Entropy **2013**, *15*, 80-112; doi:10.3390/e15010080

OPEN ACCESS

entropy

ISSN 1099-4300

www.mdpi.com/journal/entropy

Review

Machine Learning with Squared-Loss Mutual Information

Masashi Sugiyama

Department of Computer Science, Tokyo Institute of Technology 2-12-1 O-okayama, Meguro-ku, Tokyo 152-8552, Japan; E-Mail: sugi@cs.titech.ac.jp

Received: 29 October 2012; in revised form: 7 December 2012 / Accepted: 21 December 2012 / Published: 27 December 2012

Abstract: Mutual information (MI) is useful for detecting statistical independence between random variables, and it has been successfully applied to solving various machine learning problems. Recently, an alternative to MI called *squared-loss MI* (SMI) was introduced. While ordinary MI is the Kullback–Leibler divergence from the joint distribution to the product of the marginal distributions, SMI is its Pearson divergence variant. Because both the divergences belong to the f-divergence family, they share similar theoretical properties. However, a notable advantage of SMI is that it can be approximated from data in a computationally more efficient and numerically more stable way than ordinary MI. In this article, we review recent development in SMI approximation based on direct density-ratio estimation and SMI-based machine learning techniques such as independence testing, dimensionality reduction, canonical dependency analysis, independent component analysis, object matching, clustering, and causal inference.

Keywords: squared-loss mutual information; Pearson divergence; density-ratio estimation; independence testing; dimensionality reduction; independent component analysis; object matching; clustering; causal inference; machine learning

1. Introduction

Mutual information (MI) [1,2] for random variables $\boldsymbol{X}$ and $\boldsymbol{Y}$ is defined as:

$$\mathrm{MI}(\boldsymbol{X},\boldsymbol{Y}) := \iint p(\boldsymbol{x},\boldsymbol{y}) \log \frac{p(\boldsymbol{x},\boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})} \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y}$$

where $p(\boldsymbol{x},\boldsymbol{y})$ is the joint probability density of $\boldsymbol{X}$ and $\boldsymbol{Y}$, and $p(\boldsymbol{x})$ and $p(\boldsymbol{y})$ are the marginal probability densities of $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively (Precisely, $p(\boldsymbol{x},\boldsymbol{y})$, $p(\boldsymbol{x})$, and $p(\boldsymbol{y})$ are different functions and thus should be denoted, e.g., by $p_{\mathbf{X},\mathbf{Y}}(\boldsymbol{x},\boldsymbol{y})$, $p_{\mathbf{X}}(\boldsymbol{x})$, and $p_{\mathbf{Y}}(\boldsymbol{y})$, respectively. However, we use the simplified notations for the sake of brevity). Statistically, MI can be regarded as the Kullback–Leibler divergence [3] from the joint density $p(\boldsymbol{x},\boldsymbol{y})$ to the product of the marginals $p(\boldsymbol{x})p(\boldsymbol{y})$, and thus can be regarded as a measure of statistical dependency between $\boldsymbol{X}$ and $\boldsymbol{Y}$. Estimation of MI from data samples has been one of the major challenges in information science and various approaches have been developed thus far.

The most naive approach to approximating MI from data would be to use a non-parametric density estimator such as kernel density estimation (KDE) [4], *i.e.*, the densities $p(\boldsymbol{x},\boldsymbol{y})$, $p(\boldsymbol{x})$, and $p(\boldsymbol{y})$ included in MI are separately estimated from samples, and the estimated densities are used for approximating MI. However, density estimation is known to be a hard problem [5] and division by estimated densities tends to magnify the estimation error. Therefore, the KDE-based MI approximator may not be reliable in practice.

Another approach uses histogram-based density estimators with data-dependent partitioning. In the context of estimating the Kullback–Leibler divergence [3], histogram-based methods have been studied thoroughly and their consistency has been established [6–8]. However, the rate of convergence has not been elucidated yet, and such histogram-based methods are strongly influenced by the curse of dimensionality. Thus, these methods may not be reliable in high-dimensional problems.

MI can be expressed in terms of the entropies as:

$$\mathrm{MI}(\boldsymbol{X},\boldsymbol{Y}) = H(\boldsymbol{X}) + H(\boldsymbol{Y}) - H(\boldsymbol{X},\boldsymbol{Y})$$

where $H(\boldsymbol{X})$ denotes the entropy of $\boldsymbol{X}$:

$$H(\boldsymbol{X}) := -\int p(\boldsymbol{x}) \log p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}$$

Based on this expression, the nearest neighbor distance has been used for approximating MI [9]. Such a nearest neighbor approach was shown to perform better than the naive KDE-based approach [10], given that the number k of nearest neighbors is chosen appropriately—a small (large) k yields an estimator with small (large) bias and large (small) variance. However, appropriately determining the value of k so that the bias-variance trade-off is optimally controlled is not straightforward in the context of MI estimation. A similar nearest-neighbor idea has been applied to Kullback–Leibler divergence estimation [11], whose consistency has been proved for finite k—this is an interesting result since Kullback–Leibler divergence estimation is consistent even when density estimation is not consistent. However, the rate of convergence seems to be still an open research issue.

Approximation of the entropies based on the Edgeworth expansion has also been explored in the context of MI estimation [12]. Such a method works well when the target density is close to Gaussian. However, if the target density is far from Gaussian, the Edgeworth expansion method is no longer reliable.

More recently, an MI approximator via direct estimation of the density ratio $\frac{p(\boldsymbol{x},\boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})}$ has been developed [13], which is based on a Kullback–Leibler divergence approximator via direct density-ratio estimation [14–16]. The MI approximator is given as the solution of a convex optimization problem, which tends to be sparse [14]. A notable advantage of this density-ratio method is that it does not involve separate estimation of densities $p(\boldsymbol{x},\boldsymbol{y})$, $p(\boldsymbol{x})$, and $p(\boldsymbol{y})$, and it was proved to achieve the optimal non-parametric convergence rate. However, due to the "log" operation included in MI, this MI approximator is computationally rather expensive and susceptible to outliers [17,18].

To cope with these problems, a variant of MI called the *squared-loss mutual information* (SMI) [19] has been explored recently. SMI for $\boldsymbol{X}$ and $\boldsymbol{Y}$ is defined as:

$$\mathrm{SMI}(\boldsymbol{X},\boldsymbol{Y}) := \frac{1}{2}\iint p(\boldsymbol{x})p(\boldsymbol{y})\left(\frac{p(\boldsymbol{x},\boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})}-1\right)^2 \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y}$$

SMI is the Pearson divergence [20] from the joint density $p(\boldsymbol{x},\boldsymbol{y})$ to the product of the marginals $p(\boldsymbol{x})p(\boldsymbol{y})$. It is always non-negative and it vanishes if and only if $\boldsymbol{X}$ and $\boldsymbol{Y}$ are statistically independent. Note that both the Pearson divergence and the Kullback–Leibler divergence belong to the class of Ali–Silvey–Csiszár divergences (which is also known as f-divergences) [21,22], meaning that they share similar properties.

In a similar way to ordinary MI, SMI can be approximated accurately via direct estimation of the density ratio $\frac{p(\boldsymbol{x},\boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})}$ [19], which is based on a Pearson divergence approximator via direct density-ratio estimation [16,23]. This SMI approximator has various desirable properties: For example, it was proved to achieve the optimal non-parametric convergence rate [24], its solution can be obtained *analytically* just by solving a system of linear equations, it has superior numerical properties [25], and it is robust against outliers [17,18].

In particular, the property of the SMI approximator that an analytic solution is available is highly useful in machine learning, because this allows explicit computation of the *derivative* of the SMI approximator with respect to another parameter. For example, in supervised dimensionality reduction, linear transformation $\boldsymbol{U}$ for input $\boldsymbol{x}$ is optimized so that the transformed input $\boldsymbol{U}\boldsymbol{x}$ has the highest dependency on output $\boldsymbol{y}$. In this context, the derivative of the SMI estimator between $\boldsymbol{U}\boldsymbol{x}$ and $\boldsymbol{y}$ with respect to transformation $\boldsymbol{U}$ can be exploited for optimizing transformation $\boldsymbol{U}$. On the other hand, such derivative computation is not straightforward for the MI estimator whose solution is obtained via numerical optimization.

The purpose of this article is to review recent development in SMI approximation based on direct density-ratio estimation and SMI-based machine learning techniques. The remainder of this paper is structured as follows. After reviewing the SMI approximator based on direct density-ratio estimation in Section 2, we illustrate in Section 3 how the SMI approximator can be utilized for solving various machine learning tasks such as: independence testing [26], feature selection [19,27], feature extraction [28,29], canonical dependency analysis [30], independent component analysis [31], object matching [32], clustering [33,34], and causality learning [35].

2. Definition and Estimation of SMI

In this section, we review the definition of SMI and its approximator based on direct density-ratio estimation.

2.1. Definition of SMI

Let us consider two random variables $\boldsymbol{X} \in \mathcal{X}$ and $\boldsymbol{Y} \in \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ are domains of $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively. Let $p(\boldsymbol{x}, \boldsymbol{y})$ be the joint probability density of $\boldsymbol{X}$ and $\boldsymbol{Y}$, and $p(\boldsymbol{x})$ and $p(\boldsymbol{y})$ be the marginal probability densities of $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively. The *squared-loss mutual information* (SMI) [19] for $\boldsymbol{X}$ and $\boldsymbol{Y}$ is defined as:

$$\mathrm{SMI}(\boldsymbol{X}, \boldsymbol{Y}) := \frac{1}{2} \iint p(\boldsymbol{x})p(\boldsymbol{y}) \left(\frac{p(\boldsymbol{x}, \boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})} - 1 \right)^2 \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y} \tag{1}$$

SMI is always non-negative and it takes zero if and only if $\boldsymbol{X}$ and $\boldsymbol{Y}$ are statistically independent. Hence, SMI can be used for detecting statistical independence between $\boldsymbol{X}$ and $\boldsymbol{Y}$.

Below, we consider the problem of estimating SMI from paired samples $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$ drawn independently from the joint distribution with density $p(\boldsymbol{x}, \boldsymbol{y})$.

2.2. Least-Squares Estimation of SMI

Here, we review the basic idea and theoretical properties of the SMI approximator called *least-squares mutual information* (LSMI) [19].

2.2.1. SMI Approximation via Direct Density-Ratio Estimation

The basic idea of LSMI is to directly estimate the following *density-ratio* function without going through density estimation of $p(\boldsymbol{x}, \boldsymbol{y})$, $p(\boldsymbol{x})$, and $p(\boldsymbol{y})$:

$$r(\boldsymbol{x}, \boldsymbol{y}) := \frac{p(\boldsymbol{x}, \boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})} \tag{2}$$

Let $g(\boldsymbol{x}, \boldsymbol{y})$ be a model of the density ratio $r(\boldsymbol{x}, \boldsymbol{y})$. In LSMI, the model is learned so that the following squared-error J is minimized:

$$\begin{aligned} J(g) &:= \frac{1}{2} \iint \left(g(\boldsymbol{x}, \boldsymbol{y}) - r(\boldsymbol{x}, \boldsymbol{y}) \right)^2 p(\boldsymbol{x})p(\boldsymbol{y}) \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y} \\ &= \frac{1}{2} \iint g(\boldsymbol{x}, \boldsymbol{y})^2 p(\boldsymbol{x})p(\boldsymbol{y}) \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y} - \iint g(\boldsymbol{x}, \boldsymbol{y}) p(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y} + C \end{aligned} \tag{3}$$

where C is a constant defined by:

$$C := \frac{1}{2} \iint r(\boldsymbol{x}, \boldsymbol{y}) p(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y}$$

Since J contains the expectations over unknown densities $p(\boldsymbol{x})p(\boldsymbol{y})$ and $p(\boldsymbol{x}, \boldsymbol{y})$, the expectations are approximated by empirical averages. Then the LSMI optimization problem is given as follows:

$$\widehat{g} := \underset{g \in \mathcal{G}}{\mathrm{argmin}} \left[\frac{1}{2n^2} \sum_{i,j=1}^{n} g(\boldsymbol{x}_i, \boldsymbol{y}_j)^2 - \frac{1}{n} \sum_{i=1}^{n} g(\boldsymbol{x}_i, \boldsymbol{y}_i) \right] \tag{4}$$

where $\mathcal{G}$ is a function space from which g is searched.

Finally, the SMI approximator called LSMI is given as:

$$\mathrm{LSMI}(\{(\boldsymbol{x}_i,\boldsymbol{y}_i)\}_{i=1}^n) := \frac{1}{2n}\sum_{i=1}^n \widehat{g}(\boldsymbol{x}_i,\boldsymbol{y}_i) - \frac{1}{2} \tag{5}$$

or

$$\mathrm{LSMI}'(\{(\boldsymbol{x}_i,\boldsymbol{y}_i)\}_{i=1}^n) := -\frac{1}{2n^2}\sum_{i,j=1}^n \widehat{g}(\boldsymbol{x}_i,\boldsymbol{y}_j)^2 + \frac{1}{n}\sum_{i=1}^n \widehat{g}(\boldsymbol{x}_i,\boldsymbol{y}_i) - \frac{1}{2} \tag{6}$$

Equation (5) would be the simplest SMI approximator, while Equation (6) is suitable for theoretical analysis because this corresponds to the negative of the objective function (4) up to the constant $1/2$. These estimators are derived based on the following equivalent expressions of SMI:

$$\mathrm{SMI}(\boldsymbol{X},\boldsymbol{Y}) = \frac{1}{2}\iint r(\boldsymbol{x},\boldsymbol{y})p(\boldsymbol{x},\boldsymbol{y})\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y} - \frac{1}{2} \tag{7}$$

$$= -\frac{1}{2}\iint r(\boldsymbol{x},\boldsymbol{y})^2 p(\boldsymbol{x})p(\boldsymbol{y})\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y} + \iint r(\boldsymbol{x},\boldsymbol{y})p(\boldsymbol{x},\boldsymbol{y})\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y} - \frac{1}{2} \tag{8}$$

Equation (7) is obtained by expanding the squared term in Equation (1), applying Equation (2) to the squared density-ratio term once, and showing that the cross-term and the remaining terms are -1 and $1/2$, respectively. Equivalence between Equations (7) and (8) can be confirmed by applying Equation (2) to the first term in Equation (8) once. Note that Equation (8) can also be obtained via the Legendre–Fenchel duality of Equation (1), implying that the optimization problem (4) corresponds to approximately maximizing the Legendre–Fenchel lower-bound [15].

