IEICE Transactions on Information and Systems,
vol.E92-D, no.10, pp.2159–2162, 2009.
Revised on December 31, 2010.

Direct Importance Estimation with Gaussian Mixture Models

Makoto Yamada and Masashi Sugiyama
Department of Computer Science, Tokyo Institute of Technology
2-12-2 O-okayama, Meguro-ku, Tokyo 152-8552, Japan
`yamada@sg.cs.titech.ac.jp` and `sugi@cs.titech.ac.jp`

Abstract

The ratio of two probability densities is called the importance and its estimation has gathered a great deal of attention these days since the importance can be used for various data processing purposes. In this paper, we propose a new importance estimation method using Gaussian mixture models (GMMs). Our method is an extension of the Kullback-Leibler importance estimation procedure (KLIEP), an importance estimation method using linear or kernel models. An advantage of GMMs is that covariance matrices can also be learned through an iterative estimation procedure, so the proposed method—which we call the Gaussian mixture KLIEP (GM-KLIEP)—is expected to work well when the true importance function has high correlation. Through experiments, we show the validity of the proposed approach.

Keywords

Importance weight, KLIEP, Gaussian mixture models

1 Introduction

Recently, the problem of estimating the ratio of two probability density functions (a.k.a. the *importance*) has received a great deal of attention since it can be used for various data processing purposes.

Covariate shift adaptation would be a typical example [1]. Covariate shift is a situation in supervised learning where the training and test input distributions are different while the conditional distribution of output remains unchanged [2]. In many real-world applications such as robot control [3], bioinformatics [4], spam filtering [5], natural language processing [6], brain-computer interfacing [7], and speaker identification [8], covariate shift adaptation has been shown to be useful. Covariate shift is also naturally induced in selective sampling or active learning scenarios and adaptation improves the generalization performance [9, 10, 11, 12].

Another example in which the importance is useful is outlier detection [13]. The outlier detection task addressed in that paper is to identify irregular samples (i.e., outliers) in an evaluation dataset based on a model dataset that only contains regular samples (i.e., inliers). If the density ratio of two datasets is considered, the importance values for regular samples are close to one, while those for outliers tend to be significantly deviated from one. Thus the values of the importance could be used as an index of the degree of outlyingness. A similar idea can also be applied to change detection in time series [14].

A naive approach to approximating the importance function is to estimate training and test probability densities separately and then take the ratio of the estimated densities. However, density estimation itself is a difficult problem and taking the ratio of estimated densities can magnify the estimation error. In order to avoid density estimation, the *Kullback-Leibler Importance Estimation Procedure* (KLIEP) was proposed [15]. KLIEP does not involve density estimation but directly models the importance function. The parameters in the importance model is learned so that the Kullback-Leibler divergence from the true test distribution to the estimated test distribution is minimized without going through density estimation. KLIEP was shown to be useful in covariate shift adaptation [15] and outlier detection [13]. A typical implementation of KLIEP employs a spherical Gaussian kernel model and the Gaussian width is chosen by cross validation. This means that when the true importance function is correlated, the performance of KLIEP could be degraded (see Figs.1-(b) and 1-(c)).

To cope with this problem, we propose to use a Gaussian mixture model in the KLIEP algorithm and learn the covariance matrix of the Gaussian components at the same time. This will allow us to learn the importance function more adaptively even when the true importance function contains high correlation (see Fig.1-(d)). We develop an iterative estimation procedure for learning the parameters in the Gaussian mixture model. The effectiveness of the proposed method—which we call the Gaussian mixture KLIEP (GM-KLIEP)—is shown through experiments.

The rest of this paper is structured as follows. In Section 2, the importance estimation problem is formulated and the original KLIEP method is reviewed. Then the proposed method, GM-KLIEP, is introduced in Section 3 and its experimental performance is investigated in Section 4. Finally, we conclude in Section 5 with a summary of our contributions.

2 Background

In this section, we formulate the importance estimation problem and briefly review the KLIEP method.

