Improving the ERM Algorithm Through Density-Ratio Estimation

Toros Arikan

1 Introduction

It is a common assumption in the supervised learning literature that the training and data samples are drawn using the same probability distribution [1], so that the expected performance on test samples converges to that on the training set with an increasing number of samples. This basic assumption, however, may not be fulfilled in practice for a variety of reasons. In a setting that requires extrapolation, we may not be able to take training and test samples from the same region of space. In a practical experimental design, we may choose to adjust our sampling set or procedure based on observations over training data. If these basic variations are present, we may no longer be able to assume the same probability distributions for the training and test samples without suffering performance penalties.

We assume a standard supervised learning framework. Given a domain of patterns $\mathcal{X}$ and labels $\mathcal{Y}$, we obtain training samples

$$Z_{\text{tr}} = \left\{ \left(x_1^{\text{tr}}, y_1^{\text{tr}}\right), \ldots, \left(x_{n_{\text{tr}}}^{\text{tr}}, y_{n_{\text{tr}}}^{\text{tr}}\right) \right\} \subseteq \mathcal{X} \times \mathcal{Y},$$

from a probability distribution $\mathrm{P}_{\text{tr}}(x, y)$, and test samples

$$Z_{\text{te}} = \left\{ \left(x_1^{\text{te}}, y_1^{\text{te}}\right), \ldots, \left(x_{n_{\text{te}}}^{\text{te}}, y_{n_{\text{te}}}^{\text{te}}\right) \right\} \subseteq \mathcal{X} \times \mathcal{Y},$$

from a probability distribution $\mathrm{P}_{\text{te}}(x, y)$. We cannot find a general solution for an estimation problem with two different distributions $\mathrm{P}_{\text{tr}}(x, y)$ and $\mathrm{P}_{\text{te}}(x, y)$, since the distributions can be arbitrarily far apart. However, we can make the simplifying assumption that the conditional distribution of outputs given inputs is common to the training and test samples, so that $\mathrm{P}_{\text{tr}}(x, y) = \mathrm{P}(y \mid x)\,\mathrm{P}_{\text{tr}}(x)$ and $\mathrm{P}_{\text{te}}(x, y) = \mathrm{P}(y \mid x)\,\mathrm{P}_{\text{te}}(x)$. This setting for density estimation is termed a covariate shift [2].

In this report, we investigate techniques for augmenting the ERM algorithm using the density-ratio between the training and test probability densities. This estimate is used as the measure of the importance of each training sample in the data domain. Basic learning algorithms can be modified by weighting the training loss function according to these importance values, so that highly useful properties such as consistency and asymptotic unbiasedness are achieved under a covariate shift.

The report is organized as follows. In Section 2, we review methods of nonparametric probability density estimation, highlighting how the Gaussian kernel is used to extend these methods in general higher-dimensional settings. Assuming that the density-ratio has been accurately deteremined, we present the importance-weighted ERM algorithm in Section 3, and conduct simulations of basic regression and classification examples to test this algorithm. We offer a conclusion in Section 4, observing that while performance improvements are easily obtained with accurate knowledge of the density-ratio, further research must be done to ensure robustness in general settings.

2 Nonparametric Probability Density Estimation

Prior to the advent of modern learning theory, the fundamental statistical approach to the learning problem was to estimates the underlying probability distribution from the training and data samples, and to use this estimate to obtain optimal decision boundaries for the given set of samples. However, probability distribution estimation is difficult in the settings of the majority of learning problems [1], and generally leads to highly erroneous results in higher dimensions, where an infeasible number of samples is required to obtain an accurate estimate. A fundamental principle of learning theory is to bypass this difficult problem, and to directly obtain the best classifiers for the dataset. However, if the training samples are not representative of the data distribution, we have to reconsider density estimation methods in a learning framework, in order to mitigate the underlying discrepancy between the training and test samples.

