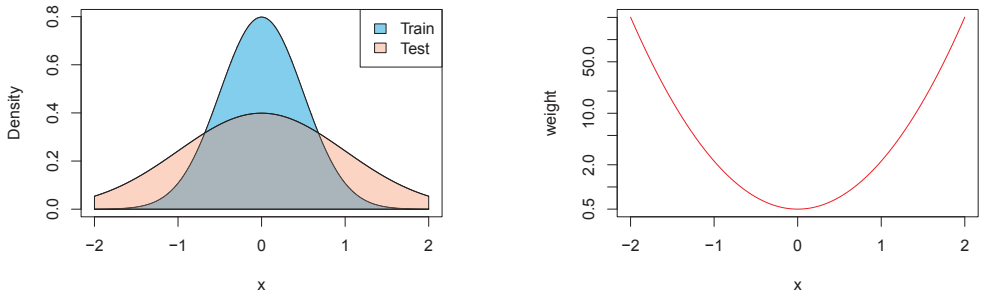
$$\hat{L}(B) = \frac{1}{n} \sum_{\substack{i \in \{1,..,n\} \\ \beta(x_i) \leq B}} \beta(x_i)[l(y_i, h(x_i, \theta_\beta)) - l(y_i, h(y_i, \theta_0))]. \tag{3.4}$$

Then $\hat{\mathcal{B}}^*(h_\theta)$ is the maximum of $\hat{L}(B)$. The complexity of calculating $\hat{L}(B)$ and finding the maximum is only $\mathcal{O}(n)$ since for any $B' > B$, $\hat{L}(B)$ can be expressed as following:

$$\hat{L}(B') = \hat{L}(B) + \frac{1}{n} \sum_{\substack{i \in \{1,..,n\} \\ B < \beta(x_i) \leq B'}} \beta(x_i)[l(y_i, h(x_i, \theta_\beta)) - l(y_i, h(y_i, \theta_0))].$$

3 Performance of Hybrid Model vs. Importance Weight

In this section, we assess the ability of our "hybrid approaches" to reduce the learning bias under covariate shift based on Theorem 13 and 12. We employed two strategies to estimate the importance weights: one is based explicitly on the true bias mechanism, the other is based on Unconstrained Least-Square Importance Fitting (uLSIF), a method that estimates $\beta(x)$ in both training and test data. Our method is not applicable to KMM since it requires the importance weight to be estimated in both the training and the test data while KMM can only estimate the former. We test our approaches on several real world benchmark data sets, from which the training examples are selected according to various biased sampling schemes as suggested in Kanamori et al. [2009].



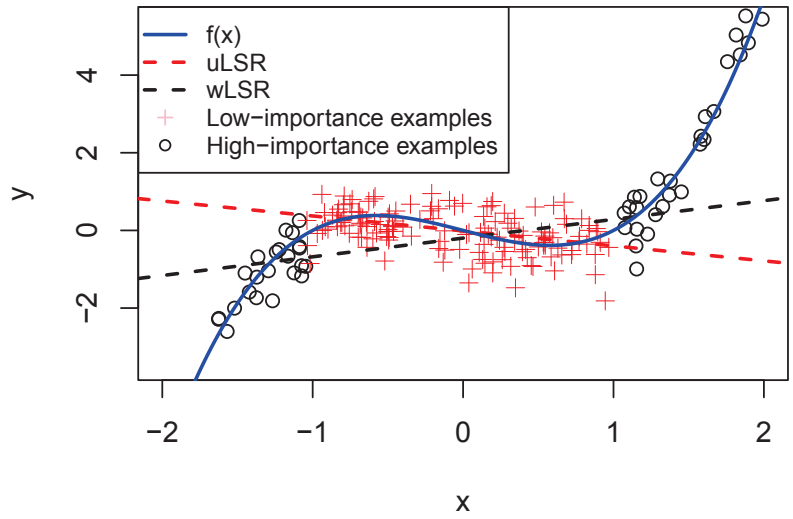
(a) Input density distribution

(b) True importance weights

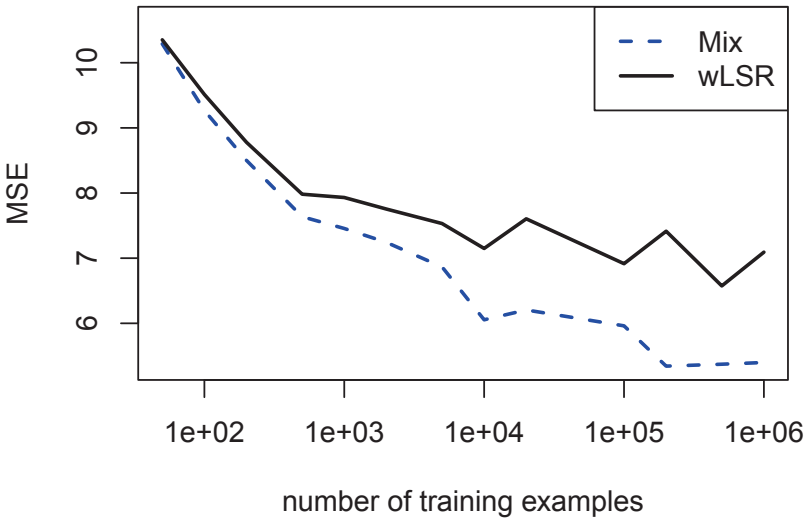
Figure 3.1: Distribution change and true importance weight in the toy problem.

3.1 Toy Regression Problem

Consider the following training data generating process: $x \sim N(\mu_0, \sigma_0)$ and $y = f(x)+\epsilon$, where $\mu_0 = 0$, $\sigma_0 = 0.5$, $f(x) = -x+x^3$, and $\epsilon \sim N(0, 0.3)$. In the test data, we have the same relationship between x and y but the distribution of the covariate x is shifted to $x \sim N(\mu_1, \sigma_1)$, where $\mu_1 = 0$, $\sigma_1 = 1$. The training and test distributions, along with their ratio are depicted in Figure 3.1a and 3.1b. The minimization of EL-Tr is obtained using the unweighted Least Square Regression (uLSR) method for the normal regression while minimization of EL-Te is performed by the weighted Least Square Regression (wLSR). As shown in Shimodaira [2000], wLSR is unbiased thus it should perform better than uLSR, which is biased, on test data. However, as can be seen in Figure 3.2a, uLSR (red dashed line) seems to better approximate the $y = f(x)$ curve (in blue) than wLSR (black dashed line) on instances in the interval $(-1, 1)$. As may be seen in Figure3.2b, the hybrid model that optimally combines wLSR and uLSR, based on Theorem 1, achieves a lower Mean Square Error (MSE) compared to wLSR. The experiment was repeated 30 times for each number of sample size. It should be noted that the hybrid model always outperforms the weighted model and the gain in performance on the test set is more noticeable for larger training sizes.



(a) True function, uSLR and wSLR on test data.



(b) MSE vs training sample size

Figure 3.2: An illustrative example of fitting a function f(x) using a linear model with or without the weight importance scheme (wLSR/uLSR) and a combination of both (termed "Mix").

3.2 Simple Step Sample Selection Distribution

In this experiment, we consider a simple step distribution with known or estimated selection probabilities and we apply this selection scheme on a variety of UCI data sets in order to assess the efficiency of our bias correction procedure in more realistic scenarios. We use a SVM classifier for both classification and regression tasks. Experiments are repeated 50 times for each data set. In each trial, we randomly select an input feature x^c to control the bias along with 100-300 training samples and 200-900 examples without label. We then apply the following single step probability distribution as discussed in Theorem 12,

$$P(s=1|x=x_i^c)=p_s=\begin{cases} p1=0.9, & \text{if } x_i^c \leq mean(x^c) \\ p2=\frac{0.9}{1+exp(r)}, & \text{otherwise} \end{cases}$$

