Canonical Dependency Analysis based on Squared-loss Mutual Information

Masayuki Karasuyama
Institute for Chemical Research, Kyoto University, Japan.
karasuyama@kuicr.kyoto-u.ac.jp

Masashi Sugiyama
Department of Computer Science, Tokyo Institute of Technology, Japan.
sugi@cs.titech.ac.jp
http://sugiyama-www.cs.titech.ac.jp/~sugi

Abstract

Canonical correlation analysis (CCA) is a classical dimensionality reduction technique for two sets of variables that iteratively finds projection directions with maximum correlation. Although CCA is still in vital use in many practical application areas, recent real-world data often contain more complicated non-linear correlations that can not be properly captured by classical CCA. In this paper, we thus propose an extension of CCA that can effectively capture such complicated non-linear correlations through statistical dependency maximization. The proposed method, which we call *least-squares canonical dependency analysis* (LSCDA), is based on a squared-loss variant of mutual information, and it has various useful properties besides its ability to capture higher-order correlations, for example, it can simultaneously find multiple projection directions (i.e., subspaces), it does not involve density estimation, and it is equipped with a model selection strategy. We demonstrate the usefulness of LSCDA through various experiments on artificial and real-world datasets.

Keywords

Canonical Correlation Analysis, Squared-loss Mutual Information, Direct Density-ratio Estimation

1 Introduction

Canonical correlation analysis (CCA) (Hotelling, 1936) is a classical dimensionality reduction technique for two data sources, which finds projection directions with maximum correlation. CCA has been successfully applied in various fields such as neuroscience (Becker, 1996; Becker & Hinton, 1992; Favorov & Ryder, 2004), econometrics (Bossaerts, 1988; Vinod, 1968), psychometrics (McKeon, 1967), meteorology (Storch & Zwiers, 2002), bioinformatics (Gumus et al., 2012; Naylor et al., 2010; Vert & Kanehisa, 2003; Yamanishi et al., 2003), and information retrieval (Hardoon et al., 2004; Li & Shawe-Taylor, 2006; Vinokourov et al., 2003).

Although CCA has been originally developed as an unsupervised learning method, it is also closely related to supervised tasks such as classification. Indeed, in some special cases, CCA is equivalent to Fisher's *Linear Discriminant Analysis* (LDA) (Bartlett, 1938). Such CCA-based classification methods have been widely studied (Farquhar et al., 2005; Kursun et al., 2011; Rai & Daume, 2009; Sun et al., 2011), which incorporate class information in various ways to learn projections that are informative for discrimination. The usefulness of these approaches have been demonstrated in various modern pattern recognition problems such as *multi-label classification* (Rai & Daume, 2009; Sun et al., 2011) that utilizes correlations among labels for improving classification performance.

However, since classical 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 developed. A simple approach uses neural networks for handling non-linear projection (Becker, 1996; Becker & Hinton, 1992; Favorov & Ryder, 2004; Fyfe & Lai, 2000), but neural networks are prone to local optima. Another approach first non-linearly transforms data samples into feature spaces and then apply linear CCA (Akaho, 2001; Gestel et al., 2001; Melzer et al., 2001). 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. Since *reproducing kernel Hilbert spaces* (RKHSs) (Aronszajn, 1950) are used as feature spaces, this approach is called *kernel CCA* (KCCA). Alternating regression is another possible way of flexibly finding dependency, which is closely related to CCA (Branco et al., 2005; Breiman & Friedman, 1985; Kursun & Favorov, 2010; Wold, 1966). A typical approach is *Alternating Conditional Expectation* (ACE) (Breiman & Friedman, 1985), which estimates transformations for two variables alternately by minimizing the squared error between transformed variables. These non-linear variants of CCA are highly flexible, but obtained results are often difficult to interpret due to non-linearity.

The above non-linear CCA approaches can be regarded as capturing correlations along non-linear projection directions. Another extension of CCA, which we call *canonical dependency analysis* (CDA), captures higher-order correlations under linear projections. It was shown in Bach & Jordan (2002) that KCCA with a *universal RKHS* (Steinwart, 2001) such as the Gaussian RKHS allows efficient detection of higher-order correlations. However, the choice of universal RKHSs affects the practical performance, and there

is no systematic method to choose a suitable RKHS (Fukumizu et al., 2009). Another approach to higher-order CCA called *informational CCA* (ICCA) (Yin, 2004) uses *mutual information* (MI) (Cover & Thomas, 2006; Shannon, 1948) as a dependency measure, where MI is estimated via kernel density estimation (KDE) (Silverman, 1986). Since KDE is equipped with systematic model selection strategies (Härdle et al., 2004; Scott, 1992; Silverman, 1986), ICCA is practically more preferable than the KCCA-based CDA method.

In the ICCA method, one-dimensional projection directions are found in an iterative manner. However, it would be more powerful if multi-dimensional projection directions (i.e., subspaces) were directly found in CDA. In the experiments in Section 4, we show that directly estimating multi-dimensional projections compares favorably with iteratively estimating one-dimensional projection directions, because the iterative approach often gets trapped into poor local optima. However, ICCA may not be reliable in such a multi-dimensional scenario since it involves the ratio of estimated densities which tends to produce large estimation error.

The purpose of this paper is to give a novel CDA method that is equipped with model selection and that is reliable even when multi-dimensional projection directions are searched. Our method, which we call *least-squares CDA* (LSCDA), uses a squared-loss MI (SMI) as a dependency measure. As ordinary MI, SMI also includes the ratio of probability densities and thus its accurate estimation is challenging. In LSCDA, we use an analytic SMI estimator called *least-squares MI* (LSMI) (Suzuki et al., 2009) that directly estimates the density ratio without going through density estimation (Kanamori et al., 2009). Thus, LSMI is more reliable than an estimator based on KDE (see Suzuki & Sugiyama (2010) for theoretical convergence analysis of the LSMI estimator). Possible benefits of our approach can be summarized as follows:

- It can capture higher-order correlations under linear projection.
- It can accurately estimate multi-dimensional projection matrices.
- It does not involve density estimation.
- It is equipped with a model selection strategy.

As examples of real-world applications, we apply our approach to multi-label classification problems in image annotation and audio tagging. In multi-label classification, extracting features that have strong dependency on labels and that incorporate correlations among labels is desirable. Through experiments, we demonstrate that the proposed method improves prediction performance of a subsequent classifier in practical multi-label classification scenarios.

The remainder of this paper is structured as follows: In Section 2, we formulate our LSCDA algorithm using SMI as a dependency measure. Section 3 describes relationships with several existing approaches. In Section 4, we present our experimental results obtained by a variety of datasets including artificial and real-world datasets, demonstrating advantages of the proposed approach over other methods. Finally, Section 5 concludes the paper.

2 Canonical Dependency Analysis via SMI Estimation

In this section, we describe our novel algorithm for finding statistical dependency between a pair of variables.

2.1 Problem Formulation

Let $\mathcal{X} \subset \mathbb{R}^m$ be the domain of $\boldsymbol{x}$ and $\mathcal{Y} \subset \mathbb{R}^n$ be the domain of $\boldsymbol{y}$. Suppose we are given ℓ independent and identically distributed (i.i.d.) paired samples,

$$\mathcal{S} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i) \mid \boldsymbol{x}_i \in \mathcal{X}, \boldsymbol{y}_i \in \mathcal{Y}, i = 1, \ldots, \ell\},$$

drawn from a joint distribution with density $p_{\mathrm{xy}}(\boldsymbol{x}, \boldsymbol{y})$.

In this paper, we would like to find the low-dimensional subspaces of $\boldsymbol{x}$ and $\boldsymbol{y}$ in which projections are maximally dependent on each other. Here, we focus on linear dimension reduction:

$$\boldsymbol{u} = \boldsymbol{U}\boldsymbol{x} \quad \text{and} \quad \boldsymbol{v} = \boldsymbol{V}\boldsymbol{y},$$

where $\boldsymbol{U} \in \mathbb{R}^{p \times m}$ and $\boldsymbol{V} \in \mathbb{R}^{q \times n}$ are transformation matrices with p and q being the projection dimensions for $\boldsymbol{u}$ and $\boldsymbol{v}$, respectively. We assume that $\boldsymbol{U}$ and $\boldsymbol{V}$ belong to the *Stiefel manifolds* $\mathbb{S}_p^m(\mathbb{R})$ and $\mathbb{S}_q^n(\mathbb{R})$, respectively:

$$\mathbb{S}_p^m(\mathbb{R}) = \{\boldsymbol{U} \in \mathbb{R}^{p \times m} \mid \boldsymbol{U}\boldsymbol{U}^\top = \boldsymbol{I}_p\},$$
$$\mathbb{S}_q^n(\mathbb{R}) = \{\boldsymbol{V} \in \mathbb{R}^{q \times n} \mid \boldsymbol{V}\boldsymbol{V}^\top = \boldsymbol{I}_q\},$$

where $\boldsymbol{I}_p$ is the p-dimensional identity matrix and $\boldsymbol{I}_q$ is the q-dimensional identity matrix. Hereafter, we assume that the projection dimensions p and q are known; practically, p and q may be chosen by cross-validation. We write $\boldsymbol{u}_i = \boldsymbol{U}\boldsymbol{x}_i$ and $\boldsymbol{v}_i = \boldsymbol{V}\boldsymbol{y}_i$ for $i = 1, \ldots, \ell$.

