REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* 17-02-2015	**2. REPORT TYPE** Final	**3. DATES COVERED** *(From - To)* 28-03-2013 – 27-03-2015

4. TITLE AND SUBTITLE Machine Learning with Distances	**5a. CONTRACT NUMBER** FA2386-13-1-4041
	5b. GRANT NUMBER Grant AOARD-134041
	5c. PROGRAM ELEMENT NUMBER 61102F
6. AUTHOR(S) Prof. Masashi Sugiyama	**5d. PROJECT NUMBER**
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Department of Computer Science, Tokyo Institute of Technology y 2-12-1-W8-74, O-okayama, Meguro-ku Tokyo 152-8552 Japan	**8. PERFORMING ORGANIZATION REPORT NUMBER** N/A
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AOARD UNIT 45002 APO AP 96338-5002	**10. SPONSOR/MONITOR'S ACRONYM(S)** AFRL/AFOSR/IOA(AOARD)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) AOARD-134041

12. DISTRIBUTION/AVAILABILITY STATEMENT

Distribution Code A: Approved for public release, distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Various machine learning tasks such as learning under non-stationarity, change detection, and dimensionality reduction can be solved by estimating some distance/ratio between two probability distributions. This project developed accurate and computationally efficient methods for estimating the distance/ratio from data, and demonstrated their usefulness in experiments. The principle idea is that when solving a problem of interest, we should not solve a more general sub-problem as an intermediate step, i,e., directly estimate the difference/ratio of the two distributions rather than estimating both separately and take the difference/ratio later. Types of the problems actually solved are change detection in time series, salient object detection in an image, measuring statistical independence, detection of structure change, covariance shift, class balance change, information maximization clustering.

15. SUBJECT TERMS

Density ratio, Density difference, Change detection, Object detection, Statistical independence, Structure change detection, Covariance shift, clustering

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Hiroshi Motoda, Ph. D.
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			
U	U	U	SAR	112	**19b. TELEPHONE NUMBER** *(Include area code)* +81-42-511-2011

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE
26 MAR 2015

2. REPORT TYPE
Final

3. DATES COVERED
28-03-2013 to 27-03-2015

4. TITLE AND SUBTITLE
Machine Learning with Distances

5a. CONTRACT NUMBER
FA2386-13-1-4041

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
61102F

6. AUTHOR(S)
Masashi Sugiyama

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Department of Computer Science, Tokyo Institute of Technology,2-12-1-W8-74, O-okayama, Meguro-ku,Tokyo 152-8552,Japan,NA,NA

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AOARD, UNIT 45002, APO, AP, 96338-5002

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/AFOSR/IOA(AOARD)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AOARD-134041

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Various machine learning tasks such as learning under non-stationarity, change detection, and dimensionality reduction can be solved by estimating some distance/ratio between two probability distributions. This project developed accurate and computationally efficient methods for estimating the distance/ratio from data, and demonstrated their usefulness in experiments. The principle idea is that when solving a problem of interest, we should not solve a more general sub-problem as an intermediate step, i,e., directly estimate the difference/ratio of the two distributions rather than estimating both separately and take the difference/ratio later. Types of the problems actually solved are change detection in time series, salient object detection in an image, measuring statistical independence, detection of structure change, covariance shift, class balance change, information maximization clustering.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**112**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Final Report for AOARD Grant FA2386-13-1-4041

Machine Learning with Distances

Masashi Sugiyama (sugi@cs.titech.ac.jp)
Tokyo Institute of Technology
2-12-1 O-okayama, Meguro, Tokyo 152-8552, Japan
Phone : +81-3-5734-2699   Fax : +81-3-5734-3907

Feb. 16, 2015

Abstract

Various machine learning tasks such as learning under non-stationarity, change detection, and dimensionality reduction can be solved by estimating some distances between probability distributions. In this project, we developed accurate and computationally efficient methods for estimating the distances from data, and demonstrated their usefulness in experiments.

1 Introduction

The goal of machine learning is to find useful knowledge behind data. Many machine learning tasks contain multiple datasets (such as data taken from different categories, different time periods, etc.) and comparing the probability distributions behind these datasets is a fundamental challenge in statistics and machine learning communities. More specifically, an estimator of a *distance* between probability distributions can be used for solving various machine learning tasks such as change detection in time-series and semi-supervised learning under class-balance change. In this project, we develop a unified framework of machine learning based on distances between probability distributions.

The Kullback-Leibler (KL) distance is the de-facto standard distance measure in statistics and machine learning, because of its high compatibility with maximum likelihood estimation. However, the KL distance has several weaknesses such as high sensitivity to outliers, high computational requirements, and non-metricity. In this project, we propose to use other distances than the KL distance, such as the relative ratio based distances and the difference based distances. These novel distance measures can potentially overcome the above weaknesses of the KL distance.

2 Divergence Estimation

2.1 Background

Let us consider the problem of approximating a divergence D between two probability distributions P and P' on $\mathbb{R}^d$ from two sets of independent and identically distributed samples $\mathcal{X} := \{\boldsymbol{x}_i\}_{i=1}^{n}$ and $\mathcal{X}' := \{\boldsymbol{x}'_{i'}\}_{i'=1}^{n'}$ following P and P'.

A divergence approximator can be used for various purposes such as two-sample testing [167, 86], change detection in time-series [92], class-prior estimation under class-balance change [45], salient object detection in images [214], and event detection from movies [213] and Twitter [112]. Furthermore, an approximator of the divergence between the joint distribution and the product of marginal distributions can be used for solving a wide range of machine learning problems [157], including independence testing [166], feature selection [179, 77], feature extraction [178, 204], canonical dependency analysis [89], object matching [208], independent component analysis [177], clustering [175, 97], and causal direction learning [207]. For this reason, accurately approximating a divergence between two probability distributions from their samples has been one of the challenging research topics in the statistics, information theory, and machine learning communities.

A naive way to approximate the divergence from P to P', denoted by $D(P\|P')$, is to first obtain estimators $\widehat{P}_{\mathcal{X}}$ and $\widehat{P}'_{\mathcal{X}'}$ of the distributions P and P' separately from their samples $\mathcal{X}$ and $\mathcal{X}'$, and then compute a plug-in approximator $D(\widehat{P}_{\mathcal{X}}\|\widehat{P}'_{\mathcal{X}'})$. However, this naive two-step approach violates *Vapnik's principle* [194]:

> *If you possess a restricted amount of information for solving some problem, try to solve the problem directly and never solve a more general problem as an intermediate step. It is possible that the available information is sufficient for a direct solution but is insufficient for solving a more general intermediate problem.*

More specifically, if we know the distributions P and P', we can immediately know their divergence $D(P\|P')$. However, knowing the divergence $D(P\|P')$ does not necessarily imply knowing the distributions P and P', because different pairs of distributions can yield the same divergence value. Thus, estimating the distributions P and P' is more general than estimating the divergence $D(P\|P')$. Following Vapnik's principle, direct divergence approximators $\widehat{D}(\mathcal{X}, \mathcal{X}')$ that do not involve the estimation of distributions P and P' have been developed recently [173, 124, 82, 212, 172].

The purpose of this article is to give an overview of the development of such direct divergence approximators. In Section 2.2, we review the definitions of the Kullback-Leibler divergence, the Pearson divergence, the relative Pearson divergence, and the L^2-distance, and discuss their pros and cons. Then, in Section 2.3, we review direct approximators of these divergences that do not involve the estimation of probability distributions. In Section 2.4, we show practical usage of divergence approximators in unsupervised change-detection in time-series, semi-supervised class-prior estimation under class-balance change, salient object detection in an image, and evaluation of statistical independence between random variables. Finally, we conclude in Section 2.5.

2.2 Divergence Measures

A function $d(\cdot,\cdot)$ is called a *distance* if and only if the following four conditions are satisfied:

- Non-negativity: $\forall x, y, \quad d(x,y) \geq 0$
- Non-degeneracy: $d(x,y) = 0 \iff x = y$
- Symmetry: $\forall x, y, \quad d(x,y) = d(y,x)$
- Triangle inequality: $\forall x, y, z \quad d(x,z) \leq d(x,y) + d(y,z)$

A divergence is a pseudo-distance that still acts like a distance, but it may violate some of the above conditions. In this section, we introduce useful divergence and distance measures between probability distributions.

2.2.1 Kullback-Leibler (KL) Divergence

The most popular divergence measure in statistics and machine learning is the KL divergence [103] defined as

$$\mathrm{KL}(p\|p') := \int p(\boldsymbol{x}) \log \frac{p(\boldsymbol{x})}{p'(\boldsymbol{x})} \mathrm{d}\boldsymbol{x},$$

where $p(\boldsymbol{x})$ and $p'(\boldsymbol{x})$ are probability density functions of P and P', respectively.

Advantages of the KL divergence are that it is compatible with maximum likelihood estimation, it is invariant under input metric change, its Riemannian geometric structure is well studied [8], and it can be approximated accurately via *direct density-ratio estimation* [173, 124, 168]. However, it is not symmetric, it does not satisfy the triangle inequality, its approximation is computationally expensive due to the log function, and it is sensitive to outliers and numerically unstable because of the strong non-linearity of the log function and possible unboundedness of the density-ratio function p/p' [36, 212].

2.2.2 Pearson (PE) Divergence

The PE divergence [128] is a squared-loss variant of the KL divergence defined as

$$\mathrm{PE}(p\|p') := \int p'(\boldsymbol{x}) \left(\frac{p(\boldsymbol{x})}{p'(\boldsymbol{x})} - 1 \right)^2 \mathrm{d}\boldsymbol{x}. \tag{1}$$

Because both the PE and KL divergences belong to the class of Ali-Silvey-Csiszár divergences (which is also known as f-divergences) [5, 39], they share similar theoretical properties such as the invariance under input metric change.

The PE divergence can also be accurately approximated via direct density-ratio estimation in the same way as the KL divergence [82, 168]. However, its approximator can be obtained *analytically* in a computationally much more efficient manner than the KL divergence, because the quadratic function the PE divergence adopts is compatible with least-squares estimation. Furthermore, the PE divergence tends to be more robust against outliers than the KL divergence [170]. However, other weaknesses of the KL divergence such as asymmetry, violation of the triangle inequality, and possible unboundedness of the density-ratio function p/p' remain unsolved in the PE divergence.

2.2.3 Relative Pearson (rPE) Divergence

To overcome the possible unboundedness of the density-ratio function p/p', the rPE divergence was recently introduced [212]. The rPE divergence is defined as

$$\begin{aligned} \mathrm{rPE}(p\|p') &:= \mathrm{PE}(p\|q_\alpha) \\ &= \int q_\alpha(\boldsymbol{x}) \left(\frac{p(\boldsymbol{x})}{q_\alpha(\boldsymbol{x})} - 1 \right)^2 \mathrm{d}\boldsymbol{x}, \end{aligned} \tag{2}$$

where, for $0 \leq \alpha < 1$, q_α is defined as the α-mixture of p and p':

$$q_\alpha = \alpha p + (1 - \alpha) p'.$$

When $\alpha = 0$, the rPE divergence is reduced to the plain PE divergence. The quantity p/q_α is called the *relative density ratio*, which is always upper-bounded by $1/\alpha$ for $\alpha > 0$ because

$$\frac{p(\boldsymbol{x})}{q_\alpha(\boldsymbol{x})} = \frac{1}{\alpha + (1 - \alpha)\frac{p'(\boldsymbol{x})}{p(\boldsymbol{x})}} < \frac{1}{\alpha}.$$

Thus, it can overcome the unboundedness problem of the PE divergence, while the invariance under input metric change is still maintained.

The rPE divergence is still compatible with least-squares estimation, and it can be approximated in almost the same way as the PE divergence via *direct relative density-ratio estimation* [212]. Indeed, an rPE-divergence approximator can still be obtained analytically in an accurate and computationally efficient manner. However, it still violates symmetry and the triangle inequality in the same way as the KL and PE divergence. Furthermore, the choice of α may not be straightforward in some applications.

2.2.4 L^2-Distance

The L^2-distance is another standard distance measure between probability distributions defined as

$$L^2(p, p') := \int \Big(p(\boldsymbol{x}) - p'(\boldsymbol{x}) \Big)^2 \mathrm{d}\boldsymbol{x}.$$

The L^2-distance is a proper distance measure, and thus it is symmetric and satisfies the triangle inequality. Furthermore, the density difference $p(\boldsymbol{x}) - p'(\boldsymbol{x})$ is always bounded as long as each density is bounded. Therefore, the L^2-distance is stable, without the need of tuning any control parameter such as α in the rPE divergence.

The L^2-distance is also compatible with least-squares estimation, and it can be accurately and analytically approximated in a computationally efficient and numerically stable manner via *direct density-difference estimation* [172]. However, the L^2-distance is not invariant under input metric change, which is a unique property inherent to ratio-based divergences.

2.3 Direct Divergence Approximation

In this section, we review recent advances in direct divergence approximation.

Suppose that we are given two sets of independent and identically distributed samples $\mathcal{X} := \{\boldsymbol{x}_i\}_{i=1}^{n}$ and $\mathcal{X}' := \{\boldsymbol{x}'_{i'}\}_{i'=1}^{n'}$ from probability distributions on $\mathbb{R}^d$ with densities $p(\boldsymbol{x})$ and $p'(\boldsymbol{x})$, respectively:

$$\begin{aligned}\mathcal{X} &:= \{\boldsymbol{x}_i\}_{i=1}^{n} \overset{\text{i.i.d.}}{\sim} p(\boldsymbol{x}),\\ \mathcal{X}' &:= \{\boldsymbol{x}'_{i'}\}_{i'=1}^{n'} \overset{\text{i.i.d.}}{\sim} p'(\boldsymbol{x}).\end{aligned}$$

Our goal is to approximate a divergence between from p to p' from samples $\mathcal{X}$ and $\mathcal{X}'$.

2.3.1 KL Divergence Approximation

The key idea of direct KL divergence approximation is to estimate the density ratio p/p' without estimating the densities p and p' [173]. More specifically, a density-ratio estimator is obtained by minimizing the KL divergence from p to $r \cdot p'$ with respect to a density-ratio model r, under the constraints that the density-ratio function is non-negative and $r \cdot p'$ is integrated to one:

$$\begin{aligned}&\min_{r} \mathrm{KL}(p \| r \cdot p')\\ &\text{subject to } r \geq 0 \text{ and } \int r(\boldsymbol{x}) p'(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = 1.\end{aligned}$$

Its empirical optimization problem, where an irrelevant constant is ignored and the expectations are approximated by the sample averages, is given by

$$\begin{aligned}&\max_{r} \frac{1}{n} \sum_{i=1}^{n} \log r(\boldsymbol{x}_i)\\ &\text{subject to } r \geq 0 \text{ and } \frac{1}{n'} \sum_{i'=1}^{n'} r(\boldsymbol{x}'_{i'}) = 1.\end{aligned}$$

Let us consider the following Gaussian density-ratio model:

$$r(\boldsymbol{x}) = \sum_{\ell=1}^{n} \theta_\ell \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}_\ell\|^2}{2\sigma^2}\right), \tag{3}$$

where $\|\cdot\|$ denotes the ℓ_2-norm. We define the vector of parameters $\{\theta_\ell\}_{\ell=1}^{n}$ as

$$\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)^\top,$$

where $^\top$ denotes the transpose. In this model, the Gaussian kernels are located on numerator samples $\{\boldsymbol{x}_i\}_{i=1}^{n}$ because the density ratio p/p' tends to take large values in the regions where the numerator samples $\{\boldsymbol{x}_i\}_{i=1}^{n}$ exist. Alternatively, Gaussian kernels may be located on both numerator and denominator samples, but this seems not to further improve the accuracy [173]. When n is very large, a (random) subset of numerator samples $\{\boldsymbol{x}_i\}_{i=1}^{n}$ may be chosen as Gaussian centers, which can reduce the computational cost.

For the Gaussian density-ratio model (3), the above optimization problem is expressed as

$$\begin{aligned}
&\max_{\boldsymbol{\theta}} \ \frac{1}{n}\sum_{i=1}^{n} \log\left(\sum_{\ell=1}^{n} \theta_\ell \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_\ell\|^2}{2\sigma^2}\right)\right)\\
&\text{subject to } \theta_1, \ldots, \theta_n \geq 0\\
&\text{and } \frac{1}{n'}\sum_{i'=1}^{n'}\sum_{\ell=1}^{n} \theta_\ell \exp\left(-\frac{\|\boldsymbol{x}'_{i'} - \boldsymbol{x}_\ell\|^2}{2\sigma^2}\right) = 1.
\end{aligned}$$

This is a convex optimization problem and thus the global optimal solution can be obtained easily, e.g., by gradient-projection iterations. Furthermore, the global optimal solution tends to be *sparse* (i.e., many parameter values become exactly zero), which can be utilized for reducing the computational cost.

The Gaussian width σ is a tuning parameter in this algorithm, and it can be systematically optimized by *cross-validation* with respect to the objective function. More specifically, the numerator samples $\mathcal{X} := \{\boldsymbol{x}_i\}_{i=1}^{n}$ are divided into T disjoint subsets $\{\mathcal{X}_t\}_{t=1}^{T}$ of (approximately) the same size. Then a density-ratio estimator $\widehat{r}_t(\boldsymbol{x})$ is obtained using $\mathcal{X}\backslash\mathcal{X}_t$ and $\mathcal{X}' := \{\boldsymbol{x}'_{i'}\}_{i'=1}^{n'}$ (i.e., all numerator samples without $\mathcal{X}_t$ and all denominator samples), and its objective value for the hold-out numerator samples $\mathcal{X}_t$ is computed:

$$\frac{1}{|\mathcal{X}_t|}\sum_{\boldsymbol{x}\in\mathcal{X}_t} \log \widehat{r}_t(\boldsymbol{x}),$$

where $|\mathcal{X}_t|$ denotes the number of elements in the set $\mathcal{X}_t$. This procedure is repeated for $t = 1, \ldots, T$, and the σ value that maximizes the average of the above hold-out objective values is chosen as the best one.

Given a density-ratio estimator $\widehat{r}$, a KL-divergence approximator $\widehat{\mathrm{KL}}(\mathcal{X}\|\mathcal{X}')$ can be constructed as

$$\widehat{\mathrm{KL}}(\mathcal{X}\|\mathcal{X}') := \frac{1}{n}\sum_{i=1}^{n} \log \widehat{r}(\boldsymbol{x}_i).$$

Variations of this procedure for other density-ratio models have been developed, including the log-linear model [187], the Gaussian mixture model [206], and the mixture of probabilistic principal component analyzers [211]. Also, an unconstrained variant, which corresponds to approximately maximizing the *Legendre-Fenchel lower bound* of the KL divergence [95], was proposed [124]:

$$\widetilde{\mathrm{KL}}(\mathcal{X}\|\mathcal{X}') := \max_{r}\left[\frac{1}{n}\sum_{i=1}^{n}\log r(\boldsymbol{x}_i) - \frac{1}{n'}\sum_{i'=1}^{n'} r(\boldsymbol{x}'_{i'}) + 1\right].$$

2.3.2 PE Divergence Approximation

The PE divergence can also be directly approximated without estimating the densities p and p' via direct estimation of the density ratio p/p' [82]. More specifically, a density-ratio estimator is obtained by minimizing the p'-weighted squared difference between a density-ratio model r and the true density-ratio function p/p':

$$\min_{r}\int p'(\boldsymbol{x})\left(r(\boldsymbol{x}) - \frac{p(\boldsymbol{x})}{p'(\boldsymbol{x})}\right)^2 \mathrm{d}\boldsymbol{x}.$$

Its empirical criterion where an irrelevant constant is ignored and the expectations are approximated by the sample averages is given by

$$\min_{r}\left[\frac{1}{n'}\sum_{i'=1}^{n'} r^2(\boldsymbol{x}'_{i'}) - \frac{2}{n}\sum_{i=1}^{n} r(\boldsymbol{x}_i)\right].$$

For the Gaussian density-ratio model (3) together with the ℓ_2-regularizer, the above optimization problem is expressed as

$$\min_{\boldsymbol{\theta}}\left[\boldsymbol{\theta}^\top\widehat{\boldsymbol{G}}'\boldsymbol{\theta} - 2\boldsymbol{\theta}^\top\widehat{\boldsymbol{h}} + \lambda\|\boldsymbol{\theta}\|^2\right], \tag{4}$$

where $\lambda \geq 0$ denotes the regularization parameter, $\widehat{\boldsymbol{G}}'$ is the $n\times n$ matrix with the (ℓ,ℓ')-th element defined by

$$\widehat{G}'_{\ell,\ell'} := \frac{1}{n'}\sum_{i'=1}^{n'}\exp\left(-\frac{\|\boldsymbol{x}'_{i'} - \boldsymbol{x}_\ell\|^2}{2\sigma^2}\right)\exp\left(-\frac{\|\boldsymbol{x}'_{i'} - \boldsymbol{x}_{\ell'}\|^2}{2\sigma^2}\right),$$

and $\widehat{\boldsymbol{h}}$ is the n-dimensional vector with the ℓ-th element defined by

$$\widehat{h}_\ell := \frac{1}{n}\sum_{i=1}^{n}\exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_\ell\|^2}{2\sigma^2}\right).$$

This is a convex optimization problem, and the global optimal solution can be computed *analytically* as

$$(\widehat{\boldsymbol{G}}' + \lambda\boldsymbol{I})^{-1}\widehat{\boldsymbol{h}},$$

where $\boldsymbol{I}$ denotes the identity matrix.

The Gaussian width σ and the regularization parameter λ are the tuning parameters in this algorithm, and they can be systematically optimized by cross-validation with respect to the objective function as follows: First, the numerator and denominator samples $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^{n}$ and $\mathcal{X}' = \{\boldsymbol{x}'_{i'}\}_{i'=1}^{n'}$ are divided into T disjoint subsets $\{\mathcal{X}_t\}_{t=1}^{T}$ and $\{\mathcal{X}'_t\}_{t=1}^{T}$, respectively. Then a density-ratio estimator $\widehat{r}_t(\boldsymbol{x})$ is obtained using $\mathcal{X}\backslash\mathcal{X}_t$ and $\mathcal{X}'\backslash\mathcal{X}'_t$ (i.e., all samples without $\mathcal{X}_t$ and $\mathcal{X}'_t$), and its objective value for the hold-out samples $\mathcal{X}_t$ and $\mathcal{X}'_t$ is computed:

$$\frac{1}{|\mathcal{X}'_t|}\sum_{\boldsymbol{x}'\in\mathcal{X}'_t}\widehat{r}_t(\boldsymbol{x}')^2 - \frac{2}{|\mathcal{X}_t|}\sum_{\boldsymbol{x}\in\mathcal{X}_t}\widehat{r}_t(\boldsymbol{x}). \tag{5}$$

This procedure is repeated for $t = 1, \ldots, T$, and the σ and λ values that maximize the average of the above hold-out objective values are chosen as the best ones.

By expanding the squared term $\left(\frac{p(\boldsymbol{x})}{p'(\boldsymbol{x})} - 1\right)^2$ in Eq.(1), the PE divergence can be expressed as

$$\mathrm{PE} = \int p(\boldsymbol{x})\frac{p(\boldsymbol{x})}{p'(\boldsymbol{x})}\mathrm{d}\boldsymbol{x} - 1 \tag{6}$$

$$= -\int p'(\boldsymbol{x})\left(\frac{p(\boldsymbol{x})}{p'(\boldsymbol{x})}\right)^2\mathrm{d}\boldsymbol{x} + 2\int p(\boldsymbol{x})\frac{p(\boldsymbol{x})}{p'(\boldsymbol{x})}\mathrm{d}\boldsymbol{x} - 1. \tag{7}$$

Note that Eq.(7) can also be obtained via *Legendre-Fenchel convex duality* of the divergence functional [140]. Based on these expressions, PE divergence approximators are obtained using a density-ratio estimator $\widehat{r}$ as

$$\widehat{\mathrm{PE}}(\mathcal{X}\|\mathcal{X}') := \frac{1}{n}\sum_{i=1}^{n}\widehat{r}(\boldsymbol{x}_i) - 1, \tag{8}$$

$$\widetilde{\mathrm{PE}}(\mathcal{X}\|\mathcal{X}') := -\frac{1}{n'}\sum_{i'=1}^{n'}\widehat{r}(\boldsymbol{x}'_{i'})^2 + \frac{2}{n}\sum_{i=1}^{n}\widehat{r}(\boldsymbol{x}_i) - 1. \tag{9}$$

Eq.(8) is suitable for algorithmic development because this would be the simplest expression, while Eq.(9) is suitable for theoretical analysis because this corresponds to the negative of the objective function in Eq.(4).

If the ℓ_2-regularizer

$$\|\boldsymbol{\theta}\|^2 := \sum_{\ell=1}^{n}\theta_\ell^2$$

in Eq.(4) is replaced with the ℓ_1-regularizer

$$\|\boldsymbol{\theta}\|_1 := \sum_{\ell=1}^{n}|\theta_\ell|,$$

the solution tends to be sparse [182]. Then the solution can be obtained in a computationally more efficient way [185], and furthermore a regularization path tracking algorithm [48] is available for efficiently computing solutions with different regularization parameter values.

2.3.3 rPE Divergence Approximation

The rPE divergence can be directly estimated in the same way as the PE divergence [212]:

$$\min_r \int q_\alpha(\boldsymbol{x}') \left(r(\boldsymbol{x}) - \frac{p(\boldsymbol{x})}{q_\alpha(\boldsymbol{x})} \right)^2 \mathrm{d}\boldsymbol{x}.$$

Its empirical criterion where an irrelevant constant is ignored and the expectations are approximated by sample averages is given by

$$\min_r \left[\frac{\alpha}{n} \sum_{i=1}^{n} r^2(\boldsymbol{x}_i) + \frac{1-\alpha}{n'} \sum_{i'=1}^{n'} r^2(\boldsymbol{x}'_{i'}) - \frac{2}{n} \sum_{i=1}^{n} r(\boldsymbol{x}_i) \right].$$

For the Gaussian density-ratio model (3) together with the ℓ_2-regularizer, the above optimization problem is expressed as

$$\min_{\boldsymbol{\theta}} \left[\boldsymbol{\theta}^\top (\alpha \widehat{\boldsymbol{G}} + (1-\alpha)\widehat{\boldsymbol{G}}')\boldsymbol{\theta} - 2\boldsymbol{\theta}^\top \widehat{\boldsymbol{h}} + \lambda \|\boldsymbol{\theta}\|^2 \right],$$

where $\widehat{\boldsymbol{G}}$ is the $n \times n$ matrix with the (ℓ, ℓ')-th element defined by

$$\widehat{G}_{\ell,\ell'} := \frac{1}{n} \sum_{i=1}^{n} \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_\ell\|^2}{2\sigma^2} \right) \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_{\ell'}\|^2}{2\sigma^2} \right).$$

This is a convex optimization problem, and the global optimal solution can be computed analytically as

$$(\alpha \widehat{\boldsymbol{G}} + (1-\alpha)\widehat{\boldsymbol{G}}' + \lambda \boldsymbol{I})^{-1} \widehat{\boldsymbol{h}}.$$

Cross-validation for tuning the Gaussian width σ and the regularization parameter λ can be carried out in the same way as the PE-divergence case, with Eq.(5) replaced by

$$\frac{\alpha}{|\mathcal{X}_t|} \sum_{\boldsymbol{x} \in \mathcal{X}_t} \widehat{r}_t(\boldsymbol{x})^2 + \frac{1-\alpha}{|\mathcal{X}'_t|} \sum_{\boldsymbol{x}' \in \mathcal{X}'_t} \widehat{r}_t(\boldsymbol{x}')^2 - \frac{2}{|\mathcal{X}_t|} \sum_{\boldsymbol{x} \in \mathcal{X}_t} \widehat{r}_t(\boldsymbol{x}).$$

By expanding the squared term $\left(\frac{p(\boldsymbol{x})}{q_\alpha(\boldsymbol{x})} - 1 \right)^2$ in Eq.(2), the rPE divergence can be expressed as

$$\mathrm{rPE} = \int p(\boldsymbol{x}) \frac{p(\boldsymbol{x})}{q_\alpha(\boldsymbol{x})} \mathrm{d}\boldsymbol{x} - 1 \tag{10}$$

$$= -\int q_\alpha(\boldsymbol{x}) \left(\frac{p(\boldsymbol{x})}{q_\alpha(\boldsymbol{x})} \right)^2 \mathrm{d}\boldsymbol{x} + 2 \int p(\boldsymbol{x}) \frac{p(\boldsymbol{x})}{q_\alpha(\boldsymbol{x})} \mathrm{d}\boldsymbol{x} - 1. \tag{11}$$

Based on these expressions, rPE divergence approximators are given using the relative density-ratio estimator $\widehat{r}_\alpha$ as

$$\widehat{\mathrm{rPE}}_\alpha(\mathcal{X}\|\mathcal{X}') := \frac{1}{n}\sum_{i=1}^{n}\widehat{r}_\alpha(\boldsymbol{x}_i) - 1, \tag{12}$$

$$\widetilde{\mathrm{rPE}}_\alpha(\mathcal{X}\|\mathcal{X}') := -\frac{\alpha}{n}\sum_{i=1}^{n}\widehat{r}_\alpha(\boldsymbol{x}_i)^2 - \frac{(1-\alpha)}{n'}\sum_{i'=1}^{n'}\widehat{r}_\alpha(\boldsymbol{x}'_{i'})^2 + \frac{2}{n}\sum_{i=1}^{n}\widehat{r}_\alpha(\boldsymbol{x}_i) - 1. \tag{13}$$

2.3.4 L^2-Distance Approximation

The key idea is to directly estimate the density difference $p - p'$ without estimating each density [172]. More specifically, a density-difference estimator is obtained by minimizing the squared difference between a density-difference model f and the true density-difference function $p - p'$:

$$\min_f \int \Big(f(\boldsymbol{x}) - \big(p(\boldsymbol{x}) - p'(\boldsymbol{x})\big)\Big)^2 \mathrm{d}\boldsymbol{x}.$$

Its empirical criterion where an irrelevant constant is ignored and the expectation is approximated by the sample average is given by

$$\min_f \left[\int f(\boldsymbol{x})^2 \mathrm{d}\boldsymbol{x} - \left(\frac{2}{n}\sum_{i=1}^{n} f(\boldsymbol{x}_i) - \frac{2}{n'}\sum_{i'=1}^{n'} f(\boldsymbol{x}'_{i'})\right)\right].$$

Let us consider the following Gaussian density-difference model:

$$f(\boldsymbol{x}) = \sum_{\ell=1}^{n+n'} \xi_\ell \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{c}_\ell\|^2}{2\sigma^2}\right), \tag{14}$$

where

$$(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_n, \boldsymbol{c}_{n+1}, \ldots, \boldsymbol{c}_{n+n'}) := (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n, \boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_{n'})$$

are Gaussian centers. Then the above optimization problem is expressed as

$$\min_{\boldsymbol{\xi}=(\xi_1,\ldots,\xi_{n+n'})^\top} \left[\boldsymbol{\xi}^\top \boldsymbol{U}\boldsymbol{\xi} - 2\boldsymbol{\xi}^\top \widehat{\boldsymbol{v}} + \lambda\|\boldsymbol{\xi}\|^2\right],$$

where the ℓ_2-regularizer $\lambda\|\boldsymbol{\xi}\|^2$ is included, $\boldsymbol{U}$ is the $(n+n') \times (n+n')$ matrix with the (ℓ,ℓ')-th element defined by

$$\begin{aligned} U_{\ell,\ell'} &:= \int \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{c}_\ell\|^2}{2\sigma^2}\right) \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{c}_{\ell'}\|^2}{2\sigma^2}\right) \mathrm{d}\boldsymbol{x} \\ &= (\pi\sigma^2)^{d/2} \exp\left(-\frac{\|\boldsymbol{c}_\ell - \boldsymbol{c}_{\ell'}\|^2}{4\sigma^2}\right), \end{aligned}$$

d denotes the dimensionality of $\boldsymbol{x}$, and $\widehat{\boldsymbol{v}}$ is the $(n+n')$-dimensional vector with the ℓ-th element defined by

$$\widehat{v}_\ell := \frac{1}{n}\sum_{i=1}^{n}\exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{c}_\ell\|^2}{2\sigma^2}\right) - \frac{1}{n'}\sum_{i'=1}^{n'}\exp\left(-\frac{\|\boldsymbol{x}'_{i'} - \boldsymbol{c}_\ell\|^2}{2\sigma^2}\right).$$

This is a convex optimization problem, and the global optimal solution can be computed *analytically* as

$$(\boldsymbol{U} + \lambda\boldsymbol{I})^{-1}\widehat{\boldsymbol{v}}.$$

The above optimization problem is essentially the same form as least-squares density-ratio approximation for the PE divergence, and therefore least-squares density-difference approximation can enjoy all the computational properties of least-squares density-ratio approximation.

