This article has been accepted for publication at the 2013 IEEE 13th International Conference on Data Mining Workshops

Auto-tuning Kernel Mean Matching

Yun-Qian Miao, Ahmed K. Farahat, Mohamed S. Kamel
University of Waterloo
Waterloo, Ontario, Canada. N2L 3G1
Email: {yqmiao, afarahat, mkamel}@uwaterloo.ca

***Abstract*—The Kernel Mean Matching (KMM) algorithm is a mathematically rigorous method that directly weights the training samples such that the mean discrepancy in a kernel space is minimized. However, the applicability of KMM is still limited, due to the existence of many parameters that are difficult to adjust. This paper presents a novel method that automatically tunes the KMM parameters by assessing the quality of distribution matching from a new perspective. While the KMM itself minimizes the mean discrepancy in a reproducing kernel Hilbert space, the tuning of KMM is achieved by adopting a different quality measure which reflects the Normalized Mean Squared Error (NMSE) between the estimated importance weights and the ratio of the estimated test and training densities. This method enables the applicability of KMM to real domains and leads to a generalized routine for the KMM to incorporate different types of kernels. The effectiveness of the proposed method is demonstrated by experiments on both synthetic and benchmark datasets.**

***Keywords*-Covariate Shift Adaptation, Density-ratio Estimation, Kernel Mean Matching.**

I. INTRODUCTION

Facing today's dynamic world, traditional learning techniques are required to demonstrate some degree of *adaptability* to cope with distribution changes. This has resulted in an intensive research in the data mining and machine learning community under the names domain adaptation [1], [2], transfer learning [3], and concept drift [4]. One particular case of domain adaptation is to handle the covariate shift problem [5], [6], where the concept is stable but the sample's marginal distributions between the training and working data are shifted.

Usually the covariate shift happens in biased sample selection scenarios. For example in building an action recognition system, the training samples are collected in a university lab setting, where young people make up a high percentage of the population. When the system is applied in reality, it is likely that we will face a more general population model.

Under the covariate shift setting, reweighting the training instances and applying cost-sensitive learning techniques can learn asymptotically optimal models to alleviate the biased sampling problem [5]. Given the probability density function of the training and test data as $p_{tr}(x)$ and $p_{ts}(x)$ respectively, it is justified that the best possible set of weights need to reflect the density-ratio, defined as:

$$\beta(x) = \frac{p_{ts}(x)}{p_{tr}(x)} \, . \tag{1}$$

The density-ratio estimation (also known as the sample importance estimation) seems not to involve any more burden than estimating two density functions and then dividing them. But this naïve approach encounters several problems [7]: 1) the information from the given limited number of samples may be sufficient to infer the density-ratio, but insufficient to infer two probability density functions; 2) a small estimation error in the denominator can lead to a large variance in the density-ratio; 3) the naïve approach would be highly unreliable for high-dimension problems because of the notable "curse-of-dimensionality".

Recently researchers made great strides in proposing methods to estimate the density-ratio directly. Kernel Mean Matching (KMM) [7] is a milestone in this trend, which minimizes the mean discrepancy in a Reproducing Kernel Hilbert Space (RKHS). The KMM algorithm shows elegance in theory, and is not specific to any distribution or density-ratio model. However, KMM lacks a systematic mechanism for tuning parameters. The heuristic choices, such as adopting the median of sample pairwise distances as Gaussian kernel width has neither strong theoretical justification nor has it been supported in practice [8]–[10]. Furthermore, there does not exist a clue on the choices for other types of kernels, such as the polynomial kernel. Yu et al. [11] analyzed the convergence rate of the KMM and revealed that the selection of kernel highly affects the performance of the KMM.

In order to choose a good kernel, one traditional approach is to use the weighted Cross-Validation (CV) applied on the subsequent learners. This approach, however, does not achieve good matching results. The reason behind this was explored in the previous work of Sugiyama et al. [12], who have shown that in learning covariate shift adaptation systems the model selection of importance estimation should be separated from the model selection of subsequent learners. If combining the two steps of model selection by the weighted CV based on the final learning system, the CV score would be estimated with bias inside the loop and accordingly the result is highly unreliable. Despite that the KMM algorithm involves minimizing an objective function called the Maximum Mean Discrepancy (MMD), the MMD

©2013 IEEE DOI 10.1109/ICDMW.2013.117

cannot serve as the criteria for tuning the KMM because they share the same parameter settings that define the kernel space, as discussed in details in Section III-C.

In this paper, we propose an auto-tuning algorithm for KMM by introducing a novel measure for assessing the quality of candidate choices. The proposed method enables the applicability of the mathematically rigorous KMM to real domains. The basic idea behind this work is to separate the process of estimating the importance weights from that of selecting parameters. This is achieved by allowing each of the two processes to see things from a different perspective. While the KMM optimizes the MMD measure, the proposed quality measure reflects the Normalized Mean Squared Error (NMSE) between the estimated importance weights and the ratio of the estimated test and training densities. Using the KMM for estimating the weights and then the NMSE to evaluate the choice of kernels and parameters allows us to combine the advantages of both methods. In specific, the new method uses the nonparametric KMM without implying any assumption on the model of the density-ratio, and at the same time it uses NMSE to provide a systematic process to tune the KMM automatically.