A common way to estimate probability densities is to assume we know the functional form of the distribution, and then to find the maximum likelihood estimate of the distribution's sufficient statistics. Given data vectors $\underline{x}$, this method essentially uses the mode of the posterior distribution $g(\theta \mid \underline{x})$ as an estimate for θ. However, this parametric approach has several important shortcomings [3]. The mode of a distribution tends to be much more unstable than the mean or the median, and although the maximum likelihood estimate is consistent and converges almost surely to the true parameter value θ^* under general assumptions [4], the result is inaccurate if the number of data points is small. We also make a strong assumption about the functional form of the probability density function, which may not hold in practice, especially in higher dimensions. We therefore consider nonparametric probability density estimation

methods in this section, avoiding any assumptions about the probability densities.

2.1 Kernel Estimators

The most common method used to estimate probability densities of unknown functional form are histograms [3], which form the basis of more sophisticated extensions. Suppose we have a random sample $\underline{x} = [x_1\, x_2\, \ldots\, x_n]$ from an unknown continuous probability density function on $[a,\, b] \in \mathbb{R}$, which we partition as $a = t_0 < t_1 < \cdots < t_m = b$. We consider estimators $\hat{f}_H\left(\underline{x}\right)$ of the form:

$$f_H\left(t\right) = c_i,\; t_i \leq t \leq t_{i+1},\; i = 0,\, \ldots\,,\, m-1$$

$$f_H\left(b\right) = c_{m-1}$$

$$f_H\left(t\right) = 0,\; t \notin \left[a,\, b\right], \tag{1}$$

where $f_H\left(t\right) \geq 0$ and $\int_a^b f_H\left(t\right) dt = 1$.

If q_i is the number of observations $x_j \in \underline{x}$ falling in the i^{th} interval, we obtain the standard histogram method for $\hat{f}_H$ by setting the weights c_i as:

$$\hat{c}_i = \frac{q_i}{n\left(t_{i+1} - t_i\right)},\; i = 0,\, \ldots\,,\, m-1 \tag{2}$$

The histogram estimator $\hat{f}_H\left(\underline{x}\right)$, as outlined in Equations (1) and (2), has the desirable properties of being a consistent estimator for the true probability density function $f^*\left(\underline{x}\right)$ [3]. However, this estimator is discontinuous and difficult to update, thus motivating various generalized kernel estimators that improve upon it in convergence and computational complexity. The Rosenblatt estimator [5] employs a shifted histogram for faster convergence to the true distribution, where the mesh that determines the bin intervals is adjusted so that each sample point in v lies at the midpoint of a bin. Given sample points $\{x_j\}_{j=1}^n$, the shifted histogram can be represented as:

$$\hat{f}_n\left(x\right) = \frac{1}{n}\sum_{j=1}^{n}\frac{1}{h_n}w\left(\frac{x - x_j}{h_n}\right), \tag{3}$$

where h_n is a real-valued constant for each n and $w\left(u\right)$ is defined as:

$$w\left(u\right) = \begin{cases} \frac{1}{2}, & \left|u\right| < 1 \\ 0, & \text{else} \end{cases} \tag{4}$$

The key generalization of the histogram method is to note that different kernels may be used in place of $w\left(u\right)$ as given in Equation 4, yielding the kernel estimator as follows [6]:

$$\begin{aligned} \hat{f}_n\left(x\right) &= \int_{-\infty}^{\infty}\frac{1}{h_n}K\left(\frac{x-y}{h_n}\right)dF_n\left(y\right) \\ &= \frac{1}{nh_n}\sum_{j=1}^{n}K\left(\frac{x - x_j}{h_n}\right), \end{aligned} \tag{5}$$

where the constraints on K are:

$$\begin{cases} \int_{-\infty}^{\infty}\left|K\left(y\right)\right|dy < \infty \\ \sup_{-\infty<y<\infty}\left|K\left(y\right)\right| < \infty \\ \lim_{y\to\infty}\left|yK\left(y\right)\right| = 0 \\ K\left(y\right) \geq 0 \\ \int_{-\infty}^{\infty}K\left(y\right)dy = 1 \end{cases} \tag{6}$$

The kernel estimator has the highly desirable properties of asymptotic unbiasedness and consistency, as proven in Theorem 1 [3]:

Theorem 1. *The kernel estimator is asymptotically unbiased if $h_n \to 0$ as $n \to \infty$, and is consistent if $nh_n \to \infty$ as $n \to \infty$.*

Proof: The asymptotic unbiasedness of the estimator follows from:

$$\begin{aligned} E\left[\hat{f}_n\left(x\right)\right] &= \frac{1}{h_n}\int K\left(\frac{x - x_j}{h_n}\right)f^*\left(x_j\right)dx_j \\ &= \int K\left(y\right)f^*\left(x - h_n y\right)dy \\ &= f^*\left(x\right) + O\left(h_n\right) \\ &\to f^*\left(x\right), \end{aligned} \tag{7}$$

where we have used the kernel estimator definition in Equation (5).