2.1 Problem Formulation

Let $\mathcal{D} \in \mathbb{R}^d$ be the data domain and suppose we are given i.i.d. training samples $\{\mathbf{x}_i^{tr}\}_{i=1}^{n_{tr}}$ from a training data distribution with density $p_{tr}(\mathbf{x})$ and i.i.d. test samples $\{\mathbf{x}_j^{te}\}_{j=1}^{n_{te}}$ from a test data distribution with density $p_{te}(\mathbf{x})$. We assume that $p_{tr}(\mathbf{x}) > 0$ for all $\mathbf{x} \in \mathcal{D}$. The goal of this paper is to develop a method of estimating the *importance* $w(\mathbf{x})$ from $\{\mathbf{x}_i^{tr}\}_{i=1}^{n_{tr}}$ and $\{\mathbf{x}_j^{te}\}_{j=1}^{n_{te}}$:

$$w(\mathbf{x}) = \frac{p_{te}(\mathbf{x})}{p_{tr}(\mathbf{x})}.$$

Our key restriction is that we avoid estimating densities $p_{te}(\mathbf{x})$ and $p_{tr}(\mathbf{x})$ when estimating the importance $w(\mathbf{x})$.

2.2 Kullback-Leibler Importance Estimation Procedure

Kullback-Leibler Importance Estimation Procedure (KLIEP) allows one to directly estimate $w(\mathbf{x})$ without going through density estimation [15]. In KLIEP, the following linear importance model is used:

$$\widehat{w}(\mathbf{x}) = \sum_{l=1}^{b} \alpha_l \varphi_l(\mathbf{x}), \tag{1}$$

where $\{\alpha_l\}_{l=1}^{b}$ are parameters, b is the number of parameters, and $\varphi_l(\mathbf{x})$ is a basis function. In the original KLIEP paper [15], the Gaussian kernel was chosen as the basis functions

$$\varphi_l(\mathbf{x}) = \exp\left(\frac{-\|\mathbf{x} - \mathbf{c}_l\|^2}{2\tau^2}\right),$$

where τ^2 is the Gaussian width and $\mathbf{c}_l$ is a template point randomly chosen from the test set $\{\mathbf{x}_i\}_{i=1}^{n_{te}}$. Using the model $\widehat{w}(\mathbf{x})$, one can estimate the test data density $p_{te}(\mathbf{x})$ as

$$\widehat{p}_{te}(\mathbf{x}) = \widehat{w}(\mathbf{x}) p_{tr}(\mathbf{x}).$$

Based on this, $\{\alpha_l\}_{l=1}^b$ is determined so that the Kullback-Leibler divergence from $p_{te}(\mathbf{x})$ to $\widehat{p}_{te}(\mathbf{x})$ minimized:

$$\begin{aligned} KL[p_{te}(\mathbf{x})\|\widehat{p}_{te}(\mathbf{x})] &= \int p_{te}(\mathbf{x}) \ln \frac{p_{te}(\mathbf{x})}{p_{tr}(\mathbf{x})\widehat{w}(\mathbf{x})} d\mathbf{x} \\ &= \int p_{te}(\mathbf{x}) \ln \frac{p_{te}(\mathbf{x})}{p_{tr}(\mathbf{x})} d\mathbf{x} - \int p_{te}(\mathbf{x}) \ln \widehat{w}(\mathbf{x}) d\mathbf{x}. \end{aligned}$$

The first term in the above equation is independent of $\{\alpha_l\}_{l=1}^b$, so it can be ignored. Let us define the second term as J:

$$J = \int p_{te}(\mathbf{x}) \ln \widehat{w}(\mathbf{x}) d\mathbf{x} \approx \frac{1}{n_{te}} \sum_{j=1}^{n_{te}} \ln \widehat{w}(\mathbf{x}_j^{te}),$$

where the expectation over the test distribution is approximated by the test sample average. Since $\widehat{p}_{te}(\mathbf{x})$ is a probability density, the following equation should hold:

$$1 = \int \widehat{p}_{te}(\mathbf{x}) d\mathbf{x} = \int p_{tr}(\mathbf{x}) \widehat{w}(\mathbf{x}) d\mathbf{x} \approx \frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} \widehat{w}(\mathbf{x}_i^{tr}),$$

