Yoshida *et al. BMC Bioinformatics* (2017) 18:108
DOI 10.1186/s12859-017-1543-x

BMC Bioinformatics

METHODOLOGY ARTICLE Open Access

Sparse kernel canonical correlation analysis for discovery of nonlinear interactions in high-dimensional data

Kosuke Yoshida[1,2*], Junichiro Yoshimoto[3] and Kenji Doya[2]

Abstract

Background: Advance in high-throughput technologies in genomics, transcriptomics, and metabolomics has created demand for bioinformatics tools to integrate high-dimensional data from different sources. Canonical correlation analysis (CCA) is a statistical tool for finding linear associations between different types of information. Previous extensions of CCA used to capture nonlinear associations, such as kernel CCA, did not allow feature selection or capturing of multiple canonical components. Here we propose a novel method, two-stage kernel CCA (TSKCCA) to select appropriate kernels in the framework of multiple kernel learning.

Results: TSKCCA first selects relevant kernels based on the HSIC criterion in the multiple kernel learning framework. Weights are then derived by non-negative matrix decomposition with L1 regularization. Using artificial datasets and nutrigenomic datasets, we show that TSKCCA can extract multiple, nonlinear associations among high-dimensional data and multiplicative interactions among variables.

Conclusions: TSKCCA can identify nonlinear associations among high-dimensional data more reliably than previous nonlinear CCA methods.

Keywords: Kernel canonical correlation analysis, Hilbert-Schmidt independent criterion, L1 regularization

Background

Canonical correlation analysis (CCA) [1] is a statistical method for finding common information from two different sources of multivariate data. This method optimizes linear projection vectors so that two random multivariate datasets are maximally correlated. With advances in high-throughput biological measurements, such as DNA sequencing, RNA microarrays, and mass spectroscopy, CCA has been extensively used for discovery of interactions between the genome, gene transcription, protein synthesis, and metabolites [2–5]. Because CCA solution is reduced to an eigenvalue problem, multiple components of interactions with sparse constraints are readily introduced [4, 6, 7].

Kernel CCA (KCCA) was introduced to capture nonlinear associations between two blocks of multivariate data [8–11]. Given two blocks of multivariate data $\mathbf{x}$ and $\mathbf{z}$, KCCA finds nonlinear transformations $f(\mathbf{x})$ and $g(\mathbf{z})$ in a reproducing kernel Hilbert space (RKHS) so that the correlation between $f(\mathbf{x})$ and $g(\mathbf{z})$ is maximized. In order to avoid overfitting and to improve interpretability of results, sparse additive functional CCA (SAFCCA) [12] constrains $f(\mathbf{x})$ and $g(\mathbf{z})$ as sparse additive models and optimizes them using the biconvex back-fitting algorithm [13]. However, it is not straightforward to obtain multiple orthogonal transformations for extracting multiple components of associations. Another method for finding nonlinear associations is to maximize measures of nonlinear matching, such as the Hilbert-Schmidt Independent Criterion (HSIC) [14] and the Kernel Target Alignment (KTA) [15] between linearly projected datasets $\mathbf{x}$ and $\mathbf{z}$ [16]. While these methods can obtain multiple orthogonal projections by iteratively analyzing residuals, it is impossible for these methods to remove irrelevant features, making them prone to overfitting.

In this paper, we propose two-stage kernel CCA (TSKCCA), which enables us (1) to select sparse features

*Correspondence: kosuke.yoshida@oist.jp
[1]Graduate School of Informatics, Kyoto University, Kyoto, Japan
[2]Neural Computation Unit, Okinawa Institute of Science and Technology, Okinawa, Japan
Full list of author information is available at the end of the article

in high-dimensional data and (2) to obtain multiple nonlinear associations. In the first stage, we represent target kernels with a weighted sum of pre-specified sub-kernels and optimize their weight coefficients based on HSIC with sparse regularization. In the second stage, we apply standard KCCA using target kernels obtained in the first stage to find multiple nonlinear correlations.

We briefly review CCA, KCCA, and two-stage MKL, and then present TSKCCA algorithm. We apply TSKCCA to three synthetic datasets and nutrigenomic experimental data to show that the method discovers multiple nonlinear associations within high-dimensional data, and provides interpretation that are robust to irrelevant features.

CCA, kernel CCA, and multiple kernel learning

In this section, we briefly review the bases of our proposed method, namely, linear canonical correlation analysis (CCA), kernel CCA (KCCA), and multiple kernel learning (MKL).

Canonical correlation analysis (CCA)

Let $D = \{(\mathbf{x}_n, \mathbf{z}_n)\}_{n=1}^N$ be N pairs of samples, where $\mathbf{x}_n$ and $\mathbf{z}_n$ are the n-th samples drawn from p- and q-dimensional Euclidian space, respectively. Let $f_w(\mathbf{x}) \equiv \mathbf{w}^T\mathbf{x}$ and $g_v(\mathbf{z}) \equiv \mathbf{v}^T\mathbf{z}$ denote the projection of $\mathbf{x} \in \mathbb{R}^p$ by $\mathbf{w} \in \mathbb{R}^p$ and that of $\mathbf{z} \in \mathbb{R}^q$ by $\mathbf{v} \in \mathbb{R}^q$, respectively. The objective of linear CCA is to find projections that maximize Pearson's correlation between $F_w \equiv \{f_w(\mathbf{x}_n)\}_{n=1}^N$ and $G_v \equiv \{g_v(\mathbf{z}_n)\}_{n=1}^N$ and formulated as the following optimization problem:

$$\max_{\mathbf{w}\in\mathbb{R}^p, \mathbf{v}\in\mathbb{R}^q} \quad \mathrm{Cov}(F_w, G_v) \tag{1a}$$

$$\text{subject to} \quad \mathrm{Var}(F_w) = \mathrm{Var}(G_v) = 1, \tag{1b}$$

where $\mathrm{Var}(\cdot)$ and $\mathrm{Cov}(\cdot,\cdot)$ denote the empirical variance and covariance of the data, respectively. The optimal solution $(\mathbf{w}^*, \mathbf{v}^*)$ of Eq. (1a and 1b) is obtained by solving generalized eigenvalue problems and successive eigenvectors represent multiple components. The projections, $f^*(\mathbf{x}) = \mathbf{w}^{*T}\mathbf{x}$ and $g^*(\mathbf{z}) = \mathbf{v}^{*T}\mathbf{z}$, are said to be canonical variables for $\mathbf{x} \in \mathbb{R}^p$ and $\mathbf{z} \in \mathbb{R}^q$, respectively. If we introduce sparse regularization on $\mathbf{w}$ and $\mathbf{v}$, we obtain sparse projections [4, 6, 7].