Our goal is to find the transformation matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ such that the dependency between $\boldsymbol{u}$ and $\boldsymbol{v}$ is maximized. To this end, we employ *squared-loss mutual information* (SMI) as our dependency measure:

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

where $p_{\mathrm{uv}}(\boldsymbol{u}, \boldsymbol{v})$ is the joint density of $\boldsymbol{u}$ and $\boldsymbol{v}$, and $p_{\mathrm{u}}(\boldsymbol{u})$ and $p_{\mathrm{v}}(\boldsymbol{v})$ are the marginal densities of $\boldsymbol{u}$ and $\boldsymbol{v}$, respectively. SMI is the *Pearson divergence* (Pearson, 1900) from $p_{\mathrm{uv}}(\boldsymbol{u}, \boldsymbol{v})$ to $p_{\mathrm{u}}(\boldsymbol{u}) p_{\mathrm{v}}(\boldsymbol{v})$, while the ordinary MI is the *Kullback-Leibler divergence* (Kullback & Leibler, 1951) from $p_{\mathrm{uv}}(\boldsymbol{u}, \boldsymbol{v})$ to $p_{\mathrm{u}}(\boldsymbol{u}) p_{\mathrm{v}}(\boldsymbol{v})$:

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

Both of the Pearson divergence and the Kullback-Leibler divergence belong to the class of f-divergences (Ali & Silvey, 1966; Csiszár, 1967) and they share similar properties. For example, SMI and MI are nonnegative and take 0 if and only if $p_{\mathrm{uv}}(\boldsymbol{u},\boldsymbol{v}) = p_{\mathrm{u}}(\boldsymbol{u})p_{\mathrm{v}}(\boldsymbol{v})$. Therefore, SMI allows us to evaluate the statistical independence between $\boldsymbol{u}$ and $\boldsymbol{v}$.

Now we want to find matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ that maximize SMI. However, we can not directly maximize SMI since densities $p_{\mathrm{vu}}(\boldsymbol{u},\boldsymbol{v})$, $p_{\mathrm{u}}(\boldsymbol{u})$, and $p_{\mathrm{v}}(\boldsymbol{v})$ are usually unknown. Below, we utilize an SMI estimator called *least-squares mutual information* (LSMI) (Suzuki et al., 2009), which involves *direct density-ratio estimation* (Kanamori et al., 2009) instead of density estimation. LSMI was shown to possesses superior convergence properties (Suzuki & Sugiyama, 2010).

2.2 SMI Approximation via Direct Density-ratio Estimation

Here, we review LSMI (Suzuki et al., 2009). A key idea of LSMI is to directly estimate the *density ratio*,

$$g^*(\boldsymbol{u},\boldsymbol{v}) := \frac{p_{\mathrm{uv}}(\boldsymbol{u},\boldsymbol{v})}{p_{\mathrm{u}}(\boldsymbol{u})p_{\mathrm{v}}(\boldsymbol{v})},$$

without estimating each density. We model the density ratio function $g^*(\boldsymbol{u},\boldsymbol{v})$ using the following linear model:

$$g(\boldsymbol{u},\boldsymbol{v}) := \boldsymbol{\alpha}^\top \boldsymbol{\varphi}(\boldsymbol{u},\boldsymbol{v}),$$

where $^\top$ denotes the transpose of a matrix or a vector,

$$\boldsymbol{\alpha} := (\alpha_1,\ldots,\alpha_b)^\top$$

are parameters to be learned from samples, and

$$\boldsymbol{\varphi}(\boldsymbol{u},\boldsymbol{v}) := (\varphi_1(\boldsymbol{u},\boldsymbol{v}),\ldots,\varphi_b(\boldsymbol{u},\boldsymbol{v}))^\top$$

is a basis function such that $\boldsymbol{\varphi}(\boldsymbol{u},\boldsymbol{v}) \geq \boldsymbol{0}_b$ for all $\boldsymbol{u}$ and $\boldsymbol{v}$. $\boldsymbol{0}_b$ denotes the b-dimensional vector with all zeros, and the inequality for a vector is applied in the element-wise manner.

Note that the number of basis functions b is not necessarily a constant; it can depend on the number of samples ℓ. Similarly, the basis function $\boldsymbol{\varphi}(\boldsymbol{u},\boldsymbol{v})$ could be dependent on the samples $\mathcal{S}$. This means that the *kernel* models (i.e., $b = \ell$ and $\varphi_i(\boldsymbol{u},\boldsymbol{v})$ is a kernel function "centered" at $\{(\boldsymbol{u}_i,\boldsymbol{v}_i)\}_{i=1}^{\ell}$) are also included in the above formulation. In Section 2.3, we explain how the basis functions $\boldsymbol{\varphi}(\boldsymbol{u},\boldsymbol{v})$ are practically chosen.

We estimate the parameter $\boldsymbol{\alpha}$ in $g(\boldsymbol{u},\boldsymbol{v})$ so that the following squared error J_0 is minimized:

$$\begin{aligned} J_0(\boldsymbol{\alpha}) &:= \frac{1}{2}\iint (g(\boldsymbol{u},\boldsymbol{v}) - g^*(\boldsymbol{u},\boldsymbol{v}))^2 p_{\mathrm{u}}(\boldsymbol{u})p_{\mathrm{v}}(\boldsymbol{v})\mathrm{d}\boldsymbol{u}\mathrm{d}\boldsymbol{v} \\ &= \frac{1}{2}\iint g(\boldsymbol{u},\boldsymbol{v})^2 p_{\mathrm{u}}(\boldsymbol{u})p_{\mathrm{v}}(\boldsymbol{v})\mathrm{d}\boldsymbol{u}\mathrm{d}\boldsymbol{v} - \iint g(\boldsymbol{u},\boldsymbol{v})p_{\mathrm{uv}}(\boldsymbol{u},\boldsymbol{v})\mathrm{d}\boldsymbol{u}\mathrm{d}\boldsymbol{v} + C, \end{aligned} \tag{1}$$

where

$$C := \frac{1}{2} \iint g^*(\boldsymbol{u}, \boldsymbol{v})^2 p_{\mathrm{u}}(\boldsymbol{u}) p_{\mathrm{v}}(\boldsymbol{v}) \mathrm{d}\boldsymbol{u} \mathrm{d}\boldsymbol{v}$$

is a constant and therefore can be safely ignored by assuming $C < \infty$. Let us denote the first two terms of (1) by J:

$$J(\boldsymbol{\alpha}) := J_0(\boldsymbol{\alpha}) - C = \frac{1}{2} \boldsymbol{\alpha}^\top \boldsymbol{H} \boldsymbol{\alpha} - \boldsymbol{h}^\top \boldsymbol{\alpha}, \tag{2}$$

where

$$\boldsymbol{H} := \iint \boldsymbol{\varphi}(\boldsymbol{u}, \boldsymbol{v}) \boldsymbol{\varphi}(\boldsymbol{u}, \boldsymbol{v})^\top p_{\mathrm{u}}(\boldsymbol{u}) p_{\mathrm{v}}(\boldsymbol{v}) \mathrm{d}\boldsymbol{u} \mathrm{d}\boldsymbol{v},$$
$$\boldsymbol{h} := \iint \boldsymbol{\varphi}(\boldsymbol{u}, \boldsymbol{v}) p_{\mathrm{uv}}(\boldsymbol{u}, \boldsymbol{v}) \mathrm{d}\boldsymbol{u} \mathrm{d}\boldsymbol{v}.$$

Approximating the expectations in $\boldsymbol{H}$ and $\boldsymbol{h}$ by empirical averages, we obtain the following optimization problem:

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

where we included $\frac{\lambda}{2} \boldsymbol{\alpha}^\top \boldsymbol{\alpha}$ ($\lambda > 0$) for regularization purposes, and

$$\widehat{\boldsymbol{H}} := \frac{1}{\ell^2} \sum_{i,i'=1}^{\ell} \boldsymbol{\varphi}(\boldsymbol{u}_i, \boldsymbol{v}_{i'}) \boldsymbol{\varphi}(\boldsymbol{u}_i, \boldsymbol{v}_{i'})^\top,$$

$$\widehat{\boldsymbol{h}} := \frac{1}{\ell} \sum_{i=1}^{\ell} \boldsymbol{\varphi}(\boldsymbol{u}_i, \boldsymbol{v}_i).$$

Taking the derivative of the above objective function and set it as zero, we obtain:

$$\widehat{\boldsymbol{\alpha}} = (\widehat{\boldsymbol{H}} + \lambda \boldsymbol{I}_b)^{-1} \widehat{\boldsymbol{h}},$$

where $\boldsymbol{I}_b$ is the b-dimensional identity matrix. Thus, the solution can be computed *analytically* by solving a system of linear equations. Then an analytical approximation of SMI called *least-squares mutual information* (LSMI) is given as

$$\widehat{\mathrm{SMI}} := \widehat{\boldsymbol{h}}^\top \widehat{\boldsymbol{\alpha}} - \frac{1}{2} \widehat{\boldsymbol{\alpha}}^\top \widehat{\boldsymbol{H}} \widehat{\boldsymbol{\alpha}} - \frac{1}{2},$$

which is obtained based on the following expression of SMI (Suzuki & Sugiyama, 2010):

$$\mathrm{SMI} = \iint g^*(\boldsymbol{u}, \boldsymbol{v}) p_{\mathrm{uv}}(\boldsymbol{u}, \boldsymbol{v}) \mathrm{d}\boldsymbol{u} \mathrm{d}\boldsymbol{v} - \frac{1}{2} \iint g^*(\boldsymbol{u}, \boldsymbol{v})^2 p_{\mathrm{u}}(\boldsymbol{u}) p_{\mathrm{v}}(\boldsymbol{v}) \mathrm{d}\boldsymbol{u} \mathrm{d}\boldsymbol{v} - \frac{1}{2}.$$

Using $J(\boldsymbol{\alpha})$, SMI can be written as follows:

$$\mathrm{SMI} = -\inf_{\boldsymbol{\alpha}} J(\boldsymbol{\alpha}) - \frac{1}{2}$$

Here, we assume that the true density ratio g^* is contained in the model, i.e., $g^* \in \{g(\boldsymbol{u}, \boldsymbol{v}) \mid \boldsymbol{\alpha} \in \mathbb{R}^b\}$. Therefore, computing SMI is reduced to finding minimizer of $J(\boldsymbol{\alpha})$ (see Suzuki & Sugiyama, 2010, for detail).

