1 Covariate Shift by Kernel Mean Matching

Arthur Gretton[1]
Alex Smola[1]
Jiayuan Huang
Marcel Schmittfull
Karsten Borgwardt
Bernhard Schölkopf

Given sets of observations of training and test data, we consider the problem of re-weighting the training data such that its distribution more closely matches that of the test data. We achieve this goal by matching covariate distributions between training and test sets in a high dimensional feature space (specifically, a reproducing kernel Hilbert space). This approach does not require distribution estimation. Instead, the sample weights are obtained by a simple quadratic programming procedure. We provide a uniform convergence bound on the distance between the reweighted training feature mean and the test feature mean, a transductive bound on the expected loss of an algorithm trained on the reweighted data, and a connection to single class SVMs. While our method is designed to deal with the case of simple covariate shift (in the sense of Chapter **??**), we have also found benefits for sample selection bias on the labels. Our correction procedure yields its greatest and most consistent advantages when the learning algorithm returns a classifier/regressor that is "simpler" than the data might suggest.

1.1 Introduction

The default assumption in many learning scenarios is that training and test data are drawn independently and identically (iid) from the *same* distribution. When the distributions on training and test set do not match, we face the problem of *dataset shift*: given a domain of patterns $\mathcal{X}$ and labels $\mathcal{Y}$, we obtain training samples $Z_{\mathrm{tr}} = \left\{(x_1^{\mathrm{tr}}, y_1^{\mathrm{tr}}), \ldots, (x_{n_{\mathrm{tr}}}^{\mathrm{tr}}, y_{n_{\mathrm{tr}}}^{\mathrm{tr}})\right\} \subseteq \mathcal{X} \times \mathcal{Y}$ from a Borel probability distribution $\mathrm{P}_{\mathrm{tr}}(x, y)$, and test samples $Z_{\mathrm{te}} = \left\{(x_1^{\mathrm{te}}, y_1^{\mathrm{te}}), \ldots, (x_{n_{\mathrm{te}}}^{\mathrm{te}}, y_{n_{\mathrm{te}}}^{\mathrm{te}})\right\} \subseteq \mathcal{X} \times \mathcal{Y}$ drawn from another such distribution $\mathrm{P}_{\mathrm{te}}(x, y)$.

1. Both authors contributed equally.

Although there exists previous work addressing this problem [Zadrozny, 2004, Rosset et al., 2004, Heckman, 1979, Lin et al., 2002, Dudík et al., 2005, Shimodaira, 2000, Sugiyama and Müller, 2005], dataset shift has typically been ignored in standard estimation algorithms. Nonetheless, in reality the problem occurs rather frequently. Below, we give some examples of where dataset shift occurs (following the terminology defined by Storkey in Chapter **??**).

1. Suppose we wish to generate a model to diagnose breast cancer. Suppose, moreover, that most women who participate in the breast screening test are middle-aged and likely to have attended the screening in the preceding 3 years. Consequently our sample includes mostly older women and those who have low risk of breast cancer because they have been tested before. This problem is referred to as *sample selection bias*. The examples do not reflect the general population with respect to age (which amounts to a bias in $\mathrm{P}_{\mathrm{tr}}(x)$) and they only contain very few diseased cases (i.e. a bias in $\mathrm{P}_{\mathrm{tr}}(y|x)$).

2. Consider the problem of data analysis using a brain computer interface, where the distribution over incoming signals is known to change as experiments go on (subjects tire, the sensor setup changes, etc). In this case it necessary to adapt the estimator to the new distribution of patterns in order to improve performance [Sugiyama et al., 2007].

3. Gene expression profile studies using DNA microarrays are used in tumor diagnosis. A common problem is that the samples are obtained using certain protocols, microarray platforms and analysis techniques, and typically have small sample sizes. The test cases are recorded under different conditions, resulting in a different distribution of gene expression values. In both this and the previous example, a *covariate shift* has occurred (see Chapter **??**).

In all cases we would intuitively want to assign more weight those observations in the training set which are most similar to those in the test set, and less weight to those which rarely occur in the test set.

In this chapter, we use unlabeled data as the basis for a dataset shift correction procedure for various learning methods. Unlike previous work, we infer the resampling weight *directly* by non-parametric distribution matching between training and testing samples. We do not need to estimate the biasing densities or selection probabilities [Zadrozny, 2004, Dudík et al., 2005, Shimodaira, 2000], or to assume advance knowledge of the different class probabilities [Lin et al., 2002]. Rather, we account for the difference between $\mathrm{P}_{\mathrm{tr}}(x, y)$ and $\mathrm{P}_{\mathrm{te}}(x, y)$ by reweighting the training points such that the means of the training and test points in a reproducing kernel Hilbert space (RKHS) are close. We call this reweighting process kernel mean matching (KMM), following our presentation in [Huang et al., 2007]. The present chapter expands on our earlier work in terms of both theoretical and experimental analysis.

Since our approach does not require density estimation, we are able to state results which apply to arbitrary domains and which do not, in principle, suffer from the curse of dimensionality that befalls high-dimensional density estimation. When

the RKHS is universal [Steinwart, 2002], the population solution to this minimisation is exactly the ratio $\mathrm{P}_{\mathrm{te}}(x,y)/\mathrm{P}_{\mathrm{tr}}(x,y)$; we derive performance guarantees which depend on the maximum ratio between the distributions (but not on the distributions themselves) and which show that our method is consistent. We remark that when this ratio is large, however, a large sample size would be required to ensure the bound is tight (and to guarantee a good approximation).

The required optimisation is a simple quadratic program, and the reweighted sample can be incorporated straightforwardly into many regression and classification algorithms and model selection procedures, such as cross validation. We apply our method to a variety of regression and classification benchmarks from UCI and elsewhere, as well as to classification of microarrays from prostate and breast cancer patients. The experiments demonstrate that sample reweighting by KMM substantially improves learning performance in cases where the class of functions output by the learning algorithm is "simpler" than the true function (for instance, such a classification algorithm would estimate a decision boundary deliberately smoother than the Bayes optimal boundary that emerges as the sample size increases to infinity). Indeed, for this case, performance can be improved from close to chance level to the point of almost matching the performance of a learning algorithm with the "correct" complexity. KMM reweighting can also improve performance in cases where the complexity of the leaned classification/regression function is chosen optimally for the data, via parameter selection by cross validation. For most such cases, however, KMM does not affect performance, or can even make it slightly worse.

In general, the estimation problem with two different distributions $\mathrm{P}_{\mathrm{tr}}(x,y)$ and $\mathrm{P}_{\mathrm{te}}(x,y)$ is unsolvable, as the two distributions could be arbitrarily far apart. In particular, for arbitrary $\mathrm{P}_{\mathrm{tr}}(y|x)$ and $\mathrm{P}_{\mathrm{te}}(y|x)$, there is no way we could infer a good estimator based on the training sample. For instance, the distributions $\mathrm{P}_{\mathrm{tr}}(y=1|x)$ and $\mathrm{P}_{\mathrm{te}}(y=-1|x)$ could be swapped in binary classification, leading to arbitrarily large error. The following assumption allows us to address the problem.

Assumption 1.1 *We make the simplifying assumption that* $\mathrm{P}_{\mathrm{tr}}(x,y)$ *and* $\mathrm{P}_{\mathrm{te}}(x,y)$ *only differ via* $\mathrm{P}_{\mathrm{tr}}(x,y) = \mathrm{P}(y|x)\mathrm{P}_{\mathrm{tr}}(x)$ *and* $\mathrm{P}_{\mathrm{te}}(x,y) = \mathrm{P}(y|x)\mathrm{P}_{\mathrm{te}}(x)$. *In other words, the conditional probabilities of* $y|x$ *remain* unchanged.

This particular case of dataset shift has been termed *covariate shift* (see examples above, Chapter **??** and [Shimodaira, 2000]). We will see experimentally that even in situations where our key assumption is not valid, our method can still be useful (see Section 1.6).

We begin our presentation in Section 1.2, where we describe the concept of sample reweighting to match empirical distributions, and show how a reweighted sample can be incorporated easily into a variety of learning algorithms (penalised 1-norm classification, penalised logistic regression, penalised LMS regression, Poisson regression). In Section 1.3, we describe our sample reweighting procedure, which entails matching the means of the reweighted training sample and the target (test) sample in a reproducing kernel Hilbert space. We discuss the convergence of the Hilbert space training and test means in the limit of large sample size, and provide

an empirical optimization procedure to select the training sample weights (this being a straightforward quadratic program). In Section 1.4, we provide transductive guarantees on the performance of learning algorithms that use the reweighted sample, subject to linearity conditions on the loss functions of these algorithms. We establish a connection between sample bias correction and novelty detection in Section 1.5, with reference to the single class SVM. We present our experiments in Section 1.6: these comprise a toy example proposed by Shimodaira [2000], a detailed analysis of performance for different classifier parameters on the UCI breast cancer data set, a broader overview of performance on many different UCI datasets, and experiments on microarray data. We provide proofs of our theoretical results in an appendix.

1.2 Sample Reweighting

We begin by stating the problem of risk minimization. In general a learning method aims to minimize the expected risk

$$R[\mathrm{P}, \theta, l(x, y, \theta)] = \mathbf{E}_{(x,y)\sim \mathrm{P}}\left[l(x, y, \theta)\right] \tag{1.1}$$

of a loss function $l(x, y, \theta)$ depending on a parameter θ. For instance, the loss function could be the negative log-likelihood $-\log \mathrm{P}(y|x, \theta)$, a misclassification loss, or some form of regression loss. However, since typically we only observe examples (x, y) drawn from $\mathrm{P}(x, y)$ rather than $\mathrm{P}(x, y)$, we resort to computing the empirical average

$$R_{\mathrm{emp}}[Z, \theta, l(x, y, \theta)] = \frac{1}{n}\sum_{i=1}^{n} l(x_i, y_i, \theta). \tag{1.2}$$

To avoid overfitting, instead of minimizing R_{emp} directly we minimize a regularized variant,

$$R_{\mathrm{reg}}[Z, \theta, l(x, y, \theta)] := R_{\mathrm{emp}}[Z, \theta, l(x, y, \theta)] + \lambda\Omega[\theta]$$

where $\Omega[\theta]$ is a regularizer.

1.2.1 Sample Correction

The problem is more involved if $\mathrm{P}_{\mathrm{tr}}(x, y)$ and $\mathrm{P}_{\mathrm{te}}(x, y)$ are different. The training set is drawn from P_{tr}, however what we would really like is to minimize $R[\mathrm{P}_{\mathrm{te}}, \theta, l]$ as we wish to generalize to test examples drawn from P_{te}. An observation from the

field of importance sampling is that

$$R[\mathrm{P}_{\mathrm{te}}, \theta, l(x,y,\theta)] = \mathbf{E}_{(x,y)\sim \mathrm{P}_{\mathrm{te}}}\left[l(x,y,\theta)\right] = \mathbf{E}_{(x,y)\sim \mathrm{P}_{\mathrm{tr}}}\Big[\underbrace{\frac{\mathrm{P}_{\mathrm{te}}(x,y)}{\mathrm{P}_{\mathrm{tr}}(x,y)}}_{:=\beta(x,y)} l(x,y,\theta)\Big]$$

$$= R[\mathrm{P}_{\mathrm{tr}}, \theta, \beta(x,y) l(x,y,\theta)],$$

provided that the support of P_{te} *is contained in the support of* P_{tr}. If this does not hold, reweighting x in order to obtain a risk estimate for $\mathrm{P}_{\mathrm{te}}(x,y)$ is impossible. In fact, the risks could be arbitrarily different, since we have no information about the behavior of $l(x,y,\theta)$ on a subset of the domain of P_{te}.

Given $\beta(x,y)$, we can thus compute the risk with respect to P_{te} using P_{tr}. Similarly, we can *estimate* the risk with respect to P_{te} by computing $R_{\mathrm{emp}}[Z, \theta, \beta(x,y) l(x,y,\theta)]$. The key problem is that the coefficients $\beta(x,y)$ are usually unknown, and must be estimated from the data. When P_{tr} and P_{te} differ in $\mathrm{P}_{\mathrm{tr}}(x)$ and $\mathrm{P}_{\mathrm{te}}(x)$ only, we have $\beta(x,y) = \mathrm{P}_{\mathrm{te}}(x)/\mathrm{P}_{\mathrm{tr}}(x)$, where β is a reweighting factor for the training examples. We thus reweight every training observation $(x_i^{\mathrm{tr}}, y_i^{\mathrm{tr}})$ such that observations that are under-represented in P_{tr} (relative to P_{te}) are assigned a higher weight, whereas over-represented cases are downweighted.

