Ensemble 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) is an elegant algorithm that produces density ratios between training and test data by minimizing their maximum mean discrepancy in a kernel space. The applicability of KMM to large-scale problems is however hindered by the quadratic complexity of calculating and storing the kernel matrices over training and test data. To address this problem, this paper proposes a novel ensemble algorithm for KMM, which divides test samples into smaller partitions, estimates a density ratio for each partition and then fuses these local estimates with a weighted sum. Our theoretical analysis shows that the ensemble KMM has a lower error bound than the centralized KMM, which uses all the test data at once to estimate the density ratio. Considering its suitability for distributed implementation, the proposed algorithm is also favorable in terms of time and space complexities. Experiments on benchmark datasets confirm the superiority of the proposed algorithm in terms of estimation accuracy and running time.**

***Keywords*—Density ratio estimation; Kernel mean matching; Ensemble method; Distributed algorithm.**

I. Introduction

Estimating the ratio of probability densities from two data collections is an emerging topic in the statistics and machine learning communities. The topic has attracted a great interest due to its potential in solving many challenging problems in data mining, such as covariate shift adaptation [1], [2], [3] , outlier detection [4], [5], semi-supervised learning [6], [7], and some others [8], [9].

This problem is referred to as density-ratio estimation [10], [11]. It is also known as the sample importance estimation problem [12] and the Radon-Nikodym Derivative problem [13]. A simple solution to this problem would be estimating two density functions and then dividing them. This naïve approach however encounters several problems [14]: 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 - the estimation of probability density is usually a more general and challenging problem; 2) a small estimation error in the denominator can lead to a large variance in the density-ratio; 3) the naïve approach is highly unreliable for high-dimensional data because of the *curse-of-dimensionality* problem.

Recently, a number of kernel methods have been proposed to estimate the density ratio directly in one-step. Kernel Mean Matching (KMM) [14] is a milestone in this trend, which minimizes the mean discrepancy in a Reproducing Kernel Hilbert Space (RKHS). The KMM algorithm has elegant property in convergence and is not specific to any distribution or density ratio model [1], [15]. However, the currently available algorithms for direct density-ratio estimation, including KMM, are centralized and their scalability is a challenging problem. This greatly limits the applicability of KMM algorithms for large scale data analysis. In specific, KMM formulates a quadratic optimization programming problem which operates on the Gram kernel matrix of all training samples and the connection kernel matrix between all training and test samples. KMM computational and space complexities are thus the functions of the size of training and test data.

To address this limitation, we propose an ensemble Kernel Mean Matching (ensKMM) algorithm, which is based on dividing test samples into smaller partitions and fusing these local estimates with a weighted sum. These weights of components are proportional to the sample size of each collection. The greatest advantage of the proposed ensemble approach is its suitability for distributed implementation, which makes it less constrained by the size of test data. While the proposed ensemble approach makes density-ratio methods scalable, our theoretical analysis reveals that ensKMM has a lower error bound than the centralized version. This is another example that a proper combination of multiple weaker learners can outpace a single complex learner. Experiments on benchmark datasets confirm the superiority of the proposed algorithm: it achieves higher estimation accuracy with lower running time and is less constrained by the size of test data.

The rest of the paper is organized as follows. Section II briefly reviews the density-ratio estimation problem and the kernel mean matching algorithm. Section III describes the details of the proposed ensemble KMM method. Section IV presents theoretical analysis on the error bound, the time and the space complexities. Experimental results on a set of benchmark datasets are given in Section V. Finally, Section VI concludes the paper.

II. Kernel Mean Matching

A. The Density-Ratio Problem

In a d-dimensional data space, we are given a collection of n_{tr} independent and identically distributed (*i.i.d.*) training samples $\mathcal{X}_{tr} = \{x_i | i = 1, \ldots, n_{tr}\}$ that are from a distribution with probability density function (PDF) $p_{tr}(x)$; and another collection of n_{ts} *i.i.d.* test samples $\mathcal{X}_{ts} = \{x_j | j = 1, \ldots, n_{ts}\}$ that are from a different distribution with density function $p_{ts}(x)$. Suppose $p_{ts}(x)$ is continuous with respect to $p_{tr}(x)$ (i.e. any region of $p_{tr}(x) = 0$ implies $p_{ts}(x) = 0$). The Density-Ratio (DR) problem is to estimate the ratio

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

This paper has been accepted for publication at the 15th IEEE International Conference on Data Mining (ICDM 2015).

from the given finite samples $\mathcal{X}_{tr}$ and $\mathcal{X}_{ts}$.

Recently there have been a number of density-ratio estimation methods being proposed. Some well-known algorithms are the Kernel Mean Matching (KMM) algorithm [14], the Kullback-Leibler Importance Estimation Procedure (KLIEP) algorithm [16], and the unconstrained Least-Squares Importance Fitting (uLSIF) algorithm [17]. The following section describes the KMM algorithm.

B. Kernel Mean Matching Algorithm

The Kernel Mean Matching (KMM) [1], [14] is a well-known algorithm for density ratio estimation based on infinite-order moment matching. The basic idea behind KMM is that two distributions are equivalent if and only if all moments are matched with each other. By making use of universal reproducing kernels, the infinite-order moment matching is implicitly implemented. To be specific, KMM estimates density ratios by minimizing the Maximum Mean Discrepancy (MMD) [18] between the weighted distribution $\beta(x)p_{tr}(x)$ and the distribution $p_{ts}(x)$ in a Reproducing Kernel Hilbert Space (RKHS) $\Phi(x) : x \to \mathcal{F}$,

$$\text{MMD}^2\left(\mathcal{F}, (\beta, p_{tr}), p_{ts}\right) = \left\| E_{x\sim p_{tr}(x)}\left[\beta(x)\Phi(x)\right] - E_{x\sim p_{ts}(x)}\left[\Phi(x)\right]\right\|^2 , \quad (2)$$

where $\|.\|$ is the l_2 norm.