Cross-validation for tuning the Gaussian width σ and the regularization parameter λ can be carried as follows: First, the numerator and denominator samples $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^{n}$ and $\mathcal{X}' = \{\boldsymbol{x}'_{i'}\}_{i'=1}^{n'}$ are divided into T disjoint subsets $\{\mathcal{X}_t\}_{t=1}^{T}$ and $\{\mathcal{X}'_t\}_{t=1}^{T}$, respectively. Then a density-difference estimator $\widehat{f}_t(\boldsymbol{x})$ is obtained using $\mathcal{X}\backslash\mathcal{X}_t$ and $\mathcal{X}'\backslash\mathcal{X}'_t$ (i.e., all samples without $\mathcal{X}_t$ and $\mathcal{X}'_t$), and its objective value for the hold-out samples $\mathcal{X}_t$ and $\mathcal{X}'_t$ is computed:

$$\int \widehat{f}_t(\boldsymbol{x})^2 \mathrm{d}\boldsymbol{x} - \frac{2}{|\mathcal{X}_t|}\sum_{\boldsymbol{x}\in\mathcal{X}_t}\widehat{f}_t(\boldsymbol{x}) + \frac{2}{|\mathcal{X}'_t|}\sum_{\boldsymbol{x}'\in\mathcal{X}'_t}\widehat{f}_t(\boldsymbol{x}').$$

Note that the first term can be computed analytically for the Gaussian density-difference model (14):

$$\int \widehat{f}_t(\boldsymbol{x})^2 \mathrm{d}\boldsymbol{x} = \widehat{\boldsymbol{\xi}}_t^\top \boldsymbol{U}\widehat{\boldsymbol{\xi}}_t,$$

where $\widehat{\boldsymbol{\xi}}_t$ is the parameter vector learned from $\mathcal{X}\backslash\mathcal{X}_t$ and $\mathcal{X}'\backslash\mathcal{X}'_t$.

For an equivalent expression of the L^2-distance,

$$L^2(p,p') = \int f(\boldsymbol{x})p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} - \int f(\boldsymbol{x}')p'(\boldsymbol{x}')\mathrm{d}\boldsymbol{x}',$$

if f is replaced with a density-difference estimator $\widehat{f}$ and approximate the expectations by empirical averages, the following L^2-distance approximator can be obtained:

$$\widehat{\boldsymbol{v}}^\top\widehat{\boldsymbol{\xi}}. \tag{15}$$

Similarly, for another expression

$$L^2(p,p') = \int f(\boldsymbol{x})^2 \mathrm{d}\boldsymbol{x},$$

replacing f with a density-difference estimator $\widehat{f}$ gives another L^2-distance approximator:

$$\widehat{\boldsymbol{\xi}}^\top \boldsymbol{U} \widehat{\boldsymbol{\xi}}. \tag{16}$$

Eq.(15) and Eq.(16) themselves give valid approximations to $L^2(p, p')$, but their linear combination

$$\widehat{L}^2(\mathcal{X}, \mathcal{X}') := 2\widehat{\boldsymbol{v}}^\top \widehat{\boldsymbol{\xi}} - \widehat{\boldsymbol{\xi}}^\top \boldsymbol{U} \widehat{\boldsymbol{\xi}},$$

was shown to have a smaller bias than than Eq.(15) and Eq.(16).

2.4 Usage of Divergence Approximators in Machine Learning

In this section, we show applications of divergence approximators in machine learning.

2.4.1 Change-Detection in Time-Series

The goal is to discover abrupt property changes behind time-series data. Let $\boldsymbol{y}(t) \in \mathbb{R}^m$ be an m-dimensional time-series sample at time t, and let

$$\boldsymbol{Y}(t) := [\boldsymbol{y}(t)^\top, \boldsymbol{y}(t+1)^\top, \ldots, \boldsymbol{y}(t+k-1)^\top]^\top \in \mathbb{R}^{km}$$

be a subsequence of time series at time t with length k. Instead of a single point $\boldsymbol{y}(t)$, the subsequence $\boldsymbol{Y}(t)$ is treated as a sample here, because time-dependent information can be naturally incorporated by this trick [92]. Let

$$\mathcal{Y}(t) := \{\boldsymbol{Y}(t), \boldsymbol{Y}(t+1), \ldots, \boldsymbol{Y}(t+r-1)\}$$

be a set of r retrospective subsequence samples starting at time t. Then a divergence between the probability distributions of $\mathcal{Y}(t)$ and $\mathcal{Y}(t+r)$ may be used as the plausibility of change points (see Figure 1).

In Figure 2, we illustrate results of unsupervised change detection for the *IPSJ SIG-SLP Corpora and Environments for Noisy Speech Recognition* (CENSREC) dataset[1] that records human voice in noisy environments such as a restaurant, and the *Human Activity Sensing Consortium (HASC) challenge 2011* dataset[2] that provides human activity information collected by portable three-axis accelerometers. These graphs show that the KL-based method is excessively sensitive to noise and thus change points are not clearly detected. On the other hand, the L^2-based method more clearly indicates the existence of change points.

It was also demonstrated that divergence-based change-detection methods are useful in event detection from movies [213] and Twitter [112].

[1] http://research.nii.ac.jp/src/en/CENSREC-1-C.html

[2] http://hasc.jp/hc2011/

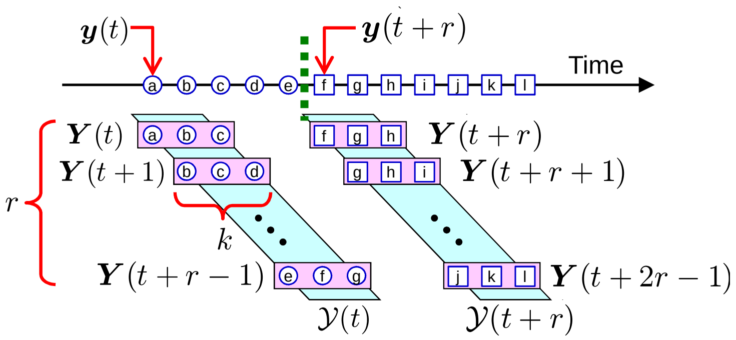


Figure 1: Schematic of change-point detection in time-series.

2.4.2 Salient Object Detection in an Image

The goal is to find salient objects in an image. This can be achieved by computing a divergence between the probability distributions of image features (such as brightness, edges, and colors) in the center window and its surroundings [214]. This divergence computation is swept over the entire image with changing scales (Figure 3).

The object detection results on the *MSRA salient object database* [114] by the rPE divergence with $\alpha = 0.1$ are described in Figure 4, where pixels in gray-scale saliency maps take brighter color if the estimated divergence value is large. The results show that visually salient objects can be successfully detected by the divergence-based approach.

2.4.3 Measuring Statistical Independence

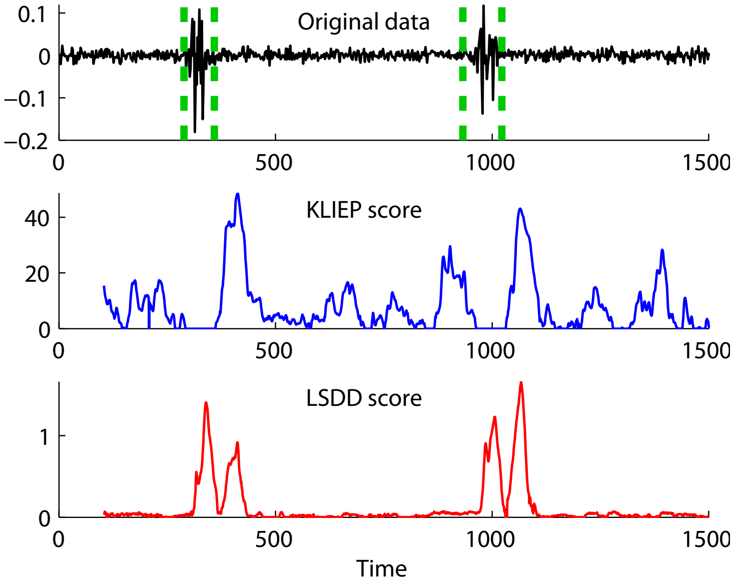
The goal is to measure how strongly two random variables $\boldsymbol{U}$ and $\boldsymbol{V}$ are statistically dependent, from paired samples $\{(\boldsymbol{u}_i, \boldsymbol{v}_i)\}_{i=1}^n$ drawn independently from the joint distribution with density $p_{\mathbf{U},\mathbf{V}}(\boldsymbol{u}, \boldsymbol{v})$. Let us consider a divergence between the joint density $p_{\mathbf{U},\mathbf{V}}$ and the product of marginal densities $p_{\mathbf{U}} \cdot p_{\mathbf{V}}$. This actually serves as a measure of statistical independence, because $\boldsymbol{U}$ and $\boldsymbol{V}$ are independent if and only if the divergence is zero (i.e., $p_{\mathbf{U},\mathbf{V}} = p_{\mathbf{U}} \cdot p_{\mathbf{V}}$), and the dependence between $\boldsymbol{U}$ and $\boldsymbol{V}$ is stronger if the divergence is larger.

Such a dependence measure can be approximated in the same way as ordinary divergences by using the two datasets formed as $\mathcal{X} = \{(\boldsymbol{u}_i, \boldsymbol{v}_i)\}_{i=1}^n$ and $\mathcal{X}' = \{(\boldsymbol{u}_i, \boldsymbol{v}_j)\}_{i,j=1}^n$. The dependence measure based on the KL divergence is called *mutual information* (MI) [147]:

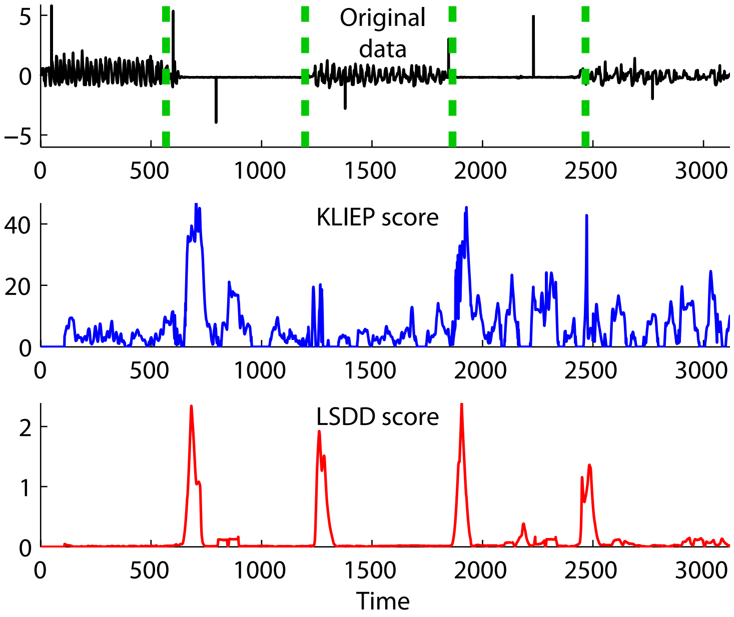
$$\mathrm{MI} := \iint p_{\mathbf{U},\mathbf{V}}(\boldsymbol{u}, \boldsymbol{v}) \log \frac{p_{\mathbf{U},\mathbf{V}}(\boldsymbol{u}, \boldsymbol{v})}{p_{\mathbf{U}}(\boldsymbol{u}) p_{\mathbf{V}}(\boldsymbol{v})} \mathrm{d}\boldsymbol{u}\mathrm{d}\boldsymbol{v}.$$

MI plays a central role in information theory [38].

On the other hand, its PE-divergence variant is called the *squared-loss mutual infor-*



(a) CENSREC dataset



(b) HASC dataset

Figure 2: Results of change-point detection. Original time-series data is plotted in the top graphs, and change scores obtained by KLIEP (Section 2.3.1) and LSDD (Section 2.3.4) are plotted in the bottom graphs.

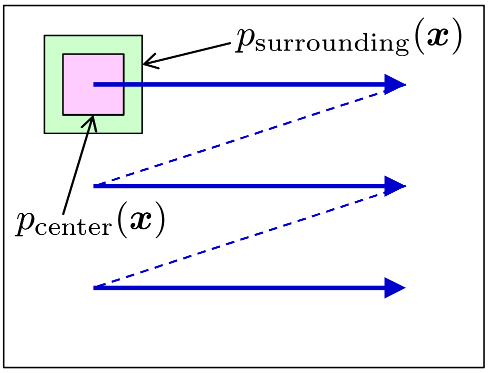

$p_{\text{surrounding}}(\boldsymbol{x})$
$p_{\text{center}}(\boldsymbol{x})$

Figure 3: Schematic of salient object detection in an image.

mation (SMI):

$$\text{SMI} := \iint p_{\mathbf{U}}(\boldsymbol{u}) p_{\mathbf{V}}(\boldsymbol{v}) \left(\frac{p_{\mathbf{U},\mathbf{V}}(\boldsymbol{u},\boldsymbol{v})}{p_{\mathbf{U}}(\boldsymbol{u}) p_{\mathbf{V}}(\boldsymbol{v})} - 1 \right)^2 \mathrm{d}\boldsymbol{u}\mathrm{d}\boldsymbol{v}.$$

SMI is useful for solving various machine learning tasks [157], including independence testing [166], feature selection [179, 77], feature extraction [178, 204], canonical dependency analysis [89], object matching [208], independent component analysis [177], clustering [175, 97], and causal direction estimation [207].

An L^2-distance variant of the dependence measure is called *quadratic mutual information* (QMI) [186]:

$$\text{QMI} := \iint \Big(p_{\mathbf{U},\mathbf{V}}(\boldsymbol{u},\boldsymbol{v}) - p_{\mathbf{U}}(\boldsymbol{u}) p_{\mathbf{V}}(\boldsymbol{v}) \Big)^2 \mathrm{d}\boldsymbol{u}\mathrm{d}\boldsymbol{v}.$$

QMI is also a useful dependence measure in practice [143].

2.5 Conclusions

In this article, we reviewed recent advances in direct divergence approximation. Direct divergence approximators theoretically achieve optimal convergence rates both in parametric and non-parametric cases and experimentally compare favorably with the naive density-estimation counterparts [124, 173, 82, 212, 172].

However, direct divergence approximators still suffer from the *curse of dimensionality*. A possible cure for this problem is to combine them with dimensionality reduction, based on the hope that two probability distributions share some commonality [159, 176, 209]. Further investigating this line would be a promising future direction.

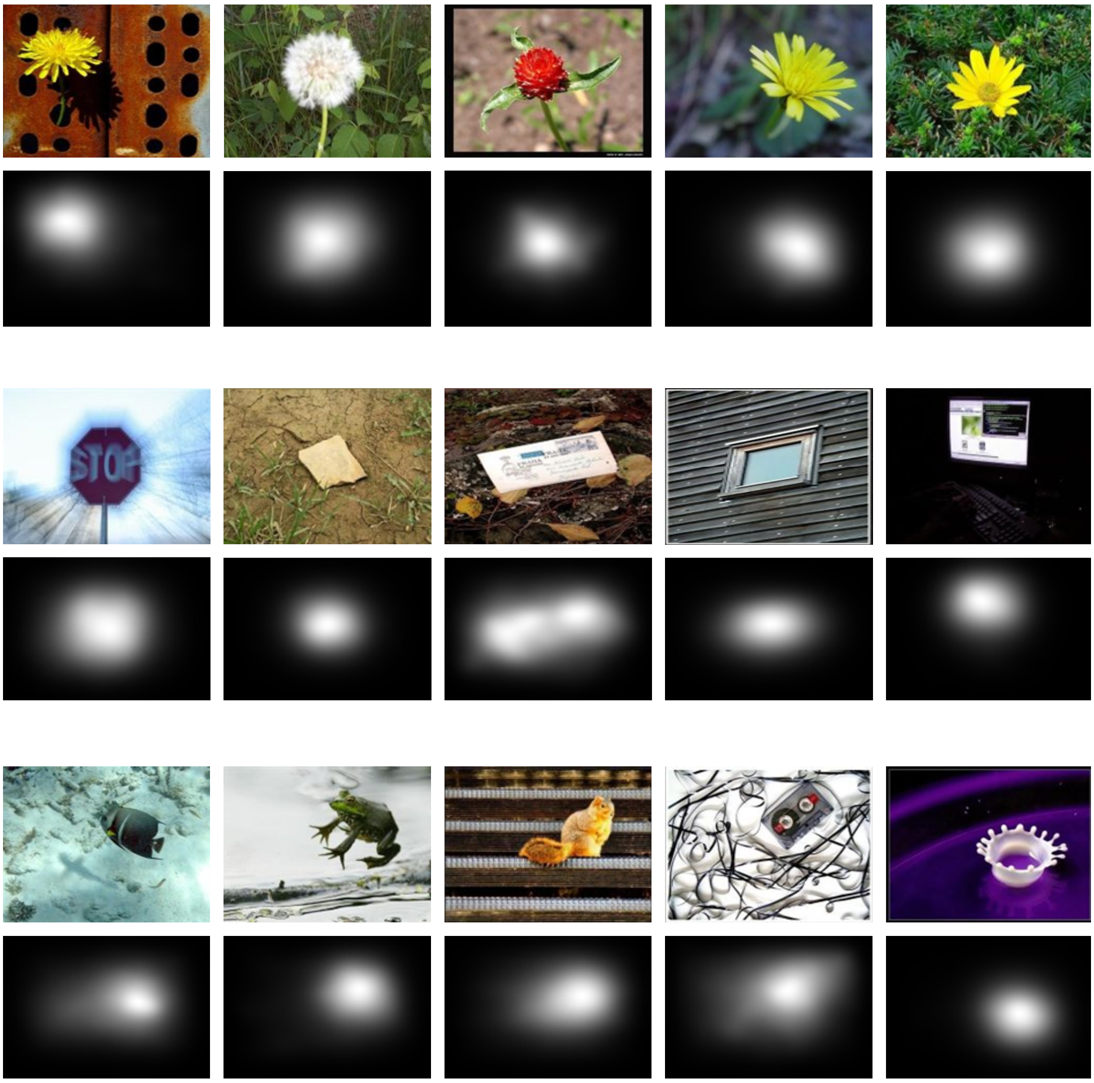


Figure 4: Results of salient object detection in an image. Upper: Original images. Lower: Obtained saliency maps (brighter color means more salient).

3 Change Detection

3.1 Background

Changes in interactions between random variables are interesting in many real-world phenomena. For example, genes may interact with each other in different ways when external stimuli change, co-occurrence between words may appear/disappear when the domains of text corpora shift, and correlation among pixels may change when a surveillance camera captures anomalous activities. Discovering such changes in interactions is a task of great interest in machine learning and data mining, because it provides useful insights into underlying mechanisms in many real-world applications.

In this paper, we consider the problem of detecting changes in conditional independence among random variables between two sets of data. Such conditional independence structure can be expressed via an undirected graphical model called a *Markov network* (MN) [23, 198, 98], where nodes and edges represent variables and their conditional dependencies, respectively. As a simple and widely applicable case, the pairwise MN model has been thoroughly studied recently [136, 105]. Following this line, we also focus on the pairwise MN model as a representative example.

A naive approach to change detection in MNs is the two-step procedure of first estimating two MNs separately from two sets of data by *maximum likelihood estimation* (MLE), and then comparing the structure of the learned MNs. However, MLE is often computationally intractable due to the normalization factor included in the density model. Therefore, Gaussianity is often assumed in practice for computing the normalization factor analytically [70], though this Gaussian assumption is highly restrictive in practice. We may utilize *importance sampling* [139] to numerically compute the normalization factor, but an inappropriate choice of the instrumental distribution may lead to an estimate with high variance [201]; for more discussions on sampling techniques, see references [56] and [72]. References [75] and [63] have explored an alternative approach to avoid computing the normalization factor which are not based on MLE.

However, the two-step procedure has a conceptual weakness that structure change is not directly learned. This indirect nature causes a crucial problem: Suppose that we want to learn a sparse structure change. For learning sparse changes, we may utilize ℓ_1-regularized MLE [16, 52, 105], which produces sparse MNs and thus the change between MNs also becomes sparse. However, this approach does not work if each MN is dense but only change is sparse.

To mitigate this indirect nature, the *fused-lasso* [183] is useful, where two MNs are simultaneously learned with a sparsity-inducing penalty on the *difference* between two MN parameters [218]. Although this fused-lasso approach allows us to learn sparse structure change naturally, the restrictive Gaussian assumption is still necessary to obtain the solution in a computationally tractable way.

The *nonparanormal* assumption [111, 110] is a useful generalization of the Gaussian assumption. A nonparanormal distribution is a *semi-parametric Gaussian copula* where each Gaussian variable is transformed by a monotone non-linear function. Nonparanormal

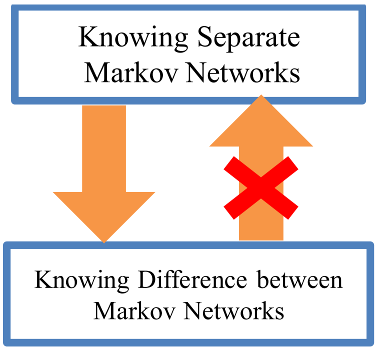

Knowing Separate Markov Networks
Knowing Difference between Markov Networks

Figure 5: The rationale of direct structural change learning: finding the difference between two MNs is a more specific task than finding the entire structures of those two networks, and hence should be possible to learn with less data.

distributions are much more flexible than Gaussian distributions thanks to the feature-wise non-linear transformation, while the normalization factors can still be computed analytically. Thus, the fused-lasso method combined with nonparanormal models would be one of the state-of-the-art approaches to change detection in MNs. However, the fused-lasso method is still based on separate modeling of two MNs, and its computation for more general non-Gaussian distributions is challenging.

In this paper, we propose a more direct approach to structural change learning in MNs based on *density ratio estimation* (DRE) [169]. Our method does not separately model two MNs, but directly models the *change* in two MNs. This idea follows Vapnik's principle [195]:

> If you possess a restricted amount of information for solving some problem, try to solve the problem directly and never solve a more general problem as an intermediate step. It is possible that the available information is sufficient for a direct solution but is insufficient for solving a more general intermediate problem.

This principle was used in the development of *support vector machines* (SVMs): rather than modeling two classes of samples, SVM directly learns a decision boundary that is sufficient for performing pattern recognition. In the current context, estimating two MNs is more general than detecting changes in MNs (Figure 5). By directly detecting changes in MNs, we can also halve the number of parameters, from two MNs to one MN-difference.

Another important advantage of our DRE-based method is that the normalization factor can be approximated efficiently, because the normalization term in a density ratio function takes the form of the expectation over a data distribution and thus it can be simply approximated by the sample average without additional sampling. Through experiments on gene expression and Twitter data analysis, we demonstrate the usefulness of our proposed approach.

The remainder of this paper is structured as follows. In Section 3.2, we formulate the problem of detecting structural changes and review currently available approaches. We then propose our DRE-based structural change detection method in Section 3.3. Results of illustrative and real-world experiments are reported in Section 3.4 and Section 3.5, respectively. Finally, we conclude our work and show the future direction in Section 3.6.

3.2 Problem Formulation and Related Methods

In this section, we formulate the problem of change detection in Markov network structure and review existing approaches.

3.2.1 Problem Formulation

Consider two sets of independent samples drawn separately from two probability distributions P and Q on $\mathbb{R}^d$:

$$\{\boldsymbol{x}_i^P\}_{i=1}^{n_P} \overset{\text{i.i.d.}}{\sim} P \text{ and } \{\boldsymbol{x}_i^Q\}_{i=1}^{n_Q} \overset{\text{i.i.d.}}{\sim} Q.$$

We assume that P and Q belong to the family of *Markov networks* (MNs) consisting of univariate and bivariate factors[3], i.e., their respective probability densities p and q are expressed as

$$p(\boldsymbol{x};\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp\left(\sum_{u,v=1,u\geq v}^{d} \boldsymbol{\theta}_{u,v}^\top \boldsymbol{f}(x^{(u)}, x^{(v)})\right), \tag{17}$$

where $\boldsymbol{x} = (x^{(1)}, \ldots, x^{(d)})^\top$ is the d-dimensional random variable, $\top$ denotes the transpose, $\boldsymbol{\theta}_{u,v}$ is the parameter vector for the elements $x^{(u)}$ and $x^{(v)}$, and

$$\boldsymbol{\theta} = (\boldsymbol{\theta}_{1,1}^\top, \ldots, \boldsymbol{\theta}_{d,1}^\top, \boldsymbol{\theta}_{2,2}^\top, \ldots, \boldsymbol{\theta}_{d,2}^\top, \ldots, \boldsymbol{\theta}_{d,d}^\top)^\top$$

is the entire parameter vector. $\boldsymbol{f}(x^{(u)}, x^{(v)})$ is a bivariate vector-valued basis function. $Z(\boldsymbol{\theta})$ is the normalization factor defined as

$$Z(\boldsymbol{\theta}) = \int \exp\left(\sum_{u,v=1,u\geq v}^{d} \boldsymbol{\theta}_{u,v}^\top \boldsymbol{f}(x^{(u)}, x^{(v)})\right) \mathrm{d}\boldsymbol{x}.$$

$q(\boldsymbol{x};\boldsymbol{\theta})$ is defined in the same way.

Given two densities which can be parameterized using $p(\boldsymbol{x};\boldsymbol{\theta}^P)$ and $q(\boldsymbol{x};\boldsymbol{\theta}^Q)$, our goal is to discover *the changes in parameters* from P to Q, i.e., $\boldsymbol{\theta}^P - \boldsymbol{\theta}^Q$.

[3] Note that the proposed algorithm itself can be applied to *any* MNs containing more than two elements in each factor.

3.2.2 Sparse Maximum Likelihood Estimation and Graphical Lasso

Maximum likelihood estimation (MLE) with group ℓ_1-regularization has been widely used for estimating the sparse structure of MNs [144, 136, 105]:

$$\max_{\boldsymbol{\theta}} \left[\frac{1}{n_P} \sum_{i=1}^{n_P} \log p(\boldsymbol{x}_i^P; \boldsymbol{\theta}) - \lambda \sum_{u,v=1,u\geq v}^{d} \|\boldsymbol{\theta}_{u,v}\| \right], \tag{18}$$

where $\|\cdot\|$ denotes the ℓ_2-norm. As λ increases, $\|\boldsymbol{\theta}_{u,v}\|$ may drop to 0. Thus, this method favors an MN that encodes more conditional independencies among variables.

Computation of the normalization term $Z(\boldsymbol{\theta})$ in Eq.(17) is often computationally intractable when the dimensionality of $\boldsymbol{x}$ is high. To avoid this computational problem, the Gaussian assumption is often imposed [52, 119]. More specifically, the following zero-mean Gaussian model is used:

$$p(\boldsymbol{x}; \boldsymbol{\Theta}) = \frac{\det(\boldsymbol{\Theta})^{1/2}}{(2\pi)^{d/2}} \exp\left(-\frac{1}{2} \boldsymbol{x}^\top \boldsymbol{\Theta} \boldsymbol{x} \right),$$

where $\boldsymbol{\Theta}$ is the inverse covariance matrix (a.k.a. the precision matrix) and $\det(\cdot)$ denotes the determinant. Then $\boldsymbol{\Theta}$ is learned as

$$\max_{\boldsymbol{\Theta}} \left[\log\det(\boldsymbol{\Theta}) - \mathrm{tr}(\boldsymbol{\Theta} \boldsymbol{S}^P) - \lambda \|\boldsymbol{\Theta}\|_1 \right],$$

where $\boldsymbol{S}^P$ is the sample covariance matrix of $\{\boldsymbol{x}_i^P\}_{i=1}^n$. $\|\boldsymbol{\Theta}\|_1$ is the ℓ_1-norm of $\boldsymbol{\Theta}$, i.e., the absolute sum of all elements. This formulation has been studied intensively in [16], and a computationally efficient algorithm called the *graphical lasso* (Glasso) has been proposed [52].

Sparse changes in conditional independence structure between P and Q can be detected by comparing two MNs estimated separately using sparse MLE. However, this approach implicitly assumes that two MNs are sparse, which is not necessarily true even if the change is sparse.

3.2.3 Fused-Lasso (Flasso) Method

To more naturally handle sparse changes in conditional independence structure between P and Q, a method based on *fused-lasso* [183] has been developed [218]. This method directly sparsifies the *difference* between parameters.

The original method conducts *feature-wise neighborhood regression* [119] jointly for P and Q, which can be conceptually understood as maximizing the local conditional Gaussian likelihood jointly on each feature [136]. A slightly more general form of the learning criterion may be summarized as

$$\max_{\boldsymbol{\theta}_s^P, \boldsymbol{\theta}_s^Q} \left[\ell_s^P(\boldsymbol{\theta}_s^P) + \ell_s^Q(\boldsymbol{\theta}_s^Q) - \lambda_1(\|\boldsymbol{\theta}_s^P\|_1 + \|\boldsymbol{\theta}_s^Q\|_1) - \lambda_2 \|\boldsymbol{\theta}_s^P - \boldsymbol{\theta}_s^Q\|_1 \right],$$

where $\ell_s^P(\boldsymbol{\theta})$ is the log conditional likelihood for the s-th element $x^{(s)} \in \mathbb{R}$ given the rest $\boldsymbol{x}^{(-s)} \in \mathbb{R}^{d-1}$:

$$\ell_s^P(\boldsymbol{\theta}) = \frac{1}{n_P} \sum_{i=1}^{n_P} \log p(x_i^{(s)P} | \boldsymbol{x}_i^{(-s)P}; \boldsymbol{\theta}).$$

$\ell_s^Q(\boldsymbol{\theta})$ is defined in the same way as $\ell_s^P(\boldsymbol{\theta})$.

Since the Flasso-based method directly sparsifies the change in MN structure, it can work well even when each MN is not sparse. However, using other models than Gaussian is difficult because of the normalization issue described in Section 3.2.2.