We could estimate P_{tr} and P_{te} and subsequently compute β based on those estimates. This is closely related to the methods of Zadrozny [2004], Lin et al. [2002], Sugiyama and Müller [2005], who either have to estimate the selection probabilities, or have prior knowledge of the class distributions. While intuitive, this approach has three major drawbacks:

1. It only works whenever the estimates for P_{tr} and P_{te} (or potentially, the selection probabilities or class distributions) are good. In particular, small errors in estimating P_{tr} can lead to large coefficients β and consequently to a serious overweighting of the corresponding observations.

2. Estimating both distributions just for the purpose of computing reweighting coefficients may be overkill: we may be able to directly estimate the coefficients $\beta_i := \beta(x_i^{\mathrm{tr}}, y_i^{\mathrm{tr}})$ without having to perform distribution estimation. Furthermore, we can regularize β_i directly with more flexibility, taking prior knowledge into account (similar to learning methods for other problems).

3. It is well known that using the exact importance-sampler weights may not be optimal, even when knowing both distributions. See e.g. [Shimodaira, 2000] for a discussion of the issue. The basic idea is that importance sampler weights β which deviate strongly from 1 increase the variance significantly. In fact, as we will see in Lemma 1.8, the effective training sample size is $n_{\mathrm{tr}}^2 / \|\beta\|_2^2$. Hence it may be worth accepting a small bias in return for a larger effective sample size.

1.2.2 Using the sample reweighting in learning algorithms

Before we describe how we will estimate the reweighting coefficients β_i, we briefly discuss how to minimize the reweighted regularized risk

$$R_{\mathrm{reg}}[Z, \beta, l(x, y, \theta)] := \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \beta_i l(x_i^{\mathrm{tr}}, y_i^{\mathrm{tr}}, \theta) + \lambda \Omega[\theta], \tag{1.3}$$

in four useful settings.

Penalized 1-norm Classification (Support Vector Classification): Using the formulation of Tsochantaridis et al. [2005], Taskar et al. [2004] we have the following minimization problem (the original SVMs can be formulated in the same way):

$$\underset{\theta,\xi}{\text{minimize}} \ \frac{1}{2} \|\theta\|^2 + C \sum_{i=1}^{n_{\mathrm{tr}}} \beta_i \xi_i \tag{1.4a}$$

$$\text{subject to } \left\langle \Phi(x_i^{\mathrm{tr}}, y_i^{\mathrm{tr}}) - \Phi(x_i^{\mathrm{tr}}, y), \theta \right\rangle \geq 1 - \xi_i / \Delta(y_i^{\mathrm{tr}}, y) \tag{1.4b}$$

$$\text{for all } \ y \in \mathcal{Y}, \text{ and } \xi_i \geq 0.$$

Here, $\Phi(x, y)$ is a feature map from $\mathcal{X} \times \mathcal{Y}$ to a feature space $\mathcal{F}$, where $\theta \in \mathcal{F}$ and $\Delta(y, y')$ denotes a discrepancy function between y and y'. The dual of (1.4) is

$$\underset{\alpha}{\text{minimize}} \ \frac{1}{2} \sum_{i,j=1;y,y' \in \mathcal{Y}}^{n_{\mathrm{tr}}} \alpha_{iy} \alpha_{jy'} k(x_i^{\mathrm{tr}}, y, x_j^{\mathrm{tr}}, y') - \sum_{i=1;y \in \mathcal{Y}}^{n_{\mathrm{tr}}} \alpha_{iy} \tag{1.5a}$$

$$\text{subject to } \alpha_{iy} \geq 0 \text{ for all } i, y \text{ and } \sum_{y \in \mathcal{Y}} \alpha_{iy} / \Delta(y_i^{\mathrm{tr}}, y) \leq \beta_i C. \tag{1.5b}$$

Here $k(x, y, x', y') := \langle \Phi(x, y), \Phi(x', y') \rangle$ denotes the inner product between the feature maps. This generalizes the observation-dependent binary SV classification described by Schmidt and Gish [1996]. Many existing solvers, such as SVMStruct [Tsochantaridis et al., 2005], can be modified easily to take sample-dependent weights into account.

Penalized Logistic Regression: This is also referred to as *Gaussian Process Classification*. In the unweighted case [Williams and Barber, 1998], we minimize $\sum_{i=1}^{n} -\log p(y_i | x_i, \theta) + \frac{\lambda}{2} \|\theta\|^2$ with respect to θ. Using (1.3) yields the following modified optimization problem:

$$\underset{\theta}{\text{minimize}} \sum_{i=1}^{n_{\mathrm{tr}}} -\beta_i \log p(y_i^{\mathrm{tr}} | x_i^{\mathrm{tr}}, \theta) + \frac{\lambda}{2} \|\theta\|^2 . \tag{1.6}$$

Using an exponential families and kernel approach for

$$\log p(y|x, \theta) = \ \langle \Phi(x, y), \theta \rangle - g(\theta|x) \tag{1.7}$$

$$\text{where } \ g(\theta|x) = \log \sum_{y \in \mathcal{Y}} \exp\left(\langle \Phi(x, y), \theta \rangle \right)$$

we can invoke the representer theorem [Kimeldorf and Wahba, 1970] which leads to

$$\begin{aligned}\underset{\alpha}{\text{minimize}} \sum_{i=1}^{n_{\text{tr}}} \beta_i g(\alpha|x_i^{\text{tr}}) - \sum_{i,j=1;y\in\mathcal{Y}}^{n_{\text{tr}}} \alpha_{iy}\beta_j k(x_i^{\text{tr}}, y, x_j^{\text{tr}}, y_j^{\text{tr}}) \\ + \sum_{i,j=1;y,y'\in\mathcal{Y}}^{n_{\text{tr}}} \alpha_{iy}\alpha_{jy'} k(x_i^{\text{tr}}, y, x_j^{\text{tr}}, y') \qquad (1.8)\\ \text{where } g(\alpha|x_i^{\text{tr}}) := \log \sum_{y\in\mathcal{Y}} \exp\left(\sum_{j=1;y'\in\mathcal{Y}}^{n_{\text{tr}}} \alpha_{jy'} k(x_i^{\text{tr}}, y, x_j^{\text{tr}}, y')\right).\end{aligned}$$

Penalized LMS Regression: Assume $l(x, y, \theta) = (y - \langle\Phi(x), \theta\rangle)^2$ and $\Omega[\theta] = \|\theta\|^2$. Here we solve

$$\underset{\theta}{\text{minimize}} \sum_{i=1}^{n_{\text{tr}}} \beta_i (y_i^{\text{tr}} - \langle\Phi(x_i^{\text{tr}}), \theta\rangle)^2 + \lambda \|\theta\|^2 . \qquad (1.9)$$

Denote by $\bar{\beta}$ the diagonal matrix with diagonal $(\beta_1, \ldots, \beta_{n_{\text{tr}}})$ and by $K \in \mathbb{R}^{m\times m}$ the kernel matrix $K_{ij} = k(x_i^{\text{tr}}, x_j^{\text{tr}})$. In this case minimizing (1.9) is equivalent to solving

$$\underset{\alpha}{\text{minimize}} \quad (y - K\alpha)^\top \bar{\beta}(y - K\alpha) + \lambda\alpha^\top K\alpha$$

with respect to α. Assuming that K and $\bar{\beta}$ have full rank, the minimization yields

$$\alpha = (\lambda\bar{\beta}^{-1} + K)^{-1} y$$

The advantage of this formulation is that it can be solved as easily as the standard penalized regression problem. Essentially, we rescale the regularizer depending on the pattern weights: the higher the weight of an observation, the less we regularize.

Poisson Regression: Assume a process of discrete events, such as the distribution of species over a geographical location or the occurrence of non-infectious diseases. This process can be modeled by a conditional Poisson distribution,

$$\log p(y|x, \theta) = y \langle\Phi(x), \theta\rangle - \log y! - \exp(\langle\Phi(x), \theta\rangle) \qquad (1.10)$$

as a member of the nonparametric exponential family (see e.g. [Cressie, 1993]), where $y \in \mathbb{N}_0$. Consequently we may obtain a re-weighted risk minimization problem,

$$\underset{\alpha}{\text{minimize}} \sum_{i=1}^{n_{\text{tr}}} \beta_i \exp([K\alpha]_i) - \beta_i y_i^{\text{tr}} [K\alpha]_i + \lambda\alpha^\top K\alpha. \qquad (1.11)$$

Here K and α are defined as in the above example. The problem is convex in α.

We provided the above examples to demonstrate that it is fairly straightforward to turn most risk minimization procedures into re-weighted ones. For those algorithms

which cannot deal with weighted data easily, one may always resort to re-sampling, see e.g. [Efron and Tibshirani, 1994].

1.3 Distribution Matching

1.3.1 Kernel Mean Matching and its relation to importance sampling

Let $\Phi : \mathcal{X} \to \mathcal{F}$ be a feature map into a feature space $\mathcal{F}$ and denote by $\mu : \mathcal{P} \to \mathcal{F}$ the expectation operator

$$\mu(\mathrm{P}) := \mathbf{E}_{x\sim\mathrm{P}(x)}\left[\Phi(x)\right]. \tag{1.12}$$

Clearly μ is a *linear* operator mapping the space of all probability distributions $\mathcal{P}$ into feature space. Denote by $\mathcal{M}(\Phi) := \{\mu(\mathrm{P}) \text{ where } \mathrm{P} \in \mathcal{P}\}$ the image of $\mathcal{P}$ under μ. This set is also often referred to as the *marginal polytope*. We have the following theorem, proved in the appendix.

Theorem 1.2 *The operator μ 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 a universal kernel $k(x, x') = \langle\Phi(x), \Phi(x')\rangle$ in the sense of Steinwart [2002] (bearing in mind that universality is defined for kernels on compact domains $\mathcal{X}$).*

The practical consequence of this (rather abstract) result is that if we know $\mu(\mathrm{P}_{\mathrm{te}})$, we can infer a suitable weighting function β by solving the following minimization problem. We first state the expectation version of the kernel mean matching (KMM) procedure:

Lemma 1.3 *The following optimization problem in β is convex.*

$$\underset{\beta}{\text{minimize}} \quad \left\|\mu(\mathrm{P}_{\mathrm{te}}) - \mathbf{E}_{x\sim\mathrm{P}_{\mathrm{tr}}(x)}\left[\beta(x)\Phi(x)\right]\right\| \tag{1.13}$$

$$\textit{subject to } \beta(x) \geq 0 \textit{ and } \mathbf{E}_{x\sim\mathrm{P}_{\mathrm{tr}}(x)}\left[\beta(x)\right] = 1. \tag{1.14}$$

Assume P_{te} is absolutely continuous with respect to P_{tr} (so $\mathrm{P}_{\mathrm{tr}}(A) = 0$ implies $\mathrm{P}_{\mathrm{te}}(A) = 0$), and that k is universal. The solution of (1.13) is then $\mathrm{P}_{\mathrm{te}}(x) = \beta(x)\mathrm{P}_{\mathrm{tr}}(x)$.

Proof: The convexity of the objective function follows from the facts that the norm is a convex function and the integral is a linear functional in β. The other constraints are also convex.

By virtue of the constraints, any feasible solution of β corresponds to a distribution, as $\int \beta(x)d\mathrm{P}_{\mathrm{tr}}(x) = 1$. Moreover, the choice of $\hat{\beta}(x) := \mathrm{P}_{\mathrm{te}}(x)/\mathrm{P}_{\mathrm{tr}}(x)$ is feasible as it obviously satisfies the constraints. Moreover, it minimizes the objective function with value 0. Note that such a $\beta(x)$ exists due to the absolute continuity of

$\mathrm{P}_{\mathrm{te}}(x)$ with respect to $\mathrm{P}_{\mathrm{tr}}(x)$. Theorem 1.2 implies that there can be only one distribution $\beta(x)\mathrm{P}_{\mathrm{tr}}$ such that $\mu(\beta(x)\mathrm{P}_{\mathrm{tr}}) = \mu(\mathrm{P}_{\mathrm{te}})$. Hence $\beta(x)\mathrm{P}_{\mathrm{tr}}(x) = \mathrm{P}_{\mathrm{te}}(x)$.

1.3.2 Convergence of reweighted means in feature space

Lemma 1.3 shows that in principle, if we knew P_{tr} and $\mu[\mathrm{P}_{\mathrm{te}}]$, we could fully recover P_{te} by solving a simple quadratic program. In practice, however, neither $\mu(\mathrm{P}_{\mathrm{te}})$ nor P_{tr} is known. Instead, we only have samples X_{tr} and X_{te} of size n_{tr} and n_{te}, drawn iid from P_{tr} and P_{te}, respectively.