2.3 Model Selection and Basis Function Design

The performance of LSMI depends on the choice of basis functions and the regularization parameter. We can choose them based on cross-validation (CV) with respect to the error criterion (2). For example, in the case of K-fold CV, we divide the samples $\mathcal{S} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{\ell}$ into K disjoint subsets $\{\mathcal{S}_k\}_{k=1}^{K}$ of (approximately) the same size. Then, an estimator $\widehat{\boldsymbol{\alpha}}_k$ is obtained using $\{\mathcal{S}_j\}_{j\neq k}$ (i.e., without $\mathcal{S}_k$) and the approximation error for the hold-out samples $\mathcal{S}_k$ is obtained as

$$\widehat{J}^{(K-\mathrm{CV})} := \frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{2}\widehat{\boldsymbol{\alpha}}_k^\top \widehat{\boldsymbol{H}}_k \widehat{\boldsymbol{\alpha}}_k - \widehat{\boldsymbol{h}}_k^\top \widehat{\boldsymbol{\alpha}}_k\right), \tag{3}$$

where, for the set $\mathcal{S}_k = \{(\boldsymbol{x}_{s_i}, y_{s_i})\}_{i=1}^{\ell_k}$,

$$\widehat{\boldsymbol{H}}_k := \frac{1}{\ell_k^2}\sum_{i,i'=1}^{\ell_k} \boldsymbol{\varphi}(\boldsymbol{u}_{s_i}, \boldsymbol{v}_{s_{i'}})\boldsymbol{\varphi}(\boldsymbol{u}_{s_i}, \boldsymbol{v}_{s_{i'}})^\top,$$

$$\widehat{\boldsymbol{h}}_k := \frac{1}{\ell_k}\sum_{i=1}^{\ell_k} \boldsymbol{\varphi}(\boldsymbol{u}_{s_i}, \boldsymbol{v}_{s_i}).$$

We compute $\widehat{J}^{(K-\mathrm{CV})}$ for each model candidate, and then choose the best one that minimizes $\widehat{J}^{(K-\mathrm{CV})}$.

To exploit the above CV procedure, we have to prepare candidates of basis functions. Here, we use the following *product kernel* as basis functions:

$$\varphi_k(\boldsymbol{u}, \boldsymbol{v}) = \phi_k^{\mathrm{u}}(\boldsymbol{u})\phi_k^{\mathrm{v}}(\boldsymbol{v}),$$

since the number of kernel evaluation when computing $\widehat{H}_{k,k'}$ is reduced from ℓ^2 to 2ℓ:

$$\widehat{H}_{k,k'} = \frac{1}{\ell^2}\left(\sum_{i=1}^{\ell}\phi_k^{\mathrm{u}}(\boldsymbol{u}_i)\phi_{k'}^{\mathrm{u}}(\boldsymbol{u}_i)\right)\left(\sum_{i=1}^{\ell}\phi_k^{\mathrm{v}}(\boldsymbol{v}_i)\phi_{k'}^{\mathrm{v}}(\boldsymbol{v}_i)\right).$$

We use the Gaussian kernel as the "base" kernels:

$$\phi_k^{\mathrm{u}}(\boldsymbol{u}) := \exp\left(-\frac{\|\boldsymbol{u}-\boldsymbol{u}_k\|_2^2}{2\sigma^2}\right),$$

$$\phi_k^{\mathrm{v}}(\boldsymbol{v}) := \exp\left(-\frac{\|\boldsymbol{v}-\boldsymbol{v}_k\|_2^2}{2\sigma^2}\right),$$

where $\{(\boldsymbol{u}_k, \boldsymbol{v}_k)\}_{k=1}^{b}$ are Gaussian centers randomly chosen from $\{(\boldsymbol{u}_i, \boldsymbol{v}_i)\}_{i=1}^{\ell}$ and $\|\cdot\|_2$ denotes the ℓ_2-norm.

In the experiments, we fix the number of basis functions at

$$b = \min(200, \ell), \tag{4}$$

and choose the Gaussian width σ and the regularization parameter λ by CV with grid search.

1: Initialize $\boldsymbol{U}$ and $\boldsymbol{V}$.
2: Optimize Gaussian width σ and regularization parameter λ by CV (explained in Section 2.3).
3: Update $\boldsymbol{U}$ and $\boldsymbol{V}$ such that $\widehat{\mathrm{SMI}}$ is maximized.
4: Repeat 2. and 3. until $\boldsymbol{U}$ and $\boldsymbol{V}$ converge.

Figure 1: The LSCDA algorithm.

2.4 Least-squares Canonical Dependency Analysis (LSCDA)

Given the SMI estimator $\widehat{\mathrm{SMI}}$, we maximize it with respect to $\boldsymbol{U}$ and $\boldsymbol{V}$:

$$\begin{aligned} \underset{\boldsymbol{U}\in\mathbb{R}^{p\times m},\boldsymbol{V}\in\mathbb{R}^{q\times n}}{\operatorname{argmax}} \quad & \widehat{\mathrm{SMI}}(\boldsymbol{U},\boldsymbol{V}) \\ \text{s.t.} \quad & \boldsymbol{U}\boldsymbol{U}^\top = \boldsymbol{I}_p \ \text{ and } \ \boldsymbol{V}\boldsymbol{V}^\top = \boldsymbol{I}_q. \end{aligned} \tag{5}$$

We call this approach to finding $\boldsymbol{U}$ and $\boldsymbol{V}$ *least-squares canonical dependency analysis* (LSCDA). The entire algorithm of LSCDA is summarized in Figure 1. We can employ various optimization techniques to obtain a solution of the above optimization problem. Below, we show several possibilities.

2.4.1 Plain Gradient Algorithm

A plain gradient ascent technique updates $\boldsymbol{U}$ and $\boldsymbol{V}$ by the following forms:

$$\boldsymbol{U} \leftarrow \boldsymbol{U} + t\frac{\partial\widehat{\mathrm{SMI}}}{\partial\boldsymbol{U}} \ \text{ and } \ \boldsymbol{V} \leftarrow \boldsymbol{V} + t\frac{\partial\widehat{\mathrm{SMI}}}{\partial\boldsymbol{V}},$$

where $t>0$ is a step size chosen by a line search method such as *Armijo's rule* (Nocedal & Wright, 1999). The gradients are given by

$$\begin{aligned} \frac{\partial\widehat{\mathrm{SMI}}}{\partial U_{k,k'}} &= \frac{\partial\widehat{\boldsymbol{h}}}{\partial U_{k,k'}}(2\widehat{\boldsymbol{\alpha}}-\widehat{\boldsymbol{\beta}}) - \widehat{\boldsymbol{\alpha}}^\top\frac{\partial\widehat{\boldsymbol{H}}}{\partial U_{k,k'}}(\frac{3}{2}\widehat{\boldsymbol{\alpha}}-\widehat{\boldsymbol{\beta}}), \\ \frac{\partial\widehat{\mathrm{SMI}}}{\partial V_{k,k'}} &= \frac{\partial\widehat{\boldsymbol{h}}}{\partial V_{k,k'}}(2\widehat{\boldsymbol{\alpha}}-\widehat{\boldsymbol{\beta}}) - \widehat{\boldsymbol{\alpha}}^\top\frac{\partial\widehat{\boldsymbol{H}}}{\partial V_{k,k'}}(\frac{3}{2}\widehat{\boldsymbol{\alpha}}-\widehat{\boldsymbol{\beta}}), \end{aligned}$$

where $\widehat{\boldsymbol{\beta}} := (\widehat{\boldsymbol{H}}+\lambda\boldsymbol{I}_b)^{-1}\widehat{\boldsymbol{H}}\widehat{\boldsymbol{\alpha}}$. A naive gradient ascent approach does not take into account the orthonormality constraints $\boldsymbol{U}\boldsymbol{U}^\top = \boldsymbol{I}_p$ and $\boldsymbol{V}\boldsymbol{V}^\top = \boldsymbol{I}_q$. Thus, we orthonormalize $\boldsymbol{U}$ and $\boldsymbol{V}$ after each update, e.g., using the *Gram-Schmidt process* (Golub & Van Loan, 1996). However, this may be rather time-consuming.

2.4.2 Sequential Quadratic Programming

We can use a *sequential quadratic programming* (SQP) (Nocedal & Wright, 1999) technique to efficiently handle the orthonormality constraints. SQP is one of the standard

optimization techniques for a non-linear objective function and nonlinear constraints. In the MI-based CCA method (Yin, 2004), SQP is used to optimize MI under similar constraints to (5). This approach is also applicable to our optimization problem (5). The SQP method iteratively solves a quadratic programming subproblem which locally approximates the true optimization problem.

Let

$$\boldsymbol{z} = \begin{bmatrix} \mathrm{vec}(\boldsymbol{U}^\top) \\ \mathrm{vec}(\boldsymbol{V}^\top) \end{bmatrix},$$

where the "vec" operator vectorizes a matrix by concatenating all of its columns. At the k-th iteration, we "model" problem (5) as the following quadratic program:

$$\begin{aligned} (\boldsymbol{U}'^{(k+1)}, \boldsymbol{V}'^{(k+1)}) := \underset{\boldsymbol{U}\in\mathbb{R}^{p\times m}, \boldsymbol{V}\in\mathbb{R}^{q\times n}}{\mathrm{argmax}} \quad & \boldsymbol{z}^\top \boldsymbol{W}^{(k)} \boldsymbol{z} + \left.\frac{\partial \widehat{\mathrm{SMI}}}{\partial \boldsymbol{z}}\right|_{\boldsymbol{z}=\boldsymbol{z}^{(k)}}^\top \boldsymbol{z} \\ \text{s.t.} \quad & \boldsymbol{U}^{(k)}\boldsymbol{U}^\top = \boldsymbol{I}_p \ \text{ and } \ \boldsymbol{V}^{(k)}\boldsymbol{V}^\top = \boldsymbol{I}_q, \end{aligned}$$

where $\boldsymbol{z}^{(k)}$, $\boldsymbol{U}^{(k)}$, and $\boldsymbol{V}^{(k)}$ are the estimates obtained at the k-th iteration and $\boldsymbol{W}^{(k)}$ is a (usually negative semi-definite) approximation of the Hessian matrix of the Lagrangian function of (5). The solution of this problem provides search directions for the $(k+1)$-th iteration:

$$\begin{aligned} \boldsymbol{U}^{(k+1)} &\leftarrow \boldsymbol{U}^{(k)} + t\left(\boldsymbol{U}'^{(k+1)} - \boldsymbol{U}^{(k)}\right), \\ \boldsymbol{V}^{(k+1)} &\leftarrow \boldsymbol{V}^{(k)} + t\left(\boldsymbol{V}'^{(k+1)} - \boldsymbol{V}^{(k)}\right), \end{aligned}$$

where t is a step size. t is chosen so that the following *merit function* is maximized:

$$\widehat{\mathrm{SMI}} - \mu\left(\|\boldsymbol{U}\boldsymbol{U}^\top - \boldsymbol{I}_p\|_{1,1} + \|\boldsymbol{V}\boldsymbol{V}^\top - \boldsymbol{I}_q\|_{1,1}\right),$$

where $\|\cdot\|_{1,1}$ denotes the sum of absolute values of all elements of a matrix and $\mu > 0$ is the penalty parameter which penalizes deviations from the feasible region of the original problem (5). We can easily determine μ so that the increase of the merit function is guaranteed (Nocedal & Wright, 1999). An implementation of SQP is available, e.g., "*fmincon*" in the MATLAB® optimization toolbox. However, since this is a general-purpose optimizer, it is sometimes inefficient in solving a problem with specific structure.