3.2.4 Nonparanormal Extensions

In the above methods, Gaussianity is required in practice to compute the normalization factor efficiently, which is a highly restrictive assumption. To overcome this restriction, it has become popular to perform structure learning under the *nonparanormal* settings [111, 110], where the Gaussian distribution is replaced by a *semi-parametric Gaussian copula.*

A random vector $\boldsymbol{x} = (x^{(1)}, \ldots, x^{(d)})^\top$ is said to follow a *nonparanormal* distribution, if there exists a set of monotone and differentiable functions, $\{h_i(x)\}_{i=1}^d$, such that $\boldsymbol{h}(\boldsymbol{x}) = (h_1(x^{(1)}), \ldots, h_d(x^{(d)}))^\top$ follows the Gaussian distribution. Nonparanormal distributions are much more flexible than Gaussian distributions thanks to the non-linear transformation $\{h_i(x)\}_{i=1}^d$, while the normalization factors can still be computed in an analytical way.

However, the nonparanormal transformation is restricted to be element-wise, which is still restrictive to express complex distributions.

3.2.5 Maximum Likelihood Estimation for Non-Gaussian Models by Importance-Sampling

A numerical way to obtain the MLE solution under general non-Gaussian distributions is *importance sampling.*

Suppose that we try to maximize the log-likelihood[4]:

$$\begin{aligned}\ell_{\mathrm{MLE}}(\boldsymbol{\theta}) &= \frac{1}{n_P} \sum_{i=1}^{n_P} \log p(\boldsymbol{x}_i^P; \boldsymbol{\theta}) \\ &= \frac{1}{n_P} \sum_{i=1}^{n_P} \sum_{u \geq v} \boldsymbol{\theta}_{u,v}^\top \boldsymbol{f}(x_i^{(u)P}, x_i^{(v)P}) - \log \int \exp\left(\sum_{u \geq v} \boldsymbol{\theta}_{u,v}^\top \boldsymbol{f}(x^{(u)}, x^{(v)}) \right) \mathrm{d}\boldsymbol{x}. \quad (19)\end{aligned}$$

The key idea of importance sampling is to compute the integral by the expectation over an easy-to-sample *instrumental density* $p'(\boldsymbol{x})$ (e.g., Gaussian) weighted according to

[4] From here on, we simplify $\sum_{u,v=1,u \geq v}^{d}$ as $\sum_{u \geq v}$.

the *importance* $1/p'(\boldsymbol{x})$. More specifically, using i.i.d. samples $\{\boldsymbol{x}'_i\}_{i=1}^{n'} \overset{\text{i.i.d.}}{\sim} p'(\boldsymbol{x})$, the last term of Eq.(19) can be approximately computed as follows:

$$\begin{aligned}\log \int \exp\left(\sum_{u\geq v} \boldsymbol{\theta}_{u,v}^\top \boldsymbol{f}(x^{(u)}, x^{(v)})\right) \mathrm{d}\boldsymbol{x} &= \log \int p'(\boldsymbol{x}) \frac{\exp\left(\sum_{u\geq v} \boldsymbol{\theta}_{u,v}^\top \boldsymbol{f}(x^{(u)}, x^{(v)})\right)}{p'(\boldsymbol{x})} \mathrm{d}\boldsymbol{x} \\ &\approx \log \frac{1}{n'} \sum_{i=1}^{n'} \frac{\exp\left(\sum_{u\geq v} \boldsymbol{\theta}_{u,v}^\top \boldsymbol{f}(x_i'^{(u)}, x_i'^{(v)})\right)}{p'(\boldsymbol{x}'_i)}.\end{aligned}$$

We refer to this implementation of Glasso as IS-Glasso below.

However, importance sampling tends to produce an estimate with large variance if the instrumental distribution is not carefully chosen. Although it is often suggested to use a density whose shape is similar to the function to be integrated but with thicker tails as p', it is not straightforward in practice to decide which p' to choose, especially when the dimensionality of $\boldsymbol{x}$ is high [201].

We can also consider an importance-sampling version of the Flasso method (which we refer to as IS-Flasso)[5]

$$\max_{\boldsymbol{\theta}^P, \boldsymbol{\theta}^Q} \left[\ell_{\mathrm{MLE}}^P(\boldsymbol{\theta}^P) + \ell_{\mathrm{MLE}}^Q(\boldsymbol{\theta}^Q) - \lambda_1(\|\boldsymbol{\theta}^P\|^2 + \|\boldsymbol{\theta}^Q\|^2) - \lambda_2 \sum_{u\geq v} \|\boldsymbol{\theta}_{u,v}^P - \boldsymbol{\theta}_{u,v}^Q\|\right],$$

where both $\ell_{\mathrm{MLE}}^P(\boldsymbol{\theta}^P)$ and $\ell_{\mathrm{MLE}}^Q(\boldsymbol{\theta}^Q)$ are approximated by importance sampling for non-Gaussian distributions. However, in the same way as IS-Glasso, the choice of instrumental distributions is not straightforward.

3.3 Direct Learning of Structural Changes via Density Ratio Estimation

The Flasso method can more naturally handle sparse changes in MNs than separate sparse MLE. However, the Flasso method is still based on separate modeling of two MNs, and its computation for general high-dimensional non-Gaussian distributions is challenging. In this section, we propose to directly learn structural changes based on *density ratio estimation* [169]. Our approach does not involve separate modeling of each MN and allows us to approximate the normalization term efficiently for *any* distributions.

3.3.1 Density Ratio Formulation for Structural Change Detection

Our key idea is to consider the ratio of p and q:

$$\frac{p(\boldsymbol{x}; \boldsymbol{\theta}^P)}{q(\boldsymbol{x}; \boldsymbol{\theta}^Q)} \propto \exp\left(\sum_{u\geq v} (\boldsymbol{\theta}_{u,v}^P - \boldsymbol{\theta}_{u,v}^Q)^\top \boldsymbol{f}(x^{(u)}, x^{(v)})\right).$$

[5] For implementation simplicity, we maximize the joint likelihood of p and q, instead of its feature-wise conditional likelihood. We also switch the first penalty term from ℓ_1 to ℓ_2.

Here $\boldsymbol{\theta}^P_{u,v} - \boldsymbol{\theta}^Q_{u,v}$ encodes the difference between P and Q for factor $\boldsymbol{f}(x^{(u)}, x^{(v)})$, i.e., $\boldsymbol{\theta}^P_{u,v} - \boldsymbol{\theta}^Q_{u,v}$ is zero if there is no change in the factor $\boldsymbol{f}(x^{(u)}, x^{(v)})$.

Once we consider the ratio of p and q, we actually do not have to estimate $\boldsymbol{\theta}^P_{u,v}$ and $\boldsymbol{\theta}^Q_{u,v}$; instead estimating their difference $\boldsymbol{\theta}_{u,v} = \boldsymbol{\theta}^P_{u,v} - \boldsymbol{\theta}^Q_{u,v}$ is sufficient for change detection:

$$r(\boldsymbol{x}; \boldsymbol{\theta}) = \frac{1}{N(\boldsymbol{\theta})} \exp\left(\sum_{u \geq v} \boldsymbol{\theta}^\top_{u,v} \boldsymbol{f}(x^{(u)}, x^{(v)})\right), \tag{20}$$

where

$$N(\boldsymbol{\theta}) = \int q(\boldsymbol{x}) \exp\left(\sum_{u \geq v} \boldsymbol{\theta}^\top_{u,v} \boldsymbol{f}(x^{(u)}, x^{(v)})\right) \mathrm{d}\boldsymbol{x}.$$

The normalization term $N(\boldsymbol{\theta})$ guarantees[6]

$$\int q(\boldsymbol{x}) r(\boldsymbol{x}; \boldsymbol{\theta}) \mathrm{d}\boldsymbol{x} = 1.$$

Thus, in this density ratio formulation, we are no longer modeling p and q separately, but we model the change from p to q *directly.* This direct nature would be more suitable for change detection purposes according to Vapnik's principle that encourages avoidance of solving more general problems as an intermediate step [195]. This direct formulation also allows us to halve the number of parameters from both $\boldsymbol{\theta}^P$ and $\boldsymbol{\theta}^Q$ to only $\boldsymbol{\theta}$.

Furthermore, the normalization factor $N(\boldsymbol{\theta})$ in the density ratio formulation can be easily approximated by the sample average over $\{\boldsymbol{x}^Q_i\}_{i=1}^{n_Q} \overset{\text{i.i.d.}}{\sim} q(\boldsymbol{x})$, because $N(\boldsymbol{\theta})$ is the

[6] If the model $q(\boldsymbol{x}; \boldsymbol{\theta}^Q)$ is correctly specified, i.e., there exists $\boldsymbol{\theta}^{Q^*}$ such that $q(\boldsymbol{x}; \boldsymbol{\theta}^{Q^*}) = q(\boldsymbol{x})$, then $N(\boldsymbol{\theta})$ can be interpreted as importance sampling of $Z(\boldsymbol{\theta}^P)$ via instrumental distribution $q(\boldsymbol{x})$. Indeed, since

$$Z(\boldsymbol{\theta}^P) = \int q(\boldsymbol{x}) \frac{\exp\left(\sum_{u \geq v} {\boldsymbol{\theta}^P_{u,v}}^\top \boldsymbol{f}(x^{(u)}, x^{(v)})\right)}{q(\boldsymbol{x}; \boldsymbol{\theta}^{Q^*})} \mathrm{d}\boldsymbol{x},$$

where $q(\boldsymbol{x}; \boldsymbol{\theta}^{Q^*}) = q(\boldsymbol{x})$, we have

$$N(\boldsymbol{\theta}^P - \boldsymbol{\theta}^{Q^*}) = \frac{Z(\boldsymbol{\theta}^P)}{Z(\boldsymbol{\theta}^{Q^*})} = \int q(\boldsymbol{x}) \exp\left(\sum_{u \geq v} (\boldsymbol{\theta}^P_{u,v} - {\boldsymbol{\theta}^Q_{u,v}}^*)^\top \boldsymbol{f}(x^{(u)}, x^{(v)})\right) \mathrm{d}\boldsymbol{x}.$$

This is exactly the normalization term $N(\boldsymbol{\theta})$ of the ratio $p(\boldsymbol{x}; \boldsymbol{\theta}^P)/q(\boldsymbol{x}; \boldsymbol{\theta}^{Q^*})$. However, we note that the density ratio estimation method we use in this paper is consistent to the optimal solution in the model even without the correct model assumption [85]. An alternative normalization term,

$$N'(\boldsymbol{\theta}, \boldsymbol{\theta}^Q) = \int q(\boldsymbol{x}; \boldsymbol{\theta}^Q) r(\boldsymbol{x}; \boldsymbol{\theta}) \mathrm{d}\boldsymbol{x},$$

may also be considered, as in the case of MLE. However, this alternative form requires an extra parameter $\boldsymbol{\theta}^Q$ which is not our main interest.

expectation over $q(\boldsymbol{x})$:

$$N(\boldsymbol{\theta}) \approx \frac{1}{n_Q}\sum_{i=1}^{n_Q} \exp\left(\sum_{u\geq v} \boldsymbol{\theta}_{u,v}^\top \boldsymbol{f}(x_i^{(u)Q}, x_i^{(v)Q})\right).$$

3.3.2 Direct Density-Ratio Estimation

Density ratio estimation has been recently introduced to the machine learning community and is proven to be useful in a wide range of applications [169]. Here, we concentrate on the density ratio estimator called the *Kullback-Leibler importance estimation procedure* (KLIEP) for log-linear models [174, 188].

For a density ratio model $r(\boldsymbol{x};\boldsymbol{\theta})$, the KLIEP method minimizes the Kullback-Leibler divergence from $p(\boldsymbol{x})$ to $\widehat{p}(\boldsymbol{x}) = q(\boldsymbol{x})r(\boldsymbol{x};\boldsymbol{\theta})$:

$$\begin{aligned}\mathrm{KL}[p\|\widehat{p}] &= \int p(\boldsymbol{x}) \log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})r(\boldsymbol{x};\boldsymbol{\theta})}\mathrm{d}\boldsymbol{x} \\ &= \mathrm{Const.} - \int p(\boldsymbol{x}) \log r(\boldsymbol{x};\boldsymbol{\theta})\mathrm{d}\boldsymbol{x}. \end{aligned} \tag{21}$$

Note that our density-ratio model (20) automatically satisfies the non-negativity and normalization constraints:

$$r(\boldsymbol{x};\boldsymbol{\theta}) \geq 0 \quad \text{and} \quad \int q(\boldsymbol{x})r(\boldsymbol{x};\boldsymbol{\theta})\mathrm{d}\boldsymbol{x} = 1.$$

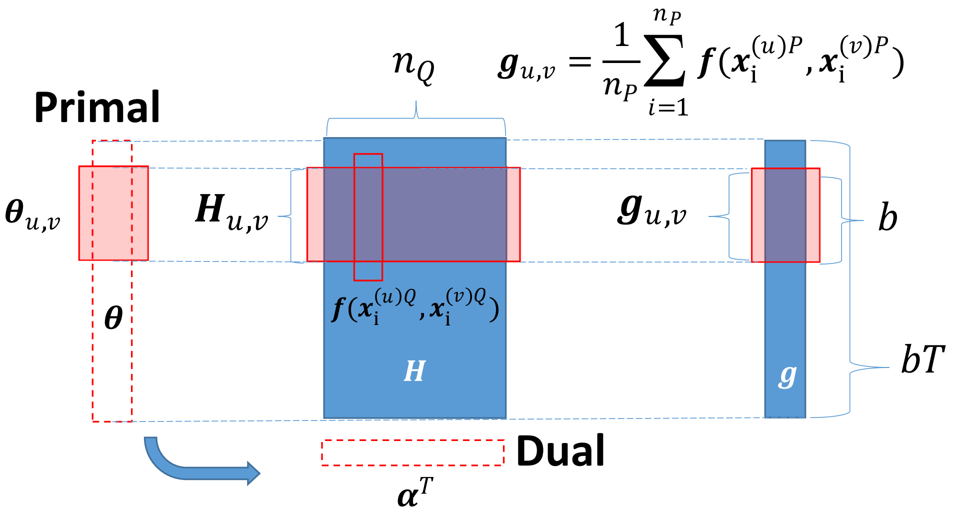
In practice, we maximize the empirical approximation of the second term in Eq.(21):

$$\begin{aligned}\ell_{\mathrm{KLIEP}}(\boldsymbol{\theta}) &= \frac{1}{n_P}\sum_{i=1}^{n_P} \log r(\boldsymbol{x}_i^P;\boldsymbol{\theta}) \\ &= \frac{1}{n_P}\sum_{i=1}^{n_P}\sum_{u\geq v} \boldsymbol{\theta}_{u,v}^\top \boldsymbol{f}(x_i^{(u)P}, x_i^{(v)P}) \\ &\quad - \log\left(\frac{1}{n_Q}\sum_{i=1}^{n_Q} \exp\left(\sum_{u\geq v} \boldsymbol{\theta}_{u,v}^\top \boldsymbol{f}(x_i^{(u)Q}, x_i^{(v)Q})\right)\right).\end{aligned}$$

Because $\ell_{\mathrm{KLIEP}}(\boldsymbol{\theta})$ is concave with respect to $\boldsymbol{\theta}$, its global maximizer can be numerically found by standard optimization techniques such as gradient ascent or quasi-Newton methods. The gradient of ℓ_{KLIEP} with respect to $\boldsymbol{\theta}_{u,v}$ is given by

$$\begin{aligned}\nabla_{\boldsymbol{\theta}_{u,v}} \ell_{\mathrm{KLIEP}}(\boldsymbol{\theta}) =& \frac{1}{n_P}\sum_{i=1}^{n_P} \boldsymbol{f}(\boldsymbol{x}_i^{(u)P}, \boldsymbol{x}_i^{(v)P}) \\ &- \frac{\frac{1}{n_Q}\sum_{i=1}^{n_Q} \exp\left(\sum_{u'\geq v'} \boldsymbol{\theta}_{u',v'}^\top \boldsymbol{f}(x_i^{(u')Q}, x_i^{(v')Q})\right) \boldsymbol{f}(x_i^{(u)Q}, x_i^{(v)Q})}{\frac{1}{n_Q}\sum_{j=1}^{n_Q} \exp\left(\sum_{u''\geq v''} \boldsymbol{\theta}_{u'',v''}^\top \boldsymbol{f}(x_j^{(u'')Q}, x_j^{(v'')Q})\right)},\end{aligned}$$

which can be computed in a straightforward manner for *any* feature vector $\boldsymbol{f}(x^{(u)}, x^{(v)})$.



n_Q
$g_{u,v} = \frac{1}{n_P}\sum_{i=1}^{n_P} f(x_i^{(u)P}, x_i^{(v)P})$
Primal
$\theta_{u,v}$
$H_{u,v}$
$g_{u,v}$
b
θ
$f(x_i^{(u)Q}, x_i^{(v)Q})$
H
g
bT
Dual
α^T

Figure 6: Schematics of primal and dual optimization. b denotes the number of basis functions and T denotes the number of factors. Because we are considering pairwise factors, $T = \mathcal{O}(d^2)$ for input dimensionality d.

3.3.3 Sparsity-Inducing Norm

To find a sparse change between P and Q, we propose to regularize the KLIEP solution with a sparsity-inducing norm $\sum_{u \geq v} \|\boldsymbol{\theta}_{u,v}\|$. Note that the MLE approach sparsifies both $\boldsymbol{\theta}^P$ and $\boldsymbol{\theta}^Q$ so that the difference $\boldsymbol{\theta}^P - \boldsymbol{\theta}^Q$ is also sparsified, while we directly sparsify the difference $\boldsymbol{\theta}^P - \boldsymbol{\theta}^Q$; thus our method can still work well even if $\boldsymbol{\theta}^P$ and $\boldsymbol{\theta}^Q$ are dense.

In practice, we may use the following *elastic-net* penalty [222] to better control overfitting to noisy data:

$$\max_{\boldsymbol{\theta}} \left[\ell_{\text{KLIEP}}(\boldsymbol{\theta}) - \lambda_1 \|\boldsymbol{\theta}\|^2 - \lambda_2 \sum_{u \geq v} \|\boldsymbol{\theta}_{u,v}\| \right], \tag{22}$$

where $\|\boldsymbol{\theta}\|^2$ penalizes the magnitude of the entire parameter vector.

3.3.4 Dual Formulation for High-Dimensional Data

The solution of the optimization problem (22) can be easily obtained by standard sparse optimization methods. However, in the case where the input dimensionality d is high (which is often the case in our setup), the dimensionality of parameter vector $\boldsymbol{\theta}$ is large, and thus obtaining the solution can be computationally expensive. Here, we derive a dual optimization problem [27], which can be solved more efficiently for high-dimensional $\boldsymbol{\theta}$ (Figure 6).

As detailed in Appendix, the dual optimization problem is given as

$$\min_{\boldsymbol{\alpha}=(\alpha_1,\ldots,\alpha_{n_Q})^\top} \sum_{i=1}^{n_Q} \alpha_i \log \alpha_i + \frac{1}{\lambda_1} \sum_{u \geq v} \max(0, \|\boldsymbol{\xi}_{u,v}\| - \lambda_2)^2$$

$$\text{subject to } \alpha_1, \ldots, \alpha_{n_Q} \geq 0 \text{ and } \sum_{i=1}^{n_Q} \alpha_i = 1, \tag{23}$$

where

$$\begin{aligned}
\boldsymbol{\xi}_{u,v} &= \boldsymbol{g}_{u,v} - \boldsymbol{H}_{u,v}\boldsymbol{\alpha}, \\
\boldsymbol{H}_{u,v} &= [\boldsymbol{f}(x_1^{(u)Q}, x_1^{(v)Q}), \ldots, \boldsymbol{f}(x_{n_Q}^{(u)Q}, x_{n_Q}^{(v)Q})], \\
\boldsymbol{g}_{u,v} &= \frac{1}{n_P} \sum_{i=1}^{n_P} \boldsymbol{f}(x_i^{(u)P}, x_i^{(v)P}).
\end{aligned}$$

The primal solution can be obtained from the dual solution as

$$\boldsymbol{\theta}_{u,v} = \begin{cases} \dfrac{1}{\lambda_1}\left(1 - \dfrac{\lambda_2}{\|\boldsymbol{\xi}_{u,v}\|}\right)\boldsymbol{\xi}_{u,v} & \text{if } \|\boldsymbol{\xi}_{u,v}\| > \lambda_2, \\ \mathbf{0} & \text{if } \|\boldsymbol{\xi}_{u,v}\| \leq \lambda_2. \end{cases} \tag{24}$$

Note that the dimensionality of the dual variable $\boldsymbol{\alpha}$ is equal to n_Q, while that of $\boldsymbol{\theta}$ is quadratic with respect to the input dimensionality d, because we are considering pairwise factors. Thus, if d is not small and n_Q is not very large (which is often the case in our experiments shown later), solving the dual optimization problem would be computationally more efficient. Furthermore, the dual objective (and its gradient) can be computed efficiently in parallel for each (u, v), which is a useful property when handling large-scale MNs. Note that the dual objective is differentiable everywhere, while the primal objective is not.

3.4 Numerical Experiments

In this section, we compare the performance of the proposed KLIEP-based method, the Flasso method, and the Glasso method for Gaussian models, nonparanormal models, and non-Gaussian models. Results are reported on datasets with three different underlying distributions: multivariate Gaussian, nonparanormal, and non-Gaussian "diamond" distributions. We also investigate the computation time of the primal and dual formulations as a function of the input dimensionality.

3.4.1 Gaussian Distribution

First, we investigate the performance of each method under Gaussianity.

Consider a 40-node sparse Gaussian MN, where its graphical structure is characterized by precision matrix $\boldsymbol{\Theta}^P$ with diagonal elements equal to 2. The off-diagonal elements are randomly chosen[7] and set to 0.2, so that the overall sparsity of $\boldsymbol{\Theta}^P$ is 25%. We then introduce changes by randomly picking 15 edges and reducing the corresponding elements in the precision matrix by 0.1. The resulting precision matrices $\boldsymbol{\Theta}^P$ and $\boldsymbol{\Theta}^Q$ are used for drawing samples as

$$\{\boldsymbol{x}_i^P\}_{i=1}^{n_P} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, (\boldsymbol{\Theta}^P)^{-1}) \quad \text{and} \quad \{\boldsymbol{x}_i^Q\}_{i=1}^{n_Q} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, (\boldsymbol{\Theta}^Q)^{-1}),$$

where $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Datasets of size $n = n_P = n_Q = 50, 100$ are tested.

We compare the performance of the KLIEP, Flasso, and Glasso methods. Because all methods use the same Gaussian model, the difference in performance is caused only by the difference in estimation methods. We repeat the experiments 20 times with randomly generated datasets and report the results in Figure 7.

The top 6 graphs are examples of regularization paths[8]. The dashed lines represent changed edges in the ground truth, while the solid lines represent unchanged edges. The top row is for $n = 100$ while the middle row is for $n = 50$. The bottom 3 graphs are the data generating distribution and averaged precision-recall (P-R) curves with standard error over 20 runs. The P-R curves are plotted by varying the group-sparsity control parameter λ_2 with $\lambda_1 = 0$ in KLIEP and Flasso, and by varying the sparsity control parameters as $\lambda = \lambda^P = \lambda^Q$ in Glasso.

In the regularization path plots, solid vertical lines show the regularization parameter values picked based on hold-out data $\{\widetilde{\boldsymbol{x}}_i^P\}_{i=1}^{3000} \overset{\text{i.i.d.}}{\sim} P$ and $\{\widetilde{\boldsymbol{x}}_i^Q\}_{i=1}^{3000} \overset{\text{i.i.d.}}{\sim} Q$ as follows:

- **KLIEP:** The *hold-out log-likelihood* (HOLL) is maximized:

$$\frac{1}{\widetilde{n}_P}\sum_{i=1}^{\widetilde{n}_P} \log \frac{\exp\left(\sum_{u\geq v} \widehat{\boldsymbol{\theta}}_{u,v}^\top \boldsymbol{f}(\widetilde{x}_i^{(u)P}, \widetilde{x}_i^{(v)P})\right)}{\frac{1}{\widetilde{n}_Q}\sum_{j=1}^{\widetilde{n}_Q} \exp\left(\sum_{u'\geq v'} \widehat{\boldsymbol{\theta}}_{u',v'}^\top \boldsymbol{f}(\widetilde{x}_j^{(u')Q}, \widetilde{x}_j^{(v')Q})\right)}.$$

- **Flasso:** The sum of feature-wise conditional HOLLs for $p(x^{(s)}|\boldsymbol{x}^{(-s)}; \boldsymbol{\theta}_s)$ and $q(x^{(s)}|\boldsymbol{x}^{(-s)}; \boldsymbol{\theta}_s)$ over all nodes is maximized:

$$\frac{1}{\widetilde{n}_P}\sum_{i=1}^{\widetilde{n}_P}\sum_{s=1}^{d} \log p(\widetilde{x}_i^{(s)P}|\widetilde{\boldsymbol{x}}_i^{(-s)P}; \widehat{\boldsymbol{\theta}}_s^P) + \frac{1}{\widetilde{n}_Q}\sum_{i=1}^{\widetilde{n}_Q}\sum_{s=1}^{d} \log q(\widetilde{x}_i^{(s)Q}|\widetilde{\boldsymbol{x}}_i^{(-s)Q}; \widehat{\boldsymbol{\theta}}_s^Q).$$

- **Glasso:** The sum of HOLLs for $p(\boldsymbol{x}; \boldsymbol{\theta})$ and $q(\boldsymbol{x}; \boldsymbol{\theta})$ is maximized:

$$\frac{1}{\widetilde{n}_P}\sum_{i=1}^{\widetilde{n}_P} \log p(\widetilde{\boldsymbol{x}}_i^P; \widehat{\boldsymbol{\theta}}^P) + \frac{1}{\widetilde{n}_Q}\sum_{i=1}^{\widetilde{n}_Q} \log q(\widetilde{\boldsymbol{x}}_i^Q; \widehat{\boldsymbol{\theta}}^Q).$$

[7] We set $\Theta_{u,v} = \Theta_{v,u}$ for not breaking the symmetry of the precision matrix.

[8] Paths of univariate factors are omitted for clear visibility.

When $n = 100$, KLIEP and Flasso clearly distinguish changed (dashed lines) and unchanged (solid lines) edges in terms of parameter magnitude. However, when the sample size is halved to $n = 50$, the separation is visually rather unclear in the case of Flasso. In contrast, the paths of changed and unchanged edges are still almost disjoint in the case of KLIEP. The Glasso method performs rather poorly in both cases. A similar tendency can be observed also in the P-R curve plot: When the sample size is $n = 100$, KLIEP and Flasso work equally well, but KLIEP gains its lead when the sample size is reduced to $n = 50$. Glasso does not perform well in both cases.

3.4.2 Nonparanormal Distribution

We post-process the Gaussian dataset used in Section 3.4.1 to construct nonparanormal samples. More specifically, we apply the power function,

$$h_i^{-1}(x) = \text{sign}(x)|x|^{\frac{1}{2}},$$

to each dimension of $\boldsymbol{x}^P$ and $\boldsymbol{x}^Q$, so that $\boldsymbol{h}(\boldsymbol{x}^P) \sim \mathcal{N}(\boldsymbol{0}, (\boldsymbol{\Theta}^P)^{-1})$ and $\boldsymbol{h}(\boldsymbol{x}^Q) \sim \mathcal{N}(\boldsymbol{0}, (\boldsymbol{\Theta}^Q)^{-1})$.

To cope with the non-linearity in the KLIEP method, we use the power nonparanormal basis functions with power $k = 2$, 3, and 4:

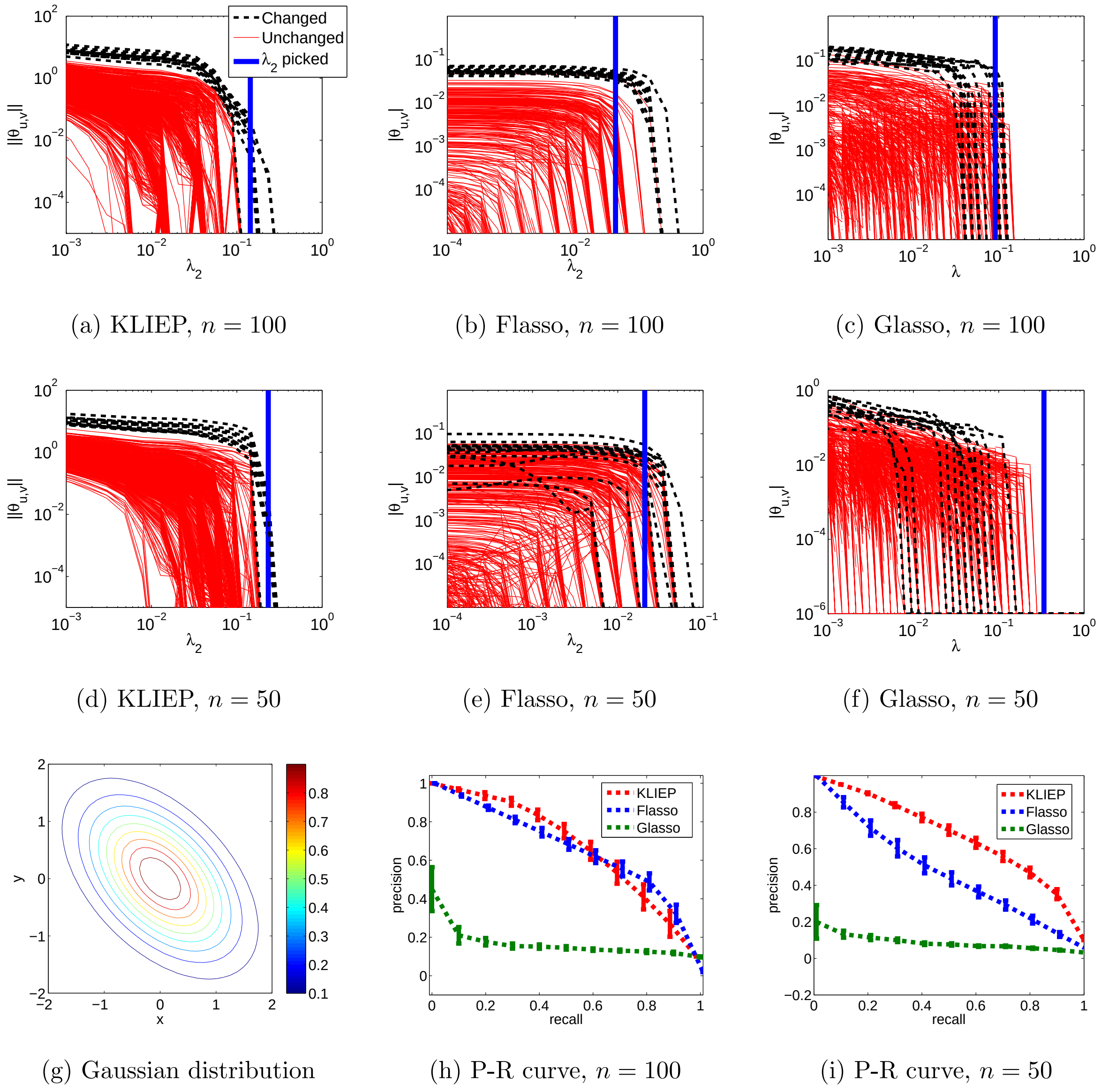
$$\boldsymbol{f}(x_i, x_j) = (\text{sign}(x_i)|x_i|^k, \text{sign}(x_j)|x_j|^k, 1)^\top.$$

Model selection of k is performed together with the regularization parameter by HOLL maximization. For Flasso and Glasso, we apply the nonparanormal transform as described in [111] before the structural change is learned.