where the expectation over the training distribution is approximated by the training sample average. Then the KLIEP optimization problem is given as follows:

$$\begin{aligned} &\max_{\{\alpha_l\}_{l=1}^b} \left[\sum_{j=1}^{n_{te}} \ln \left(\sum_{l=1}^{b} \alpha_l \varphi_l(\mathbf{x}_j^{te}) \right) \right] \\ &\text{s.t. } \sum_{i=1}^{n_{tr}} \sum_{l=1}^{b} \alpha_l \varphi_l(\mathbf{x}_i^{tr}) = n_{tr} \text{ and } \alpha_1, \ldots, \alpha_b \geq 0. \end{aligned}$$

2.3 Model Selection by Likelihood Cross Validation

The choice of the Gaussian width τ in KLIEP heavily affects the performance of importance estimation. Since KLIEP is based on the maximization of the score J, it is natural to determine τ so that J is maximized.

The expectation over $p_{te}(\mathbf{x})$ involved in J can be numerically approximated by *likelihood cross validation* (LCV) as follows [15]: First divide the test samples $\{\mathbf{x}_j^{te}\}_{j=1}^{n_{te}}$ into K disjoint subsets $\{\mathcal{X}_i^{te}\}_{i=1}^K$ of approximately the same size. Then obtain an importance estimate $\widehat{w}_k(\mathbf{x})$ from $\{\mathcal{X}_j^{te}\}_{j\neq k}$ (i.e., without $\mathcal{X}_k^{te}$) and approximate the score J using $\mathcal{X}_k^{te}$ as

$$\widehat{J}_k = \frac{1}{|\mathcal{X}_k^{te}|} \sum_{\mathbf{x}\in\mathcal{X}_k^{te}} \ln \widehat{w}_k(\mathbf{x}).$$

This procedure is repeated for $k = 1, \ldots, K$ and the average of $\widehat{J}_k$ over all k is used as an estimate of J:

$$\widehat{J} = \frac{1}{K} \sum_{k=1}^{K} \widehat{J}_k.$$

For model selection, $\widehat{J}$ is computed for all model candidates (the Gaussian width τ in the current setting) and choose the one that maximizes $\widehat{J}$.

3 KLIEP with Gaussian Mixture Models

In this section, we propose our new method, *the Gaussian mixture KLIEP* (GM-KLIEP).

Instead of the linear model (1), we use a Gaussian mixture model as an importance model:

$$w(\mathbf{x}) = \sum_{l=1}^{b} \pi_l N(\mathbf{x}|\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l),$$

where π_l are mixing coefficients, $N(\mathbf{x}|\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)$ is the Gaussian density with mean vector $\boldsymbol{\mu}_l$ and covariance matrix $\boldsymbol{\Sigma}_l$, and b is the number of mixture components. Then the GM-KLIEP optimization problem becomes

$$\max_{\{\pi_l, \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l\}_{l=1}^{b}} \left[\sum_{j=1}^{n_{te}} \ln \left(\sum_{l=1}^{b} \pi_l N(\mathbf{x}_j^{te}|\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l) \right) \right]$$

$$\text{s.t.} \sum_{i=1}^{n_{tr}} \sum_{l=1}^{b} \pi_l N(\mathbf{x}_i^{tr}|\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l) = n_{tr} \text{ and } \pi_1, \ldots, \pi_b \geq 0.$$

Here, we employ the following iterative estimation procedure for optimization (see Appendix for its derivation):

Initialization step: Initialize the means $\boldsymbol{\mu}_k$, the covariances $\boldsymbol{\Sigma}_k$, and the mixing coefficients π_k.

