Information-Maximization Clustering based on Squared-Loss Mutual Information*

Masashi Sugiyama[1], Gang Niu[1], Makoto Yamada[2], Manabu Kimura[1], and Hirotaka Hachiya[1]

[1]Tokyo Institute of Technology, Japan.
[2]Yahoo! Labs, USA.

sugi@cs.titech.ac.jp http://sugiyama-www.cs.titech.ac.jp/˜sugi

Abstract

Information-maximization clustering learns a probabilistic classifier in an unsupervised manner so that mutual information between feature vectors and cluster assignments is maximized. A notable advantage of this approach is that it only involves continuous optimization of model parameters, which is substantially simpler than discrete optimization of cluster assignments. However, existing methods still involve non-convex optimization problems, and therefore finding a good local optimal solution is not straightforward in practice. In this paper, we propose an alternative information-maximization clustering method based on a *squared-loss* variant of mutual information. This novel approach gives a clustering solution *analytically* in a computationally efficient way via kernel eigenvalue decomposition. Furthermore, we provide a practical model selection procedure that allows us to objectively optimize tuning parameters included in the kernel function. Through experiments, we demonstrate the usefulness of the proposed approach.

Keywords

Clustering, Information Maximization, Squared-Loss Mutual Information.

1 Introduction

The goal of *clustering* is to classify data samples into disjoint groups in an unsupervised manner. *K-means* (MacQueen, 1967) is a classic but still popular clustering algorithm. However, since k-means only produces linearly separated clusters, its usefulness is rather limited in practice.

To cope with this problem, various non-linear clustering methods have been developed. *Kernel k-means* (Girolami, 2002) performs k-means in a feature space induced by a reproducing kernel function (Schölkopf & Smola, 2002). *Spectral clustering* (Shi & Malik,

*An earlier version of this paper is presented at International Conference on Machine Learning in 2011 (Sugiyama et al., 2011).

2000; Ng et al., 2002) first unfolds non-linear data manifolds by a spectral embedding method, and then performs k-means in the embedded space. *Blurring mean-shift* (Fukunaga & Hostetler, 1975; Carreira-Perpiñán, 2006) uses a non-parametric kernel density estimator for modeling the data-generating probability density, and finds clusters based on the modes of the estimated density. *Discriminative clustering* learns a discriminative classifier for separating clusters, where class labels are also treated as parameters to be optimized (Xu et al., 2005; Bach & Harchaoui, 2008). *Dependence-maximization clustering* determines cluster assignments so that their dependence on input data is maximized (Song et al., 2007; Faivishevsky & Goldberger, 2010). See Section 3 for comprehensive reviews of existing clustering methods.

These non-linear clustering techniques would be capable of handling highly complex real-world data. However, they suffer from lack of objective model selection strategies[1]. More specifically, the above non-linear clustering methods contain tuning parameters such as the width of Gaussian functions and the number of nearest neighbors in kernel functions or similarity measures, and these tuning parameter values need to be manually determined in an unsupervised manner. The problem of learning similarities/kernels was addressed in earlier works (Meila & Shi, 2001; Shental et al., 2003; Cour et al., 2005; Bach & Jordan, 2006), but they considered supervised setups, i.e., labeled samples are assumed to be given. Zelnik-Manor and Perona (2005) provided a useful unsupervised heuristic to determine the similarity in a data-dependent way. However, it still requires the number of nearest neighbors to be determined manually (although the magic number "7" was shown to work well in their experiments).

Another line of clustering framework called *information-maximization clustering* exhibited the state-of-the-art performance (Agakov & Barber, 2006; Gomes et al., 2010). In this information-maximization approach, probabilistic classifiers such as a kernelized Gaussian classifier (Agakov & Barber, 2006) and a kernel logistic regression classifier (Gomes et al., 2010) are learned so that *mutual information* (MI) between feature vectors and cluster assignments is maximized in an unsupervised manner. A notable advantage of this approach is that classifier training is formulated as continuous optimization problems, which are substantially simpler than discrete optimization of cluster assignments. Indeed, classifier training can be carried out in computationally efficient manners by a gradient method (Agakov & Barber, 2006) or a quasi-Newton method (Gomes et al., 2010). Furthermore, Agakov and Barber (2006) provided a model selection strategy based on the information-maximization principle. Thus, kernel parameters can be systematically optimized in an unsupervised way.

However, in the above MI-based clustering approach, the optimization problems are non-convex, and finding a good local optimal solution is not straightforward in practice. The goal of this paper is to overcome this problem by providing a novel information-maximization clustering method. More specifically, we propose to employ a variant of MI called *squared-loss MI* (SMI), and develop a new clustering algorithm whose solution can be computed analytically in a computationally efficient way via kernel eigenvalue

[1] "Model selection" in this paper refers to the choice of tuning parameters in kernel functions or similarity measures, not the choice of the number of clusters.

decomposition. Furthermore, for kernel parameter optimization, we propose to use a non-parametric SMI estimator called *least-squares MI* (LSMI) (Suzuki et al., 2009; Sugiyama, 2013), which was proved to achieve the optimal convergence rate with an analytic-form solution. Through experiments on various real-world datasets such as images, natural languages, accelerometric sensors, and speeches, we demonstrate the usefulness of the proposed clustering method.

The rest of this paper is structured as follows. In Section 2, we describe our proposed information-maximization clustering method based on SMI and analyze its properties. Then the proposed method is compared with existing clustering methods qualitatively in Section 3 and quantitatively in Section 4. Finally, this paper is concluded in Section 5.

2 Information-Maximization Clustering with Squared-Loss Mutual Information

In this section, we describe our proposed clustering algorithm.

2.1 Formulation of Information-Maximization Clustering

Suppose that we are given d-dimensional i.i.d. feature vectors of size n,

$$\{\boldsymbol{x}_i \mid \boldsymbol{x}_i \in \mathbb{R}^d\}_{i=1}^n,$$

which are drawn independently from a probability distribution with density $p^*(\boldsymbol{x})$. The goal of clustering is to give cluster assignments,

$$\{y_i \mid y_i \in \{1, \ldots, c\}\}_{i=1}^n,$$

to the feature vectors $\{\boldsymbol{x}_i\}_{i=1}^n$, where c denotes the number of classes. Throughout this paper, we assume that c is known.

In order to solve the clustering problem, we take the *information-maximization* approach (Agakov & Barber, 2006; Gomes et al., 2010). That is, we regard clustering as an unsupervised classification problem, and learn the class-posterior probability $p^*(y|\boldsymbol{x})$ so that "information" between feature vector $\boldsymbol{x}$ and class label y is maximized.

The *dependence-maximization* approach (Song et al., 2007; Faivishevsky & Goldberger, 2010) (see also Section 3.7) is related to, but substantially different from the above information-maximization approach. In the dependence-maximization approach, cluster assignments $\{y_i\}_{i=1}^n$ are directly determined so that their dependence on feature vectors $\{\boldsymbol{x}_i\}_{i=1}^n$ is maximized. Thus, the dependence-maximization approach intrinsically involves combinatorial optimization with respect to $\{y_i\}_{i=1}^n$. On the other hand, the information-maximization approach involves continuous optimization with respect to the parameter $\boldsymbol{\alpha}$ included in a class-posterior model $p(y|\boldsymbol{x};\boldsymbol{\alpha})$. This continuous optimization of $\boldsymbol{\alpha}$ is substantially easier to solve than discrete optimization of $\{y_i\}_{i=1}^n$.

Another advantage of the information-maximization approach is that it naturally allows out-of-sample clustering based on the discriminative model $p(y|\boldsymbol{x};\boldsymbol{\alpha})$, i.e., a cluster assignment for a new feature vector can be obtained based on the learned discriminative model.

2.2 Squared-Loss Mutual Information

As an information measure, we adopt *squared-loss mutual information* (SMI). SMI between feature vector $\boldsymbol{x}$ and class label y is defined by

$$\mathrm{SMI} := \frac{1}{2}\int \sum_{y=1}^{c} p^*(\boldsymbol{x})p^*(y)\left(\frac{p^*(\boldsymbol{x},y)}{p^*(\boldsymbol{x})p^*(y)}-1\right)^2 \mathrm{d}\boldsymbol{x}, \tag{1}$$

where $p^*(\boldsymbol{x},y)$ denotes the joint density of $\boldsymbol{x}$ and y, and $p^*(y)$ is the marginal probability of y. SMI is the *Pearson divergence* (Pearson, 1900) from $p^*(\boldsymbol{x},y)$ to $p^*(\boldsymbol{x})p^*(y)$, while the ordinary MI (Cover & Thomas, 2006),

$$\mathrm{MI} := \int \sum_{y=1}^{c} p^*(\boldsymbol{x},y)\log\frac{p^*(\boldsymbol{x},y)}{p^*(\boldsymbol{x})p^*(y)}\mathrm{d}\boldsymbol{x}, \tag{2}$$

is the *Kullback-Leibler divergence* (Kullback & Leibler, 1951) from $p^*(\boldsymbol{x},y)$ to $p^*(\boldsymbol{x})p^*(y)$. The Pearson divergence and the Kullback-Leibler divergence both belong to the class of *Ali-Silvey-Csiszár divergences* (which is also known as *f-divergences*, see Ali and Silvey (1966); Csiszár (1967)), and thus they share similar properties. For example, SMI is non-negative and takes zero if and only if $\boldsymbol{x}$ and y are statistically independent, as the ordinary MI.

In the existing information-maximization clustering methods (Agakov & Barber, 2006; Gomes et al., 2010) (see also Section 3.8), MI is used as the information measure. On the other hand, in this paper, we adopt SMI because it allows us to develop a clustering algorithm whose solution can be computed analytically in a computationally efficient way via kernel eigenvalue decomposition.

2.3 Clustering by SMI Maximization

Here, we give a computationally-efficient clustering algorithm based on SMI (1).

Expanding the squared term in Eq.(1), we can express SMI as

$$\begin{aligned}\mathrm{SMI} &= \frac{1}{2}\int \sum_{y=1}^{c} p^*(\boldsymbol{x})p^*(y)\left(\frac{p^*(\boldsymbol{x},y)}{p^*(\boldsymbol{x})p^*(y)}\right)^2 \mathrm{d}\boldsymbol{x} - \int \sum_{y=1}^{c} p^*(\boldsymbol{x})p^*(y)\frac{p^*(\boldsymbol{x},y)}{p^*(\boldsymbol{x})p^*(y)}\mathrm{d}\boldsymbol{x} + \frac{1}{2}\\ &= \frac{1}{2}\int \sum_{y=1}^{c} p^*(y|\boldsymbol{x})p^*(\boldsymbol{x})\frac{p^*(y|\boldsymbol{x})}{p^*(y)}\mathrm{d}\boldsymbol{x} - \frac{1}{2}.\end{aligned} \tag{3}$$

Suppose that the class-prior probability $p^*(y)$ is set to a user-specified value π_y for $y = 1, \ldots, c$, where $\pi_y > 0$ and $\sum_{y=1}^{c} \pi_y = 1$. Without loss of generality, we assume that $\{\pi_y\}_{y=1}^{c}$ are sorted in the ascending order:

$$\pi_1 \leq \cdots \leq \pi_c.$$

If $\{\pi_y\}_{y=1}^{c}$ is unknown, we may merely adopt the uniform class-prior distribution:

$$p^*(y) = \frac{1}{c} \text{ for } y = 1, \ldots, c, \tag{4}$$

which will be non-informative and thus allow us to avoid biasing clustering solutions[2]. Substituting π_y into $p^*(y)$, we can express Eq.(3) as

$$\frac{1}{2} \int \sum_{y=1}^{c} \frac{1}{\pi_y} p^*(y|\boldsymbol{x}) p^*(\boldsymbol{x}) p^*(y|\boldsymbol{x}) \mathrm{d}\boldsymbol{x} - \frac{1}{2}. \tag{5}$$

Let us approximate the class-posterior probability $p^*(y|\boldsymbol{x})$ by the following kernel model:

$$p(y|\boldsymbol{x}; \boldsymbol{\alpha}) := \sum_{i=1}^{n} \alpha_{y,i} K(\boldsymbol{x}, \boldsymbol{x}_i), \tag{6}$$

where $\boldsymbol{\alpha} = (\alpha_{1,1}, \ldots, \alpha_{c,n})^\top$ is the parameter vector, $^\top$ denotes the transpose, and $K(\boldsymbol{x}, \boldsymbol{x}')$ denotes a kernel function with a kernel parameter t. In the experiments, we will use a sparse variant of the *local-scaling kernel* (Zelnik-Manor & Perona, 2005):

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \begin{cases} \exp\left(-\dfrac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma_i \sigma_j}\right) & \text{if } \boldsymbol{x}_i \in \mathcal{N}_t(\boldsymbol{x}_j) \text{ or } \boldsymbol{x}_j \in \mathcal{N}_t(\boldsymbol{x}_i), \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

where $\mathcal{N}_t(\boldsymbol{x})$ denotes the set of t nearest neighbors for $\boldsymbol{x}$ (t is the kernel parameter), σ_i is a local scaling factor defined as $\sigma_i = \|\boldsymbol{x}_i - \boldsymbol{x}_i^{(t)}\|$, and $\boldsymbol{x}_i^{(t)}$ is the t-th nearest neighbor of $\boldsymbol{x}_i$.

Further approximating the expectation with respect to $p^*(\boldsymbol{x})$ included in Eq.(5) by the empirical average of samples $\{\boldsymbol{x}_i\}_{i=1}^{n}$, we arrive at the following SMI approximator:

$$\widehat{\mathrm{SMI}} := \frac{1}{2n} \sum_{y=1}^{c} \frac{1}{\pi_y} \boldsymbol{\alpha}_y^\top \boldsymbol{K}^2 \boldsymbol{\alpha}_y - \frac{1}{2}, \tag{8}$$

where $\boldsymbol{\alpha}_y := (\alpha_{y,1}, \ldots, \alpha_{y,n})^\top$ and $K_{i,j} := K(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

[2]Such a cluster-balance constraint is often employed in existing clustering algorithms (Shi & Malik, 2000; Xu et al., 2005; Niu et al., 2013).

For each cluster y, we maximize $\boldsymbol{\alpha}_y^\top \boldsymbol{K}^2 \boldsymbol{\alpha}_y$ under $\|\boldsymbol{\alpha}_y\| = 1$. Since this is the *Rayleigh quotient*, the maximizer is given by the normalized principal eigenvector of $\boldsymbol{K}$ (Horn & Johnson, 1985). To avoid all the solutions $\{\boldsymbol{\alpha}_y\}_{y=1}^c$ to be reduced to the same principal eigenvector, we impose their mutual orthogonality: $\boldsymbol{\alpha}_y^\top \boldsymbol{\alpha}_{y'} = 0$ for $y \neq y'$. Then the solutions are given by the normalized eigenvectors $\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_c$ associated with the eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n \geq 0$ of $\boldsymbol{K}$. Since the sign of $\boldsymbol{\phi}_y$ is arbitrary, we set the sign as

$$\widetilde{\boldsymbol{\phi}}_y = \boldsymbol{\phi}_y \times \mathrm{sign}(\boldsymbol{\phi}_y^\top \mathbf{1}_n),$$

where $\mathrm{sign}(\cdot)$ denotes the sign of a scalar and $\mathbf{1}_n$ denotes the n-dimensional vector with all ones.

On the other hand, since

$$p^*(y) = \int p^*(y|\boldsymbol{x}) p^*(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \approx \frac{1}{n} \sum_{i=1}^n p(y|\boldsymbol{x}_i; \boldsymbol{\alpha}) = \frac{1}{n} \boldsymbol{\alpha}_y^\top \boldsymbol{K} \mathbf{1}_n,$$

and the class-prior probability $p^*(y)$ was set to π_y for $y = 1, \ldots, c$, we have the following normalization condition:

$$\frac{1}{n} \boldsymbol{\alpha}_y^\top \boldsymbol{K} \mathbf{1}_n = \pi_y.$$

Furthermore, probability estimates should be non-negative, which can be achieved by rounding up negative outputs to zero.

Taking these normalization and non-negativity issues into account, cluster assignment y_i for $\boldsymbol{x}_i$ is determined as the maximizer of the approximation of $p(y|\boldsymbol{x}_i)$:

$$y_i = \operatorname*{argmax}_y \frac{[\max(\mathbf{0}_n, \boldsymbol{K}\widetilde{\boldsymbol{\phi}}_y)]_i}{(n\pi_y)^{-1} \max(\mathbf{0}_n, \boldsymbol{K}\widetilde{\boldsymbol{\phi}}_y)^\top \mathbf{1}_n} = \operatorname*{argmax}_y \frac{\pi_y [\max(\mathbf{0}_n, \widetilde{\boldsymbol{\phi}}_y)]_i}{\max(\mathbf{0}_n, \widetilde{\boldsymbol{\phi}}_y)^\top \mathbf{1}_n},$$

where $\mathbf{0}_n$ denotes the n-dimensional vector with all zeros, the max operation for vectors is applied in the element-wise manner, and $[\cdot]_i$ denotes the i-th element of a vector. Note that we used $\boldsymbol{K}\widetilde{\boldsymbol{\phi}}_y = \lambda_y \widetilde{\boldsymbol{\phi}}_y$ in the above derivation. For out-of-sample prediction, cluster assignment y' for new sample $\boldsymbol{x}'$ may be obtained as

$$y' := \operatorname*{argmax}_y \frac{\pi_y \max\left(0, \sum_{i=1}^n K(\boldsymbol{x}', \boldsymbol{x}_i)[\widetilde{\boldsymbol{\phi}}_y]_i\right)}{\lambda_y \max(\mathbf{0}_n, \widetilde{\boldsymbol{\phi}}_y)^\top \mathbf{1}_n}.$$

We call the above method *SMI-based clustering* (SMIC).

Discussions: Given an SMI approximator $\widehat{\mathrm{SMI}}$ defined by Eq.(8), a natural optimization criterion would be to impose non-negativity and normalization constraints on the parameter $\boldsymbol{\alpha}$. However, this results in a non-convex optimization problem and it is not straightforward to obtain the global optimal solution or even a good local solution without

any prior knowledge. For this reason, we decided to introduce the unit-norm constraint $\|\boldsymbol{\alpha}_y\| = 1$ on the parameter, which allows us to obtain the global optimal solution analytically even though the optimization problem is still non-convex. Although the introduction of the unit-norm constraint is a heuristic, this formulation has an advantage that we do not have to specify a good initial solution and it will be shown to work well in experiments in Section 4.

2.4 Kernel Parameter Choice by SMI Maximization

The solution of SMIC depends on the choice of the kernel parameter t included in the kernel function $K(\boldsymbol{x}, \boldsymbol{x}')$. Since SMIC was developed in the framework of SMI maximization, it would be natural to determine the kernel parameter t so as to maximize SMI. A direct approach is to use the SMI estimator $\widehat{\mathrm{SMI}}$ given by Eq.(8) also for kernel parameter choice. However, this direct approach is not favorable because $\widehat{\mathrm{SMI}}$ is an unsupervised SMI estimator (i.e., SMI is estimated only from unlabeled samples $\{\boldsymbol{x}_i\}_{i=1}^n$). On the other hand, in the model selection stage, we have already obtained labeled samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, and thus supervised estimation of SMI is possible. For supervised SMI estimation, a non-parametric SMI estimator called *least-squares mutual information* (LSMI) (Suzuki et al., 2009) was shown to achieve the optimal convergence rate. For this reason, we propose to use LSMI for model selection, instead of $\widehat{\mathrm{SMI}}$ (8).