Naively we could just replace the expectations in (1.13) by empirical averages and hope that the resulting optimization problem will provide us with a good estimate of β. However, it is to be expected that empirical averages will differ from each other due to finite sample size effects. In this section, we explore two such effects. First, we demonstrate that in the finite sample case, for a fixed β, the empirical estimate of the expectation of β is normally distributed: this provides a natural limit on the precision with which we should enforce the constraint $\int \beta(x) d\mathrm{P}_{\mathrm{tr}}(x) = 1$ when using empirical expectations (we will return to this point in the next section).

Lemma 1.4 *If $\beta(x) \in [0, B]$ is some fixed function of $x \in \mathcal{X}$, then given $x_i^{\mathrm{tr}} \sim \mathrm{P}_{\mathrm{tr}}$ iid such that $\beta(x_i^{\mathrm{tr}})$ has finite mean and finite non-zero variance, the sample mean $\frac{1}{n_{\mathrm{tr}}} \sum_i \beta(x_i^{\mathrm{tr}})$ converges in distribution to a Gaussian with mean $\int \beta(x) d\mathrm{P}_{\mathrm{tr}}(x)$ and standard deviation bounded by $\frac{B}{2\sqrt{n_{\mathrm{tr}}}}$.*

This lemma is a direct consequence of the central limit theorem [Casella and Berger, 2002, Theorem 5.5.15]. Alternatively, it is straightforward to get a large deviation bound that likewise converges as $1/\sqrt{n_{\mathrm{tr}}}$ Hoeffding [1963]. In this case, it follows that with probability at least $1 - \delta$

$$\left| \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \beta(x_i^{\mathrm{tr}}) - 1 \right| \leq B \sqrt{\log(2/\delta)/2m}. \tag{1.15}$$

Our second result demonstrates the deviation between the empirical means of P_{te} and $\beta(x)\mathrm{P}_{\mathrm{tr}}$ in feature space, given $\beta(x)$ is chosen perfectly in the population sense.

Lemma 1.5 *In addition to the conditions of Lemma 1.4, assume that we draw $X_{\mathrm{te}} := \{x_1^{\mathrm{te}}, \ldots, x_{n_{\mathrm{te}}}^{\mathrm{te}}\}$ iid from $\mathcal{X}$ using $\mathrm{P}_{\mathrm{te}} = \beta(x)\mathrm{P}_{\mathrm{tr}}$, and $\|\Phi(x)\| \leq R$ for all $x \in \mathcal{X}$. Then with probability at least $1 - \delta$,*

$$\begin{aligned} &\left\| \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \beta(x_i^{\mathrm{tr}}) \Phi(x_i^{\mathrm{tr}}) - \frac{1}{n_{\mathrm{te}}} \sum_{i=1}^{n_{\mathrm{te}}} \Phi(x_i^{\mathrm{te}}) \right\| \\ &\qquad \leq \left(1 + \sqrt{2 \log 2/\delta}\right) R \sqrt{B^2/n_{\mathrm{tr}} + 1/n_{\mathrm{te}}}. \end{aligned} \tag{1.16}$$

The proof is in the appendix. Note that this lemma shows that for a *given* $\beta(x)$, which is correct in the population sense, we can bound the deviation between the mean and the importance-sampled mean in feature space. It is *not* a guarantee that

we will find coefficients β_i which are close to $\beta(x_i^{\mathrm{tr}})$, when solving the optimization problem in the next section.

Lemma 1.5 implies we have $O(B\sqrt{1/n_{\mathrm{tr}} + 1/n_{\mathrm{te}}B^2})$ convergence in $n_{\mathrm{tr}}, n_{\mathrm{te}}$ and B. This means that for very different distributions, we need a large equivalent sample size to get reasonable convergence. Our result also implies that it is unrealistic to assume that the empirical means (reweighted or not) should match exactly. Note that a somewhat better bound could be obtained by exploiting the interplay between $\mathrm{P}_{\mathrm{tr}}, \mathrm{P}_{\mathrm{te}}$ and $\Phi(x)$. That is, it is essentially $\|\Phi(x)\| \mathrm{P}_{\mathrm{te}}(x)/\mathrm{P}_{\mathrm{tr}}(x)$ that matters, as one can see by a simple modification of the proof. For this reason, we may be able to tolerate large deviations between the two distributions at little cost, as long as the feature vector at this location is small.

1.3.3 Empirical KMM optimization

To find suitable values of $\beta \in \mathbb{R}^{n_{\mathrm{tr}}}$ we want to minimize the discrepancy between means subject to constraints $\beta_i \in [0, B]$ and $|\frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \beta_i - 1| \leq \epsilon$. The former limits the scope of discrepancy between P_{tr} and P_{te} and ensures robustness by limiting the influence of individual observations, whereas the latter ensures that the corresponding measure $\beta(x)\mathrm{P}_{\mathrm{tr}}(x)$ is close to a probability distribution. Note that for $B \to 1$ we obtain the unweighted solution. The objective function is given by the discrepancy term between the two empirical means. Using $K_{ij} := k(x_i^{\mathrm{tr}}, x_j^{\mathrm{tr}})$ and $\kappa_i := \frac{n_{\mathrm{tr}}}{n_{\mathrm{te}}} \sum_{j=1}^{n_{\mathrm{te}}} k(x_i^{\mathrm{tr}}, x_j^{\mathrm{te}})$ one may check that

$$\left\| \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \beta_i \Phi(x_i^{\mathrm{tr}}) - \frac{1}{n_{\mathrm{te}}} \sum_{i=1}^{n_{\mathrm{te}}} \Phi(x_i^{\mathrm{te}}) \right\|^2 = \frac{1}{n_{\mathrm{tr}}^2} \beta^\top K \beta - \frac{2}{n_{\mathrm{tr}}^2} \kappa^\top \beta + \mathrm{const.}$$

Now we have all necessary ingredients to formulate a quadratic problem to find suitable β via

$$\underset{\beta}{\text{minimize}} \ \frac{1}{2} \beta^\top K \beta - \kappa^\top \beta \text{ subject to } \beta_i \in [0, B] \text{ and } \left| \sum_{i=1}^{n_{\mathrm{tr}}} \beta_i - n_{\mathrm{tr}} \right| \leq n_{\mathrm{tr}} \epsilon. \tag{1.17}$$

In accordance with Lemma 1.4, we conclude that a good choice of ϵ should be $O(B/\sqrt{n_{\mathrm{tr}}})$. That said, even a change induced by normalizing $\sum_i \beta_i = 1$ only changes the value of the objective function by at most $\epsilon^2 R^2 + 2\epsilon L$, where L^2 is the value of the objective function at optimality.

Note that (1.17) is a quadratic program which can be solved efficiently using interior point methods or any other successive optimization procedure, such as chunking [Osuna, 1998], SMO [Platt, 1999], or projected gradient methods [Dai and Fletcher, 2006]. We also point out that (1.17) resembles Single Class SVM [Schölkopf et al., 2001] using the ν-trick. Besides the approximate equality constraint, the main difference is the linear correction term by means of κ. Large values of κ_i correspond to particularly important observations x_i^{tr} and are likely to lead to large β_i. We discuss further connections in Section 1.5.

1.4 Risk Estimates

So far we have been concerned concerned only with distribution matching for the purpose of finding a reweighting scheme between the empirical feature space means on training X_{tr} and test X_{te} sets. We now show, in the case of *linear* loss functions, that as long as the feature means on the test set are well enough approximated, we will be able to obtain *almost unbiased* risk estimates *regardless* of the actual values of β_i vs. their importance sampling weights $\beta(x_i)$. The price is an increase in the variance of the estimate, where $n_{\text{tr}}^2 / \|\beta\|^2$ will act as an effective sample size.

1.4.1 Transductive Bounds

We consider the transductive case: that is, we will make uniform convergence statements with respect to $\mathbf{E}_{y|x}$ only (recall that this expectation is the same for the training and test distributions by assumption). In addition, we will require the loss functions to be linear, as described below.

Assumption 1.6 *We require that $l(x, \theta)$ be expressible as inner product in feature space, i.e. $l(x, \theta) = \langle \Psi(x), \Theta \rangle$, where $\|\Theta\| \leq C$. That is, $l(x, \theta)$ belongs to a Reproducing Kernel Hilbert Space (RKHS). Likewise, assume $l(x, y, \theta)$ can be expressed as an element of an RKHS via $\langle \Upsilon(x, y), \Lambda \rangle$ with[2] $\|\Lambda\| \leq C$ and $\|\Upsilon(x, y)\| \leq R$.*

We proceed in two steps: first we show that for the expected loss

$$l(x, \Theta) := \mathbf{E}_{y|x} l(x, y, \Lambda), \tag{1.18}$$

the coefficients β_i can be used to obtain a risk estimate with low bias. Second, we show that the random variable $\sum_i \beta_i l(x_i^{\text{tr}}, y_i^{\text{tr}}, \Lambda)$ is concentrated around $\sum_i \beta_i l(x_i^{\text{tr}}, \Theta)$, if we condition $Y|X$. The first lemma is proved in the appendix.

Lemma 1.7 *Given assumptions 1.1 and 1.6 are satisfied, and $X_{\text{tr}}, X_{\text{te}}$ iid samples drawn from P_{tr} and P_{te}, respectively. Let $\mathcal{G}$ be a class of loss-induced functions $l(x, \theta)$ with $\|\Theta\| \leq C$. Finally, assume that there exist some β_i such that*

$$\left\| \frac{1}{n_{\text{tr}}} \sum_{i=1}^{n_{\text{tr}}} \beta_i \Psi(x_i^{\text{tr}}) - \frac{1}{n_{\text{te}}} \sum_{i=1}^{n_{\text{te}}} \Psi(x_i^{\text{te}}) \right\| \leq \epsilon \, .$$

In this case we can bound the empirical risk estimates as

$$\sup_{l(\cdot,\cdot,\theta) \in \mathcal{G}} \left| \mathbf{E}_{y|x} \left[\frac{1}{n_{\text{tr}}} \sum_{i=1}^{n_{\text{tr}}} \beta_i l(x_i^{\text{tr}}, y_i^{\text{tr}}, \theta) \right] - \mathbf{E}_{y|x} \left[\frac{1}{n_{\text{te}}} \sum_{i=1}^{n_{\text{te}}} l(x_i^{\text{te}}, y_i^{\text{te}}, \theta) \right] \right| \leq C\epsilon. \tag{1.19}$$

2. We use the same constant C to bound both $\|\Theta\|$ and$\|\Lambda\|$ for ease of notation, and without loss of generality.

The next step in relating a reweighted empirical average using $(X_{\mathrm{tr}}, Y_{\mathrm{tr}})$ and the expected risk with respect to $P(y|x)$ requires us to bound deviations of the first term in (1.19). The required lemma is again proved in the appendix.

Lemma 1.8 *Given Assumption 1.6, samples y_i^{tr} drawn for each x_i^{tr} according to $\mathrm{P}(y|x)$, and $M := n_{\mathrm{tr}}^2 / \|\beta\|_2^2$, then with probability at least $1-\delta$ over all $y|x$*

$$\sup_{l(\cdot,\cdot,\theta)\in\mathcal{G}} \left| \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \beta_i l(x_i^{\mathrm{tr}}, y_i^{\mathrm{tr}}, \theta) - \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \beta_i l(x_i^{\mathrm{tr}}, \theta) \right| \leq (2 + \sqrt{2\log(2/\delta)}) CR/\sqrt{M}.$$

We can now combine the bounds from both lemmas to obtain the main result of this section.

Corollary 1.9 *Under the assumptions of Lemma 1.7 and 1.8 we have that with probability at least $1-\delta$,*

$$\sup_{l(\cdot,\cdot,\theta)\in\mathcal{G}} \left| \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \beta_i l(x_i^{\mathrm{tr}}, y_i^{\mathrm{tr}}, \theta) - \mathbf{E}_{y|x} \left[\frac{1}{n_{\mathrm{te}}} \sum_{i=1}^{n_{\mathrm{te}}} l(x_i^{\mathrm{te}}, y_i^{\mathrm{te}}, \theta) \right] \right| \leq \frac{(2 + \sqrt{2\log(2/\delta)}) CR}{\sqrt{M}} + C\epsilon. \tag{1.20}$$

This means that if we minimize the reweighted empirical risk we will, with high probability, be minimizing an upper bound on the expected risk on the test set.

Note that we have an upper bound on ϵ via Lemma 1.5, although this assumes the β_i correspond to the importance weights. The encouraging news is that as *both* n_{tr} and $n_{\mathrm{te}} \to \infty$ we will obtain a minimizer of the conditional expected risk on P_{te}. That said, if the test set is small, it is very likely that the deviations introduced by the finite test set will give rise to more uncertainty, which implies that additional training data will be of limited use.

While the above result applies in the case of linear loss functions, we expect a similar approach to hold more generally. The key requirement is that the expected loss be a *smooth* function in the patterns x.