Kernel CCA

In Kernel CCA (KCCA), we suppose that the original data are mapped into a feature space via nonlinear functions. Then linear CCA is applied in the feature space. More specifically, nonlinear functions $\phi_x : \mathbb{R}^p \to \mathbb{H}_x$ and $\phi_z : \mathbb{R}^q \to \mathbb{H}_z$ transform the original data $\{(\mathbf{x}_n, \mathbf{z}_n)\}_{n=1}^N$ to feature vectors $\{(\phi_x(\mathbf{x}_n), \phi_z(\mathbf{z}_n))\}_{n=1}^N$ in reproducing kernel Hilbert spaces (RKHS) $\mathbb{H}_x$ and $\mathbb{H}_z$. Inner-product kernels for $\mathbb{H}_x$ and $\mathbb{H}_z$ are defined as $k_x(\mathbf{x}, \mathbf{x}') = \phi_x(\mathbf{x})^T\phi_x(\mathbf{x}')$, and $k_z(\mathbf{z}, \mathbf{z}') = \phi_z(\mathbf{z})^T\phi_z(\mathbf{z}')$.

Let us implement $f_w(\mathbf{x})$ and $g_v(\mathbf{z})$ by projections $f_w(\mathbf{x}) \equiv \mathbf{w}^T\phi_x(\mathbf{x})$ and $g_v(\mathbf{z}) \equiv \mathbf{v}^T\phi_z(\mathbf{z})$. By introducing appropriate regularization terms, Eq. (1a and 1b) can be reformulated as the following optimization problem ([8, 9]):

$$\max_{\alpha\in\mathbb{R}^N, \beta\in\mathbb{R}^N} \quad \alpha^T K_x K_z \beta \tag{2a}$$

$$\text{subject to} \quad \boldsymbol{\alpha}^T \left(K_x + \frac{N\kappa}{2} I\right)^2 \boldsymbol{\alpha} = 1 \tag{2b}$$

$$\boldsymbol{\beta}^T \left(K_z + \frac{N\kappa}{2} I\right)^2 \boldsymbol{\beta} = 1, \tag{2c}$$

where K_x and K_z are N-by-N kernel matrices defined as $[K_x]_{nn'} = k_x(\mathbf{x}_n, \mathbf{x}_{n'})$ and $[K_z]_{nn'} = k_z(\mathbf{z}_n, \mathbf{z}_{n'})$ [1]. I is the N-by-N identity matrix and κ ($\kappa > 0$) is the regularization parameter.

Once having obtained the solution of Eq. (2a–2c), denoted by $(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$, canonical variables for $\mathbf{x} \in \mathbb{R}^p$ and $\mathbf{z} \in \mathbb{R}^q$ are given by

$$f^*(\mathbf{x}) = \sum_{n=1}^N k_x(\mathbf{x}, \mathbf{x}_n)\alpha_n^* \tag{3a}$$

$$g^*(\mathbf{z}) = \sum_{n=1}^N k_z(\mathbf{z}, \mathbf{z}_n)\beta_n^*, \tag{3b}$$

respectively. As indicated by Eq. (2a–2c), the nonlinear functions, ϕ_x and ϕ_z, are not explicitly used in the computation of KCCA. Instead, the kernels k_x and k_z implicitly specify the nonlinear functions, and the main goal is to solve the constrained quadratic optimization problem with $2N$-dimensional variables.

Multiple kernel learning

Kernel methods usually require users to design a particular kernel, which critically affects the performance of the algorithm. To make the design more flexible, the framework of multiple kernel learning (MKL) was proposed for classification and regression problems [17, 18]. In MKL, we manually design M_x sub-kernels $\{k_x^{(m)}\}_{m=1}^{M_x}$, where each sub-kernel $k_x^{(m)}$ uses only a distinct set of features in $\mathbf{x}$. Also, M_z sub-kernels $\{k_z^{(l)}\}_{l=1}^{M_z}$ for $\mathbf{z}$ is also designed in the same manner. Then, k_x and k_z are represented as the weighted sum of those sub-kernels:

$$k_x(\mathbf{x}, \mathbf{x}') = \sum_{m=1}^{M_x} \eta_m k_x^{(m)}(\mathbf{x}, \mathbf{x}') \tag{4a}$$

$$k_z(\mathbf{z}, \mathbf{z}') = \sum_{l=1}^{M_z} \mu_l k_z^{(l)}(\mathbf{z}, \mathbf{z}'), \tag{4b}$$

where weight coefficients of sub-kernels, $\{\eta_m\}_{m=1}^{M_x}$ and $\{\mu_l\}_{l=1}^{M_z}$ are tuned to optimize an objective function.

A specific example of this framework is the two-stage MKL approach [15, 19]: In the first stage, the weight coefficients are optimized based on a similarity criterion, such as the kernel target alignment; then, a standard kernel algorithm, such as support vector machine, is applied in the second stage.

Methods

In this section, we propose a novel nonlinear CCA method, two-stage kernel CCA (TSKCCA), inspired by the concepts of sparse multiple kernel learning and kernel CCA. In the following, we present the general framework of TSKCCA, followed by our solutions for practical issues in the implementation.

First stage: multiple kernel learning with HSIC and sparse regularizer

In TSKCCA, sub-kernels are restricted to the same class as Eq. (4a and 4b), allowing us to express the kernel matrices K_x and K_z as

$$K_x = \sum_{m=1}^{M_x} \eta_m K_x^{(m)} \tag{5a}$$

$$K_z = \sum_{l=1}^{M_z} \mu_l K_z^{(l)}, \tag{5b}$$

where $[K_x^{(m)}]_{nn'} = k_x^{(m)}(\mathbf{x}_n, \mathbf{x}_{n'})$ and $[K_z^{(l)}]_{nn'} = k_z^{(l)}(\mathbf{z}_n, \mathbf{z}_{n'})$. The goal of the first stage is to optimize the weight vector $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_{M_x})^T$ and $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_{M_z})^T$ so that kernel matrices K_x and K_z statistically depend on each other as much as possible, while irrelevant sub-kernels are filtered out.

The statistical dependence between K_x and K_z is evaluated by the Hilbert-Schmidt Independent Criterion (HSIC) and approximated by its empirical estimator [14]:

$$\mathbb{D}(K_x, K_z) = \frac{\mathrm{Tr}(K_x H K_z H)}{(N-1)^2}, \tag{6}$$

where H is an N-by-N matrix such that $[H]_{nn'} = \delta_{nn'} - \frac{1}{N}$, and $\delta_{nn'}$ is Kronecker's delta. $\mathrm{Tr}(\cdot)$ denotes the trace. In our setting, optimization problem is reduced to a simple biliear form with respect to η and μ:

$$\mathbb{D}(K_x, K_z) = \boldsymbol{\eta}^{\mathrm{T}} M \boldsymbol{\mu}, \tag{7}$$

where M is a M_x-by-M_z matrix such that

$$[M]_{ml} = \frac{\mathrm{Tr}(K_x^{(m)} H K_z^{(l)} H)}{(N-1)^2}. \tag{8}$$

In addition to maximizing the dependency measure $\mathbb{D}(K_x, K_z)$, $\boldsymbol{\eta}$ and $\boldsymbol{\mu}$ should be sparse in order to filter out irrelevant sub-kernel matrices. To this end, we determine optimal weight vectors as the solution of the following constrained optimization problem:

$$\max_{\boldsymbol{\eta}\in\mathbb{R}^{M_x}, \boldsymbol{\mu}\in\mathbb{R}^{M_z}} \quad \mathbb{D}(K_x, K_z) = \boldsymbol{\eta}^{\mathrm{T}} M \boldsymbol{\mu} \tag{9a}$$

$$\text{subject to} \quad \boldsymbol{\eta} \geq 0,\ \boldsymbol{\mu} \geq 0,$$

$$\|\boldsymbol{\eta}\|_2 = \|\boldsymbol{\mu}\|_2 = 1, \tag{9b}$$

$$\|\boldsymbol{\eta}\|_1 \leq c_1, \|\boldsymbol{\mu}\|_1 \leq c_2, \tag{9c}$$

where $\|\mathbf{x}\|_p = (\sum_i |x_i|^p)^{1/p}$ is the L^p-norm of the vector $\mathbf{x}$ and c_1 and c_2 are parameters (See also "Parameter tuning by a permutation test" section). This optimization problem is an example of penalized matrix decomposition with non-negativity constraints [4]. Accordingly, we can obtain optimal weight coefficients by performing singular value decomposition of matrix M under constraints. In this process, the i-th left singular vector $\boldsymbol{\eta}^{(i)} = (\eta_1^{(i)}, \ldots, \eta_{M_x}^{(i)})^T$ as well as the right singular vector $\boldsymbol{\mu}^{(i)} = (\mu_1^{(i)}, \ldots, \mu_{M_z}^{(i)})^T$ are obtained iteratively by Algorithm 1.

Algorithm 1 Penalized Matrix Decomposition for Learning Kernels

Input: M (Eq. 8), regularization c_1 and c_2
for $i = 1$ **to** $rank(M)$ **do**
  initialize $\boldsymbol{\eta}^i$ to a first left singular vector of M
  repeat
    $\boldsymbol{\mu}^{(i)} \leftarrow \frac{S((\boldsymbol{\eta}^{(i)T} M)_+, \Delta)}{\|S((\boldsymbol{\eta}^{(i)T} M)_+, \Delta)\|_2}$
    $\boldsymbol{\eta}^{(i)} \leftarrow \frac{S((M\boldsymbol{\mu}^{(i)})_+, \Delta)}{\|S((M\boldsymbol{\mu}^{(i)})_+, \Delta)\|_2}$
  until Convergence
  compute i-th singular value as $\sigma_i \leftarrow \boldsymbol{\eta}^{(i)T} M \boldsymbol{\mu}^{(i)}$
  obtain residual as $M \leftarrow M - \sigma_i \boldsymbol{\eta}^{(i)} \boldsymbol{\mu}^{(i)T}$
end for
Output: $\{\boldsymbol{\mu}^{(i)}\}_{i=1}^{rank(M_x)}$ and $\{\boldsymbol{\eta}^{(i)}\}_{i=1}^{rank(M_z)}$

In Algorithm 1, S denotes the element-wise soft-thresholding operator: The m-th element of $S(\mathbf{a}, c)$ is given by $\mathrm{sign}(a_m)(|a_m| - c)_+$, where $(x)_+$ is x if $x \geq 0$ and 0 if $x < 0$. In each step, Δ is chosen by a binary search so that L1 constraints $\|\boldsymbol{\eta}\|_1 \leq c_1$ and $\|\boldsymbol{\mu}\|_1 \leq c_2$ are satisfied. In general, the above iteration does not necessarily converge to a global optimum. For each iteration, we initialize $\boldsymbol{\eta}^{(i)}$ with a non-sparse, left singular vector of M, following the previous study, to obtain reasonable solutions [4].

The second stage: kernel CCA

After learning kernels via penalized matrix decomposition as above, we perform the second stage of standard kernel CCA [8, 9] to obtain optimal coefficients $\boldsymbol{\alpha}^*$ and $\boldsymbol{\beta}^*$ (Eq. 3a and 3b) with parameter κ for each pair of singular vectors $\{\boldsymbol{\eta}^{(i)}, \boldsymbol{\mu}^{(i)}\}_{i=1}^{rank(M)}$. Given test kernel $\{K_{x,test}^{(m)}\}_{m=1}^{M_x}$

and $\{K^{(l)}_{z,test}\}^{M_z}_{l=1}$, test correlation corresponding to the i-th singular vectors is defined as correlation between $\sum^{M_x}_{m=1}\eta_m K^{(m)}_{x,test}\boldsymbol{\alpha}^*$ and $\sum^{M_z}_{l=1}\mu_l K^{(l)}_{z,test}\boldsymbol{\beta}^*$.

Practical solutions for TSKCCA implementation

TSKCCA still has several options for sub-kernels to be designed manually. In this study, we focus on feature-wise kernel and pair-wise kernel defined in the following sections.

Feature-wise kernel

Feature-wise kernel was introduced to perform feature-wise nonlinear Lasso [20]. In the previous study, using feature-wise kernels as sub-kernels in sparse MKL resulted in sparsity in terms of features since each sub-kernel corresponds to each feature. With x_{nm} and z_{nl} representing the m-th feature for $\mathbf{x}_n$ and l-th feature for $\mathbf{z}_n$, respectively, we adopt the following Gaussian kernel in this study:

$$[K^{(m)}_x]_{nn'} = \exp\left\{-\gamma_x(x_{nm}-x_{n'm})^2\right\} \tag{10a}$$

$$[K^{(l)}_z]_{nn'} = \exp\left\{-\gamma_z(z_{nl}-z_{n'l})^2\right\}, \tag{10b}$$

where γ_x and γ_z are width parameters. By applying feature-wise kernels, projection functions are restricted to additive models defined as $f^*(\mathbf{x}) = \sum^p_{m=1} f_m(\mathbf{x}_{\cdot m})$ and $g^*(\mathbf{z}) = \sum^q_{l=1} g_l(\mathbf{z}_{\cdot l})$, where $f_m : \mathbb{R} \to \mathbb{R}$ $(m = 1,\ldots,p)$ and $g_l : \mathbb{R} \to \mathbb{R}$ $(l = 1,\ldots,q)$ are certain nonlinear functions [2]. Note that the number of sub-kernels, M_x and M_z, are equivalent to the number of features, p and q, respectively.