LSMI is an estimator of SMI based on paired samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$. The key idea of LSMI is to learn the following *density-ratio function* (Sugiyama et al., 2012),

$$r^*(\boldsymbol{x}, y) := \frac{p^*(\boldsymbol{x}, y)}{p^*(\boldsymbol{x})p^*(y)}, \tag{9}$$

without going through density estimation of $p^*(\boldsymbol{x}, y)$, $p^*(\boldsymbol{x})$, and $p^*(y)$. More specifically, let us employ the following density-ratio model:

$$r(\boldsymbol{x}, y; \boldsymbol{\theta}) := \sum_{\ell: y_\ell = y} \theta_\ell L(\boldsymbol{x}, \boldsymbol{x}_\ell), \tag{10}$$

where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)^\top$ and $L(\boldsymbol{x}, \boldsymbol{x}')$ is a kernel function with a kernel parameter γ. In the experiments, we will use the Gaussian kernel:

$$L(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\gamma^2}\right), \tag{11}$$

where the Gaussian width γ is the kernel parameter.

The parameter $\boldsymbol{\theta}$ in the above density-ratio model is learned so that the following squared error is minimized:

$$\min_{\boldsymbol{\theta}} \frac{1}{2} \int \sum_{y=1}^{c} \Big(r(\boldsymbol{x}, y; \boldsymbol{\theta}) - r^*(\boldsymbol{x}, y)\Big)^2 p^*(\boldsymbol{x})p^*(y) \mathrm{d}\boldsymbol{x}. \tag{12}$$

Let $\boldsymbol{\theta}_y$ be the parameter vector corresponding to the kernel bases $\{L(\boldsymbol{x}, \boldsymbol{x}_\ell)\}_{\ell:y_\ell=y}$, i.e., $\boldsymbol{\theta}_y$ is the sub-vector of $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)^\top$ consisting of indices $\{\ell \mid y_\ell = y\}$. Let n_y be the length of $\boldsymbol{\theta}_y$, i.e., the number of samples in cluster y. Then an empirical and regularized version of the optimization problem (12) is given for each y as follows:

$$\min_{\boldsymbol{\theta}_y} \left[\frac{1}{2} \boldsymbol{\theta}_y^\top \widehat{\boldsymbol{H}}^{(y)} \boldsymbol{\theta}_y - \boldsymbol{\theta}_y^\top \widehat{\boldsymbol{h}}^{(y)} + \frac{\delta}{2} \boldsymbol{\theta}_y^\top \boldsymbol{\theta}_y \right], \tag{13}$$

where δ (≥ 0) is the regularization parameter. $\widehat{\boldsymbol{H}}^{(y)}$ is the $n_y \times n_y$ matrix and $\widehat{\boldsymbol{h}}^{(y)}$ is the n_y-dimensional vector defined as

$$\widehat{H}_{\ell,\ell'}^{(y)} := \frac{n_y}{n^2} \sum_{i=1}^{n} L(\boldsymbol{x}_i, \boldsymbol{x}_\ell^{(y)}) L(\boldsymbol{x}_i, \boldsymbol{x}_{\ell'}^{(y)}),$$
$$\widehat{h}_\ell^{(y)} := \frac{1}{n} \sum_{i:y_i=y} L(\boldsymbol{x}_i, \boldsymbol{x}_\ell^{(y)}),$$

where $\boldsymbol{x}_\ell^{(y)}$ is the ℓ-th sample in class y (which corresponds to $\widehat{\theta}_\ell^{(y)}$).

A notable advantage of LSMI is that the solution $\widehat{\boldsymbol{\theta}}^{(y)}$ can be computed analytically as

$$\widehat{\boldsymbol{\theta}}^{(y)} = (\widehat{\boldsymbol{H}}^{(y)} + \delta \boldsymbol{I})^{-1} \widehat{\boldsymbol{h}}^{(y)}.$$

Then a density-ratio estimator is obtained analytically as follows[3]:

$$\widehat{r}(\boldsymbol{x}, y) = \sum_{\ell=1}^{n_y} \widehat{\theta}_\ell^{(y)} L(\boldsymbol{x}, \boldsymbol{x}_\ell^{(y)}).$$

The accuracy of the above least-squares density-ratio estimator depends on the choice of the kernel parameter γ included in $L(\boldsymbol{x}, \boldsymbol{x}')$ and the regularization parameter δ in Eq.(13). Suzuki et al. (2009) showed that these tuning parameter values can be systematically optimized based on cross-validation as follows: First, the samples $\mathcal{Z} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ are divided into M disjoint subsets $\{\mathcal{Z}_m\}_{m=1}^M$ of approximately the same size (we use $M = 5$ in the experiments). Then a density-ratio estimator $\widehat{r}_m(\boldsymbol{x}, y)$ is obtained using

[3]Note that, in the original LSMI paper (Suzuki et al., 2009), the entire parameter $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)^\top$ for all classes was optimized at once. On the other hand, we found that, when the density-ratio model $r(\boldsymbol{x}, y; \boldsymbol{\theta})$ defined by Eq.(10) is used for SMI approximation, exactly the same solution as the original LSMI paper can be computed more efficiently by class-wise optimization. Indeed, in our preliminary experiments, we confirmed that our class-wise optimization significantly reduces the computation time compared with the original all-class optimization, with the same solution. Note that the original LSMI is applicable to more general setups such as regression, multi-label classification, and structured-output prediction. Thus, our speedup was brought by focusing on classification scenarios where Kronecker's delta function is used as the kernel for class labels in the density-ratio model (10).

$\mathcal{Z}\backslash\mathcal{Z}_m$ (i.e., all samples without $\mathcal{Z}_m$), and its out-of-sample error (which corresponds to Eq.(12) without irrelevant constant) for the hold-out samples $\mathcal{Z}_m$ is computed as

$$\mathrm{CV}_m := \frac{1}{2|\mathcal{Z}_m|^2}\sum_{\boldsymbol{x},y\in\mathcal{Z}_m}\widehat{r}_m(\boldsymbol{x},y)^2 - \frac{1}{|\mathcal{Z}_m|}\sum_{(\boldsymbol{x},y)\in\mathcal{Z}_m}\widehat{r}_m(\boldsymbol{x},y),$$

where $\sum_{\boldsymbol{x},y\in\mathcal{Z}_m}$ denotes the summation over all combinations of $\boldsymbol{x}$ and y in $\mathcal{Z}_m$ (and thus $|\mathcal{Z}_m|^2$ terms), while $\sum_{(\boldsymbol{x},y)\in\mathcal{Z}_m}$ denotes the summation over all pairs $(\boldsymbol{x},y)$ in $\mathcal{Z}_m$ (and thus $|\mathcal{Z}_m|$ terms). This procedure is repeated for $m=1,\ldots,M$, and the average of the above hold-out error over all m is computed as

$$\mathrm{CV} := \frac{1}{M}\sum_{m=1}^{M}\mathrm{CV}_m.$$

Then the kernel parameter γ and the regularization parameter δ that minimize the average hold-out error, CV, are chosen as the most suitable ones.

Finally, based on an expression of SMI (1),

$$\mathrm{SMI} = -\frac{1}{2}\int\sum_{y=1}^{c} r^*(\boldsymbol{x},y)^2 p^*(\boldsymbol{x})p^*(y)\mathrm{d}\boldsymbol{x} + \int\sum_{y=1}^{c} r^*(\boldsymbol{x},y)p^*(\boldsymbol{x},y)\mathrm{d}\boldsymbol{x} - \frac{1}{2},$$

the SMI estimator called LSMI is given as follows:

$$\mathrm{LSMI} := -\frac{1}{2n^2}\sum_{i,j=1}^{n}\widehat{r}(\boldsymbol{x}_i,y_j)^2 + \frac{1}{n}\sum_{i=1}^{n}\widehat{r}(\boldsymbol{x}_i,y_i) - \frac{1}{2}, \tag{14}$$

where $\widehat{r}(\boldsymbol{x},y)$ is a density-ratio estimator obtained above. Since $\widehat{r}(\boldsymbol{x},y)$ can be computed analytically, LSMI can also be computed analytically.

We use LSMI for model selection of SMIC. More specifically, we compute LSMI as a function of the kernel parameter t of $K(\boldsymbol{x},\boldsymbol{x}')$ included in the cluster-posterior model (6), and choose the one that maximizes LSMI. See Figure 1 for a schematic. A pseudo code of the entire SMI-maximization clustering procedure is summarized in Figures 2–4. Its MATLAB implementation is available from

"`http://sugiyama-www.cs.titech.ac.jp/~sugi/software/SMIC`".

Discussions: $\widehat{\mathrm{SMI}}$ given by Eq.(8) is used for determining cluster assignments $\{y_i\}_{i=1}^n$, while LSMI is used for model selection. Since LSMI was shown to be the optimal approximator of SMI, it would be more natural to use LSMI also for determining cluster assignments in a dependence-maximizing way (Song et al., 2007; Faivishevsky & Goldberger, 2010). However, this is not practical because maximizing LSMI with respect to cluster assignments $\{y_i\}_{i=1}^n$ is a hard optimization problem and a naive greedy-search strategy may not give a good solution without any prior knowledge. For this reason, we decided to use different criteria, $\widehat{\mathrm{SMI}}$ and LSMI, for determining cluster assignments and model selection. In principle, it is possible to use an arbitrary clustering algorithm in the first step and then evaluate its validity by LSMI in the second stage, although $\widehat{\mathrm{SMI}}$ and LSMI are "consistent" in the sense that they are both approximators of SMI.

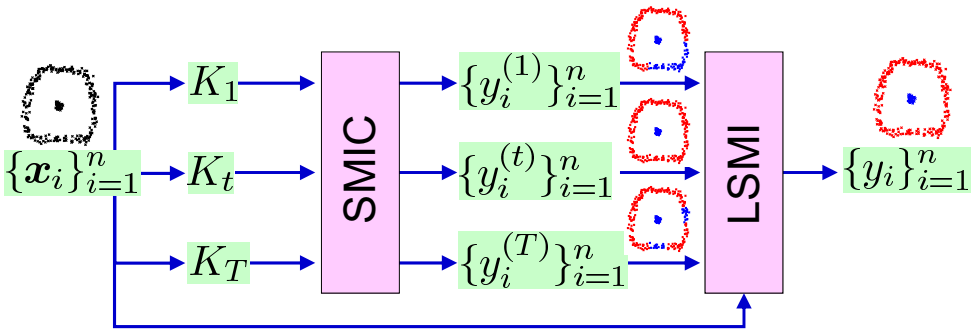


Figure 1: Schematic of the proposed clustering algorithm. We prepare T kernel candidates $\{K_t(\boldsymbol{x},\boldsymbol{x}')\}_{t=1}^T$, compute cluster assignments $\{y_i^{(t)}\}_{i=1,t=1}^{n,T}$ by SMIC, and choose the best one that maximizes LSMI.

Input: Feature vectors $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^n$ and the number c of clusters
Output: Cluster assignments $\mathcal{Y} = \{y_i\}_{i=1}^n$

For each kernel parameter candidate $t \in T$
  $\mathcal{Y}^{(t)} \longleftarrow \mathrm{SMIC}(\mathcal{X}, t, c)$;
  $\mathrm{LSMI}(t) \longleftarrow \mathrm{LSMI}(\mathcal{X}, \mathcal{Y}^{(t)})$;
end
$\widehat{t} \longleftarrow \operatorname{argmax}_{t \in T} \mathrm{LSMI}(t)$;
$\mathcal{Y} \longleftarrow \mathcal{Y}^{(\widehat{t})}$;

Figure 2: Pseudo code of information-maximization clustering based on SMIC and LSMI. The kernel parameter t refers to the tuning parameter included in the kernel function $K(\boldsymbol{x},\boldsymbol{x}')$ in the cluster-posterior model (6). Pseudo codes of SMIC and LSMI are described in Figure 3 and Figure 4, respectively.

2.5 Perturbation Stability Analysis

Here, we analyze the perturbation stability of the proposed clustering algorithm.

Let us denote the set of symmetric matrices of size n by $\mathbb{S}^n \subset \mathbb{R}^{n\times n}$, and the Frobenius norm of a matrix by $\|\cdot\|_{\mathrm{Frob}}$. For $\boldsymbol{A} \in \mathbb{S}^n$, we denote by $\lambda(\boldsymbol{A})$ the *spectra* of $\boldsymbol{A}$, i.e., the set of all eigenvalues of $\boldsymbol{A}$. For $\epsilon > 0$, a subset $\Lambda(\boldsymbol{A})$ of $\lambda(\boldsymbol{A})$ is said to be an ϵ-*cluster* of (the spectra of) $\boldsymbol{A}$, if the following two conditions are met:

1. $\Lambda(\boldsymbol{A})$ has a diameter smaller than ϵ.
2. $d_{\mathcal{H}}(\Lambda(\boldsymbol{A}), \lambda(\boldsymbol{A}) \setminus \Lambda(\boldsymbol{A})) > \epsilon$, where $d_{\mathcal{H}}$ is the Hausdorff distance.

First, we review a fundamental perturbation result given in the appendix of Koltchinskii (1998), Lemma 5.2 of Koltchinskii and Giné (2000), and pp.33–34 in von Luxburg (2004).

Input: Feature vectors $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^n$, kernel parameter t, and the number c of clusters
Output: Cluster assignments $\mathcal{Y} = \{y_i\}_{i=1}^n$

$\boldsymbol{K} \longleftarrow$ Kernel matrix for samples $\mathcal{X}$ and kernel parameter t;
$\boldsymbol{\phi}_y \longleftarrow$ y-th principal eigenvectors of $\boldsymbol{K}$ for $y = 1, \ldots, c$;
$\widetilde{\boldsymbol{\phi}}_y \longleftarrow \boldsymbol{\phi}_y \times \mathrm{sign}(\boldsymbol{\phi}_y^\top \mathbf{1}_n)$ for $y = 1, \ldots, c$;
$y_i \longleftarrow \underset{y \in \{1,\ldots,c\}}{\mathrm{argmax}} \dfrac{[\max(\mathbf{0}_n, \widetilde{\boldsymbol{\phi}}_y)]_i}{\max(\mathbf{0}_n, \widetilde{\boldsymbol{\phi}}_y)^\top \mathbf{1}_n}$ for $i = 1, \ldots, n$;
$\mathcal{Y} \longleftarrow \{y_i\}_{i=1}^n$;

Figure 3: Pseudo code of SMIC (with the uniform class-prior distribution). The kernel parameter t refers to the tuning parameter included in the kernel function $K(\boldsymbol{x}, \boldsymbol{x}')$ in the cluster-posterior model (6). If the class-prior probability $p^*(y)$ is set to a user-specified value π_y for $y = 1, \ldots, c$, y_i is determined as $\mathrm{argmax}_y \frac{\pi_y [\max(\mathbf{0}_n, \widetilde{\boldsymbol{\phi}}_y)]_i}{\max(\mathbf{0}_n, \widetilde{\boldsymbol{\phi}}_y)^\top \mathbf{1}_n}$.

Input: Feature vectors $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^n$ and cluster assignments $\mathcal{Y} = \{y_i\}_{i=1}^n$
Output: LSMI (an SMI estimate)

$\mathcal{Z} \longleftarrow \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$;
$\{\mathcal{Z}_m\}_{m=1}^M \longleftarrow M$ disjoint subsets of $\mathcal{Z}$;
For each kernel parameter candidate $\gamma \in \Gamma$
  For each regularization parameter candidate $\delta \in \Delta$
    For each fold $m = 1, \ldots, M$
      $\widehat{r}_{\gamma,\delta,m}(\boldsymbol{x}, y) \longleftarrow$ Density ratio estimator for (γ, δ) using $\mathcal{Z} \backslash \mathcal{Z}_m$;
      $\mathrm{CV}_m(\gamma, \delta) \longleftarrow$ Hold-out error of $\widehat{r}_{\gamma,\delta,m}(\boldsymbol{x}, y)$ for $\mathcal{Z}_m$;
    end
    $\mathrm{CV}(\gamma, \delta) \longleftarrow \dfrac{1}{M} \displaystyle\sum_{m=1}^M \mathrm{CV}_m(\gamma, \delta)$;
  end
end
$(\widehat{\gamma}, \widehat{\delta}) \longleftarrow \underset{\gamma \in \Gamma, \delta \in \Delta}{\mathrm{argmin}} \mathrm{CV}(\gamma, \delta)$;
$\widehat{r}(\boldsymbol{x}, y) \longleftarrow$ Density ratio estimator for $(\widehat{\gamma}, \widehat{\delta})$ using $\mathcal{Z}$;
$\mathrm{LSMI} \longleftarrow -\dfrac{1}{2n^2} \displaystyle\sum_{i,j=1}^n \widehat{r}(\boldsymbol{x}_i, y_j)^2 + \dfrac{1}{n} \sum_{i=1}^n \widehat{r}(\boldsymbol{x}_i, y_i) - \dfrac{1}{2}$;

Figure 4: Pseudo code of LSMI. The kernel parameter γ refers to the tuning parameter included in the kernel function $L(\boldsymbol{x}, \boldsymbol{x}')$ in the density-ratio model (10).

Proposition 1 (Finite-dimensional perturbation)**.** *For $\boldsymbol{A} \in \mathbb{S}^n$, let $\mu_1 > \cdots > \mu_k$ be the eigenvalues of $\boldsymbol{A}$ counted without multiplicity, and $W_1, \ldots, W_k$ be the corresponding eigenspaces. Let $\boldsymbol{P}_j(\boldsymbol{A})$ be the orthogonal projection onto W_j for $j = 1, \ldots, k$. For $1 \leq r < k$, define the eigengap*

$$\delta_r := \min_{j=1,\ldots,r} \{\mu_j - \mu_{j+1}\}.$$

Fix r, let $0 < \epsilon \leq \delta_r/4$, and assume perturbation $\boldsymbol{B} \in \mathbb{S}^n$ with $\|\boldsymbol{B}\|_{\mathrm{Frob}} < \epsilon$. Then,

1. *The spectra $\lambda(\boldsymbol{A} + \boldsymbol{B})$ of $(\boldsymbol{A} + \boldsymbol{B})$ can be partitioned into $r + 1$ subsets, i.e., r ϵ-clusters $\Lambda_j(\boldsymbol{A} + \boldsymbol{B})$ for $j = 1, \ldots, r$ and the residue R_r satisfy*

$$\Lambda_j(\boldsymbol{A} + \boldsymbol{B}) \subset \mathcal{B}(\mu_j, \epsilon), \tag{15}$$

   where $\mathcal{B}(\mu_j, \epsilon)$ denotes the open ball with center μ_j and radius ϵ, and

$$d_{\mathcal{H}}(R_r, \{\mu_1, \ldots, \mu_r\}) > \delta_r - \epsilon.$$

2. *Denote by $\boldsymbol{P}_j(\boldsymbol{A}+\boldsymbol{B})$ the orthogonal projection onto the direct sum of the eigenspaces of $(\boldsymbol{A} + \boldsymbol{B})$ with eigenvalues in the cluster $\Lambda_j(\boldsymbol{A} + \boldsymbol{B})$ for $j = 1, \ldots, k$. For all $j = 1, \ldots, r$, we have*

$$\mathrm{tr}\left(\boldsymbol{P}_j(\boldsymbol{A} + \boldsymbol{B})\right) = \mathrm{tr}\left(\boldsymbol{P}_j(\boldsymbol{A})\right) \tag{16}$$

   and

$$\|\boldsymbol{P}_j(\boldsymbol{A} + \boldsymbol{B}) - \boldsymbol{P}_j(\boldsymbol{A})\|_{\mathrm{Frob}} \leq 4\|\boldsymbol{B}\|_{\mathrm{Frob}}/\delta_r. \tag{17}$$

Intuitively speaking, Eq.(15) says that the perturbed eigenvalues are close to the original eigenvalues, Eq.(17) says that the perturbed eigenspaces are close to the original eigenspaces, and Eq.(16) guarantees the same dimensionality of the eigenspaces and thus the same multiplicity of perturbed and original eigenvalues, provided that the eigenvalues of $\boldsymbol{A}$ are well-separated, i.e., the eigengap δ_r is more than $4\|\boldsymbol{B}\|_{\mathrm{Frob}}$.