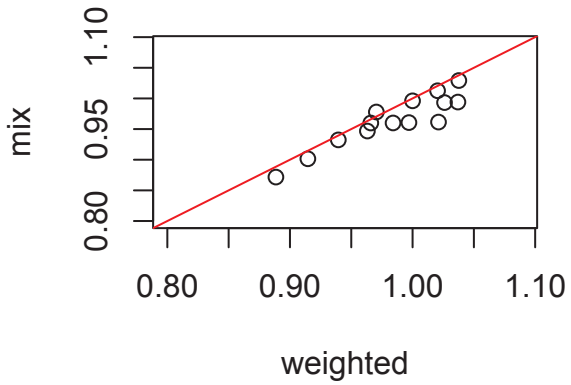
where r is a parameter that controls the strength of the selection bias. In each trial r takes a random value from a normal distribution $N(2, 0.1)$. With these parameters, the selection probability for instances having an x^c value (e.g. a degree of exposure to some risk factor) above the mean is between 7 to 10 times smaller than for those having of a lower value. This is a scenario that typically arises in epidemiological cohort studies when subjects are includes in the study according to some exposure factor. Consider the two following weighting schemes. The first one: $\beta = p_{te}(x)/p_{tr}(x) = p(s=1)/p(s=1|x) \sim 1/p_s$ assumes that the bias mechanism is known exactly.

$$\beta(x) \sim p_s^{-1} \sim \begin{cases} b1=1, & \text{if } x_i^c \leq mean(x^c) \\ b2=1+exp(r), & \text{otherwise.} \end{cases}$$

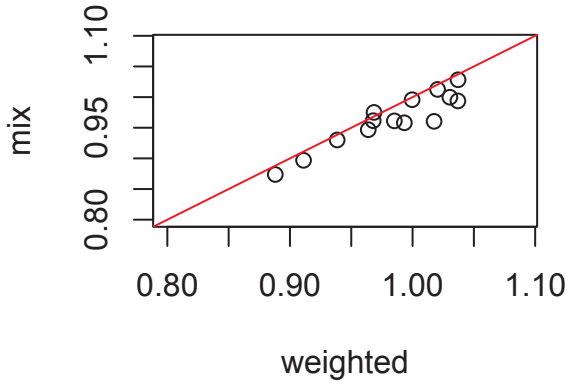
In practice, however, the selection probability is rarely known exactly. So let us assume that the estimation of β is subject to some error and let us consider the following approximate weighting scheme:

$$\hat{\beta}(x) \sim p_s^{-1} \sim \begin{cases} b1=1, & \text{if } x_i^c \leq mean(x^c) \\ b2=1+exp(\hat{r}), & \text{if otherwise} \end{cases}$$

where $\hat{r} = r + \mathcal{N}(0, 0.1)$ is our noisy estimate of r. For each weighting scheme, we fit a true weighted model (denoted as P in Table 3.1) and an ap-



(a) Importance weight with true selection probability



(b) Importance weight with uLSIF

Figure 3.3: MSE gain (over weighted model) of the mix data vs. MSE gain mix model under simple step distribution covariate shift. Points below the diagonal line indicate that the mix data outperforms the mix model. The importance weight is estimated based on the true selection probability (Figure a) and based on the estimated selection probability (Figure b).

proximated weighted model (denoted as $\hat{P}$). As $p_1 < 1$ and $p_2 > 1$, our weighting mechanism satisfies the assumptions of Theorem 12, so we set $B^* = 1$. We report the mean square errors (MSE) in Table3.1. All values are normalized by the MSE of the unweighted model (our gold standard). As may be seen from the plots in Figure3.3a and 3.3b, the combined models outperform the weighted ones. That is, when using either exact probability ratio, the results obtained with P_{mix} are better than that of P. The same observation can be made when the estimated probability ratios are used instead (i.e., $\hat{P}_{mix}$ versus $\hat{P}$) except on the Banknote data set. The gain is significant at the significance level 5% using the Wilcoxon signed rank test.

Table 3.1: Mean test error averaged over 50 trials for different weighting schemes on UCI data sets with the simple selection distribution. Data sets marked with '*' are regression problems. P denotes the weighting scheme using the true selection probability and $\hat{P}$ denotes the weighting scheme using a noisy selection probability. For each pair of weighted and mix models, the better prediction value is highlighted in boldface.

Data set	No weighting	P	P mix	$\hat{P}$	$\hat{P}$ mix
India diabetes	1.000 ± 0.020	0.966 ± 0.019	**0.960** ± 0.018	0.968 ± 0.019	**0.962** ± 0.018
Ionosphere	1.000 ± 0.128	0.915 ± 0.105	**0.902** ± 0.107	0.911 ± 0.104	**0.897** ± 0.106
BreastCancer	1.000 ± 0.039	1.020 ± 0.044	**1.013** ± 0.044	1.020 ± 0.044	**1.013** ± 0.043
GermanCredit	1.000 ± 0.008	1.000 ± 0.007	**0.996** ± 0.008	1.000 ± 0.008	**0.996** ± 0.008
Australian credit	1.000 ± 0.006	0.963 ± 0.008	**0.947** ± 0.010	0.964 ± 0.008	**0.947** ± 0.010
Mushroom	1.000 ± 0.068	0.090 ± 0.057	**0.872** ± 0.060	0.888 ± 0.058	**0.874** ± 0.056
Congressional Voting	1.000 ± 0.033	1.026 ± 0.039	**0.993** ± 0.038	1.030 ± 0.038	**1.000** ± 0.037
Banknote	1.000 ± 0.040	**0.970** ± 0.043	0.978 ± 0.038	**0.969** ± 0.042	0.975 ± 0.039
Airfoil self noise*	1.000 ± 0.023	0.997 ± 0.015	**0.961** ± 0.012	0.993 ± 0.015	**0.958** ± 0.012
Abalone*	1.000 ± 0.032	0.984 ± 0.020	**0.960** ± 0.020	0.985 ± 0.021	**0.961** ± 0.020
Auto MGP*	1.000 ± 0.084	0.939 ± 0.066	**0.933** ± 0.067	0.939 ± 0.066	**0.930** ± 0.067
Boston Housing*	1.000 ± 0.057	1.037 ± 0.053	**0.994** ± 0.050	1.037 ± 0.053	**0.994** ± 0.050
Space GA*	1.000 ± 0.009	1.021 ± 0.007	**0.962** ± 0.008	1.018 ± 0.008	**0.961** ± 0.008
Cadata*	1.000 ± 0.013	1.038 ± 0.022	**1.029** ± 0.017	1.037 ± 0.022	**1.029** ± 0.017

3.3 General Selection Mechanisms

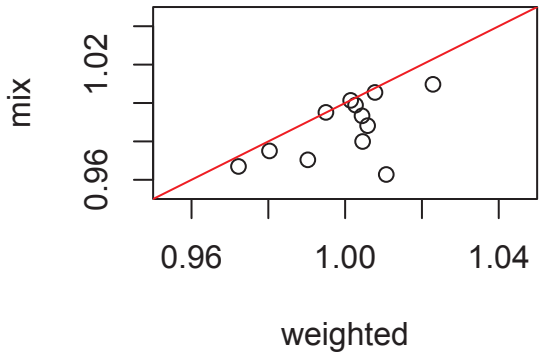
In this last experiment, we use the same setting as above but we use a more general distribution:

$$P(s=1|x=x_i^c) = ps = \begin{cases} p1 = 0.9 \text{ if } x_i^c \leq mean(x^c) \\ p2 = 0.1 \text{ if } x_i^c > mean(x^c) + 0.8 \times 2\sigma(x^c) \\ p3 = 0.9 - \frac{x_i^c - mean(x^c)}{2\sigma(x^c)} \text{ otherwise.} \end{cases}$$

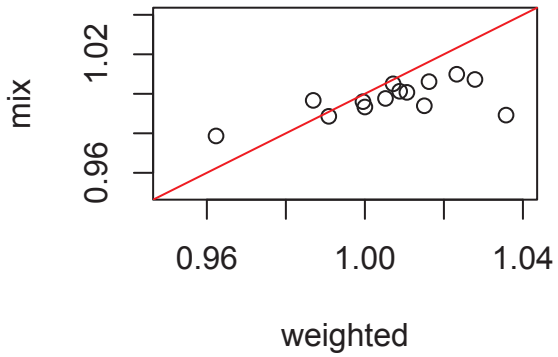
where $\sigma(x^c)$ denotes the standard deviation of x^c. As may be observed, the assumptions required in Theorem 12 do not hold anymore with this more general sample selection distribution. According to Equation 3.3, we need to estimate $\hat{B}^*(h_\theta)$ empirically from data. We consider again two importance weighting schemes: one is based on the true underlying probability and is referred to as P, while the other is based on the uLSIF estimator. As may be observed from Table 3.2, Figure 3.4a, and Figure 3.4b that performances of the hybrid models are significantly improved with respect to the weighted models, except with the Congressional Voting and Banknote data sets.