Pair-wise kernel

We introduce pair-wise kernels as sub-kernels to consider cross-feature interactions among all possible pairs of features. Since the sparseness is induced to the weight of sub-kernels, the pair-wise kernels result in selecting relevant cross-feature interactions. Projection functions are defined as $f^*(\mathbf{x}) = \sum^p_{m<m'} f_{m,m'}(\mathbf{x}_{\cdot m}, \mathbf{x}_{\cdot m'})$ and $g^*(\mathbf{z}) = \sum^q_{l<l'} g_{l,l'}(\mathbf{z}_{\cdot l}, \mathbf{z}_{\cdot l'})$, where $f_{m,m'} : \mathbb{R}^2 \to \mathbb{R}$ and $g_{l,l'} : \mathbb{R}^2 \to \mathbb{R}$ are certain nonlinear functions with two dimensional inputs. Note that the number of sub-kernels, M_x and M_z, are, $p(p-1)/2$ and $q(q-1)/2$, respectively.

Preprocessing for MKL

We normalize the sub-kernels to have uniform variance in RKHS. This is an important procedure in the context of MKL because each feature-wise kernel has a different scale. This makes it difficult to evaluate weight coefficients [21]. To compensate for that, we calculate the variance σ^2 in RKHS as

$$\sigma^2 = \frac{1}{N}\sum^N_n \left\| \phi(\mathbf{x}_n) - \frac{1}{N}\sum^N_{n'}\phi(\mathbf{x}_{n'}) \right\|^2_2 \tag{11a}$$

$$= \frac{1}{N}\sum^N_n \phi(\mathbf{x}_n)^T\phi(\mathbf{x}_n) - \frac{1}{N^2}\sum^N_{n,n'}\phi(\mathbf{x}_{n'})^T\phi(\mathbf{x}_{n'}) \tag{11b}$$

$$= \frac{1}{N}\sum^N_n [K]_{nn} - \frac{1}{N^2}\sum^N_{n,n'}[K]_{nn'}. \tag{11c}$$

Dividing each sub-kernel by its variance $K \to \frac{K}{\sigma^2}$, we can achieve normalization of each sub-kernel.

Parameter tuning by a permutation test

When the kernel matrix K_x (or K_y) is full rank, as is typically our case, KCCA with a small κ ($\kappa \ll 1$) can always find a solution such that the maximum canonical correlation nearly equals one. This property makes it difficult to tune the regularization parameters for the first stage c_1 and c_2. To solve the issue, we introduce a simple heuristics.

The key idea is to conduct a permutation test for deciding whether to reject a null hypothesis that the maximal canonical correlation induced by i-th singular vectors is no more than those attained when $\mathbf{x}$ and $\mathbf{z}$ are statistically independent. Since the p-value of this test is interpreted as the deviance between the actual outcome and those expected under the null hypothesis, we use it as a score to evaluate the significance of i-th singular vectors where smaller p-value is more significant.

Algorithm 2 summarizes our implementation for the permutation test. Only for the first singular vectors $\boldsymbol{\eta}^{(1)}$ and $\boldsymbol{\mu}^{(1)}$, this procedure is applied to various pairs of (c_1, c_2) that satisfy the constraints of $1 \le c_1 \le \sqrt{M_x}$ and $1 \le c_2 \le \sqrt{M_y}$ [4]. Among them, the pair with the lowest p-value is chosen as the optimal parameters of c_1 and c_2.

Algorithm 2 A Permutation Test

Input: $\{K^{(m)}_x\}^{M_x}_{m=1}$, $\{K^{(l)}_z\}^{M_z}_{l=1}$, c_1, and c_2
$c = \mathrm{Cor}(\{K^{(m)}_x\}^{M_x}_{m=1}, \{K^{(l)}_z\}^{M_z}_{l=1}, \boldsymbol{\eta}^{(i)}, \boldsymbol{\mu}^{(i)})$
for $b = 1$ **to** B **do**
  permute the samples of $\mathbf{x}$ and calculate $\{\tilde{K}^{(m)}_x\}^{M_x}_{m=1}$
  obtain $\tilde{M}$ where $[\tilde{M}]_{ij} = \frac{\mathrm{Tr}(\tilde{K}^{(m)}_x H K^{(l)}_z H)}{(N-1)^2}$
  perform the first stage; matrix decomposition of $\tilde{M}$ to obtain $\tilde{\boldsymbol{\eta}}^{(i)}$ and $\tilde{\boldsymbol{\mu}}^{(i)}$
  calculate $c_b = \mathrm{Cor}(\{\tilde{K}^{(m)}_x\}^{M_x}_{m=1}, \{K^{(l)}_z\}^{M_z}_{l=1}, \tilde{\boldsymbol{\eta}}^{(i)}, \tilde{\boldsymbol{\mu}}^{(i)})$
end for
$p = \frac{\sum^B_{b=1} I(|c_b| > |c|)}{B+1}$
Output: p

For simplicity, other parameters, such as γ in the Gaussian kernel and κ in KCCA, are fixed heuristically. γ^{-1} is set to the median of the Euclidean distance between data points and κ is set to 0.02 as recommended in the previous study [9].

Results

In this section, we experimentally evaluate the performance of our proposed TSKCCA, SAFCCA [12], and other methods using synthetic data and nutrigenomic experimental data.

Dataset 1: single nonlinear association

To evaluate the ability to extract a single nonlinear association, we generated simple synthetic data which consisted of a single pair of relevant features in quadratic association and noise, in which standard CCA and KCCA are known to performance poorly [12]. Let $N(\mu, s^2)$ and $U(\mathcal{A})$ denote the normal distribution with mean μ, variance s^2, and uniform distribution supported in $\mathcal{A}$, respectively. The synthetic data were generated as

$$\begin{aligned}
\mathbf{x}_{.m} &\sim U([-0.5, 0.5]) \quad m = 1, \ldots, D \\
\mathbf{z}_{.1} &= \mathbf{x}_{.1}^2 + \boldsymbol{\epsilon} \\
\mathbf{z}_{.l} &\sim U([-0.5, 0.5]) \quad l = 2, \ldots, D \\
\boldsymbol{\epsilon} &\sim N(0, s^2),
\end{aligned}$$

where D was the total number of dimensions and ϵ was independent noise.

The optimal model in each method was trained using N training samples. Here, we assumed $c_1 = c_2$ in the range of $1 \leq c_1, c_2 \leq \frac{\sqrt{D}}{2}$ and obtained optimal values using a permutation test with $B = 100$. The test correlation was evaluated with separate 100 test samples, averaged over 100 simulation runs as we varied the number of dimensions, the sample size, and the noise level.

Figure 1 shows the test correlations achieved by TSKCCA and SAFCCA with different data dimensions D, sample size N, and noise level s. In the first stage, our method selected only two sub-kernels, corresponding to $\mathbf{x}_1$ and $\mathbf{z}_1$, among $2 \times D$ sub-kernels in the first stage, especially in the case of $N = 100$ and $N = 150$. As a result, it achieved better test correlation than SAFCCA, especially with high-dimensional data, indicating that our method was sufficiently robust.

In addition, Fig. 2 shows average computation time for each method over 100 simulation runs with dataset 1. Computation time of TSKCCA was comparable with that of SAFCCA, and could scale up with the feature size. Note that all the experiments were performed on a MacBook Pro with Intel Core i7 (2.9 GHz dual core processor with 4 MB L3 cache) with 8 GB main memory. All the simulation programs were implemented in MATLAB®.