As stated in *Theorem 1.2* and *Lemma 1.3* of [1], if the kernel space is universal and $p_{ts}(x)$ is absolutely continuous with respect to $p_{tr}(x)$, the solution $\beta(x)$ of Eq. 2 converges to $p_{ts}(x) = \beta(x)p_{tr}(x)$.

Using empirical means of $\mathcal{X}_{tr}$ and $\mathcal{X}_{ts}$ to replace the expectations, a Quadratic Programming (QP) problem can be obtained as

$$\begin{aligned}
\hat{\boldsymbol{\beta}} &= \text{argmin}_{\boldsymbol{\beta}}\left[\text{MMD}^2\left(\mathcal{F}, (\beta, p_{tr}), p_{ts}\right)\right] \\
&\approx \text{argmin}_{\boldsymbol{\beta}}\left\| \frac{1}{n_{tr}}\sum_{i=1}^{n_{tr}} \boldsymbol{\beta}_i \Phi(x_i) - \frac{1}{n_{ts}}\sum_{j=1}^{n_{ts}} \Phi(x_j)\right\|^2 \\
&= \text{argmin}_{\boldsymbol{\beta}}[\frac{1}{n_{tr}^2}\sum_{i,i'=1}^{n_{tr}} \beta_i k(x_i, x_{i'})\beta_{i'} \\
&\quad - \frac{2}{n_{tr}n_{ts}}\sum_{i=1}^{n_{tr}}\sum_{j=1}^{n_{ts}} \beta_i k(x_i, x_j) \\
&\quad + \frac{1}{n_{ts}^2}\sum_{j,j'=1}^{n_{ts}} k(x_j, x_{j'})] \\
&= \text{argmin}_{\boldsymbol{\beta}}\left[\frac{1}{2}\boldsymbol{\beta}^T K_{x_{tr},x_{tr}}\boldsymbol{\beta} - \frac{n_{tr}}{n_{ts}}\boldsymbol{\beta}^T K_{x_{tr},x_{ts}}\mathbf{1}\right] ,
\end{aligned} \quad (3)$$

where $K_{x_{tr},x_{tr}}$ and $K_{x_{tr},x_{ts}}$ are two Gram kernel matrix (Radial Basis Function kernel is a well-adopted choice) and $\mathbf{1}_{n_{ts},1}$ is a size of $n_{ts} \times 1$ one vector as

$$\begin{aligned}
K_{x_{tr},x_{tr}} &= \kappa(x_i, x_{i'}),\ \{x_i, x_{i'} \in \mathcal{X}_{tr}\} & (4)\\
K_{x_{tr},x_{ts}} &= \kappa(x_i, x_j),\ \{x_i \in \mathcal{X}_{tr}, x_j \in \mathcal{X}_{ts}\} & (5)\\
\mathbf{1}_{n_{ts},1} &= [1, ..., 1]^T & (6)
\end{aligned}$$

Algorithm 1 KMM Algorithm [14]

Input: $\mathcal{X}_{tr} = \{x_i | i = 1, \ldots, n_{tr}\}$,
$\mathcal{X}_{ts} = \{x_j | j = 1, \ldots, n_{ts}\}$, B, ϵ, σ
Output: $\hat{\boldsymbol{\beta}} = \left(\hat{\beta}_1, \hat{\beta}_2, \ldots, \hat{\beta}_{n_{tr}}\right)^T$
Steps:
1: Compute $K_{x_{tr},x_{tr}}$ (Eq. 4);
2: Compute $K_{x_{tr},x_{ts}}$ (Eq. 5);
3: Boundary constraint:
$\mathbf{0} \le \hat{\boldsymbol{\beta}} \le B\mathbf{1}$;
4: Normalization constraint:
$\frac{1}{n_{tr}}\left[\mathbf{1}_{n_{tr},1}; -\mathbf{1}_{n_{tr},1}\right]^T \hat{\boldsymbol{\beta}} \le \left[(\epsilon+1), (\epsilon+1)\right]^T$;
5: $\hat{\boldsymbol{\beta}} \leftarrow$ QP_solver $\left(K_{x_{tr},x_{tr}}, K_{x_{tr},x_{ts}}, \epsilon, B\right)$ (Eq. 3);

with respect to two constraints

$$\boldsymbol{\beta}_i \in [0, B]\ i = 1, \ldots, n_{tr}, \text{ and}$$
$$\left|\frac{1}{n_{tr}}\sum_{i=1}^{n_{tr}} \boldsymbol{\beta}_i - 1\right| \le \epsilon\ .$$

The first constraint that limits the boundary of $\boldsymbol{\beta}_i$ between 0 and B, reflects the scope of discrepancy between $p_{tr}(x)$ and $p_{ts}(x)$. The second constraint $\left|\frac{1}{n_{tr}}\sum_{i=1}^{n_{tr}} \boldsymbol{\beta}_i - 1\right| \le \epsilon$ is a normalization over $\beta(x)$, since $p_{ts}(x) = \beta(x)p_{tr}(x)$ should approximate a probability density function. The small value ϵ is the normalization precision. The original KMM paper suggests setting these parameters as the boundary $B = 1000$, $\epsilon = {(\sqrt{n_{tr}}-1)}/{\sqrt{n_{tr}}}$, and kernel bandwidth σ as the median of pairwise sample distances [1]. Further study on parameter selection can be found in the work of [2].

Eq. 3 formulates a convex Quadratic Programming (QP) problem with linear constraints. Its global optimum can be computed by using any existing QP solver [19]. Our implementation uses the well-accepted 'interior-point-convex' algorithm in the Matlab toolbox as the QP solver. The detailed procedure of KMM algorithm is listed in Alg. 1.