Now we apply the above result to SMIC. Recall that SMIC maximizes the objective function defined in Eq.(8),

$$\frac{1}{2n} \sum_{y=1}^{c} \frac{1}{\pi_y} \boldsymbol{\alpha}_y^\top \boldsymbol{K}^2 \boldsymbol{\alpha}_y - \frac{1}{2},$$

subject to the orthonormality of $\{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_c\}$. We can bound the difference between empirical and optimal solutions under a kernel matrix perturbation $\boldsymbol{\Delta} \in \mathbb{S}^n$ with $\|\boldsymbol{\Delta}\|_{\mathrm{Frob}} \ll \|\boldsymbol{K}\|_{\mathrm{Frob}}$ as follows:

Theorem 2 (Kernel matrix perturbation)**.** *Suppose that the kernel function satisfies $K(\boldsymbol{x}, \boldsymbol{x}) = 1$ for all $\boldsymbol{x}$. Let $\mu_1 > \cdots > \mu_r$ be the first r eigenvalues of the kernel matrix $\boldsymbol{K}$ counted without multiplicity, such that μ_r is the c-th largest eigenvalue of $\boldsymbol{K}$ if counted with multiplicity. Define the eigengap*

$$\delta_r = \min_{j=1,\ldots,r} \{\mu_j - \mu_{j+1}\}.$$

Assume that the kernel matrix $\boldsymbol{K}$ *is perturbed as*

$$\boldsymbol{K}' = \boldsymbol{K} + \boldsymbol{\Delta},$$

where $\boldsymbol{\Delta} \in \mathbb{S}^n$ *with* $\|\boldsymbol{\Delta}\|_{\mathrm{Frob}} < \delta_r/4$. *Denote by* v *and* $\{\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_c\}$ *the optimal value and solutions of SMIC for* $\boldsymbol{K}$, *and by* v' *the optimal value of SMIC for* $\boldsymbol{K}'$. *Then we have*

$$|v - v'| < \|\boldsymbol{\Delta}\|_{\mathrm{Frob}}/\pi_1, \tag{18}$$

and there exist optimal solutions $\{\boldsymbol{\phi}'_1, \ldots, \boldsymbol{\phi}'_c\}$ *for* $\boldsymbol{K}'$ *such that*

$$\|\boldsymbol{\phi}_y - \boldsymbol{\phi}'_y\|_2 \leq 4\|\boldsymbol{\Delta}\|_{\mathrm{Frob}}/\delta_r \text{ for } y = 1, \ldots, c, \tag{19}$$

where $\|\cdot\|_2$ *denotes the* ℓ_2*-norm.*

A proof of Theorem 2 is provided in Appendix A. This theorem shows that the difference in SMIC solutions is bounded by the amount of perturbation in the kernel matrix, which is a desirable property in practice. Note that, by "there exist optimal solutions $\{\boldsymbol{\phi}'_1, \ldots, \boldsymbol{\phi}'_c\}$", we mean that $\{\boldsymbol{\phi}'_1, \ldots, \boldsymbol{\phi}'_c\}$ need to be chosen carefully, since SMIC involves non-convex optimization and thus there may exist multiple globally optimal solutions. However, if $\boldsymbol{K}$ has c distinct top eigenvalues which would be a usual case in practice, it will be easy to determine $\boldsymbol{\phi}'_y$ because the only degree of freedom is its sign.

Next, we analyze the post-processing step of SMIC.

Theorem 3 (Post-processing perturbation)**.** *Under the same assumption as Theorem 2, suppose that* $\{\boldsymbol{\phi}'_1, \ldots, \boldsymbol{\phi}'_c\}$ *satisfy Eq.*(19). *Without loss of generality, we further assume that*

$$\mathbf{1}_n^\top \boldsymbol{\phi}_y > 0 \quad \text{and} \quad \mathbf{1}_n^\top \boldsymbol{\phi}'_y > 0 \text{ for } y = 1, \ldots, c.$$

Define the soft response vectors based on the solutions $\{\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_c\}$ *and* $\{\boldsymbol{\phi}'_1, \ldots, \boldsymbol{\phi}'_c\}$ *as*

$$\boldsymbol{f}_y = \pi_y \boldsymbol{\phi}_y^+ / (\mathbf{1}_n^\top \boldsymbol{\phi}_y^+) \quad \text{and} \quad \boldsymbol{f}'_y = \pi_y \boldsymbol{\phi}_y'^+ / (\mathbf{1}_n^\top \boldsymbol{\phi}_y'^+) \text{ for } y = 1, \ldots, c,$$

respectively, where $\boldsymbol{\phi}_y^+ = \max(\mathbf{0}_n, \boldsymbol{\phi}_y)$ *and* $\boldsymbol{\phi}_y'^+ = \max(\mathbf{0}_n, \boldsymbol{\phi}'_y)$. *Then, for* $y = 1, \ldots, c$, *we have*

$$\|\boldsymbol{f}_y - \boldsymbol{f}'_y\|_2/\sqrt{n} < 16\sqrt{2}\pi_y\|\boldsymbol{\Delta}\|_{\mathrm{Frob}}/\delta_r.$$

A proof of Theorem 3 is provided in Appendix B. This theorem shows that SMIC is stable with respect to kernel matrix perturbation $\boldsymbol{\Delta}$. That is, the root-mean-square error $\|\boldsymbol{f}_y - \boldsymbol{f}'_y\|_2/\sqrt{n}$ will vanish as $n \to \infty$, if the intensity of the perturbation measured by $\|\boldsymbol{\Delta}\|_{\mathrm{Frob}}/\delta_r$ is asymptotically an infinitesimal, i.e., $\|\boldsymbol{\Delta}\|_{\mathrm{Frob}}/\delta_r \in o(1)$ in terms of n.

3 Existing Clustering Methods

In this section, we review existing clustering methods and qualitatively discuss the relation to the proposed approach.

3.1 K-Means Clustering

K-means clustering (MacQueen, 1967) would be one of the most popular clustering algorithms. It tries to minimize the following distortion measure with respect to the cluster assignments $\{y_i\}_{i=1}^n$:

$$\sum_{y=1}^{c} \sum_{i:y_i=y} \|\boldsymbol{x}_i - \boldsymbol{\mu}_y\|^2, \tag{20}$$

where $\boldsymbol{\mu}_y := \frac{1}{n_y} \sum_{i:y_i=y} \boldsymbol{x}_i$ is the centroid of cluster y and n_y is the number of samples in cluster y.

The original k-means algorithm is capable of only producing linearly separated clusters (Duda et al., 2001). However, since samples are used only in terms of their inner products, its non-linear variant can be immediately obtained by performing k-means in a feature space induced by a reproducing kernel function (Girolami, 2002).

As the optimization problem of (kernel) k-means is NP-hard (Aloise et al., 2009), a greedy optimization algorithm is usually used for finding a local optimal solution in practice. It was shown that the solution to a continuously-relaxed variant of the kernel k-means problem is given by the principal components of the kernel matrix (Zha et al., 2002; Ding & He, 2004). Thus, post-discretization of the relaxed solution may give a good approximation to the original problem, which is computationally efficient. This idea is similar to the proposed SMIC method described in Section 2.3. However, an essential difference is that SMIC handles the continuous solution directly as a parameter estimate of the class-posterior model.

The performance of kernel k-means depends heavily on the choice of kernel functions, and there is no systematic way to determine the kernel function. This is a critical weakness of kernel k-means in practice. On the other hand, our proposed approach offers a natural model selection strategy, which is a significant advantage over kernel k-means.

3.2 Spectral Clustering

The basic idea of *spectral clustering* (Shi & Malik, 2000; Ng et al., 2002) is to first unfold non-linear data manifolds by a spectral embedding method, and then perform k-means in the embedded space. More specifically, given sample-sample similarity $W_{i,j} \geq 0$ (large $W_{i,j}$ means that $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are similar), embedded samples are obtained as the minimizer of the following criterion with respect to $\{\boldsymbol{\xi}_i\}_{i=1}^n$ under some normalization constraint:

$$\sum_{i,j}^{n} W_{i,j} \left\| \frac{1}{\sqrt{D_{i,i}}} \boldsymbol{\xi}_i - \frac{1}{\sqrt{D_{j,j}}} \boldsymbol{\xi}_j \right\|^2,$$

where $\boldsymbol{D}$ is the diagonal matrix with i-th diagonal element given by $D_{i,i} := \sum_{j=1}^n W_{i,j}$. Consequently, the embedded samples are given by the principal eigenvectors of $\boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{W} \boldsymbol{D}^{-\frac{1}{2}}$, followed by normalization. Note that spectral clustering was shown to

be equivalent to a weighted variant of kernel k-means with some specific kernel (Dhillon et al., 2004).

The performance of spectral clustering depends heavily on the choice of sample-sample similarity $W_{i,j}$. Zelnik-Manor and Perona (2005) proposed a useful unsupervised heuristic to determine the similarity in a data-dependent manner, called *local scaling*:

$$W_{i,j} = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma_i\sigma_j}\right),$$

where σ_i is a local scaling factor defined as

$$\sigma_i = \|\boldsymbol{x}_i - \boldsymbol{x}_i^{(t)}\|,$$

and $\boldsymbol{x}_i^{(t)}$ is the t-th nearest neighbor of $\boldsymbol{x}_i$. t is the tuning parameter in the local scaling similarity, and $t = 7$ was shown to be useful (Zelnik-Manor & Perona, 2005; Sugiyama, 2007). However, this magic number "7" does not seem to work always well in general.

If $\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{W}\boldsymbol{D}^{-\frac{1}{2}}$ is regarded as a kernel matrix, spectral clustering will be similar to the proposed SMIC method described in Section 2.3. However, SMIC does not require the post k-means processing since the principal components have clear interpretation as parameter estimates of the class-posterior model (6). Furthermore, our proposed approach provides a systematic model selection strategy, which is a notable advantage over spectral clustering.

3.3 Blurring Mean-Shift Clustering

Blurring mean-shift (Fukunaga & Hostetler, 1975) is a non-parametric clustering method based on the *modes* of the data-generating probability density.

In the blurring mean-shift algorithm, a kernel density estimator (Silverman, 1986) is used for modeling the data-generating probability density:

$$\widehat{p}(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} K\left(\|\boldsymbol{x} - \boldsymbol{x}_i\|^2/\sigma^2\right),$$

where $K(\xi)$ is a kernel function such as a Gaussian kernel $K(\xi) = e^{-\xi/2}$. Taking the derivative of $\widehat{p}(\boldsymbol{x})$ with respect to $\boldsymbol{x}$ and equating the derivative at $\boldsymbol{x} = \boldsymbol{x}_i$ to zero, we obtain the following updating formula for sample $\boldsymbol{x}_i$ $(i = 1, \ldots, n)$:

$$\boldsymbol{x}_i \longleftarrow \frac{\sum_{j=1}^{n} W_{i,j}\boldsymbol{x}_j}{\sum_{j'=1}^{n} W_{i,j'}},$$

where $W_{i,j} := K'\left(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/\sigma^2\right)$ and $K'(\xi)$ is the derivative of $K(\xi)$. Each mode of the density is regarded as a representative of a cluster, and each data point is assigned to the cluster which it converges to.

Carreira-Perpiñán (2007) showed that the blurring mean-shift algorithm can be interpreted as an *expectation-maximization algorithm* (Dempster et al., 1977), where

$W_{i,j}/(\sum_{j'=1}^{n} W_{i,j'})$ is regarded as the posterior probability of the i-th sample belonging to the j-th cluster. Furthermore, the above update rule can be expressed in a matrix form as $\boldsymbol{X} \longleftarrow \boldsymbol{X}\boldsymbol{P}$, where $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ is a sample matrix and $\boldsymbol{P} := \boldsymbol{W}\boldsymbol{D}^{-1}$ is a *stochastic matrix* of the random walk in a graph with adjacency $\boldsymbol{W}$ (Chung, 1997). $\boldsymbol{D}$ is defined as $D_{i,i} := \sum_{j=1}^{n} W_{i,j}$ and $D_{i,j} = 0$ for $i \neq j$. If $\boldsymbol{P}$ is independent of $\boldsymbol{X}$, the above iterative algorithm corresponds to the *power method* (Golub & Loan, 1989) for finding the leading left eigenvector of $\boldsymbol{P}$. Then, this algorithm is highly related to the spectral clustering which computes the principal eigenvectors of $\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{W}\boldsymbol{D}^{-\frac{1}{2}}$ (see Section 3.2). Although $\boldsymbol{P}$ depends on $\boldsymbol{X}$ in reality, Carreira-Perpiñán (2006) insisted that this analysis is still valid since $\boldsymbol{P}$ and $\boldsymbol{X}$ quickly reach a quasi-stable state.

An attractive property of blurring mean-shift is that the number of clusters is automatically determined as the number of modes in the probability density estimate. However, this choice depends on the kernel parameter σ and there is no systematic way to determine σ, which is restrictive compared with the proposed method. Another critical drawback of the blurring mean-shift algorithm is that it eventually converges to a single point (i.e., a single cluster, see Cheng (1995)), and therefore a sensible stopping criterion is necessary in practice. Although Carreira-Perpiñán (2006) gave a useful heuristic for stopping the iteration, it is not clear whether this heuristic always works well in practice.

3.4 Discriminative Clustering

The *support vector machine* (SVM) (Vapnik, 1995) is a supervised discriminative classifier that tries to find a hyperplane separating positive and negative samples with the maximum margin. Xu et al. (2005) extended SVM to unsupervised classification scenarios (i.e., clustering), which is called *maximum-margin clustering* (MMC).

MMC inherits the idea of SVM and tries to find the cluster assignments $\boldsymbol{y} = (y_1, \ldots, y_n)^\top$ so that the margin between two clusters is maximized under proper constraints:

$$\min_{\boldsymbol{y} \in \{+1,-1\}^n} \max_{\boldsymbol{\lambda}} \quad 2\boldsymbol{\lambda}^\top \mathbf{1}_n - \langle \boldsymbol{K} \circ \boldsymbol{\lambda}\boldsymbol{\lambda}^\top, \boldsymbol{y}\boldsymbol{y}^\top \rangle$$
$$\text{subject to} \quad -\varepsilon \leq \mathbf{1}_n^\top \boldsymbol{y} \leq \varepsilon \text{ and } \mathbf{0}_n \leq \boldsymbol{\lambda} \leq C\mathbf{1}_n,$$

where $\circ$ denotes the *Hadamard product* (also known as the entry-wise product), and ε and C are tuning parameters. The constraint $-\varepsilon \leq \mathbf{1}_n^\top \boldsymbol{y} \leq \varepsilon$ corresponds to balancing the cluster size.

Since the above optimization problem is combinatorial with respect to $\boldsymbol{y}$ and thus hard to solve directly, it is relaxed to a semi-definite program by replacing $\boldsymbol{y}\boldsymbol{y}^\top$ (which is a zero-one matrix with rank one) with a real positive semi-definite matrix (Xu et al., 2005). Since then, several approaches have been developed for further improving the computational efficiency of MMC (Valizadegan & Jin, 2007; Zhao et al., 2008; Zhang et al., 2009; Li et al., 2009; Wang et al., 2010).

The performance of MMC depends heavily on the choice of the tuning parameters ε and C, but there is no systematic method to tune these parameters. The fact that our

proposed approach is equipped with a model selection strategy would practically be a strong advantage over MMC.

Following a similar line to MMC, a *discriminative and flexible framework for clustering* (DIFFRAC) (Bach & Harchaoui, 2008) was proposed. DIFFRAC tries to solve a regularized least-squares problem with respect to a linear predictor and class labels. Thanks to the simple least-squares formulation, the parameters in the linear predictor can be optimized analytically, and thus the optimization problem is much simplified. A kernelized version of the DIFFRAC optimization problem is given by

$$\min_{\boldsymbol{y}\in\{+1,-1\}^n} \mathrm{tr}\left(\boldsymbol{\Pi}\boldsymbol{\Pi}^\top \kappa\boldsymbol{\Gamma}(\boldsymbol{\Gamma}\boldsymbol{K}\boldsymbol{\Gamma}+n\kappa\boldsymbol{I}_n)^{-1}\boldsymbol{\Gamma}\right),$$

where $\boldsymbol{\Pi}$ is the $n\times c$ cluster indicator matrix, which takes 1 only at one of the elements in each row (this corresponds to the index of the cluster to which the sample belongs) and others are all zeros. κ (≥ 0) is the regularization parameter, and $\boldsymbol{\Gamma} := \boldsymbol{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$ is a centering matrix. In practice, the above optimization problem is relaxed to a semi-definite program by replacing $\boldsymbol{\Pi}\boldsymbol{\Pi}^\top$ with a real positive semi-definite matrix. However, DIFFRAC is still computationally expensive and it suffers from lack of objective model selection strategies.

3.5 Generative Clustering

In the *generative clustering* framework (Duda et al., 2001), class labels are determined by

$$\widehat{y} = \operatorname*{argmax}_y p^*(y|\boldsymbol{x}) = \operatorname*{argmax}_y p^*(\boldsymbol{x},y),$$

where $p^*(y|\boldsymbol{x})$ is the class-posterior probability and $p^*(\boldsymbol{x},y)$ is the data-generating probability. Typically, $p^*(\boldsymbol{x},y)$ is modeled as

$$p(\boldsymbol{x},y;\boldsymbol{\beta},\boldsymbol{\pi}) = p(\boldsymbol{x}|y;\boldsymbol{\beta})p(y;\boldsymbol{\pi}),$$

where $\boldsymbol{\beta}$ and $\boldsymbol{\pi}$ are parameters. Canonical model choice is the Gaussian distribution for $p(\boldsymbol{x}|y;\boldsymbol{\beta})$ and the multinomial distribution for $p(y;\boldsymbol{\pi})$.

However, since class labels $\{y_i\}_{i=1}^n$ are unknown, one may not directly learn $\boldsymbol{\beta}$ and $\boldsymbol{\pi}$ in the joint-probability model $p(\boldsymbol{x},y;\boldsymbol{\beta},\boldsymbol{\pi})$. An approach to coping with this problem is to consider a *marginal* model,

$$p(\boldsymbol{x};\boldsymbol{\beta},\boldsymbol{\pi}) = \sum_{y=1}^{c} p(\boldsymbol{x}|y;\boldsymbol{\beta})p(y;\boldsymbol{\pi}),$$

and learns the parameters $\boldsymbol{\beta}$ and $\boldsymbol{\pi}$ by maximum likelihood estimation (Duda et al., 2001):

$$\max_{\boldsymbol{\beta},\boldsymbol{\pi}} \prod_{i=1}^{n} p(\boldsymbol{x}_i;\boldsymbol{\beta},\boldsymbol{\pi}).$$

Since the likelihood function of the above mixture model is non-convex, a *gradient method* (Amari, 1967) may be used for finding a local maximizer in practice. For determining the number of clusters (mixtures) and the mixing-element model $p(\boldsymbol{x}|y;\boldsymbol{\beta})$, *likelihood cross-validation* (Härdle et al., 2004) may be used.