4 Conclusions

In this chapter, we showed that the standard importance weighting approach used to reduce the bias due to covariate shift can easily be improved when misspecified training models are used. Considering a simple class of selection bias mechanisms, we proved analytically that the unweighted model exhibits a lower prediction bias compared to the globally unbiased model in the low importance input subspace. Even for more general covariate shift scenarios, we proved that there always exist a threshold for the importance weight below which the test instances should be predicted by the globally biased model. In view of this result, we proposed a practical procedure to estimate this threshold and we discussed a simple procedure to combine the weighted and unweighted prediction models. The method was shown to be effective in reducing the bias on both synthetic and real-world data.



(a) Importance weight with true selection probability



(b) Importance weight with uLSIF

Figure 3.4: MSE gain (over weighted model) of the mix data vs. MSE gain mix model under general distribution covariate shift. Points below the diagonal line indicate that the mix data outperforms the mix model. The importance weight is estimated based on the true selection probability (Figure a) and based on the estimated selection probability (Figure b).

Table 3.2: Mean test error averaged over 50 trials for different weighting schemes on UCI data sets with the general selection distribution. Data sets marked with * are for regression problems. P denotes the weighting scheme based on the true selection probability and uLSIF denotes the weighting scheme using the uLSIF estimator. For each pair of weighted and mix models, the better prediction value is highlighted in boldface.

Data set	No weighting	P	P mix	uLSIF	uLSIF mix
India diabetes	1.000 ± 0.021	0.980 ± 0.018	**0.975** ± 0.018	1.016 ± 0.021	**1.006** ± 0.021
Ionosphere	1.000 ± 0.087	1.006 ± 0.087	**0.988** ± 0.085	1.028 ± 0.093	**1.007** ± 0.087
BreastCancer	1.000 ± 0.019	1.004 ± 0.018	**0.993** ± 0.019	1.000 ± 0.018	**0.993** ± 0.019
GermanCredit	1.000 ± 0.008	1.003 ± 0.008	**0.999** ± 0.008	1.009 ± 0.008	**1.001** ± 0.008
Australian credit	1.000 ± 0.009	0.972 ± 0.007	**0.967** ± 0.007	1.007 ± 0.008	**1.005** ± 0.008
Mushroom	1.000 ± 0.558	1.011 ± 0.054	**0.963** ± 0.051	0.991 ± 0.054	**0.989** ± 0.054
Congressional Voting	1.000 ± 0.037	1.023 ± 0.036	**1.010** ± 0.037	**0.987** ± 0.036	0.997 ± 0.036
Banknote	1.000 ± 0.060	1.083 ± 0.057	**0.962** ± 0.062	**0.962** ± 0.061	0.979 ± 0.058
Airfoil self noise*	1.000 ± 0.007	0.995 ± 0.007	**0.995** ± 0.007	1.011 ± 0.008	**1.001** ± 0.008
Abalone*	1.000 ± 0.007	1.001 ± 0.008	1.001 ± 0.007	1.005 ± 0.007	**0.998** ± 0.006
Auto MGP*	1.000 ± 0.026	0.990 ± 0.025	**0.970** ± 0.025	1.015 ± 0.027	**0.994** ± 0.026
Boston Housing*	1.000 ± 0.043	0.984 ± 0.031	**0.940** ± 0.032	1.036 ± 0.040	**0.989** ± 0.042
Space GA*	1.000 ± 0.006	1.005 ± 0.005	**0.980** ± 0.006	1.000 ± 0.005	**0.996** ± 0.005
Cadata*	1.000 ± 0.012	1.008 ± 0.013	**1.006** ± 0.012	1.023 ± 0.013	**1.010** ± 0.012

Chapter 4

Selection Bias as a Missing Data Problem

Importance weighting, even when being used partially as in previous chapter still reduces the effective sample size, which is harmful when the initial training sample size is already small. In this chapter, we show that there exists a weighting scheme on the unlabeled data such that the combination of the weighted unlabeled data and the labeled training data mimics the test distribution. We further prove that the labels are missing at random in this combined data set and thus can be imputed safely in order to mitigate the undesirable sample-size-reduction effect of importance weighting. A series of experiments on several synthetic and real-world data sets are conducted to demonstrate the efficiency of our approach. A version of this chapter has been presented at ESANN2017 conference (Tran and Aussem [2017]).

1 Introduction

In previous chapter we discussed the fact that reweighting methods do not necessarily improve the prediction accuracy as they reduce the effective training sample size and presented the hybrid model approach that used partially the weighted model and partially the unweighted model on test data. The reduction of sample size becomes more severe when the initial training sample size is small even for the hybrid model. Another drawback of current importance weighting approaches is that the unlabeled data set are usually discarded once

the importance weights are estimated. Some information is lost in the process. To our best knowledge, non of the existing methods to deal with covariate shift takes advantage of the unlabeled data in the training phase given that the importance weight was estimated.

In this chapter we show that there exists a weighting scheme on the unlabeled data so that a combination of these weighted unlabeled data and original training data forms a new data set, called the *hybrid data set*, that have label missing at random (MAR). The missing values of label in the hybrid data are then imputed using state of the art imputation methods for MAR data. This approach is particularly useful when very few labeled data are provided.

2 The Hybrid Data Method

The importance weight estimation almost always requires some unlabeled data from general population to provide an estimation of the input distribution of test data. In importance weighting approaches, after importance weight is estimated, the unlabeled data set is usually discarded, causing a lost of information that could be helpful in reducing covariate shift without increasing much variation especially when initial training sample size is small. In this section, we show that there exists a weighting scheme on unlabeled data so that a combination of these weighted unlabeled data and original data forms a new data set, called hybrid data set, free from covariate shift. We also show that the missing label in the new hybrid data set satisfies missing at random condition. Therefore it can be imputed using state of the art methods for missing at random problem. Assuming that the unlabeled data follow the input distribution $p_{te}(x)$ of test data, we first derive, in this Section, a weighting scheme $w(x)$ on the unlabeled data so that a combination of these weighted unlabeled data and the original training data forms a new data set that mimics $p_{te}(x)$. Our main result can be stated as follows:

Theorem 14 *Given n_1 training examples and n_2 unlabeled examples, that follow distributions $p_{tr}(x)$ and $p_{te}(x)$ respectively, there exists a weighting scheme*

$$w(x) = \frac{n_1}{n_2}\left(\max_{x\in\mathcal{X}} \frac{p_{tr}(x)}{p_{te}(x)} - \frac{p_{tr}(x)}{p_{te}(x)}\right)$$

on the unlabeled examples such that the mixture of n_1 unweighted training examples and n_2 weighted unlabeled examples follows the distribution $p_{te}(x)$.

Proof The hybrid data set follows a mixture distribution

$$p_{tr}(x)\frac{n_1}{n_1+n_2\int w(x)p_{te}(x)dx}+p_{te}(x)\frac{w(x)}{\int w(x)p_{te}(x)dx}\times\frac{n_2\int w(x)p_{te}(x)dx}{n_1+n_2\int w(x)p_{te}(x)dx}$$

Imposing this mixture to be $p_{te}(x)$ and solving for $w(x)$, we have:

$$w(x)=\frac{n_1}{n_2}\left(C-\frac{p_{tr}(x)}{p_{te}(x)}\right)$$

Where C is any constant that satisfies $w(x)\geq 0$ for all $x\in\mathcal{X}$ since $w(x)$ is a non-negative coefficient. That gives $C\geq\max_{x\in\mathcal{X}}\frac{p_{tr}(x)}{p_{te}(x)}$. If we increase C by ΔC, the weight of every unlabeled example will be increased by $\frac{n_1}{n_2}\Delta C$. The choice of the constant C is only depends on how much weight we would like to attribute to the unlabeled data. Unlike the semi-supervised learning, we don't assume any relationship between $p_{tr}(x)$ and $p(y|x)$, We only use the unlabeled data to improve prediction accuracy indirectly through correcting the input distribution. When there is no covariate shift, in the semi-supervised learning setting, (Castelli and Cover [1996]) showed that the labeled examples are exponentially more valuable than the unlabeled examples in constructing classification rules. Therefore, we argue that the quantity of the unlabeled data in the final hybrid training data set should be minimized using a minimal weight that allows the selection bias correction.