III. Ensemble Kernel Mean Matching

In this section, we derive an algorithm for ensemble density-ratio estimation based on: 1) arbitrarily dividing the test data into smaller partitions; 2) estimating density ratios for each of these partitions; 3) fusing the estimated density ratios using a weighted sum of the local components.

By making an *i.i.d* assumption on samples in the test set $\mathcal{X}_{ts}$, we arbitrarily divide the test data into k non-overlapping partitions as $\mathcal{X}_{ts} = S^{(1)} \cup S^{(2)} \cup \ldots \cup S^{(k)}$. In this case, the density function estimated over $\mathcal{X}_{ts}$ can be calculated in terms of the density function over different partitions as follows.

$$\begin{aligned}
\hat{p}_{ts}\left(x, x \in \mathcal{X}_{ts}\right) &= \hat{p}_{ts}\left(x,\ x \in S^{(1)} \cup \ldots \cup S^{(k)}\right) \\
&= \sum_{l=1}^{k} \hat{p}_{ts}^{(l)}(x,\ x \in S^{(l)}) \\
&= \sum_{l=1}^{k} \hat{p}_{ts}^{(l)}(x | x \in S^{(l)}) \hat{p}_{ts}^{(l)}(x \in S^{(l)}) \\
&= \sum_{l=1}^{k} \frac{|S^{(l)}|}{n_{ts}} \hat{p}_{ts}^{(l)}(x | x \in S^{(l)}) , \qquad (7)
\end{aligned}$$

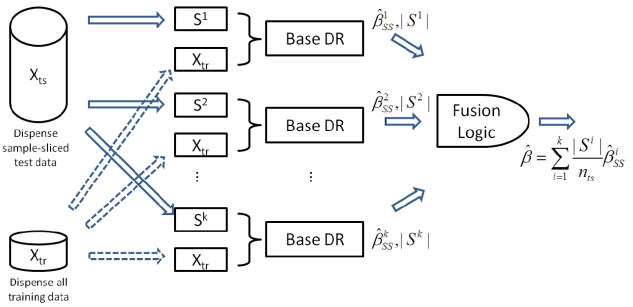

Fig. 1. The structure of the proposed ensemble KMM method.

Algorithm 2 Ensemble KMM Algorithm

Input: $\mathcal{X}_{tr} = \{x_i | i = 1, \ldots, n_{tr}\}$, $\mathcal{X}_{ts} = \{x_j | j = 1, \ldots, n_{ts}\}$, B, ϵ, σ, k

Output: $\hat{\boldsymbol{\beta}}_{\boldsymbol{ens}} = \left(\hat{\beta}_1, \hat{\beta}_2, \ldots, \hat{\beta}_{n_{tr}}\right)^T$

Steps:

1: Divide $\mathcal{X}_{ts}$ into k partitions: $S^{(1)}, \ldots, S^{(k)}$;
2: **for** $l = 1$ **to** k **do**
3: $\quad \hat{\boldsymbol{\beta}}_{\boldsymbol{ss}}^{(\boldsymbol{l})} \leftarrow$ KMM$(\mathcal{X}_{tr}, S^{(l)}, B, \epsilon, \sigma)$ (Call Alg. 1);
4: **end for**
5: $\hat{\boldsymbol{\beta}}_{\boldsymbol{ens}} = \sum_{l=1}^{k} \frac{|S^{(l)}|}{n_{ts}} \cdot \hat{\boldsymbol{\beta}}_{\boldsymbol{ss}}^{(\boldsymbol{l})}$;

where $|S^{(l)}|$ is the cardinality of the set $S^{(l)}$.

Assume we have a relatively small number of training samples $\mathcal{X}_{tr}$, and let $\hat{p}_{tr}(x)$ be its estimated density function, the density-ratio between test and training data is defined as:

$$\begin{aligned} \hat{\beta}(x) &= \frac{\hat{p}_{ts}(x,\, x \in \mathcal{X}_{ts})}{\hat{p}_{tr}(x,\, x \in \mathcal{X}_{tr})} \\ &= \frac{\sum_{l=1}^{k} \frac{|S^{(l)}|}{n_{ts}} \hat{p}_{ts}^{(l)}(x | x \in S^{(l)})}{\hat{p}_{tr}(x,\, x \in \mathcal{X}_{tr})} \\ &= \sum_{l=1}^{k} \frac{|S^{(l)}|}{n_{ts}} \cdot \frac{\hat{p}_{ts}^{(l)}(x | x \in S^{(l)})}{\hat{p}_{tr}(x,\, x \in \mathcal{X}_{tr})} \,. \end{aligned} \tag{8}$$

Define a local component density-ratio $\hat{\beta}_{ss}^{(l)}$, which is estimated using the test samples $S^{(l)}$ of partition l and the training samples $\mathcal{X}_{tr}$ as

$$\hat{\beta}_{ss}^{(l)} = \frac{\hat{p}_{ts}^{(l)}(x | x \in S^{(l)})}{\hat{p}_{tr}(x,\, x \in \mathcal{X}_{tr})} \,, \tag{9}$$

the density ratio for the ensemble can be written as a linear combination of the component density ratios as

$$\hat{\beta}_{ens} = \sum_{l=1}^{k} \frac{|S^{(l)}|}{n_{ts}} \cdot \hat{\beta}_{ss}^{(l)} \,. \tag{10}$$

This indicates that we can divide a large test set into a number of non-overlapping partitions, and estimate the density ratio for the test set as the weighted sum of the density ratios for different partitions.

The procedure of calculating ensemble density ratios can be easily implemented on distributed architectures as illustrated in Figure 1. First, the test data are divided into k non-overlapping partitions and each partition of test samples along with all training samples are transferred to a component estimator. Then, each component estimator independently produces its own estimate using its partition of test samples along with all the training samples. After having component density ratios from these k estimators, the decision fusion layer outputs the ensemble density ratios by taking the weighted sum of component estimates (Eq. 10). With KMM as its component estimators, the ensemble KMM algorithm is formulated as described in Alg. 2.