2.4.3 Natural Gradient Algorithm

Another approach to efficiently handling the orthonormality constraints in (5) is a *natural gradient algorithm* (Amari, 1998). Due to the orthonormality constraints $\boldsymbol{U}\boldsymbol{U}^\top = \boldsymbol{I}_p$ and $\boldsymbol{V}\boldsymbol{V}^\top = \boldsymbol{I}_q$, the matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ should belong to the *Stiefel manifolds* $\mathbb{S}_p^m(\mathbb{R})$ and $\mathbb{S}_q^n(\mathbb{R})$, respectively. The ordinary gradient gives the steepest direction in the Euclidean space, whereas the natural gradient gives the steepest direction on a manifold. The natural gradient is the projection of the ordinary gradient to the tangent space of its manifold.

If the tangent space is equipped with the canonical metric $\langle \boldsymbol{G}_1, \boldsymbol{G}_2 \rangle = \frac{1}{2}\mathrm{tr}(\boldsymbol{G}_1^\top \boldsymbol{G}_2)$, the natural gradients for $\boldsymbol{U}$ and $\boldsymbol{V}$ are given by

$$\nabla_{\boldsymbol{U}} \widehat{\mathrm{SMI}} = \frac{1}{2}\left(\frac{\partial \widehat{\mathrm{SMI}}}{\partial \boldsymbol{U}} - \boldsymbol{U}\frac{\partial \widehat{\mathrm{SMI}}}{\partial \boldsymbol{U}}^\top \boldsymbol{U}\right),$$
$$\nabla_{\boldsymbol{V}} \widehat{\mathrm{SMI}} = \frac{1}{2}\left(\frac{\partial \widehat{\mathrm{SMI}}}{\partial \boldsymbol{V}} - \boldsymbol{V}\frac{\partial \widehat{\mathrm{SMI}}}{\partial \boldsymbol{V}}^\top \boldsymbol{V}\right).$$

Then the *geodesics* from $\boldsymbol{U}$ and $\boldsymbol{V}$ to the directions of the natural gradients $\nabla_{\boldsymbol{U}} \widehat{\mathrm{SMI}}$ and $\nabla_{\boldsymbol{V}} \widehat{\mathrm{SMI}}$ over $\mathbb{S}_p^m(\mathbb{R})$ and $\mathbb{S}_q^n(\mathbb{R})$ can be represented as

$$\boldsymbol{U}_t := \boldsymbol{U}\exp\left(t(\boldsymbol{U}^\top \frac{\partial \widehat{\mathrm{SMI}}}{\partial \boldsymbol{U}} - \frac{\partial \widehat{\mathrm{SMI}}}{\partial \boldsymbol{U}}^\top \boldsymbol{U})\right),$$
$$\boldsymbol{V}_t := \boldsymbol{V}\exp\left(t(\boldsymbol{V}^\top \frac{\partial \widehat{\mathrm{SMI}}}{\partial \boldsymbol{V}} - \frac{\partial \widehat{\mathrm{SMI}}}{\partial \boldsymbol{V}}^\top \boldsymbol{V})\right),$$

where "exp" for a matrix denotes the *matrix exponential* and $t > 0$ is a step size which can be chosen by a line search method such as Armijo's rule. See (Nishimori & Akaho, 2005) for more details of geometric structure of the Stiefel manifold.

In the experiments, we use the natural gradient method for optimization. Our MATLAB® is available from `http://www.bic.kyoto-u.ac.jp/pathway/krsym/software/LSCDA/index.html`.

3 Relation to Existing Methods

In this section, we review existing methods for analyzing paired samples $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{\ell}$, and discuss the relationship to the proposed method.

3.1 Canonical Correlation Analysis (CCA)

CCA iteratively finds projection directions $\boldsymbol{g}$ and $\boldsymbol{h}$ so that the correlation between $\boldsymbol{g}^\top \boldsymbol{x}$ and $\boldsymbol{h}^\top \boldsymbol{y}$ is maximized (Hotelling, 1936).

More formally, given the first k solutions $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_k$ and $\boldsymbol{h}_1, \ldots, \boldsymbol{h}_k$, the $(k+1)$-st solution $\boldsymbol{g}_{k+1}, \boldsymbol{h}_{k+1}$ of CCA is given as

$$\underset{\boldsymbol{g}\in\mathbb{R}^m, \boldsymbol{h}\in\mathbb{R}^n}{\mathrm{argmax}} \quad \frac{\boldsymbol{g}^\top \widehat{\boldsymbol{\Sigma}}_{\mathrm{XY}} \boldsymbol{h}}{\sqrt{\boldsymbol{g}^\top \widehat{\boldsymbol{\Sigma}}_{\mathrm{XX}} \boldsymbol{g}}\sqrt{\boldsymbol{h}^\top \widehat{\boldsymbol{\Sigma}}_{\mathrm{YY}} \boldsymbol{h}}},$$
$$\text{s.t.} \quad \boldsymbol{g}_j^\top \widehat{\boldsymbol{\Sigma}}_{\mathrm{XX}} \boldsymbol{g} = 0 \quad \text{and} \quad \boldsymbol{h}_j^\top \widehat{\boldsymbol{\Sigma}}_{\mathrm{YY}} \boldsymbol{h} = 0, \quad j = 1, \ldots, k.$$

$\widehat{\boldsymbol{\Sigma}}_{\mathrm{XX}}$ and $\widehat{\boldsymbol{\Sigma}}_{\mathrm{YY}}$ are the sample covariance matrices of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively, and $\widehat{\boldsymbol{\Sigma}}_{\mathrm{XY}}$ is the sample cross-covariance matrix of $\boldsymbol{x}$ and $\boldsymbol{y}$.

The CCA solution $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_d$ and $\boldsymbol{h}_1, \ldots, \boldsymbol{h}_d$ for $d \leq \min(m, n)$ is given analytically by the generalized eigenvectors associated with the d largest generalized eigenvalues of the following generalized eigenvalue problem:

$$\begin{bmatrix} \boldsymbol{O} & \widehat{\boldsymbol{\Sigma}}_{\mathrm{XY}} \\ \widehat{\boldsymbol{\Sigma}}_{\mathrm{XY}}^{\top} & \boldsymbol{O} \end{bmatrix} \begin{bmatrix} \boldsymbol{g} \\ \boldsymbol{h} \end{bmatrix} = \eta \begin{bmatrix} \widehat{\boldsymbol{\Sigma}}_{\mathrm{XX}} & \boldsymbol{O} \\ \boldsymbol{O} & \widehat{\boldsymbol{\Sigma}}_{\mathrm{YY}} \end{bmatrix} \begin{bmatrix} \boldsymbol{g} \\ \boldsymbol{h} \end{bmatrix},$$

where $\boldsymbol{O}$ is the zero matrix. Finally, CCA projection matrices are constructed as

$$\boldsymbol{U} = (\boldsymbol{g}_1 \mid \cdots \mid \boldsymbol{g}_d)^{\top} \quad \text{and} \quad \boldsymbol{V} = (\boldsymbol{h}_1 \mid \cdots \mid \boldsymbol{h}_d)^{\top}. \tag{6}$$

Although the classical CCA has been widely used in a variety of fields, it can only capture correlations under linear projections. Thus, it is often insufficient to analyze complex real-world data. If $p_{\mathrm{xy}}(\boldsymbol{x}, \boldsymbol{y})$ is jointly normal, MI can be computed as a function of canonical correlations (Bach & Jordan, 2002; Kay, 1992; Kullback, 1959). Although this implies that CCA is closely related to MI under the Gaussian assumption, such relation to MI does not hold in general since CCA only evaluates the second-order correlation. On the other hand, our proposed approach can capture statistical dependency for any distribution, which is a strong advantage in real-world complex data analysis.

3.2 Kernel Canonical Correlation Analysis (KCCA)

KCCA (Akaho, 2001; Gestel et al., 2001; Melzer et al., 2001) could be a more flexible alternative to the classical CCA.

The basic idea of KCCA is to first non-linearly transform data samples into feature spaces by reproducing kernels (Aronszajn, 1950), and then apply ordinary CCA in the feature spaces. This corresponds to considering non-linear projections in the original spaces. Thanks to the reproducing property of the kernel functions, the global optimal solution of KCCA can be computed efficiently even when the dimensionality of the feature spaces is very high. This is a significant advantage over general non-linear approaches such as a method based on neural networks which suffers from local optimality (Fyfe & Lai, 2000). However, the result of KCCA is often difficult to interpret because projections are non-linear in the original space.

Bach & Jordan (2002) proposed a method that learns linear projection matrices using a KCCA-based dependence measure in the context of independent component analysis. They also proposed another kernel-based dependency measure called the *kernel generalized variance* (KGV) that is closely related to mutual information. Gretton et al. (2005) also proposed another kernel-based dependency measure called the *Hilbert-Schmidt independence criterion* (HSIC). These dependency measures can be directly applied to the CDA problem by maximizing each dependency measure with respect to the linear projection matrices (in the experiments shown later, we call these methods KCCA-CDA, KGV-CDA, and HSIC-CDA, respectively). Since these methods learn linear projection matrices in the original space, we can interpret the results easily. However, a critical limitation of these kernel-based approaches is that it contains free parameters to be tuned such as the

regularization parameter and the kernel parameter. Although practical heuristics were suggested for tuning parameter choice (Bach & Jordan, 2002; Gretton et al., 2005), there seems no theoretical justification and thus it is not clear whether such heuristics are always reliable. In a related context, cross-validation was used to choose the regularization parameter so that the first canonical correlation is maximized (Leurgans et al., 1993). However, this approach was shown not to work well for subsequent projection vectors.