Since the effective number of unlabeled data increase linearly with C, we will set it as small as possible, $C=\max_{x\in\mathcal{X}}\frac{p_{tr}(x)}{p_{te}(x)}$. Finally,

$$w(x)=\frac{n_1}{n_2}\left(\max_{x\in\mathcal{X}}\frac{p_{tr}(x)}{p_{te}(x)}-\frac{p_{tr}(x)}{p_{te}(x)}\right).\quad\blacksquare$$

We have shown that the resulting hybrid data set is unbiased but it still contains missing labels. There are circumstances under which even the best designed study is jeopardized by non-missing-at-random data. The following result shows the labels are in fact MAR:

Theorem 15 *The labels in the hybrid data set obtained from the weighting scheme in Theorem 14 are missing at random.*

Proof From Theorem 14, the hybrid data set follows the marginal distribution $p_{te}(x)$ of the test data. In addition, because of the definition of covariate shift, Let $R_Y = 1$ denotes "Y is missing" and 0 otherwise, it is easily shown that $p(y|x, R_Y = 1) = p(y|x, R_Y = 0) = p(y|x)$, which is the definition of the MAR missing mechanism.

The methods for correcting covariate shift bears similarity to the techniques employed in semi-supervised learning. The latter usually make further assumptions on the data distribution p, more specifically on the relationship between $p(y|x)$ and $p(y)$ (Zhu [2005]). When the models used for representing $p_{tr}(x)$ and $p(y|x)$ do not share common parameters, semi-supervised learning methods cannot improve the predictive performance. For example, transductive support vector machines (Chapelle et al. [2006]; Joachims [1999]), assumes that the data contains clusters that have homogeneous labels and as a result the decision boundary has to lie in low density regions. In contrast, generative models, (Baluja [1999]; Castelli and Cover [1996]) assumes that $p(x|y)$ is a mixture of distributions, allowing the decision boundary to go through some denser regions. The success of a semi-supervised learning method depend on whether the data distribution can be accurately approximated by a parameterized model and the degree to which the class distributions overlap (Zhu [2005]). On the other hand, the covariate shift supposes the input training and test distributions are different and make no further assumption on the relationship between $p_{tr}(x)$ and $p(y|x)$. That differentiate our approach from semi-supervised learning methods.

2.1 Predictive Mean Matching for the Missing Data Imputation

Give a hybrid data set that is MAR, our next step is to impute the missing labels. Missing data imputation is a well-studied topic in the statistical analysis. From the many references, we choose Predictive Mean Matching (PMM), which was first presented in Little [1988] and proved to be successful with missing data imputation, as was shown to be robust to the misspecification of the imputation model in Morris et al. [2014]. For the covariate shift problem, if we can choose a correctly specified model in the first place, there will be no learn-

ing bias. However due to the lack of domain knowledge, it is safer to assume that the imputation model for the unlabeled data is misspecified. Robustness of imputation models to misspecification is an important criterion that should be considered with great care when choosing an imputation method.

For a data set that only has missing labels as in our hybrid data set, PMM first estimates a linear regression of y on x and produces a posterior predictive distribution of the coefficient vector α that specifies the linear regression. A set of coefficient α^* is drawn from that posterior distribution. Using α^*, PMM predicts values of all cases (labeled and unlabeled). For each case with missing label x_u, we determine a set of five labeled cases $\{(x_t; y_t) : t = 1, ..., 5\}$ whose predicted labels are closest to the predicted label of x_u. One of five values in $\{y_t) : t = 1, ..., 5\}$ is randomly selected to be an imputed value of the missing case x_u. For a new imputed data set in multiple imputation, the process is repeated from drawing a new set of coefficient α^* from posterior predictive distribution.

3 Performance of Hybrid Data vs. Hybrid Model and Weighting Models

In this section, we assess the ability of our hybrid data approach to reduce the model variance due to importance weighting in the covariate shift bias reduction process. We use two strategies to estimate the importance weights $\beta(x) = \frac{p_{te}(x)}{p_{tr}(x)}$: the first is based explicitly on the true bias mechanism, the second is based on Unconstrained Least-Square Importance Fitting (uLSIF). We first study a toy regression problem to show whether covariate shift corrections based on our method can reduce the prediction error on the test set when the learning model is misspecified and the training sample size is small. Then we test our approach on real world benchmark data sets corrupted by a simple covariate shift bias selection mechanism.

3.1 Toy Regression Problem

Consider the following training data generating process: $x \sim N(\mu_0, \sigma_0)$ and $y = f(x) + \epsilon$, where $\mu_0 = 0.5$, $\sigma_0 = 0.5$, $f(x) = -x + x^3$, and $\epsilon \sim N(0, 0.3)$. In the

test data, the same relationship between x and y holds but the distribution of the covariate x is shifted because of the selection bias that causes the examples to be selected with a probability depending on x:

$$p(s=1|x) = \begin{cases} 4x^2 \text{ if } 4x^2 \in [0.01, 1] \\ 0.01 \text{ if } 4x^2 \leq 0.01 \\ 1 \text{ otherwise.} \end{cases}$$

The training and test distributions, along with their ratio are plotted in Fig. 4.1a and 4.1b. Least Square Regression is used to train a linear model to predict output y from x. We first investigate the effect of unlabeled data quantity on the performance of the hybrid data. As may be seen in Figure 4.1c, the Mean Square Error (MSE) of the regression model drops as the unlabeled-labeled sample size ratio, n_2/n_1, increases. At first, as more unlabeled data are used, n_2/n_1 varies from 0 to 1, the improvement is clearly noticeable. The smaller the initial training sample size is, the larger the margin of the improvement gets because the hybrid data approach is more effective at preserving the effective sample size. When n_2/n_1 varies from 1 to 2, a further but moderate improvement is observed. Again, the more unlabeled data are used, the smaller the weights of the unlabeled example according to Theorem 14. Consequently, the imputation variance contributes less to the final prediction error. Finally, when the value of n_2/n_1 is large enough, no further improvement is noticed since the unlabeled data are only helpful in reducing the distribution mismatch up to the point when the hybrid data mimics closely the test data distribution. This behavior is contrary to semi-supervised learning methods whose the predictive performance tend to increase as more unlabeled data are used given that their assumptions is correct. We will use in the toy problem an unlabeled data set five times larger than the labeled data set for and only twice as large in real-world data set experiments. We shall now compare the "hybrid-data approach" against respectively the unweighted, weighted, and hybrid-model approaches. In the hybrid-model approach presented in previous chapter, the predictive performance in some regions of the input space is improved by combining the weighted and the unweighted models. The average MSE of these models over 100 repeated trials is reported for every training sample size in Figure 4.2. The unweighted model (black solid line) serves as a baseline. As

expected, it performs worse than the other models. When the training sample size is large enough (say, more than 300) the hybrid-model method achieves a lower MSE because it has the lowest bias as suggested by Theorem 13. On the other hand, the hybrid-data method (blue solid line) outperforms any other method with a large margin when the training sample size is small. As sample size increases, the variance reduction becomes less significant, the hybrid data's performance is similar to that of the weighted model. From these observations, we conclude that the hybrid-data approach is more effective when the sample size is small.

3.2 Experiments on Real-world Data sets

In this series of experiments, we consider the learning problems under a covariate shift induced by an artificial selection mechanism with known or estimated selection probabilities. We apply this selection scheme on a variety of UCI data sets in order to assess the efficiency of our approach in more realistic scenarios. We use a SVM classifier for both classification and regression tasks. Experiments are repeated 50 times for each data set. In each trial, we randomly select 100 training examples, 200 unlabeled examples, and an input feature x^c that controls the probability of an an example to be selected into the training set as follows:

$$p(s=1|x=x_i^c) = ps = \begin{cases} p1 = 0.9 \text{ if } x_i^c \leq mean(x^c) \\ p2 = 0.1 \text{ if } x_i^c > mean(x^c) + 0.8 \times 2\sigma(x^c) \\ p3 = 0.9 - \frac{x_i^c - mean(x^c)}{2\sigma(x^c)} \text{ otherwise.} \end{cases}$$