To prove consistency, we first note that:

$$\text{Var}\left[\hat{f}_n\left(x\right)\right] = \frac{1}{n}\text{Var}\left[\frac{1}{h_n}K\left(\frac{x-y}{h_n}\right)\right],$$

since $\text{Var}\left[\overline{z}\right] = \frac{1}{n}\text{Var}\left[z\right]$.

It follows that:

$$\begin{aligned} \frac{1}{n}\text{Var}\left[\frac{1}{h_n}K\left(\frac{x-y}{h_n}\right)\right] &\leq \frac{1}{n}E\left[\left(\frac{1}{h_n}K\left(\frac{x-y}{h_n}\right)^2\right)\right] \\ &= \frac{1}{nh_n}\left[\frac{1}{h_n}\int K^2\left(\frac{x-y}{h_n}\right)f^*\left(y\right)dy\right] \\ &\to 0 \text{ if } \lim_{n\to\infty} nh_n = \infty \end{aligned} \tag{8}$$

We also note that:

$$\begin{aligned} \text{MSE}\left[\hat{f}_n\left(x\right)\right] &= E\left[\left(\hat{f}_n\left(x\right) - f^*\left(x\right)\right)^2\right] \\ &= \text{Var}\left[\hat{f}_n\left(x\right)\right] + \text{Bias}^2\left(\hat{f}_n\left(x\right)\right) \end{aligned}$$

We have proven that both the variance and the bias terms go to zero. Therefore, $\text{MSE}\left[\hat{f}_n\left(x\right)\right] \to 0$, and $\hat{f}_n\left(x\right)$ is thus a consistent estimator for $f^*\left(x\right)$. □

One of the advantages of this estimator formulation is that it can be extended to arbitrarily many dimensions with an appropriate kernel. The Gaussian kernel is typically chosen, although a similar kernel with finite support can sometimes be easier to implement. One problem is that kernel estimators are not generally robust against poor choices of h_n, which is generally determined numerically for particular cases. The most important concern, however, is that the convergence rates of histograms are slower than $\frac{1}{n}$, and that poor estimates are obtained for high-dimensional problems.

2.2 Density-Ratio Estimation

We have previously mentioned that density estimation is a more difficult problem than classification, where we just learn the decision boundaries for a given dataset. In fact, the density-ratio is an easier quantity to estimate than the probability densities themselves, and can be directly applied to the learning problem. Various direct density-ratio estimation methods have been proposed [7]. Here, we consider the moment-matching method, which makes very few assumptions about the underlying distributions.

Given a probability distribution $p^*_{\text{te}}(\underline{x})$ for the data samples, and a probability distribution $p^*_{\text{tr}}(\underline{x})$ for the training samples, the density-ratio $r^*(\underline{x})$ is given by:

$$r^*(\underline{x}) = \frac{p^*_{\text{te}}(\underline{x})}{p^*_{\text{tr}}(\underline{x})} \tag{9}$$

Our goal is to find a good estimator $\hat{r}(\underline{x})$ for $r^*(\underline{x})$. A naive method would have been to estimate the probability distributions separately and to take the ratio directly. However, division by an estimated quantity often makes an estimator unreliable. This is a critical problem for a high-dimensional setting, because density values tend to be small in these cases, making the density-ratio estimation ill-conditioned.