On the other hand, LSCDA is equipped with cross-validation (3) and thus all tuning parameters can be objectively determined in a data-dependent manner, which is a significant advantage.

3.3 Informational Canonical Correlation Analysis (ICCA)

ICCA (Yin, 2004) shares the same goal as the proposed LSCDA, i.e., it obtains linear projections that maximize the dependency between two projected variables. However, in ICCA, the dependency is measured by MI,

$$\mathrm{MI} := \iint p_{\mathrm{uv}}(\boldsymbol{u},\boldsymbol{v}) \log \frac{p_{\mathrm{uv}}(\boldsymbol{u},\boldsymbol{v})}{p_{\mathrm{u}}(\boldsymbol{u})p_{\mathrm{v}}(\boldsymbol{v})} \mathrm{d}\boldsymbol{u}\mathrm{d}\boldsymbol{v},$$

which is estimated using kernel density estimation (KDE).

ICCA iteratively finds projection directions $\boldsymbol{g}$ and $\boldsymbol{h}$ so that the KDE-based MI estimate between $\boldsymbol{g}^\top\boldsymbol{x}$ and $\boldsymbol{h}^\top\boldsymbol{y}$ is maximized. Let $\widehat{p}_{\mathrm{uv}}(\boldsymbol{u},\boldsymbol{v})$, $\widehat{p}_{\mathrm{u}}(\boldsymbol{u})$, and $\widehat{p}_{\mathrm{v}}(\boldsymbol{v})$ be density estimators obtained by KDE. Then, given the first k solutions $\boldsymbol{g}_1,\ldots,\boldsymbol{g}_k$ and $\boldsymbol{h}_1,\ldots,\boldsymbol{h}_k$, the $(k+1)$-st solution $\boldsymbol{g}_{k+1},\boldsymbol{h}_{k+1}$ of ICCA is given as

$$\begin{aligned}
\underset{\boldsymbol{g}\in\mathbb{R}^m,\boldsymbol{h}\in\mathbb{R}^n}{\mathrm{argmax}} \quad & \frac{1}{\ell}\sum_{i=1}^{\ell} \log \frac{\widehat{p}_{\mathrm{uv}}(\boldsymbol{g}^\top\boldsymbol{x}_i,\boldsymbol{h}^\top\boldsymbol{y}_i)}{\widehat{p}_{\mathrm{u}}(\boldsymbol{g}^\top\boldsymbol{x}_i)\widehat{p}_{\mathrm{v}}(\boldsymbol{h}^\top\boldsymbol{y}_i)} \\
\text{s.t.} \quad & \boldsymbol{g}^\top\widehat{\boldsymbol{\Sigma}}_{\mathrm{XX}}\boldsymbol{g} = \boldsymbol{h}^\top\widehat{\boldsymbol{\Sigma}}_{\mathrm{YY}}\boldsymbol{h} = 1, \\
& \boldsymbol{g}_j^\top\widehat{\boldsymbol{\Sigma}}_{\mathrm{XX}}\boldsymbol{g} = \boldsymbol{h}_j^\top\widehat{\boldsymbol{\Sigma}}_{\mathrm{YY}}\boldsymbol{h} = 0, \quad j=1,\ldots,k.
\end{aligned}$$

Although bandwidths of KDE were chosen based on Silverman's rule (Scott, 1992; Silverman, 1986) in the original ICCA paper (Yin, 2004), we found in our preliminary experiments that likelihood cross-validation (LCV) (Härdle et al., 2004) tends to perform better. For this reason, we decided to use LCV in our experiments.

ICCA iteratively estimates pairs of projection vectors $\{(\boldsymbol{g}_i,\boldsymbol{h}_i)\}_{i=1}^d$ for $d \le \min(n,m)$, and then projection matrices are constructed by (6). This means that only one or two-dimensional density estimation is involved in ICCA, and thus KDE-based MI estimation would be reasonable. However, this approach corresponds to estimating rows of $\boldsymbol{U}$ and $\boldsymbol{V}$ one by one in a greedy manner. Although greedy optimization was shown to give the global optimal solution in the case of classical CCA (Izenman, 2008), it usually leads to a local optimal solution in ICCA. Indeed, as we will show experimentally, directly estimating multi-dimensional projections (i.e., global optimization) is potentially more

powerful. We can easily extend ICCA so that entire projection matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ are estimated at once:

$$\underset{\boldsymbol{U}\in\mathbb{R}^{p\times m},\boldsymbol{V}\in\mathbb{R}^{q\times n}}{\operatorname{argmax}} \quad \frac{1}{\ell}\sum_{i=1}^{\ell}\log\frac{\widehat{p}_{\mathrm{uv}}(\boldsymbol{U}^\top\boldsymbol{x}_i,\boldsymbol{V}^\top\boldsymbol{y}_i)}{\widehat{p}_{\mathrm{u}}(\boldsymbol{U}^\top\boldsymbol{x}_i)\widehat{p}_{\mathrm{v}}(\boldsymbol{V}^\top\boldsymbol{y}_i)}.$$

However, this formulation involves higher-dimensional density estimation (e.g., p_{uv} is now defined on the $(p+q)$-dimensional space), which tends to be inaccurate. Furthermore, taking the ratio of estimated densities tends to magnify the estimation error. For this reason, multi-dimensional ICCA may not be reliable in practice.

On the other hand, our proposed approach mitigates this difficulty by directly estimating the density ratio without going through density estimation, which tends to give more reliable solutions.

4 Experiments

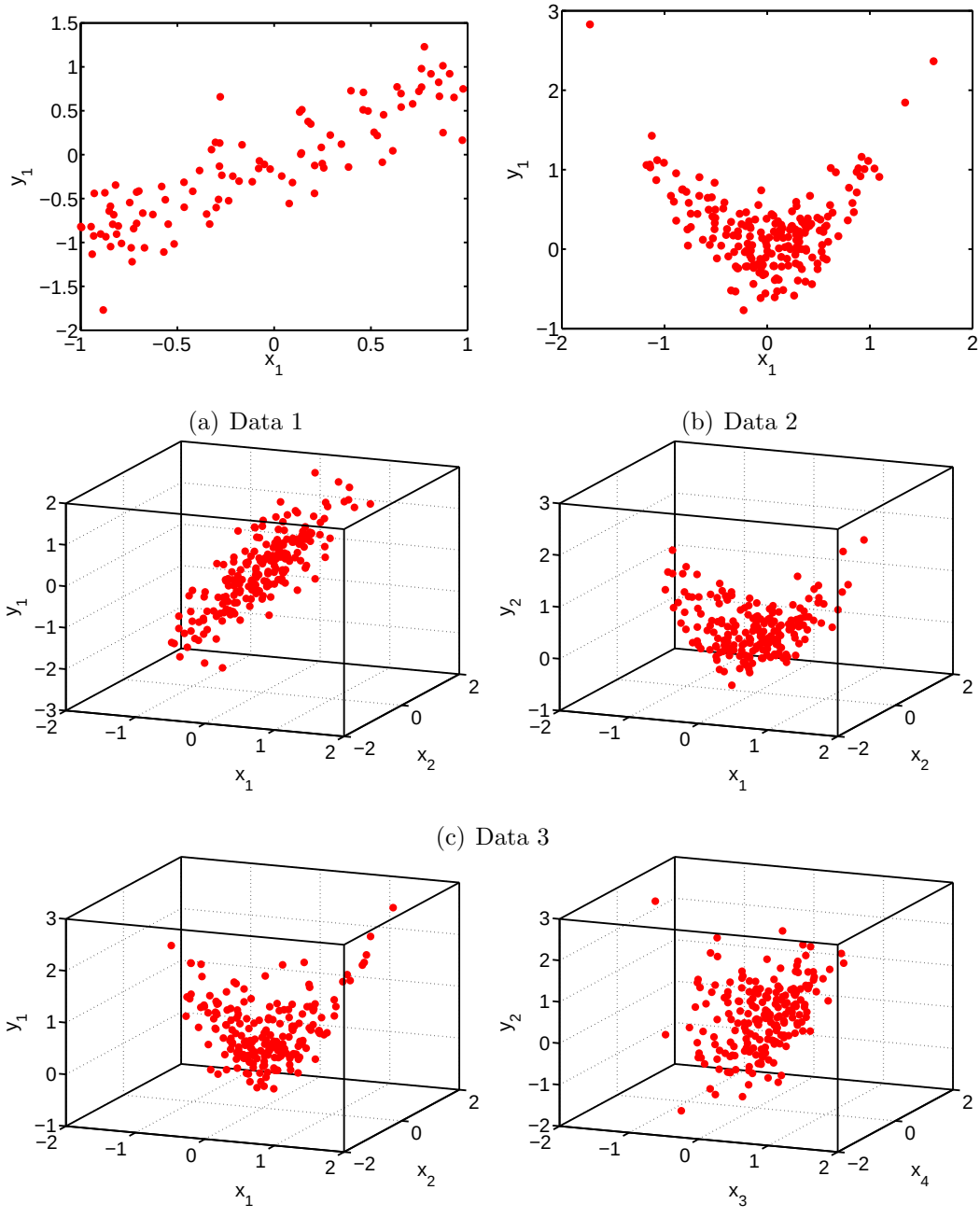
In this section, we experimentally evaluate the performance of the proposed LSCDA and existing methods. In the proposed method, we used the Gaussian kernel and fixed the number of basis functions by (4). The Gaussian width σ and regularization parameter λ were chosen based on 5-fold CV with grid search. We employed the natural gradient method as an optimization strategy for LSCDA. We restarted the gradient procedure 10 times with random initial points and chose the one having the minimum CV score (3).