Dataset 2: multiple nonlinear associations

To test whether our method could extract multiple nonlinear associations precisely, we generated the following data:

$$\begin{aligned}
\mathbf{x}_{.m} &\sim U([-0.5, 0.5]) \quad m = 1, \ldots, 25 \\
\mathbf{z}_{.1} &= \mathbf{x}_{.1} + \exp(-\mathbf{x}_{.4}^2) + \boldsymbol{\epsilon}_1 \\
\mathbf{z}_{.2} &= \mathbf{x}_{.2}^2 + \sin(\pi \mathbf{x}_{.5}/2) + \boldsymbol{\epsilon}_2 \\
\mathbf{z}_{.3} &= |\mathbf{x}_{.3}| + 1/(1 + \exp(-5\mathbf{x}_{.6})) + \boldsymbol{\epsilon}_3 \\
\mathbf{z}_{.l} &\sim U([-0.5, 0.5]) \quad l = 4, \ldots, 25 \\
\boldsymbol{\epsilon}_l &\sim N(0, 0.1^2) \quad l = 1, 2, 3.
\end{aligned}$$

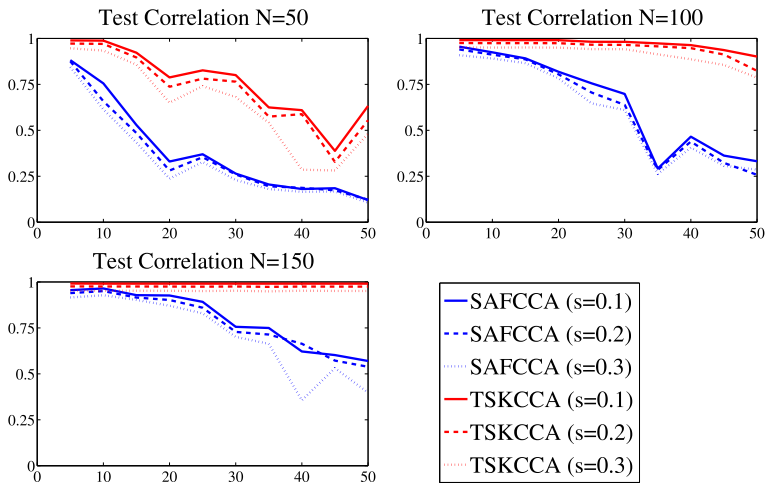

Fig. 1 Comparison of test correlation averaged over simulation runs in Data 1. The horizontal axis denotes the number of dimensions *D*, and the vertical axis denotes test correlations. The number of training samples is 50, 100, and 150. TSKCCA outperforms SAFCCA, especially with high-dimensional data

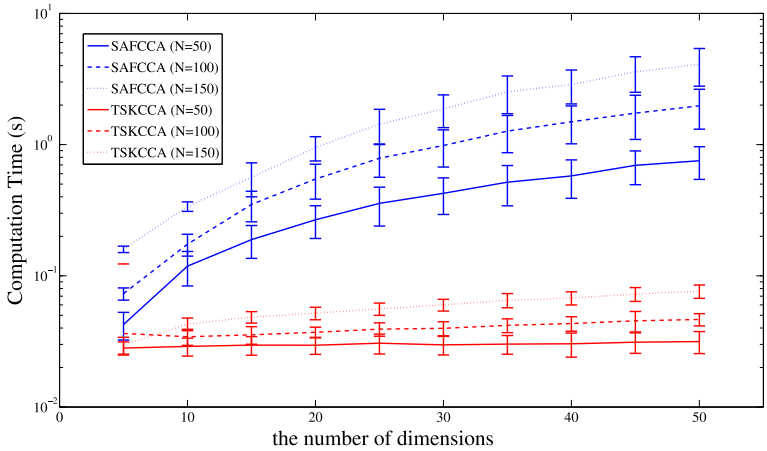

Fig. 2 Comparison of computation time for Data 1. The horizontal axis denotes the number of dimensions *D*, and the vertical axis denotes computation time in log-scale. The number of training samples is 50, 100, 150 for SAFCCA and TSKCCA. Computation time of TSKCCA is moderate and can be scaled

First, we performed a permutation test with $B = 1000$ for ten singular vectors $\{\boldsymbol{\eta}^{(i)}, \boldsymbol{\mu}^{(i)}\}_{i=1}^{10}$ corresponding to the ten highest singular values of M given by Eq. (8). P-values of the top three were significant ($p < 0.001$) and the rest were non-significant. This result suggests that only the three singular vectors included nonlinear associations.

Figure 3 shows the transformations $f(\mathbf{x})$ and $g(\mathbf{z})$ obtained with TSKCCA. In the first singular vectors, the contributions of η_1^1, η_4^1 and μ_1^1 were dominant, indicating that $\mathbf{x}_{\cdot 1}$, $\mathbf{x}_{\cdot 4}$ and $\mathbf{z}_{\cdot 1}$ were associated. The contributions of η_2^2, η_5^2 and μ_2^2 in the second singular vectors were also dominant, indicating that $\mathbf{x}_{\cdot 2}$, $\mathbf{x}_{\cdot 5}$ and $\mathbf{z}_{\cdot 2}$ were associated. Finally, the contributions of η_3^3, η_6^3 and μ_3^3 in the third singular vectors were dominant, indicating that $\mathbf{x}_{\cdot 3}$, $\mathbf{x}_{\cdot 6}$ and $\mathbf{z}_{\cdot 3}$ were associated. Some singular vectors averaged over 100 simulation runs are listed in Table 1. Our results suggest that TSKCCA achieved feature selection precisely.

We further evaluated test correlation, precision, and recall averaged over 20 simulation runs. Table 2 shows that SAFCCA failed to detect all relevant features because it is