The proposed approach only partitions the test samples across the members of the ensemble while transferring all the training samples to each component estimator. Due to the fact that the number of training samples is usually small, when implementing this procedure on a distributed architecture, the communication overhead that results from moving the training data and the calculated density ratios is expected to be relatively small.

IV. Analysis

This section first analyzes the theoretical properties of the proposed ensemble KMM algorithm, and derives a bound for the density ratio estimation error. Second, the time and space complexities are analyzed.

A. Error Bound

Similar to the [15], [20], we can make two assumptions on the density ratio and the kernel functions. For a compact domain of problem, these assumptions are not too restrictive.

Assumption 1. *Continuity Assumption: The density ratio $\beta(x)$ is well-defined and bounded with B such that $B < \infty$.*

Assumption 2. *Compactness Assumption: The kernel k is continuous in the domain X, that is $||k||_\infty \leq C < \infty$.*

Gretton et al. [1] proposed the following error-bound for the centralized KMM.

Lemma 1. *Under the* Assumption 1 and 2*, with probability at least $1 - \delta$, the convergence error of β estimated with KMM is bounded as:*

$$\left\| \frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} \beta(x_i^{tr}) \Phi(x_i^{tr}) - \frac{1}{n_{ts}} \sum_{j=1}^{n_{ts}} \Phi(x_j^{ts}) \right\| \leq \left(1 + \sqrt{2 \log\left(\frac{2}{\delta}\right)}\right) C \sqrt{\frac{B^2}{n_{tr}} + \frac{1}{n_{ts}}} \,. \tag{11}$$

Proof: See *Lemma 1.5* in [1]. ■

In the following theorem, we show that the error of the ensemble KMM has a tighter bound.

Theorem 1. *The error of the ensemble KMM method with k*

partitions of test data, each with m samples, is bounded as:

$$\left\| \frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} \beta_{ens}(x_i^{tr}) \Phi(x_i^{tr}) - \frac{1}{n_{ts}} \sum_{j=1}^{n_{ts}} \Phi(x_j^{ts}) \right\| \leq \frac{1}{k^{3/4}} C \sqrt{2 \log\left(\frac{2}{\delta}\right)} \sqrt{\frac{B^2}{n_{tr}} + \frac{1}{m}} + C \sqrt{\frac{B^2}{n_{tr}} + \frac{1}{mk}} \,. \quad (12)$$

Proof: First we provide the tail bound. The density ratio of each component KMM is estimated based on n_{tr} training samples and m test samples. According to [1], the bound of change in density-ratio for this component is $4C^2 \left(\frac{B^2}{n_{tr}} + \frac{1}{m}\right)$. Applying McDiarmid tail bound [21], the variance of the ensemble output would be

$$\sigma_{ens}^2 = \frac{1}{k^2} \sqrt{k} C^2 \left(\frac{B^2}{n_{tr}} + \frac{1}{m}\right) \,. \quad (13)$$

Define $\Gamma(\mathcal{X}_{tr}, \mathcal{X}_{ts}) :=$

$$\left\| \frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} \beta_{ens}(x_i^{tr}) \Phi(x_i^{tr}) - \frac{1}{n_{ts}} \sum_{j=1}^{n_{ts}} \Phi(x_j^{ts}) \right\| \,. \quad (14)$$

For an arbitrary small number ϵ, it can be shown that

$$P\left\{ |\Gamma(\mathcal{X}_{tr}, \mathcal{X}_{ts}) - E\left(\Gamma(\mathcal{X}_{tr}, \mathcal{X}_{ts})\right)| > \epsilon \right\} \leq 2 \exp\left(- \frac{\epsilon^2}{2 \frac{1}{k^{3/2}} C^2 \left(\frac{B^2}{n_{tr}} + \frac{1}{m}\right)} \right) \,. \quad (15)$$

Applying the concentration inequality [22], with probability of $(1 - \delta)$ the 2-side tail bound is:

$$|\Gamma(\mathcal{X}_{tr}, \mathcal{X}_{ts}) - E\left(\Gamma(\mathcal{X}_{tr}, \mathcal{X}_{ts})\right)| \leq \frac{1}{k^{3/4}} C \sqrt{2 \log\left(\frac{2}{\delta}\right)} \sqrt{\frac{B^2}{n_{tr}} + \frac{1}{m}} \,. \quad (16)$$

Second, the bound of expectation is derived by applying Jensen's inequality [23] as:

$$\begin{aligned} E\left(\Gamma(\mathcal{X}_{tr}, \mathcal{X}_{ts})\right) &\leq \sqrt{E\left(\Gamma(\mathcal{X}_{tr}, \mathcal{X}_{ts})^2\right)} \\ &\leq \sqrt{C^2 \left(\frac{B^2}{n_{tr}} + \frac{1}{mk}\right)} \,. \end{aligned} \quad (17)$$

Combining the tail bound (Eq. 16) and the expectation bound (Eq. 17) proves the theorem.

■

In order to illustrate the importance of this error bound, we first show that when we increase the number of components in the ensemble KMM method, while maintaining m test samples at each partition, it is guaranteed to achieve a lower error bound.