Another approach to coping with the unavailability of class labels is to regard $\{y_i\}_{i=1}^n$ as *latent variables*, and apply the *expectation-maximization (EM) algorithm* (Dempster et al., 1977) for finding a local maximizer of the joint likelihood:

$$\max_{\boldsymbol{\beta},\boldsymbol{\pi}} \prod_{i=1}^{n} p(\boldsymbol{x}_i, y_i; \boldsymbol{\beta}, \boldsymbol{\pi}).$$

A more flexible variant of the EM algorithm called the *split-and-merge EM algorithm* (Ueda et al., 2000) is also available, which dynamically controls the number of clusters during the EM iteration.

Instead of point-estimating the parameters $\boldsymbol{\beta}$ and $\boldsymbol{\pi}$, one can also consider their distributions in the *Bayesian* framework (Bishop, 2006). Let us introduce prior distributions $p(\boldsymbol{\beta})$ and $p(\boldsymbol{\pi})$ for the parameters $\boldsymbol{\beta}$ and $\boldsymbol{\pi}$. Then the posterior distribution of the parameters is expressed as

$$p(\boldsymbol{\beta}, \boldsymbol{\pi}|\mathcal{X}) \propto p(\mathcal{X}|\boldsymbol{\beta}, \boldsymbol{\pi}) p(\boldsymbol{\beta}) p(\boldsymbol{\pi}),$$

where $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^n$. Based on the *Bayesian predictive distribution*,

$$\widehat{p}(y|\boldsymbol{x}, \mathcal{X}) \propto \iint p(\boldsymbol{x}, y|\boldsymbol{\beta}, \boldsymbol{\pi}) p(\boldsymbol{\beta}, \boldsymbol{\pi}|\mathcal{X}) \mathrm{d}\boldsymbol{\beta} \mathrm{d}\boldsymbol{\pi},$$

class labels are determined as

$$\max_{y} \widehat{p}(y|\boldsymbol{x}, \mathcal{X}).$$

Because the integration included in the Bayesian predictive distribution is computationally expensive, *conjugate priors* are often adopted in practice. For example, for the Gaussian-cluster model $p(\boldsymbol{x}|y;\boldsymbol{\beta})$, the Gaussian prior is assumed for the mean parameter and the Wishart prior is assumed for the precision parameter (i.e., the inverse covariance) for the multinomial model $p(y;\boldsymbol{\pi})$, the Dirichlet prior is assumed. Otherwise, the posterior distribution is approximated by the *Laplace approximation* (MacKay, 2003), the *Markov chain Monte Carlo sampling* (Andrieu et al., 2003), or the *variational approximation* (Attias, 2000; Ghahramani & Beal, 2000). The number of clusters can be determined based on the maximization of the *marginal likelihood*:

$$p(\mathcal{X}) = \operatorname*{argmax}_{y} \iint p(\mathcal{X}|\boldsymbol{\beta}, \boldsymbol{\pi}) p(\boldsymbol{\beta}) p(\boldsymbol{\pi}) \mathrm{d}\boldsymbol{\beta} \mathrm{d}\boldsymbol{\pi}. \tag{21}$$

The generative clustering methods are statistically well-founded. However, density models for each cluster $p^*(\boldsymbol{x}|y)$ need to be specified in advance, which lacks flexibility in practice. Furthermore, in the Bayesian approach, the choice of cluster models and prior distributions are often limited to conjugate pairs in practice. On the other hand, in the frequentist approach, only local solutions can be obtained in practice due to the non-convexity caused by mixture modeling.

3.6 Posterior-Maximization Clustering

Another possible clustering approach based on probabilistic inference is to directly maximizes the posterior probability of class labels $\mathcal{Y} = \{y_i\}_{i=1}^n$ (Bishop, 2006):

$$\max_{\mathcal{Y}} p^*(\mathcal{Y}|\mathcal{X}).$$

Let us model the cluster-wise data distribution $p^*(\mathcal{X}|\mathcal{Y})$ by $p(\mathcal{X}|\mathcal{Y}, \boldsymbol{\beta})$.

An approximate inference method called *iterative conditional modes* (Kurihara & Welling, 2009) alternatively maximizes the posterior probabilities of $\mathcal{Y}$ and $\boldsymbol{\beta}$ until convergence:

$$\widehat{\mathcal{Y}} \longleftarrow p(\mathcal{Y}|\mathcal{X}, \widehat{\boldsymbol{\beta}}),$$
$$\widehat{\boldsymbol{\beta}} \longleftarrow p(\boldsymbol{\beta}|\mathcal{X}, \widehat{\mathcal{Y}}).$$

When the Gaussian model with covariance identity is assumed for $p(\mathcal{Y}|\mathcal{X}, \boldsymbol{\beta})$, this algorithm is reduced to the k-means algorithm (see Section 3.1) under the uniform priors.

Let us consider the class-prior probability $p^*(\mathcal{Y})$ and model it by $p(\mathcal{Y}|\boldsymbol{\pi})$. Introducing the prior distributions $p(\boldsymbol{\beta})$ and $p(\boldsymbol{\pi})$, we can approximate the posterior distribution of $\mathcal{Y}$ as

$$p(\mathcal{Y}|\mathcal{X}) \propto \iint p(\mathcal{X}|\mathcal{Y}, \boldsymbol{\beta}) p(\boldsymbol{\beta}) p(\mathcal{Y}|\boldsymbol{\pi}) p(\boldsymbol{\pi}) \mathrm{d}\boldsymbol{\beta} \mathrm{d}\boldsymbol{\pi}.$$

Similarly to generative clustering described in Section 3.5, conjugate priors such as the Gauss-Wishart prior and the Dirichlet prior are practically useful in improving the computational efficiency. The number of clusters can also be similarly determined by maximizing the marginal likelihood (21). However, direct optimization of $\mathcal{Y}$ is often computationally intractable due to c^n combinations, where c is the number of clusters and n is the number of samples. For this reason, efficient sampling schemes such as the Markov chain Monte Carlo are indispensable in this approach.

A *Dirichlet process mixture* (Ferguson, 1973; Antoniak, 1974) is a non-parametric extension of the above approach, where an infinite number of clusters are implicitly considered and the number of clusters is automatically determined based on observed data. In order to improve the computational efficiency of this infinite mixture approach, various approximation schemes such as Markov chain Monte Carlo sampling (Neal, 2000) and variational approximation (Blei & Jordan, 2006) have been introduced. Furthermore, variants of Dirichlet processes such as hierarchical Dirichlet processes (Teh et al., 2007), nested Dirichlet processes (Rodríguez et al., 2008), and dependent Dirichlet processes (Lin et al., 2010) have been developed recently.

However, even in this non-parametric Bayesian approach, density models for each cluster still need to be parametrically specified in advance, which is often restricted to Gaussian models in practice. This highly limits the flexibility of clustering.

3.7 Dependence-Maximization Clustering

The *Hilbert-Schmidt independence criterion* (HSIC) (Gretton et al., 2005) is a dependence measure based on a reproducing kernel function $K(\boldsymbol{x}, \boldsymbol{x}')$ (Aronszajn, 1950). Song et al. (2007) proposed a *dependence-maximization clustering* method called *clustering with HSIC* (CLUHSIC), which tries to determine cluster assignments $\{y_i\}_{i=1}^n$ so that their dependence on feature vectors $\{\boldsymbol{x}_i\}_{i=1}^n$ is maximized.

More specifically, CLUHSIC tries to find the cluster indicator matrix $\boldsymbol{\Pi}$ (see Section 3.4) that maximizes

$$\mathrm{tr}\left(\boldsymbol{K}\boldsymbol{\Pi}\boldsymbol{A}\boldsymbol{\Pi}^\top\right),$$

where $K_{i,j} := K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $\boldsymbol{A}$ is a $c \times c$ cluster-cluster similarity matrix. Note that $\boldsymbol{\Pi}\boldsymbol{A}\boldsymbol{\Pi}^\top$ can be regarded as the kernel matrix for cluster assignments. Song et al. (2007) used a greedy algorithm to optimize the cluster indicator matrix, which is computationally demanding. Yang et al. (2010) gave spectral and semi-definite relaxation techniques to improve the computational efficiency of CLUHSIC.

HSIC is a kernel-based independence measure and the kernel function $K(\boldsymbol{x}, \boldsymbol{x}')$ needs to be determined in advance. However, there is no systematic model selection strategy for HSIC, and using the Gaussian kernel with width set to the median distance between samples is a standard heuristic in practice (Schölkopf & Smola, 2002). On the other hand, our proposed approach is equipped with an objective model selection strategy, which is a notable advantage over CLUHSIC.

Another line of dependence-maximization clustering adopts *mutual information* (MI) as a dependency measure. Recently, a dependence-maximization clustering method called *mean nearest-neighbor* (MNN) clustering was proposed (Faivishevsky & Goldberger, 2010). MNN is based on the k-nearest-neighbor entropy estimator proposed by Kozachenko and Leonenko (1987).

The performance of the original k-nearest-neighbor entropy estimator depends on the choice of the number of nearest neighbors, k. On the other hand, MNN avoids this problem by introducing a heuristic of taking an average over all possible k. The resulting objective function is given by

$$\sum_{y=1}^{c} \frac{1}{n_y - 1} \sum_{i \neq j : y_i = y_j = y} \log(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 + \epsilon), \tag{22}$$

where ϵ (> 0) is a smoothing parameter. Then this objective function is minimized with respect to cluster assignments $\{y_i\}_{i=1}^n$ using a greedy algorithm.

Although the fact that the tuning parameter k is averaged out is convenient, this heuristic is not well justified theoretically. Moreover, the choice of the smoothing parameter ϵ is arbitrary. In the MATLAB code provided by one of the authors, $\epsilon = 1/n$ was recommended, but there seems no justification for this choice. Also, due to the greedy optimization scheme, MNN is computationally expensive. On the other hand, our proposed approach offers a well-justified model selection strategy, and the SMI-based clustering gives an analytic-form solution which can be computed efficiently.

3.8 Information-Maximization Clustering with Mutual Information

Finally, we review methods of information-maximization clustering based on *mutual information* (Agakov & Barber, 2006; Gomes et al., 2010), which belong to the same family of clustering algorithms as our proposed method.

Mutual information (MI) is defined and expressed as

$$\begin{aligned}\mathrm{MI} &:= \int \sum_{y=1}^{c} p^*(\boldsymbol{x}, y) \log \frac{p^*(\boldsymbol{x}, y)}{p^*(\boldsymbol{x}) p^*(y)} \mathrm{d}\boldsymbol{x} \\ &= \int \sum_{y=1}^{c} p^*(y|\boldsymbol{x}) p^*(\boldsymbol{x}) \log p^*(y|\boldsymbol{x}) \mathrm{d}\boldsymbol{x} - \int \sum_{y=1}^{c} p^*(y|\boldsymbol{x}) p^*(\boldsymbol{x}) \log p^*(y) \mathrm{d}\boldsymbol{x}. \end{aligned} \tag{23}$$

Let us approximate the class-posterior probability $p^*(y|\boldsymbol{x})$ by a conditional-probability model $p(y|\boldsymbol{x};\boldsymbol{\alpha})$ with parameter $\boldsymbol{\alpha}$. Then the marginal probability $p^*(y)$ can be approximated as

$$p^*(y) = \int p^*(y|\boldsymbol{x}) p^*(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \approx \frac{1}{n} \sum_{i=1}^{n} p(y|\boldsymbol{x}_i;\boldsymbol{\alpha}). \tag{24}$$

By further approximating the expectation with respect to $p^*(\boldsymbol{x})$ included in Eq.(23) by the empirical average of samples $\{\boldsymbol{x}_i\}_{i=1}^n$, the following MI estimator can be obtained (Agakov & Barber, 2006; Gomes et al., 2010):

$$\begin{aligned}\widehat{\mathrm{MI}} &:= \frac{1}{n} \sum_{i=1}^{n} \sum_{y=1}^{c} p(y|\boldsymbol{x}_i;\boldsymbol{\alpha}) \log p(y|\boldsymbol{x}_i;\boldsymbol{\alpha}) \\ &\quad - \sum_{y=1}^{c} \left(\frac{1}{n} \sum_{i=1}^{n} p(y|\boldsymbol{x}_i;\boldsymbol{\alpha}) \right) \log \left(\frac{1}{n} \sum_{j=1}^{n} p(y|\boldsymbol{x}_j;\boldsymbol{\alpha}) \right). \end{aligned} \tag{25}$$

In Agakov and Barber (2006), the Gaussian model,

$$p(y|\boldsymbol{x};\boldsymbol{\alpha}) \propto \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{c}_y\|^2}{2s_y^2} + b_y \right),$$

(or its kernelized version) is adopted, where $\boldsymbol{\alpha} = \{\boldsymbol{c}_y, s_y, b_y\}_{y=1}^c$ is the parameter. Then a local maximizer of $\widehat{\mathrm{MI}}$ with respect to the parameter $\boldsymbol{\alpha}$ is found by a gradient method. On the other hand, in Gomes et al. (2010), the logistic model

$$p(y|\boldsymbol{x};\boldsymbol{\alpha}) \propto \exp\left(\boldsymbol{\alpha}_y^\top \boldsymbol{x}\right), \tag{26}$$

(or its kernelized version) is adopted, where $\boldsymbol{\alpha} = \{\boldsymbol{\alpha}_y\}_{y=1}^c$ is the parameter. Then a local maximizer of $\widehat{\mathrm{MI}}$ with respect to the parameter $\boldsymbol{\alpha}$ is found by a quasi-Newton method.

Finally, cluster assignments $\{y_i\}_{i=1}^n$ are determined as

$$y_i = \operatorname*{argmax}_y p(y|\boldsymbol{x}_i; \widehat{\boldsymbol{\alpha}}),$$

where $\widehat{\boldsymbol{\alpha}}$ is a local maximizer of $\widehat{\mathrm{MI}}$. Below, we refer to the above method as *MI-based clustering* (MIC).

In the kernelized version of MIC, the user needs to determine parameters included in the kernel function such as the kernel width or the number of nearest neighbors. Agakov and Barber (2006) proposed to choose the kernel parameters so that $\widehat{\mathrm{MI}}$ (25) is maximized. Thus, cluster assignments and kernel parameters can be consistently determined under the common guidance of maximizing $\widehat{\mathrm{MI}}$. However, since $\widehat{\mathrm{MI}}$ is an unsupervised estimator of MI, it is not accurately enough; in the model selection stage, cluster labels $\{y_i\}_{i=1}^n$ are available and thus supervised estimation of MI is more favorable. Indeed, there exists a more powerful supervised MI estimator called *maximum-likelihood MI* (MLMI) (Suzuki et al., 2008), which was proved to achieve the optimal non-parametric convergence rate.

The derivation of MLMI follows a similar line to LSMI explained in Section 2.4, i.e., the density-ratio function (9) is learned. More specifically, the following density-ratio model $r(\boldsymbol{x}, y; \boldsymbol{\theta})$ is used:

$$r(\boldsymbol{x}, y; \boldsymbol{\theta}) := \sum_{\ell: y_i = y} \theta_\ell L(\boldsymbol{x}, \boldsymbol{x}_\ell),$$

where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)^\top$ and $L(\boldsymbol{x}, \boldsymbol{x}')$ is a kernel function with a kernel parameter γ. Then the parameter $\boldsymbol{\theta}$ is learned so that the Kullback-Leibler divergence from $p^*(\boldsymbol{x}, y)$ to $r(\boldsymbol{x}, y; \boldsymbol{\theta}) p^*(\boldsymbol{x}) p^*(y)$ is minimized[4]. An empirical version of the MLMI optimization problem is given as

$$\begin{aligned} &\max_{\boldsymbol{\theta}} \ \frac{1}{n} \sum_{i=1}^n \log r(\boldsymbol{x}_i, y_i; \boldsymbol{\theta}) \\ &\text{s.t.} \quad \frac{1}{n^2} \sum_{i,j=1}^n r(\boldsymbol{x}_i, y_j; \boldsymbol{\theta}) = 1 \quad \text{and} \quad \boldsymbol{\theta} \geq \mathbf{0}_n, \end{aligned}$$

where the inequality for vectors is applied in the element-wise manner. This is a convex optimization problem, and thus the global optimal solution $\widehat{\boldsymbol{\theta}}$, which tends to be sparse, can be easily obtained by, e.g., iteratively performing gradient ascent and projection (Sugiyama et al., 2008).

Then an MI estimator called MLMI is given as follows:

$$\mathrm{MLMI} := \frac{1}{n} \sum_{i=1}^n \log r(\boldsymbol{x}_i, y_i; \widehat{\boldsymbol{\theta}}).$$

The kernel parameter γ included in the kernel function $L(\boldsymbol{x}, \boldsymbol{x}')$ can be optimized by cross-validation, in the same way as LSMI (Suzuki et al., 2008).

[4]Note that $r(\boldsymbol{x}, y; \boldsymbol{\theta}) p^*(\boldsymbol{x}) p^*(y)$ can be regarded as a model of $p^*(\boldsymbol{x}, y)$.

4 Experiments

In this section, we experimentally evaluate the performance of the proposed and existing clustering methods.

4.1 Illustration

First, we illustrate the behavior of the proposed method using the following 4 artificial datasets with dimensionality $d = 2$ and sample size $n = 200$:

(a) Four Gaussian blobs: For the number of classes $c = 4$, samples in each class are drawn from the Gaussian distributions with mean $(2, 2)^\top$, $(-2, 2)^\top$, $(2, -2)^\top$, and $(-2, -2)^\top$ and covariance matrix $0.25\boldsymbol{I}_2$, respectively.

(b) Circle & Gaussian: For $c = 2$, samples in one class are drawn from the 2-dimensional standard normal distribution, and samples in the other class are equi-distantly located on the origin-centered circle with radius 5. Then noise following the origin-centered normal distribution with covariance matrix $0.01\boldsymbol{I}_2$ is added to each sample.

(c) Double spirals: For $c = 2$, the i-th sample in one class is given by $(\ell_i \cos(m_i), \ell_i \sin(m_i))^\top$, and the i-th sample in the other class is given by $(-\ell_i \cos(m_i), -\ell_i \sin(m_i))^\top$, where $\ell_i = 1 + 4(i-1)/n$ and $m_i = 3\pi(i-1)/n$. Then noise following the origin-centered normal distribution with covariance matrix $0.01\boldsymbol{I}_2$ is added to each sample.