We consider the derivation of the moment matching method of density-ratio estimation, where we use easily computed sample averages of the training and test data points to estimate the density-ratio at those points. Given a one-dimensional random variable X drawn from a probability distribution with density $p^*_X(x)$, the k^{th} order moment of X around the origin is defined as:

$$\mathrm{E}\left[X^k\right] = \int x^k p^*_X(x)\, dx \tag{10}$$

Noting that we can express Equation (9) as $r^*(\underline{x})\, p^*_{\text{tr}}(\underline{x}) = p^*_{\text{te}}(\underline{x})$, we observe that if we matched the first-order moments of the two sides of the equation through the following minimization step, we can obtain an approximation for the density-ratio:

$$\hat{r}(\underline{x}) = \underset{r}{\operatorname{argmin}} \left| \int \underline{x} r(\underline{x})\, p^*_{\text{tr}}(\underline{x})\, dx - \int \underline{x} p^*_{\text{te}}(\underline{x})\, dx \right|^2 \tag{11}$$

In practice, the expectations over $p^*_{\text{tr}}(\underline{x})$ and $p^*_{\text{te}}(\underline{x})$ in Equation (11) are replaced by the sample means. However, two distributions are equivalent if and only if all moments agree with each other. Directly matching infinitely many moments is not possible in reality, and although we can choose a finite number of moments and match them, consistency is not guaranteed with this approach. An alternative approach is to employ a universal reproducing kernel $K(\underline{x}, \underline{x}')$(Steinwart, 2001) as a nonlinear transformation. The Gaussian kernel:

$$K(\underline{x}, \underline{x}') = \exp\left(-\frac{(\underline{x} - \underline{x}')^2}{2\sigma^2}\right), \tag{12}$$

is an example of universal reproducing kernels. Using this kernel, mean matching leads to a consistent estimator (Huang et al., 2007). In particular, we solve the following minimization problem [8]:

$$\hat{r}(\underline{x}) = \underset{r \in \mathcal{R}}{\operatorname{argmin}} \left\| \int K(\underline{x}, \cdot)\, r(\underline{x})\, p^*_{\text{tr}}(\underline{x})\, dx - \int K(\underline{x}, \cdot)\, p^*_{\text{te}}(\underline{x})\, dx \right\|^2_{\mathcal{R}}, \tag{13}$$

where $\mathcal{R}$ denotes a universal reproducing kernel Hilbert space, and $\|\cdot\|_{\mathcal{R}}$ denotes its norm. The empirical form of Equation (13), using the test and training samples, is expressed as:

$$\hat{\underline{r}} = \underset{\underline{r} \in \mathbb{R}^{n_{\text{tr}}}}{\operatorname{argmin}} \left(\frac{1}{n^2_{\text{tr}}}\, \underline{r}^{\mathrm{T}} \underline{\underline{K}}_{\text{tr,tr}}\, \underline{r} - \frac{2}{n_{\text{tr}} n_{\text{te}}}\, \underline{r}^{\mathrm{T}} \underline{\underline{K}}_{\text{tr,te}}\, \underline{1}_{n_{\text{te}}} \right), \tag{14}$$

where $\underline{1}_{n_{\text{te}}}$ is a n_{te}-dimensional vector of all ones, and $\underline{\underline{K}}_{\text{tr,tr}}$ and $\underline{\underline{K}}_{\text{tr,te}}$ denote the kernel Gram matrices defined by:

$$\left[\underline{\underline{K}}_{\text{tr,tr}}\right]_{j,j'} = K\left(\underline{x}^{\text{tr}}_j, \underline{x}^{\text{tr}}_{j'}\right), \quad \left[\underline{\underline{K}}_{\text{tr,te}}\right]_{j,i} = K\left(\underline{x}^{\text{tr}}_j, \underline{x}^{\text{te}}_i\right) \tag{15}$$

The solution to Equation (14) can be analytically found as:

$$\hat{\underline{r}} = \frac{n_{\text{tr}}}{n_{\text{te}}}\, \underline{\underline{K}}^{-1}_{\text{tr,tr}}\, \underline{\underline{K}}_{\text{tr,te}}\, \underline{1}_{n_{\text{te}}} \tag{16}$$

Although the kernel method for density-ratio estimation leads to a consistent estimator, one of its weaknesses is that the parameter σ^2 in the Gaussian kernel of Equation (12) is selected empirically for a given dataset. We observe that the theoretical strengths and practical weaknesses of the kernel method are shared by the kernel estimator for the probability density functions themselves, as presented in Equation (5). Though this connection between the two kernel methods is not made explicit in the papers studied, the methodology that leads from the optimization problem to the kernel estimator is similar in both cases, and is inspired by the desirable properties of universal reproducing kernels for estimation.