Corollary 1. *For a fixed number of test samples per partition, the error bound of an ensemble KMM is monotonically decreasing with respect to the number of components in the ensemble.*

Proof: According to *Theorem 1*, the error bound of ensemble KMM is

$$\mathcal{B}_{\text{ensKMM}} = \frac{1}{k^{3/4}} C \sqrt{2 \log\left(\frac{2}{\delta}\right)} \sqrt{\frac{B^2}{n_{tr}} + \frac{1}{m}} + C \sqrt{\frac{B^2}{n_{tr}} + \frac{1}{mk}} \,. \quad (18)$$

Taking derivative with respect to k, we have

$$\begin{aligned} \frac{\partial \mathcal{B}_{\text{ensKMM}}}{\partial k} &= -\frac{3C}{4k^{7/4}} \sqrt{2 \log\left(\frac{2}{\delta}\right)} \sqrt{\frac{B^2}{n_{tr}} + \frac{1}{m}} - \frac{C}{2\sqrt{\frac{B^2}{n_{tr}} + \frac{1}{mk}}} \left(\frac{1}{mk^2}\right) \\ &\leq 0 \,. \end{aligned} \quad (19)$$

As both terms are less or equal to zero, $\mathcal{B}_{\text{ensKMM}}$ is a monotonically decreasing function with respect to k. ■

Another interesting result about the error bound of the ensemble KMM can be derived by comparing an ensemble KMM on a test set with a centralized KMM whose density ratio is estimated using the whole test set.

Corollary 2. *For a given test data, the ensemble KMM on k partitions has a tighter error bound than the centralized KMM which uses all the data.*

Proof: Let $\mathcal{B}_{\text{cenKMM}}$ and $\mathcal{B}_{\text{ensKMM}}$ be the error bounds of the centralized KMM and ensemble KMM. From *Lemma 1*, we have

$$\mathcal{B}_{\text{cenKMM}} = \left(1 + \sqrt{2 \log\left(\frac{2}{\delta}\right)}\right) C \sqrt{\frac{B^2}{n_{tr}} + \frac{1}{n_{ts}}} \,.$$

Because $n_{ts} = mk$, from *Theorem 1* we have

$$\begin{aligned} \mathcal{B}_{\text{ensKMM}} &= \frac{1}{k^{3/4}} C \sqrt{2 \log\left(\frac{2}{\delta}\right)} \sqrt{\frac{B^2}{n_{tr}} + \frac{1}{m}} + C \sqrt{\frac{B^2}{n_{tr}} + \frac{1}{mk}} \\ &= C \sqrt{2 \log\left(\frac{2}{\delta}\right)} \sqrt{\frac{B^2}{n_{tr}} \frac{1}{k\sqrt{k}} + \frac{k}{n_{ts}} \frac{1}{k\sqrt{k}}} + C \sqrt{\frac{B^2}{n_{tr}} + \frac{1}{n_{ts}}} \\ &= C \sqrt{2 \log\left(\frac{2}{\delta}\right)} \sqrt{\frac{B^2}{n_{tr}} \frac{1}{k^{3/2}} + \frac{1}{n_{ts}} \frac{1}{\sqrt{k}}} + C \sqrt{\frac{B^2}{n_{tr}} + \frac{1}{n_{ts}}} \,. \end{aligned} \quad (20)$$

The number of estimators in the ensemble $k \geq 2$, therefore

TABLE I. TIME AND SPACE COMPLEXITIES OF CENTRALIZED KMM AND ENSEMBLE KMM.

	Time Complexity	Space Complexity
cenKMM	$\mathcal{O}(n_{\text{tr}}^3 + n_{\text{tr}}^2 d + n_{\text{tr}} n_{\text{ts}} d)$	$\mathcal{O}(n_{\text{tr}}^2 + n_{tr} n_{ts})$
ensKMM	$\mathcal{O}(n_{\text{tr}}^3 + n_{\text{tr}}^2 d + n_{\text{tr}} m d + k n_{tr})$	$\mathcal{O}(n_{\text{tr}}^2 + n_{tr} m + k n_{tr})$

$\frac{B^2}{n_{tr}} \frac{1}{k^{3/2}} < \frac{B^2}{n_{tr}}$ and $\frac{1}{n_{ts}} \frac{1}{\sqrt{k}} < \frac{1}{n_{ts}}$. Then,

$$\begin{aligned} \mathcal{B}_{\text{ensKMM}} &\leq C\sqrt{2\log\left(\frac{2}{\delta}\right)}\sqrt{\frac{B^2}{n_{tr}} + \frac{1}{n_{ts}}} \\ &\quad + C\sqrt{\frac{B^2}{n_{tr}} + \frac{1}{n_{ts}}} \\ &= \left(1 + \sqrt{2\log\left(\frac{2}{\delta}\right)}\right) C\sqrt{\frac{B^2}{n_{tr}} + \frac{1}{n_{ts}}} \\ &= \mathcal{B}_{\text{cenKMM}} \,. \end{aligned} \tag{21}$$

This proves the corollary. ■

B. Time and Space Complexities

First we analyze the time complexity. Considering centralized KMM (cenKMM) with n_{tr} training samples and n_{ts} test samples in a d-dimensional space, the computation of kernel matrix $K_{x_{\text{tr}},x_{\text{tr}}}$ is $\mathcal{O}(n_{\text{tr}}^2 d)$. The computation of kernel matrix $K_{x_{\text{tr}},x_{\text{ts}}}$ is $\mathcal{O}(n_{\text{tr}} n_{\text{ts}} d)$. The QP problem has the complexity of $\mathcal{O}(n_{\text{tr}}^3)$.[1] Thus, the total computational complexity for centralized KMM is $\mathcal{O}(n_{\text{tr}}^3 + n_{\text{tr}}^2 d + n_{\text{tr}} n_{\text{ts}} d)$. For distributed implementation of ensemble KMM (ensKMM), the total computation will be the maximum of component estimators plus the computation of the fusion step. It is obvious that the computational complexity of all k component estimators depends on the size of partition m. The computation complexity of component KMM is $\mathcal{O}(n_{\text{tr}}^3 + n_{\text{tr}}^2 d + n_{\text{tr}} m d)$. The fusion step has complexity as $\mathcal{O}(k n_{\text{tr}})$. Therefore the time complexity of ensKMM is $\mathcal{O}(n_{\text{tr}}^3 + n_{\text{tr}}^2 d + n_{\text{tr}} m d + k n_{tr})$, which does not depend on the size of test data.