2.2.2. Convergence Analysis

Here we briefly review theoretical convergence properties of LSMI.

First, let us consider the case where the function class $\mathcal{G}$ from which the function g is searched is a parametric model:

$$\mathcal{G} = \{g_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}) \mid \boldsymbol{\theta} \in \Theta \subset \mathbb{R}^b\}$$

Suppose that the true density-ratio r is contained in the model $\mathcal{G}$, *i.e.*, there exists $\boldsymbol{\theta}^*$ $(\in \Theta)$ such that: $r = g_{\boldsymbol{\theta}^*}$. Then, it was shown [28] that, under the standard regularity conditions for consistency [for example, see Section 3.2.1 of 36], it holds that:

$$\mathrm{LSMI}'(\{(\boldsymbol{x}_i,\boldsymbol{y}_i)\}_{i=1}^n) - \mathrm{SMI}(\boldsymbol{X},\boldsymbol{Y}) = \mathcal{O}_p(n^{-1/2})$$

where $\mathcal{O}_p$ denotes the asymptotic order in probability. This shows that LSMI$'$ retains the optimality in terms of the order of convergence in n, because $\mathcal{O}_p(n^{-1/2})$ is the optimal convergence rate in the parametric setup [37].

Next, we consider non-parametric cases where the function class $\mathcal{G}$ is a reproducing kernel Hilbert space [38] on $\mathcal{X} \times \mathcal{Y}$. Let us consider a non-parametric version of the LSMI optimization problem:

$$\widehat{g} := \underset{g\in\mathcal{G}}{\mathrm{argmin}}\left[\frac{1}{2n^2}\sum_{i,j=1}^n g(\boldsymbol{x}_i,\boldsymbol{y}_j)^2 - \frac{1}{n}\sum_{i=1}^n g(\boldsymbol{x}_i,\boldsymbol{y}_i) + \frac{\lambda_n}{2}\|g\|_{\mathcal{G}}^2\right]$$

where $\|\cdot\|_{\mathcal{G}}^2$ denotes the norm in the reproducing kernel Hilbert space $\mathcal{G}$. In the above optimization problem, a regularizer $\|g\|_{\mathcal{G}}^2$ is included to avoid overfitting and $\lambda_n \geq 0$ is the regularization parameter.

Suppose that the true density-ratio function r is contained in the function space $\mathcal{G}$ and is bounded from above. Then, it was shown [28] that, if $\lambda_n \to 0$ and $\lambda_n^{-1} = o(n^{2/(2+\gamma)})$ where γ $(0 < \gamma < 2)$ denotes a complexity measure of the function space $\mathcal{G}$ based on the *bracketing entropy* (The larger the value of γ is, the more complex the function space $\mathcal{G}$ is) [see p.83 of 36], it holds that:

$$\mathrm{LSMI}'(\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n) - \mathrm{SMI}(\boldsymbol{X}, \boldsymbol{Y}) = \mathcal{O}_p\big(\max(\lambda_n, n^{-1/2})\big) \tag{9}$$

The conditions $\lambda_n \to 0$ and $\lambda_n^{-1} = o(n^{2/(2+\gamma)})$ roughly mean that the regularization parameter λ_n should be sufficiently small but not too small. Equation (9) means that the convergence rate of the non-parametric version can also be $\mathcal{O}_p(n^{-1/2})$ for an appropriate choice of λ_n, but the non-parametric method requires a milder model assumption. According to [15], the above convergence rate is the minimax optimal rate under some setup. Thus, the convergence property of the above non-parametric method would also be optimal in the same sense.

2.3. Practical Implementation of LSMI

We have seen that LSMI has desirable convergence properties. Here we review practical implementation of LSMI. A MATLAB® implementation of LSMI is publicly available [39].

2.3.1. LSMI for Linear-in-Parameter Models

Let us approximate the density ratio Equation (2) using the following linear-in-parameter model:

$$g_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{\ell=1}^{b} \theta_\ell \phi_\ell(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{\theta}^\top \boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{y}) \tag{10}$$

where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_b)^\top$ are parameters, $\boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{y}) = (\phi_1(\boldsymbol{x}, \boldsymbol{y}), \ldots, \phi_b(\boldsymbol{x}, \boldsymbol{y}))^\top$ are fixed basis functions, and $^\top$ denotes the transpose. Practical choices of the basis functions will be explained in Section 2.3.2. . For this model, the LSMI optimization problem with an ℓ_2-regularizer is expressed as:

$$\widehat{\boldsymbol{\theta}} := \mathop{\mathrm{argmin}}_{\boldsymbol{\theta} \in \mathbb{R}^b} \left[\frac{1}{2} \boldsymbol{\theta}^\top \widehat{\boldsymbol{H}} \boldsymbol{\theta} - \boldsymbol{\theta}^\top \widehat{\boldsymbol{h}} + \frac{\lambda}{2} \boldsymbol{\theta}^\top \boldsymbol{\theta} \right]$$

where $\lambda \geq 0$ is the regularization parameter that controls the strength of regularization, $\widehat{\boldsymbol{H}}$ is the $b \times b$ matrix defined by:

$$\widehat{\boldsymbol{H}} := \frac{1}{n^2} \sum_{i,j=1}^{n} \boldsymbol{\phi}(\boldsymbol{x}_i, \boldsymbol{y}_j) \boldsymbol{\phi}(\boldsymbol{x}_i, \boldsymbol{y}_j)^\top$$

and $\widehat{\boldsymbol{h}}$ is the b-dimensional vector defined by:

$$\widehat{\boldsymbol{h}} := \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\phi}(\boldsymbol{x}_i, \boldsymbol{y}_i)$$

The solution $\widehat{\boldsymbol{\theta}}$ can be analytically obtained as:

$$\widehat{\boldsymbol{\theta}} = (\widehat{\boldsymbol{H}} + \lambda \boldsymbol{I}_b)^{-1} \widehat{\boldsymbol{h}} \tag{11}$$

where $\boldsymbol{I}_b$ is the b-dimensional identity matrix. Finally, LSMI is also given analytically as:

$$\mathrm{LSMI}(\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n) = \frac{1}{2}\widehat{\boldsymbol{h}}^\top \widehat{\boldsymbol{\theta}} - \frac{1}{2} \tag{12}$$

or

$$\mathrm{LSMI}'(\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n) = -\frac{1}{2}\widehat{\boldsymbol{\theta}}^\top \widehat{\boldsymbol{H}} \widehat{\boldsymbol{\theta}} + \widehat{\boldsymbol{h}}^\top \widehat{\boldsymbol{\theta}} - \frac{1}{2} \tag{13}$$

Some elements of $\widehat{\boldsymbol{\theta}}$ may take negative values in the above formulation, which can lead to negative density-ratio values and negative LSMI values. Such negative values may be rounded up to zero if necessary, although this does not happen for sufficiently large n. Another option is to explicitly impose the non-negativity constraint $\theta_1, \ldots, \theta_b \geq 0$ on the optimization problem. However, by this modification, the solution can no longer be obtained analytically, but only numerically using a quadratic program solver. (In this case, if the ℓ_2-regularizer is replaced with the ℓ_1-regularizer, the regularization path [40,41]—*i.e.*, solutions for all different regularization parameter values—can be computed efficiently without a quadratic program solver just by solving systems of linear equation [23].)

2.3.2. Design of Basis Functions

The practical accuracy of LSMI depends on the choice of basis functions in the model Equation (10). A typical choice is a non-parametric kernel model, *i.e.*, setting the number of basis function to $b = n$ and the ℓ-th basis function to $\phi_\ell(\boldsymbol{x}, \boldsymbol{y}) = K(\boldsymbol{x}, \boldsymbol{x}_\ell) L(\boldsymbol{y}, \boldsymbol{y}_\ell)$:

$$g_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{\ell=1}^n \theta_\ell K(\boldsymbol{x}, \boldsymbol{x}_\ell) L(\boldsymbol{y}, \boldsymbol{y}_\ell) \tag{14}$$

where $K(\boldsymbol{x}, \boldsymbol{x}')$ and $L(\boldsymbol{y}, \boldsymbol{y}')$ are kernel functions for $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. If n is too large, b may be set to be smaller than n and choose a subset of data points $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$ as kernel centers.

For real vector $\boldsymbol{x} \in \mathbb{R}^d$, we may practically use the Gaussian kernel for $K(\boldsymbol{x}, \boldsymbol{x}')$ after element-wise variance normalization:

$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\sigma_{\mathrm{x}}^2}\right)$$

where $\sigma_{\mathrm{x}} > 0$ is the Gaussian width. When $\boldsymbol{x}$ is a non-vectorial structured object such as a string, a tree, and a graph, we may employ a kernel function defined for such structured data [42].

In the (multi-output) regression scenario where $\boldsymbol{y}$ is a real vector, the Gaussian kernel may also be used for $L(\boldsymbol{y}, \boldsymbol{y}')$ after element-wise variance normalization:

$$L(\boldsymbol{y}, \boldsymbol{y}') = \exp\left(-\frac{\|\boldsymbol{y} - \boldsymbol{y}'\|^2}{2\sigma_{\mathrm{y}}^2}\right)$$

where $\sigma_{\mathrm{y}} > 0$ is the Gaussian width. In the multi-class classification scenario where $y \in \{1, \ldots, c\}$ and c denotes the number of classes, we may use the delta kernel for $L(y, y')$:

$$L(y, y') = \begin{cases} 1 & \text{if } y = y' \\ 0 & \text{if } y \neq y' \end{cases}$$

Note that, in the classification case with the delta kernel, the LSMI solution can be computed efficiently in a class-wise manner [33]. In the multi-label classification scenario where $\boldsymbol{y} \in \{0, 1\}^c$ and c denotes the number of labels, we may use the normalized linear kernel function [43] for $\boldsymbol{y}$:

$$L(\boldsymbol{y}, \boldsymbol{y}') = \frac{(\boldsymbol{y} - \overline{\boldsymbol{y}})^\top (\boldsymbol{y}' - \overline{\boldsymbol{y}})}{\|\boldsymbol{y} - \overline{\boldsymbol{y}}\| \|\boldsymbol{y}' - \overline{\boldsymbol{y}}'\|}$$

where $\overline{\boldsymbol{y}} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{y}_i$ is the sample mean.

2.3.3. Model Selection by Cross-Validation

Most of the above kernels include tuning parameters such as the Gaussian width, and the practical performance of LSMI depends on the choice of such kernel parameters and the regularization parameter λ. Model selection of LSMI is possible based on cross-validation with respect to the criterion J defined by Equation (3).

More specifically, the sample set $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{n}$ is divided into M disjoint subsets $\{\mathcal{D}_m\}_{m=1}^{M}$. Then the LSMI solution $\widehat{g}_m(\boldsymbol{x})$ is obtained using $\mathcal{D} \backslash \mathcal{D}_m$ (*i.e.*, all samples without $\mathcal{D}_m$), and its J-score for the hold-out samples $\mathcal{D}_m$ is computed as:

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

where $|\mathcal{D}_m|$ denotes the number of elements in the set $\mathcal{D}_m$. $\sum_{\boldsymbol{x}, \boldsymbol{y} \in \mathcal{D}_m}$ denotes the summation over all combinations of $\boldsymbol{x}$ and $\boldsymbol{y}$ in $\mathcal{D}_m$ (and thus $|\mathcal{D}_m|^2$ terms), while $\sum_{(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{D}_m}$ denotes the summation over all pairs $(\boldsymbol{x}, \boldsymbol{y})$ in $\mathcal{D}_m$ (and thus $|\mathcal{D}_m|$ terms). This procedure is repeated for $m = 1, \ldots, M$, and the average score,

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

is computed. Finally, the model (the kernel parameter and the regularization parameter in the current setup) that minimizes $\widehat{J}^{\mathrm{CV}}$ is chosen as the most suitable one.

3. SMI-Based Machine Learning

In this section, we show how the SMI estimator, LSMI, can be used for solving various machine learning tasks.

3.1. Independence Testing

First, we show how the SMI estimator can be used for independence testing.

3.1.1. Introduction

Identifying the statistical independence between random variables is one of the fundamental challenges in statistical data analysis. A traditional independence measure between random variables is the Pearson correlation coefficient, which can be used for detecting linear dependency. Recently, kernel-based independence measures have been studied to detect non-linear dependency. The Hilbert–Schmidt independence criterion (HSIC) [44] utilizes cross-covariance operators on universal reproducing kernel Hilbert spaces (RKHSs) [45], which is an infinite-dimensional generalization of covariance matrices. HSIC allows efficient detection of non-linear dependency by making use of the reproducing property of RKHSs [38]. However, HSIC has a weakness that its performance depends on the choice of RKHSs and there is no theoretically justified way to determine the RKHS properly thus far. In practice, using the Gaussian RKHS with width set to the median distance between samples is a popular heuristic [46], but this does not always work well.

To overcome the above limitations, an SMI-based independence test called *least-squares independence test* (LSIT) was proposed [26]. Below, we review LSIT.

3.1.2. Independence Testing with SMI

Let $\boldsymbol{x} \in \mathcal{X}$ be an input feature and $\boldsymbol{y} \in \mathcal{Y}$ be an output feature, which follow a joint probability distribution with density $p(\boldsymbol{x}, \boldsymbol{y})$. Suppose that we are given a set of independent and identically distributed (i.i.d.) paired samples $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$. The objective of independence testing is to conclude whether $\boldsymbol{x}$ and $\boldsymbol{y}$ are statistically independent or not, based on the samples $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$.