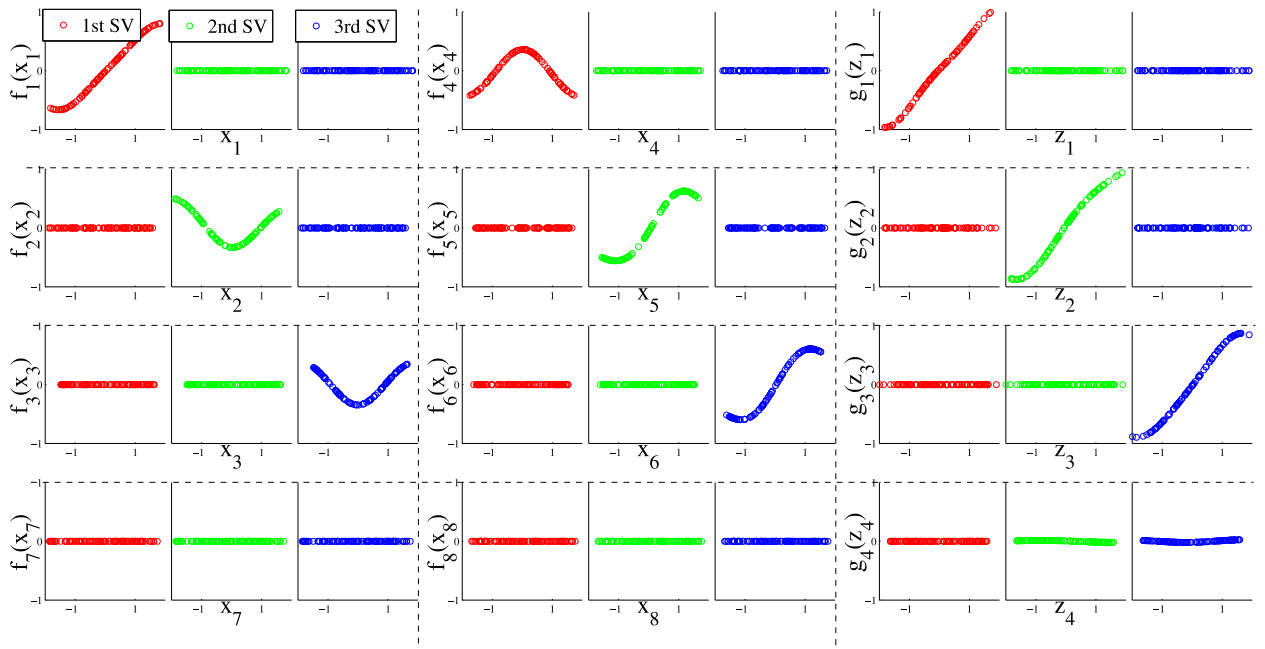

Fig. 3 Transformations $f(\mathbf{x})$ and $g(\mathbf{z})$ obtained with TSKCCA. The top three rows and the bottom row show the resulting functions corresponding to relevant and irrelevant features, respectively

Table 1 Feature selection through singular vectors (SVs) in data 2

	1st SV ($\eta^{(1)}$)	2nd SV ($\eta^{(2)}$)	3rd SV ($\eta^{(3)}$)
η_1	**0.98 (0.002)**	0.00 (0.018)	0.00 (0.001)
η_2	0.00 (0.003)	**0.21 (0.033)**	0.00 (0.001)
η_3	0.00 (0.001)	0.00 (0.010)	**0.22 (0.029)**
η_4	**0.22 (0.013)**	0.00 (0.017)	0.00 (0.005)
η_5	0.00 (0.000)	**0.98 (0.004)**	0.00 (0.005)
η_6	0.00 (0.004)	0.00 (0.002)	**0.98 (0.003)**
	1st SV ($\mu^{(1)}$)	2nd SV ($\mu^{(2)}$)	3rd SV ($\mu^{(3)}$)
μ_1	**0.99 (0.005)**	0.01 (0.022)	0.01 (0.014)
μ_2	0.01 (0.027)	**0.99 (0.004)**	0.01 (0.015)
μ_3	0.01 (0.024)	0.01 (0.018)	**0.99 (0.003)**
μ_4	0.01 (0.023)	0.01 (0.026)	0.01 (0.017)

These results show mean weight coefficients (standard deviation) in 100 simulation runs. Significant weight coefficients are bold faced

not able to obtain multiple canonical correlations, while our method detected 9 relevant sub-kernels out of 50 in the first stage in most runs. Note that the precision is the fraction of retrieved features that are relevant and recall is the fraction of relevant features that are retrieved.

Dataset 3: feature interactions

To assess the capability of TSKCCA in discovering nonlinear interactions, we generated data with a product term:

$$\begin{aligned} \mathbf{x}_{\cdot m} &\sim U([-0.5, 0.5]) \quad m = 1, \ldots, D \\ \mathbf{z}_{\cdot 1} &= \mathbf{x}_{\cdot 1}\mathbf{x}_{\cdot 2} + \boldsymbol{\epsilon} \\ \mathbf{z}_{\cdot l} &\sim U([-0.5, 0.5]) \quad l = 2, \ldots, D \\ \boldsymbol{\epsilon} &\sim N(0, 0.1^2), \end{aligned}$$

where D was the number of dimensions. For this dataset, we used feature-wise kernels and pair-wise kernels as sub-kernels in order to handle both single feature effects and cross-feature interactions like the term $\mathbf{x}_{\cdot 1}\mathbf{x}_{\cdot 2}$. There were $D + D \times (D-1)/2$ sub-kernels, the weight coefficients of which were optimized in our method.

Table 2 Comparison of test correlation, precision, and recall in data 2

	Correlation	Precision	Recall
TSKCCA	0.9670	0.9163	1
	0.9636		
	0.9732		
SAFCCA	0.7585	0.6350	0.4375

TSKCCA can identify most relevant features through three significant singular vectors, while SAFCCA can only identify a small set of them

First, to evaluate the performance of our method with feature-wise and pair-wise kernels, we obtained test correlations evaluated by individual test data ($N = 100$) in different numbers of dimensions D. Next, to evaluate the accuracy of feature selection of the model, we assessed recall and precision. Average test correlations, recall, and precision over 100 simulation runs are shown in Fig. 4. Our results illustrate that in the case of $D < 10$ (i.e. the number of sub-kernels is less than $10 + 10 \times 9/2 = 55$), our method successfully determined the relation between $\mathbf{z}_{\cdot 1}$ and $\mathbf{x}_{\cdot 1}\mathbf{x}_{\cdot 2}$.

Dataset 4: nutrigenomic data

We then analyzed a nutrigenomic dataset from a previous mouse study [22, 23]. In this study, expression of 120 genes in liver cells that would be relevant in the context of nutrition and concentrations of 21 hepatic fatty acids were measured on 20 wild-type mice and 20 PPARα-deficient mice. Mice of each genotype were fed 5 different diets with different levels of fat. For matrix notation, gene expression data were denoted by $X \in \mathbb{R}^{40\times 120}$, and data regarding concentrations of fatty acids was denoted by $Z \in \mathbb{R}^{40\times 21}$. Data were standardized to have a mean of zero and unit variance in each dimension. Several linear correlations between X and Z were detected by applying a regularized version of the linear CCA [5, 23].

First, we performed a permutation test for sparse CCA, KCCA, SAFCCA, and TSKCCA on parameters defined by equally-spaced grid points in order to identify significant associations in these data. In KCCA and SAFCCA, there were no significant associations; thus, we focused on sparse CCA and TSKCCA in the following analysis. We identified two significant linear associations in sparse CCA ($p < 0.001$ using a permutation test) and one nonlinear association in TSKCCA ($p = 0.0067$ using a permutation test) with $c_1 = 2.6257$ and $c_2 = 1.9275$.