1.4.2 Bounds in Expectation and Cross validation

There are two more important cases worth analyzing: when carrying out covariate shift correction (or transduction) we may still want to perform model selection by methods such as cross validation. In this case we need *two* estimators of the empirical test risk — one for obtaining a regularized risk minimizer and another one for assessing the performance of the former.

A first approach is to use the reweighted training set directly for this purpose similar to what was proposed by Sugiyama et al. [2006]. This will give us an estimate of the loss on the test set, albeit biased by the deviation between the reweighted means, as described in Corollary 1.9.

A second approach is to use a modification of the cross validation procedure by partitioning first and reweighting second. That is, in 10-fold cross validation one would first partition the training set and then compute correcting weights for both the $\frac{9}{10}$-th fraction used in training and the $\frac{1}{10}$-th fraction used for validation. While this increases the cost of computing weights considerably (we need to compute a total of $10 + 10 + 1 = 21$ weighting schemes for model selection and final estimates in 10-fold cross validation), "transductive cross validation" nonetheless offers a reduction in sampling bias. Again, the bounds of Corollary 1.9 apply directly.

Finally, let us briefly consider the situation where we have a reference unlabeled dataset which is drawn from the same distribution as the actual test set, yet it is not identical with the test set. In this case, risk bounds similar to Lemma 1.5 and Corollary 1.9 can be obtained. The proof is essentially identical to that of the previous section. Hence we only state the result.

Lemma 1.10 *In addition to the conditions of Lemma 1.4, assume that* $\mathrm{P}_{\mathrm{te}} = \beta(x)\mathrm{P}_{\mathrm{tr}}$*, and* $\|\Phi(x)\| \leq R$ *for all* $x \in \mathcal{X}$*. Then with probability at least* $1 - \delta$

$$\left\| \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \beta(x_i^{\mathrm{tr}})\Phi(x_i^{\mathrm{tr}}) - \mathbf{E}_{\mathrm{P}_{\mathrm{te}}}\left[\Phi(x^{\mathrm{te}})\right] \right\| \leq \left(1 + \sqrt{2\log 2/\delta}\right) RB/\sqrt{n_{\mathrm{tr}}} \quad (1.21)$$

This also can be used in combination with Lemma 1.5, via a triangle inequality, to bound deviations of $\sum_i \beta_i \Phi(x_i^{\mathrm{tr}})$ from $\mathbf{E}_{\mathrm{P}_{\mathrm{te}}}[\Phi(x)]$ whenever the deviation between the two reweighted empirical samples is minimized as in (1.17).

To obtain a large deviation result with respect to the expected loss in $\mathrm{P}_{\mathrm{te}}(x, y)$, one would simply need to combine Lemma 1.10 with a uniform convergence bound, e.g. the bounds by Mendelson [2003].

1.5 The Connection to Single Class Support Vector Machines

1.5.1 Basic Setting

In single class SVM estimation [Schölkopf et al., 2001] one aims to find a function f which satisfies

$$f(x) \begin{cases} \geq \rho & \text{for typical observations } x \\ < \rho & \text{for novel observations } x \end{cases} \quad (1.22)$$

yet at the same time, f should be smooth. For functions in Reproducing Kernel Hilbert Spaces $f(x) = \langle \Phi(x), w \rangle$ this is obtained by solving the following optimization problem:

$$\underset{w,\xi}{\text{minimize}} \; C\sum_{i=1}^{n} \xi_i + \frac{1}{2}\|w\|^2 \quad (1.23a)$$

$$\text{subject to } \langle \Phi(x_i), w \rangle \geq \rho - \xi_i \text{ and } \xi_i \geq 0. \quad (1.23b)$$

Since it is desirable to have an approximately *fixed* number of observations singled out as novel it is preferable to use the ν-formulation of the problem [Schölkopf et al., 2000], which leads to

$$\underset{w,\xi,\rho}{\text{minimize}} \quad \sum_{i=1}^{n} \xi_i - \nu n \rho + \frac{1}{2} \|w\|^2 \tag{1.24a}$$

$$\text{subject to } \langle \Phi(x_i), w \rangle \geq \rho - \xi_i \text{ and } \xi_i \geq 0. \tag{1.24b}$$

The key difference is that the fixed threshold ρ has been replaced by a variable threshold, which is penalized by $\nu n \rho$. Schölkopf et al. [2000] show that for $n \to \infty$ the fraction of constraints (1.24b) being active converges to ν.

1.5.2 Relative Novelty Detection

Smola et al. [2005] show that novelty detection can also be understood as density estimation, where low-density regions are particularly emphasized, whereas high density regions beyond a certain threshold are ignored, and normalization is discarded. This means that the formulation (1.23) is equivalent to minimizing

$$C \sum_{i=1}^{n} \max \left(0, \frac{p(x_i; w)}{p_0 \exp\left(g(w)\right)} \right) + \frac{1}{2} \|w\|^2 \tag{1.25}$$

where $p(x; w)$ is a member of the exponential family, i.e. $p(x; w) = \exp\left(\langle \Phi(x), w \rangle - g(w)\right)$. Here $p_0 \exp(g(w))$ acts as a reference threshold. Observations whose density exceeds this threshold are considered typical, whereas observations below the threshold are viewed as novel. Note that $g(w)$ is the log-partition function which ensures that p is suitably normalized.

Having a fixed reference threshold may not be the most desirable criterion for novelty:

- Assume that we have a density $p(x)$ on the domain $\mathcal{X}$. Now assume that we perform a variable transformation $\psi : \mathcal{X} \to \mathcal{Z}$. In this case the measure $dp(x)$ is transformed into $dp(z) = dp(x) \left| \frac{dz(x)}{dx} \right|$. Thus a simple variable transformation could render observations novel which were considered typical before and vice versa. This is clearly undesirable.
- Assume that we already have a density model of the typical distribution of the data, e.g. a model of how stars *should* be distributed in the sky, based on prior knowledge from astrophysics. We would want to test this assumption subsequently, to discover whether and where the model has defects. This would provide us with a list of observations which are particularly *rare* with respect to this model.

Hence we would need to modify the denominator in (1.25) to reflect this modification via $p_0 \longleftarrow p_0 \left| \frac{dz(x)}{dx} \right|$ or $p_0 \longleftarrow p_{\text{model}}$.

These cases can be taken care of effectively by extending (1.23) and (1.25) to take a variable margin into account. For convenience, we do so for the variant using

the ν-trick, as it is easier to parametrize the optimization problem using ν rather than C.

$$\underset{w,\xi,\rho}{\text{minimize}} \ \sum_{i=1}^{n} \xi_i - \nu n \rho + \frac{1}{2} \|w\|^2 \tag{1.26a}$$

$$\text{subject to } \langle \Phi(x_i), w \rangle \geq \rho_i + \rho - \xi_i \text{ and } \xi_i \geq 0. \tag{1.26b}$$

Here $\rho_i = \log p_0(x_i)$, i.e. ρ_i denotes a reference threshold. By using standard Lagrange multiplier techniques we see that the dual problem of (1.26) is given by

$$\underset{\alpha}{\text{minimize}} \ \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j k(x_i, x_j) - \sum_{i=1}^{n} \rho_i \alpha_i \tag{1.27a}$$

$$\text{subject to } \sum_{i=1}^{n} \alpha_i = \nu n \text{ and } \alpha_i \in [0, 1]. \tag{1.27b}$$

The only difference to standard ν-style novelty detection is that in the objective function (1.27a) we have the additional linear term $\sum_i \rho_i \alpha_i$. This biases the solution towards nonzero α_i for which ρ_i is large. In other words, where the reference density $p_0(x_i)$ is large, the algorithm is more likely to find novel observations (where now novelty is defined with respect to $p_0(x)$). We state without proof an extension of the ν-property, as the proof is identical to that of Schölkopf et al. [2001]. Note that changing $\rho_i \to \rho_i + \text{const.}$ leaves the problem unchanged, as a constant offset in ρ_i with a corresponding change of $\rho \to \rho + \text{const.}$ does not change the optimality of the solution but merely leads to a constant shift in the objective function.

Theorem 1.11 (ν-Property) *Assume the solution of (1.26) satisfies $\rho \neq 0$. The following statements hold:*

1. *ν is an upper bound on the fraction of outliers.*
2. *ν is a lower bound on the fraction of SVs.*
3. *Suppose the data X were generated independently from a distribution $\mathrm{P}(x)$ which does not contain discrete components with respect to $p_0(x)$. Suppose, moreover, that the kernel is analytic and non-constant. With probability 1, asymptotically, ν equals both the fraction of SVs and the fraction of outliers.*

1.5.3 From Novelty Detection to Sample Bias Correction

Note the similarity between (1.27) and (1.17). In fact, a simple re-parametrization of (1.17) ($\beta_i \longrightarrow B\alpha_i$) makes the connection even more clear:

Lemma 1.12 *The problems (1.17) and (1.27) are equivalent subject to:*

- *The fraction of nonzero terms is set to $\nu = \frac{1}{B}$.*

- *The linear term ρ_i is given by*

$$\rho_i = \frac{n_{\mathrm{tr}}}{n_{\mathrm{te}} B} \sum_{j=1}^{n_{\mathrm{te}}} k(x_i^{\mathrm{tr}}, x_j^{\mathrm{te}}). \tag{1.28}$$

In other words, we typically will choose only a fraction of $1/B$ points for the covariate shift correction. Moreover, we will impose a higher threshold of 'typicality' for those points which are very well aligned with the mean operator. That is, typical points are more likely to be recruited for covariate shift correction.

Remark 1.13 (Connection to Parzen Windows) *Note that ρ_i can also be expressed as $\frac{n_{\mathrm{tr}}}{B} \hat{P}_{\mathrm{te}}(x)$, that is, the Parzen window density estimate of P_{te} at location x rescaled by $\frac{n_{\mathrm{tr}}}{B}$. In keeping with the reasoning above this means that we require a higher level estimate for observations which are relatively typical with respect to the test set, and a lower threshold for observations not so typical with respect to the test set.*

1.6 Experiments

1.6.1 Toy regression example

Our first experiment is on toy data, and is intended mainly to provide a comparison with the approach of Shimodaira [2000]. This method uses an information criterion to optimise the weights, under certain restrictions on P_{tr} and P_{te} (namely, P_{te} must be known, while P_{tr} can be either known exactly, Gaussian with unknown parameters, or approximated via kernel density estimation).

Our data is generated according to the polynomial regression example from [Shimodaira, 2000, Section 2], for which $\mathrm{P}_{\mathrm{tr}} \sim \mathcal{N}(0.5, 0.5^2)$ and $\mathrm{P}_{\mathrm{te}} \sim \mathcal{N}(0, 0.3^2)$ are two normal distributions. The observations are generated according to $y = -x + x^3$, and are observed in Gaussian noise with standard deviation 0.3 (see the left hand plot in Figure 1.6.1; the blue curve is the noise-free signal).

We sampled 100 training (darker circles) and testing (lighter crosses) points from P_{tr} and P_{te} respectively. We attempted to model the observations with a degree 1 polynomial. The black dashed line is a best-case scenario, which is shown for reference purposes: it represents the model fit using ordinary least squared (OLS) on the labeled test points. The solid grey line is a second reference result, derived only from the training data via OLS, and predicts the test data very poorly. The other three dashed lines are fit with weighted ordinary least square (WOLS), using one of three weighting schemes: the ratio of the underlying training and test densities, KMM, and the information criterion of Shimodaira [2000]. A summary of the performance over 100 trials is shown in Figure 1.6.1. In this case, our method outperforms the two other reweighting methods. Note that in this case the model (linear) is much simpler than the equation describing the underlying curve (higher order polynomial).

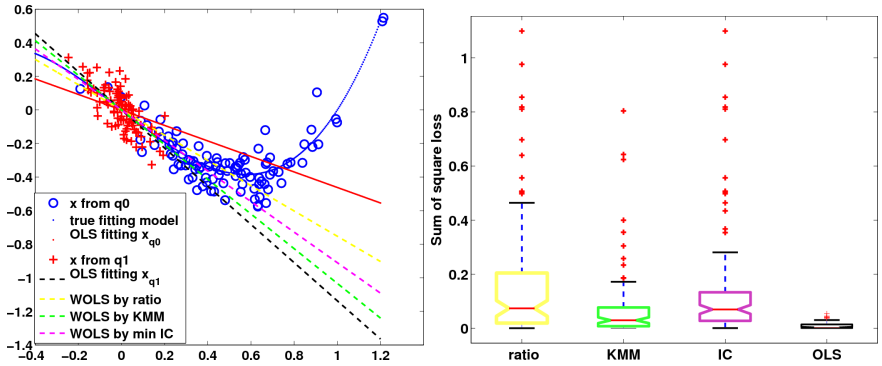

Figure 1.1 **Left:** Polynomial models of degree 1 fit with OLS and WOLS; **Right:** Average performances of three WOLS methods and OLS on this example. Labels are *Ratio* for ratio of test to training density; KMM for our approach; *min IC* for the approach of Shimodaira [2000]; and *OLS* for the model trained on the labeled test points.