4.1 Artificial Datasets

Here, we compared the performance of the LSCDA algorithm with classical CCA (Hotelling, 1936), KCCA-CDA (Bach & Jordan, 2002), KGV-CDA (Bach & Jordan, 2002), HSIC-CDA (Gretton et al., 2005), and ICCA (Yin, 2004) using artificial datasets. For classical CCA, we used the "*canoncorr*" function in the MATLAB® Statistics Toolbox. In ICCA, we estimated densities in MI using KDE following the original paper (Yin, 2004). We optimized the bandwidth parameters of KDE based on likelihood cross-validation (LCV), as mentioned in Section 3.3. In KCCA-CDA and KGV-CDA, the Gaussian width and the regularization parameter were set by the heuristics suggested in the original paper (Bach & Jordan, 2002). In HSIC-CDA, the Gaussian width was set to the median of sample distances, following the suggestions of the original paper (Gretton et al., 2005). We evaluated the performance of each method by

$$\frac{1}{2}\left\{\frac{1}{\sqrt{2p}}\|\widehat{\boldsymbol{U}}^\top\widehat{\boldsymbol{U}}-\boldsymbol{U}^{*\top}\boldsymbol{U}^*\|_{\mathrm{Fro}}+\frac{1}{\sqrt{2q}}\|\widehat{\boldsymbol{V}}^\top\widehat{\boldsymbol{V}}-\boldsymbol{V}^{*\top}\boldsymbol{V}^*\|_{\mathrm{Fro}}\right\},$$

where $\|\cdot\|_{\mathrm{Fro}}$ denotes the Frobenius norm, $\widehat{\boldsymbol{U}}$ and $\widehat{\boldsymbol{V}}$ are estimated projection matrices, and $\boldsymbol{U}^*$ and $\boldsymbol{V}^*$ are the optimal projection matrices. Note that the above measure takes its value in $[0,1]$, and smaller is better.



(a) Data 1

(b) Data 2

(c) Data 3

(d) Data 4

Figure 2: Artificial datasets. We generated four types of artificial datasets which have linear and/or nonlinear dependencies. The definitions of $\boldsymbol{x}$ and $\boldsymbol{y}$ are in Section 4.1. In Data 3, although we have $\boldsymbol{x} \in \mathbb{R}^5$ and $\boldsymbol{y} \in \mathbb{R}^5$, we only plot the relationships between y_1 and (x_1, x_2), and y_2 and (x_1, x_2), respectively. For Data 4, we plot the relationships between y_1 and (x_1, x_2), and y_2 and (x_3, x_4), respectively.

Let $U(\mathcal{D})$ denote the uniform distribution on $\mathcal{D}$, and let $N(\mu, \sigma^2)$ denote the normal distribution with mean μ and variance σ^2. We used the following 4 datasets (see Figure 2) for performance comparison, where the reduced dimensions of $\boldsymbol{x}$ and $\boldsymbol{y}$ were set to be the same number $d \leq \min(m, n)$.

(a) Data 1: $\boldsymbol{x} = (x_1, x_2)^\top$, $\boldsymbol{y} = (y_1, y_2)^\top$, and $d = 1$, where

$$y_1 = x_1 + \varepsilon,$$

$x_1, x_2 \sim U([-1, 1]^2)$, $y_2 \sim U([-1, 1])$, $\varepsilon \sim N(0, 0.1)$, and $\ell = 100$.

(b) Data 2: $\boldsymbol{x} = (x_1, x_2)^\top$, $\boldsymbol{y} = (y_1, y_2)^\top$, and $d = 1$, where

$$y_1 = x_1^2 + \varepsilon,$$

$x_1, x_2 \sim N(0, 0.25)$, $y_2 \sim N(0, 0.25)$, $\varepsilon \sim N(0, 0.1)$, and $\ell = 200$.

(c) Data 3: $\boldsymbol{x} = (x_1, \ldots, x_5)^\top$, $\boldsymbol{y} = (y_1, \ldots, y_5)^\top$, and $d = 2$, where

$$\begin{aligned} y_1 &= x_1 + x_2 + \varepsilon, \\ y_2 &= x_1^2 + \varepsilon, \end{aligned}$$

$x_1, \ldots, x_5 \sim N(0, 0.25)$, $y_3, \ldots, y_5 \sim N(0, 0.25)$, $\varepsilon \sim N(0, 0.1)$, and $\ell = 200$.

(d) Data 4: $\boldsymbol{x} = (x_1, \ldots, x_8)^\top$, $\boldsymbol{y} = (y_1, \ldots, y_8)^\top$, and $d = 5$, where

$$\begin{aligned} y_1 &= x_1^2 + x_2^2 + \varepsilon, \\ y_2 &= x_3^2 + x_4 + \varepsilon, \\ y_3 &= x_1^2 + x_3^2 + \varepsilon, \\ y_4 &= x_4^2 + x_2 + \varepsilon, \\ y_5 &= x_5^2 + x_3 + \varepsilon, \end{aligned}$$

$x_1, \ldots, x_8 \sim N(0, 0.25)$, $y_6, \ldots, y_8 \sim N(0, 0.25)$, $\varepsilon \sim N(0, 0.1)$, and $\ell = 200$.

The performance of each method is summarized in Table 1, which shows the mean and standard deviation of the Frobenius-norm error over 10 trials. For both of the ICCA and LSCDA methods, we evaluated the following two types of optimization strategies:

- Directly estimate d-dimensional projection matrices using the natural gradient method. In the table, ICCA and LSCDA indicate ICCA and LSCDA with this approach.
- Iteratively estimate pairs of projection vectors $\{(\boldsymbol{g}_i, \boldsymbol{h}_i)\}_{i=1}^d$ using the SQP method. ICCA' and LSCDA' in the table indicate the methods where projection matrices are optimized by this greedy approach. The original paper of ICCA (Yin, 2004) takes this approach.

Table 1: Mean and standard deviation of the Frobenius norm error for artificial datasets. The best method in terms of the mean error and comparable ones according to the t-test at the significance level 1 % are specified by boldface. In the table, ICCA' and LSCDA' indicate that projection matrices are optimized by one-dimensional greedy approach, while ICCA and LSCDA estimate d-dimensional projection matrices directly. Since LSCDA and LSCDA' (and also ICCA and ICCA') are equivalent in the case of $d = 1$, for datasets (a) and (b), we only reported results of LSCDA (and ICCA).

Dataset	CCA	KCCA-CDA	KGV-CDA	HSIC-CDA	ICCA'	ICCA	LSCDA'	LSCDA
(a)	**.05(.03)**	.06(.03)	**.05(.03)**	**.05(.03)**	-	.07(.04)	-	**.05(.03)**
(b)	.54(.28)	**.06(.05)**	**.06(.04)**	.08(.06)	-	.08(.04)	-	**.07(.04)**
(c)	.56(.13)	.59(.16)	.20(.06)	.27(.17)	.56(.15)	.42(.19)	.46(.17)	**.16(.06)**
(d)	.51(.05)	.49(.08)	.30(.06)	**.22(.09)**	.41(.05)	.53(.04)	.40(.06)	**.23(.07)**

Note that the above two approaches are equivalent for the datasets (a) and (b) where $d = 1$.

In Table 1, we can see that the proposed LSCDA has better or comparable performance to other methods. Although KGV-CDA and HSIC-CDA also tend to work reasonably well, LSCDA outperforms them for some datasets. These differences of performance seem to come from inappropriate choices of tuning parameters in KGV-CDA and HSIC-CDA.

For the dataset (a) having simple linear dependency, most of the methods have comparable performance. For the dataset (b) having dependency but no correlation, the classical CCA did not work well, whereas other methods more successfully captured the dependency of variables.

For the dataset (c), LSCDA performed well compared with LSCDA'. The reason why LSCDA' failed to estimate projection matrices for this dataset may be explained as follows. In the first iteration, $\boldsymbol{g}_1 \propto (1, 1, 0, 0, 0)^\top$ and $\boldsymbol{h}_1 \propto (1, 0, 0, 0, 0)^\top$ are typically obtained. Let g_{ij} be the jth component of $\boldsymbol{g}_i$ and h_{ij} be the jth component of $\boldsymbol{h}_i$. Then, in the second iteration, $g_{23}, \ldots, g_{25}$ must be 0 because x_3, x_4, and x_5 are irrelevant to $\boldsymbol{y}$. Therefore, due to the orthogonality $\boldsymbol{g}_1^\top \boldsymbol{g}_2 = 0$, $\boldsymbol{g}_2 \propto (1, -1, 0, 0, 0)^\top$ should be obtained. Similarly, since $h_{23}, \ldots, h_{25}$ must be 0, $\boldsymbol{h}_2 \propto (0, 1, 0, 0, 0)^\top$ should be obtained. However, since there seems no strong dependency between $\boldsymbol{g}_2^\top \boldsymbol{x} = (x_1 - x_2)/\sqrt{2}$ and $\boldsymbol{h}_2^\top \boldsymbol{y} = y_2$ (see Figure 3), it is difficult to find this solution by the iterative approach.

The dataset (d) contains rather complicated dependencies of variables in higher-dimensional subspaces. Due to the same reason as the dataset (c), the one-dimensional greedy strategy did not work well for the dataset (d). Furthermore, since ICCA involved higher-dimensional density estimation (10- and 5-dimensional density estimations are needed for the dataset (c)), it did not perform well compared with LSCDA. On the other hand, LSCDA mitigated this difficulty thanks to the direct density-ratio estimation approach.

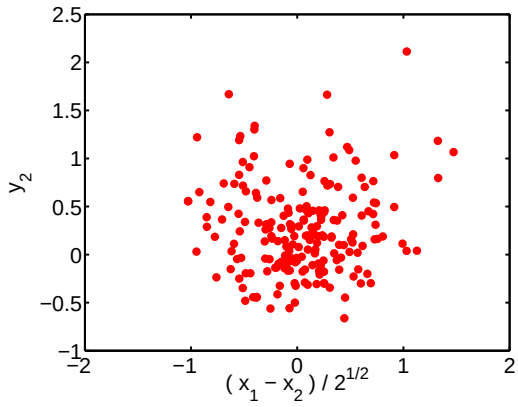


Figure 3: Relationship between $(x_1 - x_2)/\sqrt{2}$ and y_2 in dataset (c). There seems no strong dependency between these two quantities.