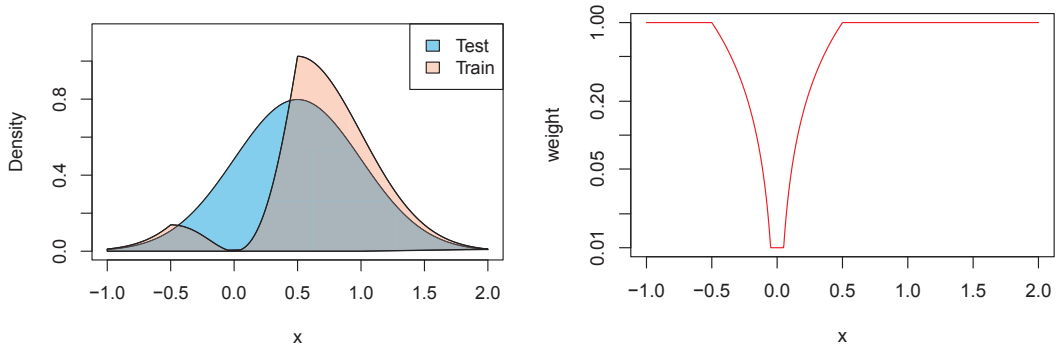
where $\sigma(x^c)$ denotes the standard deviation of x^c. Each of three approaches, namely the weighted data, hybrid model, and hybrid data is applied with both the true important weights and the important weights estimated with uLSIF. The MSE of each model is normalized by that of the unweighted model (our gold standard) and plotted in Fig.4.3 and 4.4. As may be observed, the hybrid data approach always outperforms the weighted model by a noticeable margin except when ulSIF is used on the Cadata data set. However, we suspect that the estimation of importance ratio on this data set fails as all other methods

using ulSIF performs worse than the basic unweighted method on this data set. The hybrid data method also outperforms the hybrid model method in most situations, except on the Australian credit data set with true important weight and on the Cadata and Ionosphere data sets with ulSIF. Our results strongly suggest that our bias correction method combined with missing at random label imputation is effective at increasing the prediction performance when few labeled data are available.

4 Conclusion and Open Problems

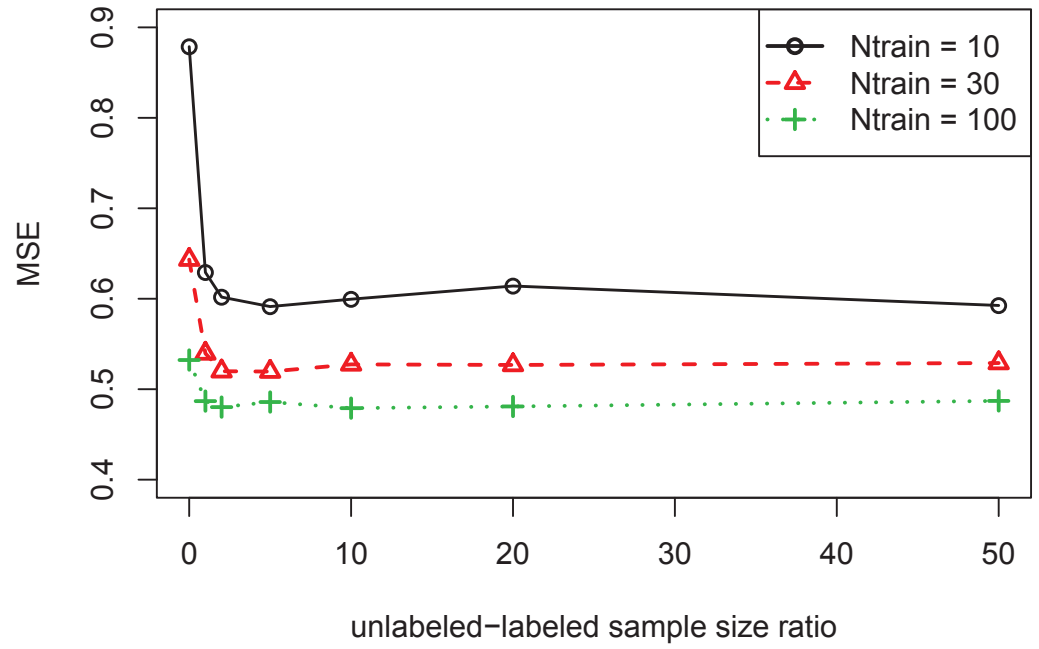
We have shown that given training data with covariate shift and unbiased unlabeled data there exists a weighting scheme on the unlabeled data such that the combination of the weighted unlabeled data and the labeled training data mimics the test distribution. The fact that the labels are missing at random in this combined data set allows effective imputation in order to mitigate the undesirable sample-size-reduction effect of importance weighting. Both experiments on synthetic and real-world data demonstrate the efficiency of our approach.

In our study, PMM has shown to be an effective method for imputation given the combined data set is missing at random. According to whether there is assumption about the relationship between $p(y|x)$ and $p_{tr}(x)$ or not, we can take semi-supervised learning methods as alternative approaches to use the combined data set more efficiently. However, we have to keep in mind that the predictive performance of semi-supervised learning methods depends heavily on matching of problem structure with model assumption. Therefore, a good understanding of the specific problem is required to use semi-supervised learning methods effectively for covariate shift problem.



(a)

(b)



(c)

Figure 4.1: A function f(x) is fitted by a linear model: a) Input density distribution; b) True importance weights; c) MSE of hybrid-data model vs. unlabeled/labeled ratio for different training sample sizes.

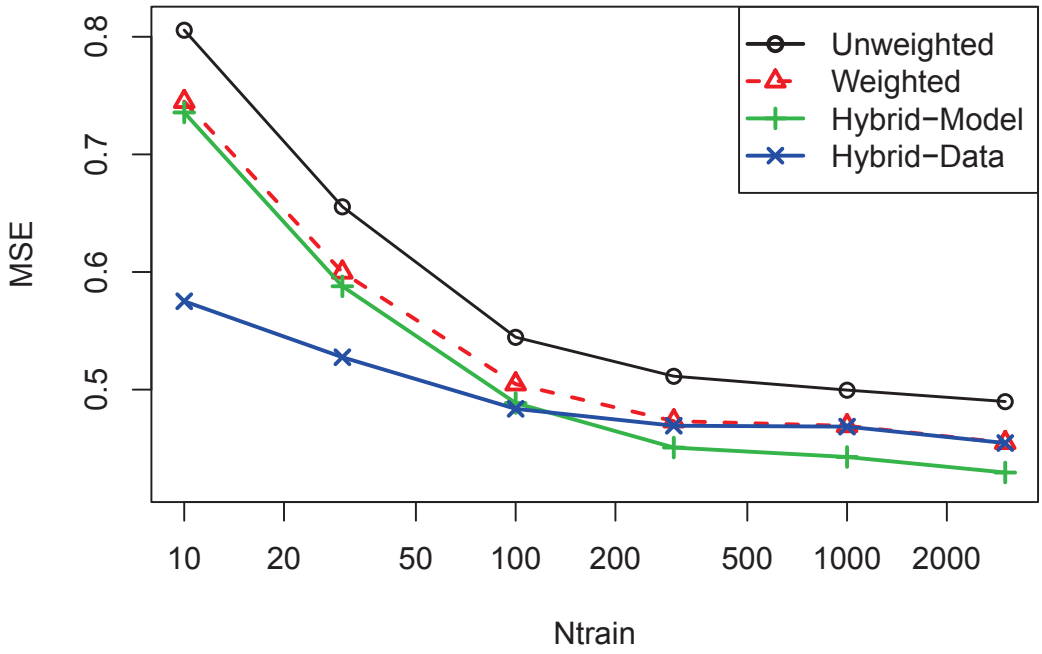


Figure 4.2: Toy regression problem: MSE vs training sample size (on log scale) with unweighted data, weighted data, hybrid model, and hybrid data.

Table 4.1: Normalized MSE averaged over 50 trials on UCI data set of weighted, hybrid data, and hybrid model methods that use important weight derived from true selection probability. For each data set, the method that yields the lowest error is described in bold face.

Data set	P	P mix data	P mix model
India diabetes	0.982 ± 0.014	**0.967 ± 0.016**	0.975 ± 0.015
Ionosphere	0.937 ± 0.067	**0.880 ± 0.064**	0.937 ± 0.066
BreastCancer	1.008 ± 0.032	1.000 ± 0.033	1.000 ± 0.034
GermanCredit	1.000 ± 0.005	**0.995 ± 0.005**	0.999 ± 0.005
Australian credit	0.987 ± 0.016	0.974 ± 0.013	**0.968 ± 0.018**
Mushroom	0.981 ± 0.055	**0.949 ± 0.056**	0.967 ± 0.054
Congressional Voting	1.008 ± 0.043	**0.943 ± 0.047**	1.008 ± 0.047
Banknote	0.980 ± 0.060	**0.972 ± 0.070**	1.028 ± 0.063
Airfoil self noise	1.053 ± 0.026	**0.847 ± 0.017**	1.003 ± 0.025
Abalone	0.997 ± 0.014	**0.985 ± 0.013**	0.993 ± 0.014
Auto MGP	1.008 ± 0.044	**0.959 ± 0.041**	0.988 ± 0.044
Boston Housing	1.010 ± 0.021	**0.692 ± 0.023**	0.978 ± 0.021
Space ga	1.003 ± 0.012	**0.981 ± 0.008**	0.986 ± 0.011
Cadata	0.997 ± 0.020	**0.950 ± 0.019**	0.994 ± 0.019

Table 4.2: Normalized MSE averaged over 50 trials on UCI data set of weighted, hybrid data, and hybrid model methods that use important weight estimated by uLSIF. For each data set, the method that yields the lowest error is described in bold face.