The effectiveness of the proposed auto-tuning KMM is investigated by conducting experiments on both synthetic data and benchmark datasets, comparing with baselines and the state-of-the-art methods. The results demonstrate the prominent contribution of our proposed method. The proposed tuning mechanism in fact leads to a generalized routine for the KMM to operate with different types of kernels or even with multi-kernel [13], where the weighting coefficients of different kernels need to be well tuned.

The remainder of the paper is organized as follows: The rest of this section describes the notations used in the paper. Section II discusses the importance estimation problem and reviews the key points of the KMM algorithm. Section III describes our approach of using NMSE as the objective criterion to form the KMM tuning mechanism. In Section IV, empirical evaluations are conducted on synthetic and real datasets. Section V concludes the paper.

A. Notation

In this paper, scales, vectors, and matrix are shown in small, bold, and capital letters respectively. When discussing covariate shift problem, we use the following notations:

$\mathcal{X}$	$\mathcal{X} \subseteq \mathbb{R}^d$, the d-dimension input space, $x \in \mathcal{X}$ is an input sample
$\mathcal{Y}$	the class label space, $y \in \mathcal{Y}$ is an output variable
p_{tr}	the probability density of the training data
p_{ts}	the probability density of the test data
n_{tr}	the number of training samples
n_{ts}	the number of test samples
β	the density-ratio to be estimated (Equation 1)
$\tilde{\beta}$	an estimate of β

II. Background

A. Learning under Covariate Shift

With the empirical risk minimization framework, the general purpose of a supervised learning problem is to minimize the expected risk of:

$$R\left[\theta, p, l\right] = \iint l\left(x, y, \theta\right) p\left(x, y\right) dxdy \,, \tag{2}$$

where θ is a learned model, $l\left(x, y, \theta\right)$ is a loss function for the problem with a joint distribution $p\left(x, y\right)$.

If we are facing the case where the training data distribution $p_{tr}(x, y)$ differs from the test data distribution $p_{ts}(x, y)$, in order to obtain the optimal model in the test domain θ^*_{ts}, we can derive the following reweighting scheme:

$$\begin{aligned}
\theta^*_{ts} &= \operatorname*{argmin}_{\theta \in \boldsymbol{\theta}} R_{ts}\left[\theta, p_{ts}\left(x, y\right), l\left(x, y, \theta\right)\right] \\
&= \operatorname*{argmin}_{\theta \in \boldsymbol{\theta}} R_{tr}\left[\theta, p_{tr}\left(x, y\right), \frac{p_{ts}\left(x, y\right)}{p_{tr}\left(x, y\right)} l\left(x, y, \theta\right)\right] \\
&\approx \operatorname*{argmin}_{\theta \in \boldsymbol{\theta}} \frac{1}{n_{tr}} \sum_{(x,y) \in \pi_{tr}} \frac{p_{ts}\left(x, y\right)}{p_{tr}\left(x, y\right)} l\left(x, y, \theta\right) \,.
\end{aligned} \tag{3}$$

Further, covariate shift assumes that the conditional distributions are the same across the training and test data (i.e. $p_{ts}\left(y|x\right) = p_{tr}\left(y|x\right)$), but that the marginal distributions are different. Hence θ^*_{ts} can be expressed as follows:

$$\begin{aligned}
\theta^*_{ts} &\approx \operatorname*{argmin}_{\theta \in \boldsymbol{\theta}} \frac{1}{n_{tr}} \sum_{(x,y) \in \pi_{tr}} \frac{p_{ts}\left(y|x\right) p_{ts}\left(x\right)}{p_{tr}\left(y|x\right) p_{tr}\left(x\right)} l\left(x, y, \theta\right) \\
&= \operatorname*{argmin}_{\theta \in \boldsymbol{\theta}} \frac{1}{n_{tr}} \sum_{(x,y) \in \pi_{tr}} \frac{p_{ts}\left(x\right)}{p_{tr}\left(x\right)} l\left(x, y, \theta\right) \\
&= \operatorname*{argmin}_{\theta \in \boldsymbol{\theta}} \frac{1}{n_{tr}} \sum_{(x,y) \in \pi_{tr}} \beta(x) l\left(x, y, \theta\right) \,.
\end{aligned} \tag{4}$$

Now, the learning system objective in the new test domain would be evaluated by the importance-weighted training samples to reflect the changes of distribution, where the sample importance is equal to its density-ratio.

Having the weighted training instances, there are numerous cost-sensitive learning algorithms that can be applied. Instead of minimizing the loss of misclassification, the cost-sensitive learning aims at minimizing the instance-dependent cost of wrong prediction [14], [15].