Figures 5 and 6 show the results of feature selection of sparse CCA and TSKCCA, respectively. Genes selected by the first singular vector of our method have different expression levels in different genotypes (marked with asterisk), suggesting that our method successfully extracted the nonlinear correlation associated with genotypes.

For further analysis, cross-validation was performed in 100 runs. In each run, 40 samples were randomly split into 30 training samples used for fitting models and 10 validation samples used for evaluating the canonical correlation for fitted models. Figure 7 shows box plots of correlation coefficients in sparse CCA and TSKCCA. Left one represents the first canonical correlation coefficient in sparse CCA and right one represents correlation coefficient obtained with the first singular vectors. Significantly higher test correlation ($p < 10^{-6}$ with a t-test) were achieved by the first singular vectors of TSKCCA,

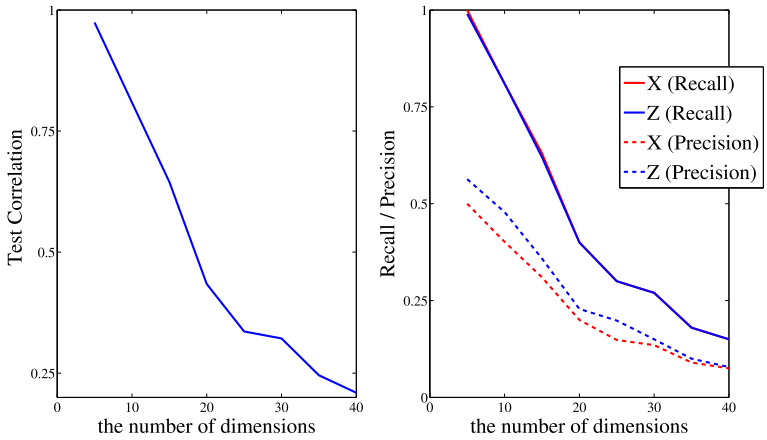

Fig. 4 The performance of pair-wise kernels in Data 3. (*Left*) Test correlations averaged over 100 simulation runs in different numbers of dimensions. (*Right*) Recall and precision averaged over 100 simulation runs in different numbers of dimensions. Our method successfully extracts nonlinear associations with relevant features

indicating that it avoided overfitting despite having nonlinearity.

To account for interactions between features into our model, we calculated pair-wise kernels for nutrigenomic data. Although the number of sub-kernels was huge ($120+120\times 119/2=7260$ sub-kernels for genes, $21+21\times 20/2=231$ sub-kernels for fatty acids), TSKCCA successfully extracted a significant association ($p<0.001$ using a permutation test). To evaluate the stability of feature selection, we performed TSKCCA on 1000 runs with data generated by random sampling of empirical data with replacement. Table 3 shows the frequencies of features (i.e. pairs of features) selected across 1000 runs, suggesting that *PMDCI* played an important role within the interactions.

Discussion

Other researchers have employed the sparse additive model [13] to extend KCCA to high-dimensional problems, and have defined two equivalent formulations, such as sparse additive functional CCA (SAFCCA) and sparse additive kernel CCA (SAKCCA) [12]. The former was defined in a second order Sobolev space and solved using the biconvex back-fitting procedure. The latter, defined in RKHS, was derived by applying representer theorem to the former. Given some function $f_m \in \mathbb{H}_m$, these algorithms optimize the additive model, $f_1 \in \mathbb{H}_1, f_2 \in \mathbb{H}_2, \ldots, f_p \in \mathbb{H}_p$. In contrast, our formulation supposes an additive kernel, such as $\sum \eta_m K_m$ associated with RKHS $\mathbb{H}_{add}$ and finds correlations in this space. This approach enables us to reveal multiple components of associations.

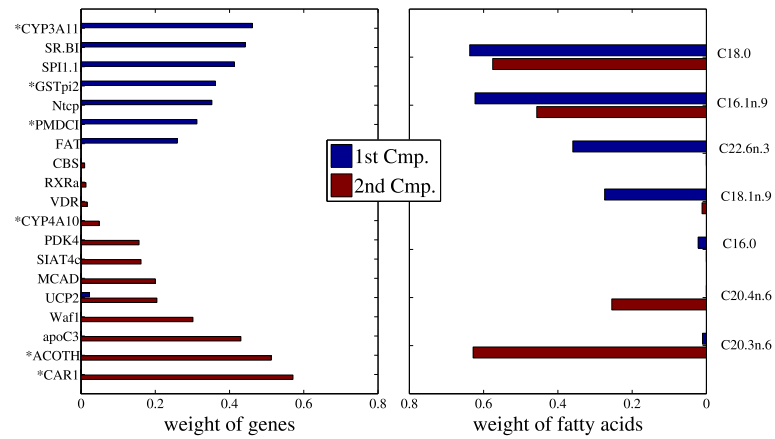

Fig. 5 Feature selection of sparse CCA in nutrigenomic data. *Left* and *right panels* show selected genes and fatty acids, respectively. Genes marked with asterisks show significantly different expression in different genotypes

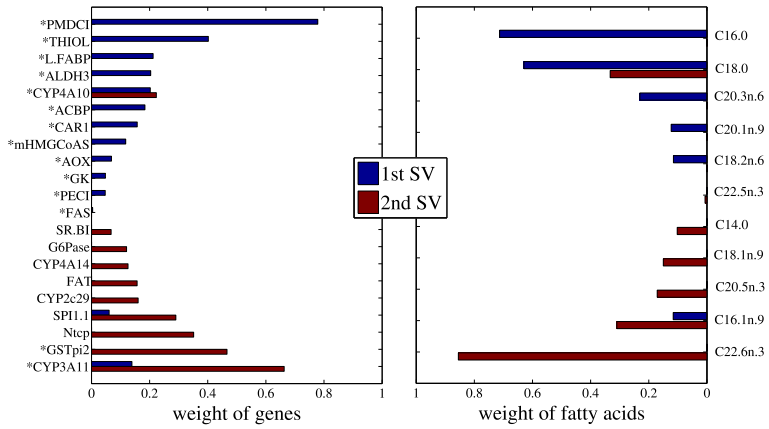

Fig. 6 Feature selection of TSKCCA using nutrigenomic data. *Left* and *right panels* show selected genes and fatty acids, respectively. Genes marked with asterisks show significantly different expression in different genotypes. The left panel shows that the 1st singular vector extracts nonlinear correlations associated with the genotype

Some problems specific to KCCA, such as choosing two parameters (i.e. regularization parameter κ and the width parameter γ) and the number of components, remain unsolved. While cross validation is applicable to set these values [24], they are fixed for simplicity in our study, based on the previous study [9].

Next, we discuss the validity of feature selection in nutrigenomic data performed using sparse CCA and TSKCCA. In the original study, the authors focused on the role of PPARα as a major transcriptional regulator of lipid metabolism and determined that PPARα regulates the expression of many genes in mouse liver under lower dietary fat conditions [22]. They provided a list of genes that have significantly different expression levels between wild-type and PPARα-deficient mice. While only a few genes selected by sparse CCA were included in the list, 13 out of 14 genes selected with the 1st singular vector in TSKCCA were included in the list. This result shows that TSKCCA successfully extracts meaningful nonlinear associations induced by PPARα-deficiency.