The experiments are conducted on 20 randomly generated datasets with $n = 50$ and 100, respectively. The regularization paths, data generating distribution, and averaged P-R curves are plotted in Figure 8. The results show that Flasso clearly suffers from the performance degradation compared with the Gaussian case, perhaps because the number of samples is too small for the complicated nonparanormal distribution. Due to the two-step estimation scheme, the performance of Glasso is poor. In contrast, KLIEP separates changed and unchanged edges still clearly for both $n = 50$ and $n = 100$. The P-R curves also show the same tendency.

3.4.3 “Diamond” Distribution with No Pearson Correlation

In the experiments in Section 3.4.2, though samples are non-Gaussian, the *Pearson correlation* is not zero. Therefore, methods assuming Gaussianity can still capture some linear correlation between random variables. Here, we consider a more challenging case with a diamond-shaped distribution within the exponential family that has zero Pearson correlation between variables. Thus, the methods assuming Gaussianity cannot extract any information in principle from this dataset.



(a) KLIEP, $n = 100$

(b) Flasso, $n = 100$

(c) Glasso, $n = 100$

(d) KLIEP, $n = 50$

(e) Flasso, $n = 50$

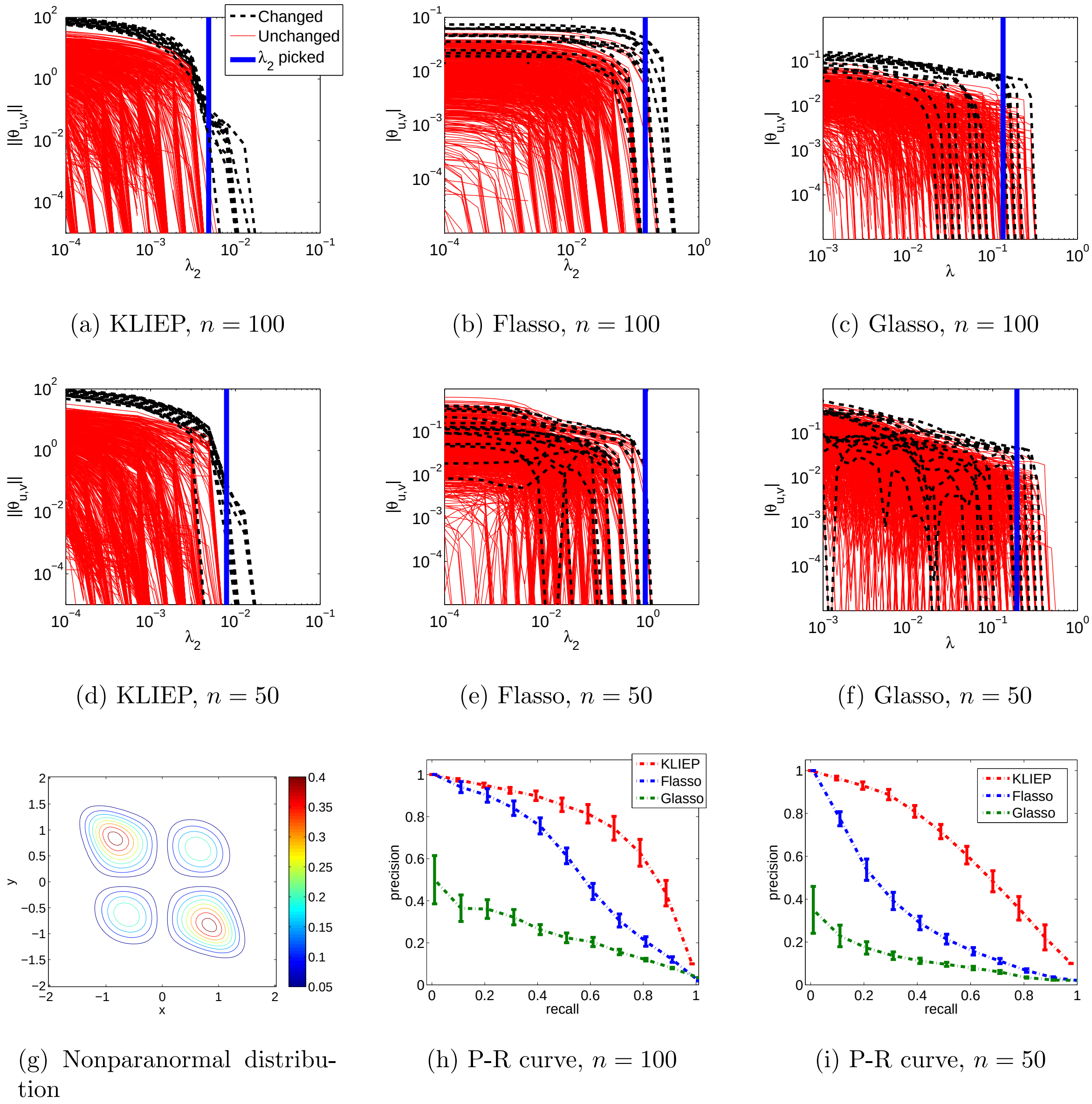
(f) Glasso, $n = 50$

(g) Gaussian distribution

(h) P-R curve, $n = 100$

(i) P-R curve, $n = 50$

Figure 7: Experimental results on the Gaussian dataset.



(a) KLIEP, $n = 100$

(b) Flasso, $n = 100$

(c) Glasso, $n = 100$

(d) KLIEP, $n = 50$

(e) Flasso, $n = 50$

(f) Glasso, $n = 50$

(g) Nonparanormal distribution

(h) P-R curve, $n = 100$

(i) P-R curve, $n = 50$

Figure 8: Experimental results on the nonparanormal dataset.

The probability density function of the diamond distribution is defined as follows (Figure 9(a)):

$$p(\boldsymbol{x}) \propto \exp\left(-\sum_{i=1}^{d} 2x_i^2 - \sum_{(i,j):A_{i,j}\neq 0} 20x_i^2 x_j^2\right), \tag{25}$$

where the adjacency matrix $\boldsymbol{A}$ describes the MN structure. Note that this distribution cannot be transformed into a Gaussian distribution by any nonparanormal transformations.

We set $d = 9$ and $n_P = n_Q = 5000$. $\boldsymbol{A}^P$ is randomly generated with 35% sparsity, while $\boldsymbol{A}^Q$ is created by randomly removing edges in $\boldsymbol{A}^P$ so that the sparsity level is dropped to 15%. Samples from the above distribution are drawn by using a *slice sampling* method [122]. Since generating samples from high-dimensional distributions is non-trivial and time-consuming, we focus on a relatively low-dimensional case. To avoid sampling error which may mislead the experimental evaluation, we also increase the sample size, so that the erratic points generated by accident will not affect the overall population.

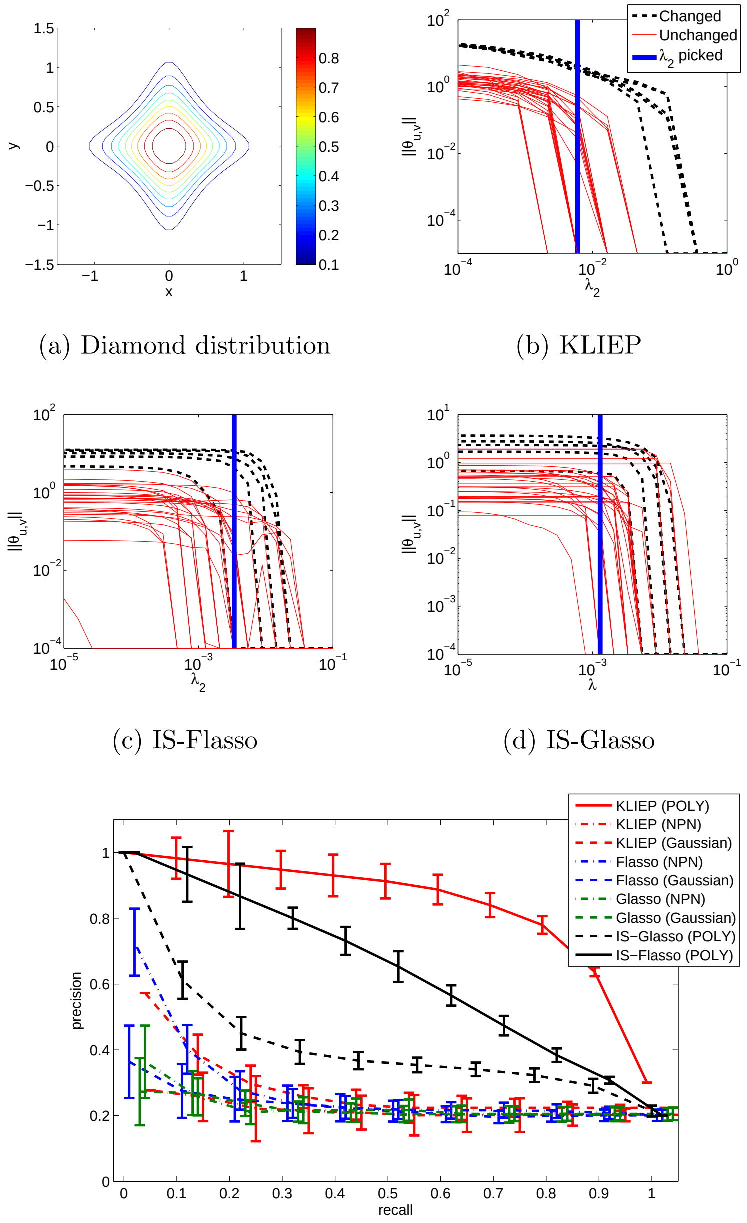
In this experiment, we compare the performance of KLIEP, Flasso, and Glasso with the Gaussian model, the power nonparanormal model, and the polynomial model:

$$\boldsymbol{f}(x_i, x_j) = (x_i^k, x_j^k, x_i x_j^{k-1}, \ldots, x_i^{k-1}x_j, x_i^{k-1}, x_j^{k-1}, \ldots, x_i, x_j, 1)^\top \text{ for } i \neq j.$$

The univariate polynomial transform is defined as $\boldsymbol{f}(x_i, x_i) = \boldsymbol{f}(x_i, 0)$. We test $k = 2, 3, 4$ and choose the best one in terms of HOLL. The Flasso and Glasso methods for the polynomial model are computed by importance sampling, i.e., we use the IS-Flasso and IS-Glasso methods (see Section 3.2.5). Since these methods are computationally very expensive, we only test $k = 4$ which we found to be a reasonable choice. We set the instrumental distribution p' as the standard normal $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, and use sample $\{\boldsymbol{x}'_i\}_{i=1}^{70000} \sim p'$ for approximating integrals. p' is purposely chosen so that it has a similar "bell" shape to the target densities but with larger variance on each dimension.

The averaged P-R curves over 20 datasets are shown in Figure 9(e). KLIEP with the polynomial model significantly outperforms all the other methods, while the IS-Glasso and especially IS-Flasso give better result than the KLIEP, Flasso, and Glasso methods with the Gaussian and nonparanormal models. This means that the polynomial basis function is indeed helpful in handling completely non-Gaussian data. However, as discussed in Section 3.2.2, it is difficult to use such a basis function in Glasso and Flasso because of the computational intractability of the normalization term. Although IS-Glasso can approximate integrals, the result shows that such approximation of integrals does not lead to a very good performance. In comparison, the result of the IS-Flasso method is much improved thanks to the coupled sparsity regularization, but it is still not comparable to KLIEP.

The regularization paths of KLIEP with the polynomial model illustrated in Figure 9(b) show the usefulness of the proposed method in change detection under non-Gaussianity. We also give regularization paths obtained by the IS-Flasso and IS-Glasso



(a) Diamond distribution

(b) KLIEP

(c) IS-Flasso

(d) IS-Glasso

(e) P-R curve

Figure 9: Experimental results on the diamond dataset. “NPN” and “POLY” denote the nonparanormal and polynomial models, respectively. Note that the precision rate of 100% recall for a random guess is approximately 20%.

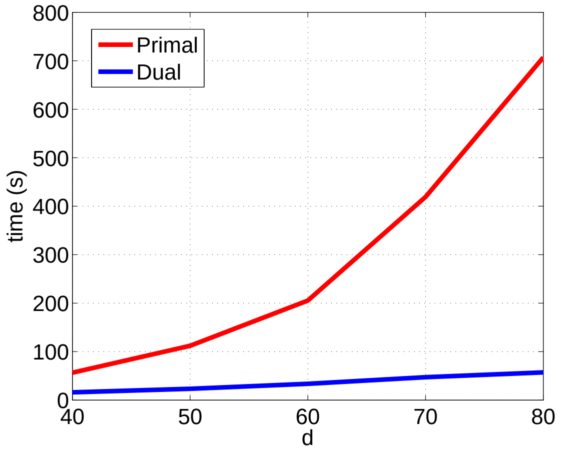


Figure 10: Comparison of computation time for solving primal and dual optimization problems.

methods on the same dataset in Figures 9(c) and 9(d), respectively. The graphs show that both methods do not separate changed and unchanged edges well, though the IS-Flasso method works slightly better.

3.4.4 Computation Time: Dual versus Primal Optimization Problems

Finally, we compare the computation time of the proposed KLIEP method when solving the dual optimization problem (23) and the primal optimization problem (22). Both the optimization problems are solved by using the same convex optimizer *minFunc*[9]. The datasets are generated from two Gaussian distributions constructed in the same way as Section 3.4.1. 150 samples are separately drawn from two distributions with dimension $d = 40, 50, 60, 70, 80$. We then perform change detection by computing the regularization paths using 20 choices of λ_2 ranging from 10^{-4} to 10^{0} and fix $\lambda_1 = 0.1$. The results are plotted in Figure 10.

It can be seen from the graph that as the dimensionality increases, the computation time for solving the primal optimization problem is sharply increased, while that for solving the dual optimization problem grows only moderately: when $d = 80$, the computation time for obtaining the primal solution is almost 10 times more than that required for obtaining the dual solution. Thus, the dual formulation is computationally much more efficient than the primal formulation.

3.5 Applications

In this section, we report the experimental results on a synthetic gene expression dataset and a Twitter dataset.

[9] `http://www.di.ens.fr/~mschmidt/Software/minFunc.html`

3.5.1 Synthetic Gene Expression Dataset

A gene regulatory network encodes interactions between DNA segments. However, the way genes interact may change due to environmental or biological stimuli. In this experiment, we focus on detecting such changes. We use *SynTReN*, which is a generator of gene regulatory networks used for benchmark validation of bioinformatics algorithms [192].

We first choose a sub-network containing 13 nodes from an existing signaling network in *Saccharomyces cerevisiae* (shown in Figure 11(a)). Three types of interactions are modeled: activation (ac), deactivation (re), and dual (du). 50 samples are generated in the first stage, after which we change the types of interactions in 6 edges, and generate 50 samples again. Four types of changes are considered: ac → re, re → ac, du → ac, and du → re.

We use KLIEP and IS-Flasso with the polynomial transform function for $k \in \{2, 3, 4\}$. The regularization parameter λ_1 in KLIEP and Flasso is tested with choices $\lambda_1 \in \{0.1, 1, 10\}$. We set the instrumental distribution p' as the standard normal $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$, and use sample $\{\boldsymbol{x}'_i\}_{i=1}^{70000} \sim p'$ for approximating integrals in IS-Flasso.

The regularization paths on one example dataset for KLIEP, IS-Flasso, and the plain Flasso with the Gaussian model are plotted in Figures 11(b), 11(c), and 11(d), respectively. Averaged P-R curves over 20 simulation runs are shown in Figure 11(e). We can see clearly from the KLIEP regularization paths shown in Figure 11(b) that the magnitude of estimated parameters on the changed pairwise interactions is much higher than that of the unchanged edges. IS-Flasso also achieves rather clear separation between changed and unchanged interactions, though there are a few unchanged interactions drop to zero at the final stage. Flasso gives many false alarms by assigning non-zero values to the unchanged edges, even after some changed edges hit zeros.

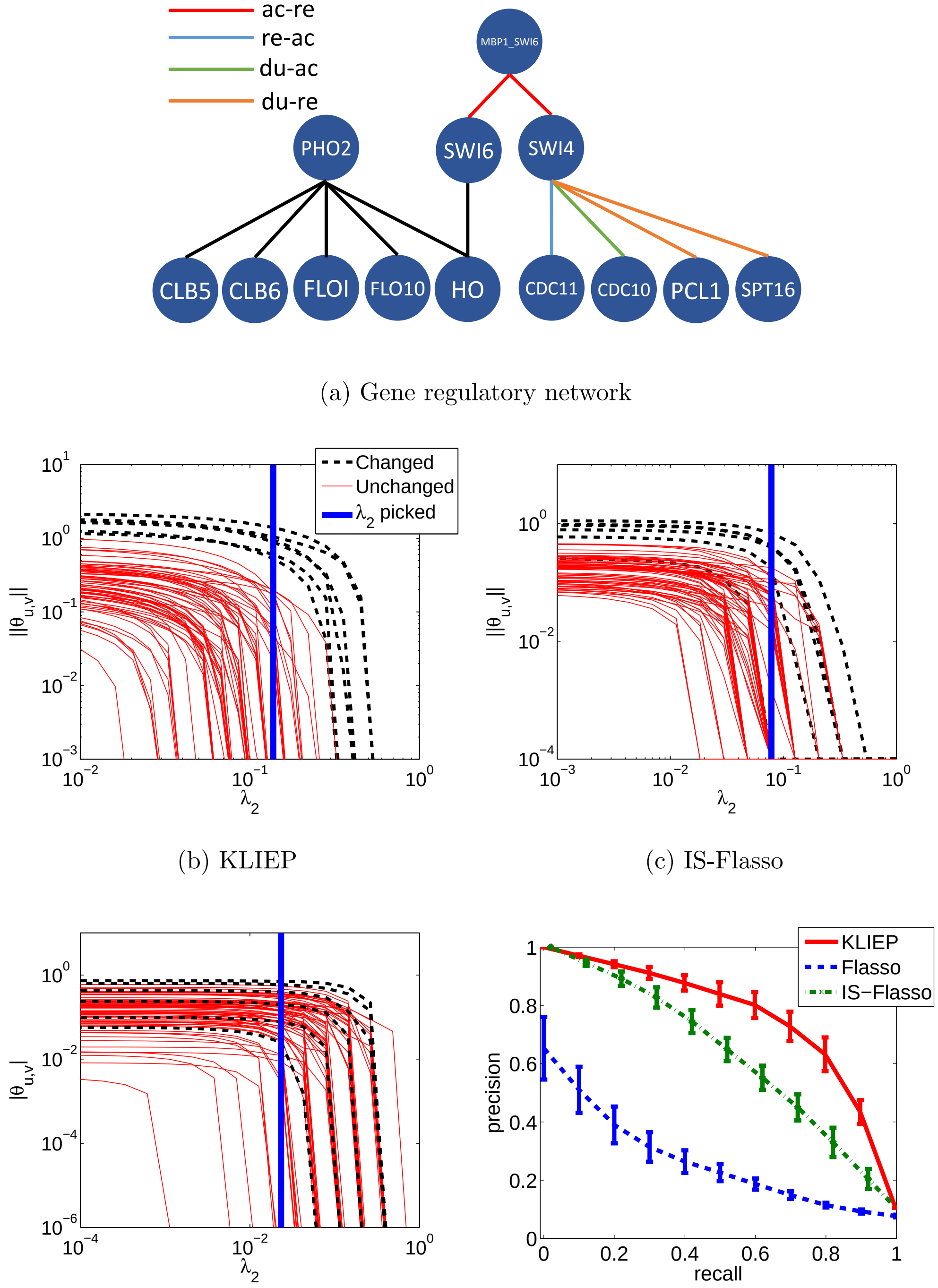
Reflecting a similar pattern, the P-R curves plotted in Figure 11(e) show that the proposed KLIEP method has the best performance among all three methods. We can also see that the IS-Flasso method achieves significant improvement over the plain Flasso method with the Gaussian model. The improvement from Flasso to IS-Flasso shows that the use of the polynomial basis is useful on this dataset, and the improvement from IS-Flasso to KLIEP shows that the direct estimation can further boost the performance.

3.5.2 Twitter Story Telling

Finally, we use KLIEP and Flasso as event detectors from Twitter. More specifically, we choose the *Deepwater Horizon oil spill*[10] as the target event, and we hope that our method can recover some story lines from Twitter as the news events develop. Counting the frequencies of 10 keywords (BP, oil, spill, Mexico, gulf, coast, Hayward, Halliburton, Transocean, and Obama), we obtain a dataset by sampling 4 times per day from February 1st, 2010 to October 15th, 2010, resulting in 1061 data samples.

We segment the data into two parts: the first 300 samples collected before the day of oil spill (April 20th, 2010) are regarded as conforming to a 10-dimensional joint distribution

[10] `http://en.wikipedia.org/wiki/Deepwater_Horizon_oil_spill`



(a) Gene regulatory network

(b) KLIEP

(c) IS-Flasso

(d) Flasso

(e) P-R curve

Figure 11: Experiments on synthetic gene expression datasets.

Q, while the second set of samples that are in an arbitrary 50-day window after the oil spill accident happened is regarded as following distribution P. Thus, the MN of Q encodes the original conditional independence of frequencies between 10 keywords, while the underlying MN of P has changed since an event occurred. We expect that unveiling changes in MNs between P and Q can recover the drift of popular topic trends on Twitter in terms of the dependency among keywords.

The detected change graphs (i.e., the graphs with only detected changing edges) on 10 keywords are illustrated in Figure 12. The edges are selected at a certain value of λ_2 indicated by the maximal *cross-validated log-likelihood* (CVLL). Since the edge set that is picked by CVLL may not be sparse in general, we sparsify the graph based on the permutation test as follows: we randomly shuffle the samples between P and Q and repeatedly run change detection algorithms for 100 times; then we observe detected edges by CVLL. Finally, we select the edges that are detected using the original non-shuffled dataset and remove those that were detected in the shuffled datasets for more than 5 times (i.e., the significance level 5%). In Figure 12, we plot detected change graphs which are generated using samples of P starting from April 17th, July 6th, and July 26th, respectively.

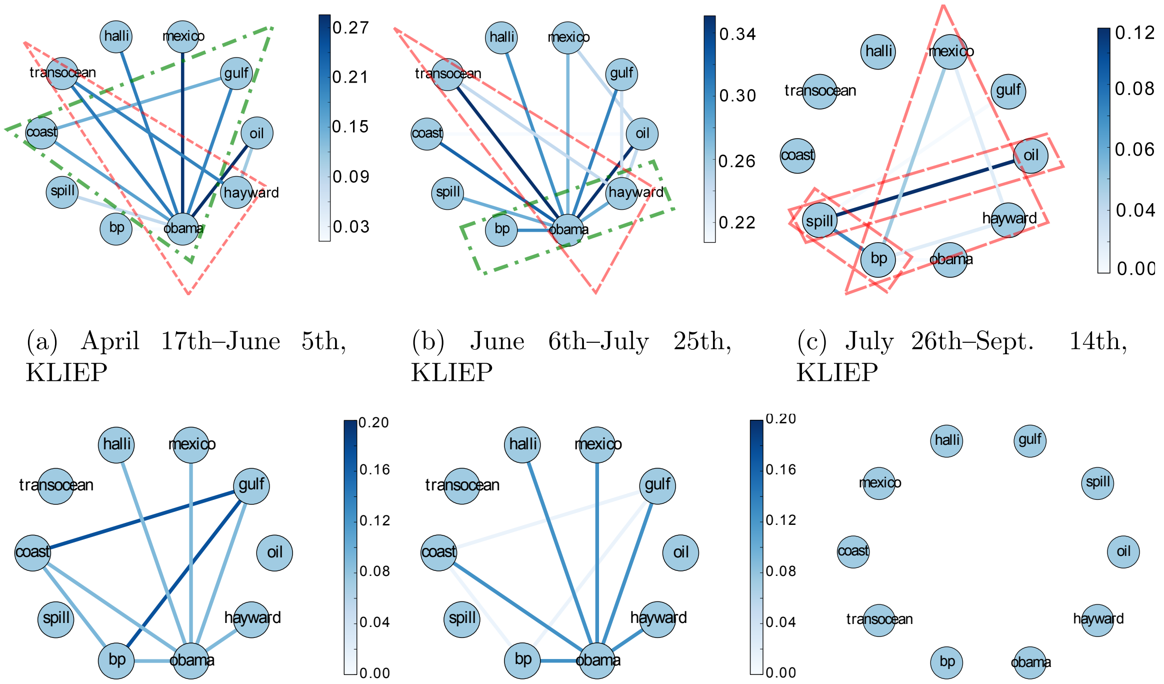
The initial explosion happened on April 20th, 2010. Both methods discover dependency changes between keywords. Generally speaking, KLIEP captures more conditional independence changes between keywords than the Flasso method, especially when comparing Figure 12(c) and Figure 12(f). At the first two stages (Figures 12(a), 12(b), 12(d) and 12(e)), the keyword "Obama" is very well connected with other keywords in the results given by both methods. Indeed, at the early development of this event, he lies in the center of the news stories, and his media exposure peaks after his visit to the Louisiana coast (May 2nd, May 28nd, and June 5th) and his meeting with BP CEO Tony Hayward on June 16th. Notably, both methods highlight the "gulf-obama-coast" triangle in Figures 12(a) and 12(d) and the "bp-obama-hayward" chain in Figures 12(b) and 12(e).

However, there are some important differences worth mentioning. First, the Flasso method misses the "transocean-hayward-obama" triangle in Figures 12(d) and 12(e). Transocean is the contracted operator in the Deepwater Horizon platform, where the initial explosion happened. On Figure 12(c), the chain "bp-spill-oil" may indicate that the phrase "bp spill" or "oil spill" has been publicly recognized by the Twitter community since then, while the "hayward-bp-mexico" triangle, although relatively weak, may link to the event that Hayward stepped down from the CEO position on July 27th.

It is also noted that Flasso cannot find any changed edges in Figure 12(f), perhaps due to the Gaussian restriction.

3.6 Discussion, Conclusion, and Future Works

In this paper, we proposed a *direct* approach to learning sparse changes in MNs by density ratio estimation. Rather than fitting two MNs separately to data and comparing them to detect a change, we estimated the ratio of the probability densities of two MNs where changes can be naturally encoded as sparsity patterns in estimated parameters.



(a) April 17th–June 5th, KLIEP

(b) June 6th–July 25th, KLIEP

(c) July 26th–Sept. 14th, KLIEP

(d) April 17th–June 5th, Flasso

(e) June 6th–July 25th, Flasso

(f) July 26th–Sept. 14th, Flasso

Figure 12: Change graphs captured by the proposed KLIEP method (top) and the Flasso method (bottom). The date range beneath each figure indicates when P was sampled, while Q is fixed to dates from February 1st to April 20th. Notable structures shared by the graph of both methods are surrounded by the dash-dotted lines. Unique structures that only appear in the graph of the proposed KLIEP method are surrounded by the dashed lines.

This direct modeling allows us to halve the number of parameters and approximate the normalization term in the density ratio model by a sample average without sampling. We also showed that the number of parameters to be optimized can be further reduced with the dual formulation, which is highly useful when the dimensionality is high. Through experiments on artificial and real-world datasets, we demonstrated the usefulness of the proposed method over state-of-the-art methods including nonparanormal-based methods and sampling-based methods.

Our important future work is to theoretically elucidate the advantage of the proposed method, beyond the Vapnik's principle of solving the target problem directly. The relation to *score matching* [75], which avoids computing the normalization term in density estimation, is also an interesting issue to be further investigated. Considering higher-order MN models such as the *hierarchical log-linear model* [144] is a promising direction for extension.

In the context of change detection, we are mainly interested in the situation where

p and q are close to each other (if p and q are completely different, it is straightforward to detect changes). When p and q are similar, density ratio estimation for $p(\boldsymbol{x})/q(\boldsymbol{x})$ or $q(\boldsymbol{x})/p(\boldsymbol{x})$ perform similarly. However, given the asymmetry of density ratios, the solutions for $p(\boldsymbol{x})/q(\boldsymbol{x})$ or $q(\boldsymbol{x})/p(\boldsymbol{x})$ are generally different. The choice of the numerator and denominator in the ratio is left for future investigation.

Detecting changes in MNs is the main target of this paper. On the other hand, estimating the difference/divergence between two probability distributions has been studied under a more general context in the statistics and machine learning communities [8, 49, 200, 171, 161]. In fact, the estimation of the *Kullback-Leibler divergence* [103] is related to the KLIEP-type density ratio estimation method [125], and the estimation of the *Pearson divergence* [129] is related to the squared-loss density ratio estimation method [83]. However, the density ratio based divergences tend to be sensitive to outliers. To overcome this problem, a divergence measure based on relative density ratios was introduced, and its direct estimation method was developed [212]. L^2-distance is another popular difference measure between probability density functions. L^2-distance is symmetric, unlike the Kullback-Leibler divergence and the Pearson divergence, and its direct estimation method has been investigated recently [172, 96].

Change detection in time-series a related topic. A straightforward approach is to evaluate the difference (dissimilarity) between two consecutive segments of time-series signals. Various methods have been developed to identify the difference by fitting two models to two segments of time-series separately, e.g., the singular spectrum transform [120, 76], subspace identification [94], and the method based on the one-class support vector machine [42]. In the same way as the current paper, directly modeling of the change has also been explored for change detection in time-series [93, 113, 172].

4 Learning under Non-Stationarity

4.1 Background

The goal of supervised learning such as regression and classification is to learn an input-output dependency from input-output paired training samples so that test output y' for unseen test input $\boldsymbol{x}'$ can be accurately estimated. Various supervised learning algorithms were developed thus far, and they have been demonstrated to be useful in a wide range of applications. Most of the popular machine learning algorithms assume that training and test data follow the same probability distribution, based on which learning machines can generalize to unseen test data from training data [194, 71, 24]. However, this fundamental assumption is often violated in practice, and this causes standard supervised learning algorithms suffer significant estimation bias.

In this article, we consider two scenarios. The first setup is the *covariate shift* [150, 158], where training and test input data follow different distributions but the input-output relation does not change between training and test phases. The other setup is called *class-balance change* in classification [142, 45], where the class-prior probabilities are different in training and test phases but the input distribution of each class does not change. For these two scenarios, we review semi-supervised adaptation techniques, where *importance weighting* plays an essential role.