B. Kernel Mean Matching

To correct distribution difference caused by sampling bias, the Kernel Mean Matching (KMM) method is proposed to reweight the sample's importance such that the Maximum Mean Discrepancy (MMD) between the weighted training samples and the test samples is minimized [7]. The empirical KMM optimization is formulated as a convex quadratic

©2013 IEEE DOI 10.1109/ICDMW.2013.117

program to obtain the training sample's weights, $\tilde{\boldsymbol{\beta}}$:

$$\begin{aligned}\tilde{\boldsymbol{\beta}} &= \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left(\mathrm{MMD}^2\left[\mathcal{F}, \beta p_{tr}, p_{ts}\right]\right) \\ &\approx \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\| \frac{1}{n_{tr}} \sum_{x \in \pi_{tr}} \beta(x)\Phi(x) \right. \\ &\qquad \left. - \frac{1}{n_{ts}} \sum_{x \in \pi_{ts}} \Phi(x) \right\|^2 \\ &= \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left[\frac{1}{2}\boldsymbol{\beta}^{\boldsymbol{T}}\boldsymbol{K}\boldsymbol{\beta} - \boldsymbol{k}^{\boldsymbol{T}}\boldsymbol{\beta}\right] \qquad (5)\end{aligned}$$

$$\text{subject to: } \boldsymbol{\beta_i} \in [0, b] \text{ and } \mid \frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} \boldsymbol{\beta_i} - 1 \mid \leq \varepsilon \,,$$

where $\boldsymbol{\beta}$ is the weights over the training samples, $\mathcal{F}$ is a RKHS space with a mapping function $\Phi(x)$, the kernel matrix $\boldsymbol{K}$ is defined at the training samples as $K_{ij} = k(x_i, x_j)$, while the vector $\boldsymbol{k}$ is defined at the training samples as $\boldsymbol{k_i} := \frac{n_{tr}}{n_{ts}} \sum_{j=1}^{n_{ts}} k(x_i, x_j)$. The first asserted constraint is to limit the scope of the distribution changes, and the second constraint ensures that $\beta(x)p_{tr}$ is close to a Probability Density Function (PDF) with a precision ε.

III. Tuning KMM

A. Parameters in KMM

The KMM algorithm involves the following factors: the boundary b, the normalization precision ε, and most importantly the kernel parameters.

1) The boundary b: This reflects the discrepancy between the two distributions to be matched, and acts as a constraint that limits the range of the estimation to $\beta \in [0, b]$. Therefore, the parameter b is expected to be different according to the given problem. Setting $b = 1000$ is reasonable for most applications as suggested by Huang et al. [7].

2) The normalization precision ε: Referring to Lemma 3 in [7], the normalization constraint $\int \beta(x)p_{tr}(x)dx = 1$ is applied and the empirical estimate is used as $|\frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} \beta_i - 1| \leq \varepsilon$, where the parameter ε reflects the normalization precision. Huang et al. [7] explained that selection of ε should be $\mathcal{O}\left(\frac{b}{\sqrt{n_{tr}}}\right)$, and suggested KMM to adopt the setting as $\varepsilon = \left(\frac{\sqrt{n_{tr}}-1}{\sqrt{n_{tr}}}\right)$.

3) The kernel and kernel parameters: This is the most important set of parameters that affect the algorithm performance, but has not been well-studied in the literature.

The next section discusses the previous work on selecting the kernel parameters for KMM. In Section III-C, we propose a novel quality measure for conducting the kernel or parameter tuning.

B. Related Work

In the literature of parameter selection for kernelized methods, Gaussian kernels are commonly used and a popular heuristic for setting the Gaussian bandwidth is to use the median of the pairwise distances between samples [9], [16]. Another important work [10] of studying the covariate shift adaptation adopts the Gaussian bandwidth as $\sqrt{d/2}$, where d is the dimension of the data. These two settings are just heuristics that lack strong justification, and it has been shown in previous work [9] that using these settings in KMM does not produce very good matching results. For other types of kernels, to the best of our knowledge there is no report of prior work which studies a systematic methodology for tuning the parameters.

Gretton et al. [8] proposed to integrate the model selection of the KMM with the model selection of subsequent learning procedures using Cross-Validation (CV). Sugiyama et al. [17], however, pointed out that the CV score will be biased under the covariate shift conditions, and using importance-weighed CV will require the use of fixed importance weights. This means that these CV scores will be dependent on the estimated weights and accordingly cannot be used to decide which estimate is better. In summary, the model selection of importance estimation and the model selection of the classifier have to be evaluated separately and the objective of model selection for importance estimation is supposed to be based on the effectiveness of distribution matching.

Quite recently, Gretton et al. [13] and Sugiyama et al. [18] proposed methods to solve the parameter selection problems for the Maximum Mean Discrepancy (MMD) and Hilbert-Schmidt Independence Criterion (HSIC), respectively. In these papers, MMD and HSIC are used as statistical tests for determining the dependence of two sets of samples. This is totally different from the covariate shift problem we are trying to solve.

C. Tuning KMM Using NMSE

In this section, we are going to describe the proposed method for tuning KMM. Before we start, one direct question that might be asked is the following: If the KMM algorithm takes MMD as its objective function, then why can MMD not serve as the criteria for parameter selection? We first answer this question and support our answer with an illustrative example.

Referring to (5), MMD also has its own parameters on the kernel's choice to be determined, which shares the same parameter with KMM. For any defined kernels in calculating the MMD, the KMM which minimizes the MMD will fall into the same kernel space and lead to the same choice. To demonstrate this difficulty, we use this illustrative example. Suppose that KMM is used to match two one-dimension Gaussians, where the distributions differ at the means from 0 to 1, and the variances are the same ($\sigma^2 = 1$). Figure 1 shows the results of KMM and the corresponding MMD value with different kernel widths (X-axis). We can see that using different parameters for calculating MMD will lead the KMM to arrive at an optimum at different bandwidths too.