We note that the kernel method outlined in this section is one of many parametric and nonparametric density ratio estimation methods. If there is a model for the probability density-ratio, then maximum likelihood estimation can be used to determine the sufficient statistics that characterize the density-ratio distribution. These direct density-ratio estimation methods still perform poorly when the dimensionality of the data domain is high. A framework of direct density-ratio estimation with dimensionality reduction was introduced in [9]. The basic idea is to find a subspace in which the numerator and denominator densities are significantly different, and then carry out density-ratio estimation only within this subspace. It is noted, however, that these techniques are still largely heuristic ones, and have to be finely tuned to the given problem setting.

3 Augmentation of the ERM Algorithm for Covariate Shift Adaptation

Having outlined several methods for density-ratio estimation, we now use the ratio estimate obtained for each training sample to augment the standard formulation of the ERM algorithm, stated as:

$$\hat{\theta}_{\text{ERM}} = \underset{\theta}{\operatorname{argmin}} \left[\frac{1}{n_{\text{tr}}} \sum_{i=1}^{n_{\text{tr}}} l\left(f\left(\underline{x}^{\text{tr}}_i, \theta\right), y^{\text{tr}}_i\right) \right], \tag{17}$$

where $l(\underline{x}, \underline{x}')$ is a loss function of our choosing. If a covariate shift is present, and if the model we are assuming is incorrectly specified, then $\hat{\theta}_{\text{ERM}}$ may not converge to the true parameter θ^* as $n_{\text{tr}} \to \infty$ [2]. A useful method for ensuring this convergence is importance weighting, in which we assign a larger weight to the training samples which are more representative of the data set. We make use of the following identity [10]:

$$\begin{aligned} \mathrm{E}_P[g(X)] &= \sum_{x \in X} P(x)\, g(x) \\ &= \sum_{x \in X} Q(x) \frac{P(x)}{Q(x)} g(x) \\ &= \mathrm{E}_Q\left[\frac{P(X)}{Q(X)} g(X)\right], \end{aligned} \tag{18}$$

where P and Q are probability distributions with the same support. When we apply the identity in Equation (18) to our learning framework, we obtain:

$$\mathrm{E}_{\underline{x}^{\text{te}} \sim p^*_{\text{te}}(\underline{x})}\left[g\left(\underline{x}^{\text{te}}\right)\right] = \mathrm{E}_{\underline{x}^{\text{tr}} \sim p^*_{\text{tr}}(\underline{x})}\left[g\left(\underline{x}^{\text{tr}}\right) \frac{p^*_{\text{te}}(\underline{x})}{p^*_{\text{tr}}(\underline{x})}\right] \tag{19}$$

We observe that, by weighting the expectation of $g(\underline{x})$ over the training samples with the correct probability density-ratio, we get a result that is equal to the expectation over the test samples, which we have yet to observe. We apply the importance weighting as stated in Equation (19) to the formulation of the ERM algorithm in Equation (17) to obtain an algorithm called importance-weighted ERM [11]:

$$\hat{\theta}_{\text{IWERM}} = \underset{\theta}{\operatorname{argmin}}\left[\frac{1}{n_{\text{tr}}} \sum_{i=1}^{n_{\text{tr}}} \left(\frac{p^*_{\text{te}}(\underline{x}_i^{\text{tr}})}{p^*_{\text{tr}}(\underline{x}_i^{\text{tr}})}\right)^{\gamma} l\left(f\left(\underline{x}_i^{\text{tr}}, \theta\right), y_i^{\text{tr}}\right)\right], \tag{20}$$

where $0 \leq \gamma \leq 1$ is a parameter that controls the weighting of the samples, with $\gamma = 0$ corresponding to the case of the standard ERM algorithm.