In terms of space complexity, the centralized KMM requires the storage of two kernel matrix $K_{x_{\text{tr}},x_{\text{tr}}}$ and $K_{x_{\text{tr}},x_{\text{ts}}}$, which is $\mathcal{O}(n_{\text{tr}}^2 + n_{tr} n_{ts})$. Considering ensemble KMM, the space requirement of component estimators is $\mathcal{O}(n_{\text{tr}}^2 + n_{tr} m)$. The fusion step needs a space of $\mathcal{O}(k n_{tr})$. Therefore, the space complexity of ensemble KMM is $\mathcal{O}(n_{\text{tr}}^2 + n_{tr} m + k n_{tr})$. Compared to centralized KMM, the space complexity does not depend on n_{ts}. Table I shows a clear difference between the centralized KMM and the proposed ensemble method.

V. EXPERIMENTS

A. Datasets and Setup

To examine the effectiveness and efficiency of the proposed ensemble approach, we conducted experiments on four benchmark datasets: PIE, CovType, MNIST, RCV1-4Class, whose properties are listed in Table II. The RCV1-4Class is a subset of RCV dataset containing four categories of documents within the original collection. The CovType dataset is formed with the first 50K samples from the original collection. These datasets are publicly available.[2]

TABLE II. THE PROPERTIES OF DATASETS USED TO EVALUATE ENSEMBLE KMM METHOD.

Dataset	Type	# Instances	# Features
PIE	face images	11,554	4,096
CovType	multivariate	50,000	54
MNIST	digit images	70,000	784
RCV1-4Class	documents	9,625	29,992

We take the same sampling bias scheme based on joint features as described in [14]. Samples in each dataset are selected with known probability $P(s = 1|x_i) = \exp(-\|x_i - \bar{x}\|^2)$ to formulate a training collection with the size of 500. The remaining data are reserved to formulate the test collection.

Since the probabilities of biased selected samples are known, we can obtain the ground truth of density ratios for each sample in the training collection, i.e. the reciprocal of $P(s = 1|x_i)$. Therefore, similar to previous work on the density-ratio estimation [16], the performance of different estimation methods are evaluated with the Normalized Mean Squared Error (NMSE), which is defined as

$$\text{NMSE} = \frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} \left(\frac{\hat{\beta}_i}{\sum_{j=1}^{n_{tr}} \hat{\beta}_j} - \frac{\beta_i}{\sum_{j=1}^{n_{tr}} \beta_j} \right)^2 ,$$

where $\hat{\beta}_i$ is the estimate of the density ratio at training data point x_i, and β_i is the ground truth of the density ratio calculated as $\beta_i = {}^1\!/_{P(s=1|x_i)}$.

For each dataset, under the same setup we compared the proposed ensemble KMM (ensKMM) method with two methods:

1) Centralized KMM method: This centralized method uses all training and all test samples to produce density ratios. Obviously, the scalability of this centralized method is limited.
2) Uniform sampling method: This is a baseline method which uniformly sub-samples a fraction of test data equal to that of each component of the ensemble KMM and uses them to estimate density ratios.

B. Results

a) Increasing number of component KMMs with fixed sample size in each partition: This set of experiments is conducted by varying the number of component KMMs from 4 to 20 with increments of 2. Each KMM component takes $m = 5\% n_{ts}$ test samples. The performance and running times are plotted in Figure 2.

The left figures show the estimation Normalized Mean Squared Error (NMSE). It can be observed that by increasing the number of component estimators k, the estimation errors of the ensemble KMM and centralized KMM decrease. The decreasing estimation error for centralized KMM is consistent with *Lemma 1*, because more test data is being introduced.

[1] We noticed that for convex quadratic programming problem there are some variants which have improved time complexity [24]. This does not affect the result of complexity comparison presented here.

[2] Datasets were downloaded from http://www.cad.zju.edu.cn/home/dengcai/Data/data.html.

The uniform sampling method, on the other hand, is the case of estimation with the same number of test samples as one component in the ensemble KMM, which performs the worst. The ensemble KMM achieves the smallest error. The right figures plot running times for corresponding setups. It is not a surprise that the centralized KMM takes the longest time and keeps increasing because its estimation is executed on all test data. The running time of the ensemble KMM and the uniform sampling KMM is nearly at the same level, much less than the centralized KMM.

b) Using all test data and changing the number of component KMM estimators: The second set of experiments uses all test samples and examines performance by changing the number of component KMM estimators from 4 to 20 with increments of 2. Figure 3 plots estimation errors in terms of NMSE and the running time for each setup.

As seen from the figures, the centralized KMM estimates density ratios based on all test data. Therefore, the estimation error and running time is one number for each dataset. The error of uniform sampling method increases due to the decreasing number of test samples being included. The ensemble KMM has the most accurate estimation, while its running time is just slightly higher than the uniform sampling method, which is understandable due to the calculation of the final density-ratio fusion step.

VI. Conclusion

This paper presents an ensemble algorithm for Kernel Mean Matching (KMM) which depends on fusing density ratios estimated from different partitions of test data. The proposed algorithm achieves very promising results compared to that of the centralized KMM algorithms and it is characterized by strong theoretical guarantees. Moreover, the distributed nature of the ensemble KMM algorithm allows its implementation on modern architectures. When there are abundant amounts of data in the test domain, the algorithm allows for multiple instances of KMM to run in parallel and then fuses multiple results to obtain a more accurate density ratio. When the size of the test data is relatively small, the algorithm can still be used to create many weak learners into a strong estimator for density ratio. In both cases, the ensemble KMM results in a significant reduction in the run-time when compared to the centralized KMM that runs on the same amount of test data.