1.6.2 Real world datasets

We next test our approach on real world data sets, from which we select training examples using a deliberately biased procedure (as in Zadrozny [2004], Rosset et al. [2004]). To describe our biased selection scheme, we need to define an additional random variable s_i for each point in the pool of possible training samples, where $s_i = 1$ means the ith sample is included, and $s_i = 0$ indicates an excluded sample. Two situations are considered: the selection bias corresponds to our key assumption 1.1 regarding the relation between the training and test distributions, and $P(s_i = 1|x_i, y_i) = P(s_i|x_i)$; or s_i is dependent only on y_i, i.e. $P(s_i|x_i, y_i) = P(s_i|y_i)$, which potentially creates a greater challenge since it violates this assumption. The training and test data were generated by splitting the original dataset at random, and then resampling the training data according to the biasing scheme. The combination of splitting and biased resampling was repeated to obtain an averaged value of test performance. Note that all data features were normalized to zero mean and unit standard deviation *before* any other procedure was applied (including training/test set splits and biased resampling of the training set).

In the following, we compare our method (labeled *KMM*) against two others: a baseline unweighted method (*unweighted*), in which no modification is made, and a weighting by the inverse of the true sampling distribution (*importance sampling*), as in Zadrozny [2004], Rosset et al. [2004]. We emphasise, however, that our method does *not* require any prior knowledge of the true sampling probabilities. We used a Gaussian kernel $\exp(-|x_i - x_j|^2/(2\sigma^2))$ in our kernel classification and regression algorithms, besides for the microarray data (in Section 1.6.3), where we used a linear kernel. For kernel mean matching, we always used a Gaussian kernel with identical size to the kernel in the learning algorithm. In the case of the microarray data, we did not have this reference value, and thus set the kernel size to the median distance between sample points. We set the parameters $\epsilon = (\sqrt{m} - 1)/\sqrt{m}$ and $B = 1000$ in the optimization (1.17). Note that using the same kernel size for the learning

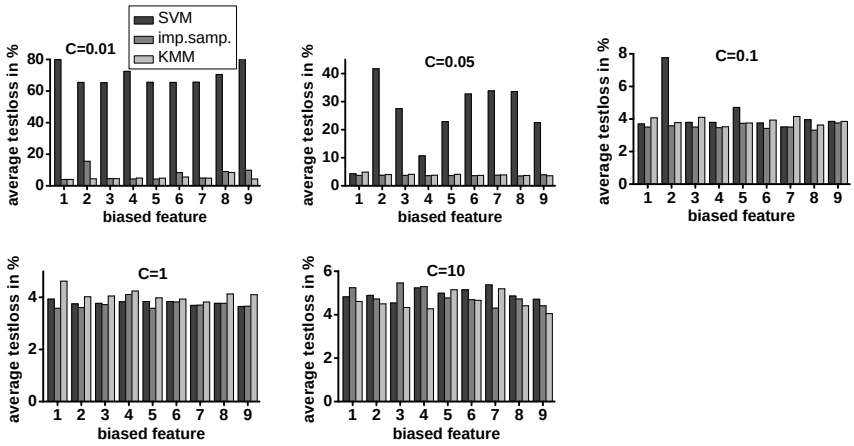

Figure 1.2 Classification performance on UCI breast cancer data. An individual feature bias scheme was used. Test error is reported on the y-axis, and the feature being biased on the x-axis.

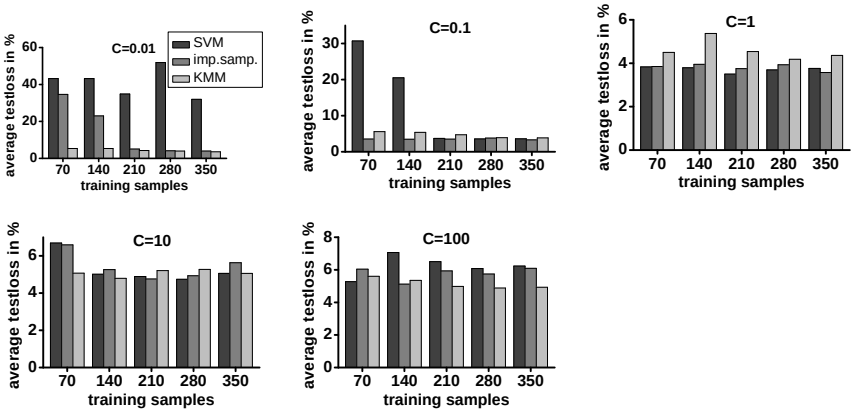

Figure 1.3 Classification performance on UCI breast cancer data. A joint feature bias scheme was used. Test error is reported on the y-axis, and the initial number of training points (prior to biased training point selection) on the x-axis.

algorithms and the bias correction has no guarantee of being optimal. The choice of optimal kernel size for KMM remains an open question (see conclusions for a suggestion on further work in this direction). The choice of B above is likewise a heuristic, and was sufficiently large that none of the β_i reached the upper bound. When B was reduced to the point where a small percentage of the β_i reached B, we found empirically on several Table 1.1 datasets that performance either did not change, or worsened.

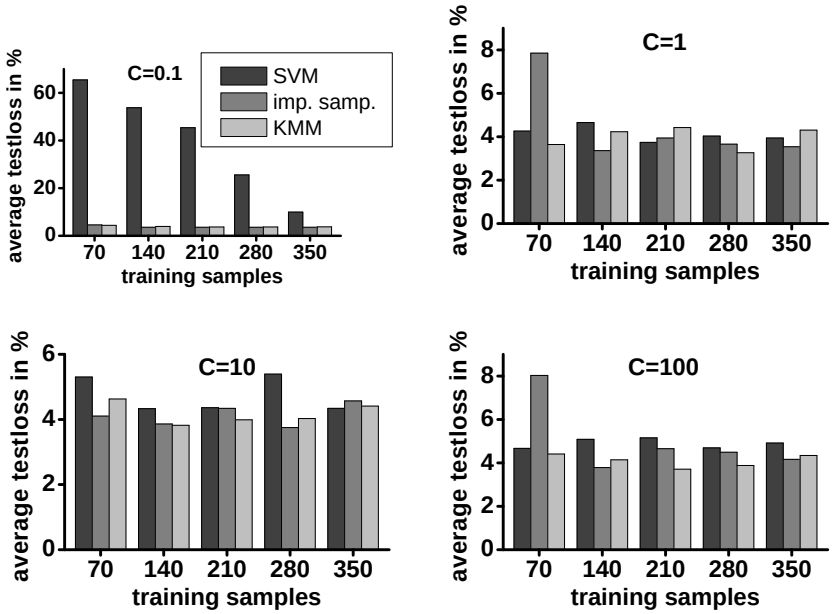

Figure 1.4 Classification performance on UCI breast cancer data. A label bias scheme was used. Test error is reported on the y-axis, and the initial number of training points (prior to biased training point selection) on the x-axis.

1.6.2.1 Breast Cancer Dataset

Before providing a general analysis across multiple datasets, we take a detailed look at one particular example: the Breast Cancer dataset from the UCI Archive. This is a binary classification task, and includes 699 examples from 2 classes: benign (positive label) and malignant (negative label). Our first experiments explore the effect of varying C on the performance of covariate shift correction, in the case of a support vector classifier. This is of particular interest since C controls the tradeoff between regularization and test error (see eq. (1.4)): small values of C favour smoothness of the decision boundary over minimizing the loss. We fix the kernel size to $\sigma = \sqrt{5}$, and vary C over the range $C \in \{0.01, 0.1, 1, 10, 100\}$. Test results always represent an average over 15 trials (a trial being a particular random split of the data into training and test sets).

First, we consider a biased sampling scheme based on the input features, of which there are nine, with integer values from 0 to 9. The data were first split into training and test sets, with 25% of data reserved for training. Since smaller feature values predominate in the unbiased data, the test set was subsampled according to $P(s = 1|x \leq 5) = 0.2$ and $P(s = 1|x > 5) = 0.8$. This subsampling was repeated for each of the features in turn. Around 30%-50% of the training points were retained by the biased sampling procedure (the exact percentage depending on the feature in question). Average performance is shown in Figure 1.2.

Second, we consider a sampling bias that operates jointly across multiple features. The data were randomly split into training and test sets, where the proportion of examples used for training varied from 10% to 50%. We then subsampled the training set, selecting samples less often when they were further from the sample mean $\overline{x}$ over the training data, i.e. $P(s_i|x_i) \propto \exp(-\gamma\|x_i - \overline{x}\|^2)$ where $\gamma = 1/20$. Around 70% of the training points were retained after the resampling. A performance comparison is given in Figure 1.3.

Finally, we consider a simple biased sampling scheme which depends only on the label y: $P(s = 1|y = 1) = 0.1$ and $P(s = 1|y = -1) = 0.9$ (the data have on average twice as many positive as negative examples when uniformly sampled). Prior to this sampling, the data were again randomly split into training and test sets, with a training proportion from 10% to 50%. Around 40% of the training points were retained following the biased sampling procedure. Average performance is plotted in Figure 1.4.

In all three of the above examples, by far the greatest performance advantage for both importance sampling and KMM-based reweighting is for small values of C (and thus, for classifiers which put a high priority on a smooth decision boundary). It is remarkable how great an improvement is found in these cases: the error reduces to the point where it is very close to its value for optimal choice of C, even though the unweighted error is on occasion extremely high. This advantage also holds for bias over the labels, despite this violating our key assumption 1.1. Somewhat surprisingly, we also see that covariate shift correction confers a small advantage for very large values of C. While this is seen in all three experiments, it is particularly apparent in the case of joint bias on the features (Figure 1.2), where - besides for the smallest training sample size - KMM consistently outperforms the unweighted and importance sampling cases.

For values $C \in \{1, 10\}$ which fall between these extremes, however, KMM does not have a consistent effect on performance, and often makes performance slightly worse. In other words, the classifier is sufficiently powerful that it is able to learn correctly over the entire input space, regardless of the weighting of particular training points.

We conclude that for the UCI breast cancer data, covariate shift correction (whether by importance sampling or KMM) has the advantage of widening the range of C values for which good performance can be expected (and in particular, greatly enhancing performance at the lowest C levels), at the risk of slightly worsening performance at the optimal C range. Our conclusions are mixed, however, regarding the effect on classifier performance of the number of training points. For small C values and label bias, the unweighted classification performance approaches the importance sampling and KMM performance with increasing training sample size (Figure 1.4). No such effect is seen in the case of joint feature bias (Figure 1.3), nor are there any clear trends for larger values of C.

We now address the question of cross-validating over σ, in accordance with the first procedure described in Section 1.4.2: i.e., on the weighted training sample, without using a second weighting procedure on the validation set. This can be very

costly due to our use of the same σ for kernel mean matching as for classification: we need to recompute the β for each new σ value. That said, we anticipate that for close to optimal parameter settings, for a sufficiently powerful class of learning algorithms, the performance optimum for cross validation over σ will occur at roughly the same location for the weighted and unweighted sample (we bear in mind the point made by Sugiyama et al. [2007] that cross-validation on the unweighted training data introduces an additional source of bias in the resulting test error estimate, for cases of covariate shift). We are led to this conjecture by the similar performance of the classifier at intermediate C values for the weighted and unweighted data (Figures 1.2, 1.3, and 1.4). The cross validation (CV) performance of the classifier for fixed $C = 10$, $\sigma \in \{0.1, 1, 10, 100, 1000\}$, and a 9:1 training:validation split is shown in Figure 1.6.2.1, in the case of joint bias on the features and an initial training sample size of 70 (prior to resampling; around 75% of training points were retained following resampling). We note that the optimum performance is obtained for the same value $\sigma = 10$ in all cases (unweighted, importance weighted with unweighted CV, importance weighted with weighted CV, KMM with unweighted CV, KMM with weighted CV), although in both KMM cases the advantage of $\sigma = 10$ over $\sigma = 1$ is negligible. Thus, in subsequent experiments, we cross-validate on the unweighted data.