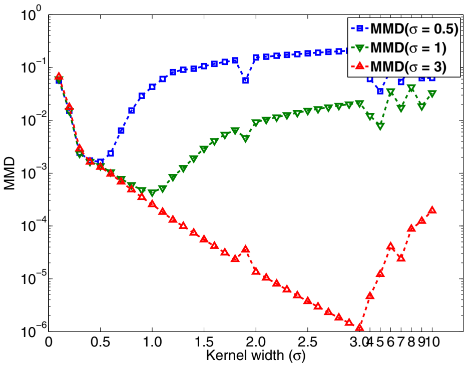


Figure 1: KMM with different kernel widths and the corresponding MMD values.

The shared parameter and hence the dependency between MMD and KMM imply that the MMD (i.e., the objective function of KMM) is not suitable to act as an objective criterion for KMM to process model selection.

Inspired by the work of Kanamori et al. [19], we propose to introduce the Normalized Mean Squared Error (NMSE) to assess the goodness of parameter settings. The key idea is that while each KMM procedure minimizes the mean discrepancy between the weighted training samples and the test samples, an overhead parameter selection procedure minimizes the NMSE, which is computed from the matching results. A NMSE-based quality measure for estimating the goodness of candidate parameter values can be derived as follows.

The NMSE between the ground-truth and approximate density ratios is defined as:

$$\text{NMSE} = \frac{1}{n_{tr}} \sum_{x \in \pi_{tr}} \left(\frac{\tilde{\beta}(x)}{\sum_{z \in \pi_{tr}} \tilde{\beta}(z)} - \frac{\beta(x)}{\sum_{z \in \pi_{tr}} \beta(z)} \right)^2 , \tag{6}$$

where $\beta(x)$ is the ground-truth density-ratio for a training sample x, $\tilde{\beta}(x)$ is the approximate density-ratio for x estimated using the KMM, π_{tr} is the set of training samples, and n_{tr} is the number of training samples.

Our goal is to define a criterion based on NMSE that can be used to assess the quality of a given parameter value, such that the best parameter value is the one that minimizes NMSE. Without loss of generality, we can assume that $\beta(x)$ and $\tilde{\beta}(x)$ are normalized over the training samples: $\sum_{x \in \pi_{tr}} \tilde{\beta}(x) = \sum_{x \in \pi_{tr}} \beta(x) = 1$. In this case, the NMSE can be calculated as:

$$\text{NMSE} = \frac{1}{n_{tr}} \sum_{x \in \pi_{tr}} \left(\tilde{\beta}(x) - \beta(x) \right)^2 . \tag{7}$$

Equation (7) is the empirical average corresponding to the following integral:

$$\begin{aligned} E[\text{NMSE}] &= \int \left(\tilde{\beta}(x) - \beta(x) \right)^2 p_{tr}(x)\, dx \\ &= \int \left(\tilde{\beta}^2(x) - 2\tilde{\beta}(x)\beta(x) \right) p_{tr}(x)\, dx \\ &\quad + \int \beta^2(x)\, p_{tr}(x) dx . \end{aligned} \tag{8}$$

Algorithm 1 Parameter Tuning for KMM

```
Input: $\pi_{tr}$, $\pi_{ts}$, $\mathcal{M}$ (a family of setups to be selected)
Output: chosen parameters $m^*$, and the estimates $\tilde{\boldsymbol{\beta}}^*$
for each $m$ in $\mathcal{M}$ do
    $\tilde{\boldsymbol{\beta}}_i^{(m)} \leftarrow \text{KMM}(\pi_{tr}, \pi_{ts}, m)$ for $x_i \in \pi_{tr}$;
    Estimate $\tilde{\boldsymbol{\beta}}_j^{(m)}$ for $x_j \in \pi_{ts}$ using RLS;
    $J(m) \leftarrow$ Equation$(11)$
end for
$m^* \leftarrow \text{argmin}_{m \in \mathcal{M}} J(m)$;
$\tilde{\boldsymbol{\beta}}^* \leftarrow \tilde{\boldsymbol{\beta}}^{(m*)}$.
```

The term $\int \beta^2(x)\, p_{tr}(x) dx$ does not depend on the density-ratio estimation method and accordingly the choice of the method parameters. This means that the parameter values that minimize NMSE will also lead to the minimization of a new score J, which can be defined as:

$$J = \int \left(\tilde{\beta}^2(x) - 2\tilde{\beta}(x)\beta(x) \right) p_{tr}(x)\, dx . \tag{9}$$

Substituting with $\beta(x) = p_{ts}(x)/p_{tr}(x)$ in (9), the J score can be simplified as follows:

$$J = \int \tilde{\beta}^2(x)\, p_{tr}(x)\, dx - 2 \int \tilde{\beta}(x)\, p_{ts}(x) dx . \tag{10}$$

Using the empirical averages corresponding to the integrals, the right-hand of (10) can be expressed as:

$$J = \frac{1}{n_{tr}} \sum_{x \in \pi_{tr}} \tilde{\beta}^2(x) - \frac{2}{n_{ts}} \sum_{x \in \pi_{ts}} \tilde{\beta}(x) . \tag{11}$$

The first section of the J score can use the estimated $\tilde{\beta}$ at the training samples given by KMM directly. The second section considers the $\tilde{\beta}$ scores at the test samples, which are not available. We formulate this scenario as a regression problem to model the $\tilde{\beta}$ and then deduce values at the test samples. In this paper, we use the Regularized Least Squares (RLS) [12] as the regression method.