To observe how importance weighting helps mitigate a covariate shift, we consider two examples given in [11], where perfect knowledge of the density-ratio is assumed, and the training and test data distributions are Gaussians centered in different locations - thus making our problem similar to extrapolation. We first consider the case of a linear approximation to the true function $f^*(x) = \operatorname{sinc}(x)$ with mean-squared error, yielding the problem setting:

$$l(\hat{y}, y) = (\hat{y} - y)^2, \; f(x, \theta) = \theta_1 x + \theta_2$$

$$\Rightarrow \hat{\theta}_{\text{IWERM}} = \underset{\theta}{\operatorname{argmin}}\left[\frac{1}{n_{\text{tr}}} \sum_{i=1}^{n_{\text{tr}}} \left(\frac{p^*_{\text{te}}(x_i^{\text{tr}})}{p^*_{\text{tr}}(x_i^{\text{tr}})}\right)^{\gamma} \left(\theta_1 x_i^{\text{tr}} + \theta_2 - y_i^{\text{tr}}\right)^2\right] \tag{21}$$

The solution to this minimization problem can be found through weighted least-squares fitting. We perform simulations with $n_{\text{tr}} = n_{\text{te}} = 150$, using the training and test probability distributions given by:

$$p^*_{\text{tr}}(x) = N\left(x;\, 1,\, \left(\frac{1}{2}\right)^2\right), \; p^*_{\text{te}}(x) = N\left(x;\, 2,\, \left(\frac{1}{4}\right)^2\right) \tag{22}$$

The data and training distributions are centered around different regions, so that if we wish to get an accurate fit for the data, we need to extrapolate. The importance-weighting yields the correct extrapolation within a learning framework. We observe this in Figure 1, where we plot the training and data points to observe how importance-weighted ERM fits the data points much better than the standard least-squares solution that is equivalent to the ERM algorithm. The importance-weighting minimizes the contributions of the training points that are farther away from the test distribution, so that the linear fit is close to one on the data points that we care to classify well. We quantify this improvement in Table 1, from which we observe that the ERM and IWERM algorithms perform well on the training and test distributions, respectively.

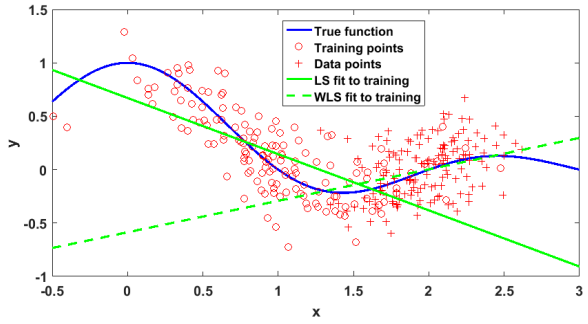

Fig. 1. Regression problem with weighted least squares solution.

TABLE 1
Mean-squared error for 100 regression trials.

Training, LS	Test, LS	Training, WLS	Test, WLS
0.085	0.249	0.474	0.057

For our second example, we consider a linear classification problem on noisy samples in $\mathbb{R}^2$, with 0-1 loss. For a general linear model, we have the estimator expressed as:

$$\hat{f}(\underline{x}; \theta) = \theta_0 + \sum_{i=1}^{d} \theta_i \underline{x}^{(i)}, \tag{23}$$

The analytical solution to the IWERM algorithm in Equation (20) with the linear estimator in Equation(23) is given by:

$$\hat{\theta}_{\text{AIWLS}} = \left(X^{\text{T}} D^{\gamma} X\right)^{-1} X^{\text{T}} D^{\gamma} \underline{y}, \tag{24}$$

where we have:

$$X \equiv \begin{bmatrix} 1 & \underline{x}_1^{\text{T}} \\ \vdots & \vdots \\ 1 & \underline{x}_n^{\text{T}} \end{bmatrix}, \tag{25}$$

$$D_{i,i} = \frac{p^*_{\text{te}}(\underline{x}_i)}{p^*_{\text{tr}}(\underline{x}_i)} \tag{26}$$

The classification result of a sample point z is given by the sign of the output of the learned function:

$$\hat{u} = \operatorname{sgn}\left(\hat{f}\left(z; \hat{\theta}_{\text{AIWLS}}\right)\right) \tag{27}$$

We consider a noisy sample setting where the conditional probabilities of the labels y given the sample points x are given by:

$$p\left(y=1 \mid \underline{x}\right)=\frac{1+\tanh\left(\underline{x}^{(1)}+\min\left(0,\, \underline{x}^{(2)}\right)\right)}{2}, \tag{28}$$

$$p\left(y=-1 \mid \underline{x}\right)=1-p\left(y=1 \mid \underline{x}\right) \tag{29}$$