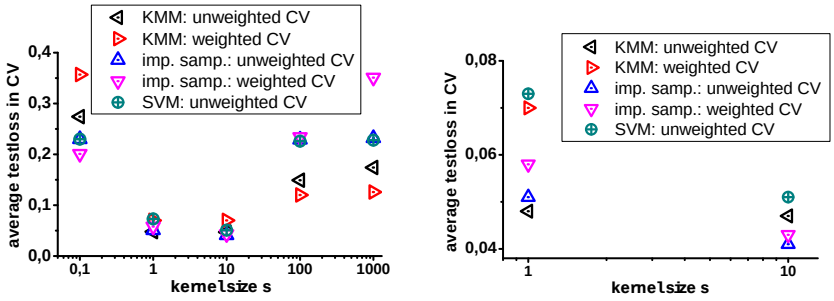

Figure 1.5 **Left:** Cross validation error vs σ for unweighted SVM, and weighted and unweighted cross validation scores for SVM with importance sampling and KMM reweighted data; **Right:** Zoomed version of the left hand plot, showing performance for $\sigma = 1$ and $\sigma = 10$. Note: in the case of weighted cross-validation, the *weighted CV error* $\frac{1}{\sum_i \beta_i} \sum_i \beta_i I_{y_i \neq f(x_i)}$ is plotted.

1.6.2.2 Further Benchmark Datasets

A question of particular interest is whether dataset shift correction can improve performance when the learning algorithm parameters are chosen by cross-validation, rather than being chosen to be "simpler" than suggested by the data (as we saw in Figures 1.2, 1.3, and 1.4 with small C values). Thus, we compare performance of various learning algorithms on both unweighted and weighted training data from

further benchmark datasets.[3] We selected training data via three biased sampling schemes. For sampling distribution bias on labels, we used either $P(s = 1|y) = \exp(a + by)/(1 + \exp(a + by))$ (denoted *label(a,b)*), or the simple step distribution $P(s = 1|y = 1) = a$, $P(s = 1|y = -1) = b$ (denoted *simple label*). For the remaining datasets, we generated biased sampling schemes over the features. We first did PCA, selecting the first principal component of the training data and the corresponding projection values. Denoting the minimum value of the projection as m and the mean as $\overline{m}$, we applied a normal distribution with mean $m + (\overline{m} - m)/a$ and variance $(\overline{m} - m)/b$ as the biased sampling scheme. Detailed parameter settings are given in Table 1.1. Our learning algorithms were penalized LMS for regression, and SVM for classification. We used a Gaussian kernel for both the kernel mean matching and the SVM/LMS regression. The kernel size was chosen by ten-fold cross validation on the unweighted training data over the set $\sigma \in \{0.1, 1, 10, 100, 1000\}$. This cross-validation procedure was also used to search over the remaining parameters $C \in \{0.1, 1, 10, 100, 1000\}$ (for classification) or $\lambda \in \{1e-3, 1e-2, 0.1, 1, 10\}$ (for regression). To evaluate generalization performance, we applied the normalized mean square error (NMSE) given by $\frac{1}{n_{\text{te}}} \sum_{i=1}^{n_{\text{te}}} \frac{(y_i^{\text{te}} - \mu_i)^2}{\text{var } y^{\text{te}}}$ for regression problems, and the average test error for classification problems. Results are listed in Table 1.1.

The results from our experiments are mixed. In certain cases, both importance sampling and KMM give similar results, which improve on the performance of the unweighted case. These datasets are (7b,11,13,14). In one case (8), KMM alone improves performance; in two further cases (3,7a), importance sampling improves performance, whereas KMM does not. That said, the sampling bias for the latter two datasets violates assumption 1.1, and the result is not surprising.

In a large number of cases, however, both for classification and regression, there is very little difference between the original, importance weighted, and KMM-corrected results. In the case of regression, these datasets are (1,4,9,10); for classification, they are (5a,6b). Performance can even worsen due to the application of KMM weighting and/or importance sampling. In some cases, the KMM correction alone gives worse results (2,6a). In the case of dataset 6a, the failure of KMM is unsurprising, since assumption 1.1 does not hold. KMM does not necessarily fail in this circumstance, however: in dataset 7a, there is little difference compared with the unweighted case (although importance sampling improves performance). In yet further instances, importance sampling worsens performance, but KMM has no effect (3,15). Finally, there exist cases where both KMM and importance sampling worsen performance (5b,12). We note that mixed results were also reported independently for KMM by Sugiyama et al. [2008, Table 1], with performance being improved or unchanged for good kernel size choice (KMM(0.3) in this table), and worsening for poor kernel choice.

3. Regression data from `http://www.liacc.up.pt/~ltorgo/Regression/DataSets.html`; classification data from UCI.

In comparison with the results of Huang et al. [2007, Table 1], the current results are less favourable to both KMM and importance sampling: in particular, in the earlier work, KMM always improved performance. This is because our earlier experiments used parameters resulting in an overly simple classification/regression function (in particular, the kernels sizes used were relatively large: see the corresponding column in [Huang et al., 2007, Table 1]). We conclude from our Table 1.1 results that while covariate shift can still improve performance in cases where the classification/regression parameters are chosen by cross validation, this is not guaranteed; moreover, we have yet to determine what properties of these particular data are favourable to covariate shift. On the other hand, the application of covariate shift correction through KMM/importance sampling can decrease performance in this case, though the penalty is not generally too large.

Table 1.1 Test results for three methods on 15 datasets with different sampling schemes. The results are averages over 10 trials for regression problems (marked *) and 20 trials for classification problems. Sampling schemes: *simple label:* $P(s = 1|y = 1) = 0.1$ and $P(s = 1|y = -1) = 0.9$, *label(a,b):* $P(s = 1|y) = \exp(a + by)/(1 + \exp(a + by))$, *PCA(a,b,$\sigma_{PCA}$):* bias using kernel PCA with parameters a, b, σ_{PCA}. The training set size is in column n_{tr}, the number of training points after biased subsampling is in column *sel.*, and the number of test points is in column n_{tst}.

					NMSE/test error ± std. error		
Dataset	sampling scheme	n_{tr}	*sel.*	n_{tst}	SVM	importance samp.	KMM
1. Abalone*	label(1,10)	2000	973	2177	0.83±0.02	0.80±0.04	0.83±0.03
2. CA Housing*	PCA(10,5,0.1)	8000	1591	12640	0.694±0.005	0.684±0.009	0.728±0.007
3. Delta Ailerons*	label(1,10)	4000	1980	3129	0.64±0.01	0.405±0.009	0.613±0.008
4. Ailerons*	PCA(1e3,4,0.1)	7154	726	6596	0.25±0.04	0.27±0.04	0.24±0.03
5a. Haberman	label(0.2,0.8)	150	68	156	0.30±0.02	0.32±0.02	0.33±0.01
5b. Haberman	PCA(2,2,0.01)	150	82	156	0.266±0.008	0.318±0.008	0.33±0.02
6a. USPS(6vs8)	simple label	500	264	1042	0.035±0.004	0.034±0.004	0.047±0.004
6b. USPS(6vs8)	PCA(3,3,1/128)	500	169	1042	0.17±0.04	0.19±0.05	0.19±0.04
7a. USPS(3vs9)	simple label	500	261	1145	0.020±0.004	0.014±0.002	0.020±0.003
7b. USPS(3vs9)	PCA(3,3,1/128)	500	165	1145	0.15±0.03	0.056±0.007	0.08±0.02
8. Bank8FM*	PCA(3,6,0.1)	4500	589	3692	0.10±0.02	0.12±0.02	0.068±0.003
9. Bank32nh*	PCA(3,6,0.01)	4500	673	3692	0.523±0.008	0.54±0.03	0.555±0.008
10. cpu-act*	PCA(4,2,1e-12)	4000	1672	4192	0.09±0.03	0.08±0.03	0.10±0.02
11. cpu-small*	PCA(4,2,1e-12)	4000	1674	4192	0.32±0.09	0.15±0.07	0.11±0.02
12. Delta Ailerons*	PCA(1e3,4,0.1)	4000	511	3129	0.38±0.02	0.41±0.03	0.44±0.03
13. Boston house*	PCA(2,4,1e-4)	300	100	206	0.63±0.08	0.5±0.2	0.50±0.04
14. kin8nm*	PCA(8,5,0.1)	5000	292	3192	1.0±0.3	0.72±0.02	0.74±0.04
15. puma8nh*	PCA(4,4,0.1)	4499	685	3693	0.75±0.03	0.83±0.06	0.75±0.02

Table 1.2 Covariate shift correction for microarray data. The notation "Gruvberger→West" indicates that we train on the data of Gruvberger and test on that of West.

	test error		
Dataset	SVM	importance sampling	KMM
Singh	0.40±0.02	0.091±0.006	0.083±0.005
Gruvberger→West	0.061	—	0.061
West→Gruvberger	0.086	—	0.052
Dhanasekaran→Welsh	0.03	—	0.09
Welsh→Dhanasekaran	0.26	—	0.17

1.6.3 Tumor Diagnosis using Microarrays

Our next benchmark is a dataset of 102 microarrays from prostate cancer patients [Singh et al., 2002]. Each of these microarrays measures the expression levels of 12,600 genes. The dataset comprises 50 samples from normal tissues (positive label) and 52 from tumor tissues (negative label). We simulate the realistic scenario that two sets of microarrays A and B are given with dissimilar proportion of tumor samples, and we want to perform cancer diagnosis via classification, training on A and predicting on B. As a preprocessing step, the data were normalised to have zero mean and unit variance for each feature. We selected training examples via a biased selection scheme as $P(s = 1|y = 1) = 0.85$ and $P(s = 1|y = -1) = 0.15$; the remaining data points form the test set. We performed SVM classification using a linear SVM setting $C = 1000$ (there being too little data for cross-validation), for the unweighted, the KMM, and the importance sampling approaches. In the case of KMM, the kernel size was the median distance between training sample points. Results are given in Table 1.2, and represent the average performance over 50 training/test splits. We note that both importance sampling and KMM result in a substantial performance improvement, with KMM outperforming importance sampling (despite the violation of assumption 1.1).

We now use the same setting to investigate dataset shift for microarray studies on the same tissue by different labs. We first consider two breast cancer microarray datasets from Gruvberger et al. [2001] and West et al. [2001], measuring the expression levels of 2,166 common genes for normal and cancer patients [Warnat et al., 2005]. All settings for the data preprocessing, the SVM, and KMM, were identical to our first experiment. Results are listed in Table 1.2, and describe both training on *West* and testing on *Gruvberger*, as well as training on *Gruvberger* and testing on *West*.[4] In the former case, KMM causes a performance improvement compared with the unweighted data; in the latter case, performance remains constant.

4. Note: since the biasing scheme for these data is not know, there is no importance sampling result.

Finally, we study the same scenario for two prostate cancer datasets, Dhanasekaran et al. [2001] vs Welsh et al. [2001]. Results are again in Table 1.2. In this case our results are mixed: while training on *Welsh* and testing on *Dhanasekaran* demonstrates a substantial performance gain when using KMM, the reverse procedure results in a (smaller) performance reduction for KMM. We conclude that while KMM more often results in performance increases in microarray data than in the UCI benchmark sets of the previous section, this performance improvement is not guaranteed.

1.7 Conclusion

We present a new approach, kernel mean matching (KMM), for dealing with sampling bias in various learning problems. We directly estimate the resampling weights by matching training and test distribution feature means in a reproducing kernel Hilbert space. In addition, we develop bounds on the mean matching error, and transductive risk bounds, based on the maximum ratio of the distributions and the sample sizes.

In our experiments, it appears that with properly chosen parameters (via cross validation), kernel classification and regression methods occasionally benefit from covariate shift correction, but for the most part do not. This is true both when the correction is made using KMM, and via the "optimal" reweighting given by the ratio of test and training probabilities (note that the latter is unavailable in real world applications). We also emphasise that our results were obtained using the heuristic that the KMM kernel size was set to the kernel size of the classification/regression algorithm. Sugiyama et al. [2008, Table 1] demonstrated that kernel size has a strong effect on KMM performance (though no comparison was made between the optimal KMM kernel size and that chosen by cross validation for the learning algorithm). Thus, performance of KMM might be further improved by a more principled strategy for KMM kernel choice.[5]

Major benefits can obtained from covariate shift correction when a simple classification/regression function is used. There are several reasons for not using a "correct" model, but rather a deliberately simpler one: these include interpretability on one hand; and on the other hand difficulties in correct model selection by cross-validation, especially for higher dimensional data and small sample sizes (for instance in our microarray experiments, where we used a linear classifier, and where the performance of KMM was generally better). Covariate shift correction allows us to make use of these simpler models without too significant a performance penalty.

5. One approach might be along the lines of [Fukumizu et al., 2008], where the variance of a kernel dependence statistic was computed via both a closed form expression and random permutations of the sample: a good kernel size caused these quantities to match. In our case, the relevant statistic is a difference in RKHS means, so an appropriate closed form variance expression might derive from Gretton et al. [2007].