The SMI-based independence test, where the null hypothesis is that $\boldsymbol{x}$ and $\boldsymbol{y}$ are statistically independent, is based on the permutation test procedure [47]. More specifically, LSMI is first run using the original dataset $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$, and an SMI estimate, $\mathrm{LSMI}(\mathcal{D})$, is obtained. Next, $\{\boldsymbol{y}_i\}_{i=1}^n$ are randomly permuted and a shuffled dataset $\widetilde{\mathcal{D}} = \{(\boldsymbol{x}_i, \widetilde{\boldsymbol{y}}_i)\}_{i=1}^n$ is formed, where $\{\widetilde{\boldsymbol{y}}_i\}_{i=1}^n$ denote permuted samples. Then LSMI is run again using the shuffled dataset $\widetilde{\mathcal{D}}$, and an SMI estimate $\mathrm{LSMI}(\widetilde{\mathcal{D}})$ is obtained. Note that the random permutation eliminates the dependency between $\boldsymbol{x}$ and $\boldsymbol{y}$ (if it exists), and therefore $\mathrm{LSMI}(\widetilde{\mathcal{D}})$ would take a value close to zero. This random permutation procedure is repeated many times, and the distribution of $\mathrm{LSMI}(\widetilde{\mathcal{D}})$ under the null-hypothesis that $\boldsymbol{x}$ and $\boldsymbol{y}$ are statistically independent is constructed. Finally, the p-value is approximated by evaluating the relative ranking of $\mathrm{LSMI}(\mathcal{D})$ in the distribution of $\mathrm{LSMI}(\widetilde{\mathcal{D}})$.

This procedure is called the *least-squares independence test* (LSIT) [26]. A MATLAB® implementation of LSIT is publicly available [48].

3.2. Supervised Feature Selection

Next, we show how the SMI estimator can be used for supervised feature selection.

3.2.1. Introduction

The objective of supervised learning is to learn an input-output relation from input-output paired samples. However, when the dimensionality of input vectors is large, using all input elements could lead to a model interpretability problem. Feature selection is aimed at finding a subset of input elements that is useful for predicting output values [49].

Feature ranking is a simple implementation of feature selection that ranks each feature according to its relevance. In this feature ranking scenario, SMI between a single input variable and an output was shown to be useful [19]. However, feature ranking does not take feature interaction into account, and thus it is not useful when each single feature is not capable of predicting outputs, but multiple features are necessary for a valid prediction of outputs (e.g., an XOR problem). Two criteria, relevancy and redundancy, are often used to select multiple features simultaneously: A feature is said to be relevant if it can explain outputs, and features are said to be redundant if they are similar. Ideally, we want to find a subset of features that has high relevance and low redundancy.

Another important issue in feature selection is the computational cost: Naively selecting multiple features causes computational infeasibility because the number of possible feature combinations is exponential with respect to the number of input features. To cope with this problem, a computationally efficient method to handle multiple features called the least absolute shrinkage and selection operator (LASSO) [50] was proposed. In LASSO, a predictor consisting of a weighted sum of each feature is fitted to output values using the least-squares method, while the weight vector is confined in an ℓ_1-ball. The ℓ_1-ball restriction actually provides a notable property that the solution is sparsified, meaning that some of the weight parameters become exactly zero. Thus, LASSO automatically removes irrelevant features from its predictor, which can be achieved through convex optimization in a computationally efficient way [51,52].

However, LASSO can only handle linear predictors and its feature selection characteristic explicitly depends on the squared-loss function used in the least-squares method. To go beyond these limitations, an SMI-based feature selection method called ℓ_1-*LSMI* was proposed [27]. Below, we review ℓ_1-LSMI.

3.2.2. Feature Selection with SMI

The objective of feature selection is, from input feature vector $\boldsymbol{x} = (x^{(1)}, \ldots, x^{(d)})^\top \in \mathbb{R}^d$, to choose a subset of its elements that are useful for the prediction of output $\boldsymbol{y} \in \mathcal{Y}$. Suppose that we are given n i.i.d. paired samples $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$ drawn from a joint distribution with density $p(\boldsymbol{x}, \boldsymbol{y})$. Let $w_1, \ldots, w_d$ be feature weights for $x^{(1)}, \ldots, x^{(d)}$, and we learn the weights as:

$$\max_{w_1,\ldots,w_d} \quad \mathrm{LSMI}\left(\left\{\left((w_1 x_i^{(1)}, \ldots, w_d x_i^{(d)})^\top, \boldsymbol{y}_i\right)\right\}_{i=1}^n\right)$$
$$\text{subject to} \quad \sum_{i=1}^d w_i \leq \eta \text{ and } w_1, \ldots, w_d \geq 0$$

where $\eta \geq 0$ is the regularization parameter that controls the number of features. Because the sign of feature weights is not relevant in feature selection, they are restricted to be non-negative. For non-negative weights, $\sum_{i=1}^d w_i$ is reduced to the ℓ_1-norm of the feature weight vector $(w_1, \ldots, w_d)^\top$. The features having zero weights are regarded as irrelevant in this formulation.

To compute the solution, a simple gradient ascent may be used, where the weight vector is projected onto the positive orthant of the ℓ_1-ball in each iteration to guarantee the feasibility. This can be performed by first projecting the weight vector onto the positive orthant by rounding up negative elements to zero, and then projecting it onto the ℓ_1-ball [54].

This SMI-based feature selection algorithm is called the *ℓ_1-LSMI* [27]. A MATLAB® implementation of ℓ_1-LSMI is publicly available [53].

3.3. Supervised Feature Extraction

While feature selection chooses a subset of features for enhancing model interpretability, feature extraction finds a low-dimensional representation of features that can depend on all features (e.g., through linear combination) for improving the prediction accuracy. Here, we show how the SMI estimator can be used for supervised feature extraction.

3.3.1. Introduction

The goal of sufficient dimension reduction (SDR) is to map input features to low-dimensional expressions while "sufficient" information for predicting output values is maintained [55]. Earlier SDR methods developed in the statistics community, such as sliced inverse regression [56], principal Hessian direction [57], and sliced average variance estimation [58], rely on the ellipticity of the data (e.g., Gaussian), but such an assumption may not be fulfilled in practice. To overcome the limitations of these approaches, kernel dimension reduction (KDR) was proposed [59]. KDR employs a kernel-based dependence measure that is distribution-free, and thus KDR is flexible. However, it lacks systematic model selection strategies for kernel and regularization parameters. Furthermore, KDR scales poorly to massive datasets and there is no good way to set an initial solution for its gradient-based optimization. In practice, many restarts from different initial solutions may be needed for finding a good local optimum, which makes the entire procedure even slower and the performance of dimension reduction unreliable.

To overcome the above limitations, an SMI-based SDR method called *least-squares dimension reduction* (LSDR) was proposed [28]. An advantage of LSDR is that its tuning parameters can be systematically optimized based on cross-validation. To further address the computational and initialization issues, a heuristic search strategy for LSDR called *sufficient component analysis* (SCA) was proposed [29]. Below, we review LSDR and SCA.

3.3.2. Sufficient Dimension Reduction with SMI

First, we formulate the problem of SDR [55]. Let $\boldsymbol{x} \in \mathbb{R}^{d_{\mathrm{x}}}$ be an input vector and $\boldsymbol{y} \in \mathcal{Y}$ be an output. The goal of SDR is to find a subspace of input domain $\mathbb{R}^{d_{\mathrm{x}}}$ that contains "sufficient" information about output $\boldsymbol{y}$. We assume that there exists an orthogonal matrix $\boldsymbol{U}^* \in \mathbb{R}^{d_{\mathrm{u}} \times d_{\mathrm{x}}}$ for $d_{\mathrm{u}} \leq d_{\mathrm{x}}$ such that

$$\boldsymbol{y} \perp\!\!\!\perp \boldsymbol{x} \mid \boldsymbol{U}^* \boldsymbol{x} \tag{15}$$

That is, given the projected feature $\boldsymbol{U}^*\boldsymbol{x}$, the (remaining) feature $\boldsymbol{x}$ is conditionally independent of output $\boldsymbol{y}$ and thus can be discarded without sacrificing the predictability of $\boldsymbol{y}$. The objective of SDR is to find such $\boldsymbol{U}^*$ from n i.i.d. paired samples, $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$, drawn from a joint distribution with density $p(\boldsymbol{x}, \boldsymbol{y})$. We assume that the projection dimensionality d_{u} is known.

SMI can be used for characterizing the optimal projection matrix $\boldsymbol{U}^*$ [28]. Indeed, it was shown that inequality,

$$\mathrm{SMI}(\boldsymbol{X}, \boldsymbol{Y}) \geq \mathrm{SMI}(\boldsymbol{U}\boldsymbol{X}, \boldsymbol{Y})$$

holds, and the equality holds if and only if Equation (15) holds. Thus, maximizing $\mathrm{SMI}(\boldsymbol{U}\boldsymbol{X}, \boldsymbol{Y})$ with respect to $\boldsymbol{U}$ leads to $\boldsymbol{U}^*$. In practice, the following optimization problem is solved:

$$\max_{\boldsymbol{U} \in \mathbb{R}^{d_{\mathrm{u}} \times d_{\mathrm{x}}}} \mathrm{LSMI}(\{(\boldsymbol{U}\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n)$$
$$\text{subject to } \boldsymbol{U}\boldsymbol{U}^\top = \boldsymbol{I}_{d_{\mathrm{u}}}$$

This formulation is called *least-squares dimension reduction* (LSDR) [28].

3.3.3. Gradient-Based Subspace Search

A simple approach to solving the above LSDR optimization problem is the following iterative procedure:

- $\boldsymbol{U}$ is updated to ascend the gradient of $\mathrm{LSMI}(\{(\boldsymbol{U}\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n)$ with respect to $\boldsymbol{U}$.
- $\boldsymbol{U}$ is projected onto the feasible region specified by $\boldsymbol{U}\boldsymbol{U}^\top = \boldsymbol{I}_{d_{\mathrm{u}}}$.

The gradient of $\mathrm{LSMI}(\{(\boldsymbol{U}\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n)$ with respect to $\boldsymbol{U}$ is given by:

$$\frac{\partial \mathrm{LSMI}}{\partial \boldsymbol{U}} = \sum_{\ell=1}^{b} \widehat{\theta}_\ell \frac{\partial \widehat{h}_\ell}{\partial \boldsymbol{U}} - \frac{1}{2} \sum_{\ell,\ell'=1}^{b} \widehat{\theta}_\ell \widehat{\theta}_{\ell'} \frac{\partial \widehat{H}_{\ell,\ell'}}{\partial \boldsymbol{U}}$$

If the kernel model Equation (14) with the Gaussian kernel,

$$K(\boldsymbol{U}\boldsymbol{x}, \boldsymbol{U}\boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{U}\boldsymbol{x} - \boldsymbol{U}\boldsymbol{x}'\|^2}{2\sigma^2}\right)$$

is used, $\frac{\partial \widehat{h}_\ell}{\partial \boldsymbol{U}}$ and $\frac{\partial \widehat{H}_{\ell,\ell'}}{\partial \boldsymbol{U}}$ (for $\ell, \ell' = 1, \ldots, n$) are given by:

$$\frac{\partial \widehat{h}_\ell}{\partial \boldsymbol{U}} = -\frac{1}{n\sigma^2} \sum_{i=1}^{n} (\boldsymbol{U}\boldsymbol{x}_i - \boldsymbol{U}\boldsymbol{x}_\ell)(\boldsymbol{x}_i - \boldsymbol{x}_\ell)^\top \exp\left(-\frac{\|\boldsymbol{U}\boldsymbol{x}_i - \boldsymbol{U}\boldsymbol{x}_\ell\|^2}{2\sigma^2}\right) L(\boldsymbol{y}_i, \boldsymbol{y}_\ell),$$
$$\frac{\partial \widehat{H}_{\ell,\ell'}}{\partial \boldsymbol{U}} = \left[-\frac{1}{n\sigma^2} \sum_{i=1}^{n} \left((\boldsymbol{U}\boldsymbol{x}_i - \boldsymbol{U}\boldsymbol{x}_\ell)(\boldsymbol{x}_i - \boldsymbol{x}_\ell)^\top + (\boldsymbol{U}\boldsymbol{x}_i - \boldsymbol{U}\boldsymbol{x}_{\ell'})(\boldsymbol{x}_i - \boldsymbol{x}_{\ell'})^\top \right) \right.$$
$$\left. \times \exp\left(-\frac{\|\boldsymbol{U}\boldsymbol{x}_i - \boldsymbol{U}\boldsymbol{x}_\ell\|^2 + \|\boldsymbol{U}\boldsymbol{x}_i - \boldsymbol{U}\boldsymbol{x}_{\ell'}\|^2}{2\sigma^2}\right) \right] \times \left[\frac{1}{n} \sum_{i=1}^{n} L(\boldsymbol{y}_i, \boldsymbol{y}_\ell) L(\boldsymbol{y}_i, \boldsymbol{y}_{\ell'}) \right]$$

The projection of $\boldsymbol{U}$ onto the feasible region specified by $\boldsymbol{U}\boldsymbol{U}^\top = \boldsymbol{I}_{d_{\mathrm{u}}}$ may be carried out by the Gram–Schmidt process [60], although this is time-consuming.

An alternative way to solve the LSDR optimization problem is to perform gradient ascent on the Grassmann manifold [61]. In the Euclidean space, the ordinary gradient gives the steepest direction.