On the $\underline{x}^{(2)}$ versus $\underline{x}^{(1)}$ plane in $\mathbb{R}^2$, the optimal decision-making boundary $p\left(y=1 \mid \underline{x}\right)=p\left(y=-1 \mid \underline{x}\right)$ for these distributions is $\underline{x}^{(2)}=-\underline{x}^{(1)}$ for $\underline{x}^{(1)}>0$, and $\underline{x}^{(1)}=0$ for $\underline{x}^{(2)}>0$. The training and test probability distributions are given by:

$$p_{\mathrm{tr}}^{*}\left(\underline{x}\right)=\frac{1}{2}N\left(\underline{x};\begin{bmatrix}-2\\3\end{bmatrix},\begin{bmatrix}1&0\\0&2\end{bmatrix}\right)+\frac{1}{2}N\left(\underline{x};\begin{bmatrix}2\\3\end{bmatrix},\begin{bmatrix}1&0\\0&2\end{bmatrix}\right),$$

$$p_{\mathrm{te}}^{*}\left(\underline{x}\right)=\frac{1}{2}N\left(\underline{x};\begin{bmatrix}0\\-1\end{bmatrix},\begin{bmatrix}1&0\\0&1\end{bmatrix}\right)+\frac{1}{2}N\left(\underline{x};\begin{bmatrix}4\\-1\end{bmatrix},\begin{bmatrix}1&0\\0&1\end{bmatrix}\right)$$

These distributions each consist of two Gaussian random variables, centered on opposite sides of the optimal decision-making boundary.

We conduct our simulations with $\gamma=0.5$, observing that $\gamma=1$ can yield an unstable estimator with a positive slope that misses the test distribution. This observed in the original paper [11], where a regularization parameter λ is introduced to make the IWERM algorithm more stable at the expense of introducing a second parameter to be optimized for a given setting. In Figure 2 and the corresponding Table 2, we observe that for $n_{\mathrm{tr}}=n_{\mathrm{te}}=250$, the ERM algorithm obtains a result close to the optimal boundary for the training set, while the IWERM algorithm is sufficiently biased to correctly classify the majority of data zeros. IWERM cannot approach the optimal boundary for the test sample set, because the probability distributions are not sufficiently close along the y-coordinate to obtain the correct y-intercept.

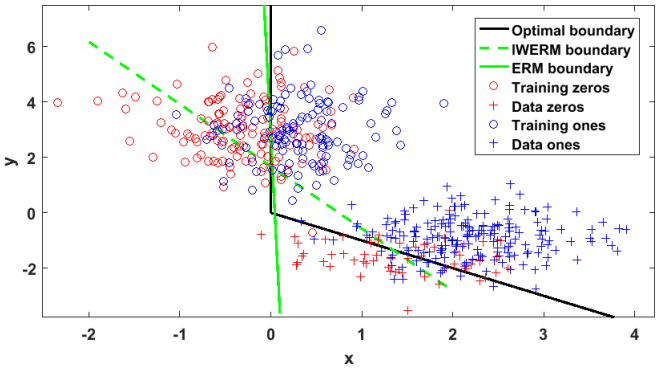

Fig. 2. Classification problem with weighted least squares solution, $n=250$.

A key issue with the IWERM algorithm is illustrated by repeating this classification experiment with $n_{\mathrm{tr}}=n_{\mathrm{te}}=500$, as illustrated in Figure 3 and the corresponding Table 3. In this case, the importance-weighting with the same parameters is not strong enough to move the IWERM boundary sufficiently far towards the data set - a stronger bias is required. As a result, the performance on test data is poorer compared to the previous case with 250 samples, contradicting the general assumption that an increase in the number of training samples would directly lead to better performance on test data. In the literature, these parameters are optimized by exhaustive searching for a given setting, which is feasible for the example analyzed here, but may not be the case for a larger and higher-dimensional dataset.

TABLE 2
Mean-squared error for 100 classification trials, $n=250$.