More specifically, we consider the semi-supervised learning problem where input-output training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ and input-only test samples $\{\boldsymbol{x}'_{i'}\}_{i'=1}^{n'}$ are available. In the standard semi-supervised learning setup, training and test samples are regarded as being drawn from the same probability distribution [31]. In contrast, in this article, we suppose that they are drawn from different distributions: $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ are drawn independently from a joint probability distribution with density $p(\boldsymbol{x}, y)$ and $\{\boldsymbol{x}'_{i'}\}_{i'=1}^{n'}$ are drawn independently from a marginal probability distribution with density $\int p'(\boldsymbol{x}, y)\mathrm{d}y$, where $p(\boldsymbol{x}, y)$ and $p'(\boldsymbol{x}, y)$ are different:

$$p(\boldsymbol{x}, y) \neq p'(\boldsymbol{x}, y).$$

Our goal is to learn the input-output relation for test samples. The situation where training and test samples follow different distributions is also referred to as *non-stationarity adaptation*, *dataset-shift adaptation*, *transfer learning*, and *domain adaptation*. The semi-supervised learning setup with differing training and testing distributions is sometimes called *unsupervised transfer* or *unsupervised adaptation* in literature because no supervision is available from the test domain.

4.2 Adaptation Techniques for Covariate Shift

The *covariate shift* [150, 158] is the situation where input distributions change but the conditional distribution of outputs given inputs remains unchanged:

$$p(\boldsymbol{x}) \neq p'(\boldsymbol{x}) \quad \text{and} \quad p(y|\boldsymbol{x}) = p'(y|\boldsymbol{x}).$$

Figure 13 illustrates an example of covariate shift regression: Training input samples $\{\boldsymbol{x}_i\}_{i=1}^{n}$ are drawn from the left-hand side of the domain, whereas test input samples $\{\boldsymbol{x}'_{i'}\}_{i'=1}^{n'}$ are drawn from the right-hand side. This problem is similar to *extrapolation* since the prediction is made in a low density region of the training set.

4.2.1 Importance-Weighted Learning

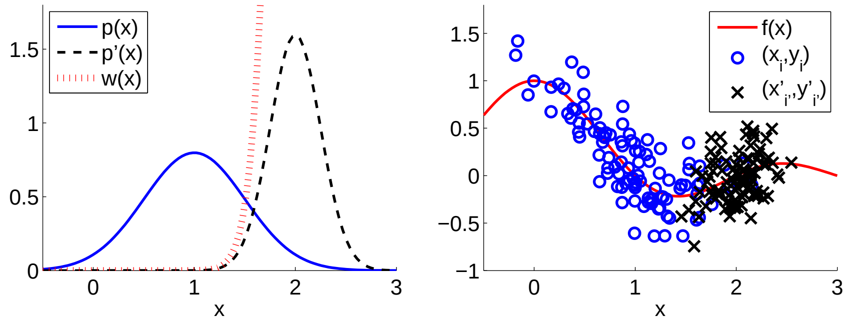
For this covariate-shift regression problem, let us use a simple linear model,

$$M(x) = \theta_1 + \theta_2 x,$$

and train this model by *ordinary least-squares*:

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \Big(M(\boldsymbol{x}_i) - y_i \Big)^2.$$

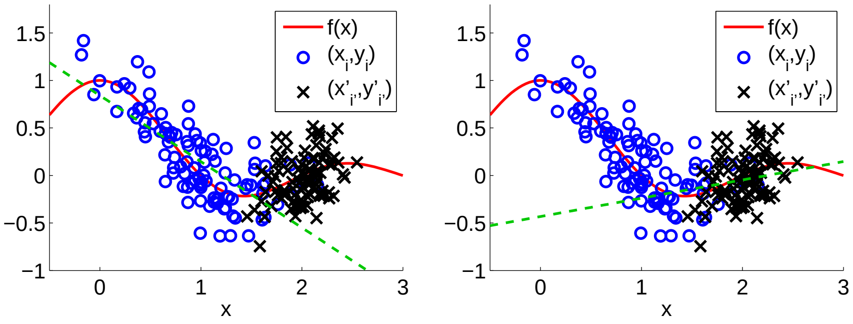
The learned result illustrated in Figure 14(a) shows that the obtained function fits the training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ very well, but it does not give good prediction of outputs for the test input samples $\{\boldsymbol{x}'_{i'}\}_{i'=1}^{n'}$ (i.e., samples denoted by “×”).



(a) Input densities and importance

(b) Learning target function, training samples, and test samples

Figure 13: Covariate shift. Input distributions change but the conditional distribution of outputs given inputs does not changed.



(a) Ordinary least-squares

(b) Importance-weighted least-squares

Figure 14: Regression under covariate shift. Dashed lines denote learned functions.

Under the covariate shift, it is expected that only training samples whose input points are close to test input samples $\{\boldsymbol{x}'_{i'}\}_{i'=1}^{n'}$ are useful. This intuitive idea can be realized by weighting the training loss according to the *importance*, which is the ratio between $p'(\boldsymbol{x})$ and $p(\boldsymbol{x})$.

$$w(\boldsymbol{x}) := \frac{p'(\boldsymbol{x})}{p(\boldsymbol{x})}.$$

In Figure 14(b), the learned result obtained by *importance-weighted least-squares* [150],

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} w(\boldsymbol{x}_i)\Big(M(\boldsymbol{x}_i) - y_i\Big)^2,$$

is illustrated. This shows that importance weighting can improve the accuracy of predicting outputs for the test input samples $\{\boldsymbol{x}'_{i'}\}_{i'=1}^{n'}$.

The above importance-weighted least-squares can be regarded as an application of *importance weighting* to approximating the generalization error (or the expected test loss):

$$G := \iint \mathrm{loss}(y, M(\boldsymbol{x}))p'(\boldsymbol{x}, y)\mathrm{d}\boldsymbol{x}\mathrm{d}y,$$

where $\mathrm{loss}(y, \widehat{y})$ denotes a point-wise loss when y is predicted by $\widehat{y}$. More specifically, the generalization error G can be approximated by the importance-weighted average of the training loss:

$$\begin{aligned}
G &= \iint \mathrm{loss}(y, M(\boldsymbol{x}))p'(y|\boldsymbol{x})p'(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\mathrm{d}y \\
&= \iint \mathrm{loss}(y, M(\boldsymbol{x}))p'(y|\boldsymbol{x})\frac{p'(\boldsymbol{x})}{p(\boldsymbol{x})}p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\mathrm{d}y \\
&= \iint \mathrm{loss}(y, M(\boldsymbol{x}))w(\boldsymbol{x})p(\boldsymbol{x}, y)\mathrm{d}\boldsymbol{x}\mathrm{d}y \\
&\approx \frac{1}{n}\sum_{i=1}^{n} \mathrm{loss}(y_i, M(\boldsymbol{x}_i))w(\boldsymbol{x}_i).
\end{aligned}$$

Note that this importance weighting idea can be applied to *any* likelihood/loss-based learning algorithms, including *Fisher discriminant analysis*, *logistic regression*, the *support vector machine*, *boosting*, and the *conditional random field*, and it also plays an important role for reducing the estimation bias in active learning and experimental design scenarios [202, 84, 155, 81, 165, 163]. See [158] for more thorough discussion on importance-weighted learning.

To implement importance-weighted learning, importance values $\{w(\boldsymbol{x}_i)\}_{i=1}^{n}$ are necessary. However, training and test input densities $p(\boldsymbol{x})$ and $p'(\boldsymbol{x})$ are unknown in practice, and thus the importance values should be estimated from data. A naive approach is to estimate $p(\boldsymbol{x})$ from $\{\boldsymbol{x}_i\}_{i=1}^{n}$ and $p'(\boldsymbol{x})$ from $\{\boldsymbol{x}'_{i'}\}_{i'=1}^{n'}$ separately and then take their ratio. However, such a two-step procedure is not accurate because the error incurred in the estimation of $p(\boldsymbol{x})$ and $p'(\boldsymbol{x})$ can be increased when their ratio is computed in the second stage. Thus, *directly* estimating the ratio $w(\boldsymbol{x})$ without estimating $p(\boldsymbol{x})$ and $p'(\boldsymbol{x})$ is more preferable.

Following this idea, various methods of importance estimation have been developed, for example, based on density estimation of $p'(\boldsymbol{x})$ after uniformization of $p(\boldsymbol{x})$ [40, 32], logistic regression for discriminating data from $p(\boldsymbol{x})$ and $p'(\boldsymbol{x})$ [131, 33, 20], moment matching between $p'(\boldsymbol{x})$ and $p(\boldsymbol{x})w(\boldsymbol{x})$ [131, 62, 87], integral equations between $p'(\boldsymbol{x})$ and $p(\boldsymbol{x})w(\boldsymbol{x})$ [196, 132], density matching between $p'(\boldsymbol{x})$ and $p(\boldsymbol{x})w(\boldsymbol{x})$ under the Kullback-Leibler divergence [173, 124, 187, 206, 211], least-squares importance fitting of $w(\boldsymbol{x})$ to $p'(\boldsymbol{x})/p(\boldsymbol{x})$ [82, 87], and importance fitting of $w(\boldsymbol{x})$ to $p'(\boldsymbol{x})/p(\boldsymbol{x})$ under the Bregman divergence [170].

Among them, the least-squares importance fitting method has various practical advantages, for example, an analytic-form solution that can be computed efficiently is available, cross-validation is available for hyperparameter tuning, the optimal convergence rate is achieved both in parametric and non-parametric settings [82, 87], and the highest numerical stability in terms of condition numbers is achieved among a class of importance estimators [88]. Furthermore, dimensionality reduction methods for improving the accuracy of importance estimation in high-dimensional problems have been developed [159, 176, 209]. See [168] for more comprehensive discussion on direct importance estimation.

4.2.2 Relative Importance-Weighted Learning

Let us continue using the illustrative example described in Figure 13 and Figure 14. The true importance function $w(x)$ is plotted in Figure 13(a). This shows that, among many training samples, only a small number of samples at around $x = 2$ have large importance weights and other samples have almost zero weights. This implies that importance-weighted learning in this example is rather unreliable because the learned function is essentially obtained from only a few training samples.

Such unreliable behavior is caused by the fact that the importance function $w(\boldsymbol{x})$ can take very large values. To cope with this problem, the *relative importance weight* is useful [212]:

$$w^{(\beta)}(\boldsymbol{x}) = \frac{p'(\boldsymbol{x})}{\beta p'(\boldsymbol{x}) + (1-\beta)p(\boldsymbol{x})},$$

where $\beta \in [0,1]$ is the relativity parameter. The relative importance weight $w^{(\beta)}(\boldsymbol{x})$ is reduced to the ordinary importance weight $w(\boldsymbol{x})$ when $\beta = 0$. As β is increased, the relative importance weight gets flatter and is reduced to the uniform weight $w^{(\beta)}(\boldsymbol{x}) = 1$ when $\beta = 1$ (Figure 15). The non-negativity of the importance function, $p'(\boldsymbol{x})/p(\boldsymbol{x}) \geq 0$, assures that the relative importance weight is bounded from above by $1/\beta$:

$$w^{(\beta)}(\boldsymbol{x}) = \frac{1}{\beta + (1-\beta)\frac{p(\boldsymbol{x})}{p'(\boldsymbol{x})}} \leq \frac{1}{\beta}.$$

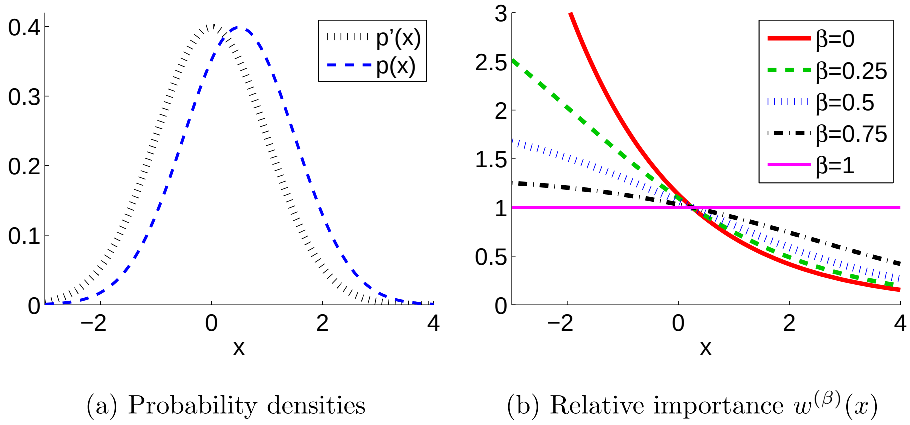
The least-squares method combined with the relative importance weight is called *relative importance-weighted least-squares*:

$$\min_{\boldsymbol{\theta}} \frac{1}{2}\sum_{i=1}^{n} w^{(\beta)}(\boldsymbol{x}_i)\Big(M(\boldsymbol{x}_i) - y_i\Big)^2,$$

where the relativity parameter β controls the trade-off between bias and variance.

Now let us consider the problem of estimating the relative importance weight $w^{(\beta)}(\boldsymbol{x})$ from $\{\boldsymbol{x}_i\}_{i=1}^{n}$ and $\{\boldsymbol{x}'_{i'}\}_{i'=1}^{n'}$. We use the following linear-in-parameter model $w_{\boldsymbol{\alpha}}(\boldsymbol{x})$ for learning the relative importance weight $w^{(\beta)}(\boldsymbol{x})$:

$$w_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \sum_{j=1}^{b} \alpha_j \psi_j(\boldsymbol{x}) = \boldsymbol{\alpha}^\top \boldsymbol{\psi}(\boldsymbol{x}),$$



(a) Probability densities (b) Relative importance $w^{(\beta)}(x)$

Figure 15: Relative importance. $p'(x)$ is the normal distribution with mean 0 and variance 1, and $p(x)$ is the normal distribution with mean 0.5 and variance 1.

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_b)^\top$ is the parameter vector and $\boldsymbol{\psi}(\boldsymbol{x}) = (\psi_1(\boldsymbol{x}), \ldots, \psi_b(\boldsymbol{x}))^\top$ is the basis function vector. As basis functions, we may use, for example, the Gaussian kernels:

$$w_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \sum_{j=1}^{n'} \alpha_j \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'_j\|^2}{2\sigma^2}\right),$$

where σ^2 denotes the Gaussian width.

Then the parameter $\boldsymbol{\alpha}$ is learned so that the following criterion $J(\boldsymbol{\alpha})$ is minimized:

$$\begin{aligned} J(\boldsymbol{\alpha}) &= \int \left(w_{\boldsymbol{\alpha}}(\boldsymbol{x}) - w^{(\beta)}(\boldsymbol{x})\right)^2 \left(\beta p'(\boldsymbol{x}) + (1-\beta)p(\boldsymbol{x})\right)\mathrm{d}\boldsymbol{x} \\ &= \int \boldsymbol{\alpha}^\top \boldsymbol{\psi}(\boldsymbol{x})\boldsymbol{\psi}(\boldsymbol{x})^\top \boldsymbol{\alpha} \left(\beta p'(\boldsymbol{x}) + (1-\beta)p(\boldsymbol{x})\right)\mathrm{d}\boldsymbol{x} \\ &\quad - 2\int \boldsymbol{\alpha}^\top \boldsymbol{\psi}(\boldsymbol{x}) p'(\boldsymbol{x})\mathrm{d}\boldsymbol{x} + C, \end{aligned}$$

where the third term,

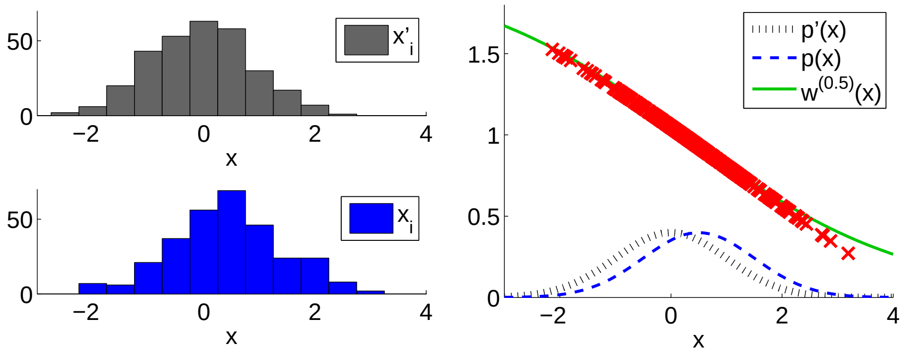
$$C = \int w^{(\beta)}(\boldsymbol{x}) p'(\boldsymbol{x})\mathrm{d}\boldsymbol{x},$$

is a constant irrelevant to the parameter $\boldsymbol{\alpha}$ and thus can be ignored. Approximating the expectations in the first and second terms by sample averages and adding the ℓ_2-regularizer, we have the following training criterion:

$$\min_{\boldsymbol{\alpha}} \left[\boldsymbol{\alpha}^\top \widehat{\boldsymbol{G}}_\beta \boldsymbol{\alpha} - 2\boldsymbol{\alpha}^\top \widehat{\boldsymbol{h}} + \lambda\|\boldsymbol{\alpha}\|^2\right],$$

where $\widehat{\boldsymbol{G}}_\beta$ and $\widehat{\boldsymbol{h}}$, a $b \times b$ matrix and a b-dimensional vector, are defined as

$$\widehat{\boldsymbol{G}}_\beta = \frac{\beta}{n'} \sum_{i'=1}^{n'} \boldsymbol{\psi}(\boldsymbol{x}'_{i'})\boldsymbol{\psi}(\boldsymbol{x}'_{i'})^\top + \frac{1-\beta}{n} \sum_{i=1}^{n} \boldsymbol{\psi}(\boldsymbol{x}_i)\boldsymbol{\psi}(\boldsymbol{x}_i)^\top \quad \text{and} \quad \widehat{\boldsymbol{h}} = \frac{1}{n'} \sum_{i'=1}^{n'} \boldsymbol{\psi}(\boldsymbol{x}'_{i'}).$$



(a) Training and test data

(b) Relative importance weight ($\beta = 0.5$)

Figure 16: Illustration of RuLSIF. "×" in Figure 16(b) denotes an estimated relative importance value at $\boldsymbol{x}_i$.

This training criterion is a convex quadratic function of $\boldsymbol{\alpha}$ and its minimizer $\widehat{\boldsymbol{\alpha}}$ can be obtained analytically as

$$\widehat{\boldsymbol{\alpha}} = \left(\widehat{\boldsymbol{G}}_\beta + \lambda \boldsymbol{I}\right)^{-1} \widehat{\boldsymbol{h}}.$$

This method is called *relative unconstrained least-squares importance fitting* (RuLSIF) [212]. Tuning parameters such as the regularization parameter λ and the Gaussian width σ^2 can be optimized via cross-validation with respect to J.

An example of relative importance estimation by RuLSIF is illustrated in Figure 16.

4.2.3 Importance-Weighted Model Selection

Choice of the relativity parameter β as well as other tuning parameters such as basis functions and regularization parameters is crucial for obtaining better performance in practice. For model selection, various methods such as the *Akaike information criterion* [3], the *subspace information criterion* [164], and *cross-validation* [154] are available. However, under the covariate shift, these model selection techniques based on training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ do not give valid evaluation of the prediction accuracy of outputs for test inputs $\{\boldsymbol{x}'_{i'}\}_{i'=1}^{n'}$.

Under the covariate shift, importance-weighted variants of such model selection methods are useful [150, 162, 160]. The simplest model selection method called *importance-weighted cross-validation* is given as follows:

1. Randomly split training samples $\mathcal{T} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ into m disjoint subsets $\{\mathcal{T}_i\}_{i=1}^m$ of (approximately) the same size.

2. Repeat for $i = 1, \ldots, m$;

(a) Obtain a learned function f_i from $\mathcal{T}\backslash\mathcal{T}_i$ (i.e., all samples without $\mathcal{T}_i$).

(b) Evaluate the generalization error using hold-out samples $\mathcal{T}_i$ as

$$\widehat{G}_i = \begin{cases} \dfrac{1}{|\mathcal{T}_i|} \displaystyle\sum_{(\boldsymbol{x},y)\in\mathcal{T}_i} w(\boldsymbol{x})\Big(f_i(\boldsymbol{x}) - y\Big)^2 & \text{(Regression)}, \\ \dfrac{1}{|\mathcal{T}_i|} \displaystyle\sum_{(\boldsymbol{x},y)\in\mathcal{T}_i} \dfrac{w(\boldsymbol{x})}{2}\Big(1 - \mathrm{sign}\big(f_i(\boldsymbol{x})y\big)\Big) & \text{(Classification)}, \end{cases}$$

where $|\mathcal{T}_i|$ denotes the number of elements in the set $\mathcal{T}_i$.

3. Output the average of $\widehat{G}_1, \ldots, \widehat{G}_m$ as the final evaluation $\widehat{G}$ of the generalization error:

$$\widehat{G} = \frac{1}{m}\sum_{i=1}^{m} \widehat{G}_i.$$

4.2.4 Applications

Importance-weighted learning has been successfully applied to various real-world problems, including brain-computer interface [160, 107], robot control [64, 4, 65, 221], speaker identification [210], age prediction from face images [190], activity recognition from accelerometers [66], natural language processing [187], spam filtering [22], targeted advertising [21], HIV therapy screening [19], and wafer alignment in semiconductor exposure apparatus [163]. Below, we describe application of covariate shift adaptation in 3D human-pose estimation from monocular videos [205].

We use the HumanEva-I dataset [151], which contains synchronized multi-view videos and motion-capture data for 3 subjects performing multiple activities: Walking, jogging, boxing, throwing and catching, and gesturing. As input $\boldsymbol{x}$, we extract the *histogram-of-oriented-gradient (HoG) feature* [26] of 270 dimensions from videos taken by 3 color cameras with 9630 image-pose frames for each camera. Output $\boldsymbol{y}$ is a corresponding pose vector, which means that we consider a multi-dimensional regression problem. We randomly select n samples from the set of $3 \times 4815 = 14445$ frames for training and use the remaining 14445 frames for testing.

We consider the following scenarios:

Selection bias: The training set contains data from all 3 subjects, whereas the test set only contains data from a single subject.

Subject transfer: The training set contains data from 2 subjects, whereas the test set contains data from the remaining subject not included in the training set.

As regression algorithms, we use *kernel regression* (KR) [2], *twin Gaussian processes regression* (TGP) [26], and the *weighted k-nearest neighbor* (WkNN) method [146]. See [205] for the details of these algorithms. For KR and TGP, we consider their importance-weighted variants which are referred to as IWKR and IWTGP.

Each pose is represented by 20 3D-joint markers: $\boldsymbol{y} = [\boldsymbol{y}^{(1)\top}, \ldots, \boldsymbol{y}^{(20)\top}]^\top \in \mathbb{R}^{60}$, where $\boldsymbol{y}^{(m)} \in \mathbb{R}^3$ for $m = 1, \ldots, 20$. Error between true pose $\boldsymbol{y}^*$ and its estimate $\widehat{\boldsymbol{y}}$ is measured by the average Euclidean distance:

$$\mathrm{Error}(\boldsymbol{y}^*, \widehat{\boldsymbol{y}}) = \frac{1}{20} \sum_{m=1}^{20} \|\widehat{\boldsymbol{y}}^{(m)} - \boldsymbol{y}^{*(m)}\|.$$

Figure 17 shows the pose estimation error as a function of the training sample size n averaged over all motions and 10 runs. The graphs clearly show that IWTGP and IWKR outperform their non-adaptive counterparts and the baseline WkNN method.

4.3 Adaptation Techniques for Class-Balance Change

Class-balance change [142, 45] is the classification problem where class-prior probabilities change but the conditional distribution of input $\boldsymbol{x}$ given class y remains unchanged:

$$p(y) \neq p'(y) \quad \text{and} \quad p(\boldsymbol{x}|y) = p'(\boldsymbol{x}|y). \tag{26}$$

Figure 18 illustrates an example of classification under class-balance change. When the class balances are different in the training and test phases, naive training of a classifier yields significant estimation bias even if the class-conditional input density is unchanged.

In the same way as covariate shift adaptation, estimation bias caused by class-balance change can be canceled by weighting the training loss according to the *class-balance ratio*:

$$w(y) = \frac{p'(y)}{p(y)}.$$

Below, we focus on binary classification where label y takes either $+1$ or -1 for simplicity.

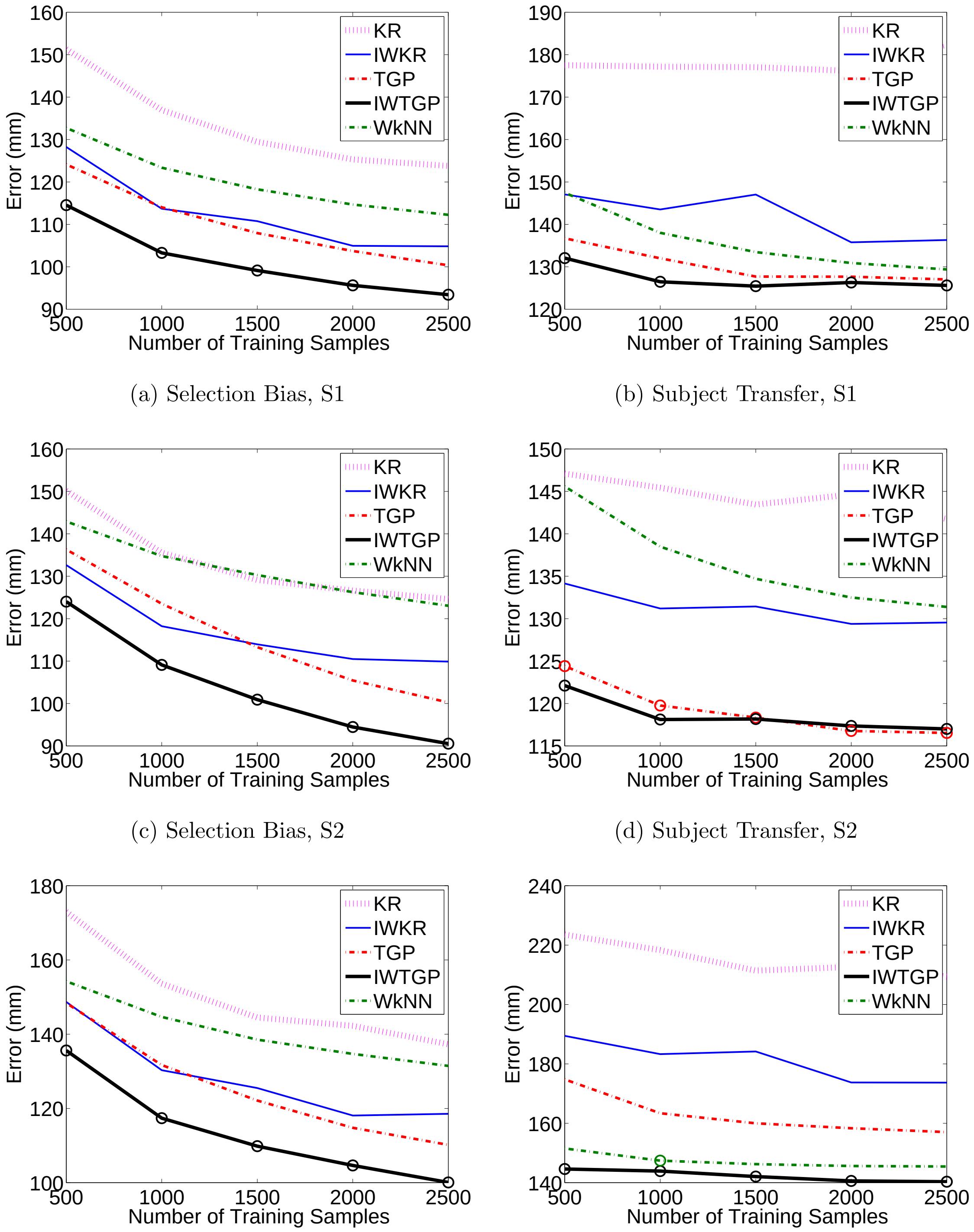
4.3.1 Class-Balance Estimation

The training class-balance $p(y)$ can be naively estimated by n_y/n if n_y samples belong to class y in the training set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$. The test class-balance $p'(y)$ can also be estimated in the same way if a labeled test set $\{(\boldsymbol{x}'_{i'}, y'_{i'})\}_{i'=1}^{n'}$ is available. However, we are considering a semi-supervised learning setup where only an unlabeled test set $\{\boldsymbol{x}'_{i'}\}_{i'=1}^{n'}$ is available. Thus, $p'(y)$ cannot be estimated naively.

In the semi-supervised learning setup under Eq.(26), $p'(y)$ can be estimated by fitting a mixture $q_\pi(\boldsymbol{x})$ of training class-wise densities $p(\boldsymbol{x}|y)$ to test input density $p'(\boldsymbol{x})$ (see Figure 19):

$$q_\pi(\boldsymbol{x}) = \pi p(\boldsymbol{x}|y = +1) + (1 - \pi) p(\boldsymbol{x}|y = -1).$$

The value of the parameter π corresponds to $p'(y = +1)$, whereas $1 - \pi$ corresponds to $p'(y = -1)$.



(a) Selection Bias, S1

(b) Subject Transfer, S1

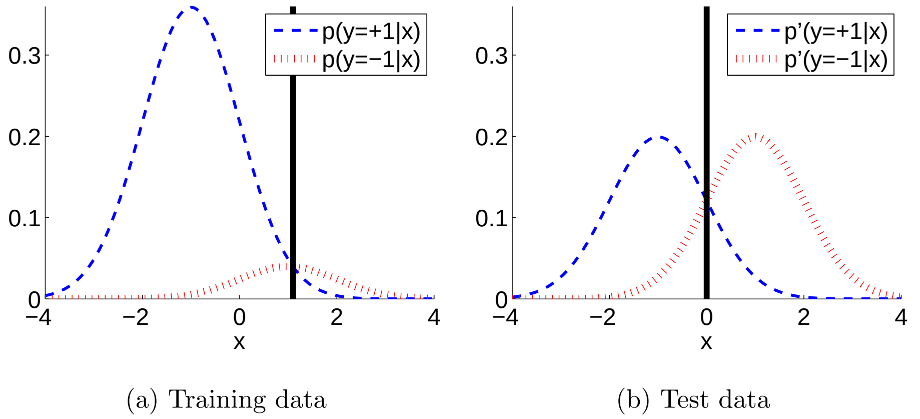
(c) Selection Bias, S2

(d) Subject Transfer, S2

(e) Selection Bias, S3

(f) Subject Transfer, S3

Figure 17: 3D human-pose estimation error as a function of the number of training samples averaged over all motions for each subject. The best method and comparable ones in terms of the average error according to the paired *t-test* at the significance level 5% are specified by '○'.



(a) Training data

(b) Test data

Figure 18: Change in class balances shifts the optimal classification boundary. Class-conditional input density is the same between the training and test phases (i.e., $p(\boldsymbol{x}|y) = p'(\boldsymbol{x}|y)$), but class-prior probabilities are different (i.e., $p(y) \neq p'(y)$).

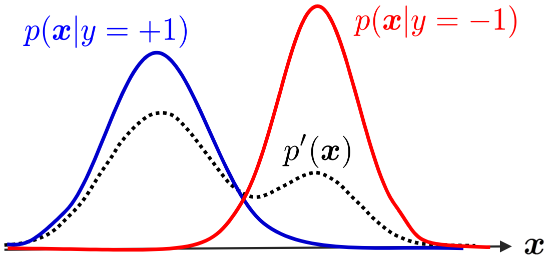


Figure 19: $p'(y)$ can be estimated by fitting a mixture of training class-wise densities $p(\boldsymbol{x}|y)$ to test input density $p'(\boldsymbol{x})$.