Step1: Evaluate the responsibility values γ_{kj} and β_{ki} using the current parameters:

$$\gamma_{kj} = \frac{\pi_k N(\mathbf{x}_j^{te}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{l=1}^{b} \pi_l N(\mathbf{x}_j^{te}|\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)},$$

$$\beta_{ki} = \frac{n_{te}}{n_{tr}} \pi_k N(\mathbf{x}_i^{tr}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k).$$

Step2: Re-estimate the parameters using the current responsibility values:

$$\begin{aligned}
\boldsymbol{\mu}_k^{new} &= (1-\eta)\boldsymbol{\mu}_k^{old} + \eta \frac{\sum_{j=1}^{n_{te}} \gamma_{kj}\mathbf{x}_j^{te} - \sum_{i=1}^{n_{tr}} \beta_{ki}\mathbf{x}_i^{tr}}{\sum_{j=1}^{n_{te}} \gamma_{kj} - \sum_{i=1}^{n_{tr}} \beta_{ki}}, \\
\boldsymbol{\Sigma}_k^{new} &= (1-\eta)\boldsymbol{\Sigma}_k^{old} + \eta \left(\frac{\sum_{j=1}^{n_{te}} \gamma_{kj}(\mathbf{x}_j^{te} - \boldsymbol{\mu}_k^{old})(\mathbf{x}_i^{te} - \boldsymbol{\mu}_k^{old})^\top}{\sum_{j=1}^{n_{te}} \gamma_{kj} - \sum_{i=1}^{n_{tr}} \beta_{ki}} \right. \\
&\qquad \left. - \frac{\sum_{i=1}^{n_{tr}} \beta_{ki}(\mathbf{x}_i^{tr} - \boldsymbol{\mu}_k^{old})(\mathbf{x}_i^{tr} - \boldsymbol{\mu}_k^{old})^\top}{\sum_{j=1}^{n_{te}} \gamma_{kj} - \sum_{i=1}^{n_{tr}} \beta_{ki}} \right) + \delta \mathrm{I}, \\
\pi_k^{new} &= \frac{n_{tr} \sum_{j=1}^{n_{te}} \gamma_{kj}}{n_{te} \sum_{i=1}^{n_{tr}} N(\mathbf{x}_i^{tr} | \boldsymbol{\mu}_k^{old}, \boldsymbol{\Sigma}_k^{old})},
\end{aligned}$$

where $0 < \eta \leq 1$ is a step parameter for stabilizing the algorithm, δ is the regularization parameter, and I is the identity matrix.

Evaluation step: Evaluate the log-likelihood:

$$\ln p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \pi) = \sum_{j=1}^{n_{te}} \ln \left(\sum_{l=1}^{b} \pi_l^{new} N(\mathbf{x}_j^{te} | \boldsymbol{\mu}_l^{new}, \boldsymbol{\Sigma}_l^{new}) \right).$$

Repeat the Step1 and Step2 until the log-likelihood converges.

Practically, we may use the k-means clustering algorithm for parameter initialization [16] and LCV is used for tuning the number of mixtures b.