Data set	uLSIF	uLSIF mix data	uLSIF mix model
India diabetes	1.026 ± 0.015	**0.918 ± 0.014**	1.004 ± 0.014
Ionosphere	0.986 ± 0.058	0.993 ± 0.067	**0.972 ± 0.061**
BreastCancer	0.998 ± 0.035	**0.988 ± 0.034**	0.990 ± 0.034
GermanCredit	1.003 ± 0.005	**0.999 ± 0.005**	1.000 ± 0.005
Australian credit	1.001 ± 0.010	**0.942 ± 0.011**	0.993 ± 0.011
Mushroom	0.987 ± 0.053	**0.885 ± 0.054**	0.985 ± 0.054
Congressional Voting	0.955 ± 0.036	**0.873 ± 0.054**	0.971 ± 0.040
Banknote	1.034 ± 0.055	**0.891 ± 0.057**	0.993 ± 0.058
Airfoil self noise	1.027 ± 0.023	**0.861 ± 0.017**	0.996 ± 0.024
Abalone	1.006 ± 0.014	**0.956 ± 0.013**	0.999 ± 0.014
Auto MGP	1.020 ± 0.040	**0.947 ± 0.039**	0.990 ± 0.040
Boston Housing	1.036 ± 0.022	**0.751 ± 0.030**	1.000 ± 0.022
Space ga	0.998 ± 0.010	**0.930 ± 0.008**	0.989 ± 0.010
Cadata	1.026 ± 0.019	1.031 ± 0.028	**1.006 ± 0.020**

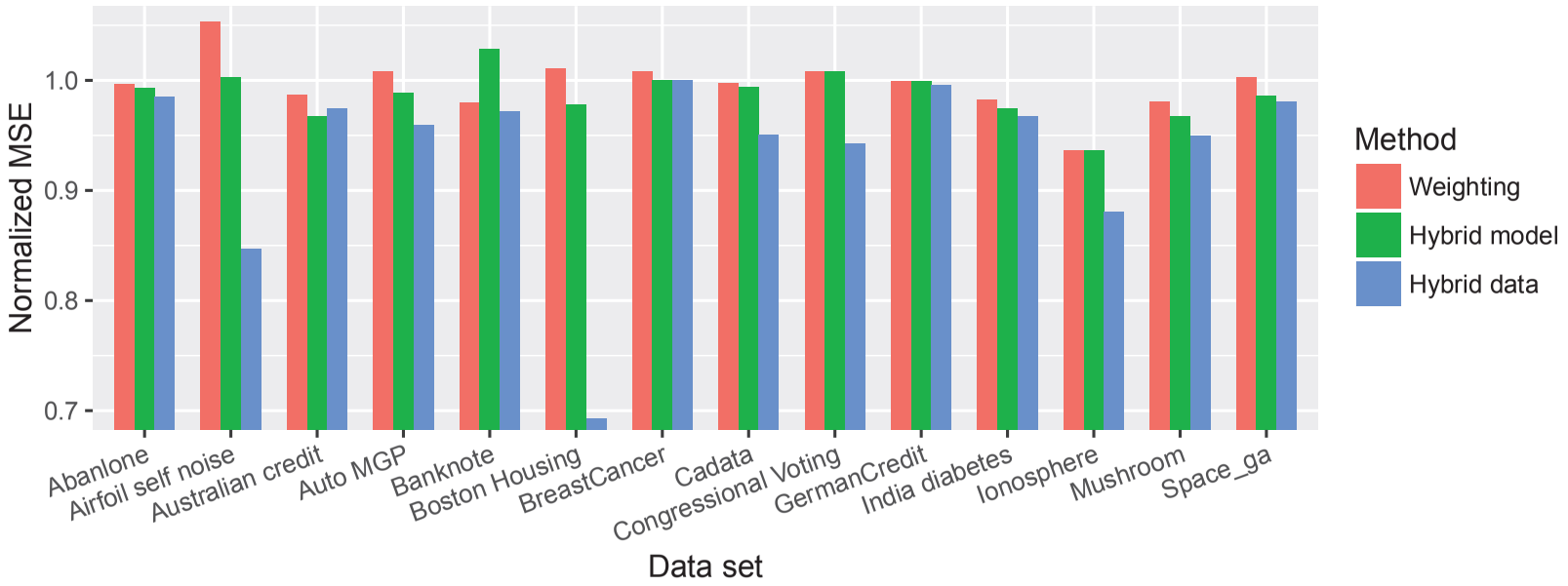


Figure 4.3: MSE gain of the weighted hybrid model (over the unweighted model) and the hybrid data method on each real-world data set when the importance weights are derived from true selection probability.

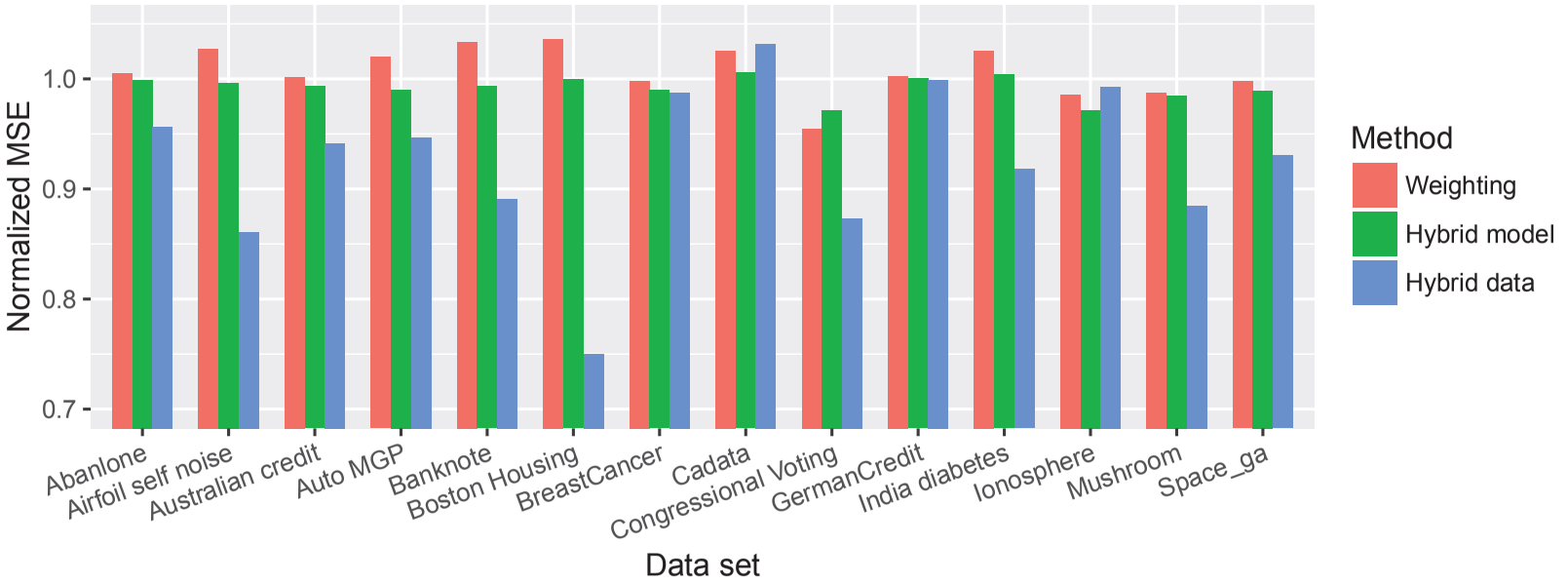


Figure 4.4: MSE gain of the weighted hybrid model (over the unweighted model) and the hybrid data method on each real-world data set when the importance weights are estimated using ulSIF.