For the fitting of q_π to p', we may use the *Kullback-Leibler (KL) divergence* [103] or the *Pearson (PE) divergence* [128]:

$$\mathrm{KL}(p'\|q_\pi) = \int p'(\boldsymbol{x}) \log \frac{p'(\boldsymbol{x})}{q_\pi(\boldsymbol{x})} \mathrm{d}\boldsymbol{x},$$
$$\mathrm{PE}(p'\|q_\pi) = \int q_\pi(\boldsymbol{x}) \left(\frac{p'(\boldsymbol{x})}{q_\pi(\boldsymbol{x})} - 1 \right)^2 \mathrm{d}\boldsymbol{x}.$$

These divergences can be accurately approximated from samples by directly estimating the density ratio $p'(\boldsymbol{x})/q_\pi(\boldsymbol{x})$ without density estimation of $p'(\boldsymbol{x})$ and $q_\pi(\boldsymbol{x})$ [168]. However, the density ratio function $p'(\boldsymbol{x})/q_\pi(\boldsymbol{x})$ is sensitive to small variation, and therefore it is not robust against outliers.

Here we consider the L^2*-distance* between p' and q_π:

$$L^2(p', q_\pi) = \int \Big(p'(\boldsymbol{x}) - q_\pi(\boldsymbol{x}) \Big)^2 \mathrm{d}\boldsymbol{x}.$$

The L^2-distance can also be accurately approximated from samples by directly estimating the density difference $p'(\boldsymbol{x}) - q_\pi(\boldsymbol{x})$, without density estimation of $p'(\boldsymbol{x})$ and $q_\pi(\boldsymbol{x})$ [172].

Historically, non-parametric estimation of mixture proportion π under the L^2-distance was first investigated in [67], which uses empirical distribution functions. Following this seminal work, its variant based on kernel density estimation has been developed [184], and this is further extended to choosing the kernel bandwidths jointly [68]. In the related context of two-sample homogeneity testing under the L^2-distance, the use of kernel density estimators with fixed and equal bandwidths has been investigated [9].

4.3.2 L^2-Distance Approximation

Here, we explain how the L^2-distance can be directly approximated from data via direct density-difference estimation [96, 172]. For simplicity, we consider the approximation problem of the L^2-distance between p and p',

$$L^2(p, p') = \int f(\boldsymbol{x})^2 \mathrm{d}\boldsymbol{x}, \quad \text{where } f(\boldsymbol{x}) = p(\boldsymbol{x}) - p'(\boldsymbol{x}), \tag{27}$$

from $\{\boldsymbol{x}_i\}_{i=1}^{n}$ and $\{\boldsymbol{x}'_{i'}\}_{i'=1}^{n'}$.

We use the following Gaussian density-difference model:

$$g(\boldsymbol{x}) = \sum_{j=1}^{n+n'} \alpha_j \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{c}_j\|^2}{2\sigma^2}\right),$$

where

$$(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_n, \boldsymbol{c}_{n+1}, \ldots, \boldsymbol{c}_{n+n'}) = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n, \boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_{n'})$$

are Gaussian centers. The parameter $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{n+n'})^\top$ in the density-difference model is learned so that the following criterion $J(\boldsymbol{\alpha})$ is minimized:

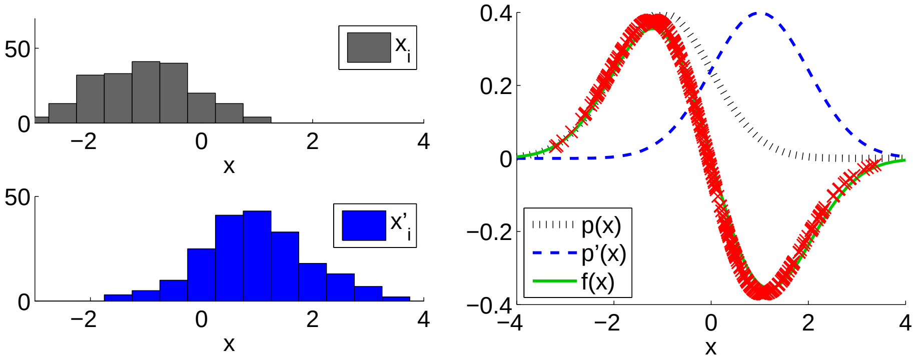
$$\begin{aligned} J(\boldsymbol{\alpha}) &= \int \Big(g(\boldsymbol{x}) - f(\boldsymbol{x})\Big)^2 \mathrm{d}\boldsymbol{x} \\ &= \int g(\boldsymbol{x})^2 \mathrm{d}\boldsymbol{x} - 2\int g(\boldsymbol{x}) f(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} + C, \end{aligned}$$

where the third term,

$$C = \int f(\boldsymbol{x})^2 \mathrm{d}\boldsymbol{x},$$

is a constant irrelevant to the parameter $\boldsymbol{\alpha}$ and thus can be ignored. The first term can be computed analytically as

$$\int g(\boldsymbol{x})^2 \mathrm{d}\boldsymbol{x} = \boldsymbol{\alpha}^\top \boldsymbol{U} \boldsymbol{\alpha},$$



(a) Data (b) Density difference

Figure 20: Illustration of LSDD. "×" in Figure 20(b) denotes an estimated density difference value at $\boldsymbol{x}_i$ and $\boldsymbol{x}'_{i'}$.

where $\boldsymbol{U}$ is the $(n+n') \times (n+n')$ matrix with the (j,j')-th element defined by

$$U_{j,j'} = \int \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{c}_j\|^2}{2\sigma^2}\right) \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{c}_{j'}\|^2}{2\sigma^2}\right) \mathrm{d}\boldsymbol{x}$$
$$= (\pi\sigma^2)^{d/2} \exp\left(-\frac{\|\boldsymbol{c}_j-\boldsymbol{c}_{j'}\|^2}{4\sigma^2}\right).$$

Approximating the expectations in the second term by sample averages and adding the ℓ_2-regularizer, we have the following training criterion:

$$\min_{\boldsymbol{\alpha}} \left[\boldsymbol{\alpha}^\top \boldsymbol{U}\boldsymbol{\alpha} - 2\boldsymbol{\alpha}^\top \widehat{\boldsymbol{v}} + \lambda\|\boldsymbol{\alpha}\|^2\right],$$

where $\widehat{\boldsymbol{v}}$ is the $(n+n')$-dimensional vector with the j-th element defined by

$$\widehat{v}_j = \frac{1}{n}\sum_{i=1}^{n} \exp\left(-\frac{\|\boldsymbol{x}_i-\boldsymbol{c}_j\|^2}{2\sigma^2}\right) - \frac{1}{n'}\sum_{i'=1}^{n'} \exp\left(-\frac{\|\boldsymbol{x}'_{i'}-\boldsymbol{c}_j\|^2}{2\sigma^2}\right).$$

This training criterion is a convex quadratic function of $\boldsymbol{\alpha}$ and its minimizer $\widehat{\boldsymbol{\alpha}}$ can be obtained analytically as

$$\widehat{\boldsymbol{\alpha}} = (\boldsymbol{U} + \lambda \boldsymbol{I})^{-1}\widehat{\boldsymbol{v}}.$$

This method is called the *least-squares density-difference* (LSDD) estimator [172]. Tuning parameters such as the regularization parameter λ and basis function $\boldsymbol{\psi}$ can be optimized via cross-validation with respect to J. An example of density-difference estimation by LSDD is illustrated in Figure 20.

If the true density-difference f in Eq.(27) is replaced with the LSDD estimator, we obtain the following L^2-distance estimator:

$$\widehat{\boldsymbol{\alpha}}^\top \boldsymbol{U}\widehat{\boldsymbol{\alpha}}.$$

Similarly, from another expression of the L^2-distance estimator,

$$L^2(p, p') = \int f(\boldsymbol{x})\Big(p(\boldsymbol{x}) - p'(\boldsymbol{x})\Big)\mathrm{d}\boldsymbol{x},$$

we obtain the following L^2-distance estimator:

$$\widehat{\boldsymbol{v}}^\top \widehat{\boldsymbol{\alpha}}.$$

It was shown that the linear combination of these estimators,

$$2\widehat{\boldsymbol{v}}^\top \widehat{\boldsymbol{\alpha}} - \widehat{\boldsymbol{\alpha}}^\top \boldsymbol{U} \widehat{\boldsymbol{\alpha}},$$

tends to have smaller bias [172], and thus this would be a more reliable L^2-distance estimator in practice.

4.3.3 Experiments

Here, we use four *UCI benchmark datasets*[11] for experiments, where we randomly choose 10 labeled training samples from each class and 50 unlabeled test samples following true class-prior:

$$\pi^* = 0.1, 0.2, \ldots, 0.9.$$

The LSDD method is compared with the following methods:

KDEi: Kernel density estimation (KDE) is used to approximate $p'(\boldsymbol{x})$ and $q_\pi(\boldsymbol{x})$ from data and then the L^2-distance is computed [184]. Two Gaussian widths are *independently* chosen based on 5-fold least-squares cross-validation [69].

KDEj In the KDE-based method, two Gaussian widths are *jointly* chosen based on 5-fold cross-validation in terms of the LSDD criterion [68]. That is, the cross-validated LSDD criterion is computed as a function of two Gaussian widths and the best pair that minimizes the criterion is selected.

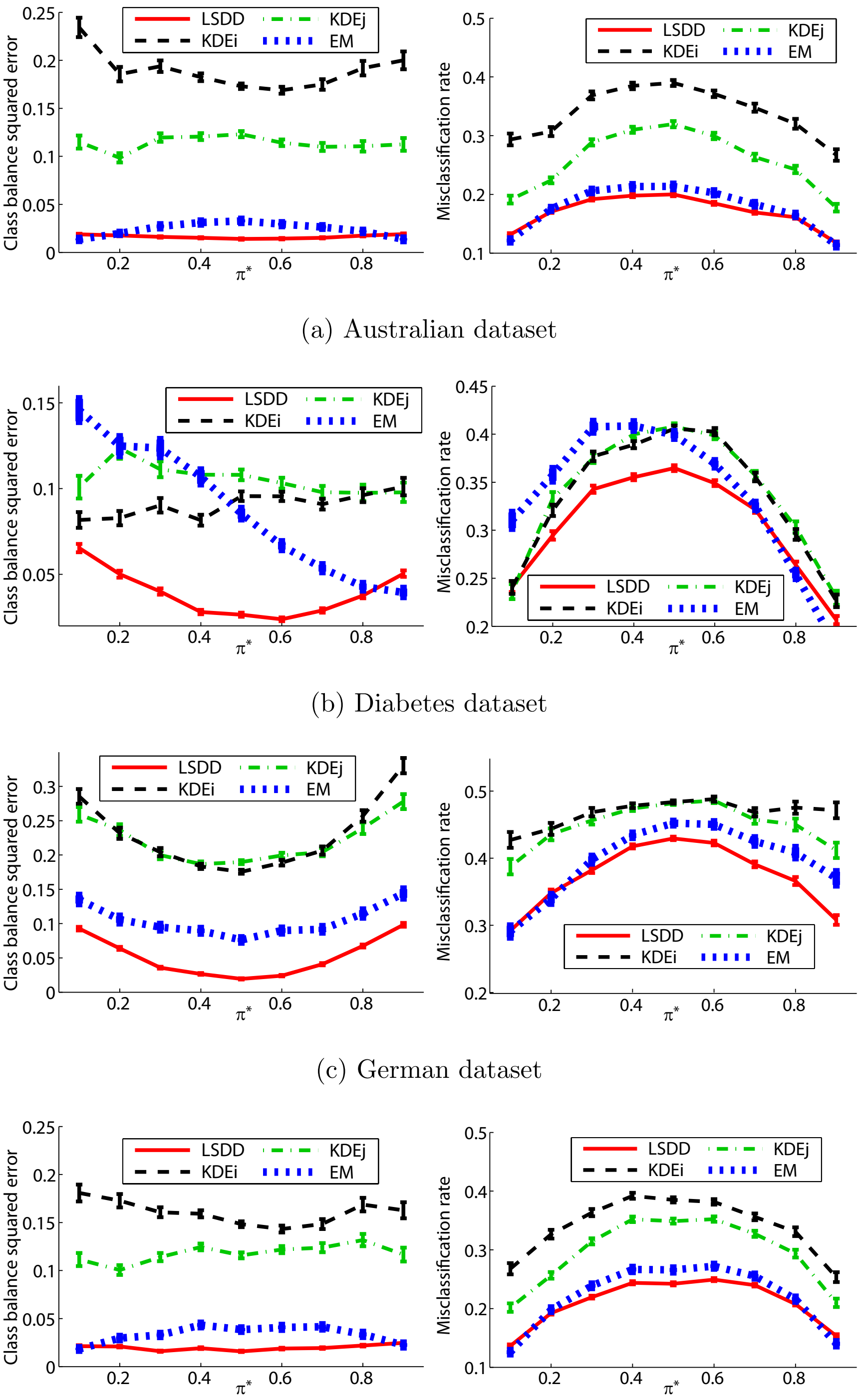
EM: The class-prior estimation method based on the expectation-maximization algorithm [142]. This method actually corresponds to distribution matching under the KL divergence.

The left graphs in Figure 21 plot the mean and standard error of the squared difference between true and estimated class-balances π over 1000 runs. These graphs show that LSDD tends to provide better class-balance estimates than alternative approaches.

Next, we use the estimated class balance to train a classifier. We use a weighted ℓ_2-regularized least-squares classifier [138]. That is, a class label $\widehat{y}$ for a test input $\boldsymbol{x}$ is estimated by

$$\widehat{y} = \mathrm{sign}\left(\sum_{\ell=1}^{n} \widehat{\theta}_\ell K(\boldsymbol{x}, \boldsymbol{x}_\ell)\right),$$

[11] `http://archive.ics.uci.edu/ml/`



(a) Australian dataset

(b) Diabetes dataset

(c) German dataset

(d) Statlogheart dataset

Figure 21: Results of class-balance adaptation. Left: Squared error of class-balance estimation. Right: Misclassification error by a weighted ℓ_2-regularized least-squares classifier with weighted cross-validation.

where $K(\boldsymbol{x}, \boldsymbol{x}')$ is the Gaussian kernel function with kernel width κ. $\{\widehat{\theta}_\ell\}_{\ell=1}^n$ are learned parameters given by

$$(\widehat{\theta}_1, \ldots, \widehat{\theta}_n) := \operatorname*{argmin}_{\theta_1,\ldots,\theta_n} \left[\sum_{i=1}^n \frac{\pi_{y_i}}{n_{y_i}/n} \left(\sum_{\ell=1}^n \theta_\ell K(\boldsymbol{x}_i, \boldsymbol{x}_\ell) - y_i \right)^2 + \delta \sum_{\ell=1}^n \theta_\ell^2 \right],$$

where $\pi_{+1} = \widehat{\pi}$, $\pi_{-1} = 1 - \widehat{\pi}$, $\widehat{\pi}$ is a class-balance estimate, and δ (≥ 0) is the regularization parameter. The Gaussian width κ and the regularization parameter δ are chosen by 5-fold weighted cross-validation [160] in terms of the misclassification error.

The right graphs in Figure 21 plot the test misclassification error over 1000 runs. The results show the LSDD-based method provides lower classification errors, which would be brought by good estimates of test class-balances.

4.4 Conclusion

In this article, we reviewed semi-supervised adaptive learning techniques for the covariate shift and class-balance change scenarios. In both cases, importance weighting plays an essential role. See [133] for more general discussion on learning under different training and test distributions.

If input-output samples are available from both training and test domains, weighted learning according to the joint importance $p'(\boldsymbol{x}, y)/p(\boldsymbol{x}, y)$ can in principle be used for transferring training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ to the test domain even when $p(\boldsymbol{x}, y)$ and $p'(\boldsymbol{x}, y)$ do not have an explicit link such as the covariate shift and class-balance change [19, 168]. In this situation, not only transferring information from the training domain to the test domain, but also the opposite transfer from the test domain to the training domain is possible simultaneously. This is the idea of *multi-task learning* [30] and is also an important branch of modern machine learning research.

Learning from input-output samples has already been studied extensively in statistics and machine learning. However, collecting input-output samples is often expensive and time-consuming in practice. Therefore, learning with side information such as additional input-only samples (semi-supervised learning) and additional related learning tasks (transfer learning and multi-task learning), as well as new models of input-output data collection such as *crowdsourcing* [137] and *self-taught learning* [134], will be important challenges in the arriving *big data* era.

5 Information-Maximization Clustering

5.1 Introduction

The goal of *clustering* is to classify data samples into disjoint groups in an unsupervised manner. *K-means* [117] 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* [58] performs k-means in a feature space induced by a reproducing kernel function [145]. *Spectral clustering* [149, 123] first unfolds non-linear data manifolds by a spectral embedding method, and then performs k-means in the embedded space. *Blurring mean-shift* [53, 28] 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 [203, 14]. *Dependence-maximization clustering* determines cluster assignments so that their dependence on input data is maximized [153, 50]. See Section 5.3 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[12]. 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 [118, 148, 37, 15], but they considered supervised setups, i.e., labeled samples are assumed to be given. [216] 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 [1, 60]. In this information-maximization approach, probabilistic classifiers such as a kernelized Gaussian classifier [1] and a kernel logistic regression classifier [60] 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 [1] or a quasi-Newton method [60]. Furthermore, [1] 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 decomposition. Furthermore, for kernel parameter optimization, we propose to use a non-parametric SMI estimator called *least-squares MI* (LSMI) [179, 157], which was proved

[12] "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.

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 5.2, 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 5.3 and quantitatively in Section 5.4. Finally, this paper is concluded in Section 5.5.

5.2 Information-Maximization Clustering with Squared-Loss Mutual Information

In this section, we describe our proposed clustering algorithm.

5.2.1 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 [1, 60]. 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 [153, 50] (see also Section 5.3.7) 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 $p_{\boldsymbol{\theta}}(y|\boldsymbol{x}; \boldsymbol{\alpha})$. 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 $p_{\boldsymbol{\theta}}(y|\boldsymbol{x}; \boldsymbol{\alpha})$, i.e., a cluster assignment for a new feature vector can be obtained based on the learned discriminative model.

5.2.2 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

$$\mathrm{SMI} := \frac{1}{2}\int \sum_{y=1}^{c} p(\boldsymbol{x})p(y)\left(\frac{p(\boldsymbol{x},y)}{p(\boldsymbol{x})p(y)} - 1\right)^2 \mathrm{d}\boldsymbol{x}, \tag{28}$$

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* [128] from $p(\boldsymbol{x},y)$ to $p(\boldsymbol{x})p(y)$, while the ordinary MI [38],

$$\mathrm{MI} := \int \sum_{y=1}^{c} p(\boldsymbol{x},y)\log\frac{p(\boldsymbol{x},y)}{p(\boldsymbol{x})p(y)}\mathrm{d}\boldsymbol{x}, \tag{29}$$

is the *Kullback-Leibler divergence* [103] 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 [5, 39]), 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 [1, 60] (see also Section 5.3.8), 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.

5.2.3 Clustering by SMI Maximization

Here, we give a computationally-efficient clustering algorithm based on SMI (28).

Expanding the squared term in Eq.(28), we can express SMI as

$$\begin{aligned}\mathrm{SMI} &= \frac{1}{2}\int \sum_{y=1}^{c} p(\boldsymbol{x})p(y)\left(\frac{p(\boldsymbol{x},y)}{p(\boldsymbol{x})p(y)}\right)^2 \mathrm{d}\boldsymbol{x} - \int \sum_{y=1}^{c} p(\boldsymbol{x})p(y)\frac{p(\boldsymbol{x},y)}{p(\boldsymbol{x})p(y)}\mathrm{d}\boldsymbol{x} + \frac{1}{2}\\ &= \frac{1}{2}\int \sum_{y=1}^{c} p(y|\boldsymbol{x})p(\boldsymbol{x})\frac{p(y|\boldsymbol{x})}{p(y)}\mathrm{d}\boldsymbol{x} - \frac{1}{2}. \end{aligned}\tag{30}$$

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:

$$p(y) = \frac{1}{c} \text{ for } y = 1, \ldots, c, \tag{31}$$

which will be non-informative and thus allow us to avoid biasing clustering solutions[13]. Substituting π_y into $p(y)$, we can express Eq.(30) as

$$\frac{1}{2}\int\sum_{y=1}^{c}\frac{1}{\pi_y}p(y|\boldsymbol{x})p(\boldsymbol{x})p(y|\boldsymbol{x})\mathrm{d}\boldsymbol{x}-\frac{1}{2}. \tag{32}$$

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

$$p_{\boldsymbol{\theta}}(y|\boldsymbol{x};\boldsymbol{\alpha}) := \sum_{i=1}^{n}\alpha_{y,i}K(\boldsymbol{x},\boldsymbol{x}_i), \tag{33}$$

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* [216]:

$$K(\boldsymbol{x}_i,\boldsymbol{x}_j)=\begin{cases}\exp\left(-\dfrac{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2}{2\sigma_i\sigma_j}\right) & \text{if } \boldsymbol{x}_i\in\mathcal{N}_t(\boldsymbol{x}_j) \text{ or } \boldsymbol{x}_j\in\mathcal{N}_t(\boldsymbol{x}_i),\\ 0 & \text{otherwise,}\end{cases} \tag{34}$$

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.(32) by the empirical average of samples $\{\boldsymbol{x}_i\}_{i=1}^n$, we arrive at the following SMI approximator:

$$\widehat{\mathrm{SMI}} := \frac{1}{2n}\sum_{y=1}^{c}\frac{1}{\pi_y}\boldsymbol{\alpha}_y^\top\boldsymbol{K}^2\boldsymbol{\alpha}_y-\frac{1}{2}, \tag{35}$$

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

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 $\boldsymbol{K}$ [73]. 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

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

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

[13] Such a cluster-balance constraint is often employed in existing clustering algorithms [149, 203, 126].

On the other hand, since

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

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

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

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 $p(y|\boldsymbol{x}_i)$:

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

where $\boldsymbol{0}_n$ 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

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

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

Discussions: Given an SMI approximator $\widehat{\mathrm{SMI}}$ defined by Eq.(35), 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 5.4.

5.2.4 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.(35) 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) [179] was shown to achieve the optimal convergence rate. For this reason, we propose to use LSMI for model selection, instead of $\widehat{\mathrm{SMI}}$ (35).

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* [168],

$$r(\boldsymbol{x}, y) := \frac{p(\boldsymbol{x}, y)}{p(\boldsymbol{x})p(y)}, \tag{36}$$

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:

$$r_{\boldsymbol{\theta}}(\boldsymbol{x}, y; \boldsymbol{\theta}) := \sum_{\ell : y_\ell = y} \theta_\ell L(\boldsymbol{x}, \boldsymbol{x}_\ell), \tag{37}$$

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:

$$L(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\gamma^2}\right), \tag{38}$$

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:

$$\min_{\boldsymbol{\theta}} \frac{1}{2} \int \sum_{y=1}^{c} \Big(r_{\boldsymbol{\theta}}(\boldsymbol{x}, y; \boldsymbol{\theta}) - r(\boldsymbol{x}, y)\Big)^2 p(\boldsymbol{x})p(y)\mathrm{d}\boldsymbol{x}. \tag{39}$$

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 (39) is given for each y as follows:

$$\min_{\boldsymbol{\theta}_y} \left[\frac{1}{2}\boldsymbol{\theta}_y^\top \widehat{\boldsymbol{H}}^{(y)} \boldsymbol{\theta}_y - \boldsymbol{\theta}_y^\top \widehat{\boldsymbol{h}}^{(y)} + \frac{\delta}{2}\boldsymbol{\theta}_y^\top \boldsymbol{\theta}_y\right], \tag{40}$$

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

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

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[14]:

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

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.(40). [179] 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 $\mathcal{Z} \backslash \mathcal{Z}_m$ (i.e., all samples without $\mathcal{Z}_m$), and its out-of-sample error (which corresponds to Eq.(39) 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 (28),

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

[14] Note that, in the original LSMI paper [179], 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 $r_{\boldsymbol{\theta}}(\boldsymbol{x}, y; \boldsymbol{\theta})$ defined by Eq.(37) 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 (37).

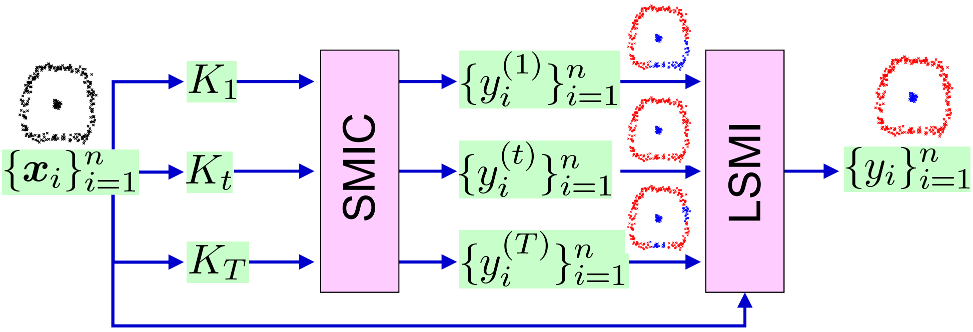


Figure 22: Schematic of the proposed clustering algorithm. We prepare T kernel candidates $\{K_t(\boldsymbol{x}, \boldsymbol{x}')\}_{t=1}^T$, compute cluster assignments $\{y_i^{(t)}\}_{i=1,t=1}^{n,T}$ by SMIC, and choose the best one that maximizes LSMI.

```
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 23: 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 (33). Pseudo codes of SMIC and LSMI are described in Figure 24 and Figure 25, respectively.

the SMI estimator called LSMI is given as follows:

$$\mathrm{LSMI} := -\frac{1}{2n^2}\sum_{i,j=1}^{n} \widehat{r}(\boldsymbol{x}_i, y_j)^2 + \frac{1}{n}\sum_{i=1}^{n} \widehat{r}(\boldsymbol{x}_i, y_i) - \frac{1}{2}, \tag{41}$$

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 (33), and choose the one that maximizes LSMI. See Figure 22 for a schematic. A pseudo code of the entire SMI-maximization clustering procedure is summarized in Figures 23–25.

Discussions: $\widehat{\mathrm{SMI}}$ given by Eq.(35) 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

```
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 $y = 1, \ldots, c$;
$\widetilde{\boldsymbol{\phi}}_y \longleftarrow \boldsymbol{\phi}_y \times \mathrm{sign}(\boldsymbol{\phi}_y^\top \boldsymbol{1}_n)$ for $y = 1, \ldots, c$;
$y_i \longleftarrow \mathop{\mathrm{argmax}}_{y \in \{1,\ldots,c\}} \frac{[\max(\boldsymbol{0}_n, \widetilde{\boldsymbol{\phi}}_y)]_i}{\max(\boldsymbol{0}_n, \widetilde{\boldsymbol{\phi}}_y)^\top \boldsymbol{1}_n}$ for $i = 1, \ldots, n$;
$\mathcal{Y} \longleftarrow \{y_i\}_{i=1}^n$;
```

Figure 24: 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 (33). 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 $\mathrm{argmax}_y \frac{\pi_y [\max(\boldsymbol{0}_n, \widetilde{\boldsymbol{\phi}}_y)]_i}{\max(\boldsymbol{0}_n, \widetilde{\boldsymbol{\phi}}_y)^\top \boldsymbol{1}_n}$.