Acknowledgements: The authors thank Patrick Warnat (DKFZ, Heidelberg) for providing the microarray datasets, and Paul von Bünau, Olivier Chapelle, Matthias Hein, Quoc Le, and Klaus-Robert Müller for helpful discussions. The work is partially supported by by the German Ministry for Education, Science, Research and Technology (BMBF) under grant 031U112F within the Bioinformatics for the Functional Analysis of Mammalian Genomes project, which is part of the German Genome Analysis Network. National ICT Australia is funded through the Australian Government's *Backing Australia's Ability* initiative, in part through the Australian Research Council. This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778.

1.8 Proofs

Proof: [Proof of Theorem 1.2] By definition μ is surjective on the marginal polytope, since the latter is defined as the set of all expectations of $\Phi(x)$. We now prove injectivity.

Let $\mathcal{F}$ be a universal RKHS, and let $\mathcal{G}$ be the unit ball in $\mathcal{F}$. We need to prove that $\mathrm{P}_{\mathrm{tr}} = \mathrm{P}_{\mathrm{te}}$ if $\mu(\mathrm{P}_{\mathrm{tr}}) = \mu(\mathrm{P}_{\mathrm{te}})$, or equivalently $\|\mu(\mathrm{P}_{\mathrm{tr}}) - \mu(\mathrm{P}_{\mathrm{te}})\| = 0$. We have

$$\begin{aligned}\|\mu(\mathrm{P}_{\mathrm{tr}}) - \mu(\mathrm{P}_{\mathrm{te}})\| &= \sup_{f \in \mathcal{G}} \langle f, \mu(\mathrm{P}_{\mathrm{tr}}) - \mu(\mathrm{P}_{\mathrm{te}})\rangle \\ &= \sup_{f \in \mathcal{G}} \left(\mathbf{E}_{\mathrm{P}_{\mathrm{tr}}}[f] - \mathbf{E}_{\mathrm{P}_{\mathrm{te}}}[f]\right) \\ &=: \Delta\left[\mathcal{G}, \mathrm{P}_{\mathrm{tr}}, \mathrm{P}_{\mathrm{te}}\right].\end{aligned}$$

We use a result from Dudley [2002, Lemma 9.3.2]: If $\mathrm{P}_{\mathrm{tr}}, \mathrm{P}_{\mathrm{te}}$ are two Borel probability measures defined on a separable metric space $\mathcal{X}$, then $\mathrm{P}_{\mathrm{tr}} = \mathrm{P}_{\mathrm{te}}$ if and only if $\mathbf{E}_{\mathrm{P}_{\mathrm{tr}}}[f] = \mathbf{E}_{\mathrm{P}_{\mathrm{te}}}[f]$ for all $f \in C(\mathcal{X})$, where $C(\mathcal{X})$ is the space of continuous bounded functions on $\mathcal{X}$. If we can show that $\Delta\left[C(\mathcal{X}), \mathrm{P}_{\mathrm{tr}}, \mathrm{P}_{\mathrm{te}}\right] = D$ for some $D > 0$ implies $\Delta\left[\mathcal{G}, \mathrm{P}_{\mathrm{tr}}, \mathrm{P}_{\mathrm{te}}\right] > 0$: this is equivalent to $\Delta\left[\mathcal{G}, \mathrm{P}_{\mathrm{tr}}, \mathrm{P}_{\mathrm{te}}\right] = 0$ implying $\Delta\left[C(\mathcal{X}), \mathrm{P}_{\mathrm{tr}}, \mathrm{P}_{\mathrm{te}}\right] = 0$ (where this last result implies $\mathrm{P}_{\mathrm{tr}} = \mathrm{P}_{\mathrm{te}}$). If $\Delta\left[C(\mathcal{X}), \mathrm{P}_{\mathrm{tr}}, \mathrm{P}_{\mathrm{te}}\right] = D$, then there exists some $\tilde{f} \in C(\mathcal{X})$ for which $\mathbf{E}_{\mathrm{P}_{\mathrm{tr}}}\left[\tilde{f}\right] - \mathbf{E}_{\mathrm{P}_{\mathrm{te}}}\left[\tilde{f}\right] \geq D/2$. By definition of universality, $\mathcal{F}$ is dense in $C(\mathcal{X})$ with respect to the L_∞ norm: this means that for all $\epsilon \in (0, D/8)$, we can find some $f^* \in \mathcal{F}$ satisfying $\left\|f^* - \tilde{f}\right\|_\infty < \epsilon$. Thus, we obtain $\left|\mathbf{E}_{\mathrm{P}_{\mathrm{tr}}}\left[f^*\right] - \mathbf{E}_{\mathrm{P}_{\mathrm{tr}}}\left[\tilde{f}\right]\right| < \epsilon$ and consequently

$$\left|\mathbf{E}_{\mathrm{P}_{\mathrm{tr}}}\left[f^*\right] - \mathbf{E}_{\mathrm{P}_{\mathrm{te}}}\left[f^*\right]\right| > \left|\mathbf{E}_{\mathrm{P}_{\mathrm{tr}}}\left[\tilde{f}\right] - \mathbf{E}_{\mathrm{P}_{\mathrm{te}}}\left[\tilde{f}\right]\right| - 2\epsilon > \tfrac{D}{2} - 2\tfrac{D}{8} = \tfrac{D}{4} > 0.$$

Finally, using $\|f^*\| < \infty$, we have

$$\left[\mathbf{E}_{\mathrm{P}_{\mathrm{tr}}}\left[f^*\right] - \mathbf{E}_{\mathrm{P}_{\mathrm{te}}}\left[f^*\right]\right] / \|f^*\| \geq D/(4\,\|f^*\|) > 0,$$

and hence $\Delta\left[\mathcal{G}, \mathrm{P}_{\mathrm{tr}}, \mathrm{P}_{\mathrm{te}}\right] > 0$.

For the proof of Lemma 1.5 we need a result by McDiarmid [1989].

Theorem 1.14 *Denote by $f(x_1, \ldots, x_n)$ a function of n independent random variables. Moreover let*

$$|f(x_1, \ldots, x_n) - f(x_1, \ldots, x_{i-1}, \bar{x}, x_{i+1}, \ldots, x_n)| \leq c_i \tag{1.29}$$

for all $x_1, \ldots, x_n$ and $\bar{x}$. Denote by $C := \sum_i c_i^2$. In this case

$$\mathrm{P}\left\{|f(x_1, \ldots, x_n) - \mathbf{E}_{x_1, \ldots x_n}\left[f(x_1, \ldots, x_n)\right]| > \epsilon\right\} < 2\exp(-2\epsilon^2/C). \tag{1.30}$$

Proof: [Proof of Lemma 1.5] Let

$$\Xi(X_{\mathrm{tr}}, X_{\mathrm{te}}) := \left\| \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \beta(x_i^{\mathrm{tr}}) \Phi(x_i^{\mathrm{tr}}) - \frac{1}{n_{\mathrm{te}}} \sum_{i=1}^{n_{\mathrm{te}}} \Phi(x_i^{\mathrm{te}}) \right\|. \tag{1.31}$$

The proof follows firstly by its tail behavior using a concentration inequality, and subsequently by bounding the expectation.

To apply McDiarmid's tail bound, we need to bound the change in $\Xi(X_{\mathrm{tr}}, X_{\mathrm{te}})$ if we replace any x_i^{tr} by an arbitrary $x \in \mathcal{X}$ and likewise if we replace any x_i^{te} by some $x \in \mathcal{X}$. By the triangle inequality the replacement of x_i^{tr} by x can change $\Xi(X_{\mathrm{tr}}, X_{\mathrm{te}})$ by at most $\frac{1}{n_{\mathrm{tr}}} \left\| \beta(x_i^{\mathrm{tr}}) \Phi(x_i^{\mathrm{tr}}) - \beta(x) \Phi(x) \right\| \leq \frac{2BR}{n_{\mathrm{tr}}}$. Likewise, a replacement of x_i^{te} by x changes $\Xi(X_{\mathrm{tr}}, X_{\mathrm{te}})$ by at most $\frac{2R}{n_{\mathrm{te}}}$. Since $n_{\mathrm{tr}}(2BR/n_{\mathrm{tr}})^2 + n_{\mathrm{te}}(2R/n_{\mathrm{te}})^2 = 4R^2(B^2/n_{\mathrm{tr}} + 1/n_{\mathrm{te}})$ we have

$$\begin{aligned} &\mathrm{P}\left\{ \left| \Xi(X_{\mathrm{tr}}, X_{\mathrm{te}}) - \mathbf{E}_{X_{\mathrm{tr}}, X_{\mathrm{te}}} \left[\Xi(X_{\mathrm{tr}}, X_{\mathrm{te}}) \right] \right| > \epsilon \right\} \\ &\qquad \leq 2 \exp\left(-\epsilon^2 / 2R^2 (B^2/n_{\mathrm{tr}} + 1/n_{\mathrm{te}}) \right). \end{aligned}$$

Hence with probability $1 - \delta$ the deviation of the random variable from its expectation is bounded by $\left| \Xi(X_{\mathrm{tr}}, X_{\mathrm{te}}) - \mathbf{E}_{X_{\mathrm{tr}}, X_{\mathrm{te}}} \left[\Xi(X_{\mathrm{tr}}, X_{\mathrm{te}}) \right] \right| \leq R \sqrt{2 \log \frac{2}{\delta} \left(\frac{B^2}{n_{\mathrm{tr}}} + \frac{1}{n_{\mathrm{te}}} \right)}$.

To bound the expected value of $\Xi(X_{\mathrm{tr}}, X_{\mathrm{te}})$ we use

$$\mathbf{E}_{X_{\mathrm{tr}}, X_{\mathrm{te}}} \left[\Xi(X_{\mathrm{tr}}, X_{\mathrm{te}}) \right] \leq \sqrt{\mathbf{E}_{X_{\mathrm{tr}}, X_{\mathrm{te}}} \left[\Xi(X_{\mathrm{tr}}, X_{\mathrm{te}})^2 \right]}.$$

Expanding out the expectation, and denoting by $\widetilde{x^{\mathrm{te}}}$ a random variable drawn from P_{te} and independent of x^{te}, we get

$$\begin{aligned} &\mathbf{E}_{X_{\mathrm{tr}}, X_{\mathrm{te}}} \left\| \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \beta(x_i^{\mathrm{tr}}) \Phi(x_i^{\mathrm{tr}}) - \frac{1}{n_{\mathrm{te}}} \sum_{i=1}^{n_{\mathrm{te}}} \Phi(x_i^{\mathrm{te}}) \right\|^2 \\ =& \frac{1}{n_{\mathrm{tr}}^2} \mathbf{E}_{X_{\mathrm{tr}}} \left[\sum_{i,j=1}^{n_{\mathrm{tr}}} \beta(x_i^{\mathrm{tr}}) \beta(x_j^{\mathrm{tr}}) k(x_i^{\mathrm{tr}}, x_j^{\mathrm{tr}}) \right] + \frac{1}{n_{\mathrm{te}}^2} \mathbf{E}_{X_{\mathrm{te}}} \left[\sum_{i,j=1}^{n_{\mathrm{te}}} k(x_i^{\mathrm{te}}, x_j^{\mathrm{te}}) \right] \\ &\qquad - 2 \mathbf{E}_{X_{\mathrm{tr}}, X_{\mathrm{te}}} \frac{1}{n_{\mathrm{tr}} n_{\mathrm{te}}} \left[\sum_{i=1}^{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{te}}} \beta(x_i^{\mathrm{tr}}) k(x_i^{\mathrm{tr}}, x_j^{\mathrm{te}}) \right] \\ =& \mathbf{E}_{\mathrm{P}_{\mathrm{te}}} k(x^{\mathrm{te}}, \widetilde{x^{\mathrm{te}}}) + \frac{1}{n_{\mathrm{tr}}} \mathbf{E}_{\mathrm{P}_{\mathrm{te}}} \left[\beta(x^{\mathrm{te}}) k(x^{\mathrm{te}}, x^{\mathrm{te}}) \right] + \mathbf{E}_{\mathrm{P}_{\mathrm{te}}} k(x^{\mathrm{te}}, \widetilde{x^{\mathrm{te}}}) \\ &\qquad + \frac{1}{n_{\mathrm{te}}} \mathbf{E}_{\mathrm{P}_{\mathrm{te}}} \left[k(x^{\mathrm{te}}, x^{\mathrm{te}}) \right] - 2 \mathbf{E}_{\mathrm{P}_{\mathrm{te}}} k(x^{\mathrm{te}}, \widetilde{x^{\mathrm{te}}}) + O\left(n_{\mathrm{tr}}^{-2} \right) + O\left(n_{\mathrm{te}}^{-2} \right) \\ \lessapprox& R^2 \left[B/n_{\mathrm{tr}} + 1/n_{\mathrm{te}} \right] < R^2 \left[B^2/n_{\mathrm{tr}} + 1/n_{\mathrm{te}} \right]. \end{aligned}$$

The final line uses that $B < B^2$ since $B > 1$ (due to the constraint (1.14)). Combining the bounds on the mean and the tail proves the claim.