At this point, we have all the necessary components to calculate the J score of (11). The parameter tuning procedure of KMM is conducted by minimizing J, as summarized in Algorithm 1. This mechanism gives the KMM the ability to be tuned based on evaluating the goodness of density-ratio estimation from a different perspective, which minimizes the NMSE as the objective.

It should be noted that using the estimation of $\tilde{\beta}$ at the training samples and extending the estimation to the test

©2013 IEEE DOI 10.1109/ICDMW.2013.117

Table I: Three cases of distribution shifting.

	p_{tr}	p_{ts}
Case-1	$\mathcal{N}(0, 1^2)$	$\mathcal{N}(1, 1^2)$
Case-2	$\mathcal{N}(0, (\frac{1}{2})^2)$	$\mathcal{N}(0, (\frac{1}{4})^2)$
Case-3	$\mathcal{N}(0, 1^2)$	$\mathcal{N}(1, (\frac{1}{2})^2)$

samples create another regression model with covariate shift problem. However, we observed that since the second term (an average over $\tilde{\beta}$) is relatively small compared to the first term (an average over $\tilde{\beta}^2$), and slight errors in estimating the second term due to covariate shift is not going to affect the values of the final quality measure. This is especially true when we have relatively reasonable number of training samples, referring to Theorem 4 in [11].

IV. Experiments

In this section, we first demonstrate the performance of our proposed method using three synthetic examples. Then, we compare our method with several state-of-the-art approaches over ten benchmark datasets. Lastly, we explore the usefulness of the method to polynomial kernels.

A. Illustrative Examples

To demonstrate the ability of the proposed parameter tuning mechanism, we use one-dimension Gaussians and examine three distribution drifting cases as shown below. The ground truth density-ratio $\boldsymbol{\beta}$ can be obtained using the known underlying distribution functions. Therefore we can calculate the quality of the importance estimates $\tilde{\boldsymbol{\beta}}$ by the NMSE [12], as defined in (6).

Table I lists three cases to be examined where the training and test distributions are drifting either by a shifting of means, a shifting of variances, or both. In all three cases, 200 training samples and 1000 test samples are randomly generated from the distributions as given. The KMM with Gaussian kernel is studied regarding different settings of kernel width.

The J scores calculated from the matching results and the NMSE calculated from ground-truth are plotted in Figure 2, corresponding to the three cases. From these results, some general facts can be observed:

1) The optimal parameter of KMM kernel width differs in different scenarios. The optimal parameters σ for the three cases are 10, 0.3 and 2.8.
2) This infers that a predefined value of the parameter may work in some cases, while failing in others. If we do not have strong prior knowledge on a given task, an automatic parameter tuning method is greatly needed.
3) The J scores calculated from (11) reflect the goodness of the KMM matching results to a great extent, and usually lead to a proper choice of the parameters, even though it may not be the most optimal.

Table II: Overview of datasets and the training test split.

Dataset	#Samples	#Features	n_{tr}	n_{ts}
ImageSeg	2310	18	716	770
BreastCancer	683	9	283	228
Diabetes	768	8	266	256
PenDigits(6vs8)	1498	16	530	499
USPS(6vs8)	1508	256	483	503
GermanCredit	1000	24	334	333
Cod_RNA	3000	8	1015	1000
Splice	3175	60	1025	1058
Australian	690	14	235	230
Adult_a1a	3000	123	1009	1000

B. Benchmark Datasets

Further experiments have been conducted on ten benchmark datasets, whose properties are summarized in Table II. These frequently used datasets are from the UCI[1] and LibSVM[2] archives.

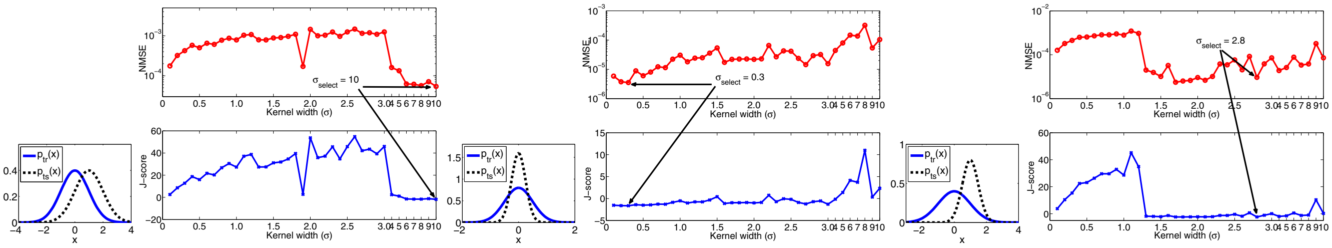
In our experiments, before any further process, all the data are normalized to the range $[-1, 1]^d$. The covariate shift classification tasks are formulated with the deliberately biased sampling procedures by following the work of [10]. First, one third of the data is uniformly sampled to form the test partition. Then, the rest of data is sub-sampled to form the the biased training set with probability $P(s = 1|x) = \frac{e^v}{1+e^v}$, where $P(s = 1)$ means that the sample x is included in the training set, and $v = \frac{4\boldsymbol{w}^{\boldsymbol{T}}(x-\overline{x})}{\sigma_{\boldsymbol{w}^{\boldsymbol{T}}(x-\overline{x})}}$. $\boldsymbol{w} \in \mathbb{R}^d$ is a projection vector randomly chosen from $[-1, 1]^d$. For each run, we randomly generate ten values of $\boldsymbol{w}$ and select the $\boldsymbol{w}$ which maximizes the difference between the unweighted method and the weighted method with ideal sampling weights. The typical reserved number training samples and the number of test samples are listed in Table II.