It is worth noting that the proposed ensemble KMM approach is derived based on the density-ratio problem formulation. Therefore, it is not limited to any specific density-ratio method and accordingly can easily be extended to build an ensemble of a variety of estimators other than KMM, or even a combination of them. Future work includes the derivation of theoretical guarantees for other density-ratio estimation methods.

References

[1] A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf, "Covariate shift by kernel mean matching," in *Dataset shift in machine learning*. Cambridge, MA: MIT press, 2009, pp. 131–160.

[2] Y.-Q. Miao, A. K. Farahat, and M. S. Kamel, "Auto-tuning kernel mean matching," in *Workshop on Incremental Clustering, Concept Drift and Novelty Detection at the IEEE 13th International Conference on Data Mining (ICDM)*, 2013, pp. 560–567.

[3] ——, "Discriminative density-ratio estimation," in *Proceedings of the 2014 SIAM International Conference on Data Mining (SDM)*, 2014, pp. 830–838.

[4] S. Hido, Y. Tsuboi, H. Kashima, M. Sugiyama, and T. Kanamori, "Statistical outlier detection using direct density ratio estimation," *Knowledge and Information Systems*, vol. 26, no. 2, pp. 309–336, 2011.

[5] Y.-Q. Miao, A. K. Farahat, and M. S. Kamel, "Locally adaptive density ratio for detecting novelty in twitter streams," in *The 6th International Workshop on Modeling Social Media (MSM) - Behavioral Analytics in Social Media, Big Data and the Web*, 2015.

[6] M. Kawakita and T. Kanamori, "Semi-supervised learning with density-ratio estimation," *Machine Learning*, vol. 91, no. 2, pp. 189–209, 2013.

[7] Y. Tan and X. Zhu, "Dragging: Density-ratio bagging," Computer Science, University of Wisconsin-Madison, Tech. Rep. TR1795, 2013.

[8] C. Elkan, "Preserving privacy in data mining via importance weighting," in *Privacy and Security Issues in Data Mining and Machine Learning*. Springer, 2011, pp. 15–21.

[9] S. Liu, J. A. Quinn, M. U. Gutmann, and M. Sugiyama, "Direct learning of sparse changes in markov networks by density ratio estimation," in *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML)*. Springer, 2013, pp. 596–611.

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

[11] V. Vapnik, I. Braga, and R. Izmailov, "A constructive setting for the problem of density ratio estimation," in *Proceedings of the 2014 SIAM International Conference on Data Mining (SDM)*, 2014, pp. 434–442.

[12] G. S. Fishman, *Monte Carlo: concepts, algorithms, and applications*. Springer, 1996.

[13] J. Jacod, "Multivariate point processes: predictable projection, Radon-Nikodym derivatives, representation of martingales," *Probability Theory and Related Fields*, vol. 31, no. 3, pp. 235–253, 1975.

[14] 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 (NIPS)*, vol. 19, 2007, pp. 601–608.

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

[16] 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 (NIPS)*, vol. 20, 2008, pp. 1433–1440.

[17] T. Kanamori, S. Hido, and M. Sugiyama, "Efficient direct density ratio estimation for non-stationarity adaptation and outlier detection," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 21, 2008, pp. 809–816.

[18] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two sample problem," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 19. Cambridge, MA: MIT press, 2007, pp. 513–520.

[19] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[20] 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.

[21] C. McDiarmid, "Concentration," in *Probabilistic methods for algorithmic discrete mathematics*. Springer, 1998, pp. 195–248.

[22] P. L. Bartlett, S. Boucheron, and G. Lugosi, "Model selection and error estimation," *Machine Learning*, vol. 48, no. 1-3, pp. 85–113, 2002.

[23] M. Kuczma, *An introduction to the theory of functional equations and inequalities: Cauchy's equation and Jensen's inequality*. Springer Science & Business Media, 2009.

[24] R. Bellman, *Introduction to matrix analysis*. SIAM, 1970, vol. 960.