4.2 Application to Real-world Multi-label Classification Problem

Finally, we evaluate the performance of the proposed method in real-world multi-label classification problems. Here, the proposed LSCDA is compared to the classical CCA, HSIC-CDA, and ICCA. We used a real-world image dataset called the *PASCAL Visual Object Classes (VOC) 2010* dataset (Everingham et al., 2010) and a real-world audio dataset called the *Freesound* dataset (Akkermans et al., 2011). In both of the datasets, we consider a set of binary classification problems that is to predict a binary label vector $\boldsymbol{y} \in \{-1, +1\}^n$ from an input vector $\boldsymbol{x} \in \mathbb{R}^m$. In these problems, the labels in each dimension are not independent of each other, but often have strong correlations. Then, extracting informative features while incorporating co-occurrence patterns among labels $\boldsymbol{y}$ is important for better prediction. Therefore, here we compared each method by prediction performance of subsequent classifier.

To evaluate each method, the one nearest-neighbor classifier was applied to transformed samples. We employed the receiver operating characteristic (ROC) analysis (Fawcett, 2006) and calculated the area under the ROC curve (AUC) as a performance measure of each method. The AUC corresponds to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance. The AUC is often calculated through varying a threshold of a classifier. Since the usual nearest-neighbor classifier does not have a threshold, we decided to use weighted distances to neighboring instances of each class. More specifically, we used the following distance function $\mathrm{dist} : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ for the nearest-neighbor classifier:

$$\mathrm{dist}(\boldsymbol{x}, \boldsymbol{x}_i) = \begin{cases} \alpha \|\boldsymbol{U}(\boldsymbol{x} - \boldsymbol{x}_i)\|_2^2 & \text{for } y_i = 1, \\ (1-\alpha)\|\boldsymbol{U}(\boldsymbol{x} - \boldsymbol{x}_i)\|_2^2 & \text{for } y_i = -1, \end{cases}$$

where $\boldsymbol{x}$ is a test data point, $\boldsymbol{x}_i$ is a training data point, and $\alpha \in [0, 1]$ is a weight parameter that corresponds to a threshold of the classifier. For instance, when α is set

as 0, the classifier always outputs $y = 1$. On the other hand, it always outputs $y = -1$ when $\alpha = 1$. Changing α from 0 to 1, we calculate the AUC for each element of $\boldsymbol{y}$ and show their average.

4.2.1 PASCAL VOC 2010 Dataset

The PASCAL VOC 2010 dataset consists of 20 binary classification tasks of identifying the existence of objects in given images such as a person and an aeroplane. The total number of images in the dataset is 11319 and we randomly divided it into training and test dataset 10 times. The number of training instances is 500 and the rest are used for testing.

In this experiment, we first extracted visual features from each image using the *speed up robust features* (SURF) algorithm (Bay et al., 2008), and obtained 500 *visual words* as the cluster centers in the SURF space. Then, we computed a 500-dimensional *bag-of-feature* vector by counting the number of visual words in each image.

We reduced the dimensionalities of $\boldsymbol{x}$ and $\boldsymbol{y}$ from $m = 500$ and $n = 20$ to $p \in \{15, 10, 5, 1\}$ and $q = 10$, and observed performance for each p. In this setting, except for the classical CCA, we have to run each method 4 times for $p \in \{15, 10, 5, 1\}$. For computational efficiency, we employed the following re-starting strategy in HSIC-CDA, ICCA, and LSCDA. Let $\widehat{\boldsymbol{U}}^{(1)} \in \mathbb{R}^{15\times 500}$ be an estimated projection matrix for $p = 15$ and $\widehat{\boldsymbol{u}}^{(1)} := \widehat{\boldsymbol{U}}^{(1)}\boldsymbol{x}$. Using this, we further reduce the dimensionality from $\widehat{\boldsymbol{u}}^{(1)}$ by estimating an "additional" projection matrix $\widehat{\boldsymbol{U}}^{(2)} \in \mathbb{R}^{10\times 15}$. Thus, the 10-dimensional projected vector is written as $\widehat{\boldsymbol{U}}^{(2)}\widehat{\boldsymbol{u}}^{(1)}$. Note that $\widehat{\boldsymbol{U}}^{(2)}\widehat{\boldsymbol{U}}^{(1)}$ lies also in the Stiefel manifold: $\widehat{\boldsymbol{U}}^{(2)}\widehat{\boldsymbol{U}}^{(1)}\widehat{\boldsymbol{U}}^{(1)\top}\widehat{\boldsymbol{U}}^{(2)\top} = \boldsymbol{I}_{10}$. Since this strategy reduces the size of projection matrices, we can save the computation cost for large dimensional problems. The same technique can also be applied to the projection matrix for $\boldsymbol{y}$.

The results are plotted in Figure 4. In the plot, "NDR" (no dimension reduction) corresponds to the one nearest-neighbor classification in the original space. The results show that the classical CCA has the worst performance and its AUC value is close to the chance level. Therefore, more flexible approaches seem to be required for this problem. Although ICCA and HSIC-CDA had better performance compared to the classical CCA, the proposed LSCDA outperformed all of them. LSCDA achieved almost the same performance as "NDR" with only a 5-dimensional subspace.

Figure 5 is CPU time comparison of each dimensionality reduction method. For classical CCA, we can obtain projection matrices for every dimension by only solving an eigenvalue problem once. On the other hand, HSIC, ICCA and LSCDA needed relatively large computational costs to maximize their dependency measures. Since HSIC, ICCA and LSCDA employed re-start strategy from a solution of large p to small p, their CPU time decreased with the increase of p.

4.2.2 Freesound Dataset

The Freesound dataset (Akkermans et al., 2011) consists of various audio files annotated with word tags such as "people", "noisy", and "restaurant". We used 230 tags in this

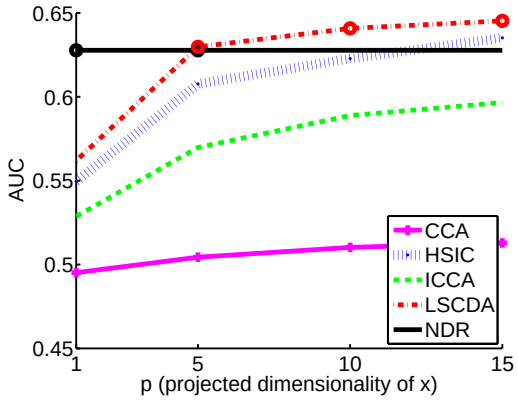


Figure 4: Comparison of the average AUC score for the PASCAL VOC 2010 dataset. The best method and comparable methods according to the t-test at the significance level 1% are specified by "○". "NDR" denotes the original data without dimension reduction. The horizontal axis is corresponding to projected dimensionality of $\boldsymbol{x}$ (i.e., dimensionality of $\boldsymbol{u}$).

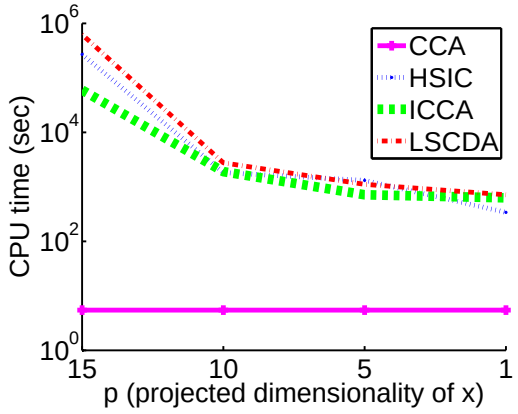


Figure 5: CPU time comparison of dimensionality reduction. The horizontal axis is corresponding to projected dimensionality of $\boldsymbol{x}$ (i.e., dimensionality of $\boldsymbol{u}$). Since HSIC, ICCA and LSCDA employed re-start strategy from solution of large p to small p, the CPU time is decreasing with increase of p. For CCA, we can obtain projections for every dimension by performing the eigenvalue decomposition only once.

experiment. The total number of audio files in the dataset is 5905 and we used 500 randomly chosen audio files for training and the rest for testing.

We first extracted the *mel-frequency cepstrum coefficient* (MFCC) (Rabiner & Juang, 1993) from each audio file, and obtained 1024 *audio features* as the cluster centers in the MFCC space. Then, we computed a 1024-dimensional *bag-of-feature* vector by counting the number of audio features in each audio file. We randomly chose the training and test datasets 10 times.

We reduced the dimensionalities of $\boldsymbol{x}$ and $\boldsymbol{y}$ from $m = 500$ and $n = 230$ to $p = \{15, 10, 5, 1\}$ and $q = 30$. As in the case of the PASCAL VOC dataset, we employed the same re-starting strategy for efficient computations.

The results are plotted in Figure 6, showing again that LSCDA outperformed the existing methods and it had comparable performance to "NDR" with only $p = 5$.

Figure 7 is CPU time comparison of each dimensionality reduction method. Here again, HSIC, ICCA and LSCDA were computationally expensive compared to classical CCA.

5 Conclusions

In this paper, we proposed a novel dimensionality reduction method for paired data, called *least-squares canonical dependency analysis* (LSCDA), that maximizes dependency between two projected variables. The proposed LSCDA can capture higher-order correlations which can not be detected by classical *canonical correlation analysis* (CCA). As a criterion of dependency, we employed *squared-loss mutual information* (SMI) which can be accurately and analytically estimated by *least-squares mutual information* (LSMI). Our method does not involve density estimation which is often difficult in higher-dimensional problems, but we estimate the ratio of densities directly. Through experiments, we demonstrated the effectiveness of our LSCDA method using artificial datasets and real-world image and audio datasets. In our future work, we will improve the computational efficiency of LSCDA.

Acknowledgement

Masayuki Karasuyama was supported by JSPS and Masashi Sugiyama was supported by the JST PRESTO program and AOARD.

References

Akaho, S. (2001). A kernel method for canonical correlation analysis. In *In Proceedings of the International Meeting of the Psychometric Society (IMPS2001)*. Springer-Verlag.