4 Experiments

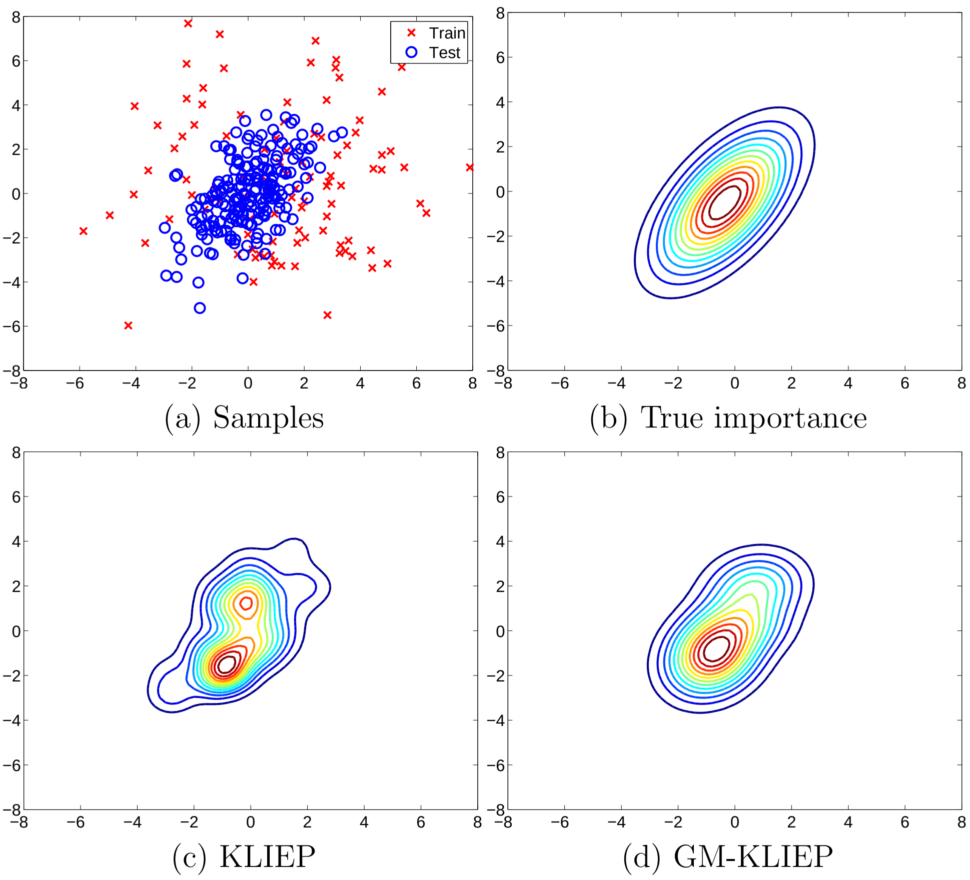
In this section, we compare the performance of GM-KLIEP with the original KLIEP. In these experiments, we set $\eta = 0.1$ and $\delta = 10^{-10}$ and choose the number of mixtures by 5-fold LCV from

$$b \in \{1, 2, 3, 4, 5\}.$$

4.1 Illustrative Example

Let us consider a toy two-dimensional importance estimation problem, where the true training and test density functions are defined as

$$\begin{aligned}
p_{tr}(\mathbf{x}) &= N\left(\mathbf{x} \,\middle|\, \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}\right), \\
p_{te}(\mathbf{x}) &= N\left(\mathbf{x} \,\middle|\, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1.5 & 1 \\ 1 & 2.5 \end{bmatrix}\right).
\end{aligned}$$



(a) Samples (b) True importance

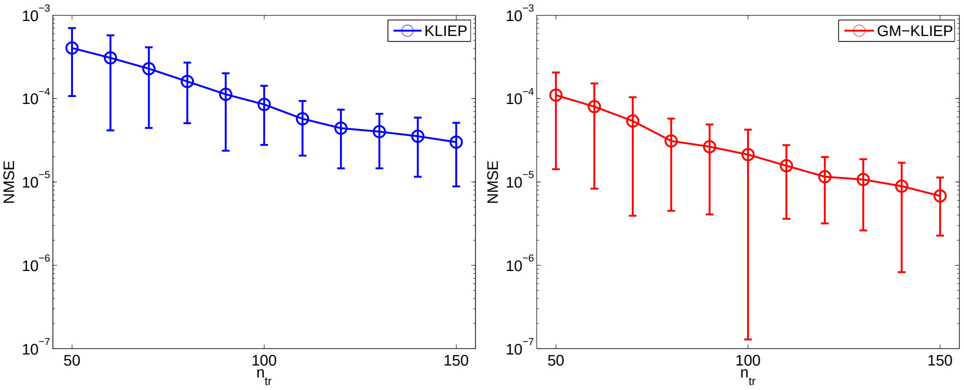
(c) KLIEP (d) GM-KLIEP

Figure 1: Samples and contour plots of the true importance function, the estimated importance function by KLIEP, and an estimated importance function by GM-KLIEP in the illustrative example.

In KLIEP, we set $b = 100$ and use the spherical Gaussian kernel as the basis function; the kernel width is chosen based on 5-fold LCV. In GM-KLIEP, we use the k-means clustering algorithm for parameter initialization [16], and choose the number of mixtures.