We set the baseline method as fitting a model on the training set without any modifications and predicting the test samples. The following state-of-the-art sample importance estimation methods are included as comparison:

- **KDE**: Using Kernel Density Estimator [20] to estimate the training PDF and test PDF separately, then dividing the two densities.
- **KLIEP**: The Kullback-Leibler Importance Estimation Procedure [17], which minimizes the Kullback-Leibler divergence.
- **uLSIF**: unconstrained Least Squares Importance Fitting [19], which models density-ratio as multi-Gaussians and minimizes LSIF. The above three methods have out-of-sample ability, and the parameters are chosen using the likelihood 10-fold Cross-Validation.
- **KMM(med)**: KMM algorithm with the kernel width being set as the median of pairwise distances of all training and test samples.

[1] UCI datasets: http://archive.ics.uci.edu/ml/datasets.html

[2] LibSVM datasets: http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/ For Cod_RNA and Adult_a1a, the first 3000 samples are taken.



(a) Case-1

(b) Case-2

(c) Case-3

Figure 2: The minimum of J scores and their corresponding kernel width.

- **KMM($\sqrt{d/2}$)**: KMM algorithm with the kernel width being set as $\sqrt{d/2}$, where d is the number of features of the data. This setup was used by Cortes et al. [10].
- **KMM(auto-tune)**: The proposed method with Gaussian kernels, using NMSE to tune the parameter of kernel width, scanning a range of kernel widths $([0.1 : 0.1 : 3, 4 : 1 : 10] * \sigma_{med}$, where σ_{med} is the median of sample's pairwise distances, and taking the choice with the minimal J score. In the above KMM methods, the other parameters are set to $\varepsilon = \left(\sqrt{n_{tr}} - 1\right) / \sqrt{n_{tr}}$, $b = 1000$.

After acquiring the instance-dependent weights using the above importance estimation methods, taking the same method as in previous work [12], we train the Instance-Weighted Regularized Least Squares Probabilistic Classifiers (IWRLS) and evaluate their prediction accuracy on the test sets respectively. In our experiments, each setup is repeated 30 times and the average performance measures are reported.

C. Results and Discussion

Similar to previous work [17], we use the Normalized Error (NE) to show the effectiveness of a method by considering the error of baseline unweighted method as one and calculate the metric as:

$$\text{NE}_{method} = \frac{\text{Err}_{method}}{\text{Err}_{baseline}} \times 100\% \tag{12}$$

Table III reports the Normalized Error on the ten datasets by using different importance estimation methods. It shows that the KMM equipped with the proposed auto-tuning mechanism outperforms other methods in mostly all the cases. As mentioned earlier, the KDE method is a two-step approach of importance estimation, which has an inherent likelihood Cross-Validation mechanism for parameter selection. In low dimensionality cases, it performs well. But, when encountering a high-dimensional problem, the weakness of this method is noticeable. On the other hand, the conventional heuristic setups of KMM, which uses the median of sample's distances or $\sqrt{d/2}$, do not have strong evidence of effectiveness. This is consistent with the findings of other reported results [8]–[10].

An interesting observation can be noted for the German Credit dataset. In this case, there is no improvement in classification performance when comparing all importance weighting methods with the simple unweighted approach. In such a scenario, the distribution changes are probably far away from the decision boundaries. And any reweighting strategy will not be effective in dealing with the shift.

D. Extension to Other Kernels

In this section, we extend the proposed auto-tuning method to another type of kernels, the polynomial kernels. We observed that when using the polynomial kernel with classification problems (in a setup similar to the one explored in Section IV-B), the classification accuracy does not change that much with different parameter values. This however does not reduce the usefulness of applying the auto-tuning method to polynomial kernels. This is because KMM is essentially a method for importance estimation which could have applicability in other machine learning tasks such as anomaly detection. Based on this observation, we use a different approach to evaluate how the auto-tuning method works with polynomial kernels. In specific, we use NMSE to evaluate the estimated density-ratio using different parameter values in comparison to using the auto-tuning method.

The following two commonly used polynomial kernels are to be investigated: 1) the polynomial kernel of degree 2: $k(x_i, x_j) = (x_i^T x_j + c)^2$; 2) the polynomial kernel of degree 3: $k(x_i, x_j) = (x_i^T x_j + c)^3$. The parameter c in these kernels is to be tuned using the proposed method.

Similar to the work of Sugiyama et al. [12], experiments are conducted based on the setup of Case-1 listed in Table I. We fixed the number of test samples as $n_{ts} = 1000$, and considered the following two scenarios:

1) Fix the number of training samples as $n_{tr} = 200$, and change the input dimension as $d = 1, 2, ..., 20$;
2) Fix the input dimension as $d = 10$, and increase the training sample size as $n_{tr} = 100 : 10 : 300$.

For each setting, the experiments are repeated 100 times. The matching quality is evaluated by the normalized mean square error (6). For the auto-tuning method, the c is automatically chosen from 0 to 2 with increments of 0.1.