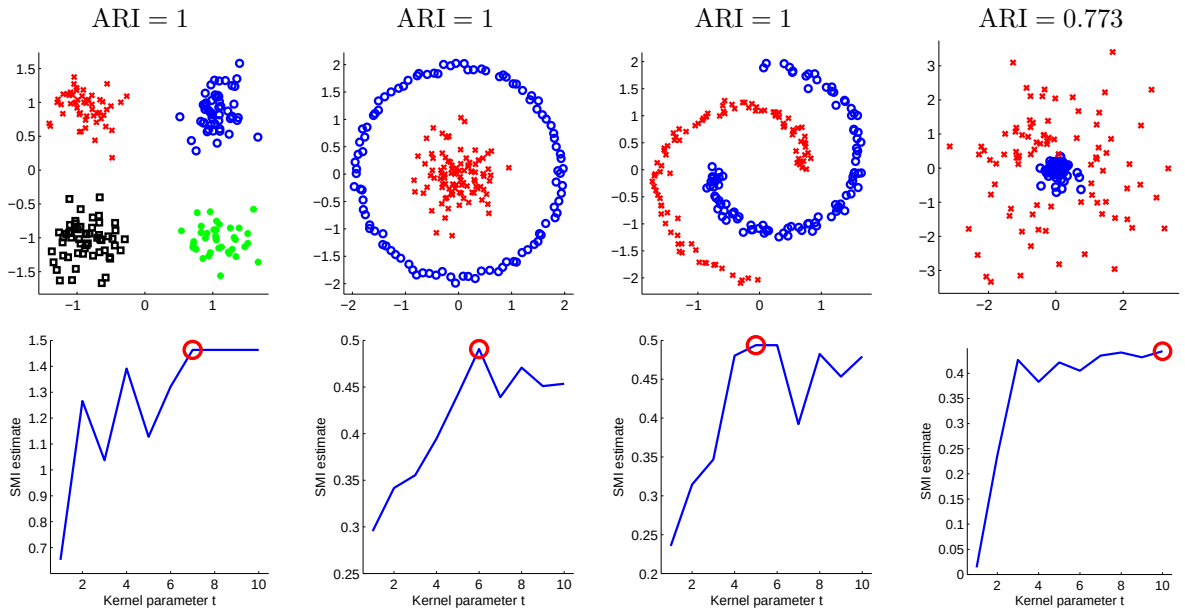
(d) High & low densities: For $c = 2$, samples in one class are drawn from the 2-dimensional standard normal distribution, and samples in the other class are drawn from the 2-dimensional origin-centered normal distribution with covariance matrix $0.01\boldsymbol{I}_2$.

The class-prior probability was set to be uniform. The generated samples were centralized and their variance was normalized in the dimension-wise manner (see the top row of Figure 5). A MATLAB code for generating these samples are available from

"`http://sugiyama-www.cs.titech.ac.jp/~sugi/software/SMIC`".

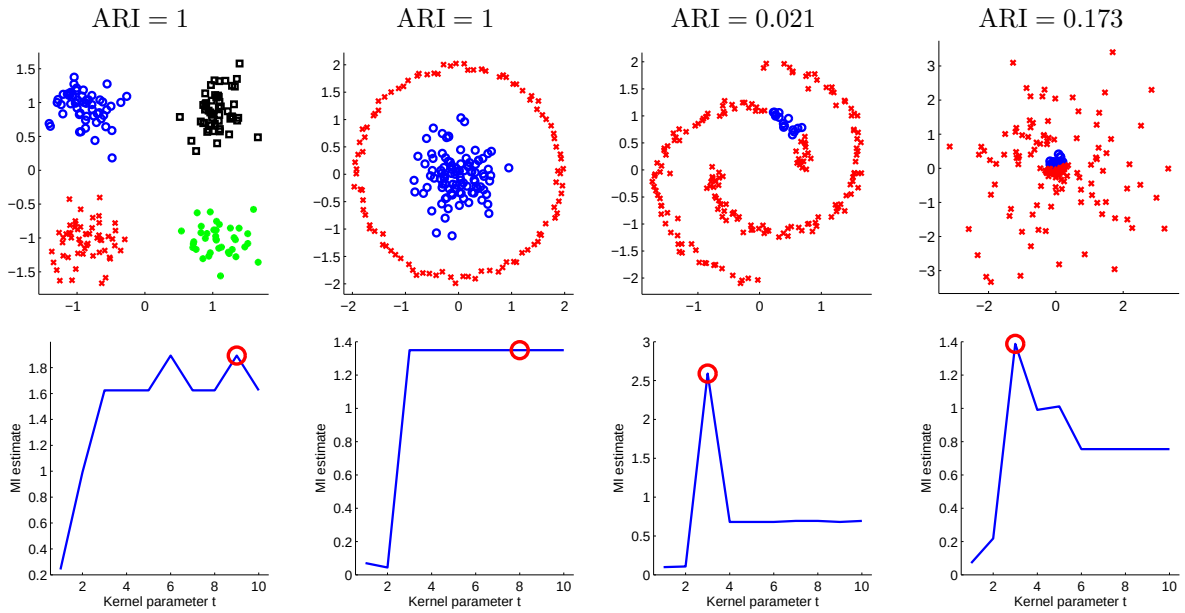
As a kernel function, we used the sparse local-scaling kernel (7) for SMIC, where the kernel parameter t was chosen from $\{1, \ldots, 10\}$ based on LSMI with the Gaussian kernel (11).

The top graphs in Figure 5 depict the cluster assignments obtained by SMIC with the uniform class-prior, and the bottom graphs in Figure 5 depict the model selection curves obtained by LSMI (i.e., the values of LSMI as functions of the model parameter t). The clustering performance was evaluated by the *adjusted Rand index* (ARI) (Hubert & Arabie, 1985) between inferred cluster assignments and the ground truth categories (see Appendix C for the details of ARI). Larger ARI values mean better performance, and



(a) Four Gaussian blobs (b) Circle & Gaussian (c) Double spirals (d) High & low densities

Figure 5: Illustrative examples. Cluster assignments obtained by SMIC (top) and model selection curves obtained by LSMI (bottom).



(a) Four Gaussian blobs (b) Circle & Gaussian (c) Double spirals (d) High & low densities

Figure 6: Illustrative examples. Cluster assignments obtained by MIC (top) and model selection curves obtained by MLMI (bottom).

ARI takes its maximum value 1 when two sets of cluster assignments are identical. The results show that SMIC combined with LSMI works well for these toy datasets.

Figure 6 depicts the cluster assignments and model selection curves obtained by MIC with MLMI (see Section 3.8), where pre-training of the kernel logistic model using the cluster assignments obtained by *self-tuning spectral clustering* (Zelnik-Manor & Perona, 2005) was carried out for initializing MIC (Gomes et al., 2010). The figure shows that qualitatively good clustering results were obtained for the datasets (a) and (b). However, for the datasets (c) and (d), poor results were obtained due to local optima of the objective function (25).

Figure 7 and Figure 8 depict class-posterior probabilities estimated by SMIC and MIC, respectively. The plots show that, for the datasets (a), (b), and (c) where the clusters are clearly separated, the estimated class-posterior probabilities are almost zero-one functions and thus the class prediction is highly certain. On the other hand, for the dataset (d) where the two clusters are overlapped, the estimated class-posterior probabilities tend to take intermediate class-posterior probabilities.

4.2 Influence of Imbalanced Class-Prior Probabilities

Next, we experimentally investigate how imbalanced class-prior probabilities (i.e., the sample size in each cluster is significantly different) influence the clustering performance of SMIC.

We continue using the 4 artificial datasets used in Section 4.1, but we set the true class-prior probability as

$$p^*(y=1)=p^*(y=2)=0.1, 0.15, 0.2, 0.25,$$
$$p^*(y=3)=p^*(y=4)=\frac{1-p^*(y=1)-p^*(y=2)}{2},$$

for the dataset (a), and

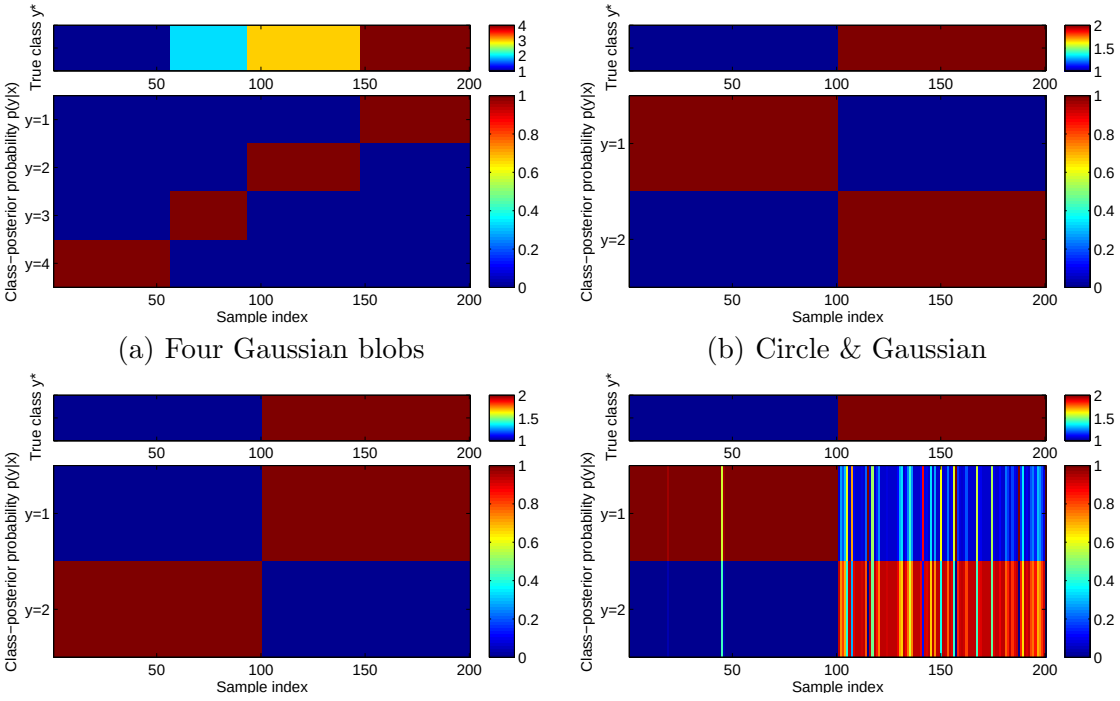
$$p^*(y=1)=0.2, 0.3, 0.4, 0.5,$$
$$p^*(y=2)=1-p^*(y=1),$$

for the datasets (b)–(d). The following 2 approaches are compared:

SMIC: SMIC with the uniform class-prior probabilities $\pi_1=\pi_2=1/2$.

SMIC*: SMIC with the true class-prior probabilities $\pi_1=p^*(y=1)$ and $\pi_2=p^*(y=2)$.

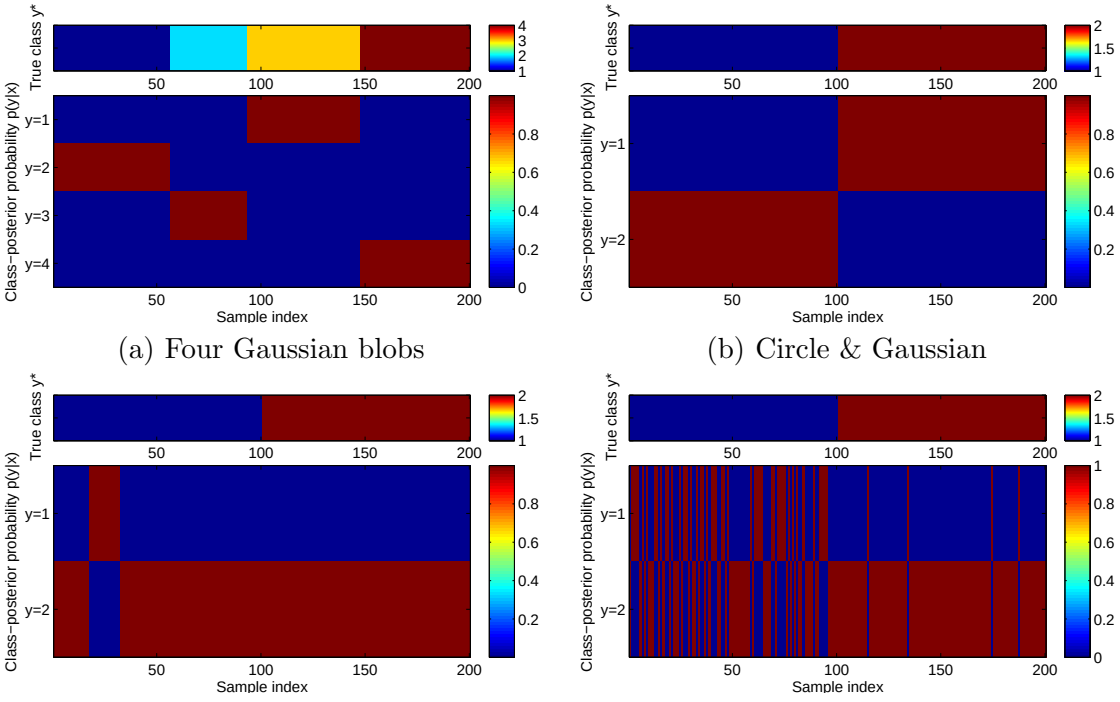
The mean and standard deviation of ARI over 100 runs are plotted in Figure 9, showing that the difference between SMIC and SMIC* is negligibly small. Indeed, the two methods were judged to be comparable to each other in terms of the average ARI by the *t-test* at the significance level 1% for all tested cases. This would be a natural result in clustering because class-prior probabilities only mildly affect cluster boundaries and such mild change in cluster boundaries do not significantly affect clustering solutions.



(a) Four Gaussian blobs (b) Circle & Gaussian

(c) Double spirals (d) High & low densities

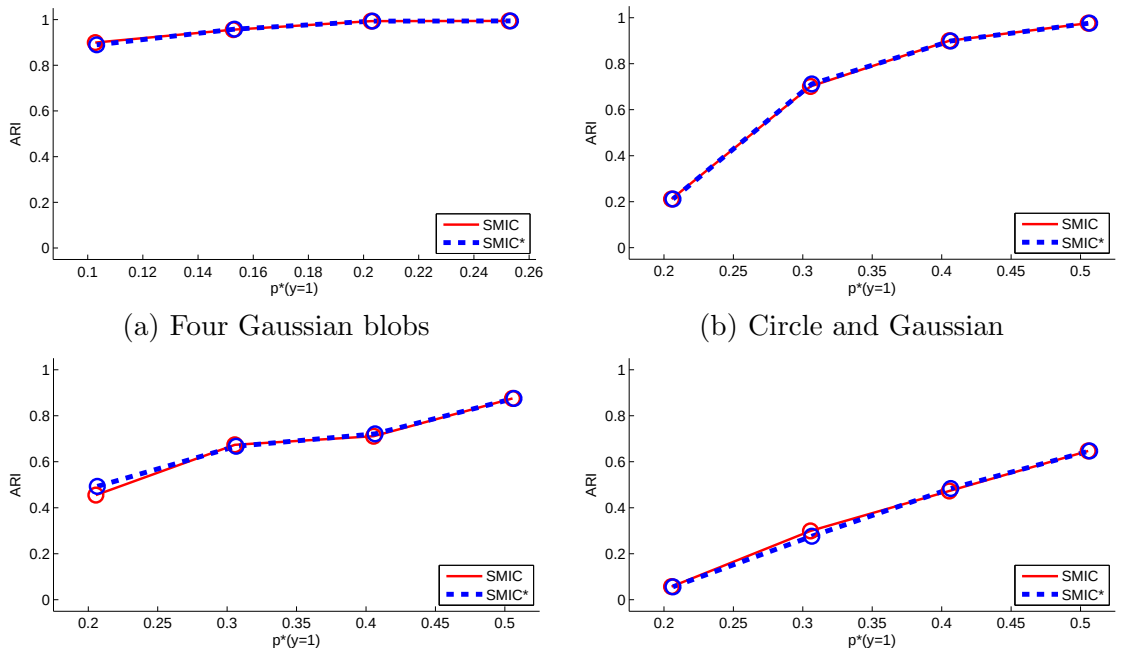
Figure 7: Illustrative examples. Class-posterior probabilities estimated by SMIC.



(a) Four Gaussian blobs (b) Circle & Gaussian

(c) Double spirals (d) High & low densities

Figure 8: Illustrative examples. Class-posterior probabilities estimated by MIC.



(a) Four Gaussian blobs

(b) Circle and Gaussian

(c) Double spirals

(d) High and low densities

Figure 9: Illustrative examples. The mean ARI over 100 runs as functions of the class-prior probability $p^*(y = 1)$. The two methods were judged to be comparable in terms of the average ARI by the *t-test* at the significance level 1%.

The above results imply that SMIC is not sensitive to the choice of class-prior probabilities. Thus, in practice, SMIC with the uniform class-prior distribution may be used when the true class-prior is unknown.

4.3 Performance Comparison

Finally, we systematically compare the performance of the proposed and existing clustering methods using various real-world datasets such as images, natural languages, accelerometric sensors, and speeches.

4.3.1 Setup

We compared the performance of the following methods, all of which do not contain open tuning parameters and therefore experimental results are fair and objective:

KM: K-means (MacQueen, 1967) (see also Section 3.1). We used the software included in the MATLAB Statistics Toolbox, where initial values were randomly generated 100 times and the best result in terms of the k-means objective value was chosen as the final solution.

SC1: Spectral clustering (Shi & Malik, 2000; Ng et al., 2002) (see also Section 3.2) with

the Gaussian similarity. The Gaussian width is set to the median distance between all samples, which is a popular heuristic in kernel methods (Schölkopf & Smola, 2002). We used the publicly available MATLAB code[5], where the post k-means processing was repeated 10 times with heuristic initialization: The first center was chosen randomly from samples, and then the next center was iteratively set to the farthest sample from the previous ones. The best result in terms of the k-means objective value over 10 repetitions was chosen as the final solution.

SC2: Spectral clustering with the self-tuning local-scaling similarity (Zelnik-Manor & Perona, 2005), instead of the Gaussian similarity.

MNN: Mean nearest-neighbor clustering (Faivishevsky & Goldberger, 2010) (see also Section 3.7). We used the MATLAB code provided by one of the authors[6]. Following the suggestions provided in the program code, the number of iterations was set to 10 and the smoothing parameter ϵ (see Eq.(22)) was set to $\epsilon = 1/n$.

MIC: MI-based clustering with kernel logistic models and the sparse local-scaling kernel (Gomes et al., 2010) (see also Section 3.8), where model selection is carried out by maximum-likelihood MI (MLMI) (Suzuki et al., 2008). We implemented this method using MATLAB, which is a combination of the MIC code personally provided by one of the authors, and the MLMI code available from the web page of one of the authors[7]. Following the suggestion provided in the original program code, MIC was initialized by pre-training of the kernel logistic model using the cluster assignments obtained by spectral clustering. The tuning parameter t included in the sparse local-scaling kernel (7) was chosen from $\{1, \ldots, 10\}$ based on MLMI with Gaussian kernels (see Section 3.8). The Gaussian kernel width in MLMI was chosen from $\{10^{-2}, 10^{-1.5}, 10^{-1}, \ldots, 10^{2}\}$ based on cross-validation. As suggested in the MLMI code provided by the author, the number of kernel bases in MLMI was limited to 200, which were randomly chosen from all n kernels.

SMIC: SMI-based clustering with the sparse local-scaling kernel and the uniform class-prior distribution (see Section 2.3), where model selection is carried out by least-squares MI (LSMI) (Suzuki et al., 2009) (see also Section 2.4). We implemented SMIC and LSMI using MATLAB by ourselves. The tuning parameter t included in the sparse local-scaling kernel (7) was chosen from $\{1, \ldots, 10\}$ based on LSMI with Gaussian kernels (see Section 2.4). The Gaussian kernel width and regularization parameter included in LSMI were chosen from $\{10^{-2}, 10^{-1.5}, 10^{-1}, \ldots, 10^{2}\}$ and $\{10^{-3}, 10^{-2.5}, 10^{-2}, \ldots, 10^{1}\}$, respectively, based on cross-validation. Similarly to MLMI, the number of kernel bases in LSMI was limited to 200, which were randomly chosen from all n kernels.

[5] `http://webee.technion.ac.il/~lihi/Demos/SelfTuningClustering.html`

[6] `http://www.levfaivishevsky.webs.com/NIC.rar`

[7] `http://sugiyama-www.cs.titech.ac.jp/~sugi/software/MLMI/index.html`

In addition to the clustering quality in terms of ARI, we also evaluated the computational efficiency of each method by the CPU computation time.