```
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 $(\gamma, \delta)$ 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
        $\mathrm{CV}(\gamma, \delta) \longleftarrow \frac{1}{M} \sum_{m=1}^M \mathrm{CV}_m(\gamma, \delta)$;
    end
end
$(\widehat{\gamma}, \widehat{\delta}) \longleftarrow \mathop{\mathrm{argmin}}_{\gamma \in \Gamma, \delta \in \Delta} \mathrm{CV}(\gamma, \delta)$;
$\widehat{r}(\boldsymbol{x}, y) \longleftarrow$ Density ratio estimator for $(\widehat{\gamma}, \widehat{\delta})$ using $\mathcal{Z}$;
$\mathrm{LSMI} \longleftarrow -\frac{1}{2n^2} \sum_{i,j=1}^n \widehat{r}(\boldsymbol{x}_i, y_j)^2 + \frac{1}{n} \sum_{i=1}^n \widehat{r}(\boldsymbol{x}_i, y_i) - \frac{1}{2}$;
```

Figure 25: 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 (37).

assignments in a dependence-maximizing way [153, 50]. 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.

5.2.5 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 [99], Lemma 5.2 of [100], and pp.33–34 in [197].

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*

   $$\Lambda_j(\boldsymbol{A}+\boldsymbol{B}) \subset \mathcal{B}(\mu_j, \epsilon), \tag{42}$$

   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*

   $$\mathrm{tr}(\boldsymbol{P}_j(\boldsymbol{A}+\boldsymbol{B})) = \mathrm{tr}(\boldsymbol{P}_j(\boldsymbol{A})) \tag{43}$$

   and

   $$\|\boldsymbol{P}_j(\boldsymbol{A}+\boldsymbol{B}) - \boldsymbol{P}_j(\boldsymbol{A})\|_{\mathrm{Frob}} \leq 4\|\boldsymbol{B}\|_{\mathrm{Frob}}/\delta_r. \tag{44}$$

Intuitively speaking, Eq.(42) says that the perturbed eigenvalues are close to the original eigenvalues, Eq.(44) says that the perturbed eigenspaces are close to the original eigenspaces, and Eq.(43) 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.(35),

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

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

$$|v-v'|<\|\boldsymbol{\Delta}\|_{\mathrm{Frob}}/\pi_1, \tag{45}$$

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

$$\|\boldsymbol{\phi}_y-\boldsymbol{\phi}_y'\|_2\le 4\|\boldsymbol{\Delta}\|_{\mathrm{Frob}}/\delta_r \textit{ for } y=1,\ldots,c, \tag{46}$$

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

A proof of Theorem 2 is provided in Appendix 5.6. 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.(46). Without loss of generality, we further assume that*

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

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

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

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 5.7. 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.

5.3 Existing Clustering Methods

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

5.3.1 K-Means Clustering

K-means clustering [117] would be one of the most popular clustering algorithms. It tries to minimize the following distortion measure with respect to the cluster assignments $\{y_i\}_{i=1}^n$:

$$\sum_{y=1}^{c} \sum_{i:y_i=y} \|\boldsymbol{x}_i - \boldsymbol{\mu}_y\|^2, \tag{47}$$

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 [46]. 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 [58].

As the optimization problem of (kernel) k-means is NP-hard [6], 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 [217, 44]. 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 5.2.3. 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.

5.3.2 Spectral Clustering

The basic idea of *spectral clustering* [149, 123] 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 [43].

The performance of spectral clustering depends heavily on the choice of sample-sample similarity $W_{i,j}$. [216] 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 [216, 156]. 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 5.2.3. 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 (33). Furthermore, our proposed approach provides a systematic model selection strategy, which is a notable advantage over spectral clustering.

5.3.3 Blurring Mean-Shift Clustering

Blurring mean-shift [53] 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 [152] 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.

[29] showed that the blurring mean-shift algorithm can be interpreted as an *expectation-maximization algorithm* [41], 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 $\boldsymbol{W}$ [35]. $\boldsymbol{D}$ 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* [59] 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 5.3.2). Although $\boldsymbol{P}$ depends on $\boldsymbol{X}$ in reality, [28] 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 [34]), and therefore a sensible stopping criterion is necessary in practice. Although [28] gave a useful heuristic for stopping the iteration, it is not clear whether this heuristic always works well in practice.

5.3.4 Discriminative Clustering

The *support vector machine* (SVM) [193] is a supervised discriminative classifier that tries to find a hyperplane separating positive and negative samples with the maximum margin.

[203] 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:

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

where $\circ$ denotes the *Hadamard product* (also known as the entry-wise product), and ε and C are tuning parameters. The constraint $-\varepsilon \le \mathbf{1}_n^\top \boldsymbol{y} \le \varepsilon$ 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 [203]. Since then, several approaches have been developed for further improving the computational efficiency of MMC [191, 220, 219, 108, 199].

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) [14] 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

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

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.

5.3.5 Generative Clustering

In the *generative clustering* framework [46], class labels are determined by

$$\widehat{y} = \underset{y}{\mathrm{argmax}}\ p(y|\boldsymbol{x}) = \underset{y}{\mathrm{argmax}}\ p(\boldsymbol{x}, y),$$

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

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

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

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 $p_{\boldsymbol{\theta}}(\boldsymbol{x}, y; \boldsymbol{\beta}, \boldsymbol{\pi})$. An approach to coping with this problem is to consider a *marginal* model,

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

and learns the parameters $\boldsymbol{\beta}$ and $\boldsymbol{\pi}$ by maximum likelihood estimation [46]:

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

Since the likelihood function of the above mixture model is non-convex, a *gradient method* [7] may be used for finding a local maximizer in practice. For determining the number of clusters (mixtures) and the mixing-element model $p_{\boldsymbol{\theta}}(\boldsymbol{x}|y; \boldsymbol{\beta})$, *likelihood cross-validation* [69] 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* [41] for finding a local maximizer of the joint likelihood:

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

A more flexible variant of the EM algorithm called the *split-and-merge EM algorithm* [189] 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 [24]. Let us introduce prior distributions $p_{\boldsymbol{\theta}}(\boldsymbol{\beta})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{\pi})$ for the parameters $\boldsymbol{\beta}$ and $\boldsymbol{\pi}$. Then the posterior distribution of the parameters is expressed as

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

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

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

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 $p_{\boldsymbol{\theta}}(\boldsymbol{x}|y;\boldsymbol{\beta})$, 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 $p_{\boldsymbol{\theta}}(y;\boldsymbol{\pi})$, the Dirichlet prior is assumed. Otherwise, the posterior distribution is approximated by the *Laplace approximation* [116], the *Markov chain Monte Carlo sampling* [10], or the *variational approximation* [13, 57]. The number of clusters can be determined based on the maximization of the *marginal likelihood*:

$$p_{\boldsymbol{\theta}}(\mathcal{X}) = \operatorname*{argmax}_y \iint p_{\boldsymbol{\theta}}(\mathcal{X}|\boldsymbol{\beta}, \boldsymbol{\pi}) p_{\boldsymbol{\theta}}(\boldsymbol{\beta}) p_{\boldsymbol{\theta}}(\boldsymbol{\pi}) \mathrm{d}\boldsymbol{\beta} \mathrm{d}\boldsymbol{\pi}. \tag{48}$$

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.

5.3.6 Posterior-Maximization Clustering

Another possible clustering approach based on probabilistic inference is to directly maximizes the posterior probability of class labels $\mathcal{Y} = \{y_i\}_{i=1}^n$ [24]:

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

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

An approximate inference method called *iterative conditional modes* [104] alternatively maximizes the posterior probabilities of $\mathcal{Y}$ and $\boldsymbol{\beta}$ until convergence:

$$\begin{aligned}\widehat{\mathcal{Y}} &\longleftarrow p_{\boldsymbol{\theta}}(\mathcal{Y}|\mathcal{X}, \widehat{\boldsymbol{\beta}}),\\ \widehat{\boldsymbol{\beta}} &\longleftarrow p_{\boldsymbol{\theta}}(\boldsymbol{\beta}|\mathcal{X}, \widehat{\mathcal{Y}}).\end{aligned}$$

When the Gaussian model with covariance identity is assumed for $p_{\boldsymbol{\theta}}(\mathcal{Y}|\mathcal{X}, \boldsymbol{\beta})$, this algorithm is reduced to the k-means algorithm (see Section 5.3.1) under the uniform priors.

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

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

Similarly to generative clustering described in Section 5.3.5, 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 (48). 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* [51, 11] 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 [121] and variational approximation [25] have been introduced. Furthermore, variants of Dirichlet processes such as hierarchical Dirichlet processes [181], nested Dirichlet processes [141], and dependent Dirichlet processes [109] 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.

5.3.7 Dependence-Maximization Clustering

The *Hilbert-Schmidt independence criterion* (HSIC) [61] is a dependence measure based on a reproducing kernel function $K(\boldsymbol{x}, \boldsymbol{x}')$ [12]. [153] 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 5.3.4) that maximizes

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

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. [153] used a greedy algorithm to optimize the cluster indicator matrix, which is computationally demanding. [215] 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 [145]. 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 [50]. MNN is based on the k-nearest-neighbor entropy estimator proposed by [102].

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

$$\sum_{y=1}^{c} \frac{1}{n_y - 1} \sum_{i \neq j : y_i = y_j = y} \log(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 + \epsilon), \tag{49}$$

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.

5.3.8 Information-Maximization Clustering with Mutual Information

Finally, we review methods of information-maximization clustering based on *mutual information* [1, 60], which belong to the same family of clustering algorithms as our proposed method.

Mutual information (MI) is defined and expressed as

$$\begin{aligned}\mathrm{MI} &:= \int \sum_{y=1}^{c} p(\boldsymbol{x}, y) \log \frac{p(\boldsymbol{x}, y)}{p(\boldsymbol{x})p(y)} \mathrm{d}\boldsymbol{x} \\ &= \int \sum_{y=1}^{c} p(y|\boldsymbol{x})p(\boldsymbol{x}) \log p(y|\boldsymbol{x}) \mathrm{d}\boldsymbol{x} - \int \sum_{y=1}^{c} p(y|\boldsymbol{x})p(\boldsymbol{x}) \log p(y) \mathrm{d}\boldsymbol{x}. \end{aligned} \tag{50}$$

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

$$p(y) = \int p(y|\boldsymbol{x})p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \approx \frac{1}{n} \sum_{i=1}^{n} p(y|\boldsymbol{x}_i; \boldsymbol{\alpha}). \tag{51}$$

By further approximating the expectation with respect to $p(\boldsymbol{x})$ included in Eq.(50) by the empirical average of samples $\{\boldsymbol{x}_i\}_{i=1}^n$, the following MI estimator can be obtained [1, 60]:

$$\begin{aligned}\widehat{\mathrm{MI}} := \frac{1}{n}\sum_{i=1}^n \sum_{y=1}^c p(y|\boldsymbol{x}_i;\boldsymbol{\alpha}) \log p(y|\boldsymbol{x}_i;\boldsymbol{\alpha}) \\ -\sum_{y=1}^c \left(\frac{1}{n}\sum_{i=1}^n p(y|\boldsymbol{x}_i;\boldsymbol{\alpha})\right) \log \left(\frac{1}{n}\sum_{j=1}^n p(y|\boldsymbol{x}_j;\boldsymbol{\alpha})\right). \qquad (52)\end{aligned}$$

In [1], 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 [60], the logistic model

$$p(y|\boldsymbol{x};\boldsymbol{\alpha}) \propto \exp\left(\boldsymbol{\alpha}_y^\top \boldsymbol{x}\right), \qquad (53)$$

(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

$$y_i = \underset{y}{\mathrm{argmax}}\ p(y|\boldsymbol{x}_i;\widehat{\boldsymbol{\alpha}}),$$

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. [1] proposed to choose the kernel parameters so that $\widehat{\mathrm{MI}}$ (52) 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) [180], which was proved to achieve the optimal non-parametric convergence rate.

The derivation of MLMI follows a similar line to LSMI explained in Section 5.2.4, i.e., the density-ratio function (36) is learned. More specifically, the following density-ratio model $r_{\boldsymbol{\theta}}(\boldsymbol{x}, y; \boldsymbol{\theta})$ is used:

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

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 $r_{\boldsymbol{\theta}}(\boldsymbol{x}, y; \boldsymbol{\theta})p(\boldsymbol{x})p(y)$ is minimized[15]. An empirical version of the MLMI optimization problem is given as

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

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 [173].

Then an MI estimator called MLMI is given as follows:

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

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 [180].

5.4 Experiments

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

5.4.1 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 equidistantly 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.

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

(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 26). As a kernel function, we used the sparse local-scaling kernel (34) for SMIC, where the kernel parameter t was chosen from $\{1, \ldots, 10\}$ based on LSMI with the Gaussian kernel (38).

The top graphs in Figure 26 depict the cluster assignments obtained by SMIC with the uniform class-prior, and the bottom graphs in Figure 26 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) [74] between inferred cluster assignments and the ground truth categories (see Appendix 5.8 for the details of ARI). Larger ARI values mean better performance, and 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 27 depicts the cluster assignments and model selection curves obtained by MIC with MLMI (see Section 5.3.8), where pre-training of the kernel logistic model using the cluster assignments obtained by *self-tuning spectral clustering* [216] was carried out for initializing MIC [60]. 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 (52).

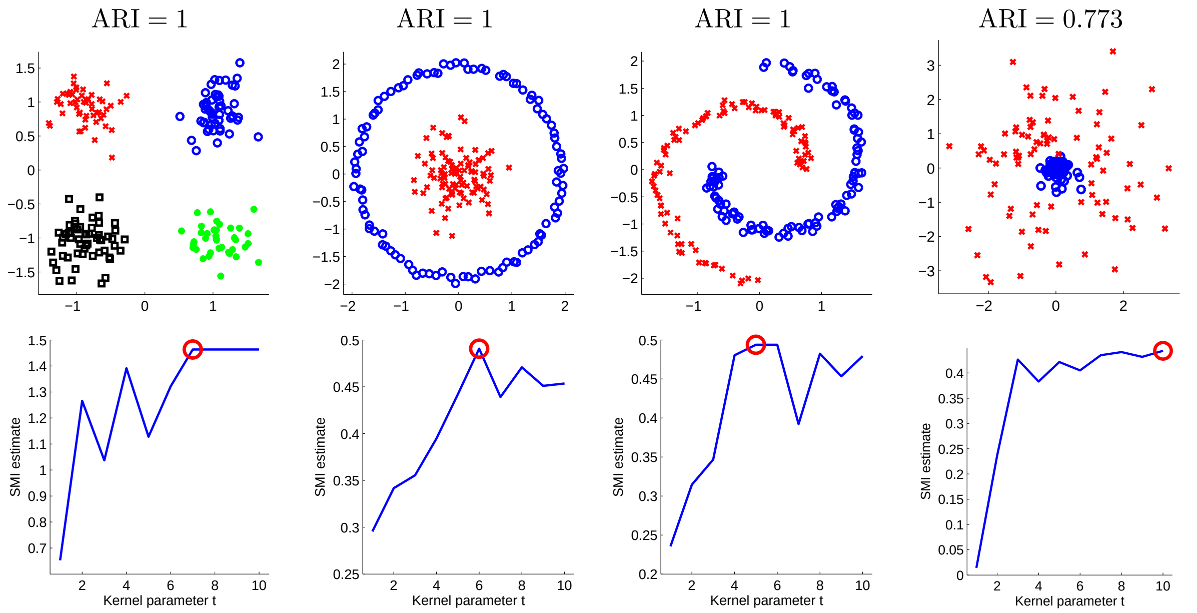
Figure 28 and Figure 29 depict class-posterior probabilities estimated by SMIC and MIC, respectively. The plots show that, for the datasets (a), (b), and (c) where the clusters are clearly separated, the estimated class-posterior probabilities are almost zero-one functions and thus the class prediction 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.

5.4.2 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.

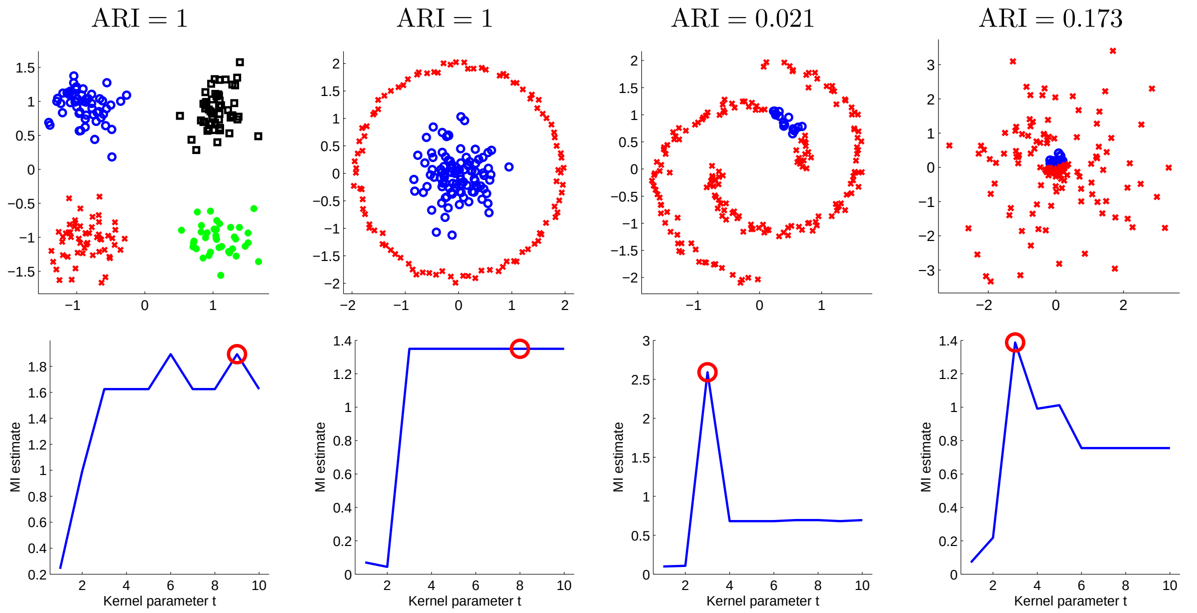
We continue using the 4 artificial datasets used in Section 5.4.1, but we set the true class-prior probability as

$$\begin{aligned} p(y=1) &= p(y=2) = 0.1, 0.15, 0.2, 0.25, \\ p(y=3) &= p(y=4) = \frac{1 - p(y=1) - p(y=2)}{2}, \end{aligned}$$



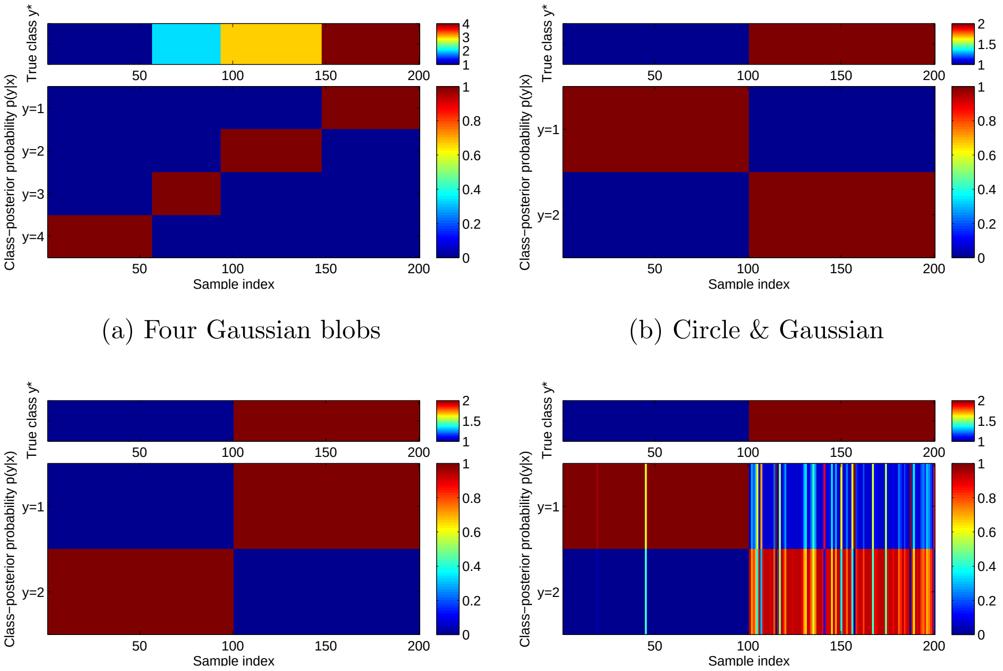
(a) Four Gaussian blobs (b) Circle & Gaussian (c) Double spirals (d) High & low densities

Figure 26: 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 27: Illustrative examples. Cluster assignments obtained by MIC (top) and model selection curves obtained by MLMI (bottom).



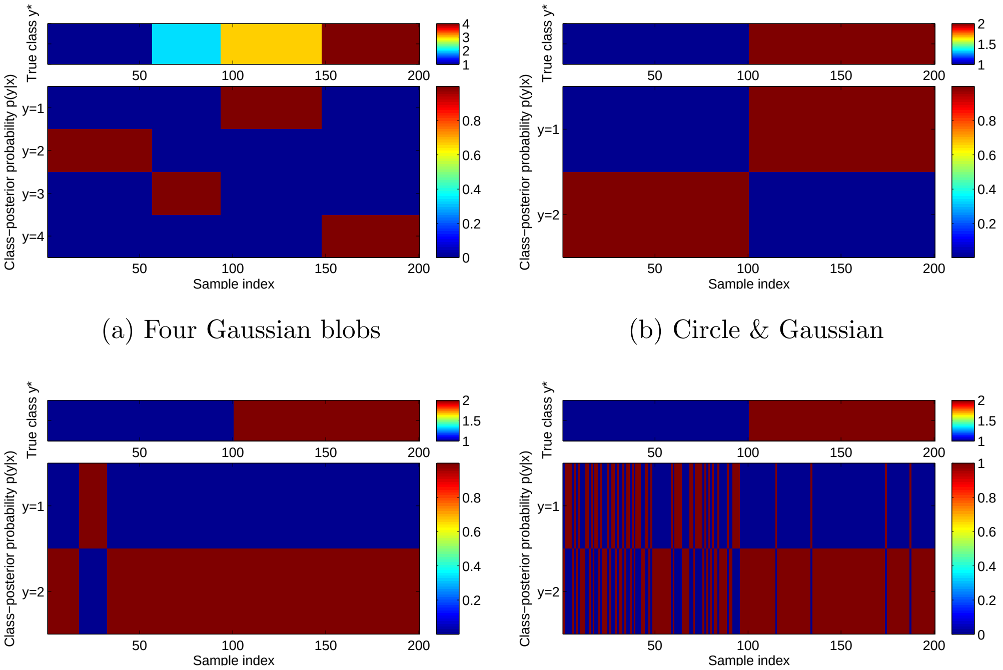
(a) Four Gaussian blobs

(b) Circle & Gaussian

(c) Double spirals

(d) High & low densities

Figure 28: Illustrative examples. Class-posterior probabilities estimated by SMIC.



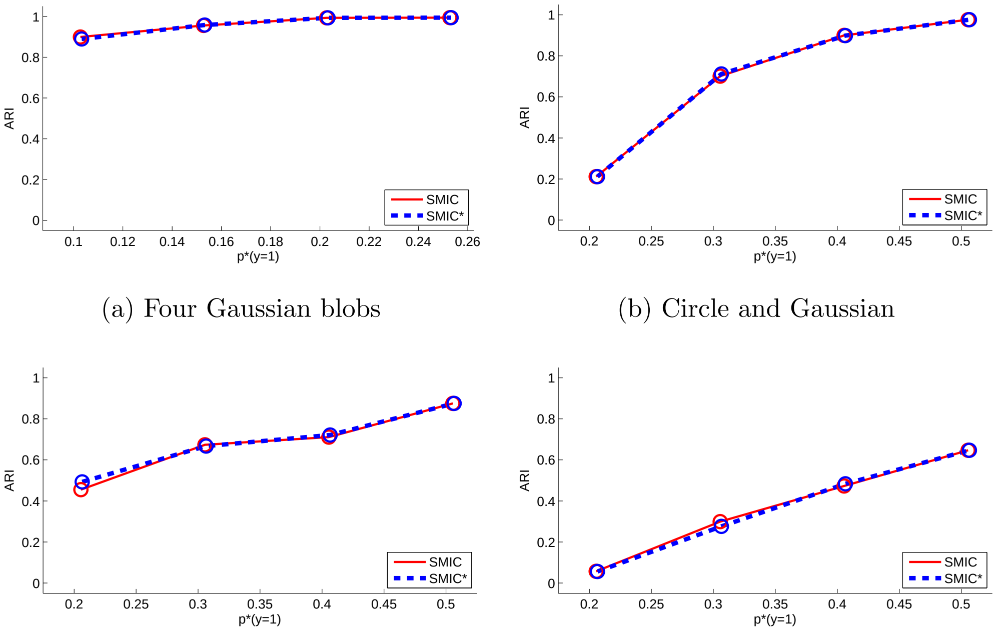
(a) Four Gaussian blobs

(b) Circle & Gaussian

(c) Double spirals

(d) High & low densities

Figure 29: 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 30: 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%.

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 30, 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.

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.

5.4.3 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.

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 [117] (see also Section 5.3.1). 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 [149, 123] (see also Section 5.3.2) with the Gaussian similarity. The Gaussian width is set to the median distance between all samples, which is a popular heuristic in kernel methods [145]. We used the publicly available MATLAB code[16], 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 [216], instead of the Gaussian similarity.

MNN: Mean nearest-neighbor clustering [50] (see also Section 5.3.7). We used the MATLAB code provided by one of the authors[17]. Following the suggestions provided in the program code, the number of iterations was set to 10 and the smoothing parameter ϵ (see Eq.(49)) was set to $\epsilon = 1/n$.

MIC: MI-based clustering with kernel logistic models and the sparse local-scaling kernel [60] (see also Section 5.3.8), where model selection is carried out by maximum-likelihood MI (MLMI) [180]. 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[18]. 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 (34) was chosen from $\{1, \ldots, 10\}$ based on MLMI with Gaussian kernels (see Section 5.3.8). The Gaussian kernel width in MLMI was chosen from $\{10^{-2}, 10^{-1.5}, 10^{-1}, \ldots, 10^{2}\}$

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

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

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

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 5.2.3), where model selection is carried out by least-squares MI (LSMI) [179] (see also Section 5.2.4). We implemented SMIC and LSMI using MATLAB by ourselves. The tuning parameter t included in the sparse local-scaling kernel (34) was chosen from $\{1, \ldots, 10\}$ based on LSMI with Gaussian kernels (see Section 5.2.4). 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.

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

Datasets We used the following 6 real-world datasets.

Digit ($d = 256, n = 5000,$ and $c = 10$)**:** The *USPS* hand-written digit dataset[19], which contains 9298 digit images. Each image consists of 256 (= 16 × 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[20], which contains 400 gray-scale face images (40 people; 10 images per person). Each image consists of 4096 (= 64 × 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[21], 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 [78], 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) [130, 79] to the 700 samples, and extracted 50-dimensional feature vectors.

[19] `http://www.gaussianprocess.org/gpml/data/`

[20] `http://www.cs.toronto.edu/~roweis/data.html`

[21] `http://people.csail.mit.edu/jrennie/20Newsgroups/`

Word ($d = 50, n = 300,$ and $c = 3$)**:** The *SENSEVAL-2* dataset[22] 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 [127], 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* [106]. 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.

Accelerometry ($d = 5, n = 300,$ and $c = 3$)**:** The *ALKAN* dataset[23], 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 [66]. 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* [17, 18]. 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 [80]. 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.

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

[22] `http://www.senseval.org/`

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

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 5.2.4).

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.

5.5 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 [1, 60], 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

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)**

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 (34) 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 (38) 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 [153], 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 [115, 47, 90, 101, 91, 55, 54]. 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 [145]. 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.(35) for finding clustering solutions and LSMI defined by Eq.(41) 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.

5.6 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 $\pi_1 \le \cdots \le \pi_c$ are class-prior probabilities, $\lambda_1(\boldsymbol{K}) \ge \cdots \ge \lambda_c(\boldsymbol{K}) \ge 0$ 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

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

where, in the third line, we used $|\lambda_y(\boldsymbol{K}) - \lambda_y(\boldsymbol{K}')| < \|\boldsymbol{\Delta}\|_{\mathrm{Frob}}$ implied by Eqs.(42) and (43), and we used in the last line $\mathrm{tr}(\boldsymbol{K}) = \mathrm{tr}(\boldsymbol{K}') = n$ implied by the assumption $K(\boldsymbol{x}, \boldsymbol{x}) = 1$ for all $\boldsymbol{x}$. Thus, Eq.(45) was proved.

Eq.(46) is immediately implied by Eq.(44). 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. □

5.7 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*

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

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*

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

Proof. By definition,

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

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

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

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

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

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	$m_{\bar{\mathcal{C}},\mathcal{C}^*}$	$m_{\bar{\mathcal{C}},\bar{\mathcal{C}}^*}$

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.(46). Finally, dividing the above inequality by $\sqrt{n}$ completes the proof. □

5.8 Rand Index and Adjusted Rand Index

Here, we review the definitions of the *Rand index* (RI) [135] and the *adjusted Rand index* (ARI) [74], 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:

$$\begin{aligned}\mathcal{C}_y &= \{y_i \mid y_i = y\},\\ \mathcal{C}_y^* &= \{y_i^* \mid y_i^* = y\}.\end{aligned}$$

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

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

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) [135] is defined and expressed as

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

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 [74]. 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

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

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.

6 List of Publications and Significant Collaborations

6.1 Papers published in peer-reviewed journals

1. Kanamori, T., Suzuki, T., & Sugiyama, M. Computational complexity of kernel-based density-ratio estimation: A condition number analysis. Machine Learning, vol.90, no.3, pp.431-460, 2013.

2. * Yamada, M., Suzuki, T., Kanamori, T., Hachiya, H., & Sugiyama, M. Relative density-ratio estimation for robust distribution comparison. Neural Computation, vol.25, no.5, pp.1324-1370, 2013.

3. Sakai, T. & Sugiyama, M. Computationally efficient estimation of squared-loss mutual information with multiplicative kernel models. IEICE Transactions on Information and Systems, vol.E97-D, no.4, pp.968-971, 2014.

4. * Sugiyama, M., Suzuki, T., Kanamori, T., du Plessis, M. C., Liu, S., & Takeuchi, I. Density-difference estimation. Neural Computation, vol.25, no.10, pp.2734-2775, 2013.

5. Kanamori, T. & Sugiyama, M. Statistical analysis of distance estimators with density differences and density ratios. Entropy, vol.16, no.2, pp.921-942, 2014.

6. * Sugiyama, M., Liu, S., du Plessis, M. C., Yamanaka, M., Yamada, M., Suzuki, T., & Kanamori, T. Direct divergence approximation between probability distributions and its applications in machine learning. Journal of Computing Science and Engineering, vol.7, no.2, pp.99-111, 2013.

7. * Sugiyama, M., Yamada, M., & du Plessis, M. C. Learning under non-stationarity: Covariate shift and class-balance change. WIREs Computational Statistics, 13 pages, 2013.

8. du Plessis, M. C. & Sugiyama, M. Class prior estimation from positive and unlabeled data. IEICE Transactions on Information and Systems, vol.E97-D, no.5, pp.1358-1362, 2014.

9. * Liu, S., Quinn, J., Gutmann, M. U., & Sugiyama, M. Direct learning of sparse changes in Markov networks by density ratio estimation. Neural Computation, vol.26, no.6, pp.1169-1197, 2014.

10. Suzuki, T. & Sugiyama, M. Sufficient dimension reduction via squared-loss mutual information estimation. Neural Computation, vol.25, no.3, pp.725-758, 2013.

11. * Sugiyama, M., Niu, G., Yamada, M., Kimura, M., & Hachiya, H. Information-maximization clustering based on squared-loss mutual information. Neural Computation, vol.26, no.1, pp.84-131, 2014.

12. Sainui, J. & Sugiyama, M. Direct approximation of quadratic mutual information and its application to dependence-maximization clustering. IEICE Transactions on Information and Systems, vol.E96-D, no.19, pp.2282-2285, 2013.

13. Yamada, M., Sugiyama, M., & Sese, J. Least-squares independence regression for non-linear causal inference under non-Gaussian noise. Machine Learning, vol.96, no.3, pp.249-267, 2014.

14. Calandriello, D., Niu, G., & Sugiyama, M. Semi-supervised information-maximization clustering. Neural Networks, vol.57, pp.103-111, 2014.

15. Nguyen, T. D., du Plessis, M. C., Kanamori, T., & Sugiyama, M. Constrained least-squares density-difference estimation. IEICE Transactions on Information and Systems, vol.E97-D, no.7, pp.1822-1829, 2014.

16. Kanamori, T. & Sugiyama, M. Statistical analysis of distance estimators with density differences and density ratios. Entropy, vol.16, no.2, pp.921-942, 2014.

17. Sainui, J. & Sugiyama, M. Unsupervised dimension reduction via least-squares quadratic mutual information. IEICE Transactions on Information and Systems, vol.E97-D, no.10, pp.2806-2809, 2014.

18. Sugiyama, M., Niu, G., Yamada, M., Kimura, M., & Hachiya, H. Information-maximization clustering based on squared-loss mutual information. Neural Computation, vol.26, no.1, pp.84-131, 2014.

19. Shiga, M., Tangkaratt, V., & Sugiyama, M. Direct conditional probability density estimation with sparse feature selection. Machine Learning, to appear.

List of interactions with collaborators

- Makoto Yamada (Yahoo Labs)
- Taiji Suzuki (Tokyo Institute of Technology)
- Takafumi Kanamori (Nagoya University)
- Gang Niu (Baido Inc.)
- Marthinus Christoffel du Plessis (University of Tokyo)
- Makoto Kimura (Weblio Inc.)
- Hirotaka Hachiya (Canon Corp.)
- Masao Yamanaka (Canon Corp.)
- Tomoya Sakai (Tokyo Institute of Technology)
- Nam Hyunha (Tokyo Institute of Technology)
- Ichiro Takeuchi (Nagoya Institute of Technology)
- Jun Sese (National Institute of Advanced Industrial Science and Technology)
- Janya Sainui (Tokyo Institute of Technology)
- Daniele Calandriello (INRIA Lille)
- Motoki Shiga (Gifu University)

Attachments

The journal publications indicated by “*”are attached.

References

[1] F. Agakov and D. Barber. Kernelized infomax clustering. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 17–24, Cambridge, MA, USA, 2006. MIT Press.

[2] A. Agarwal and B. Triggs. Monocular human motion capture with a mixture of regressors. In *Proceedings of IEEE Workshop on Vision for Human Computer Interaction at Computer Vision and Pattern Recognition*, page 72, 2005.

[3] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, AC-19(6):716–723, 1974.

[4] T. Akiyama, H. Hachiya, and M. Sugiyama. Efficient exploration through active learning for value function approximation in reinforcement learning. *Neural Networks*, 23(5):639–648, 2010.

[5] S. M. Ali and S. D. Silvey. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society, Series B*, 28(1):131–142, 1966.

[6] D. Aloise, A. Deshpande, P. Hansen, and P. Popat. NP-hardness of Euclidean sum-of-squares clustering. *Machine Learning*, 75(2):245–249, 2009.

[7] S. Amari. Theory of adaptive pattern classifiers. *IEEE Transactions on Electronic Computers*, EC-16(3):299–307, 1967.

[8] S. Amari and H. Nagaoka. *Methods of Information Geometry.* Oxford University Press, Providence, RI, USA, 2000.

[9] N. Anderson, P. Hall, and D. Titterington. Two-sample test statistics for measuring discrepancies between two multivariate probability density functions using kernel-based density estimates. *Journal of Multivariate Analysis*, 50(1):41–54, 1994.

[10] C. Andrieu, N. de Freitas, A. Doucet, and M.l I. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50(1-2):5–43, 2003.

[11] C. Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, 2(6):1152–1174, 1974.

[12] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404, 1950.

[13] H.i Attias. A variational Baysian framework for graphical models. In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 209–215. MIT Press, 2000.

[14] F. Bach and Z. Harchaoui. DIFFRAC: A discriminative and flexible framework for clustering. In J. C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 49–56, Cambridge, MA, USA, 2008. MIT Press.

[15] F. Bach and M. I. Jordan. Learning spectral clustering, with application to speech separation. *Journal of Machine Learning Research*, 7:1963–2001, 2006.

[16] O. Banerjee, L. El Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *Journal of Machine Learning Research*, 9:485–516, March 2008.

[17] L. Bao and S. S. Intille. Activity recognition from user-annotated acceleration data. In *Proceedings of 2nd IEEE International Conference on Pervasive Computing*, pages 1–17, 2004.

[18] N. B. Bharatula, M. Stager, P. Lukowicz, and G. Troster. Empirical study of design choices in multi-sensor context ecognition. In *Proceedings of International Forun on Applied Wearable Computing*, pages 79–93, 2005.