Figures 3 and 4 show the average NMSE when using different parameter values for polynomial kernels of degree 2 and 3, respectively. From Figure 3a, we can find that

©2013 IEEE DOI 10.1109/ICDMW.2013.117

Table III: The normalized testing error of different importance estimation methods. For each dataset, the best performing group of methods (according to the Wilcoxon signed rank significance test at a confidence level of 95%) are highlighted in bold. The second-best method is underlined. The absolute error rate of the baseline is also reported in the first column.

Dataset	**Baseline**		**KDE**	**KLIEP**	**uLSIF**	**KMM**	**KMM**	**KMM**
	abs err	**norm err**				**(med)**	**($\sqrt{d/2}$)**	**(auto-tune)**
ImageSeg	0.1869	100.00	*59.03*	61.44	66.63	62.44	62.99	**55.70**
BreastCancer	0.3115	100.00	109.57	42.56	70.44	21.59	**21.21**	*21.49*
Diabetes	0.3469	100.00	98.24	95.46	*95.42*	98.39	99.36	**95.38**
PenDigits(6vs8)	0.0140	100.00	22.86	22.38	**21.90**	31.90	26.19	**21.43**
USPS(6vs8))	0.1262	100.00	81.73	*25.98*	36.90	30.50	30.34	**25.30**
GermanCredit	0.3160	**100.00**	106.37	103.99	*103.64*	105.16	107.92	104.91
Cod_RNA	0.3316	100.00	*85.81*	89.03	86.65	**86.16**	87.52	**83.77**
Splice	0.3792	100.00	125.23	90.95	113.93	85.30	**81.84**	*83.60*
Australian	0.2354	100.00	93.90	**72.60**	79.86	84.73	86.45	*79.68*
Adult_a1a	0.2640	100.00	118.71	*94.38*	100.37	**90.49**	96.94	96.72
Average		100.00	90.14	69.88	77.58	*69.67*	70.08	**66.80**

small values of the parameter c gives better matching scores for lower values of d. On the contrary, for high dimension cases it can be observed that large value of c tends to produce better matching results. The degree 3 kernel (Figure 4a) shows its own characteristics in response to dimension changes, but the effects of parameter c demonstrate the same trend. From these observations, we can conclude that using a fixed value for the c parameter is not going to give the best results for all dimensions, while using the auto-tuning method achieves the best matching results for all dimensions and outperforms the performance of the fixed values.

As observed from Figures 3b and 4b, it is not a surprise that in general the matching errors are shrinking as the training sample size increases. For the degree 2 kernel, the large value of c performs better when there is small number of training samples. On the other hand, small value of c is more suitable for large number of training samples. For the degree 3 kernel, the large value of c tends to perform well in different sizes of training samples. Similarly, we can observe that using auto-tuning method achieves the best matching scores for different sizes of training samples.

V. Conclusion

In this paper, we introduced an auto-tuning KMM method with a novel quality measure for evaluating the goodness of the distribution matching. This quality measure reflects the Normalized Mean Square Error (NMSE) between the estimated importance weights and the ratio of the estimated test and training densities. This measure accordingly introduces a new perspective for tuning KMM which is different from the Maximum Mean Discrepancy (MMD) used by KMM to estimate the importance weights. In addition, the proposed quality measure does not depend on the classifier and accordingly allows the model selection procedures for importance estimation and classifier learning to be completely separated. We demonstrated the superiority of applying the proposed technique to different types of kernels. The experimental comparison on synthetic data and benchmark datasets with the state-of-the-art approaches demonstrates the effectiveness of the proposed method.

References

[1] H. Daumé III and D. Marcu, “Domain adaptation for statistical classifiers,” *Journal of Artificial Intelligence Research*, vol. 26, no. 1, pp. 101–126, 2006.

[2] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, “A theory of learning from different domains,” *Machine Learning*, vol. 79, no. 1, pp. 151–175, 2010.

[3] S. J. Pan and Q. Yang, “A survey on transfer learning,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 22, no. 10, pp. 1345–1359, 2010.

[4] R. P. J. C. Bose, W. van der Aalst, I. Žliobaitė, and M. Pechenizkiy, “Handling concept drift in process mining,” in *Advanced Information Systems Engineering*. Springer, 2011, pp. 391–405.

[5] 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.

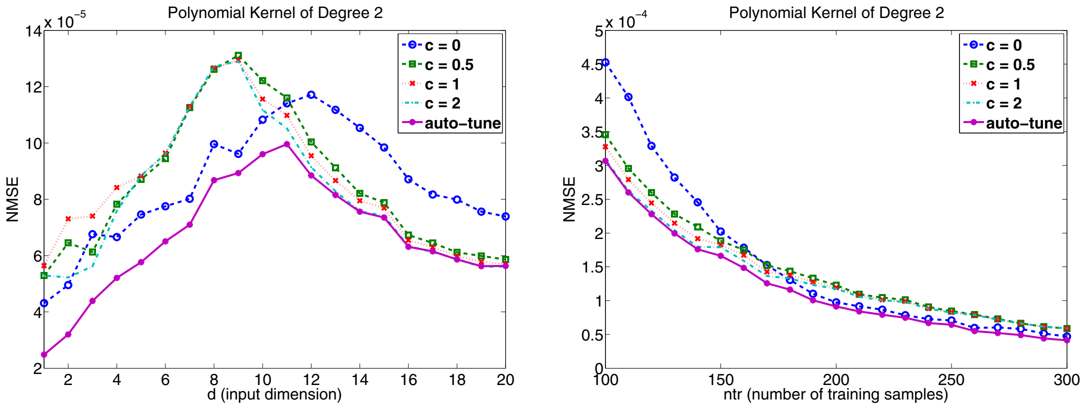
[6] J. G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. V. Chawla, and F. Herrera, “A unifying view on dataset shift in classification,” *Pattern Recognition*, vol. 45, no. 1, pp. 521–530, 2012.