4.3.2 Datasets

We used the following 6 real-world datasets.

Digit ($d = 256, n = 5000,$ and $c = 10$)**:** The *USPS* hand-written digit dataset[8], which contains 9298 digit images. Each image consists of 256 ($= 16 \times 16$) pixels and represents a digit in $\{0, 1, 2, \ldots, 9\}$. Each pixel takes a value in $[-1, +1]$ corresponding to the intensity level in gray-scale. We randomly chose 500 samples from each of the 10 classes, and used 5000 samples in total.

Face ($d = 4096, n = 100,$ and $c = 10$)**:** The *Olivetti Face* dataset[9], which contains 400 gray-scale face images (40 people; 10 images per person). Each image consists of 4096 ($= 64 \times 64$) pixels and each pixel takes an integer value between 0 and 255 as the intensity level. We randomly chose 10 people, and used 100 samples in total.

Document ($d = 50, n = 700,$ and $c = 7$)**:** The *20-Newsgroups* dataset[10], which contains 20000 newsgroup documents across 20 different newsgroups. We merged the 20 newsgroups into the following 7 top-level categories: "*comp*", "*rec*", "*sci*", "*talk*", "*alt*", "*misc*", and "*soc*". Each document is expressed by a 10000-dimensional *bag-of-words* vector of *term-frequencies*. Following the convention (Joachims, 2002), we transformed the term-frequency vectors to the *term frequency/inverse document frequency* (TFIDF) vector, i.e., we multiplied the term-frequency by the logarithm of the inverse ratio of the documents containing the corresponding word. We randomly chose 100 samples from each of the 7 classes, and used 700 samples in total. We applied *principal component analysis* (PCA) (Pearson, 1901; Jolliffe, 1986) to the 700 samples, and extracted 50-dimensional feature vectors.

Word ($d = 50, n = 300,$ and $c = 3$)**:** The *SENSEVAL-2* dataset[11] for word-sense disambiguation. We took the noun "*interest*" appeared in 1930 contexts, having 3 different meanings: "advantage, advancement or favor", "a share in a company or business", and "money paid for the use of money" (i.e., 3 classes). From each surrounding context, we extracted a 14936-dimensional feature vector (Niu et al., 2005), which includes three types of features: *part-of-speech* of neighboring words with position information, *bag-of-words* in the surrounding context, and *local collocation* (Lee & Ng, 2002). We randomly chose 100 samples from each of the 3 classes, and used 300 samples in total. We applied PCA to the 300 samples, and extracted 50-dimensional feature vectors.

[8] `http://www.gaussianprocess.org/gpml/data/`
[9] `http://www.cs.toronto.edu/~roweis/data.html`
[10] `http://people.csail.mit.edu/jrennie/20Newsgroups/`
[11] `http://www.senseval.org/`

Accelerometry ($d = 5, n = 300,$ and $c = 3$)**:** The *ALKAN* dataset[12], which contains 3-axis (i.e., x-, y-, and z-axes) accelerometric data collected by the *iPod touch*. In the data collection procedure, subjects were asked to perform three specific tasks: *walking*, *running*, and *standing up*. The duration of each task was arbitrary, and the sampling rate was 20Hz with small variations. Each data-stream was then segmented in a sliding window manner with window width 5 seconds and sliding step 1 second (Hachiya et al., 2012). Depending on subjects, the position and orientation of the accelerometer was arbitrary—held by hand or kept in a pocket or a bag. For this reason, we took the ℓ_2-norm of the 3-dimensional acceleration vector at each time step, and computed the following 5 orientation-invariant features from each window: *mean*, *standard deviation*, *fluctuation of amplitude*, *average energy*, and *frequency-domain entropy* (Bao & Intille, 2004; Bharatula et al., 2005). We randomly chose 100 samples from each of the 3 classes, and used 300 samples in total.

Speech ($d = 50, n = 400,$ and $c = 2$)**:** An in-house speech dataset, which contains short utterance samples recorded from 2 male subjects speaking in French with sampling rate 44.1kHz. From each utterance sample, we extracted a 50-dimensional *line spectral frequencies* vector (Kain & Macon, 1988). We randomly chose 200 samples from each class, and used 400 samples in total.

For each dataset, the experiment was repeated 100 times with random choice of samples from the database, where the cluster size is balanced. Samples were centralized and their variance was normalized in the dimension-wise manner, before feeding them to clustering algorithms.

4.3.3 Results

The experimental results are described in Table 1. For the *digit* dataset, MIC and SMIC outperform KM, SC1, SC2, and MNN in terms of ARI. The entire computation time of SMIC including model selection is faster than the other methods. For the *face* dataset, SC2, MIC, and SMIC are comparable to each other and are better than KM, SC1, and MNN in terms of ARI. For the *document* and *word* datasets, SMIC tends to outperform the other methods. For the *accelerometry* dataset, MNN performs the best and SMIC follows. Finally, for the *speech* dataset, MIC and SMIC work comparably well, and are significantly better than the other methods.

The above results showed that MIC worked reasonably well, implying that the MLMI-based model selection strategy is practically useful. However, SMIC was shown to work even better than MIC, with much less computation time. The accuracy improvement of SMIC over MIC was gained by computing the SMIC solution in a closed-form without any heuristic initialization. The computational efficiency of SMIC was brought by the analytic computation of the optimal solution and the class-wise optimization of LSMI (see Section 2.4).

[12] `http://alkan.mns.kyutech.ac.jp/web/data.html`

Table 1: Experimental results on real-world datasets (with equal cluster size). The average clustering accuracy (and its standard deviation in the bracket) in terms of ARI and the average CPU computation time in second over 100 runs are described. Larger ARI is better, and shorter computation time is preferable. The best method in terms of the average ARI and methods judged to be comparable to the best one by the *t-test* at the significance level 1% are described in boldface. Computation time of MIC and SMIC corresponds to the time for computing a clustering solution after model selection has been carried out. For references, computation time for the entire procedure including model selection is described in the square bracket, which depends on the number of model candidates (in the current setup, we had 81 ($= 9 \times 9$) candidates).

Digit ($d = 256,\ n = 5000,$ and $c = 10$)

	KM	SC1	SC2	MNN	MIC	SMIC
ARI	0.42(0.01)	0.46(0.01)	0.24(0.02)	0.44(0.03)	**0.63(0.08)**	**0.63(0.05)**
Time	1414.6	561.3	495.1	228.4	69.1[1728.9]	7.1[144.1]

Face ($d = 4096,\ n = 100,$ and $c = 10$)

	KM	SC1	SC2	MNN	MIC	SMIC
ARI	0.60(0.11)	0.37(0.08)	**0.62(0.11)**	0.47(0.10)	**0.64(0.12)**	**0.65(0.12)**
Time	127.6	1.8	1.6	0.6	1.7[34.3]	0.0[14.9]

Document ($d = 50,\ n = 700,$ and $c = 7$)

	KM	SC1	SC2	MNN	MIC	SMIC
ARI	0.00(0.00)	0.00(0.00)	0.09(0.02)	0.09(0.02)	0.01(0.02)	**0.19(0.03)**
Time	28.5	9.9	11.1	4.5	9.7[226.9]	0.6[41.2]

Word ($d = 50,\ n = 300,$ and $c = 3$)

	KM	SC1	SC2	MNN	MIC	SMIC
ARI	0.04(0.05)	0.01(0.02)	0.02(0.01)	0.02(0.02)	0.04(0.04)	**0.08(0.05)**
Time	2.4	1.7	1.8	1.7	1.4[85.6]	0.3[36.7]

Accelerometry ($d = 5,\ n = 300,$ and $c = 3$)

	KM	SC1	SC2	MNN	MIC	SMIC
ARI	0.50(0.03)	0.20(0.26)	0.60(0.16)	**0.73(0.05)**	0.61(0.24)	0.68(0.12)
Time	0.2	1.7	1.7	1.8	1.3[137.2]	0.6[36.4]

Speech ($d = 50,\ n = 400,$ and $c = 2$)

	KM	SC1	SC2	MNN	MIC	SMIC
ARI	0.00(0.00)	0.00(0.00)	0.00(0.00)	0.04(0.15)	**0.18(0.16)**	**0.21(0.25)**
Time	0.4	2.1	1.9	1.8	1.3[134.3]	0.5[43.0]

Table 2: Experimental results on real-world datasets for different numbers of samples. ARI values are described in the table. The results for n are the same as the ones reported in Table 1.

Digit ($d = 256,\ n = 5000,$ and $c = 10$)

ARI	KM	SC1	SC2	MNN	MIC	SMIC
n	0.42(0.01)	0.46(0.01)	0.24(0.02)	0.44(0.03)	**0.63(0.08)**	**0.63(0.05)**
$n*3/4$	0.43(0.01)	0.47(0.01)	0.25(0.02)	0.45(0.03)	**0.64(0.09)**	**0.65(0.05)**
$n*1/2$	0.43(0.02)	0.47(0.01)	0.26(0.02)	0.44(0.04)	**0.61(0.12)**	**0.64(0.05)**
$n*1/4$	0.41(0.02)	0.45(0.02)	0.28(0.03)	0.43(0.04)	**0.60(0.10)**	**0.59(0.06)**

Face ($d = 4096,\ n = 100,$ and $c = 10$)

ARI	KM	SC1	SC2	MNN	MIC	SMIC
n	0.60(0.11)	0.37(0.08)	**0.62(0.11)**	0.47(0.10)	**0.64(0.12)**	**0.65(0.12)**
$n*3/4$	0.59(0.12)	0.29(0.07)	0.53(0.12)	0.41(0.11)	**0.62(0.12)**	**0.64(0.12)**
$n*1/2$	**0.60(0.14)**	0.17(0.08)	0.36(0.12)	0.26(0.11)	**0.55(0.12)**	**0.57(0.13)**

Document ($d = 50,\ n = 700,$ and $c = 7$)

ARI	KM	SC1	SC2	MNN	MIC	SMIC
n	0.00(0.00)	0.00(0.00)	0.09(0.02)	0.09(0.02)	0.01(0.02)	**0.19(0.03)**
$n*3/4$	0.00(0.00)	0.01(0.03)	0.10(0.02)	0.09(0.02)	0.01(0.02)	**0.20(0.03)**
$n*1/2$	0.00(0.00)	0.04(0.05)	0.10(0.02)	0.09(0.02)	0.02(0.03)	**0.19(0.03)**
$n*1/4$	0.00(0.00)	0.10(0.05)	0.11(0.03)	0.10(0.03)	0.03(0.04)	**0.19(0.05)**

Word ($d = 50,\ n = 300,$ and $c = 3$)

ARI	KM	SC1	SC2	MNN	MIC	SMIC
n	0.04(0.05)	0.01(0.02)	0.02(0.01)	0.02(0.02)	0.04(0.04)	**0.08(0.05)**
$n*3/4$	0.02(0.03)	0.00(0.01)	0.02(0.02)	0.02(0.02)	0.04(0.04)	**0.07(0.05)**
$n*1/2$	0.02(0.02)	0.00(0.00)	0.02(0.03)	0.02(0.02)	0.03(0.03)	**0.07(0.05)**
$n*1/4$	0.02(0.04)	-0.00(0.02)	0.02(0.03)	0.02(0.03)	**0.04(0.06)**	**0.05(0.05)**

Accelerometry ($d = 5,\ n = 300,$ and $c = 3$)

ARI	KM	SC1	SC2	MNN	MIC	SMIC
n	0.50(0.03)	0.20(0.26)	0.60(0.16)	**0.73(0.05)**	0.61(0.24)	0.68(0.12)
$n*3/4$	0.50(0.05)	0.25(0.29)	0.64(0.18)	**0.72(0.08)**	0.60(0.25)	**0.69(0.12)**
$n*1/2$	0.51(0.09)	0.33(0.30)	0.65(0.18)	**0.71(0.09)**	0.62(0.24)	**0.72(0.13)**
$n*1/4$	0.54(0.14)	0.56(0.21)	**0.65(0.18)**	0.66(0.14)	0.58(0.23)	**0.71(0.14)**

Speech ($d = 50,\ n = 400,$ and $c = 2$)

ARI	KM	SC1	SC2	MNN	MIC	SMIC
n	0.00(0.00)	0.00(0.00)	0.00(0.00)	0.04(0.15)	**0.18(0.16)**	**0.21(0.25)**
$n*3/4$	0.00(0.01)	0.00(0.01)	0.00(0.01)	0.01(0.09)	0.17(0.14)	**0.24(0.26)**
$n*1/2$	0.00(0.01)	0.01(0.01)	0.00(0.01)	0.01(0.05)	**0.13(0.11)**	**0.17(0.22)**
$n*1/4$	0.01(0.03)	0.01(0.02)	0.00(0.02)	0.02(0.07)	**0.12(0.12)**	**0.09(0.18)**

Table 3: Experimental results on real-world datasets under imbalanced setup. ARI values are described in the table. Class-imbalance was realized by setting the sample size of the first class m times larger than other classes. SMIC was computed with the uniform prior (i.e., the non-informative prior). The results for $m = 1$ are the same as the ones reported in Table 1.

Digit ($d = 256,\ n = 5000,$ and $c = 10$)

	KM	SC	MNN	MIC	SMIC
$m = 1$	0.42(0.01)	0.24(0.02)	0.44(0.03)	**0.63(0.08)**	**0.63(0.05)**
$m = 2$	0.52(0.01)	0.21(0.02)	0.43(0.04)	0.60(0.05)	**0.63(0.05)**

Document ($d = 50,\ n = 700,$ and $c = 7$)

	KM	SC	MNN	MIC	SMIC
$m = 1$	0.00(0.00)	0.09(0.02)	0.09(0.02)	0.01(0.02)	**0.19(0.03)**
$m = 2$	0.01(0.01)	0.10(0.03)	0.10(0.02)	0.01(0.02)	**0.19(0.04)**
$m = 3$	0.01(0.01)	0.10(0.03)	0.09(0.02)	-0.01(0.03)	**0.16(0.05)**
$m = 4$	0.02(0.01)	0.09(0.03)	0.08(0.02)	-0.00(0.04)	**0.14(0.05)**

Word ($d = 50,\ n = 300,$ and $c = 3$)

	KM	SC	MNN	MIC	SMIC
$m = 1$	0.04(0.05)	0.02(0.01)	0.02(0.02)	0.04(0.04)	**0.08(0.05)**
$m = 2$	0.00(0.07)	-0.01(0.01)	0.01(0.02)	-0.02(0.05)	**0.03(0.05)**

Accelerometry ($d = 5,\ n = 300,$ and $c = 3$)

	KM	SC	MNN	MIC	SMIC
$m = 1$	0.49(0.04)	0.58(0.14)	**0.71(0.05)**	0.57(0.23)	**0.68(0.12)**
$m = 2$	0.48(0.05)	0.54(0.14)	0.58(0.11)	0.49(0.19)	**0.69(0.16)**
$m = 3$	0.49(0.05)	0.47(0.10)	0.42(0.12)	0.42(0.14)	**0.66(0.20)**
$m = 4$	0.49(0.06)	0.38(0.11)	0.31(0.09)	0.40(0.18)	**0.56(0.22)**

The performance of MNN and SC2 was rather unstable because of the heuristic averaging of the number of nearest neighbors in MNN and the heuristic choice of local scaling in SC. In terms of computation time, they are relatively efficient for small- to medium-sized datasets, but they are expensive for the largest dataset, *digit*. SC1 did not perform as well as SC2, except for the *digit* dataset. KM was not reliable for the *document* and *speech* datasets because of the restriction that the cluster boundaries are linear. For the *digit*, *face*, and *document* datasets, KM was computationally very expensive since a large number of iterations were needed until convergence to a local optimum solution.

We also performed similar experiments with smaller numbers of samples. Table 2 describes the results, showing that the tendency of the experimental does not change significantly and the proposed SMIC still performs well.

Finally, we considered the imbalanced setup where the sample size of the first class

was set to be m times larger than other classes with the total number of samples fixed to the same number. The results are summarized in Table 3, showing that the performance of all methods tends to be degraded as the degree of cluster imbalance increases. This implies that clustering becomes more challenging if the cluster size is imbalanced. Among the compared methods, SMIC (with the uniform prior) still worked better than other methods.

Overall, the proposed SMIC combined with LSMI was shown to be a practically useful alternative to existing clustering approaches.

5 Conclusions

In this paper, we proposed a novel *information-maximization clustering* method that learns class-posterior probabilities in an unsupervised manner so that the *squared-loss mutual information* (SMI) between feature vectors and cluster assignments is maximized. The proposed algorithm, called *SMI-based clustering* (SMIC), allows us to obtain clustering solutions *analytically* by solving a kernel eigenvalue problem. Thus, unlike the previous information-maximization clustering methods (Agakov & Barber, 2006; Gomes et al., 2010), SMIC does not suffer from the problem of local optima. Furthermore, we proposed to use an optimal non-parametric SMI estimator called *least-squares mutual information* (LSMI) for data-driven parameter optimization. Through experiments, SMIC combined with LSMI was demonstrated to compare favorably with existing clustering methods.

In experiments, the proposed clustering method was shown to be useful for various types of data. However, the amount of improvement is large for some datasets, while it is mild for other datasets. It is thus practically important to gain more insights on in what case the proposed method is advantageous. Also, theoretically elucidating statistical consistency of the proposed method as well as investigating the perturbation stability in more details is also an important challenge. We will also analyze properties of other popular clustering algorithms within the framework of information-maximization clustering.

The sparse local-scaling kernel (7) was shown to be useful in experiments. Since this produces a sparse kernel matrix, the computation of SMIC (i.e., solving a kernel eigenvalue problem) can be carried out very efficiently. However, if model selection is taken into account, the proposed clustering procedure is still computationally rather demanding due to the repeated computation of LSMI, which requires to solve a system of linear equations. In the experiments, we used the Gaussian kernel (11) for LSMI and found it useful in practice. However, it produces a dense kernel matrix and thus a dense system of linear equations need to be solved, which is computationally expensive. If a sparse kernel is used also for LSMI, its computational efficiency will be highly improved. In our preliminary experiments, the use of the sparse local-scaling kernel for LSMI improved the computational efficiency, but it did not perform as well as the Gaussian kernel. Thus, our important future work is to find a sparse kernel that gives an accurate approximation of SMI with high computational efficiency.

As addressed in Song et al. (2007), kernelized methods can be applied to clustering of *non-vectorial structured objects* such as *strings*, *trees*, and *graphs* by employing kernel functions defined for such structured data (Lodhi et al., 2002; Duffy & Collins, 2002; Kashima & Koyanagi, 2002; Kondor & Lafferty, 2002; Kashima et al., 2003; Gärtner et al., 2003; Gärtner, 2003). Since these structured kernels usually contain tuning parameters, the performance of clustering methods without systematic model selection strategies depends on subjective parameter tuning, which is not preferable in practice. For Gaussian kernels, there exists a popular heuristic that the Gaussian width is set to the median distance between samples (Schölkopf & Smola, 2002). However, there seems no such common heuristic for structured kernels. In such scenarios, the proposed method will be highly advantageous because it allows systematic model selection for any kernels. We will explore this direction in our future work.

We experimentally showed that the proposed method with the uniform class-prior distribution still works well even when the true class-prior probability is not uniform. This is a useful property in practice since the true class-prior probability is often unknown. Another way to address this issue is to estimate the true class-prior probability in a data-driven fashion, for example, iteratively performing clustering and updating the class-prior probabilities. We will investigate such an adaptive approach in our future work.