Akkermans, V., Font, F., Funollet, J., de Jong, B., Roma, G., Togias, S., & Serra, X. (2011). Freesound 2: An improved platform for sharing audio clips. In

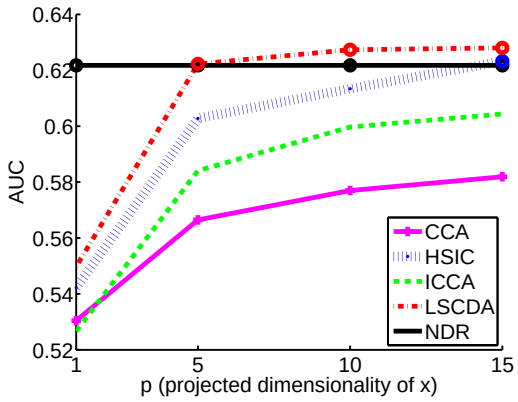


Figure 6: Comparison of the average AUC score for the Freesound dataset. The best method and comparable methods according to the t-test at the significance level 1% are specified by "○". "NDR" denotes the original data without dimension reduction. The horizontal axis is corresponding to projected dimensionality of $\boldsymbol{x}$ (i.e., dimensionality of $\boldsymbol{u}$).

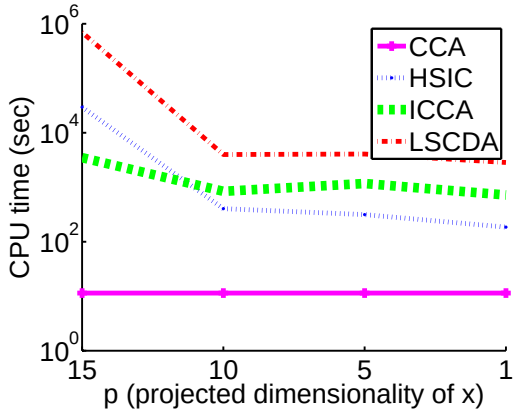


Figure 7: CPU time comparison of dimensionality reduction. The horizontal axis is corresponding to projected dimensionality of $\boldsymbol{x}$ (i.e., dimensionality of $\boldsymbol{u}$).

International Society for Music Information Retrieval Conference (ISMIR 2011), Late-breaking Demo Session. `http://mtg.upf.edu/system/files/publications/freesound_ismir.pdf`.

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

Amari, S. (1998). Natural gradient works efficiently in learning. *Neural Computation*, *10*, 251–276.

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

Bach, F. R., & Jordan, M. I. (2002). Kernel independent component analysis. *Journal of Machine Learning Research*, *3*, 1–48.

Bartlett, M. S. (1938). Further aspects of the theory of multiple regression. *Mathematical Proceedings of the Cambridge Philosophical Society*, *34*, 33–40.

Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). SURFS: peeded-up robust features. *Computer Vision Image Understanding*, *110*, 346–359.

Becker, S. (1996). Mutual Information Maximization: models of cortical self-organization. *Network : Computation in Neural Systems*, *7*, 7–31.

Becker, S., & Hinton, G. E. (1992). A self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, *355*, 161–163.

Bossaerts, P. (1988). Common nonstationary components of asset prices. *Journal of Economic Dynamics and Control*, *12*, 347 – 364.

Branco, J., Croux, C., Filzmoser, P., & Oliveira, M. (2005). Robust canonical correlations: A comparative study. *Computational Statistics*, *20*, 203–229.

Breiman, L., & Friedman, J. H. (1985). Estimating optimal transformations for multiple regression and correlation. *Journal of the American Statistical Association*, *80*, 580–598.

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

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

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. `http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html`.

Farquhar, J. D. R., Hardoon, D. R., Meng, H., Shawe-Taylor, J., & Szedmák, S. (2005). Two view learning: SVM-2K, theory and practice. In Y. Weiss, B. Schölkopf, & J. Platt (Eds.), *Advances in Neural Information Processing Systems 18*. Cambridge, MA: MIT Press.

Favorov, O. V., & Ryder, D. (2004). Sinbad: A neocortical mechanism for discovering environmental variables and regularities hidden in sensory input. *Biological Cybernetics*, *90*, 191–202.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, *27*, 861 – 874.

Fukumizu, K., Bach, F. R., & Jordan, M. I. (2009). Kernel dimension reduction in regression. *The Annals of Statistics*, *37*, 1871–1905.

Fyfe, C., & Lai, P. L. (2000). Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, *10*, 365–377.

Gestel, T. V., Suykens, J., Brabanter, J. D., Moor, B. D., & Vandewalle, J. (2001). Kernel canonical correlation analysis and least squares support vector machines. In G. Dorffner, H. Bischof, & K. Hornik (Eds.), *International Conference on Artificial Neural Networks (ICANN)* (pp. 384–389). Springer Berlin / Heidelberg volume 2130 of *Lecture Notes in Computer Science*.

Golub, G. H., & Van Loan, C. F. (1996). *Matrix computations*. Baltimore, MD, USA: Johns Hopkins University Press.

Gretton, A., Bousquet, O., Smola, A., & Scholkopf, B. (2005). Measuring statistical dependence with Hilbert-Schmidt norms. In S. Jain, H. Simon, & E. Tomita (Eds.), *Algorithmic Learning Theory* (pp. 63–77). Springer Berlin / Heidelberg volume 3734 of *Lecture Notes in Computer Science*.

Gumus, E., Kursun, O., Sertbas, A., & Ustek, D. (2012). Application of canonical correlation analysis for identifying viral integration preferences. *Bioinformatics*, *to appear*.

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

Hardoon, D. R., Szedmak, S. R., & Shawe-taylor, J. R. (2004). Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, *16*, 2639–2664.

Hotelling, H. (1936). Relations between two sets of variates. *Biometrika*, *28*, 321–377.

Izenman, A. J. (2008). *Modern multivariate statistical techniques : regression, classification, and manifold learning*. Springer Texts in Statistics. Springer New York.

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

Kay, J. (1992). Feature discovery under contextual supervision using mutual information. In *International Joint Conference on Neural Networks* (pp. 79–84). volume 4.

Kullback, S. (1959). *Information Theory and Statistics*. New York: Wiley.

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

Kursun, O., Alpaydin, E., & Favorov, O. V. (2011). Canonical correlation analysis using within-class coupling. *Pattern Recognition Letters*, *32*, 134–144.

Kursun, O., & Favorov, O. V. (2010). Feature selection and extraction using an unsupervised biologically-suggested approximation to gebelein's maximal correlation. *International Journal of Pattern Recognition and Artificial Intelligence*, *24*, 337–358.

Leurgans, S. E., Moyeed, R. A., & Silverman, B. W. (1993). Canonical correlation analysis when the data are curves. *Journal of the Royal Statistical Society. Series B*, *55*, 725–745.

Li, Y., & Shawe-Taylor, J. (2006). Using KCCA for Japanese-English cross-language information retrieval and document classification. *Journal of Intelligent Information Systems*, *27*, 117–133.

McKeon, J. J. (1967). Canonical analysis: Some relations between canonical correlation, factor analysis, discriminant function analysis, and scaling theory. *Psychometric Monograph*, .

Melzer, T., Reiter, M., , & Bischof, H. (2001). *Kernel Canonical Correlation Analysis*. Technical Report PRIP-TR-65, Pattern Recognition and Image Processing Group, TU Wien.

Naylor, M. G., Lin, X., Weiss, S. T., Raby, B. A., & Lange, C. (2010). Using canonical correlation analysis to discover genetic regulatory variants. *PLoS ONE*, *5*, e10395.

Nishimori, Y., & Akaho, S. (2005). Learning algorithms utilizing quasi-geodesic flows on the stiefel manifold. *Neurocomputing*, *67*, 106–135.

Nocedal, J., & Wright, S. J. (1999). *Numerical optimization*. Springer.

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

Rabiner, L., & Juang, B.-H. (1993). *Fundamentals of speech recognition*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Rai, P., & Daume, H. (2009). Multi-label prediction via sparse infinite CCA. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, & A. Culotta (Eds.), *Advances in Neural Information Processing Systems 22* (pp. 1518–1526).

Scott, D. W. (1992). *Multivariate Density Estimation: Theory, Practice and Visualization*. New York, NY, USA: Wiley.

Shannon, C. (1948). A mathematical theory of communication. *Bell Systems Technical Journal*, *27*, 379–423.

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

Steinwart, I. (2001). On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, *2*, 67–93.

Storch, H. V., & Zwiers, F. W. (2002). *Statistical Analysis in Climate Research*. Cambridge University Press.

Sun, L., Ji, S., & Ye, J. (2011). Canonical correlation analysis for multilabel classification: A least-squares formulation, extensions, and analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*, 194–200.

Suzuki, T., & Sugiyama, M. (2010). Sufficient dimension reduction via squared-loss mutual information estimation. In Y. W. Teh, & M. Tiggerington (Eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS2010)* (pp. 804–811). Sardinia, Italy volume 9 of *JMLR Workshop and Conference Proceedings*.

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

Vert, J.-P., & Kanehisa, M. (2003). Graph-driven feature extraction from microarray data using diffusion kernels and kernel CCA. In S. T. S. Becker, & K. Obermayer (Eds.), *Advances in Neural Information Processing Systems 15* (pp. 1425–1432). Cambridge, MA: MIT Press.

Vinod, H. D. (1968). Econometrics of joint production. *Econometrica*, *36*, 322–336.

Vinokourov, A., Shawe-Taylor, J., & Cristianini, N. (2003). Inferring a semantic representation of text via cross-language correlation analysis. In S. T. S. Becker, & K. Obermayer (Eds.), *Advances in Neural Information Processing Systems 15* (pp. 1473–1480). Cambridge, MA: MIT Press.

Wold, H. (1966). Nonlinear estimation by iterative least squares procedures. In F. David (Ed.), *Research Papers in Statistics, Festschrift for Lerzy Newman* (pp. 441–444). Wiley, New York.

Yamanishi, Y., Vert, J. P., Nakaya, A., & Kanehisa, M. (2003). Extraction of correlated gene clusters from multiple genomic data by generalized kernel canonical correlation analysis. *Bioinformatics*, *19 (Suppl 1)*.

Yin, X. (2004). Canonical correlation analysis based on information theory. *Journal of Multivariate Analysis*, *91*, 161 – 176.