[19] S. Bickel, J. Bogojeska, T. Lengauer, and T. Scheffer. Multi-task learning for HIV therapy screening. In A. McCallum and S. Roweis, editors, *Proceedings of 25th Annual International Conference on Machine Learning (ICML2008)*, pages 56–63, 2008.

[20] S. Bickel, M. Brückner, and T. Scheffer. Discriminative learning for differing training and test distributions. In *Proceedings of the 24th International Conference on Machine Learning (ICML2007)*, pages 81–88, 2007.

[21] S. Bickel, C. Sawade, and T. Scheffer. Transfer learning by distribution matching for targeted advertising. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 145–152, 2009.

[22] S. Bickel and T. Scheffer. Dirichlet-enhanced spam filtering based on biased samples. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 161–168, Cambridge, MA, 2007. MIT Press.

[23] C. M. Bishop. *Pattern Recognition and Machine Learning.* Springer, New York, NY, USA, 2006.

[24] C. M. Bishop. *Pattern Recognition and Machine Learning.* Springer, New York, NY, USA, 2006.

[25] D. M. Blei and M. I. Jordan. Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–144, 2006.

[26] L. Bo and C. Sminchisescu. Twin Gaussian processes for structured prediction. *International Journal of Computer Vision*, 87(1-2):28–52, 2010.

[27] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, Cambridge, UK, 2004.

[28] M. Á. Carreira-Perpiñán. Fast nonparametric clustering with Gaussian blurring mean-shift. In W. Cohen and A. Moore, editors, *Proceedings of 23rd International Conference on Machine Learning (ICML2006)*, pages 153–160, Pittsburgh, PA, Jun. 25–29 2006.

[29] M. Á. Carreira-Perpiñán. Gaussian mean shift is an EM algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:767–776, 2007.

[30] R. Caruana. Multitask learning. *Machine Learning*, 28:41–75, 1997.

[31] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning.* MIT Press, Cambridge, MA, USA, 2006.

[32] S.-M. Chen, Y.-S. Hsu, and J.-T. Liaw. On kernel estimators of density ratio. *Statistics*, 43(5):463–479, 2009.

[33] K. F. Cheng and C. K. Chu. Semiparametric density estimation under a two-sample density ratio model. *Bernoulli*, 10(4):583–604, 2004.

[34] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:790–799, 1995.

[35] F. R. K. Chung. *Spectral Graph Theory.* American Mathematical Society, Providence, RI, USA, 1997.

[36] C. Cortes, Y. Mansour, and M. Mohri. Learning bounds for importance weighting. In J. Lafferty, C. K. I. Williams, R. Zemel, J. Shawe-Taylor, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 442–450, 2010.

[37] T. Cour, N. Gogin, and J. Shi. Learning spectral graph segmentation. In R. G. Cowell and Z. Ghahramani, editors, *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, pages 65–72. Society for Artificial Intelligence and Statistics, 2005.

[38] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* John Wiley & Sons, Inc., Hoboken, NJ, USA, 2nd edition, 2006.

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

[40] J. Ćwik and J. Mielniczuk. Estimating density ratio with application to discriminant analysis. *Communications in Statistics: Theory and Methods*, 18(8):3057–3069, 1989.

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

[42] F. Desobry, M. Davy, and C. Doncarli. An online kernel change detection algorithm. *IEEE Transactions on Signal Processing*, 53(8):2961–2974, 2005.

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

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

[45] M. C. du Plessis and M. Sugiyama. Semi-supervised learning of class balance under class-prior change by distribution matching. In J. Langford and J. Pineau, editors, *Proceedings of 29th International Conference on Machine Learning (ICML2012)*, pages 823–830, Edinburgh, Scotland, Jun. 26–Jul. 1 2012.

[46] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley, New York, NY, USA, second edition, 2001.

[47] N. Duffy and M. Collins. Convolution kernels for natural language. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 625–632, Cambridge, MA, USA, 2002. MIT Press.

[48] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004.

[49] S. Eguchi and J. Copas. Interpreting Kullback-Leibler divergence with the Neyman-Pearson lemma. *Journal of Multivariate Analysis*, 97(9):2034–2040, 2006.

[50] L. Faivishevsky and J. Goldberger. A nonparametric information theoretic clustering algorithm. In A. T. Joachims and J. Fürnkranz, editors, *Proceedings of 27th International Conference on Machine Learning (ICML2010)*, pages 351–358, Haifa, Israel, Jun. 21–25 2010.

[51] T. S. Ferguson. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2):209–230, 1973.

[52] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

[53] K. Fukunaga and L. D. Hostetler. The estimation of the gradient of a density function, with application in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40, 1975.

[54] T. Gärtner. A survey of kernels for structured data. *SIGKDD Explorations*, 5(1):S268–S275, 2003.

[55] T. Gärtner, P. Flach, and S. Wrobel. On graph kernels: Hardness results and efficient alternatives. In B. Schölkopf and M. Warmuth, editors, *Proceedings of the Sixteenth Annual Conference on Computational Learning Theory*, pages 129–143, 2003.

[56] A. Gelman. Method of moments using Monte Carlo simulation. *Journal of Computational and Graphical Statistics*, 4(1):36–54, 1995.

[57] Z. Ghahramani and M. J. Beal. Variational inference for Bayesian mixtures of factor analysers. In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 449–455. MIT Press, 2000.

[58] M. Girolami. Mercer kernel-based clustering in feature space. *IEEE Transactions on Neural Networks*, 13(3):780–784, 2002.

[59] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, USA, second edition, 1989.

[60] R. Gomes, A. Krause, and P. Perona. Discriminative clustering by regularized information maximization. In J. Lafferty, C. K. I. Williams, R. Zemel, J. Shawe-Taylor, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 766–774, 2010.

[61] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf. Measuring statistical dependence with Hilbert-Schmidt norms. In S. Jain, H. U. Simon, and E. Tomita, editors, *Algorithmic Learning Theory*, Lecture Notes in Artificial Intelligence, pages 63–77, Berlin, Germany, 2005. Springer-Verlag.

[62] A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf. Covariate shift by kernel mean matching. In J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. Lawrence, editors, *Dataset Shift in Machine Learning*, pages 131–160, Cambridge, MA, USA, 2009. MIT Press.

[63] M. U. Gutmann and A. Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13:307–361, 2012.

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

[65] H. Hachiya, J. Peters, and M. Sugiyama. Reward weighted regression with sample reuse. *Neural Computation*, 23(11):2798–2832, 2011.

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

[67] P. Hall. On the non-parametric estimation of mixture proportions. *Journal of the Royal Statistical Society, Series B*, 43(2):147–156, 1981.

[68] P. Hall and M. P. Wand. On nonparametric discrimination using density differences. *Biometrika*, 75(3):541–547, 1988.

[69] W. Härdle, M. Müller, S. Sperlich, and A. Werwatz. *Nonparametric and Semiparametric Models.* Springer, Berlin, Germany, 2004.

[70] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, New York, NY, USA, 2001.

[71] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, New York, NY, USA, 2001.

[72] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[73] R. A. Horn and C. A. Johnson. *Matrix Analysis.* Cambridge University Press, Cambridge, UK, 1985.

[74] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.

[75] A. Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6:695–709, 2005.

[76] T. Ide and K. Tsuda. Change-point detection using Krylov subspace learning. In *Proceedings of the SIAM International Conference on Data Mining*, pages 515–520, 2007.

[77] W. Jitkrittum, H. Hachiya, and M. Sugiyama. Feature selection via ℓ_1-penalized squared-loss mutual information. *IEICE Transactions on Information and Systems*, E96-D(7):1513–1524, 2013.

[78] T. Joachims. *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms.* Kluwer Academic Publishers, Boston, MA, USA, 2002.

[79] I. T. Jolliffe. *Principal Component Analysis.* Springer-Verlag, New York, NY, USA, 1986.

[80] A. Kain and M. W. Macon. Spectral voice conversion for text-to-speech synthesis. In *Proceedings of 1998 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP1998)*, pages 285–288, Washington, DC, U.S.A, May. 12–15 1988.

[81] T. Kanamori. Pool-based active learning with optimal sampling distribution and its information geometrical interpretation. *Neurocomputing*, 71(1–3):353–362, 2007.

[82] T. Kanamori, S. Hido, and M. Sugiyama. A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, 10(Jul.):1391–1445, 2009.

[83] T. Kanamori, S. Hido, and M. Sugiyama. A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, 10:1391–1445, 2009.

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

[85] T. Kanamori, T. Suzuki, and M. Sugiyama. Theoretical analysis of density ratio estimation. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E93-A(4):787–798, 2010.

[86] T. Kanamori, T. Suzuki, and M. Sugiyama. f-divergence estimation and two-sample homogeneity test under semiparametric density-ratio models. *IEEE Transactions on Information Theory*, 58(2):708–720, 2012.

[87] T. Kanamori, T. Suzuki, and M. Sugiyama. Statistical analysis of kernel-based least-squares density-ratio estimation. *Machine Learning*, 86(3):335–367, 2012.

[88] T. Kanamori, T. Suzuki, and M. Sugiyama. Computational complexity of kernel-based density-ratio estimation: A condition number analysis. *Machine Learning*, 90(3):431–460, 2013.

[89] M. Karasuyama and Sugiyama. Canonical dependency analysis based on squared-loss mutual information. *Neural Networks*, 34:46–55, 2012.

[90] H. Kashima and T. Koyanagi. Kernels for semi-structured data. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 291–298, San Francisco, CA, USA, 2002. Morgan Kaufmann.

[91] H. Kashima, K. Tsuda, and A. Inokuchi. Marginalized kernels between labeled graphs. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 321–328, San Francisco, CA, USA, 2003. Morgan Kaufmann.

[92] Y. Kawahara and M. Sugiyama. Sequential change-point detection based on direct density-ratio estimation. *Statistical Analysis and Data Mining*, 5(2):114–127, 2012.

[93] Y. Kawahara and M. Sugiyama. Sequential change-point detection based on direct density-ratio estimation. *Statistical Analysis and Data Mining*, 5(2):114–127, 2012.

[94] Y. Kawahara, T. Yairi, and K. Machida. Change-point detection in time-series data based on subspace identification. In *Proceedings of the 7th IEEE International Conference on Data Mining*, pages 559–564, 2007.

[95] A. Keziou. Dual representation of ϕ-divergences and applications. *Comptes Rendus Mathématique*, 336(10):857–862, 2003.

[96] J. Kim and C. Scott. L_2 kernel classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(10):1822–1831, 2010.

[97] M. Kimura and M. Sugiyama. Dependence-maximization clustering with least-squares mutual information. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 15(7):800–805, 2011.

[98] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques.* MIT Press, 2009.

[99] V. Koltchinskii. Asymptotics of spectral projections of some random matrices approximating integral operators. In *Progress in Probabilty*, volume 43, pages 191–227, 1998.

[100] V. Koltchinskii and E. Giné. Random matrix approximation of spectra of integral operators. *Bernoulli*, 6(1):113–167, 2000.

[101] R. I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 315–322, 2002.

[102] L. F. Kozachenko and N. N. Leonenko. Sample estimate of entropy of a random vector. *Problems of Information Transmission*, 23(9):95–101, 1987.

[103] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22:79–86, 1951.

[104] K. Kurihara and M. Welling. Bayesian k-means as a "maximization-expectation" algorithm. *Neural Computation*, 21(4):1145–1172, 2009.

[105] S.-I. Lee, V. Ganapathi, and D. Koller. Efficient structure learning of Markov networks using l_1-regularization. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 817–824, Cambridge, MA, 2007. MIT Press.

[106] Y. K. Lee and H. T. Ng. An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, pages 41–48, 2002.

[107] Y. Li, H. Kambara, Y. Koike, and M. Sugiyama. Application of covariate shift adaptation techniques in brain computer interfaces. *IEEE Transactions on Biomedical Engineering*, 57(6):1318–1324, 2010.

[108] Y. F. Li, I. W. Tsang, J. T. Kwok, and Z.-H. Zhou. Tighter and convex maximum margin clustering. In D. van Dyk and M. Welling, editors, *Proceedings of Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS2009)*, volume 5 of *JMLR Workshop and Conference Proceedings*, pages 344–351, Clearwater Beach, FL, USA, Apr. 16–18 2009.

[109] D. Lin, E. Grimson, and J. Fisher. Construction of dependent Dirichlet processes based on Poisson processes. In J. Lafferty, C. K. I. Williams, R. Zemel, J. Shawe-Taylor, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1387–1395, 2010.

[110] H. Liu, F. Han, M. Yuan, J. Lafferty, and L. Wasserman. The nonparanormal skeptic. In *Proceedings of the 29th International Conference on Machine Learning (ICML2012)*, 2012.

[111] H. Liu, J. Lafferty, and L. Wasserman. The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *Journal of Machine Learning Research*, 10:2295–2328, 2009.

[112] S. Liu, M. Yamada, N. Collier, and M. Sugiyama. Change-point detection in time-series data by relative density-ratio estimation. *Neural Networks*, 43:72–83, 2013.

[113] S. Liu, M. Yamada, N. Collier, and M. Sugiyama. Change-point detection in time-series data by relative density-ratio estimation. *Neural Networks*, 43:72–83, 2013.

[114] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H. Y. Shum. Learning to detect a salient object. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(2):353–367, 2011.

[115] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444, 2002.

[116] D. J. C. MacKay. *Information Theory, Inference, and Learning Algorithms.* Cambridge University Press, Cambridge, UK, 2003.

[117] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics*

and Probability, volume 1, pages 281–297, Berkeley, CA, USA, 1967. University of California Press.

[118] M. Meila and J. Shi. Learning segmentation by random walks. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 873–879, Cambridge, MA, USA, 2001. MIT Press.

[119] N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3):1436–1462, 2006.

[120] V. Moskvina and A. Zhigljavsky. Change-point detection algorithm based on the singular-spectrum analysis. *Communications in Statistics: Simulation and Computation*, 32:319–352, 2003.

[121] R. M. Neal. Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, 2000.

[122] R. M Neal. Slice sampling. *The Annals of Statistics*, 31(3):705–741, 2003.

[123] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 849–856, Cambridge, MA, USA, 2002. MIT Press.

[124] X. Nguyen, M. J. Wainwright, and M. I. Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010.

[125] X. Nguyen, M. J. Wainwright, and M. I. Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010.

[126] G. Niu, B. Dai, L. Shang, and M. Sugiyama. Maximum volume clustering: A new discriminative clustering approach. *Journal of Machine Learning Research*, 14(Sep.):2641–2687, 2013.

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

[128] K. Pearson. 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(302):157–175, 1900.

[129] K. Pearson. 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*, 50:157–175, 1900.

[130] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572, 1901.

[131] J. Qin. Inferences for case-control and semiparametric two-sample density ratio models. *Biometrika*, 85(3):619–630, 1998.

[132] Q. Que and M. Belkin. Inverse density as an inverse problem: The fredholm equation approach. Technical Report 1304.5575, arXiv, 2013.

[133] J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. Lawrence, editors. *Dataset Shift in Machine Learning.* MIT Press, Cambridge, Massachusetts, USA, 2009.

[134] R. Raina, A. Battle, H. Lee, B. Packer, and A. Ng. Self-taught learning: Transfer learning from unlabeled data. In Zoubin Ghahramani, editor, *Proceedings of the 24th Annual International Conference on Machine Learning (ICML2007)*, pages 759–766. Omnipress, 2007.

[135] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[136] P. Ravikumar, M. J. Wainwright, and J. D. Lafferty. High-dimensional Ising model selection using ℓ_1-regularized logistic regression. *The Annals of Statistics*, 38(3):1287–1319, 2010.

[137] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy. Learning from crowds. *Journal of Machine Learning Research*, 11:1297–1322, 2010.

[138] R. Rifkin, G. Yeo, and T. Poggio. Regularized least-squares classification. In J. A. K. Suykens, G. Horvath, S. Basu, C. Micchelli, and J. Vandewalle, editors, *Advances in Learning Theory: Methods, Models and Applications*, volume 190 of *NATO Science Series III: Computer & Systems Sciences*, pages 131–154. IOS Press, Amsterdam, the Netherlands, 2003.

[139] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods.* Springer-Verlag, Secaucus, NJ, USA, 2005.

[140] R. T. Rockafellar. *Convex Analysis.* Princeton University Press, Princeton, NJ, USA, 1970.

[141] A. Rodríguez, D. B. Dunson, and A. E Gelfand. The Nested dirichlet process. *Journal of the American Statistical Association*, 103(483):1131–1154, 2008.

[142] M. Saerens, P. Latinne, and C. Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure. *Neural Computation*, 14(1):21–41, 2002.

[143] J. Sainui and M. Sugiyama. Direct approximation of quadratic mutual information and its application to dependence-maximization clustering. *IEICE Transactions on Information and Systems*, E96-D(10):2282–2285, 2013.

[144] M. W. Schmidt and K. P. Murphy. Convex structure learning in log-linear models: Beyond pairwise potentials. *Journal of Machine Learning Research - Proceedings Track*, 9:709–716, 2010.

[145] B. Schölkopf and A. J. Smola. *Learning with Kernels.* MIT Press, Cambridge, MA, USA, 2002.

[146] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *Proceedings of International Conference on Computer Vision (ICCV2003)*, volume 2, pages 750–757, 2003.

[147] C. Shannon. A mathematical theory of communication. *Bell Systems Technical Journal*, 27:379–423, 1948.

[148] N. Shental, A. Zomet, T. Hertz, and Y. Weiss. Learning and inferring image segmentations using the GBP typical cut algorithm. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1243–1250, 2003.

[149] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

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

[151] L. Sigal and M. J. Black. Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. Technical Report CS-06-08, Brown University, 2006.

[152] B. W. Silverman. *Density Estimation for Statistics and Data Analysis.* Chapman and Hall, London, UK, 1986.

[153] L. Song, A. Smola, A. Gretton, and K. Borgwardt. A dependence maximization view of clustering. In Z. Ghahramani, editor, *Proceedings of the 24th Annual International Conference on Machine Learning (ICML2007)*, pages 815–822, 2007.

[154] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society, Series B*, 36:111–147, 1974.

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

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

[157] M. Sugiyama. Machine learning with squared-loss mutual information. *Entropy*, 15(1):80–112, 2013.

[158] M. Sugiyama and M. Kawanabe. *Machine Learning in Non-Stationary Environments: Introduction to Covariate Shift Adaptation.* MIT Press, Cambridge, Massachusetts, USA, 2012.

[159] M. Sugiyama, M. Kawanabe, and P. L. Chui. Dimensionality reduction for density ratio estimation in high-dimensional spaces. *Neural Networks*, 23(1):44–59, 2010.

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

[161] M. Sugiyama, S. Liu, M. C. du Plessis, M. Yamanaka, M. Yamada, T. Suzuki, and T. Kanamori. Direct divergence approximation between probability distributions and its applications in machine learning. *Journal of Computing Science and Engineering*, 7(2):99–111, 2013.

[162] M. Sugiyama and K.-R. Müller. Input-dependent estimation of generalization error under covariate shift. *Statistics & Decisions*, 23(4):249–279, 2005.

[163] M. Sugiyama and S. Nakajima. Pool-based active learning in approximate linear regression. *Machine Learning*, 75(3):249–274, 2009.

[164] M. Sugiyama and H. Ogawa. Subspace information criterion for model selection. *Neural Computation*, 13(8):1863–1889, 2001.

[165] M. Sugiyama and N. Rubens. A batch ensemble approach to active learning with model selection. *Neural Networks*, 21(9):1278–1286, 2008.

[166] M. Sugiyama and T. Suzuki. Least-squares independence test. *IEICE Transactions on Information and Systems*, E94-D(6):1333–1336, 2011.

[167] M. Sugiyama, T. Suzuki, Y. Itoh, T. Kanamori, and M. Kimura. Least-squares two-sample test. *Neural Networks*, 24(7):735–751, 2011.

[168] M. Sugiyama, T. Suzuki, and T. Kanamori. *Density Ratio Estimation in Machine Learning.* Cambridge University Press, Cambridge, UK, 2012.

[169] M. Sugiyama, T. Suzuki, and T. Kanamori. *Density Ratio Estimation in Machine Learning.* Cambridge University Press, Cambridge, UK, 2012.

[170] M. Sugiyama, T. Suzuki, and T. Kanamori. Density ratio matching under the Bregman divergence: A unified framework of density ratio estimation. *Annals of the Institute of Statistical Mathematics*, 64(5):1009–1044, 2012.

[171] M. Sugiyama, T. Suzuki, and T. Kanamori. Density-ratio matching under the Bregman divergence: a unified framework of density-ratio estimation. *Annals of the Institute of Statistical Mathematics*, 64(5):1009–1044, 2012.

[172] M. Sugiyama, T. Suzuki, T. Kanamori, M. C. du Plessis, S. Liu, and I. Takeuchi. Density-difference estimation. *Neural Computation*, 25(10):2734–2775, 2013.

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

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

[175] M. Sugiyama, M. Yamada, M. Kimura, and H. Hachiya. On information-maximization clustering: Tuning parameter selection and analytic solution. In L. Getoor and T. Scheffer, editors, *Proceedings of 28th International Conference on Machine Learning (ICML2011)*, pages 65–72, Bellevue, Washington, USA, Jun. 28–Jul. 2 2011.

[176] M. Sugiyama, M. Yamada, P. von Bünau, T. Suzuki, T. Kanamori, and M. Kawanabe. Direct density-ratio estimation with dimensionality reduction via least-squares hetero-distributional subspace search. *Neural Networks*, 24(2):183–198, 2011.

[177] T. Suzuki and M. Sugiyama. Least-squares independent component analysis. *Neural Computation*, 23(1):284–301, 2011.

[178] T. Suzuki and M. Sugiyama. Sufficient dimension reduction via squared-loss mutual information estimation. *Neural Computation*, 25(3):725–758, 2013.

[179] T. Suzuki, M. Sugiyama, T. Kanamori, and J. Sese. Mutual information estimation reveals global associations between stimuli and biological processes. *BMC Bioinformatics*, 10(1):S52 (12 pages), 2009.

[180] T. Suzuki, M. Sugiyama, J. Sese, and T. Kanamori. Approximating mutual information by maximum likelihood density ratio estimation. In Y. Saeys, H. Liu, I. Inza, L. Wehenkel, and Y. Van de Peer, editors, *Proceedings of ECML-PKDD2008 Workshop on New Challenges for Feature Selection in Data Mining and Knowledge*

Discovery (FSDM2008), volume 4 of *JMLR Workshop and Conference Proceedings*, pages 5–20, Antwerp, Belgium, Sep. 15 2008.

[181] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2007.

[182] R. Tibshirani. Regression shrinkage and subset selection with the lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267–288, 1996.

[183] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.

[184] D. M. Titterington. Minimum distance non-parametric estimation of mixture proportions. *Journal of the Royal Statistical Society, Series B*, 45(1):37–46, 1983.

[185] R. Tomioka, T. Suzuki, and M. Sugiyama. Super-linear convergence of dual augmented Lagrangian algorithm for sparsity regularized estimation. *Journal of Machine Learning Research*, 12(May):1537–1586, 2011.

[186] K. Torkkola. Feature extraction by non-parametric mutual information maximization. *Journal of Machine Learning Research*, 3:1415–1438, 2003.

[187] Y. Tsuboi, H. Kashima, S. Hido, S. Bickel, and M. Sugiyama. Direct density ratio estimation for large-scale covariate shift adaptation. *Journal of Information Processing*, 17:138–155, 2009.

[188] Y. Tsuboi, H. Kashima, S. Hido, S. Bickel, and M. Sugiyama. Direct density ratio estimation for large-scale covariate shift adaptation. *Journal of Information Processing*, 17:138–155, 2009.

[189] N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton. SMEM algorithm for mixture models. *Neural Computation*, 12(9):2109–2128, 2000.

[190] K. Ueki, M. Sugiyama, and Y. Ihara. Lighting condition adaptation for perceived age estimation. *IEICE Transactions on Information and Systems*, E94-D(2):392–395, 2011.

[191] H. Valizadegan and R. Jin. Generalized maximum margin clustering and unsupervised kernel learning. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 1417–1424, Cambridge, MA, USA, 2007. MIT Press.

[192] T. Van den Bulcke, K. Van Leemput, B. Naudts, P. van Remortel, H. Ma, A. Verschoren, B. De Moor, and K. Marchal. SynTReN: A generator of synthetic gene expression data for design and analysis of structure learning algorithms. *BMC Bioinformatics*, 7(1):43, 2006.

[193] V. N. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag, Berlin, Germany, 1995.

[194] V. N. Vapnik. *Statistical Learning Theory.* Wiley, New York, NY, USA, 1998.

[195] V. N. Vapnik. *Statistical Learning Theory.* Wiley, New York, NY, USA, 1998.

[196] V. N. Vapnik, I. Braga, and R. Izmailov. Constructive setting of the density ratio estimation problem and its rigorous solution. Technical Report 1306.0407, arXiv, 2013.

[197] U. von Luxburg. *Statistical Learning with Similarity and Dissimilarity Functions.* PhD thesis, Technical University of Berlin, Berlin, Germany, 2004.

[198] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.

[199] F. Wang, B. Zhao, and C. Zhang. Linear time maximum margin clustering. *IEEE Transactions on Neural Networks*, 21(2):319–332, 2010.

[200] Q. Wang, S. R. Kulkarni, and S. Verdú. Divergence estimation for multidimensional densities via k-nearest-neighbor distances. *IEEE Transactions on Information Theory*, 55(5):2392–2405, 2009.

[201] L. Wasserman. *All of Statistics: A Concise Course in Statistical Inference.* Springer Publishing Company, Incorporated, 2010.

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

[203] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans. Maximum margin clustering. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1537–1544, Cambridge, MA, USA, 2005. MIT Press.

[204] M. Yamada, G. Niu, J. Takagi, and M. Sugiyama. Computationally efficient sufficient dimension reduction via squared-loss mutual information. In C.-N. Hsu and W. S. Lee, editors, *Proceedings of the Third Asian Conference on Machine Learning (ACML2011)*, volume 20 of *JMLR Workshop and Conference Proceedings*, pages 247–262, Taoyuan, Taiwan, Nov. 13-15 2011.

[205] M. Yamada, L. Sigal, and M. Raptis. No bias left behind: Covariate shift adaptation for discriminative 3D pose estimation. In *Proceedings of European Conference on Computer Vision (ECCV2012)*, pages 674–687, 2012.

[206] M. Yamada and M. Sugiyama. Direct importance estimation with Gaussian mixture models. *IEICE Transactions on Information and Systems*, E92-D(10):2159–2162, 2009.

[207] M. Yamada and M. Sugiyama. Dependence minimizing regression with model selection for non-linear causal inference under non-Gaussian noise. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI2010)*, pages 643–648, Atlanta, Georgia, USA, Jul. 11–15 2010. The AAAI Press.

[208] M. Yamada and M. Sugiyama. Cross-domain object matching with model selection. In G. Gordon, D. Dunson, and M. Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS2011)*, volume 15 of *JMLR Workshop and Conference Proceedings*, pages 807–815, Fort Lauderdale, Florida, USA, Apr. 11-13 2011.

[209] M. Yamada and M. Sugiyama. Direct density-ratio estimation with dimensionality reduction via hetero-distributional subspace analysis. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI2011)*, pages 549–554, San Francisco, California, USA, Aug. 7–11 2011. The AAAI Press.

[210] M. Yamada, M. Sugiyama, and T. Matsui. Semi-supervised speaker identification under covariate shift. *Signal Processing*, 90(8):2353–2361, 2010.

[211] M. Yamada, M. Sugiyama, G. Wichern, and J. Simm. Direct importance estimation with a mixture of probabilistic principal component analyzers. *IEICE Transactions on Information and Systems*, E93-D(10):2846–2849, 2010.

[212] M. Yamada, T. Suzuki, T. Kanamori, H. Hachiya, and M. Sugiyama. Relative density-ratio estimation for robust distribution comparison. *Neural Computation*, 25(5):1324–1370, 2013.

[213] M. Yamanaka, M. Matsugu, and M. Sugiyama. Detection of activities and events without explicit categorization. *IPSJ Transactions on Mathematical Modeling and Its Applications*, 6(2):86–92, 2013.

[214] M. Yamanaka, M. Matsugu, and M. Sugiyama. Salient object detection based on direct density-ratio estimation. *IPSJ Transactions on Mathematical Modeling and Its Applications*, 6(2):78–85, 2013.

[215] W.-Y. Yang, J. T. Kwok, and B.-L. Lu. Spectral and semidefinite relaxation of the CLUHSIC algorithm. In *Proceedings of the 2010 SIAM International Conference on Data Mining*, pages 106–117, 2010.

[216] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1601–1608, Cambridge, MA, USA, 2005. MIT Press.

[217] H. Zha, X. He, C. Ding, M. Gu, and H. Simon. Spectral relaxation for k-means clustering. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 1057–1064, Cambridge, MA, USA, 2002. MIT Press.

[218] B. Zhang and Y.J. Wang. Learning structural changes of Gaussian graphical models in controlled experiments. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence (UAI2010)*, pages 701–708, 2010.

[219] K. Zhang, I. W. Tsang, and J. T. Kwok. Maximum margin clustering made practical. *IEEE Transactions on Neural Networks*, 20(4):583–596, 2009.

[220] B. Zhao, F. Wang, and C Zhang. Maximum margin clustering via cutting plane algorithm. In *Proceedings of the 2007 SIAM International Conference on Data Mining*, pages 751–762, 2008.

[221] T. Zhao, H. Hachiya, V. Tangkaratt, J. Morimoto, and M. Sugiyama. Efficient sample reuse in policy gradients with parameter-based exploration. *Neural Computation*, 25(6):1512–1547, 2013.

[222] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67(2):301–320, 2005.