We draw $n_{tr} = 100$ training samples and $n_{te} = 1000$ test samples following the above densities, which are depicted in Fig.1-(a). Figures 1-(b), 1-(c), and 1-(d) are the contour plots of the true importance function, the importance function estimated by KLIEP, and an importance function estimated by GM-KLIEP, respectively. The results show that GM-KLIEP can capture the correlated profile of the true importance function better than the original KLIEP. The result of KLIEP seems to be rather overfitted due to high flexibility of the kernel model.

Next, we vary the number of training samples as $n_{tr} = 50, 60, \ldots, 150$ and quantitatively compare the performance of KLIEP and GM-KLIEP. We run the experiments 100 times for each n_{tr}, and evaluate the quality of an importance estimate $\widehat{w}(\mathbf{x})$ by the



(a) KLIEP

(b) GM-KLIEP

Figure 2: NMSEs averaged over 100 trials (log scale) in the illustrative examples.

normalized mean squared error (NMSE) [15]:

$$\mathrm{NMSE} = \frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} \left(w(\mathbf{x}_i^{tr}) - \widehat{w}(\mathbf{x}_i^{tr}) \right)^2 ,$$

where $\Sigma_{i=1}^{n_{tr}} \widehat{w}(\mathbf{x}_i^{tr})$ and $\Sigma_{i=1}^{n_{tr}} w(\mathbf{x}_i^{tr})$ are normalized to be one, respectively.

NMSEs averaged over 100 trials are plotted in Figs.2-(a) and 2-(b), showing that the errors of both methods tend to decrease as the number of training samples grows. GM-KLIEP tends to outperform the plain KLIEP, especially when the number of training samples is small; indeed, GM-KLIEP is shown to be significantly better than KLIEP by the *t-test* at the significance level 5%.

4.2 Application to Inlier-based Outlier Detection

Finally, we compare the performance of the original KLIEP with the proposed GM-KLIEP in inlier-based outlier detection.

The datasets provided by IDA [17] are used for performance evaluation; we exclude the 'splice' dataset since it is discrete. The datasets are binary classification and each one consists of positive/negative and training/test samples. We use all positive test samples as inliers and the first 5% of negative test samples as outliers in the "evaluation" set; we use positive training samples as inliers in the "model" set. Thus, the positive samples are treated as inliers and the negative samples are treated as outliers. We assign the evaluation set to $p_{tr}(\mathbf{x})$ and the model set to $p_{te}(\mathbf{x})$. Thus if the importance value is small, the sample is more plausible to be an outlier.

In the evaluation of outlier detection performance, it is important to take into account both the *detection rate* (the amount of true outliers an outlier detection algorithm can find) and the *detection accuracy* (the amount of true inliers that an outlier detection algorithm misjudges as outliers). Since there is a trade-off between the detection rate and

Table 1: Mean AUC values (and their standard deviation in the bracket) over 20 trials in the outlier detection experiments. If the performance of the two methods are significantly different by the *t-test* at the significance level 5%, the better method is indicated by bold face.

Datasets	KLIEP	GM-KLIEP
banana	53.5 (3.5)	**76.3** (12.4)
brestcancer	71.1 (12.3)	62.7 (18.9)
diabetes	58.8 (9.5)	55.0 (6.6)
flaresolar	50.6 (7.9)	**66.0** (18.6)
german	59.4 (8.1)	56.1 (8.7)
heart	71.6 (14.2)	70.7 (15.0)
image	60.6 (9.6)	**80.5** (8.0)
thyroid	65.6 (13.2)	72.6 (14.6)
titanic	66.0 (5.0)	66.9 (15.5)
twonorm	**92.3** (3.2)	89.6 (2.2)
waveform	87.5 (3.5)	88.1 (2.8)
Average	67.0 —	**71.3** —

the detection accuracy, we decided to adopt the *area under the ROC curve* (AUC) as our error metric.

The results are summarized in Tab.1, showing that GM-KLIEP compares favorably with the plain KLIEP.

5 Conclusions

In this paper, we proposed a new importance estimation method using Gaussian mixture models. Optimization of the proposed algorithm, GM-KLIEP, can be efficiently carried out through the iterative estimation procedure. The usefulness of the proposed approach was illustrated through experiments.