The proposed method uses SMI as the common guidance for clustering, although we are using two SMI approximators: $\widehat{\mathrm{SMI}}$ defined by Eq.(8) for finding clustering solutions and LSMI defined by Eq.(14) for selecting models. Since $\widehat{\mathrm{SMI}}$ does not explicitly include cluster labels $\{y_i\}_{i=1}^n$, it has a simple form and therefore is suited for efficient maximization. Indeed, we can obtain an optimal solution analytically by solving an eigenvalue problem. However, since $\widehat{\mathrm{SMI}}$ is an unsupervised estimator where the cluster labels $\{y_i\}_{i=1}^n$ are not used, it may not be accurate enough for model selection purposes. Indeed, our preliminary experiments showed that the use of $\widehat{\mathrm{SMI}}$ is not appropriate as a model selection criterion. On the other hand, since LSMI achieves the optimal non-parametric convergence rate, its high accuracy is suitable for model selection purposes. However, LSMI explicitly requires cluster labels $\{y_i\}_{i=1}^n$ and thus is not suited for efficient maximization. Based on the optimality of LSMI, we ideally want to use LSMI consistently for *both* finding clustering solutions and selecting models. However, its optimization involves discrete optimization of $\{y_i\}_{i=1}^n$, which is cumbersome in practice. Our future challenge is to develop a practical clustering algorithm based directly on LSMI or alternative information measures.

Acknowledgments

We would like to thank Ryan Gomes for providing us his program code of information-maximization clustering. MS was supported by SCAT, AOARD, and MEXT Grant-in-Aid for Young Scientists (A) 25700022, GN was supported by the MEXT scholarship, MY and MK were supported by the JST PRESTO program, and HH was supported by the FIRST program.

A Proof of Theorem 2

For the kernel matrix $\boldsymbol{K}$, the optimal value v can be expressed as

$$v = \frac{1}{2n}\sum_{y=1}^{c}\frac{1}{\pi_y}\lambda_y^2(\boldsymbol{K}) - \frac{1}{2},$$

where $\pi_1 \leq \cdots \leq \pi_c$ are class-prior probabilities, $\lambda_1(\boldsymbol{K}) \geq \cdots \geq \lambda_c(\boldsymbol{K}) \geq 0$ are eigenvalues of $\boldsymbol{K}$, and the solutions $\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_c$ are given by the eigenvectors associated with $\lambda_1(\boldsymbol{K}), \ldots, \lambda_c(\boldsymbol{K})$. The optimal value v' and solutions $\boldsymbol{\phi}'_1, \ldots, \boldsymbol{\phi}'_c$ for $\boldsymbol{K}'$ can be characterized similarly. Then we have

$$\begin{aligned}
|v - v'| &= \frac{1}{2n}\left|\sum_{y=1}^{c}\frac{1}{\pi_y}\left(\lambda_y^2(\boldsymbol{K}) - \lambda_y^2(\boldsymbol{K}')\right)\right| \\
&= \frac{1}{2n}\left|\sum_{y=1}^{c}\frac{1}{\pi_y}\left(\lambda_y(\boldsymbol{K}) + \lambda_y(\boldsymbol{K}')\right)\left(\lambda_y(\boldsymbol{K}) - \lambda_y(\boldsymbol{K}')\right)\right| \\
&\leq \frac{1}{2n}\sum_{y=1}^{c}\frac{1}{\pi_y}\left(\lambda_y(\boldsymbol{K}) + \lambda_y(\boldsymbol{K}')\right)\left|\lambda_y(\boldsymbol{K}) - \lambda_y(\boldsymbol{K}')\right| \\
&\leq \frac{\|\boldsymbol{\Delta}\|_{\mathrm{Frob}}}{2n}\sum_{y=1}^{c}\frac{1}{\pi_y}\left(\lambda_y(\boldsymbol{K}) + \lambda_y(\boldsymbol{K}')\right) \\
&\leq \frac{\|\boldsymbol{\Delta}\|_{\mathrm{Frob}}}{2n\pi_1}\sum_{y=1}^{c}\left(\lambda_y(\boldsymbol{K}) + \lambda_y(\boldsymbol{K}')\right) \\
&= \frac{\|\boldsymbol{\Delta}\|_{\mathrm{Frob}}}{2n\pi_1}\left(\mathrm{tr}\left(\boldsymbol{K}\right) + \mathrm{tr}\left(\boldsymbol{K}'\right)\right) \\
&= \|\boldsymbol{\Delta}\|_{\mathrm{Frob}}/\pi_1,
\end{aligned}$$

where, in the third line, we used $|\lambda_y(\boldsymbol{K}) - \lambda_y(\boldsymbol{K}')| < \|\boldsymbol{\Delta}\|_{\mathrm{Frob}}$ implied by Eqs.(15) and (16), and we used in the last line $\mathrm{tr}\left(\boldsymbol{K}\right) = \mathrm{tr}\left(\boldsymbol{K}'\right) = n$ implied by the assumption $K(\boldsymbol{x}, \boldsymbol{x}) = 1$ for all $\boldsymbol{x}$. Thus, Eq.(18) was proved.

Eq.(19) is immediately implied by Eq.(17). More specifically, $\boldsymbol{\phi}'_y$ needs to be carefully chosen from the corresponding eigenspace of $\boldsymbol{K}'$ by minimizing the angle between $\boldsymbol{\phi}'_y$ and $\boldsymbol{\phi}_y$ (i.e., maximizing $\boldsymbol{\phi}_y^\top \boldsymbol{\phi}'_y$), since the optimal solution to SMIC is not necessarily unique. However, if $\boldsymbol{\phi}'_y$ is set to be the eigenvector associated to eigenvalue μ_j with multiplicity one, we only need to determine its sign. □

B Proof of Theorem 3

We use the following two lemmas in the proof of Theorem 3:

Lemma 4. *For $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^n$, we have*

$$\|\boldsymbol{\alpha} - \boldsymbol{\beta}\|_2^2 \geq \|\boldsymbol{\alpha}^+ - \boldsymbol{\beta}^+\|_2^2 + \|\boldsymbol{\alpha}^- - \boldsymbol{\beta}^-\|_2^2,$$

where $\boldsymbol{\alpha}^+ = \max(\mathbf{0}_n, \boldsymbol{\alpha})$ and $\boldsymbol{\alpha}^- = \min(\mathbf{0}_n, \boldsymbol{\alpha})$, and max *and* min *for vectors are computed in element-wise manners.*

Proof. Denote by α_i and β_i the i-th components of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, respectively. Then, for all i, we have

$$\begin{aligned}(\alpha_i - \beta_i)^2 &= (\alpha_i^+ - \beta_i^+)^2 + (\alpha_i^- - \beta_i^-)^2, \quad \text{if } \alpha_i\beta_i \geq 0,\\ (\alpha_i - \beta_i)^2 &> (\alpha_i^+ - \beta_i^+)^2 + (\alpha_i^- - \beta_i^-)^2, \quad \text{if } \alpha_i\beta_i < 0,\end{aligned}$$

which complete the proof. □

Lemma 5. *For $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^n$, we have*

$$\|\boldsymbol{\alpha}\boldsymbol{\beta}^\top - \boldsymbol{\beta}\boldsymbol{\alpha}^\top\|_{\mathrm{Frob}} \leq \sqrt{2}\|\boldsymbol{\alpha}\|_2\|\boldsymbol{\beta}\|_2.$$

Proof. By definition,

$$\begin{aligned}\|\boldsymbol{\alpha}\boldsymbol{\beta}^\top - \boldsymbol{\beta}\boldsymbol{\alpha}^\top\|_{\mathrm{Frob}}^2 &= \mathrm{tr}\left((\boldsymbol{\alpha}\boldsymbol{\beta}^\top - \boldsymbol{\beta}\boldsymbol{\alpha}^\top)^\top(\boldsymbol{\alpha}\boldsymbol{\beta}^\top - \boldsymbol{\beta}\boldsymbol{\alpha}^\top)\right)\\ &= \mathrm{tr}\left((\boldsymbol{\beta}\boldsymbol{\alpha}^\top - \boldsymbol{\alpha}\boldsymbol{\beta}^\top)(\boldsymbol{\alpha}\boldsymbol{\beta}^\top - \boldsymbol{\beta}\boldsymbol{\alpha}^\top)\right)\\ &= \mathrm{tr}\left(\boldsymbol{\beta}\boldsymbol{\alpha}^\top\boldsymbol{\alpha}\boldsymbol{\beta}^\top - \boldsymbol{\beta}\boldsymbol{\alpha}^\top\boldsymbol{\beta}\boldsymbol{\alpha}^\top - \boldsymbol{\alpha}\boldsymbol{\beta}^\top\boldsymbol{\alpha}\boldsymbol{\beta}^\top + \boldsymbol{\alpha}\boldsymbol{\beta}^\top\boldsymbol{\beta}\boldsymbol{\alpha}^\top\right)\\ &= \|\boldsymbol{\alpha}\|_2^2\mathrm{tr}\left(\boldsymbol{\beta}\boldsymbol{\beta}^\top\right) - \boldsymbol{\alpha}^\top\boldsymbol{\beta}\mathrm{tr}\left(\boldsymbol{\beta}\boldsymbol{\alpha}^\top\right) - \boldsymbol{\beta}^\top\boldsymbol{\alpha}\mathrm{tr}\left(\boldsymbol{\alpha}\boldsymbol{\beta}^\top\right) + \|\boldsymbol{\beta}\|_2^2\mathrm{tr}\left(\boldsymbol{\alpha}\boldsymbol{\alpha}^\top\right)\\ &= \|\boldsymbol{\alpha}\|_2^2\|\boldsymbol{\beta}\|_2^2 - (\boldsymbol{\alpha}^\top\boldsymbol{\beta})^2 - (\boldsymbol{\beta}^\top\boldsymbol{\alpha})^2 + \|\boldsymbol{\beta}\|_2^2\|\boldsymbol{\alpha}\|_2^2\\ &\leq 2\|\boldsymbol{\alpha}\|_2^2\|\boldsymbol{\beta}\|_2^2.\end{aligned}$$

The lemma follows by taking square roots of the beginning and the end of the above chain of equations. □

Using the above lemmas, we prove Theorem 3. First of all, we have

$$\boldsymbol{f}_y - \boldsymbol{f}'_y = \frac{\pi_y(\boldsymbol{\phi}_y^+\boldsymbol{\phi}_y'^{+\top} - \boldsymbol{\phi}_y'^+\boldsymbol{\phi}_y^{+\top})\mathbf{1}_n}{\mathbf{1}_n^\top\boldsymbol{\phi}_y^+ \cdot \mathbf{1}_n^\top\boldsymbol{\phi}_y'^+}.$$

Since

$$\|\boldsymbol{\phi}_y\|_1 = \|\boldsymbol{\phi}_y^+\|_1 + \|\boldsymbol{\phi}_y^-\|_1, \quad \mathbf{1}_n^\top\boldsymbol{\phi}_y = \|\boldsymbol{\phi}_y^+\|_1 - \|\boldsymbol{\phi}_y^-\|_1, \quad \text{and} \quad \mathbf{1}_n^\top\boldsymbol{\phi}_y > 0,$$

we can know that $\|\boldsymbol{\phi}_y^+\|_1 > \|\boldsymbol{\phi}_y\|_1/2$, and

$$\mathbf{1}_n^\top\boldsymbol{\phi}_y^+ = \|\boldsymbol{\phi}_y^+\|_1 > \|\boldsymbol{\phi}_y\|_1/2 > \|\boldsymbol{\phi}_y\|_2/2 = 1/2.$$

Similarly, $\mathbf{1}_n^\top\boldsymbol{\phi}_y'^+ > 1/2$ and thus it turns out that

$$(\mathbf{1}_n^\top\boldsymbol{\phi}_y^+ \cdot \mathbf{1}_n^\top\boldsymbol{\phi}_y'^+) > 1/4.$$

Table 4: Notation for Rand index and adjusted Rand index.

(a)

	$\mathcal{C}_1^*$	$\cdots$	$\mathcal{C}_c^*$	Sum
$\mathcal{C}_1$	$n_{1,1}$	$\cdots$	$n_{1,c}$	n_1
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$\mathcal{C}_c$	$n_{c,1}$	$\cdots$	$n_{c,c}$	n_c
Sum	n_1^*	$\cdots$	n_c^*	n

(b)

		Pairs in $\{\mathcal{C}_{y'}^*\}_{y'=1}^c$	
		Same	Different
Pairs in $\{\mathcal{C}_y\}_{y=1}^c$	Same	$m_{\mathcal{C},\mathcal{C}^*}$	$m_{\mathcal{C},\bar{\mathcal{C}}^*}$
	Different	$m_{\bar{\mathcal{C}},\mathcal{C}^*}$	$m_{\bar{\mathcal{C}},\bar{\mathcal{C}}^*}$

Next, let $\boldsymbol{\alpha} = \boldsymbol{\phi}_y^+$ and $\boldsymbol{\beta} = \boldsymbol{\phi}_y'^+ - \boldsymbol{\phi}_y^+$. Then it holds that

$$\boldsymbol{\phi}_y^+ \boldsymbol{\phi}_y'^{+\top} - \boldsymbol{\phi}_y'^+ \boldsymbol{\phi}_y^{+\top} = \boldsymbol{\alpha}(\boldsymbol{\alpha}+\boldsymbol{\beta})^\top - (\boldsymbol{\alpha}+\boldsymbol{\beta})\boldsymbol{\alpha}^\top = \boldsymbol{\alpha}\boldsymbol{\beta}^\top - \boldsymbol{\beta}\boldsymbol{\alpha}^\top.$$

Consequently, we have

$$\begin{aligned}\|\boldsymbol{f}_y - \boldsymbol{f}'_y\|_2 &< 4\pi_y \|(\boldsymbol{\alpha}\boldsymbol{\beta}^\top - \boldsymbol{\beta}\boldsymbol{\alpha}^\top)\mathbf{1}_n\|_2 \\ &< 4\pi_y \|\mathbf{1}_n\|_2 \|\boldsymbol{\alpha}\boldsymbol{\beta}^\top - \boldsymbol{\beta}\boldsymbol{\alpha}^\top\|_2 \\ &\leq 4\sqrt{n}\pi_y \|\boldsymbol{\alpha}\boldsymbol{\beta}^\top - \boldsymbol{\beta}\boldsymbol{\alpha}^\top\|_{\mathrm{Frob}},\end{aligned}$$

where $\|\cdot\|_2$ on $\mathbb{R}^{n\times n}$ means the operator norm induced by $\|\cdot\|_2$ on $\mathbb{R}^n$, and the last line is due to the fact that $\|\cdot\|_2$ is the ℓ_∞-norm of the spectra and $\|\cdot\|_{\mathrm{Frob}}$ is the ℓ_2-norm of the spectra. According to Lemma 5, it holds that

$$\begin{aligned}\|\boldsymbol{f}_y - \boldsymbol{f}'_y\|_2 &< 4\sqrt{n}\pi_y\sqrt{2}\|\boldsymbol{\alpha}\|_2\|\boldsymbol{\beta}\|_2 \\ &= 4\sqrt{2n}\pi_y \|\boldsymbol{\phi}_y^+\|_2 \|\boldsymbol{\phi}_y'^+ - \boldsymbol{\phi}_y^+\|_2 \\ &< 4\sqrt{2n}\pi_y \|\boldsymbol{\phi}_y\|_2 \|\boldsymbol{\phi}'_y - \boldsymbol{\phi}_y\|_2 \\ &\leq 16\sqrt{2n}\pi_y \|\boldsymbol{\Delta}\|_{\mathrm{Frob}}/\delta_r,\end{aligned}$$

where the third line is due to Lemma 4, and we used in the last line the facts that $\boldsymbol{\phi}_y$ is an eigenvector of $\boldsymbol{K}$ and $\boldsymbol{\phi}'_y$ satisfies Eq.(19). Finally, dividing the above inequality by $\sqrt{n}$ completes the proof. □

C Appendix: Rand Index and Adjusted Rand Index

Here, we review the definitions of the *Rand index* (RI) (Rand, 1971) and the *adjusted Rand index* (ARI) (Hubert & Arabie, 1985), which are used for evaluating the quality of clustering results. Let $\{y_i^*\}_{i=1}^n$ be the ground-truth cluster assignments, and let $\{y_i\}_{i=1}^n$ be a clustering solution obtained by some algorithm. The goal is to quantitatively evaluate the similarity between $\{y_i\}_{i=1}^n$ and $\{y_i^*\}_{i=1}^n$.

The most direct way to evaluate the discrepancy between $\{y_i\}_{i=1}^n$ and $\{y_i^*\}_{i=1}^n$ would be to naively verify the correctness of the predicted labels. However, in clustering, predicted

class labels $\{y_i\}_{i=1}^n$ do not have to be equal to the true labels $\{y_i^*\}_{i=1}^n$, but only their *partition* matters. The correctness of the partition may be evaluated by verifying the correctness of the predicted labels for all possible label permutations. However, this is computationally expensive if the number of classes is large. RI and ARI are alternative performance measures that can overcome this computational problem in a systematic way.

For the two partitions $\{y_i\}_{i=1}^n$ and $\{y_i^*\}_{i=1}^n$, let $\mathcal{C}_y$ and $\mathcal{C}_y^*$ $(y = 1, \ldots, c)$ be sets of indices of samples in cluster y, respectively:

$$\mathcal{C}_y = \{y_i \mid y_i = y\},$$
$$\mathcal{C}_y^* = \{y_i^* \mid y_i^* = y\}.$$

Let $n_{y,y'}$ be the number of samples that are assigned to the cluster $\mathcal{C}_y$ and the cluster $\mathcal{C}_{y'}^*$. Let n_y (resp. n_y^*) be the number of samples that are assigned to the cluster $\mathcal{C}_y$ (resp. $\mathcal{C}_{y'}^*$). The notation is summarized in Table 4(a).

Let $m_{\mathcal{C},\mathcal{C}^*}$, $m_{\mathcal{C},\bar{\mathcal{C}}^*}$, $m_{\bar{\mathcal{C}},\mathcal{C}^*}$, and $m_{\bar{\mathcal{C}},\bar{\mathcal{C}}^*}$ be defined as

$$m_{\mathcal{C},\mathcal{C}^*} := \sum_{y,y'=1}^{c} \binom{n_{y,y'}}{2},$$
$$m_{\mathcal{C},\bar{\mathcal{C}}^*} := \sum_{y=1}^{c} \binom{n_y}{2} - m_{\mathcal{C},\mathcal{C}^*},$$
$$m_{\bar{\mathcal{C}},\mathcal{C}^*} := \sum_{y'=1}^{c} \binom{n_{y'}^*}{2} - m_{\mathcal{C},\mathcal{C}^*},$$
$$m_{\bar{\mathcal{C}},\bar{\mathcal{C}}^*} := \binom{n}{2} - m_{\mathcal{C},\mathcal{C}^*} - m_{\mathcal{C},\bar{\mathcal{C}}^*} - m_{\bar{\mathcal{C}},\mathcal{C}^*},$$

where $m_{\mathcal{C},\mathcal{C}^*}$ denotes the number of pairs of samples that are assigned to the same cluster both in $\{\mathcal{C}_y\}_{y=1}^c$ and $\{\mathcal{C}_{y'}^*\}_{y'=1}^c$, $m_{\mathcal{C},\bar{\mathcal{C}}^*}$ denotes the number of pairs of samples that are assigned to the same cluster in $\{\mathcal{C}_y\}_{y=1}^c$ but are assigned to different clusters in $\{\mathcal{C}_{y'}^*\}_{y'=1}^c$, $m_{\bar{\mathcal{C}},\mathcal{C}^*}$ denotes the number of pairs of samples that are assigned to the same cluster in $\{\mathcal{C}_{y'}^*\}_{y'=1}^c$ but are assigned to different clusters in $\{\mathcal{C}_y\}_{y=1}^c$, and $m_{\bar{\mathcal{C}},\bar{\mathcal{C}}^*}$ denotes the number of pairs of samples that are assigned to different clusters both in $\{\mathcal{C}_y\}_{y=1}^c$ and $\{\mathcal{C}_{y'}^*\}_{y'=1}^c$. $m_{\mathcal{C},\mathcal{C}^*} + m_{\bar{\mathcal{C}},\bar{\mathcal{C}}^*}$ can be considered as the number of "agreements" between $\{\mathcal{C}_y\}_{y=1}^c$ and $\{\mathcal{C}_{y'}^*\}_{y'=1}^c$, while $m_{\mathcal{C},\bar{\mathcal{C}}^*} + m_{\bar{\mathcal{C}},\mathcal{C}^*}$ can be regarded as the number of "disagreements" between $\{\mathcal{C}_y\}_{y=1}^c$ and $\{\mathcal{C}_{y'}^*\}_{y'=1}^c$. The notation is summarized in Table 4(b).