Chapter 5

Conclusions

Selection bias is pervasive in almost all empirical studies, including Machine Learning. This thesis focus on the problems of supervised learning in the presence of selection bias. We have presented a general importance weighting framework to correct for selection bias with Bayesian Networks and two techniques to improve the importance weighting for the covariate shift. In this closing Chapter, we draw several conclusions from our work and suggest avenues for future research.

In the first part of this thesis, we discussed the importance weighting framework for generative and discriminative learning. We then present two methods of using the importance weight to correct for selection bias in discriminative learning: one with sampling and the other with modification of the loss function. Our results show that the importance weighting method that exploits the assumptions deemed plausible about the sampling mechanism achieves significant improvements in regression and classification accuracy over the unweighted method. Our analysis show that the importance weighted cross validation provides an almost unbiased estimate of the generalization error. In addition, we show that the IWCV can reliably decide when to use the weighted model to correct for selection bias and when to use the unweighted model and accept that the training sample size is not sufficient for the importance weighting.

There are several interesting future directions for selection bias correction with the importance weighting method. First, instead of requiring some assumptions about the sampling mechanism, one may expect to be able to infer them - at least partially - from several sources of data under some milder as-

sumptions. This approach shares some similar intuition with transfer learning. Second, it would be interesting to consider a formal sensitivity analysis to test the robustness of the importance weighting method against the uncertainty of the S-control feature vector X_s. The problem is that, given that we accept the existence of a S-control feature vector, X_s, the choice for the variables to be included in X_s may be subject to some uncertainty. With real-world data, it is almost impossible to make a firm statement regarding the appropriateness of X_s, or to promise to reduce the selection bias, or even to refrain from creating new bias where none exists. This problem is also well known in causal inference from observational data: all conclusions are extremely sensitive to which variables one chooses to hold constant (known as the "confounders") when we are assessing the causal effect of X on Y. For bias correction as for causal inference, such factors may be identified by simple graphical means when a (causal) graphical model is provided. Otherwise, no one is able to to tell us exactly which factors should be included in the analysis. This is why the so-called adjustment problem is so critical in the analysis of observational studies. We are facing the sample problem here. While the importance weighting scheme was shown to perform well despite our wrong assumptions about X_s in our simulations, it is fairly easy design a synthetic selection scenario such that the importance weighting relying on invalid assumptions performs worse than the baseline unweighted approach. Therefore, we believe there are circumstances under which even the best designed and run study is jeopardized by selection bias: improper handling of biased data can potentially distort the conclusions drawn from a study.

In the second part of this thesis, we presented a simple, yet effective, procedure that combines the weighted and unweighted prediction models in order to improve the standard importance weighting approach when misspecified training models are used. Our results showed that, while the unweighted model is globally more biased than the weighted one, it may locally be less biased on low importance instances. The hybrid model combining the weighted and the unweighted prediction models was shown to improve significantly the prediction performance with respect to the weighted or unweighted prediction models alone. Our method bears many resemblance to local learning techniques, which assign each training example a weight that depends on the location of

the training point in the input space relative to that of the point to be predicted (Bottou and Vapnik [1992]). Local learning is known to reduce the estimation bias at the expense of increasing model complexity. Therefore, it would be interesting to study the overall performance of local learning techniques under covariate shift with and without taking the importance weight into consideration.

In the last part of this thesis, we investigated the relationship between the covariate shift and the missing data problems and explored the possibility of using missing data imputation to improve the covariate shift correction. We established formally that, given a training set corrupted by covariate shift and an additional unbiased unlabeled data set, there exists a way to combine the weighted unlabeled data and the labeled training data such that the resulting data set follows the test distribution. In addition, the labels in this hybrid data set were proven to be missing at random (MAR), allowing the use of standard imputation methods. Our experiments on synthetic and real-world data demonstrated the efficiency of the approach with small sample sizes training data sets. The main caveat of the hybrid data approach is that its performance depends heavily on the imputation method being used. To our best knowledge, there are very few imputation methods (like PMM) that are robust to model misspecification, a property that is arguably crucial for the success of our hybrid data approach.

In term of future research directions, we think it should be useful to consider semi-supervised learning techniques for each specific problem with covariate shift. A good matching between the semi-supervised learning techniques and the data structure may greatly improve the prediction accuracy with small sample size data sets. Another idea is to exploit directly the unlabeled data to correct the covariate shift without estimating the importance weight as an intermediate step. Such a direct approach would render covariate shift correction independent on the importance weight estimation methods. Therefore, we hope this work will open up many avenues of future possible research topics on bias correction.

References

Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404, 1950. 46

Shumeet Baluja. Probabilistic modeling for face orientation discrimination: Learning from labeled and unlabeled data. In M. J. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems 11*, pages 854–860. MIT Press, 1999. 101

Elias Bareinboim and Judea Pearl. Controlling selection bias in causal inference. *Journal of Machine Learning Research - Proceedings Track*, 22: 100–108, 2012. 52

Elias Bareinboim, Jin Tian, and Judea Pearl. Recovering from selection bias in causal and statistical inference. In *AAAI*, pages 2410–2416, 2014. 52

Shai Ben-david, John Blitzer, Koby Crammer, and O Pereira. Analysis of representations for domain adaptation. In *In NIPS*. MIT Press, 2007. 24

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010. 24

Steffen Bickel, Michael Brückner, and Tobias Scheffer. Discriminative learning under covariate shift. *The Journal of Machine Learning Research*, 10:2137–2155, 2009. 24, 45

John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman. Learning bounds for domain adaptation. In *Advances in neural information processing systems*, pages 129–136, 2008. 24

Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred Warmuth. Classifying learnable geometric concepts with the vapnik-chervonenkis dimension. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pages 273–282. ACM, 1986. 17

Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Learnability and the vapnik-chervonenkis dimension. *Journal of the ACM (JACM)*, 36(4):929–965, 1989. 17

Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Occams razor. *Readings in machine learning*, pages 201–204, 1990. 17

Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992. 36

Léon Bottou and Vladimir Vapnik. Local learning algorithms. *Neural computation*, 4(6):888–900, 1992. 114

George EP Box and George C Tiao. *Bayesian inference in statistical analysis*, volume 40. John Wiley & Sons, 2011. 7

Leo Breiman and Philip Spector. Submodel selection and evaluation in regression. the x-random case. *International statistical review/revue internationale de Statistique*, pages 291–319, 1992. 21

P. Caillet, S. Klemm, M. Ducher, A. Aussem, and AM. Schott. Hip fracture in the elderly: a re-analysis of the epidos study with causal bayesian networks, to appear. In *Plos One*, 2015. 75

J. Quiñonero Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset Shift in Machine Learning*. MIT Press, 2009. (Editors). 23

Vittorio Castelli and Thomas M Cover. The relative value of labeled and unlabeled samples in pattern recognition with an unknown mixing parameter. *IEEE Transactions on information theory*, 42(6):2102–2117, 1996. 100, 101

Olivier Chapelle, Mingmin Chi, and Alexander Zien. A continuation method for semi-supervised svms. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 185–192, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2. doi: 10.1145/1143844.1143868. 101

G Cooper. Causal discovery from data in the presence of selection bias. In *Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics*, pages 140–150, 1995. 23

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995. 37

Corinna Cortes, Yishay Mansour, and Mehryar Mohri. Learning bounds for importance weighting. In J.D. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 442–450. Curran Associates, Inc., 2010. 24, 45, 88

Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 2016. 24

David Roxbee Cox and Nanny Wermuth. *Multivariate dependencies: Models, analysis and interpretation*, volume 67. CRC Press, 1996. 23

Hal Daumé III. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*, 2009. 24

S Rodrigues de Morais and A Aussem. An efficient learning algorithm for local bayesian network structure discovery. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECMLPKDD10*, pages 164–169, 2010. 49

Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic theory of pattern recognition*, volume 31. Springer Science & Business Media, 2013. 11

Vanessa Didelez, Svend Kreiner, and Niels Keiding. Graphical models for inference under outcome-dependent sampling. *Statistical Science*, pages 368–387, 2010. 23

Miroslav Dudík, Steven J Phillips, and Robert E Schapire. Correcting sample selection bias in maximum entropy density estimation. In *Advances in neural information processing systems*, pages 323–330, 2005. 25

Charles Elkan. The foundations of cost-sensitive learning. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'01, pages 973–978, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1-55860-812-5, 978-1-558-60812-2. 25