However, on a manifold, the natural gradient [62] gives the steepest direction. The natural gradient $\nabla\mathrm{LSMI}(\boldsymbol{U})$ at $\boldsymbol{U}$ is given as follows [63]:

$$\nabla\mathrm{LSMI}(\boldsymbol{U}) = \frac{\partial\mathrm{LSMI}}{\partial\boldsymbol{U}} - \frac{\partial\mathrm{LSMI}}{\partial\boldsymbol{U}}\boldsymbol{U}^\top\boldsymbol{U} = \frac{\partial\mathrm{LSMI}}{\partial\boldsymbol{U}}\boldsymbol{U}_\perp^\top\boldsymbol{U}_\perp$$

where $\boldsymbol{U}_\perp$ is any $(d-m)\times d$ matrix such that $[\boldsymbol{U}^\top\ \boldsymbol{U}_\perp^\top]$ is orthogonal. Then the geodesic from $\boldsymbol{U}$ to the direction of the natural gradient $\nabla\mathrm{LSMI}(\boldsymbol{U})$ over the Grassmann manifold can be expressed using $t\in\mathbb{R}$ as:

$$\boldsymbol{U}_t := \begin{bmatrix}\boldsymbol{I}_{d_\mathrm{x}} & \boldsymbol{O}_{d_\mathrm{x}-d_\mathrm{u}}\end{bmatrix}\exp\left(t\begin{bmatrix}\boldsymbol{O}_{d_\mathrm{u}} & \frac{\partial\mathrm{LSMI}}{\partial\boldsymbol{U}}\boldsymbol{U}_\perp^\top \\ -\boldsymbol{U}_\perp\frac{\partial\mathrm{LSMI}}{\partial\boldsymbol{U}}^\top & \boldsymbol{O}_{d_\mathrm{x}-d_\mathrm{u}}\end{bmatrix}\right)\begin{bmatrix}\boldsymbol{U}\\ \boldsymbol{U}_\perp\end{bmatrix}$$

where "exp" for a matrix denotes the matrix exponential, and $\boldsymbol{O}_{d_\mathrm{x}}$ is the $d_\mathrm{x}\times d_\mathrm{x}$ zero matrix. Thus, line search along the geodesic in the natural gradient direction is equivalent to finding the maximizer from $\{\boldsymbol{U}_t \mid t\geq 0\}$. For choosing the step size of each gradient update, some approximate line search method such as Armijo's rule [64] or backtracking line search [51] may be used.

A MATLAB® implementation of LSDR is publicly available [65].

3.3.4. Heuristic Subspace Search

Although the natural gradient method is computationally more efficient than the plain gradient method, it is still time consuming. Moreover, many restarts from different initial solutions may be needed for finding a good local optimum. Here, we introduce a heuristic method that can address these issues [29].

A key idea is to use a truncated negative quadratic function called the Epanechnikov kernel [66] as a kernel function for $\boldsymbol{U}\boldsymbol{x}$:

$$K(\boldsymbol{U}\boldsymbol{x},\boldsymbol{U}\boldsymbol{x}') = \max\left(0, 1-\frac{\|\boldsymbol{U}\boldsymbol{x}-\boldsymbol{U}\boldsymbol{x}'\|^2}{2\sigma_\mathrm{z}^2}\right)$$

Let $I(c)$ be the indicator function, *i.e.*, $I(c)=1$ if c is true and zero otherwise. Then, for the above kernel function, LSMI can be expressed as:

$$\mathrm{LSMI} = \frac{1}{2}\mathrm{tr}(\boldsymbol{U}\boldsymbol{D}_{\boldsymbol{U}}\boldsymbol{U}^\top) - \frac{1}{2}$$

where $\mathrm{tr}(\cdot)$ denotes the trace of a matrix and $\boldsymbol{D}_{\boldsymbol{U}}$ is the $d_\mathrm{x}\times d_\mathrm{x}$ matrix defined by:

$$\boldsymbol{D}_{\boldsymbol{U}} = \frac{1}{n}\sum_{i=1}^{n}\sum_{\ell=1}^{n}\widehat{\theta}_\ell(\boldsymbol{U})I\left(\frac{\|\boldsymbol{U}\boldsymbol{x}_i-\boldsymbol{U}\boldsymbol{x}_\ell\|^2}{2\sigma_\mathrm{z}^2}<1\right)L(\boldsymbol{y}_i,\boldsymbol{y}_\ell)\left[\frac{1}{d_\mathrm{u}}\boldsymbol{I}_{d_\mathrm{x}} - \frac{1}{2\sigma_\mathrm{z}^2}(\boldsymbol{x}_i-\boldsymbol{x}_\ell)(\boldsymbol{x}_i-\boldsymbol{x}_\ell)^\top\right]$$

Here, the fact that $\widehat{\theta}_\ell$ depends on $\boldsymbol{U}$ is explicitly indicated by $\widehat{\theta}_\ell(\boldsymbol{U})$.

If $\boldsymbol{U}$ in $\boldsymbol{D}_{\boldsymbol{U}}$ is replaced by $\boldsymbol{U}'$, where $\boldsymbol{U}'$ is a transformation matrix obtained in the previous iteration, the SMI estimator is simplified as:

$$\frac{1}{2}\mathrm{tr}\left(\boldsymbol{U}\boldsymbol{D}_{\boldsymbol{U}'}\boldsymbol{U}^\top\right) - \frac{1}{2} \tag{16}$$

Because $\boldsymbol{D}_{\boldsymbol{U}'}$ is independent of $\boldsymbol{U}$, a maximizer of Equation (16) with respect to $\boldsymbol{U}$ can be analytically obtained by $(\boldsymbol{u}_1|\cdots|\boldsymbol{u}_{d_\mathrm{u}})^\top$, where $\{\boldsymbol{u}_i\}_{i=1}^{d_\mathrm{u}}$ are the d_u principal components of $\boldsymbol{D}'$. The same technique can also be utilized for determining an initial transformation matrix, by computing the above solution for $\boldsymbol{U}' = \boldsymbol{I}_{d_{\mathrm{x}}}$ (*i.e.*, no dimensionality reduction).

The above heuristic search method for LSDR is called *sufficient component analysis* (SCA) [29]. A MATLAB® implementation of SCA is publicly available [67].

3.4. Canonical Dependency Analysis

Next, we show how the SMI estimator can be used for feature extraction from two sets of data.

3.4.1. Introduction

Canonical correlation analysis (CCA) [68] is a classical dimensionality reduction technique for two data sources, and it iteratively finds projection directions with maximum correlation. However, because CCA only captures correlations under linear projections, it is often insufficient to analyze complex real-world data that contain higher-order correlations. To be more flexible, non-linear CCA methods have been explored. A simple approach uses neural networks to handle non-linear projections [69,70], but neural networks are prone to local optima. Another approach first non-linearly transforms data samples into feature spaces and then apply linear CCA [71,72]. Given that the non-linear transformation is fixed, this two-step approach allows analytic computation of the global optimal solution via a generalized eigenvalue problem in the same way as linear CCA. This non-linear approach is called kernel CCA (KCCA) because reproducing kernels [38] are used as non-linear transforms. Alternating regression such as the alternating conditional expectation [73] is another possible way to find dependency in a flexible manner, which estimates transformations for two variables alternately by minimizing the squared error between transformed variables. These non-linear variants of CCA are highly flexible, although obtained results are often difficult to interpret due to the non-linearity.

The above non-linear CCA approaches can be regarded as capturing correlations along non-linear projection directions. Another extension of CCA called canonical dependency analysis (CDA) [30] captures higher-order correlations under linear projections. It was shown that KCCA with a universal kernel [45] such as the Gaussian kernel allows efficient detection of higher-order correlations [74]. However, the choice of universal kernels affects the practical performance, and there is no systematic method to choose a suitable kernel function. Another approach to higher-order CCA called informational CCA (ICCA) [75] uses mutual information (MI) as a dependency measure, where MI is estimated via kernel density estimation (KDE). Because systematic model selection strategies are available for KDE [76], ICCA could be more practical than the KCCA-based CDA method. In the ICCA method, one-dimensional projection directions are found in an iterative manner. Thus, it would be more powerful if multi-dimensional projection directions (*i.e.*, a subspace) could be directly found in CDA [30]. However, ICCA may not be reliable in such a subspace search scenario because it involves the ratio of estimated densities, which tends to produce large estimation error if the dimensionality is not small.

To overcome the above limitation, an SMI-based CDA method called *least-squares CDA* (LSCDA) was proposed [30]. Below, we review LSCDA.

3.4.2. Canonical Dependency Analysis with SMI

Suppose that we are given n i.i.d. paired samples $\{(\boldsymbol{x}_i, \boldsymbol{y}_i) \mid \boldsymbol{x}_i \in \mathbb{R}^{d_\mathrm{x}},\ \boldsymbol{y}_i \in \mathbb{R}^{d_\mathrm{y}}\}_{i=1}^{n}$ drawn from a joint distribution with density $p(\boldsymbol{x}, \boldsymbol{y})$. CDA is aimed at finding the low-dimensional expressions of $\boldsymbol{x}$ and $\boldsymbol{y}$ that are maximally dependent on each other. Here, we focus on linear dimension reduction, *i.e.*, $\boldsymbol{x}$ and $\boldsymbol{y}$ are transformed as $\boldsymbol{U}\boldsymbol{x}$ and $\boldsymbol{V}\boldsymbol{y}$, where $\boldsymbol{U} \in \mathbb{R}^{d_\mathrm{u} \times d_\mathrm{x}}$ and $\boldsymbol{V} \in \mathbb{R}^{d_\mathrm{v} \times d_\mathrm{y}}$ are orthogonal matrices with known dimensionalities d_u and d_v. The objective of CDA is to find the transformation matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ such that the statistical dependency between $\boldsymbol{U}\boldsymbol{x}$ and $\boldsymbol{V}\boldsymbol{y}$ is maximized. Let us use the SMI estimator, $\mathrm{LSMI}(\{(\boldsymbol{U}\boldsymbol{x}_i, \boldsymbol{V}\boldsymbol{y}_i)\}_{i=1}^{n})$, as the dependency measure, *i.e.*, we solve,

$$\begin{aligned} &\underset{\boldsymbol{U}\in\mathbb{R}^{d_\mathrm{u}\times d_\mathrm{x}},\boldsymbol{V}\in\mathbb{R}^{d_\mathrm{v}\times d_\mathrm{y}}}{\mathrm{argmax}} && \mathrm{LSMI}(\{(\boldsymbol{U}\boldsymbol{x}_i, \boldsymbol{V}\boldsymbol{y}_i)\}_{i=1}^{n}) \\ &\text{subject to} && \boldsymbol{U}\boldsymbol{U}^\top = \boldsymbol{I}_{d_\mathrm{u}} \ \text{ and } \ \boldsymbol{V}\boldsymbol{V}^\top = \boldsymbol{I}_{d_\mathrm{v}} \end{aligned}$$

This formulation is called *least-squares CDA* (LSCDA) [30].

The above optimization problem can be solved in the same way as LSDR presented in Section 3.3.3. A MATLAB® implementation of LSCDA is publicly available [77].

3.5. Independent Component Analysis

Here, we show how the SMI estimator can be used for independent component analysis.

3.5.1. Introduction

Suppose that there exist statistically independent sources of signals, and we observe their mixtures. The purpose of independent component analysis (ICA) [78] is to separate the mixed signals into the original source signals. An approach to ICA is to separate the mixed signals such that statistical independence among separated signals is maximized under some independence measure.

Various methods for evaluating the statistical independence among random variables from samples have been explored so far. A naive approach is to estimate probability densities based on parametric or non-parametric density estimation methods. However, finding an appropriate parametric model is not straightforward without strong prior knowledge and non-parametric estimation is not generally accurate in high-dimensional problems. Thus, this naive approach is not reliable in practice. Another approach is to approximate the entropy based on the Gram–Charlier expansion [79] or the Edgeworth expansion [80]. An advantage of this entropy-based approach is that a hard task of density estimation is not directly involved. However, these expansion techniques are based on the assumption that the target density is close to Gaussian, and violation of this assumption can cause large approximation error.

The above approaches are based on the probability densities of signals. Another line of research that does not explicitly involve probability densities employs non-linear correlation—signals are statistically independent if and only if all non-linear correlations among signals vanish. Following this line, computationally efficient algorithms have been developed based on a contrast function [81,82], which is an approximation of the entropy or mutual information. However, non-linearities in the contrast function need to be pre-specified in these methods, and thus they could be inaccurate if the predetermined non-linearities do not match the target distribution. To cope with this problem, the kernel trick has been applied in ICA, which allows computationally efficient evaluation of all non-linear correlations citeJMLR:Bach+Jordan:2002. However, its practical performance depends on the choice of kernels (more specifically, the Gaussian kernel width) and there seems no theoretically justified method to determine the kernel width. This is a critical problem in unsupervised learning tasks such as ICA.

To cope with this problem, an SMI-based ICA algorithm called *least-squares independent component analysis* (LICA) has been developed [31]. Below, we review LICA.

3.5.2. Independent Component Analysis with SMI

Suppose there are d signal sources and let: $\{\boldsymbol{x}_i \mid \boldsymbol{x}_i = (x_i^{(1)}, \ldots, x_i^{(d)})^\top \in \mathbb{R}^d\}_{i=1}^n$ be i.i.d. samples drawn from a distribution with density $p(\boldsymbol{x})$. We assume that elements $x^{(1)}, \ldots, x^{(d)}$ are statistically independent of each other, *i.e.*, $p(\boldsymbol{x})$ is factorized as:

$$p(\boldsymbol{x}) = p(x^{(1)}) \cdots p(x^{(d)})$$

We cannot directly observe $\{\boldsymbol{x}_i\}_{i=1}^n$, but only their linearly mixed samples $\{\boldsymbol{y}_i\}_{i=1}^n$:

$$\boldsymbol{y}_i := \boldsymbol{U}\boldsymbol{x}_i$$

where $\boldsymbol{U}$ is a $d \times d$ invertible matrix called the mixing matrix.