Acknowledgements

The authors would like to thank Taketo Akama for pointing out an error in an earlier manuscript. MS thanks MEXT Grant-in-Aid for Young Scientists (A), 20680007, SCAT, and AOARD for financial support.

Appendix

Here, we show the derivation of the iterative estimation procedure for GM-KLIEP given in Section 3.

The Lagrangian of the GM-KLIEP optimization problem (2) is given by

$$J(\boldsymbol{\pi}, \mathrm{M}, \boldsymbol{\Sigma}) = \sum_{j=1}^{n_{te}} \ln \left(\sum_{l=1}^{b} \pi_l N(\mathbf{x}_j^{te} | \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l) \right) + \lambda \left(\sum_{i=1}^{n_{tr}} \sum_{l=1}^{b} \pi_l N(\mathbf{x}_i^{tr} | \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l) - n_{tr} \right), \qquad (2)$$

where λ is the Lagrange multiplier.

Differentiating this with respect to π_k and equating it to zero, we have

$$\frac{\partial J(\boldsymbol{\pi}, \mathrm{M}, \boldsymbol{\Sigma})}{\partial \pi_k} = \sum_{j=1}^{n_{te}} \frac{N(\mathbf{x}_j^{te} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{l=1}^{b} \pi_l N(\mathbf{x}_j^{te} | \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)} + \lambda \sum_{i=1}^{n_{tr}} N(\mathbf{x}_i^{tr} | \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l) = 0.$$

Let us multiply this by π_k:

$$\lambda \sum_{i=1}^{n_{tr}} \pi_k N(\mathbf{x}_i^{tr} | \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l) = - \sum_{j=1}^{n_{te}} \gamma_{kj}, \qquad (3)$$

where

$$\gamma_{kj} = \frac{\pi_k N(\mathbf{x}_j^{te} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{l=1}^{b} \pi_l N(\mathbf{x}_j^{te} | \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}.$$

Summing this up for all $k = 1, \ldots, b$ and solving this with respect to λ, we have

$$\lambda = -\frac{n_{te}}{n_{tr}}.$$

Inserting this back into Eq.(3), we obtain

$$\pi_k = \frac{n_{tr} \sum_{j=1}^{n_{te}} \gamma_{kj}}{n_{te} \sum_{i=1}^{n_{tr}} N(\mathbf{x}_i^{tr} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}.$$

This gives the update equation for π_k.

Differentiating Eq.(2) with respect to $\boldsymbol{\mu}_k$, we have

$$\frac{\partial J(\boldsymbol{\pi}, \mathrm{M}, \boldsymbol{\Sigma})}{\partial \boldsymbol{\mu}_k} = \sum_{j=1}^{n_{te}} \gamma_{kj} \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_j^{te} - \boldsymbol{\mu}_k) - \sum_{i=1}^{n_{tr}} \beta_{ki} \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_i^{tr} - \boldsymbol{\mu}_k),$$

where

$$\beta_{ki} = \frac{n_{te}}{n_{tr}} \pi_k N(\mathbf{x}_i^{tr} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k).$$

Equating this to zero and solving this with respect to $\boldsymbol{\mu}_k$, we have

$$\boldsymbol{\mu}_k \;\; = \;\; \frac{\sum_{j=1}^{n_{te}} \gamma_{kj} \mathbf{x}_j^{te} - \sum_{i=1}^{n_{tr}} \beta_{ki} \mathbf{x}_i^{tr}}{\sum_{j=1}^{n_{te}} \gamma_{kj} - \sum_{i=1}^{n_{tr}} \beta_{ki}}.$$

This gives the update equation for $\boldsymbol{\mu}_k$.