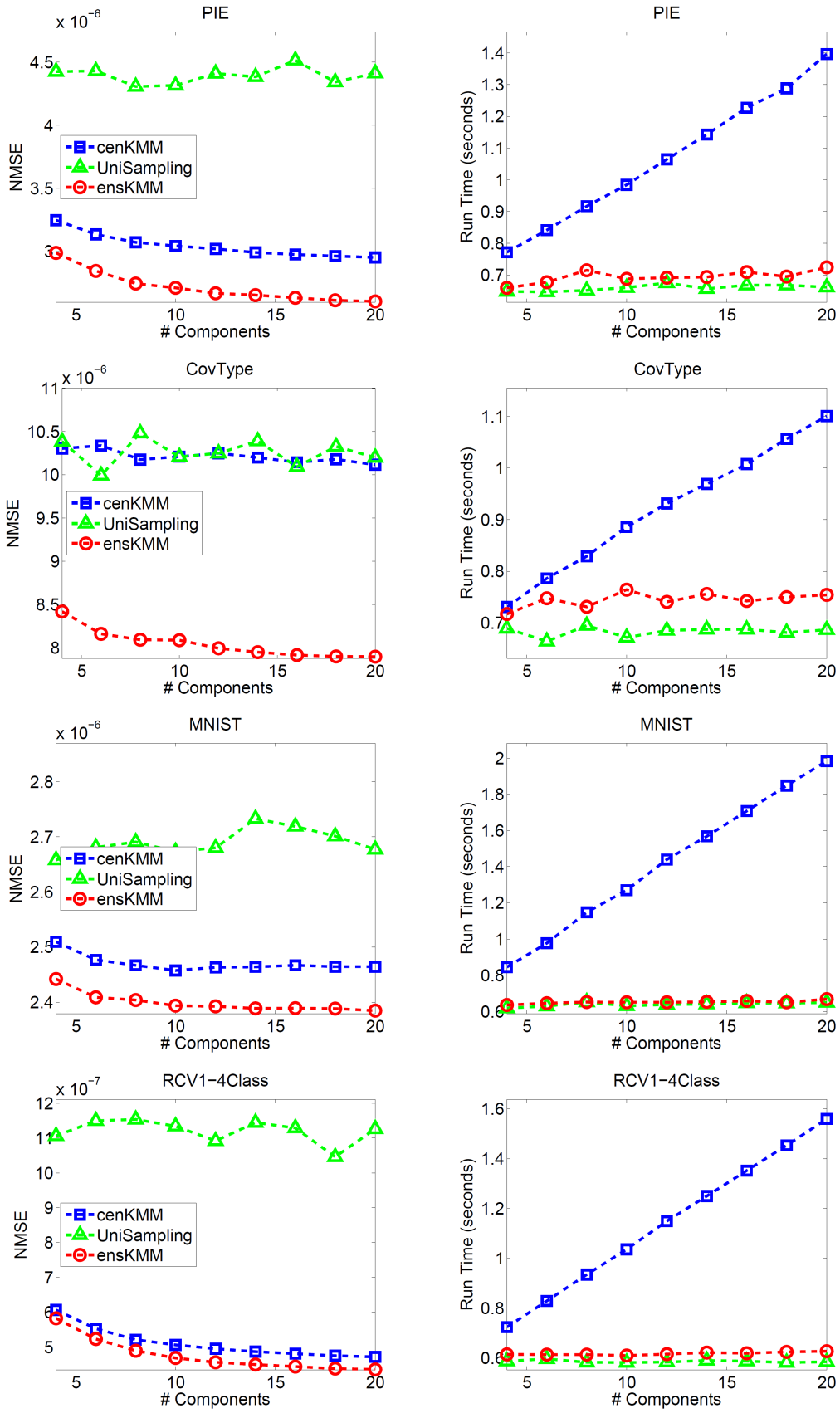

Fig. 2. The error and running time for different number of component KMMs with a fixed number of test samples in each partition.

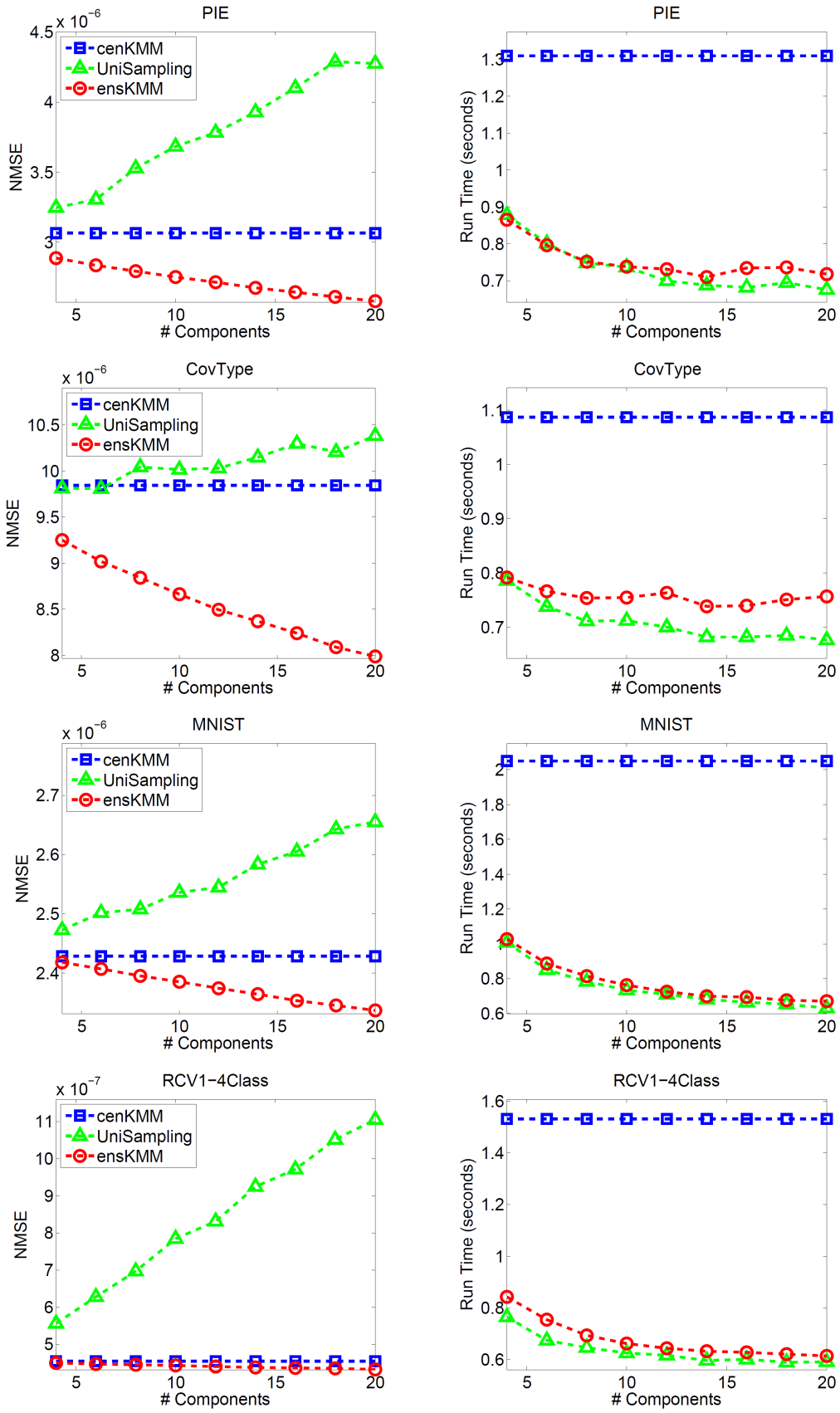

Fig. 3. The error and running time for different number of component KMMs using all test data.