The *Rand index* (RI) (Rand, 1971) is defined and expressed as

$$\mathrm{RI} := \frac{m_{\mathcal{C},\mathcal{C}^*} + m_{\bar{\mathcal{C}},\bar{\mathcal{C}}^*}}{m_{\mathcal{C},\mathcal{C}^*} + m_{\mathcal{C},\bar{\mathcal{C}}^*} + m_{\bar{\mathcal{C}},\mathcal{C}^*} + m_{\bar{\mathcal{C}},\bar{\mathcal{C}}^*}}$$
$$= (m_{\mathcal{C},\mathcal{C}^*} + m_{\bar{\mathcal{C}},\bar{\mathcal{C}}^*}) \Big/ \binom{n}{2}.$$

The Rand index lies between 0 and 1, and takes 1 if the two clustering solutions $\{\mathcal{C}_y\}_{y=1}^c$ and $\{\mathcal{C}_{y'}^*\}_{y'=1}^c$ agree with each other perfectly.

A potential drawback of the Rand index is that its expected value is not a constant (say, 0) if two clustering solutions are completely random. To overcome this problem, the *adjusted Rand index* (ARI) was proposed (Hubert & Arabie, 1985). ARI is defined as

$$\mathrm{ARI} := \frac{m_{\mathcal{C},\mathcal{C}^*} + m_{\bar{\mathcal{C}},\bar{\mathcal{C}}^*} - \mu}{m_{\mathcal{C},\mathcal{C}^*} + m_{\mathcal{C},\bar{\mathcal{C}}^*} + m_{\bar{\mathcal{C}},\mathcal{C}^*} + m_{\bar{\mathcal{C}},\bar{\mathcal{C}}^*} - \mu}.$$

μ is the expected value of $m_{\mathcal{C},\mathcal{C}^*} + m_{\bar{\mathcal{C}},\bar{\mathcal{C}}^*}$:

$$\mu := \mathbb{E}\left[m_{\mathcal{C},\mathcal{C}^*} + m_{\bar{\mathcal{C}},\bar{\mathcal{C}}^*}\right],$$

where $\mathbb{E}$ denotes the expectation over cluster assignments. ARI takes the maximum value 1 when two sets of cluster assignments are identical, and takes 0 if the index equals its expected value.

Under the assumption that the clustering solutions $\{\mathcal{C}_y\}_{y=1}^{c}$ and $\{\mathcal{C}^*_{y'}\}_{y'=1}^{c}$ are randomly drawn from a generalized hyper-geometric distribution, it holds that

$$\mathbb{E}\left[m_{\mathcal{C},\mathcal{C}^*}\right] = (m_{\mathcal{C},\mathcal{C}^*} + m_{\mathcal{C},\bar{\mathcal{C}}^*})(m_{\mathcal{C},\mathcal{C}^*} + m_{\bar{\mathcal{C}},\mathcal{C}^*}) \Big/ \binom{n}{2},$$

$$\mathbb{E}\left[m_{\bar{\mathcal{C}},\bar{\mathcal{C}}^*}\right] = (m_{\mathcal{C},\bar{\mathcal{C}}^*} + m_{\bar{\mathcal{C}},\bar{\mathcal{C}}^*})(m_{\bar{\mathcal{C}},\mathcal{C}^*} + m_{\bar{\mathcal{C}},\bar{\mathcal{C}}^*}) \Big/ \binom{n}{2}.$$

Then ARI can be expressed as

$$\mathrm{ARI} = \frac{\binom{n}{2}\sum_{y,y'=1}^{c}\binom{n_{y,y'}}{2} - \sum_{y=1}^{c}\binom{n_y}{2}\sum_{y'=1}^{c}\binom{n^*_{y'}}{2}}{\frac{1}{2}\binom{n}{2}\left[\sum_{y=1}^{c}\binom{n_y}{2} + \sum_{y'=1}^{c}\binom{n^*_{y'}}{2}\right] - \sum_{y=1}^{c}\binom{n_y}{2}\sum_{y'=1}^{c}\binom{n^*_{y'}}{2}}.$$

Note that RI and ARI can be defined even when two sets of cluster assignments $\{y_i\}_{i=1}^{n}$ and $\{y_i^*\}_{i=1}^{n}$ have different numbers of clusters, i.e., $\{\mathcal{C}_y\}_{y=1}^{c}$ and $\{\mathcal{C}^*_{y'}\}_{y'=1}^{c'}$ with $c \neq c'$. This is highly convenient in practice since, when the number of true clusters is large, clustering algorithms often produce clustering solutions with a smaller number of clusters (i.e., some of the clusters have no samples). Even in such cases, RI and ARI can still be used for evaluating the quality of clustering solutions.

References

Agakov, F., & Barber, D. (2006). Kernelized infomax clustering. *Advances in Neural Information Processing Systems 18* (pp. 17–24). Cambridge, MA, USA: MIT Press.

Ali, S. M., & Silvey, S. D. (1966). A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society, Series B*, *28*, 131–142.

Aloise, D., Deshpande, A., Hansen, P., & Popat, P. (2009). NP-hardness of Euclidean sum-of-squares clustering. *Machine Learning*, *75*, 245–249.

Amari, S. (1967). Theory of adaptive pattern classifiers. *IEEE Transactions on Electronic Computers*, *EC-16*, 299–307.

Andrieu, C., de Freitas, N., Doucet, A., & Jordan, M. I. (2003). An introduction to MCMC for machine learning. *Machine Learning*, *50*, 5–43.

Antoniak, C. (1974). Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, *2*, 1152–1174.

Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American Mathematical Society*, *68*, 337–404.

Attias, H. (2000). A variational Baysian framework for graphical models. *Advances in Neural Information Processing Systems 12* (pp. 209–215). MIT Press.

Bach, F., & Harchaoui, Z. (2008). DIFFRAC: A discriminative and flexible framework for clustering. *Advances in Neural Information Processing Systems 20* (pp. 49–56). Cambridge, MA, USA: MIT Press.

Bach, F., & Jordan, M. I. (2006). Learning spectral clustering, with application to speech separation. *Journal of Machine Learning Research*, *7*, 1963–2001.

Bao, L., & Intille, S. S. (2004). Activity recognition from user-annotated acceleration data. *Proceedings of 2nd IEEE International Conference on Pervasive Computing* (pp. 1–17).

Bharatula, N. B., Stager, M., Lukowicz, P., & Troster, G. (2005). Empirical study of design choices in multi-sensor context ecognition. *Proceedings of International Forun on Applied Wearable Computing* (pp. 79–93).

Bishop, C. M. (2006). *Pattern recognition and machine learning*. New York, NY, USA: Springer.

Blei, D. M., & Jordan, M. I. (2006). Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, *1*, 121–144.

Carreira-Perpiñán, M. A. (2006). Fast nonparametric clustering with Gaussian blurring mean-shift. *Proceedings of 23rd International Conference on Machine Learning (ICML2006)* (pp. 153–160). Pittsburgh, PA.

Carreira-Perpiñán, M. A. (2007). Gaussian mean shift is an EM algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*, 767–776.

Cheng, Y. (1995). Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *17*, 790–799.

Chung, F. R. K. (1997). *Spectral graph theory*. Providence, RI, USA: American Mathematical Society.

Cour, T., Gogin, N., & Shi, J. (2005). Learning spectral graph segmentation. *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics* (pp. 65–72). Society for Artificial Intelligence and Statistics.

Cover, T. M., & Thomas, J. A. (2006). *Elements of information theory*. Hoboken, NJ, USA: John Wiley & Sons, Inc. 2nd edition.

Csiszár, I. (1967). Information-type measures of difference of probability distributions and indirect observation. *Studia Scientiarum Mathematicarum Hungarica*, *2*, 229–318.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, series B*, *39*, 1–38.

Dhillon, I. S., Guan, Y., & Kulis, B. (2004). Kernel k-means, spectral clustering and normalized cuts. *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 551–556). New York, NY, USA: ACM Press.

Ding, C., & He, X. (2004). K-means clustering via principal component analysis. *Proceedings of the Twenty-First International Conference on Machine Learning (ICML2004)* (pp. 225–232). New York, NY, USA: ACM Press.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification*. New York, NY, USA: Wiley. Second edition.

Duffy, N., & Collins, M. (2002). Convolution kernels for natural language. *Advances in Neural Information Processing Systems 14* (pp. 625–632). Cambridge, MA, USA: MIT Press.

Faivishevsky, L., & Goldberger, J. (2010). A nonparametric information theoretic clustering algorithm. *Proceedings of 27th International Conference on Machine Learning (ICML2010)* (pp. 351–358). Haifa, Israel.

Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, *1*, 209–230.

Fukunaga, K., & Hostetler, L. D. (1975). The estimation of the gradient of a density function, with application in pattern recognition. *IEEE Transactions on Information Theory*, *21*, 32–40.

Gärtner, T. (2003). A survey of kernels for structured data. *SIGKDD Explorations*, *5*, S268–S275.

Gärtner, T., Flach, P., & Wrobel, S. (2003). On graph kernels: Hardness results and efficient alternatives. *Proceedings of the Sixteenth Annual Conference on Computational Learning Theory* (pp. 129–143).

Ghahramani, Z., & Beal, M. J. (2000). Variational inference for Bayesian mixtures of factor analysers. *Advances in Neural Information Processing Systems 12* (pp. 449–455). MIT Press.

Girolami, M. (2002). Mercer kernel-based clustering in feature space. *IEEE Transactions on Neural Networks*, *13*, 780–784.

Golub, G. H., & Loan, C. F. V. (1989). *Matrix computations*. Baltimore, MD, USA: Johns Hopkins University Press. Second edition.

Gomes, R., Krause, A., & Perona, P. (2010). Discriminative clustering by regularized information maximization. *Advances in Neural Information Processing Systems 23* (pp. 766–774).

Gretton, A., Bousquet, O., Smola, A., & Schölkopf, B. (2005). Measuring statistical dependence with Hilbert-Schmidt norms. *Algorithmic Learning Theory* (pp. 63–77). Berlin, Germany: Springer-Verlag.

Hachiya, H., Sugiyama, M., & Ueda, N. (2012). Importance-weighted least-squares probabilistic classifier for covariate shift adaptation with application to human activity recognition. *Neurocomputing*, *80*, 93–101.

Härdle, W., Müller, M., Sperlich, S., & Werwatz, A. (2004). *Nonparametric and semiparametric models*. Berlin, Germany: Springer.

Horn, R. A., & Johnson, C. A. (1985). *Matrix analysis*. Cambridge, UK: Cambridge University Press.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification*, *2*, 193–218.

Joachims, T. (2002). *Learning to classify text using support vector machines: Methods, theory and algorithms*. Boston, MA, USA: Kluwer Academic Publishers.

Jolliffe, I. T. (1986). *Principal component analysis*. New York, NY, USA: Springer-Verlag.

Kain, A., & Macon, M. W. (1988). Spectral voice conversion for text-to-speech synthesis. *Proceedings of 1998 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP1998)* (pp. 285–288). Washington, DC, U.S.A.

Kashima, H., & Koyanagi, T. (2002). Kernels for semi-structured data. *Proceedings of the Nineteenth International Conference on Machine Learning* (pp. 291–298). San Francisco, CA, USA: Morgan Kaufmann.

Kashima, H., Tsuda, K., & Inokuchi, A. (2003). Marginalized kernels between labeled graphs. *Proceedings of the Twentieth International Conference on Machine Learning* (pp. 321–328). San Francisco, CA, USA: Morgan Kaufmann.

Koltchinskii, V. (1998). Asymptotics of spectral projections of some random matrices approximating integral operators. *Progress in Probabilty* (pp. 191–227).

Koltchinskii, V., & Giné, E. (2000). Random matrix approximation of spectra of integral operators. *Bernoulli*, *6*, 113–167.

Kondor, R. I., & Lafferty, J. (2002). Diffusion kernels on graphs and other discrete input spaces. *Proceedings of the Nineteenth International Conference on Machine Learning* (pp. 315–322).

Kozachenko, L. F., & Leonenko, N. N. (1987). Sample estimate of entropy of a random vector. *Problems of Information Transmission*, *23*, 95–101.

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, *22*, 79–86.

Kurihara, K., & Welling, M. (2009). Bayesian k-means as a "maximization-expectation" algorithm. *Neural Computation*, *21*, 1145–1172.

Lee, Y. K., & Ng, H. T. (2002). An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. *Proceedings of Conference on Empirical Methods in Natural Language Processing* (pp. 41–48).

Li, Y. F., Tsang, I. W., Kwok, J. T., & Zhou, Z.-H. (2009). Tighter and convex maximum margin clustering. *Proceedings of Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS2009)* (pp. 344–351). Clearwater Beach, FL, USA.

Lin, D., Grimson, E., & Fisher, J. (2010). Construction of dependent Dirichlet processes based on Poisson processes. *Advances in Neural Information Processing Systems 23* (pp. 1387–1395).

Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., & Watkins, C. (2002). Text classification using string kernels. *Journal of Machine Learning Research*, *2*, 419–444.

MacKay, D. J. C. (2003). *Information theory, inference, and learning algorithms*. Cambridge, UK: Cambridge University Press.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability* (pp. 281–297). Berkeley, CA, USA: University of California Press.

Meila, M., & Shi, J. (2001). Learning segmentation by random walks. *Advances in Neural Information Processing Systems 13* (pp. 873–879). Cambridge, MA, USA: MIT Press.

Neal, R. M. (2000). Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, *9*, 249–265.

Ng, A. Y., Jordan, M. I., & Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems 14* (pp. 849–856). Cambridge, MA, USA: MIT Press.

Niu, G., Dai, B., Shang, L., & Sugiyama, M. (2013). Maximum volume clustering: A new discriminative clustering approach. *Journal of Machine Learning Research*. to appear.

Niu, Z.-Y., Ji, D.-H., & Tan, C. L. (2005). A semi-supervised feature clustering algorithm with application to word sense disambiguation. *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing* (pp. 907–914).

Pearson, K. (1900). On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *Philosophical Magazine Series 5*, *50*, 157–175.

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, *2*, 559–572.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, *66*, 846–850.

Rodríguez, A., Dunson, D. B., & Gelfand., A. E. (2008). The Nested dirichlet process. *Journal of the American Statistical Association*, *103*, 1131–1154.

Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels*. Cambridge, MA, USA: MIT Press.

Shental, N., Zomet, A., Hertz, T., & Weiss, Y. (2003). Learning and inferring image segmentations using the GBP typical cut algorithm. *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1243–1250).

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*, 888–905.

Silverman, B. W. (1986). *Density estimation for statistics and data analysis*. London, UK: Chapman and Hall.

Song, L., Smola, A., Gretton, A., & Borgwardt, K. (2007). A dependence maximization view of clustering. *Proceedings of the 24th Annual International Conference on Machine Learning (ICML2007)* (pp. 815–822). Corvallis, OR.

Sugiyama, M. (2007). Dimensionality reduction of multimodal labeled data by local Fisher discriminant analysis. *Journal of Machine Learning Research*, *8*, 1027–1061.

Sugiyama, M. (2013). Machine learning with squared-loss mutual information. *Entropy*, *15*, 80–112.

Sugiyama, M., Suzuki, T., & Kanamori, T. (2012). *Density ratio estimation in machine learning*. Cambridge, UK: Cambridge University Press.

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

Sugiyama, M., Yamada, M., Kimura, M., & Hachiya, H. (2011). On information-maximization clustering: Tuning parameter selection and analytic solution. *Proceedings of 28th International Conference on Machine Learning (ICML2011)* (pp. 65–72). Bellevue, Washington, USA.

Suzuki, T., Sugiyama, M., Kanamori, T., & Sese, J. (2009). Mutual information estimation reveals global associations between stimuli and biological processes. *BMC Bioinformatics*, *10*, S52 (12 pages).

Suzuki, T., Sugiyama, M., Sese, J., & Kanamori, T. (2008). Approximating mutual information by maximum likelihood density ratio estimation. *Proceedings of ECML-PKDD2008 Workshop on New Challenges for Feature Selection in Data Mining and Knowledge Discovery 2008 (FSDM2008)* (pp. 5–20). Antwerp, Belgium.

Teh, Y. W., M. I. Jordan, M. J. B., & Blei, D. M. (2007). Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, *101*, 1566–1581.

Ueda, N., Nakano, R., Ghahramani, Z., & Hinton, G. E. (2000). SMEM algorithm for mixture models. *Neural Computation*, *12*, 2109–2128.

Valizadegan, H., & Jin, R. (2007). Generalized maximum margin clustering and unsupervised kernel learning. *Advances in Neural Information Processing Systems 19* (pp. 1417–1424). Cambridge, MA, USA: MIT Press.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. Berlin, Germany: Springer-Verlag.

von Luxburg, U. (2004). *Statistical learning with similarity and dissimilarity functions*. Doctoral dissertation, Technical University of Berlin, Berlin, Germany.

Wang, F., Zhao, B., & Zhang, C. (2010). Linear time maximum margin clustering. *IEEE Transactions on Neural Networks*, *21*, 319–332.

Xu, L., Neufeld, J., Larson, B., & Schuurmans, D. (2005). Maximum margin clustering. *Advances in Neural Information Processing Systems 17* (pp. 1537–1544). Cambridge, MA, USA: MIT Press.

Yang, W.-Y., Kwok, J. T., & Lu, B.-L. (2010). Spectral and semidefinite relaxation of the CLUHSIC algorithm. *Proceedings of the 2010 SIAM International Conference on Data Mining* (pp. 106–117).

Zelnik-Manor, L., & Perona, P. (2005). Self-tuning spectral clustering. *Advances in Neural Information Processing Systems 17* (pp. 1601–1608). Cambridge, MA, USA: MIT Press.

Zha, H., He, X., Ding, C., Gu, M., & Simon, H. (2002). Spectral relaxation for k-means clustering. *Advances in Neural Information Processing Systems 14* (pp. 1057–1064). Cambridge, MA, USA: MIT Press.

Zhang, K., Tsang, I. W., & Kwok, J. T. (2009). Maximum margin clustering made practical. *IEEE Transactions on Neural Networks*, *20*, 583–596.

Zhao, B., Wang, F., & Zhang, C. (2008). Maximum margin clustering via cutting plane algorithm. *Proceedings of the 2007 SIAM International Conference on Data Mining* (pp. 751–762).