Proof: [Proof of Lemma 1.7] To see the claim, first note that by Assumption 1.1 the conditional distributions $\mathrm{P}(y|x)$ are the same for P_{tr} and P_{te}. By linearity we can apply the expectation $\mathbf{E}_{Y|X}$ to each summand individually. Finally, by

Assumption 1.6 the expected loss $l(x, \theta)$ can be written as $\langle \Psi(x), \theta \rangle$. Hence we may rewrite the LHS of (1.19) as

$$\sup_{l(\cdot,\theta)\in\mathcal{G}} \left| \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \beta_i l(x_i^{\mathrm{tr}}, \theta) - \frac{1}{n_{\mathrm{te}}} \sum_{i=1}^{n_{\mathrm{te}}} l(x_i^{\mathrm{te}}, \theta) \right| \\ \leq \sup_{\|\Theta\|\leq C} \left| \left\langle \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \beta_i \Psi(x_i^{\mathrm{tr}}) - \frac{1}{n_{\mathrm{te}}} \sum_{i=1}^{n_{\mathrm{te}}} \Psi(x_i^{\mathrm{te}}), \Theta \right\rangle \right|$$

By the definition of norms this is bounded by $C\epsilon$, which proves the claim.

Proof: [Proof of Lemma 1.8] The strategy is almost identical to that of Lemma 1.5 and of Mendelson [2003]. Let

$$\Xi(Y_{\mathrm{tr}}) := \sup_{l(\cdot,\cdot,\theta)\in\mathcal{G}} \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \beta_i \left[l(x_i^{\mathrm{tr}}, y_i^{\mathrm{tr}}, \theta) - l(x_i^{\mathrm{tr}}, \theta) \right] \tag{1.32}$$

be the maximum deviation between empirical mean and expectation. Key is that the random variables $y_1^{\mathrm{tr}}, \ldots, y_m^{\mathrm{tr}}$ are conditionally independent given X_{tr}. Replacing one y_i^{tr} by an arbitrary $y \in \mathcal{Y}$ leads to a change in $\Xi(Y_{\mathrm{tr}})$ which is bounded by $\frac{\beta_i}{n_{\mathrm{tr}}} C \left\| \Upsilon(x_i^{\mathrm{tr}}, y_i^{\mathrm{tr}}) - \Upsilon(x_i^{\mathrm{tr}}, y) \right\| \leq 2CR\beta_i/m$. Using McDiarmid's theorem we can bound

$$\mathrm{P}_{Y|X} \left\{ |\Xi(Y_{\mathrm{tr}}) - \mathbf{E}_{Y|X}\Xi(Y_{\mathrm{tr}})| > \epsilon \right\} \leq 2 \exp\left(-\epsilon^2 n_{\mathrm{tr}}^2 / \left(2C^2R^2 \left\|\beta\right\|_2^2 \right) \right). \tag{1.33}$$

In other words, $M := n_{\mathrm{tr}}^2 / \left\|\beta\right\|_2^2$ acts as an effective sample size when it comes to determining large deviations. Next we use symmetrization to obtain a bound on the expectation of $\Xi(Y_{\mathrm{tr}})$, that is

$$\mathbf{E}_{Y|X}[\Xi(Y_{\mathrm{tr}})] \leq \frac{1}{n_{\mathrm{tr}}} \mathbf{E}_{Y|X} \mathbf{E}_{\widetilde{Y}|X} \left[\sup_{l(\cdot,\cdot,\theta)\in\mathcal{G}} \left| \sum_{i=1}^{n_{\mathrm{tr}}} \beta_i l(x_i^{\mathrm{tr}}, y_i, \theta) - \beta_i l(x_i^{\mathrm{tr}}, \tilde{y}_i, \theta) \right| \right] \\ \leq \frac{2}{n_{\mathrm{tr}}} \mathbf{E}_{Y|X} \mathbf{E}_{\sigma} \left[\sup_{l(\cdot,\cdot,\theta)\in\mathcal{G}} \left| \sum_{i=1}^{n_{\mathrm{tr}}} \sigma_i \beta_i l(x_i^{\mathrm{tr}}, y_i, \theta) \right| \right]. \tag{1.34}$$

where the σ_i take values in $\{\pm 1\}$ with equal probability, and $\tilde{y}_i$ is drawn from $\mathrm{P}(\tilde{y}_i | x_i^{\mathrm{tr}})$ independently of y_i. The first inequality follows from convexity. The second one follows from the fact that all $y_i, \tilde{y}_i$ pairs are independently and identically distributed, hence we can swap these pairs.

For constant β_i the RHS in (1.34) is referred to as the Rademacher average. To make actual progress in computing this, we use the condition in assumption 1.6 that $l(x, y, \theta) = \langle \Upsilon(x, y), \Lambda \rangle$ for some Λ with $\|\Lambda\| \leq C$. This allows us to bound the supremum. This, and the convexity of x^2 yields a series of bounds on the RHS in

(1.34),

$$\begin{aligned}
\text{RHS} &\leq \frac{2}{n_{\text{tr}}} \mathbf{E}_{Y|X} \mathbf{E}_{\sigma} C \left\| \sum_{i=1}^{n_{\text{tr}}} \sigma_i \beta_i \Upsilon(x_i^{\text{tr}}, y_i) \right\| \\
&\leq \frac{2}{n_{\text{tr}}} C \sqrt{\mathbf{E}_{Y|X} \mathbf{E}_{\sigma} \left\| \sum_{i=1}^{n_{\text{tr}}} \sigma_i \beta_i \Upsilon(x_i^{\text{tr}}, y_i) \right\|^2} \\
&= \frac{2}{n_{\text{tr}}} C \sqrt{\sum_{i=1}^{n_{\text{tr}}} \beta_i^2 \mathbf{E}_{y_i|x_i^{\text{tr}}} \left\| \Upsilon(x_i^{\text{tr}}, y_i) \right\|^2} \\
&\leq \frac{2}{n_{\text{tr}}} CR \left\| \beta \right\|_2 = \frac{2CR}{\sqrt{M}}.
\end{aligned}$$

Combined with the bound on the expectation and solving the tail bound for ϵ proves the claim.

References

G. Casella and R. Berger. *Statistical Inference*. Duxbury, Pacific Grove, CA, 2nd edition, 2002.

N. A. C. Cressie. *Statistics for Spatial Data*. John Wiley and Sons, New York, 1993.

Y.-H. Dai and R. Fletcher. New algorithms for singly linearly constrained quadratic programs subject to lower and upper bounds. *Mathematical Programming: Series A and B archive*, 106 (3):403–421, 2006.

S. M. Dhanasekaran, T. R. Barrette, D. Ghosh, R. Shah, S. Varambally, K. Kurachi, K. J. Pienta, M. A. Rubin, and A. M. Chinnaiyan. Delineation of prognostic biomarkers in prostate cancer. *Nature*, 412(6849):822–826, Aug 2001.

M. Dudík, R. E. Schapire, and S. J. Phillips. Correcting sample selection bias in maximum entropy density estimation. In *Advances in Neural Information Processing Systems 17*, 2005.

R. M. Dudley. *Real analysis and probability*. Cambridge University Press, Cambridge, UK, 2002.

B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall, New York, 1994.

K. Fukumizu, A. Gretton, X. Sun, and B. Schölkopf. Kernel measures of conditional dependence. In Daphne Koller and Yoram Singer, editors, *Advances in Neural Information Processing Systems 20*, Cambridge, MA, 2008. MIT Press.

A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola. A kernel method for the two-sample-problem. In B. Schölkopf, J. Platt, and T. Hofmann, editors, *Advances in Neural Information Processing Systems 19*, volume 19. MIT Press, 2007.

S. Gruvberger, M. Ringner, Y. Chen, S. Panavally, L. H. Saal, C. Peterson A. Borg, M. Ferno, and P. S. Meltzer. Estrogen receptor status in breast cancer is associated with remarkably distinct gene expression patterns. *Cancer Research*, 61, 2001.

J. J. Heckman. Sample selection bias as a specification error. *Econometrica*, 47:153–161, 1979.

W. Hoeffding. Probability inequalities for sums of bounded random variables. *J. Amer. Statist. Assoc.*, 58:13–30, 1963.

J. Huang, A. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf. Correcting sample selection bias by unlabeled data. In B. Schölkopf, J. Platt, and T. Hofmann, editors, *Advances in Neural Information Processing Systems 19*, pages 601–608, 2007.

G. S. Kimeldorf and G. Wahba. A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. *Annals of Mathematical Statistics*, 41:495–502, 1970.

Y. Lin, Y. Lee, and G. Wahba. Support vector machines for classification in nonstandard situations. *Machine Learning*, 46:191–202, 2002.

C. McDiarmid. On the method of bounded differences. In *Survey in Combinatorics*, pages 148–188. Cambridge University Press, 1989.

S. Mendelson. A few notes on statistical learning theory. In S. Mendelson and A. J. Smola, editors, *Advanced Lectures on Machine Learning*, number 2600 in LNAI, pages 1–40. Springer-Verlag, Heidelberg, Germany, 2003.

E. Osuna. *Support Vector Machines: Training and Applications*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1998.

J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods — Support Vector Learning*, pages 185–208, Cambridge, MA, 1999. MIT Press.

S. Rosset, J. Zhu, H. Zou, and T. Hastie. A method for inferring label sampling mechanisms in semi-supervised learning. In *Advances in Neural Information Processing Systems 17*, 2004.

M. Schmidt and H. Gish. Speaker identification via support vector classifiers. In *Proc. ICASSP'96*, pages 105–108, Atlanta, GA, May 1996.

B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett. New support vector algorithms. *Neural Computation*, 12:1207–1245, 2000.

B. Schölkopf, J. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Comput.*, 13(7):1443–1471, 2001.

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

D. Singh, P. Febbo, K. Ross, D. Jackson, J. Manola, C. Ladd, P. Tamayo, A. Renshaw, A. DAmico, and J. Richie. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1 (2), 2002.

A. J. Smola, S. V. N. Vishwanathan, and T. Hofmann. Kernel methods for missing variables. In R. G. Cowell and Z. Ghahramani, editors, *Proceedings of International Workshop on Artificial Intelligence and Statistics*, pages 325–332. Society for Artificial Intelligence and Statistics, 2005.

I. Steinwart. Support vector machines are universally consistent. *J. Complexity*, 18:768–791, 2002.

M. Sugiyama and K.-R. Müller. Input-dependent estimation of generalization error under covariate shift. *Statistics & Decisions*, 23(4):249–279, 2005.

M. Sugiyama, B. Blankertz, M. Krauledat, G. Dornhege, and K.-R. Müller. Importance weighted cross-validation for covariate shift. In K. Franke, K.-R. Müller, B. Nickolay, and R. Schäfer, editors, *DAGM 2006*, pages 354–363. Springer LNCS 4174, 2006.

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

M. Sugiyama, S. Nakajima, H. Kashima, P. von Bünau, and M. Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In Daphne Koller and Yoram Singer, editors, *Advances in Neural Information Processing Systems 20*, Cambridge, MA, 2008. MIT Press.

B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 25–32, Cambridge, MA, 2004. MIT Press.

I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *J. Mach. Learn. Res.*, 6:1453–1484, 2005.

P. Warnat, R. Eils, and B. Brors. Cross-platform analysis of cancer microarray data improves gene expression based classification of phenotypes. *BMC Bioinformatics*, 6:265, Nov 2005.

J. B. Welsh, L. M. Sapinoso, A. I. Su, S. G. Kern, J. Wang-Rodriguez, C. A. Moskaluk, J. r. Frierson HF, and G. M. Hampton. Analysis of gene expression identifies candidate markers and pharmacological targets in prostate cancer. *Cancer Res*, 61(16):5974–5978, Aug 2001.

M. West, C. Blanchette, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, J. A. Olson Jr, J. R. Marks, and J. R. Nevins. Predicting the clinical status of human breast cancer by using gene expression profiles. *PNAS*, 98(20), 2001.

C. K. I. Williams and D. Barber. Bayesian classification with Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI*, 20(12):1342–1351, 1998.

B. Zadrozny. Learning and evaluating classifiers under sample selection bias. In R. Greiner and D. Schuurmans, editors, *Proc. of the 21st Int. Conf. on Machine Learning (ICML)*, pages 114–122, 2004.

Notation and Symbols

Sets of Numbers

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

Data

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

Kernels

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

Vectors, Matrices and Norms

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

Functions

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

Probability

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

Graphs

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

SVM-related

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

Miscellaneous

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