The goal of ICA is, from the mixed samples $\{\boldsymbol{y}_i\}_{i=1}^n$, to obtain a demixing matrix $\boldsymbol{V}$ that recovers the original source samples $\{\boldsymbol{x}_i\}_{i=1}^n$. We denote the demixed samples by $\{\boldsymbol{z}_i\}_{i=1}^n$:

$$\boldsymbol{z}_i = \boldsymbol{V}\boldsymbol{y}_i$$

The ideal solution is $\boldsymbol{V} = \boldsymbol{U}^{-1}$, but we can only recover the source signals up to permutation and scaling of components of $\boldsymbol{x}$ due to non-identifiability of the ICA setup [78]. Let us denote the demixed samples by:

$$\boldsymbol{z}_i = (z_i^{(1)}, \ldots, z_i^{(d)})^\top := \boldsymbol{V}\boldsymbol{y}_i$$

for $i = 1, \ldots, n$.

A direct approach to ICA is to determine $\boldsymbol{V}$ so that elements of $\boldsymbol{z}$ are as statistically independent as possible. Here, we adopt SMI as the independence measure:

$$\mathrm{SMI}(Z^{(1)}, \ldots, Z^{(d)}) := \frac{1}{2} \int \cdots \int p(z^{(1)}) \cdots p(z^{(d)}) \left(\frac{p(z^{(1)}, \ldots, z^{(d)})}{p(z^{(1)}) \cdots p(z^{(d)})} - 1 \right)^2 \mathrm{d}z^{(1)} \cdots \mathrm{d}z^{(d)}$$

We try to find the demixing matrix $\boldsymbol{V}$ that minimizes SMI. In practice, the following optimization problem is solved:

$$\min_{\boldsymbol{V} \in \mathbb{R}^{d \times d}} \mathrm{LSMI}(\{\boldsymbol{V}\boldsymbol{y}_i\}_{i=1}^n)$$

where $\mathrm{LSMI}(\{\boldsymbol{V}\boldsymbol{y}_i\}_{i=1}^n)$ is given by the same form as Equation (12) (or Equation (13)), but the matrix $\widehat{\boldsymbol{H}}$ and the vector $\widehat{\boldsymbol{h}}$ are defined in a slightly different way. For the Gaussian kernel,

$$K(\boldsymbol{V}\boldsymbol{y}, \boldsymbol{V}\boldsymbol{y}') = \exp\left(-\frac{\|\boldsymbol{V}\boldsymbol{y} - \boldsymbol{V}\boldsymbol{y}'\|^2}{2\sigma^2} \right)$$

$\widehat{\boldsymbol{H}}$ and $\widehat{\boldsymbol{h}}$ are given by:

$$\widehat{H}_{\ell,\ell'} = \frac{1}{n^d} \prod_{m=1}^{d} \left[\sum_{i=1}^{n} \exp\left(-\frac{(z_\ell^{(m)} - z_i^{(m)})^2 + (z_{\ell'}^{(m)} - z_i^{(m)})^2}{2\sigma^2} \right) \right]$$

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

This formulation is called *least-squares independent component analysis* (LICA) [31].

3.5.3. Gradient-Based Demixing Matrix Search

Based on the plain gradient technique, an update rule of $\boldsymbol{V}$ is given by:

$$\boldsymbol{V} \longleftarrow \boldsymbol{V} - t \frac{\partial \mathrm{LSMI}}{\partial \boldsymbol{V}} \tag{17}$$

where $t\ (>0)$ is the step size. The gradient $\frac{\partial \mathrm{LSMI}}{\partial \boldsymbol{V}}$ is given by:

$$\frac{\partial \mathrm{LSMI}}{\partial \boldsymbol{V}} = \sum_{\ell=1}^{n} \widehat{\theta}_\ell \frac{\partial \widehat{h}_\ell}{\partial \boldsymbol{V}} - \frac{1}{2} \sum_{\ell,\ell'=1}^{n} \widehat{\theta}_\ell \widehat{\theta}_{\ell'} \frac{\partial \widehat{H}_{\ell,\ell'}}{\partial \boldsymbol{V}}$$

where

$$\frac{\partial \widehat{h}_\ell}{\partial V_{k,k'}} = -\frac{1}{n\sigma^2} \sum_{i=1}^{n} (z_i^{(k)} - z_\ell^{(k)})(y_i^{(k')} - y_\ell^{(k')})^\top \exp\left(-\frac{\|\boldsymbol{z}_i - \boldsymbol{z}_k\|^2}{2\sigma^2} \right)$$

$$\begin{aligned} \frac{\partial \widehat{H}_{\ell,\ell'}}{\partial V_{k,k'}} = {} & \frac{1}{n^{d-1}} \prod_{m \neq k} \left[\sum_{i=1}^{n} \exp\left(-\frac{(z_i^{(m)} - z_\ell^{(m)})^2 + (z_i^{(m)} - z_{\ell'}^{(m)})^2}{2\sigma^2} \right) \right] \\ & \times \Bigg[-\frac{1}{n\sigma^2} \sum_{i=1}^{n} \left((z_i^{(k)} - z_\ell^{(k)})(y_i^{(k')} - y_\ell^{(k')}) + (z_i^{(k)} - z_{\ell'}^{(k)})(y_i^{(k')} - y_{\ell'}^{(k')}) \right) \\ & \quad \times \exp\left(-\frac{(z_i^{(k)} - v_\ell^{(k)})^2 + (z_i^{(k)} - z_{\ell'}^{(k)})^2}{2\sigma^2} \right) \Bigg] \end{aligned}$$

In ICA, scaling of components of $\boldsymbol{z}$ can be arbitrary. This implies that the above gradient updating rule can lead to a solution with poor scaling, which is not preferable from a numerical viewpoint. To avoid possible numerical instability, $\boldsymbol{V}$ is normalized at each gradient iteration as:

$$V_{k,k'} \longleftarrow \frac{V_{k,k'}}{\sqrt{\sum_{m=1}^{d} V_{k,m}^2}}$$

3.5.4. Natural Gradient Demixing Matrix Search

Suppose that data samples are whitened, *i.e.*, samples $\{\boldsymbol{y}_i\}_{i=1}^n$ are pre-transformed as:

$$\boldsymbol{y}_i \longleftarrow \widehat{\boldsymbol{\Sigma}}^{-\frac{1}{2}} \boldsymbol{y}_i$$

where $\widehat{\boldsymbol{\Sigma}}$ is the sample covariance matrix:

$$\widehat{\boldsymbol{\Sigma}} := \frac{1}{n} \sum_{i=1}^{n} \left(\boldsymbol{y}_i - \frac{1}{n} \sum_{j=1}^{n} \boldsymbol{y}_j \right) \left(\boldsymbol{y}_i - \frac{1}{n} \sum_{j=1}^{n} \boldsymbol{y}_j \right)^\top$$

Then it can be shown that any demixing matrix that eliminates the second order correlation is an orthogonal matrix [78]. Thus, for whitened data, the search space of $\boldsymbol{V}$ can be restricted to the orthogonal group without loss of generality. The natural gradient [62] update rule on the orthogonal group is given by:

$$\boldsymbol{V} \longleftarrow \boldsymbol{V} \exp\left(-t\left(\boldsymbol{V}^\top \frac{\partial \mathrm{LSMI}}{\partial \boldsymbol{V}} - \frac{\partial \mathrm{LSMI}}{\partial \boldsymbol{V}}^\top \boldsymbol{V}\right)\right)$$

where "exp" for a matrix denotes the matrix exponential and t (> 0) is the step size.

A MATLAB® implementation of LICA is publicly available [83].

3.6. Cross-Domain Object Matching

Next, we show how the SMI estimator can be used for cross-domain object matching.

3.6.1. Introduction

The objective of cross-domain object matching is to match two sets of unpaired objects in different domains. For example, in photo album summarization, we are given a set of photos and a designed photo frame expressed as a set of photo slots in the Cartesian coordinate system, and we want to automatically assign the photos into the designed photo frame. A typical approach of cross-domain object matching is to find a mapping from objects in one domain (photos) to objects in the other domain (frame) so that the pairwise dependency is maximized. In this scenario, accurately evaluating the dependence between objects is a key issue.

Kernelized sorting [84] tries to find the mapping between two domains that maximizes mutual information under the Gaussian assumption. However, because the Gaussian assumption may not be fulfilled in practice, this method tends to perform poorly. To overcome the above limitation, the kernel-based dependence measure called the Hilbert–Schmidt independence criterion (HSIC) [85] was proposed to use in kernelized sorting [86]. Because HSIC is distribution-free, HSIC-based kernelized sorting is more flexible than the original method based on the Gaussian assumption. However, HSIC includes a tuning parameter (more specifically, the Gaussian kernel width), and its choice is crucial to obtain better performance [87].

To cope with this problem, an SMI-based cross-domain object matching method called *least-squares object matching* (LSOM) was developed [32]. Below, we review LSOM.

3.6.2. Cross-Domain Object Matching with SMI

The goal of cross-domain object matching is, given two sets of *unpaired* samples of the same size, $\{\boldsymbol{x}_i \mid \boldsymbol{x}_i \in \mathcal{X}\}_{i=1}^n$ and $\{\boldsymbol{y}_i \mid \boldsymbol{y}_i \in \mathcal{Y}\}_{i=1}^n$, to find a mapping that well "matches" them. Let π be a permutation function over $\{1, \ldots, n\}$. The optimal permutation, denoted by π^*, can be obtained as the maximizer of the dependency between the two sets $\{\boldsymbol{x}_i\}_{i=1}^n$ and $\{\boldsymbol{y}_{\pi(i)}\}_{i=1}^n$. Here, we use the SMI approximator, $\mathrm{LSMI}(\{(\boldsymbol{x}_i, \boldsymbol{y}_{\pi(i)})\}_{i=1}^n)$, as the dependency measure, *i.e.*, we solve,

$$\max_{\pi} \mathrm{LSMI}(\{(\boldsymbol{x}_i, \boldsymbol{y}_{\pi(i)})\}_{i=1}^n)$$

Let $\boldsymbol{K}$ and $\boldsymbol{L}$ be the $n \times n$ kernel matrices defined by $K_{i,j} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $L_{i,j} = L(\boldsymbol{y}_i, \boldsymbol{y}_j)$. Then LSMI for $\{(\boldsymbol{x}_i, \boldsymbol{y}_{\pi(i)})\}_{i=1}^n$ can be expressed as:

$$\mathrm{LSMI}(\{(\boldsymbol{x}_i, \boldsymbol{y}_{\pi(i)})\}_{i=1}^n) = \frac{1}{2n}\mathrm{tr}\left(\boldsymbol{\Pi}^\top \boldsymbol{L}\boldsymbol{\Pi}\widehat{\boldsymbol{\Theta}}_{\boldsymbol{\Pi}}\boldsymbol{K}\right) - \frac{1}{2} \tag{18}$$

where $\boldsymbol{\Pi}$ is the permutation matrix corresponding to π, *i.e.*, $\boldsymbol{\Pi}$ is the $n \times n$ zero-one matrix such that $\Pi_{i,j} = 1$ if $i = \pi(j)$ for $j = 1, \ldots, n$ and $\Pi_{i,j} = 0$ otherwise. $\widehat{\boldsymbol{\Theta}}_{\boldsymbol{\Pi}}$ is the diagonal matrix with diagonal elements given by the LSMI solution $\widehat{\boldsymbol{\theta}}_\pi$ obtained by paired data $\{(\boldsymbol{x}_i, \boldsymbol{y}_{\pi(i)})\}_{i=1}^n$ (see Equation (11)).

Because maximizing Equation (18) with respect to $\boldsymbol{\Pi}$ is computationally infeasible, greedy update from previous solution $\boldsymbol{\Pi}'$ is used in practice:

$$\boldsymbol{\Pi}^{\mathrm{new}} = (1-t)\boldsymbol{\Pi}' + t \cdot \underset{\boldsymbol{\Pi}}{\mathrm{argmax}}\ \mathrm{tr}\left(\boldsymbol{\Pi}^\top \boldsymbol{L}\boldsymbol{\Pi}'\widehat{\boldsymbol{\Theta}}_{\boldsymbol{\Pi}'}\boldsymbol{K}\right)$$

where $0 < t \leq 1$ is the step size. Maximization of the second term is called a linear assignment problem, which can be solved efficiently by the Hungarian method [88].

The above method is called *least-squares object matching* (LSOM) [32]. A MATLAB® implementation of LSOM is publicly available [89].

3.7. Clustering

Here, we show how SMI can be effectively used for clustering.

3.7.1. Introduction

The objective of clustering is to classify data samples into disjoint groups in an unsupervised manner. K-means [90] is a classic but still popular clustering algorithm. However, k-means only produces linearly separated clusters, and thus its usefulness is rather limited in practice. To cope with this problem, various non-linear clustering methods have been developed. Kernel k-means [91] performs k-means in a feature space induced by a reproducing kernel function [46]. Spectral clustering [92,93] first unfolds non-linear data manifolds by a spectral embedding method, and then performs k-means in the embedded space. Blurring mean-shift [94,95] 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 [96,97]. Dependence-maximization clustering determines cluster assignments so that their dependence on input data is maximized [34,98,99].

Information-maximization clustering exhibited the state-of-the-art performance [100,101], where probabilistic classifiers such as a kernelized Gaussian classifier [100] and a kernel logistic regression classifier [101] are learned so that mutual information between feature vectors and cluster assignments is maximized in an unsupervised manner. A notable advantage of information-maximization clustering is that classifier training is formulated as continuous optimization, which is substantially simpler than discrete optimization of cluster assignments. Indeed, classifier training can be carried out in computationally efficient manners by a gradient method [100] or a quasi-Newton method [101]. Furthermore, a model selection strategy based on the information-maximization principle is also provided [100]. Thus, kernel parameters can be systematically optimized in an unsupervised way. However, the optimization problems of these clustering methods are non-convex and finding a good local optimal solution is not straightforward in practice.