Finally, differentiating Eq.(2) with respect to $\boldsymbol{\Sigma}_k$, we have

$$\begin{aligned}\frac{\partial J(\boldsymbol{\pi}, \mathrm{M}, \boldsymbol{\Sigma})}{\partial \boldsymbol{\Sigma}_k} &= \sum_{j=1}^{n_{te}} \gamma_{kj} \left(-\frac{1}{2}(\boldsymbol{\Sigma}_k^{-1} - \boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_j^{te} - \boldsymbol{\mu}_k)(\mathbf{x}_j^{te} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}) \right) \\ &\quad - \sum_{i=1}^{n_{tr}} \beta_{ki} \left(-\frac{1}{2}(\boldsymbol{\Sigma}_k^{-1} - \boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_i^{tr} - \boldsymbol{\mu}_k)(\mathbf{x}_i^{tr} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}) \right).\end{aligned}$$

Equating this to zero and solving this with respect to $\boldsymbol{\Sigma}_k$, we have

$$\boldsymbol{\Sigma}_k = \frac{\sum_{j=1}^{n_{te}} \gamma_{kj}(\mathbf{x}_j^{te} - \boldsymbol{\mu}_k)(\mathbf{x}_i^{te} - \boldsymbol{\mu}_k)^\top}{\sum_{j=1}^{n_{te}} \gamma_{kj} - \sum_{i=1}^{n_{tr}} \beta_{ki}} - \frac{\sum_{i=1}^{n_{tr}} \beta_{ki}(\mathbf{x}_i^{tr} - \boldsymbol{\mu}_k)(\mathbf{x}_i^{tr} - \boldsymbol{\mu}_k)^\top}{\sum_{j=1}^{n_{te}} \gamma_{kj} - \sum_{i=1}^{n_{tr}} \beta_{ki}}.$$

This gives the update equation for $\boldsymbol{\Sigma}_k$.

References

[1] J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset Shift in Machine Learning.* MIT Press, Cambridge, MA, 2008.

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

[3] H. Hachiya, T. Akiyama, M. Sugiyama, and J. Peters. Adaptive importance sampling for value function approximation in off-policy reinforcement learning. *Neural Networks*, 22(10):1399–1410, 2009.

[4] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola. Interesting structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57, 2006.

[5] S. Bickel and T. Scheffer. Dirichlet-enhanced spam filtering based on biased samples. In *Advances in Neural Information Processing Systems*, pages 161–168, Cambridge, MA, 2007. MIT Press.

[6] Y. Tsuboi, H. Kashima, S. Hido, S. Bickel, and M. Sugiyama. Direct density ratio estimation for large-scale covariate shift adaptation. *IPSJ Journal*, 50(4):1–19, 2009.

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

[8] M. Yamada, M. Sugiyama, and T. Matsui. Covariate shift adaptation for semi-supervised speaker identification. In *Proceedings of 2009 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP2009)*, pages 1661–1664, Taipei, Taiwan, Apr. 19–24 2009.

[9] D. P. Wiens. Robust weights and designs for biased regression models: Least squares and generalized M-estimation. *Journal of Statistical Planning and Inference*, 83(2):395–412, 2000.

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

[11] M. Sugiyama. Active learning in approximately linear regression based on conditional expectation of generalization error. *Journal of Machine Learning Research*, 7:141–166, 2006.

[12] M. Sugiyama and S. Nakajima. Pool-based active learning in approximate linear regression. *Machine Learning*, 78(1–2):35–61, 2010.

[13] S. Hido, Y. Tsuboi, H. Kashima, M. Sugiyama, and T. Kanamori. Inlier-based outlier detection via direct density ratio estimation. In *Proceedings of IEEE International Conference on Data Mining (ICDM2008)*, pages 223–232, Pisa, Italy, Dec. 15–19 2008.

[14] Y. Kawahara and M. Sugiyama. Change-point detection in time-series data by direct density-ratio estimation. In *Proceedings of 2009 SIAM International Conference on Data Mining (SDM2009)*, pages 389–400, Sparks, Nevada, USA, Apr. 30–May 2 2009.

[15] M. Sugiyama, T. Suzuki, S. Nakajima, H. Kashima, P. von Bünau, and M. Kawanabe. Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60(4):699–746, 2008.

[16] C. M. Bishop. *Pattern recognition and machine learning.* Springer-Verlag, New York, 2006.

[17] G. Rätsch, T. Onoda, and K.-R. Müller. Soft margins for adaboost. *Machine Learning*, 42:287–320, 2001.