[7] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf, “Correcting sample selection bias by unlabeled data,” in *Advances in Neural Information Processing Systems 19*, 2007, pp. 601–608.

[8] A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf, “Covariate shift by kernel mean matching,” *Dataset shift in machine learning*, pp. 131–160, 2009.

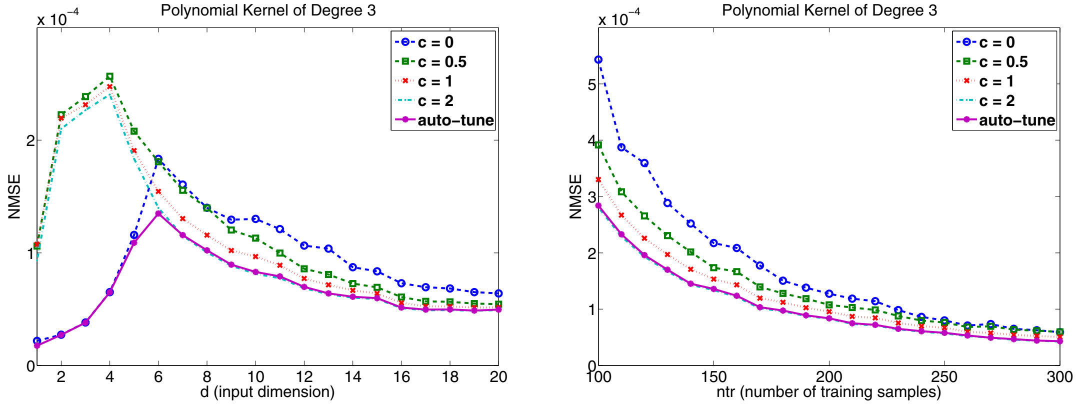
[9] M. Sugiyama, T. Kanamori, T. Suzuki, S. Hido, J. Sese, I. Takeuchi, and L. Wang, “A density-ratio framework for statistical data processing,” *IPSJ Transactions on Computer Vision and Applications*, vol. 1, no. 0, pp. 183–208, 2009.

©2013 IEEE DOI 10.1109/ICDMW.2013.117



(a) Changing the input dimension from 1 to 20 (b) Changing the training sample size from 100 to 300

Figure 3: The polynomial kernel of degree 2.



(a) Changing the input dimension from 1 to 20 (b) Changing the training sample size from 100 to 300

Figure 4: The polynomial kernel of degree 3.

[10] C. Cortes, M. Mohri, M. Riley, and A. Rostamizadeh, "Sample selection bias correction theory," in *Algorithmic Learning Theory*. Springer, 2008, pp. 38–53.

[11] Y. Yu and C. Szepesvári, "Analysis of kernel mean matching under covariate shift," in *The 29th International Conference on Machine Learning (ICML 2012)*, 2012, pp. 607–614.

[12] M. Sugiyama, M. Krauledat, and K.-R. Müller, "Covariate shift adaptation by importance weighted cross validation," *Machine Learning Research*, vol. 8, pp. 985–1005, 2007.

[13] A. Gretton, B. Sriperumbudur, D. Sejdinovic, H. Strathmann, S. Balakrishnan, M. Pontil, and K. Fukumizu, "Optimal kernel choice for large-scale two-sample tests," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1214–1222.

[14] B. Zadrozny, J. Langford, and N. Abe, "Cost-sensitive learning by cost-proportionate example weighting," in *Third IEEE International Conference on Data Mining (ICDM 2003)*, 2003, pp. 435–442.

[15] Y. Sun, M. S. Kamel, A. K. Wong, and Y. Wang, "Cost-sensitive boosting for classification of imbalanced data," *Pattern Recognition*, vol. 40, no. 12, pp. 3358–3378, 2007.

[16] B. Schölkopf and A. J. Smola, *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

[17] 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 *Advances in Neural Information Processing Systems 20*, 2008, pp. 1433–1440.

[18] M. Sugiyama and M. Yamada, "On kernel parameter selection in hilbert-schmidt independence criterion." *IEICE Transactions on Information and Systems,*, vol. E95-D, no. 10, pp. 1–9, 2012.

[19] T. Kanamori, S. Hido, and M. Sugiyama, "Efficient direct density ratio estimation for non-stationarity adaptation and outlier detection," *Advances in neural information processing systems*, vol. 21, pp. 809–816, 2008.

[20] G. R. Terrell and D. W. Scott, "Variable kernel density estimation," *The Annals of Statistics*, vol. 20, no. 3, pp. 1236–1265, 1992.

©2013 IEEE DOI 10.1109/ICDMW.2013.117