To overcome the above limitation, an SMI-based clustering method called *SMI clustering* (SMIC) was proposed [33]. Below, we review SMIC.

3.7.2. Clustering with SMI

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 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 clusters. c is assumed to be pre-fixed below. To solve the clustering problem, the information-maximization approach is taken [100,101]. That is, clustering is regarded as an unsupervised classification problem, and the class-posterior probability $p(y|\boldsymbol{x})$ is learned so that "information" between feature vector $\boldsymbol{x}$ and cluster label y is maximized.

As an information measure, SMI Equation (1) is adopted, which can expressed as:

$$\mathrm{SMI} = \frac{1}{2}\int \sum_{y=1}^{c} p(y|\boldsymbol{x})p(\boldsymbol{x})\frac{p(y|\boldsymbol{x})}{p(y)}\mathrm{d}\boldsymbol{x} - \frac{1}{2} \tag{19}$$

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, $\{\pi_y\}_{y=1}^c$ are assumed to be sorted in the ascending order:

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

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

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

Substituting π_y into $p(y)$, we can express Equation (19) as:

$$\frac{1}{2}\int \sum_{y=1}^{c} \frac{1}{\pi_y} p(y|\boldsymbol{x})p(\boldsymbol{x})p(y|\boldsymbol{x})\mathrm{d}\boldsymbol{x} - \frac{1}{2} \tag{20}$$

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

$$q_{\boldsymbol{\alpha}}(y|\boldsymbol{x}) := \sum_{i=1}^{n} \alpha_{y,i}K(\boldsymbol{x},\boldsymbol{x}_i), \tag{21}$$

where $\boldsymbol{\alpha} = (\alpha_{1,1},\ldots,\alpha_{c,n})^\top$ is the parameter vector and $K(\boldsymbol{x},\boldsymbol{x}')$ denotes a kernel function. A useful example of kernel functions is the local-scaling kernel [102] defined as:

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

where $\mathcal{N}_k(\boldsymbol{x})$ denotes the set of k nearest neighbors for $\boldsymbol{x}$ (k is the kernel parameter), σ_i is a local scaling factor defined as $\sigma_i = \|\boldsymbol{x}_i - \boldsymbol{x}_i^{(k)}\|$, and $\boldsymbol{x}_i^{(k)}$ is the k-th nearest neighbor of $\boldsymbol{x}_i$. Note that we did not include the normalization term in Equation (21) because model outputs will be normalized later (see Equation (22)).

Further approximating the expectation with respect to $p(\boldsymbol{x})$ included in Equation (20) by the empirical average of samples $\{\boldsymbol{x}_i\}_{i=1}^n$, we arrive at the following SMI approximator:

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

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

For each cluster y, $\boldsymbol{\alpha}_y^\top \boldsymbol{K}^2 \boldsymbol{\alpha}_y$ is maximized under $\|\boldsymbol{\alpha}_y\| = 1$. Since this is the Rayleigh quotient, the maximizer is given by the normalized principal eigenvector of $\boldsymbol{K}$ [104]. To avoid all the solutions $\{\boldsymbol{\alpha}_y\}_{y=1}^c$ to be reduced to the same principal eigenvector, their mutual orthogonality is imposed:

$$\boldsymbol{\alpha}_y^\top \boldsymbol{\alpha}_{y'} = 0 \quad \text{for } y \neq y'$$

Then the solutions are given by the normalized eigenvectors $\boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_c$ associated with the eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n \geq 0$ of $\boldsymbol{K}$. Since the sign of $\boldsymbol{\psi}_y$ is arbitrary, the sign is set as:

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

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

On the other hand, because

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

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

$$\boldsymbol{\alpha}_y^\top \boldsymbol{K} \boldsymbol{1}_n = \pi_y \tag{22}$$

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

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

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

where 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 $\boldsymbol{K}\widetilde{\boldsymbol{\psi}}_y = \lambda_y \widetilde{\boldsymbol{\psi}}_y$ was used in the above derivation. For out-of-sample prediction, cluster assignment y' for new sample $\boldsymbol{x}'$ may be obtained as:

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

The above method is called *SMI-based clustering* (SMIC) [33]. LSMI can be used for model selection of SMIC, *i.e.*, LSMI is computed as a function of the kernel parameter included in $K(\boldsymbol{x}, \boldsymbol{x}')$ and the maximizer of LSMI is chosen as the most promising one. A MATLAB® implementation of SMIC is publicly available [103].

3.8. Causal Direction Estimation

Finally, we show how the SMI estimator can be used for causal direction estimation.

3.8.1. Introduction

Learning causality from data is one of the important challenges in the artificial intelligence, statistics, and machine learning communities [105]. A traditional method of learning causal relationship from observational data is based on the linear-dependence Gaussian-noise model [106]. However, the linear-Gaussian assumption is too restrictive and may not be fulfilled in practice. Recently, non-Gaussianity and non-linearity have been shown to be beneficial in causal inference, because it can break symmetry between observed variables [107,108]. Since then, much attention has been paid to the discovery of non-linear causal relationship through non-Gaussian noise models [109].

In the framework of non-linear non-Gaussian causal inference, the relation between a cause X and an effect Y is assumed to be described by $Y = f(X) + E$, where f is a non-linear function and E is non-Gaussian additive noise that is independent of the cause X. Under this additive noise assumption, it was shown [108] that the causal direction between X and Y can be identified based on a hypothesis test of whether the causal model $Y = f(X) + E$ or the alternative model $X = f'(Y) + E'$ fits the data well—here, the goodness of fit is measured by independence between inputs and residuals (*i.e.*, estimated noise). In [108], the functions f and f' were learned by the Gaussian process (GP) regression [110], and the independence between inputs and residuals was evaluated by the Hilbert–Schmidt independence criterion (HSIC) [85].

However, standard regression methods such as GP are designed to handle Gaussian noise, and thus they may not be suited for discovering causality in the non-Gaussian additive noise formulation. To cope with this problem, an alternative regression method called HSIC regression was proposed [109], which learns a function so that the dependence between inputs and residuals is directly minimized based on HSIC. Through experiments, HSIC regression was shown to outperform the GP-based method [109]. However, the choice of the kernel width in HSIC regression heavily affects the sensitivity of the independence measure, and systematic model selection strategies are not available. Another weakness of HSIC regression is that the kernel width of the regression model is fixed to the same value as HSIC. This crucially limits the flexibility of function approximation in HSIC regression.

To overcome the above weaknesses, an SMI-based regression method for causal inference called *least-squares independence regression* (LSIR) was developed [35]. Below, we review LSIR.

3.8.2. Dependence Minimizing Regression with SMI

Suppose random variables $X \in \mathbb{R}$ and $Y \in \mathbb{R}$ are connected by the following additive noise model [108]:

$$Y = f(X) + E$$

where $f : \mathbb{R} \to \mathbb{R}$ is some non-linear function and $E \in \mathbb{R}$ is a zero-mean random variable that is independent of X. The goal of dependence minimizing regression is, from i.i.d. paired samples $\{(x_i, y_i)\}_{i=1}^{n}$, to obtain a function $\widehat{f}$ such that input X and estimated additive noise $\widehat{E} = Y - \widehat{f}(X)$ are independent.

Let us employ a linear model for dependence minimizing regression:

$$f_{\boldsymbol{\beta}}(x) = \sum_{l=1}^{m} \beta_l \psi_l(x) = \boldsymbol{\beta}^\top \boldsymbol{\psi}(x)$$

where m is the number of basis functions, $\boldsymbol{\beta} = (\beta_1, \dots, \beta_m)^\top$ are regression parameters, and $\boldsymbol{\psi}(x) = (\psi_1(x), \dots, \psi_m(x))^\top$ are basis functions. In LSMI-based dependence minimization regression, the regression parameters $\boldsymbol{\beta}$ are learned as:

$$\min_{\boldsymbol{\beta}} \left[\mathrm{LSMI}\big(\{(x_i, e_i)\}_{i=1}^{n}\big) + \frac{\gamma}{2} \boldsymbol{\beta}^\top \boldsymbol{\beta} \right]$$

where $e_i = y_i - f_{\boldsymbol{\beta}}(x_i)$ is the residual and $\gamma > 0$ is the regularization parameter to avoid overfitting.

For regression parameter learning, a gradient descent method may be used:

$$\boldsymbol{\beta} \longleftarrow \boldsymbol{\beta} - t \left(\frac{\partial \mathrm{LSMI}}{\partial \boldsymbol{\beta}} + \gamma \boldsymbol{\beta} \right)$$

where t is the step size. The gradient $\frac{\partial \mathrm{LSMI}}{\partial \boldsymbol{\beta}}$ can be approximately expressed as:

$$\frac{\partial \mathrm{LSMI}}{\partial \boldsymbol{\beta}} = \sum_{\ell=1}^{n} \widehat{\theta}_\ell \frac{\partial \widehat{h}_\ell}{\partial \boldsymbol{\beta}} - \frac{1}{2} \sum_{\ell,\ell'=1}^{n} \widehat{\theta}_\ell \widehat{\theta}_{\ell'} \frac{\partial \widehat{H}_{\ell,\ell'}}{\partial \boldsymbol{\beta}}$$

where

$$\frac{\partial \widehat{h}_\ell}{\partial \boldsymbol{\beta}} \approx -\frac{1}{2n\sigma^2} \sum_{j=1}^{n} \exp\left(-\frac{(x_i - x_\ell)^2 + (e_i - e_\ell)^2}{2\sigma^2} \right) (e_i - e_\ell) \boldsymbol{\psi}(x_i),$$

$$\frac{\partial \widehat{H}_{\ell,\ell'}}{\partial \boldsymbol{\beta}} \approx -\frac{1}{2n^2\sigma^2} \sum_{i,j=1}^{n} \exp\left(-\frac{(x_i - x_\ell)^2 + (e_j - e_\ell)^2 + (x_i - x_{\ell'})^2 + (e_j - e_{\ell'})^2}{2\sigma^2} \right) \times \Big((e_j - e_\ell) \boldsymbol{\psi}(x_i) + (e_i - e_\ell) \boldsymbol{\psi}(x_j) \Big)$$

Note that, in the above derivation, the dependence of $\boldsymbol{\beta}$ on e_i is ignored for simplicity. Although it is possible to exactly compute the derivative in principle, this approximated expression is computationally more efficient with good performance in practice.

By taking into account the assumption that the mean of noise E is zero, the final regressor is obtained as:

$$\widehat{f}(x) = f_{\widehat{\boldsymbol{\beta}}}(x) + \frac{1}{n}\sum_{i=1}^{n}\Big(y_i - f_{\widehat{\boldsymbol{\beta}}}(x_i)\Big)$$

This method is called *least-squares independence regression* (LSIR) [35]. A MATLAB® implementation of LSIR is publicly available [111].

3.8.3. Causal Direction Inference by LSIR

Our final goal is, given i.i.d. paired samples $\{(x_i, y_i)\}_{i=1}^n$, to determine whether X causes Y or vice versa under the additive noise assumption. To this end, we test whether the causal model $Y = f_Y(X) + E_Y$ or the alternative model $X = f_X(Y) + E_X$ fits the data well, where the goodness of fit is measured by independence between inputs and residuals (*i.e.*, estimated noise). Independence of inputs and residuals may be decided in practice based on the permutation test procedure [47].

More specifically, LSIR is first run for $\{(x_i, y_i)\}_{i=1}^n$ as usual, and obtain a regression function $\widehat{f}$. This procedure also provides an SMI estimate, $\mathrm{LSMI}(\{(x_i, \widehat{e}_i)\}_{i=1}^n)$, where $\widehat{e}_i = y_i - \widehat{f}(x_i)$. Next, pairs of input and residual $\{(x_i, \widehat{e}_i)\}_{i=1}^n$ are randomly permuted as $\{(x_i, \widehat{e}_{\pi(i)})\}_{i=1}^n$, where $\pi(\cdot)$ is a randomly generated permutation function. Note that the permuted pairs of samples are independent of each other because the random permutation breaks the dependency between X and $\widehat{E}$ (if it exists). Then, an SMI estimate for the permuted data, $\mathrm{LSMI}(\{(x_i, \widehat{e}_{\pi(i)})\}_{i=1}^n)$, is computed. This random permutation process is repeated many times, and the distribution of LSMI values under the null-hypothesis that X and $\widehat{E}$ are independent is constructed. Finally, the p-value is approximated by evaluating the relative ranking of LSMI computed from the original input-residual data, $\mathrm{LSMI}(\{(x_i, \widehat{e}_i)\}_{i=1}^n)$, over the distribution of LSMI values for randomly permuted data.

In order to decide the causal direction, the p-values $p_{X\to Y}$ and $p_{X\leftarrow Y}$ for both directions $X \to Y$ (*i.e.*, X causes Y) and $X \leftarrow Y$ (*i.e.*, Y causes X) are computed. Then, for a given significance level δ, the causal direction is determined as follows:

- If $p_{X\to Y} > \delta$ and $p_{X\leftarrow Y} \leq \delta$, the causal model $X \to Y$ is chosen.
- If $p_{X\leftarrow Y} > \delta$ and $p_{X\to Y} \leq \delta$, the causal model $X \leftarrow Y$ is selected.
- If $p_{X\to Y}, p_{X\leftarrow Y} \leq \delta$, perhaps there is no causal relation between X and Y or our modeling assumption is not correct (e.g., an unobserved confounding variable exists).
- If $p_{X\to Y}, p_{X\leftarrow Y} > \delta$, perhaps our modeling assumption is not correct or it is not possible to identify a causal direction (*i.e.*, X, Y, and E are Gaussian random variables).

When we have prior knowledge that there exists a causal relation between X and Y but the causal direction is unknown, the values of $p_{X\to Y}$ and $p_{X\leftarrow Y}$ may be simply compared for determining the causal direction as follows:

- If $p_{X\to Y} > p_{X\leftarrow Y}$, we conclude that X causes Y.
- Otherwise, we conclude that Y causes X.