Tr, LS	Te, LS	Tr, WLS	Te, WLS	Tr, Best	Te, Best
0.268	0.365	0.384	0.172	0.267	0.159

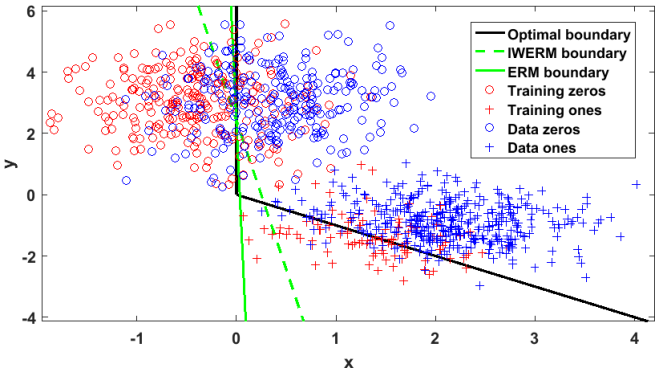

Fig. 3. Classification problem with weighted least squares solution, $n=500$.

TABLE 3
Mean-squared error for 100 classification trials, $n=500$.

Tr, LS	Te, LS	Tr, WLS	Te, WLS	Tr, Best	Te, Best
0.266	0.381	0.267	0.379	0.266	0.162

4 Conclusions

In this report, we questioned the common assumption that the training and test samples of a learning problem are drawn from the same probability distribution, and considered methods of augmenting the ERM algorithm to compensate for this setting. Researching well-established methods for density estimation, we noted that this task is generally harder to solve than the classical learning problem, so that most learning algorithms only make implicit use of probability densities. Density-ratio estimation, though still a difficult problem, is more general and therefore easier than direct density estimation. We reviewed the moment-matching method for density-ratio estimation, which is non-parametric except for kernel size selection and operates directly on the training and data samples. Assuming that the density-ratio has been perfectly estimated, we presented the IWERM algorithm that improved upon ERM under a covariate shift. We verified this claim by simulating simple regression and classification problems.

It is observed that while covariate shift adaptation leads to better performance on the test samples given an accurate density ratio, the method is not robust and does not

automatically lead to good results, even for the basic classification example we have provided. Most significantly, the performance is strongly dependent on the number of samples, the particular training and data distributions, and the parameters λ and γ introduced to tune the ERM algorithm. In a practical higher-dimensional setting where the density ratio is unknown and has to be estimated, we can expect performance and stability to decline further. Density-ratio estimation for machine learning applications clearly remains an open problem, and although performance improvement is demonstrated on practical datasets, more work needs to be done to provide robustness and generalizability. The literature on probability density estimation is vast and widely applicable, and the brief exposure presented in this report encourages research in this field from an electrical engineering standpoint.

REFERENCES

[1] V. Vapnik, *Statistical Learning Theory*. Wiley New York, 1998, vol. 1.

[2] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *Journal of Statistical Planning and Inference*, vol. 90, no. 2, pp. 227–244, 2000.

[3] R. A. Tapia and J. R. Thompson, *Nonparametric probability density estimation*. Johns Hopkins Univ., 1978.

[4] A. Wald, "Note on the consistency of the maximum likelihood estimate," *The Annals of Mathematical Statistics*, vol. 20, no. 4, pp. 595–601, 1949.

[5] M. Rosenblatt, "Remarks on some nonparametric estimates of a density function," *The Annals of Mathematical Statistics*, vol. 27, no. 3, pp. 832–837, 1956.

[6] E. Parzen, "On estimation of a probability density function and mode," *The Annals of Mathematical Statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

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

[8] A. Gretton, A. Smola, J. Huang, M. Schmittfulll, K. Borgwardt, and B. Schölkopf, "Covariate shift by kernel mean matching," *Dataset Shift in Machine Learning*, vol. 3, no. 4, p. 5, 2009.

[9] T. Kanamori, T. Suzuki, and M. Sugiyama, "Theoretical analysis of density ratio estimation," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 93, no. 4, pp. 787–798, 2010.

[10] C. P. Robert, *Monte Carlo Methods*. Wiley Online Library, 2004.

[11] M. Sugiyama, M. Krauledat, and K. Muller, "Covariate shift adaptation by importance weighted cross validation," *Journal of Machine Learning Research*, vol. 8, no. May, pp. 985–1005, 2007.