Theodoros Evgeniou, Massimiliano Pontil, and Tomaso Poggio. Regularization networks and support vector machines. *Advances in computational mathematics*, 13(1):1, 2000. 36

Wei Fan and Ian Davidson. On sample selection bias and its efficient correction via model averaging and unlabeled examples. In *SDM*. SIAM, 2007. 24

Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Subspace alignment for domain adaptation. *arXiv preprint arXiv:1409.5241*, 2014. 24

Yoav Freund and Robert E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995. 38

Sara Geneletti, Sylvia Richardson, and Nicky Best. Adjusting for selection bias in retrospective case-control studies. *Biostatistics (Oxford, England)*, 10(1): 17–31, January 2009. ISSN 1468-4357. doi: 10.1093/biostatistics/kxn010. 23, 50

Federico Girosi, Michael Jones, and Tomaso Poggio. Regularization theory and neural networks architectures. *Neural computation*, 7(2):219–269, 1995. 34

M Maria Glymour. Using causal diagrams to understand common problems in social epidemiology. *Methods in social epidemiology*, pages 393–428, 2006. 48

Peter J Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, pages 711–732, 1995. 8

Sander Greenland, Judea Pearl, and James M Robins. Causal diagrams for epidemiologic research. *Epidemiology*, pages 37–48, 1999. 48

Arthur Gretton, Alex Smola, Jiayuan Huang, Marcel Schmittfull, Karsten Borgwardt, and Bernhard Schölkopf. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4):5, 2009. 46, 88

Miguel A Hernán, Sonia Hernández-Díaz, Martha M Werler, and Allen A Mitchell. Causal knowledge as a prerequisite for confounding evaluation: an application to birth defects epidemiology. *American journal of epidemiology*, 155(2):176–184, 2002. 48

Miguel A Hernán, Sonia Hernández-Díaz, and James M Robins. A structural approach to selection bias. *Epidemiology*, 15(5):615–625, 2004. 51, 54

Thomas Hofmann, Bernhard Schölkopf, and Alexander J Smola. Kernel methods in machine learning. *The annals of statistics*, pages 1171–1220, 2008. 46

Ralph I Horwitz and Alvan R Feinstein. Alternative analytic methods for case-control studies of estrogens and endometrial cancer. *New England journal of medicine*, 299(20):1089–1094, 1978. 50

Jiayuan Huang, Alexander J. Smola, Arthur Gretton, Karsten M. Borgwardt, and Bernhard Schölkopf. Correcting sample selection bias by unlabeled data. In *NIPS*, pages 601–608. MIT Press, 2006. ISBN 0-262-19568-2. 25, 45, 56

T. Joachims. Making large-scale svm learning practical. LS8-Report 24, Universität Dortmund, LS VIII-Report, 1998. 37, 69

Thorsten Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, pages 200–209, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. ISBN 1-55860-612-2. 101

Takafumi Kanamori, Shohei Hido, and Masashi Sugiyama. A least-squares approach to direct importance estimation. *J. Mach. Learn. Res.*, 10:1391–1445, December 2009. ISSN 1532-4435. 25, 45, 47, 89

Takafumi Kanamori, Taiji Suzuki, and Masashi Sugiyama. Statistical analysis of kernel-based least-squares density-ratio estimation. *Machine Learning*, 86 (3):335–367, 2012. 25

Michael J Kearns and Umesh Virkumar Vazirani. *An introduction to computational learning theory*. MIT press, 1994. 11

Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145, 1995. 21

Kaname Kojima, Eric Perrier, Seiya Imoto, and Satoru Miyano. Optimal search on clustered structural constraint for learning bayesian network structure. *Journal of Machine Learning Research*, 11(Jan):285–310, 2010. 49

Roderick JA Little. Missing-data adjustments in large surveys. *Journal of Business & Economic Statistics*, 6(3):287–296, 1988. 101

David JC MacKay. *Bayesian methods for adaptive models*. PhD thesis, California Institute of Technology, 1992. 7

Shahar Mendelson. A few notes on statistical learning theory. In *Advanced lectures on machine learning*, pages 1–40. Springer, 2003. 12

Jose G Moreno-Torres, Troy Raeder, RocíO Alaiz-RodríGuez, Nitesh V Chawla, and Francisco Herrera. A unifying view on dataset shift in classification. *Pattern Recognition*, 45(1):521–530, 2012a. 24

Jose G Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V Chawla, and Francisco Herrera. A unifying view on dataset shift in classification. *Pattern Recognition*, 45(1):521–530, 2012b. 23

Tim P Morris, Ian R White, and Patrick Royston. Tuning multiple imputation by predictive mean matching and local residual draws. *BMC medical research methodology*, 14(1):1, 2014. 101

XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *Information Theory, IEEE Transactions on*, 56(11):5847–5861, 2010. 45

Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962. 45

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988. ISBN 0-934613-73-7. 49

Judea Pearl. On a class of bias-amplifying variables that endanger effect estimates. In *UAI 2010, Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence, Catalina Island, CA, USA, July 8-11, 2010*, pages 417–424, 2010. 67

Judea Pearl. A solution to a class of selection-bias problems. Technical report R-405, Department of Computer Science, University of California, December 2012. 50

Jose M Peña. Finding consensus bayesian network structures. *Journal of Artificial Intelligence Research*, 42:661–687, 2011. 49

Murray Rosenblatt et al. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27(3):832–837, 1956. 45

Bernhard Scholkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001. 36

Marco Scutari and Adriana Brogini. Bayesian network structure learning with permutation tests. *Communications in Statistics-Theory and Methods*, 41 (16-17):3233–3243, 2012. 49

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, October 2000. 25, 41, 60, 83, 90

Alex J Smola and Bernhard Schölkopf. *Learning with kernels*. Citeseer, 1998. 21

H. W. (Harold Wayne) Sorenson. *Parameter estimation : principles and problems*. New York : Marcel Dekker, 1980. ISBN 0824769872. 8

Ingo Steinwart. Support vector machines are universally consistent. *Journal of Complexity*, 18(3):768–791, 2002. 46

Mervyn Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 111–147, 1974. 21

Masashi Sugiyama and Motoaki Kawanabe. *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT Press, 2012. 25

Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert MÃžller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(May):985–1005, 2007a. 39

Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul von Bünau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *NIPS*, 2007b. 25, 45, 89

Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *AAAI*, volume 6, page 8, 2016. 24

Kai Ming Ting. A study on the effect of class distribution using cost-sensitive learning. In *Proceedings of the 5th International Conference on Discovery Science*, DS '02, pages 98–112, London, UK, UK, 2002. Springer-Verlag. ISBN 3-540-00188-3. 25

Van-Tinh Tran and Alex Aussem. A practical approach to reduce the learning bias under covariate shift. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part II*, pages 71–86, 2015a. 80

Van-Tinh Tran and Alex Aussem. Correcting a class of complete selection bias with external data based on importance weight estimation. In *International Conference on Neural Information Processing*, pages 111–118. Springer, 2015b. 23

Van-Tinh Tran and Alex Aussem. Reducing variance due to importance weightingin covariate shift bias correction. In *25th European Symposium on Artificial Neural Networks, ESANN 2017*, page to appear, 2017. 98

Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013. 12, 37

Vladimir N. Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998. 10, 12, 15

Edwin Villanueva and Carlos Dias Maciel. Optimized algorithm for learning bayesian network super-structures. In *ICPRAM (1)*, pages 217–222, 2012. 49

John Von Neumann. Various techniques used in connection with random digits. *National Bureau of Standards, Applied Mathematics Series*, 12:36–38, 1951. 31

Grace Wahba. *Spline models for observational data*, volume 59. Siam, 1990. 21

David H Wolpert. The lack of a priori distinctions between learning algorithms. *Neural computation*, 8(7):1341–1390, 1996. 11

Yao-liang Yu and Csaba Szepesvári. Analysis of kernel mean matching under covariate shift. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 607–614, 2012. 25

Bianca Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 114–, New York, NY, USA, 2004. ACM. ISBN 1-58113-838-5. doi: 10.1145/1015330.1015425. 24, 25, 31, 45, 83

Bianca Zadrozny, John Langford, and Naoki Abe. Cost-sensitive learning by cost-proportionate example weighting. In *Proceedings of the Third IEEE International Conference on Data Mining*, ICDM '03, pages 435–, Washington, DC, USA, 2003. IEEE Computer Society. ISBN 0-7695-1978-4. 31, 32

Xiaojin Zhu. Semi-supervised learning literature survey. Technical report, WisconsinMadison, 2005. 101