This simplified procedure does not include the computational expensive permutation process and thus it is computationally very efficient.

4. Conclusions

In this article, we reviewed recent development in the estimation of *squared-loss mutual information* (SMI) and its application to machine learning. The key idea for accurately estimating SMI is to directly estimate the ratio of probability densities without separately estimating each density. A notable advantage of the SMI estimator called *least-squares mutual information* (LSMI) [19] is that it can be computed analytically in a computationally more efficient and numerically more stable way than ordinary MI.

We have introduced SMI as a measure of statistical independence between random variables. On the other hand, ordinary MI has a rich information-theoretic interpretation via entropies. Thus, it is important to investigate an information-theoretic meaning of SMI, which remains to be an open question currently.

Various methods of direct density-ratio estimation have been explored so far [16,18], and such density ratio estimators were shown to be applicable to an even wider class of machine learning tasks beyond SMI estimation, such as non-stationarity adaptation [112], outlier detection [113], change detection [114,115], class-balance estimation [116], two-sample homogeneity testing [117,118], probabilistic classification [119,120], and conditional density estimation [121].

Improving the accuracy of density ratio estimation contributes to enhancing the performance of the above machine learning solutions. Recent advances in this line of research include dimensionality reduction for density ratio estimation [122–124], a unified statistical framework of density ratio estimation [18], and extensions to relative density ratios [125] and density differences [126]. Further improving the accuracy and computational efficiency and exploring new application areas are important future directions to pursue.

More program codes are publicly available [127].

Acknowledgements

The author would like to thank Taiji Suzuki, Makoto Yamada, Wittawat Jitkrittum, Masayuki Karasuyama, Marthinus Christoffel du Plessis, and John Quinn for their valuable comments. This work was supported by the JST PRESTO program, the FIRST program, and AOARD.

References

1. Shannon, C. A mathematical theory of communication. *AT&T Tech. J.* **1948**, *27*, 379–423.
2. Cover, T.M.; Thomas, J.A. *Elements of Information Theory*, 2nd ed.; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2006.
3. Kullback, S.; Leibler, R.A. On information and sufficiency. *Ann. Math. Stat.* **1951**, *22*, 79–86.
4. Fraser, A.M.; Swinney, H.L. Independent coordinates for strange attractors from mutual information. *Phys. Rev. A* **1986**, *33*, 1134–1140.
5. Vapnik, V.N. *Statistical Learning Theory*; Wiley: New York, NY, USA, 1998.
6. Darbellay, G.A.; Vajda, I. Estimation of the information by an adaptive partitioning of the observation space. *IEEE Trans. Inf. Theory* **1999**, *45*, 1315–1321.

7. Wang, Q.; Kulkarmi, S.R.; Verdú, S. Divergence estimation of continuous distributions based on data-dependent partitions. *IEEE Trans. Inf. Theory* **2005**, *51*, 3064–3074.
8. Silva, J.; Narayanan, S. Universal Consistency of Data-Driven Partitions for Divergence Estimation. In Proceedings of IEEE International Symposium on Information Theory, Nice, France, 24–29 June 2007; pp. 2021–2025.
9. Kraskov, A.; Stögbauer, H.; Grassberger, P. Estimating mutual information. *Phys. Rev. E* **2004**, *69*, 066138.
10. Khan, S.; Bandyopadhyay, S.; Ganguly, A.; Saigal, S. Relative performance of mutual information estimation methods for quantifying the dependence among short and noisy data. *Phys. Rev. E* **2007**, *76*, 026209.
11. Pérez-Cruz, F. Kullback-Leibler Divergence Estimation of Continuous Distributions. In Proceedings of IEEE International Symposium on Information Theory, Toronto, Canada, 6–11 July 2008; pp. 1666–1670.
12. Van Hulle, M.M. Edgeworth approximation of multivariate differential entropy. *Neural Comput.* **2005**, *17*, 1903–1910.
13. Suzuki, T.; Sugiyama, M.; Sese, J.; Kanamori, T. Approximating Mutual Information by Maximum Likelihood Density Ratio Estimation. In Proceedings of ECML-PKDD2008 Workshop on New Challenges for Feature Selection in Data Mining and Knowledge Discovery 2008 (FSDM2008); Saeys, Y., Liu, H., Inza, I., Wehenkel, L., de Peer, Y.V., Eds.; 2008; Volume 4, *JMLR Workshop and Conference Proceedings*, pp. 5–20.
14. Sugiyama, M.; Suzuki, T.; Nakajima, S.; Kashima, H.; von Bünau, P.; Kawanabe, M. Direct importance estimation for covariate shift adaptation. *Ann. I. Stat. Math.* **2008**, *60*, 699–746.
15. Nguyen, X.; Wainwright, M.J.; Jordan, M.I. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Trans. Inf. Theory* **2010**, *56*, 5847–5861.
16. Sugiyama, M.; Suzuki, T.; Kanamori, T. *Density Ratio Estimation in Machine Learning*; Cambridge University Press: Cambridge, UK, 2012.
17. Basu, A.; Harris, I.R.; Hjort, N.L.; Jones, M.C. Robust and efficient estimation by minimising a density power divergence. *Biometrika* **1998**, *85*, 549–559.
18. Sugiyama, M.; Suzuki, T.; Kanamori, T. Density ratio matching under the bregman divergence: A unified framework of density ratio estimation. *Ann. I. Stat. Math.* **2012**, *64*, 1009–1044.
19. Suzuki, T.; Sugiyama, M.; Kanamori, T.; Sese, J. Mutual information estimation reveals global associations between stimuli and biological processes. *BMC Bioinf.* **2009**, *10*, S52:1–S52:12.
20. Pearson, K. 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. *Philos. Mag. Series 5* **1900**, *50*, 157–175.
21. Ali, S.M.; Silvey, S.D. A general class of coefficients of divergence of one distribution from another. *J. R. Stat. Soc. Series B* **1966**, *28*, 131–142.
22. Csiszár, I. Information-type measures of difference of probability distributions and indirect observation. *Stud. Sci. Math. Hung.* **1967**, *2*, 229–318.
23. Kanamori, T.; Hido, S.; Sugiyama, M. A least-squares approach to direct importance estimation. *J. Mach. Learn. Res.* **2009**, *10*, 1391–1445.

24. Kanamori, T.; Suzuki, T.; Sugiyama, M. Statistical Analysis of kernel-based least-squares density-ratio estimation. *Mach. Learn.* **2012**, *86*, 335–367.
25. Kanamori, T.; Suzuki, T.; Sugiyama, M. Computational complexity of kernel-based density-ratio estimation: A condition number analysis. **2009**, arXiv:0912.2800.
26. Sugiyama, M.; Suzuki, T. Least-squares independence test. *IEICE T. Inf. Syst.* **2011**, *E94-D*, 1333–1336.
27. Jitkrittum, W.; Hachiya, H.; Sugiyama, M. Feature Selection via ℓ_1-Penalized Squared-Loss Mutual Information. Technical Report 1210.1960, arXiv, 2012.
28. Suzuki, T.; Sugiyama, M. Sufficient dimension reduction via squared-loss mutual information estimation. Available online: sugiyama-www.cs.titech.ac.jp/.../AISTATS2010b.pdf (accessed on 26 December 2012).
29. Yamada, M.; Niu, G.; Takagi, J.; Sugiyama, M. Computationally Efficient Sufficient Dimension Reduction via Squared-Loss Mutual Information. In Proceedings of the Third Asian Conference on Machine Learning (ACML2011); Hsu, C.N., Lee, W.S., Eds.; 2011; Volume 20, *JMLR Workshop and Conference Proceedings*, pp. 247–262.
30. Karasuyama, M.; Sugiyama. Canonical dependency analysis based on squared-loss mutual information. *Neural Netw.* **2012**, *34*, 46–55.
31. Suzuki, T.; Sugiyama, M. Least-squares independent component analysis. *Neural Comput.* **2011**, *23*, 284–301.
32. Yamada, M.; Sugiyama, M. Cross-Domain Object Matching with Model Selection. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS2011); Gordon, G., Dunson, D.; Dudík, M., Eds.; 2011; Volume 15, *JMLR Workshop and Conference Proceedings*, pp. 807–815.
33. Sugiyama, M.; Yamada, M.; Kimura, M.; Hachiya, H. On Information-Maximization Clustering: Tuning Parameter Selection and Analytic Solution. In Proceedings of 28th International Conference on Machine Learning (ICML2011); Getoor, L., Scheffer, T., Eds.; 2011; pp. 65–72.
34. Kimura, M.; Sugiyama, M. Dependence-maximization clustering with least-squares mutual information. *J. Adv. Comput. Intell. Intell. Inf.* **2011**, *15*, 800–805.
35. Yamada, M.; Sugiyama, M. 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)*; The AAAI Press: Atlanta, Georgia, USA, 2010; pp. 643–648.
36. Van der Vaart, A.W.; Wellner, J.A. *Weak Convergence and Empirical Processes with Applications to Statistics*; Springer: New York, NY, USA, 1996.
37. Van der Vaart, A.W. *Asymptotic Statistics*; Cambridge University Press: Cambridge, MA, USA, 2000.
38. Aronszajn, N. Theory of reproducing kernels. *T. Am. Math. Soc.* **1950**, *68*, 337–404.
39. Least-Squares Mutual Information (LSMI). Available online: http://sugiyama-www.cs.titech.ac.jp/~sugi/software/LSMI/ (accessed on 7 December 2012).
40. Efron, B.; Hastie, T.; Johnstone, I.; Tibshirani, R. Least angle regression. *Ann. Stat.* **2004**, *32*, 407–499.

41. Hastie, T.; Rosset, S.; Tibshirani, R.; Zhu, J. The entire regularization path for the support vector machine. *J. Mach. Learn. Res.* **2004**, *5*, 1391–1415.
42. Gärtner, T. A survey of kernels for structured data. *SIGKDD Explor.* **2003**, *5*, S268–S275.
43. Sarwar, B.; Karypis, G.; Konstan, J.; Reidl, J. Item-Based Collaborative Filtering Recommendation Algorithms. In Proceedings of the 10th International Conference on World Wide Web (WWW2001), Hong Kong, China, 1–5 May 2001; pp. 285–295.
44. Gretton, A.; Fukumizu, K.; Teo, C.H.; Song, L.; Schölkopf, B.; Smola, A. A Kernel Statistical Test of Independence. Advances in Neural Information Processing Systems 20; Platt, J.C., Koller, D., Singer, Y., Roweis, S., Eds.; MIT Press: Cambridge, MA, USA, 2008; pp. 585–592.
45. Steinwart, I. On the influence of the kernel on the consistency of support vector machines. *J. Mach. Learn. Res.* **2001**, *2*, 67–93.
46. Schölkopf, B.; Smola, A.J. *Learning with Kernels*; MIT Press: Cambridge, MA, USA, 2002.
47. Efron, B.; Tibshirani, R.J. *An Introduction to the Bootstrap*; Chapman & Hall/CRC: New York, NY, USA, 1993.
48. Least-Squares Independence Test (LSIT). Available online: http://sugiyama-www.cs.titech.ac.jp/~sugi/software/LSIT/ (accessed on 7 December 2012).
49. Guyon, I.; Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res.* **2003**, *3*, 1157–1182.
50. Tibshirani, R. Regression shrinkage and subset selection with the lasso. *J. R. Stat. Soc. Series B* **1996**, *58*, 267–288.
51. Boyd, S.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, UK, 2004.
52. Tomioka, R.; Suzuki, T.; Sugiyama, M. Super-linear convergence of dual augmented Lagrangian algorithm for sparsity regularized estimation. *J. Mach. Learn. Res.* **2011**, *12*, 1537–1586.
53. ℓ_1-Ball. Available online: http://wittawat.com/software/l1lsmi/ (accessed on 7 December).
54. Duchi, J.; Shalev-Shwartz, S.; Singer, Y.; Chandra, T. Efficient Projections onto the ℓ_1-Ball for Learning in High Dimensions. In Proceedings of the 25th Annual International Conference on Machine Learning (ICML2008); McCallum, A., Roweis, S., Eds.; Helsinki, Finland, 5–9 July 2008; pp. 272–279.
55. Cook, R.D. *Regression Graphics: Ideas for Studying Regressions through Graphics*; Wiley: New York, NY, USA, 1998.
56. Li, K. Sliced inverse regression for dimension reduction. *J. Am. Stat. Assoc.* **1991**, *86*, 316–342.
57. Li, K. On principal hessian directions for data visualization and dimension reduction: another application of Stein's lemma. *J. Am. Stat. Assoc.* **1992**, *87*, 1025–1039.
58. Cook, R.D. SAVE: A method for dimension reduction and graphics in regression. *Commun. Stat. Theory* **2000**, *29*, 2109–2121.
59. Fukumizu, K.; Bach, F.R.; Jordan, M.I. Kernel dimension reduction in regression. *Ann. Stat.* **2009**, *37*, 1871–1905.
60. Golub, G.H.; Loan, C.F.V. *Matrix Computations*, 2nd ed.; Johns Hopkins University Press: Baltimore, MD, USA, 1989.

61. Nishimori, Y.; Akaho, S. Learning algorithms utilizing quasi-geodesic flows on the Stiefel manifold. *Neurocomputing* **2005**, *67*, 106–135.
62. Amari, S. Natural gradient works efficiently in learning. *Neural Comput.* **1998**, *10*, 251–276.
63. Edelman, A.; Arias, T.A.; Smith, S.T. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix. Anal. A.* **1998**, *20*, 303–353.
64. Patriksson, M. *Nonlinear Programming and Variational Inequality Problems*; Kluwer Academic: Dordrecht, The Netherlands, 1999.
65. Least-Squares Dimensionality Reduction (LSDR). Available online: http://sugiyama-www.cs.titech.ac.jp/~sugi/software/LSDR/ (accessed on 7 December 2012).
66. Epanechnikov, V. Nonparametric estimates of a multivariate probability density. *Theor. Probab. Appl.* **1969**, *14*, 153–158.
67. Sufficient Component Analysis (SCA). Available online: http://sugiyama-www.cs.titech.ac.jp/~yamada/sca.html (accessed on 7 December 2012).
68. Hotelling, H. Relations between two sets of variates. *Biometrika* **1936**, *28*, 321–377.
69. Becker, S.; Hinton, G.E. A self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature* **1992**, *355*, 161–163.
70. Fyfe, C.; Lai, P.L. Kernel and nonlinear canonical correlation analysis. *Int. J. Neural Syst.* **2000**, *10*, 365–377.
71. Akaho, S. A Kernel Method For Canonical Correlation Analysis. In Proceedings of the International Meeting of the Psychometric Society, Osaka, Japan, 15–19 July 2001.
72. Gestel, T.V.; Suykens, J.; Brabanter, J.D.; Moor, B.D.; Vandewalle, J. Kernel Canonical Correlation Analysis and Least Squares Support Vector Machines. In Proceedings of the International Conference on Artificial Neural Networks; Springer Berlin/Heidelberg, Germany, 2001; Volume 2130, *Lecture Notes in Computer Science*, pp. 384–389.
73. Breiman, L.; Friedman, J.H. Estimating optimal transformations for multiple regression and correlation. *J. Am. Stat. Assoc.* **1985**, *80*, 580–598.
74. Bach, F.; Jordan, M.I. Kernel independent component analysis. *J. Mach. Learn. Res.* **2002**, *3*, 1–48.
75. Yin, X. Canonical correlation analysis based on information theory. *J. Multivariate Anal.* **2004**, *91*, 161–176.
76. Härdle, W.; Müller, M.; Sperlich, S.; Werwatz, A. *Nonparametric and Semiparametric Models*; Springer: Berlin, Germany, 2004.
77. Least-Squares Canonical Dependency Analysis (LSCDA). Available online: http://www.bic.kyoto-u.ac.jp/pathway/krsym/software/LSCDA/index.html (accessed on 7 December 2012).
78. Hyvärinen, A.; Karhunen, J.; Oja, E. *Independent Component Analysis*; Wiley: New York, NY, USA, 2001.
79. Amari, S.; Cichocki, A.; Yang, H.H. A New Learning Algorithm for Blind Signal Separation. Advances in Neural Information Processing Systems 8; Touretzky, D.S., Mozer, M.C., Hasselmo, M.E., Eds.; The MIT Press: Cambridge, MA, USA, 1996; pp. 757–763.
80. Van Hulle, M.M. Sequential fixed-point ICA based on mutual information minimization. *Neural Comput.* **2008**, *20*, 1344–1365.

81. Jutten, C.; Herault, J. Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. *Signal Process.* **1991**, *24*, 1–10.
82. Hyvärinen, A. Fast and robust fixed-point algorithms for independent component analysis. *IEEE T. Neural Networ.* **1999**, *10*, 626.
83. Least-squares Independent Component Analysis. Available online: http://www.simplex.t.u-tokyo.ac.jp/~s-taiji/software/LICA/index.html (accessed on 7 December 2012).
84. Jebara, T. Kernelized Sorting, Permutation and Alignment for Minimum Volume PCA. In *Proceedings of the 17th Annual Conference on Learning Theory (COLT2004)*, Banff, Canada, 1–4 July 2004; pp. 609–623.
85. Gretton, A.; Bousquet, O.; Smola, A.; Schölkopf, B. Measuring Statistical Dependence with Hilbert-Schmidt Norms. In *Algorithmic Learning Theory*; Jain, S., Simon, H.U., Tomita, E., Eds.; Springer-Verlag: Berlin, Germany, 2005; Lecture Notes in Artificial Intelligence, pp. 63–77.
86. Quadrianto, N.; Smola, A.J.; Song, L.; Tuytelaars, T. Kernelized sorting. *IEEE Trans. Patt. Anal.* **2010**, *32*, 1809–1821.
87. Jagarlamudi, J.; Juarez, S.; Daumé III, H. Kernelized Sorting for Natural Language Processing. In Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI2010), Atlanta, Georgia, USA, 11–15 July 2010; pp. 1020–1025.
88. Kuhn, H.W. The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **1955**, *2*, 83–97.
89. Least-Squares Object Matching (LSOM). Available online: http://sugiyama-www.cs.titech.ac.jp/~yamada/lsom.html (accessed on 7 December 2012).
90. MacQueen, J.B. Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*; University of California Press: Berkeley, CA, USA, 1967; Vol. 1, pp. 281–297.
91. Girolami, M. Mercer kernel-based clustering in feature space. *IEEE Trans. Neural Networ.* **2002**, *13*, 780–784.
92. Shi, J.; Malik, J. Normalized cuts and image segmentation. *IEEE Trans. Patt. Anal.* **2000**, *22*, 888–905.
93. Ng, A.Y.; Jordan, M.I.; Weiss, Y. On Spectral Clustering: Analysis and An Algorithm. Advances in Neural Information Processing Systems 14; Dietterich, T.G., Becker, S., Ghahramani, Z., Eds.; MIT Press: Cambridge, MA, USA, 2002; pp. 849–856.
94. Fukunaga, K.; Hostetler, L.D. The estimation of the gradient of a density function, with application in pattern recognition. *IEEE Trans. Inf. Theory* **1975**, *21*, 32–40.
95. Carreira-Perpiñán, M.A. Fast Nonparametric Clustering with Gaussian Blurring Mean-Shift. In Proceedings of 23rd International Conference on Machine Learning (ICML2006); Cohen, W., Moore, A., Eds.; Pittsburgh, Pennsylvania, USA, 25–29 June 2006; pp. 153–160.
96. Xu, L.; Neufeld, J.; Larson, B.; Schuurmans, D. Maximum Margin Clustering. Advances in Neural Information Processing Systems 17; Saul, L.K., Weiss, Y., Bottou, L., Eds.; MIT Press: Cambridge, MA, USA, 2005; pp. 1537–1544.

97. Bach, F.; Harchaoui, Z. DIFFRAC: A Discriminative and Flexible Framework for Clustering. Advances in Neural Information Processing Systems 20; Platt, J.C., Koller, D., Singer, Y., Roweis, S., Eds.; MIT Press: Cambridge, MA, USA, 2008; pp. 49–56.
98. Song, L.; Smola, A.; Gretton, A.; Borgwardt, K. A Dependence Maximization View of Clustering. In Proceedings of the 24th Annual International Conference on Machine Learning (ICML2007); Ghahramani, Z., Ed.; Corvallis, Oregon, USA, 20–24 June 2007; pp. 815–822.
99. Faivishevsky, L.; Goldberger, J. A Nonparametric Information Theoretic Clustering Algorithm. In Proceedings of 27th International Conference on Machine Learning (ICML2010); Joachims, A.T., Fürnkranz, J., Eds.; Haifa, Israel, 21–24 June 2010; pp. 351–358.
100. Agakov, F.; Barber, D. Kernelized Infomax Clustering. Advances in Neural Information Processing Systems 18; Weiss, Y., Schölkopf, B., Platt, J., Eds.; MIT Press: Cambridge, MA, USA, 2006; pp. 17–24.
101. Gomes, R.; Krause, A.; Perona, P. Discriminative Clustering by Regularized Information Maximization. Advances in Neural Information Processing Systems 23; Lafferty, J., Williams, C.K.I., Zemel, R., Shawe-Taylor, J., Culotta, A., Eds.; 2010; pp. 766–774.
102. Zelnik-Manor, L.; Perona, P. Self-Tuning Spectral Clustering. Advances in Neural Information Processing Systems 17; Saul, L.K., Weiss, Y., Bottou, L., Eds.; MIT Press: Cambridge, MA, USA, 2005; pp. 1601–1608.
103. SMI-based Clustering (SMIC). Available online: http://sugiyama-www.cs.titech.ac.jp/~sugi/software/SMIC/ (accessed on 7 December 2012).
104. Horn, R.A.; Johnson, C.A. *Matrix Analysis*; Cambridge University Press: Cambridge, UK, 1985.
105. Pearl, J. *Causality: Models, Reasoning and Inference*; Cambridge University Press: New York, NY, USA, 2000.
106. Geiger, D.; Heckerman, D. Learning Gaussian Networks. In Proceedings of the 10th Annual Conference on Uncertainty in Artificial Intelligence (UAI1994), Seattle, Washington, USA, 29–31 July 1994; pp. 235–243.
107. Shimizu, S.; Hoyer, P.O.; Hyvärinen, A.; Kerminen, A.J. A linear non-gaussian acyclic model for causal discovery. *J. Mach. Learn. Res.* **2006**, *7*, 2003–2030.
108. Hoyer, P.O.; Janzing, D.; Mooij, J.M.; Peters, J.; Schölkopf, B. Nonlinear Causal Discovery with Additive Noise Models. Advances in Neural Information Processing Systems 21; Koller, D., Schuurmans, D., Bengio, Y., Bottou, L., Eds.; MIT Press: Cambridge, MA, USA, 2009; pp. 689–696.
109. Mooij, J.; Janzing, D.; Peters, J.; Schölkopf, B. Regression by Dependence Minimization and Its Application to Causal Inference in Additive Noise Models. In Proceedings of the 26th Annual International Conference on Machine Learning (ICML2009), Montreal, Canada Jun. 14–18, 2009; pp. 745–752.
110. Rasmussen, C.E.; Williams, C.K.I. *Gaussian Processes for Machine Learning*; MIT Press: Cambridge, MA, USA, 2006.
111. Least-Squares Independence Regression (LSIR). Availble online: http://sugiyama-www.cs.titech.ac.jp/~yamada/lsir.html (accessed on 7 December 2012).

112. Sugiyama, M.; Kawanabe, M. *Machine Learning in Non-Stationary Environments: Introduction to Covariate Shift Adaptation*; MIT Press: Cambridge, Massachusetts, USA, 2012.
113. Hido, S.; Tsuboi, Y.; Kashima, H.; Sugiyama, M.; Kanamori, T. Statistical outlier detection using direct density ratio estimation. *Knowl. Inf. Syst.* **2011**, *26*, 309–336.
114. Kawahara, Y.; Sugiyama, M. Sequential change-point detection based on direct density-ratio estimation. *Stat. Anal. Data Min.* **2012**, *5*, 114–127.
115. Liu, S.; Yamada, M.; Collier, N.; Sugiyama, M. Change-Point Detection in Time-Series Data by Relative Density-Ratio Estimation. In *Structural, Syntactic, and Statistical Pattern Recognition*; Gimel'farb, G., Hancock, E., Imiya, A., Kuijper, A., Kudo, M., Omachi, S., Windeatt, T., Yamada, K., Eds.; Springer: Berlin, Germany, 2012; Volume 7626, *Lecture Notes in Computer Science*, pp. 363–372.
116. Du Plessis, M.C.; Sugiyama, M. Semi-Supervised Learning of Class Balance under Class-Prior Change by Distribution Matching. In Proceedings of 29th International Conference on Machine Learning (ICML2012); Langford, J., Pineau, J., Eds.; Edinburgh, Scotland, 26 June–1 July 2012; pp. 823–830.
117. Sugiyama, M.; Suzuki, T.; Itoh, Y.; Kanamori, T.; Kimura, M. Least-squares two-sample test. *Neural Netw.* **2011**, *24*, 735–751.
118. Kanamori, T.; Suzuki, T.; Sugiyama, M. f-divergence estimation and two-sample homogeneity test under semiparametric density-ratio models. *IEEE Trans. Inf. Theory* **2012**, *58*, 708–720.
119. Sugiyama, M. Superfast-trainable multi-class probabilistic classifier by least-squares posterior fitting. *IEICE Trans. Inf. Syst.* **2010**, *E93-D*, 2690–2701.
120. Sugiyama, M.; Hachiya, H.; Yamada, M.; Simm, J.; Nam, H. Least-Squares Probabilistic Classifier: A Computationally Efficient Alternative to Kernel Logistic Regression. In Proceedings of International Workshop on Statistical Machine Learning for Speech Processing (IWSML2012), Kyoto, Japan, Mar. 31, 2012; pp. 1–10.
121. Sugiyama, M.; Takeuchi, I.; Suzuki, T.; Kanamori, T.; Hachiya, H.; Okanohara, D. Least-squares conditional density estimation. *IEICE Trans. Inf. Syst.* **2010**, *E93-D*, 583–594.
122. Sugiyama, M.; Kawanabe, M.; Chui, P.L. Dimensionality reduction for density ratio estimation in high-dimensional spaces. *Neural Netw.* **2010**, *23*, 44–59.
123. Sugiyama, M.; Yamada, M.; von Bünau, P.; Suzuki, T.; Kanamori, T.; Kawanabe, M. Direct density-ratio estimation with dimensionality reduction via least-squares hetero-distributional subspace search. *Neural Netw.* **2011**, *24*, 183–198.
124. Yamada, M.; Sugiyama, M. Direct Density-Ratio Estimation with Dimensionality Reduction via Hetero-Distributional Subspace Analysis. In Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI2011); The AAAI Press: San Francisco, California, USA, 2011; pp. 549–554.
125. Yamada, M.; Suzuki, T.; Kanamori, T.; Hachiya, H.; Sugiyama, M. Relative Density-Ratio Estimation for Robust Distribution Comparison. Advances in Neural Information Processing Systems 24; Shawe-Taylor, J., Zemel, R.S., Bartlett, P., Pereira, F.C.N., Weinberger, K.Q., Eds.; 2011; pp. 594–602.

126. Sugiyama, M.; Suzuki, T.; Kanamori, T.; Du Plessis, M.C.; Liu, S.; Takeuchi, I. Density-Difference Estimation. Advances in Neural Information Processing Systems 25, 2012.
127. Software. Available online: http://sugiyama-www.cs.titech.ac.jp/~sugi/software/ (accessed on 7 December 2012).