Moreover, in our analysis of pair-wise kernels, most of the frequently selected pairs of genes retained *PMDCI* known as a sort of enoyl-CoA isomerases involved in β-oxidation of polyunsaturated fatty acids. This implies that the interactions of *PMDCI* and other genes contribute to lipid metabolism in PPARα-deficient mice.

Many variants of sub-kernels, such as string kernels or graph kernels, can be employed in the same framework. In the field of bioinformatics, Yamanishi et al. adopted integrated KCCA (IKCCA), which exploited the simple sum of multiple kernels to combine many sorts of biological data [11]. This technique can be improved by optimizing weight coefficients of each kernel in the frame of TSKCCA. Finally, if kernels are defined on groups of features, it enables us to perform group-wise feature selection, just like group sparse CCA [25–27]. It is beneficial

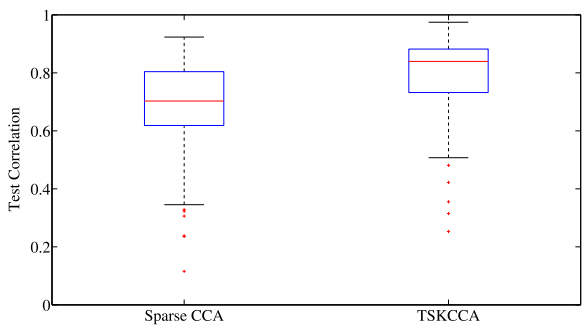

Fig. 7 Box plot of test correlations in nutrigenomic data. *Left* and *right panels* show the box plot of 100 times test correlation using sparse CCA and TSKCCA, respectively. TSKCCA achieves significantly higher test correlation through its first weight vector ($p < 10^{-6}$ with a t-test)

Table 3 Frequency of selection per sub-kernel corresponding to genes (*left*) and fatty acids (*right*) in nutrigenomic data

Genes/Pair of genes	Freq.	Fatty acids/Pair of fatty acids	Freq
PMDCI	643	C16.0-C18.0	622
CAR1-PMDCI	564	C18.0	485
PMDCI-THIOL	563	C16.0-C20.3n.6	429
ACBP-PMDCI	473	C16.0	340
L.FABP-PMDCI	451	C18.0-C20.3n.6	315
CYP4A10-PMDCI	379	-	-
CYP3A11-PMDCI	370	-	-
ALDH3-PMDCI	369	-	-
Ntcp-PMDCI	354	-	-
PMDCI-SPI1.1	347	-	-
ACOTH-PMDCI	330	-	-
PMDCI-SR.BI	306	-	-

to consider group-wise feature selection for biomarker detection problems.

Conclusions

This paper proposes a novel extension of kernel CCA that we call two-stage kernel CCA, which is able to identify multiple canonical variables from sparse features. This method optimizes the sparse weight coefficients of pre-specified sub-kernels as a sparse matrix decomposition before performing standard kernel CCA. This procedure enables us to achieve interpretability by removing irrelevant features in the context of nonlinear correlational analysis.

Through three numerical experiments, we have demonstrated that TSKCCA is more useful for higher dimensional data and for extracting multiple nonlinear associations than an existing method, SAFCCA. Using nutrigenomic data, our results show that TSKCCA can retrieve information about genotype and may reveal an interactive mechanism of lipid metabolism in PPARα-deficient mice.

Endnotes

[1] In this article, $[\cdot]_{nn'}$ denotes the (n, n')-th elements of the matrix enclosed by the brackets.

[2] In this article, $\mathbf{x}_{,m}$ denotes the m-th feature of $\mathbf{x}$.

Abbreviations

CCA: Canonical correlation analysis; HSIC: Hilbert-Schmidt independent criterion; KCCA: Kernel canonical correlation analysis; MKL: Multiple kernel learning; RKHS: Reproducing kernel Hilbert space; SAFCCA: Sparse additive functional canonical correlation analysis; TSKCCA: Two-stage kernel canonical correlation analysis

Acknowledgements

We thank Dr. Mitsuo Kawato, Dr. Noriaki Yahata, and Dr. Jun Morimoto for their valuable comments, and Dr. Steven D. Aird for editing the manuscript.

Funding

This work was supported by a Grant-in-Aid for Scientific Research on Innovative Areas: Artificial Intelligence and Brain Science (16H06563), the Strategic Research Program for Brain Sciences from Japan Agency for Medical Research and Development, AMED, and Okinawa Institute of Science and Technology Graduate University.

Availability of data and materials

The datasets analyzed during the current study are available from https://cran.r-project.org/web/packages/CCA.
Our code for TSKCCA and synthetic data is available in https://github.com/kosyoshida/TSKCCA.

Authors' contributions

KY designed the model and developed the algorithm. KY, JY, and KD participated in writing of the manuscript. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Consent for publication

Not applicable.

Ethics approval and consent to participate

Not applicable.

Author details

[1] Graduate School of Informatics, Kyoto University, Kyoto, Japan. [2] Neural Computation Unit, Okinawa Institute of Science and Technology, Okinawa, Japan. [3] Graduate School of Information Science, Nara Institute of Science and Technology, Nara, Japan.

Received: 14 October 2016 Accepted: 8 February 2017
Published online: 14 February 2017

References

1. Hotelling H. Relations between two sets of variates. Biometrika. 1936;28:321–77.
2. Yamanishi Y, Vert JP, Kanehisa M. Protein network inference from multiple genomic data: a supervised approach. Bioinformatics. 2004;20(suppl 1):363–70.
3. Waaijenborg S, Verselewel de Witt Hamer PC, Zwinderman AH. Quantifying the association between gene expressions and dna-markers by penalized canonical correlation analysis. Stat Appl Genet Mol Biol. 2008;7(1):3.
4. Witten DM, Tibshirani R, Hastie T. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. Biostatistics. 2009;10(3):515–34.
5. González I, Déjean S, Martin PG, Gonçalves O, Besse P, Baccini A. Highlighting relationships between heterogeneous biological data through graphical displays based on regularized canonical correlation analysis. J Biol Syst. 2009;17(02):173–99.
6. Wilms I, Croux C. Sparse canonical correlation analysis from a predictive point of view. Biom J. 2015;57(5):834–51.
7. Parkhomenko E, Tritchler D, Beyene J, et al. Sparse canonical correlation analysis with application to genomic data integration. Stat Appl Genet Mol Biol. 2009;8(1):1–34.
8. Akaho S. A kernel method for canonical correlation analysis. In: Proceedings of the International Meeting of the Psychometric Society (IMPS2001). Osaka: Springer Japan; 2001.
9. Bach FR, Jordan MI. Kernel independent component analysis. J Mach Learn Res. 2003;3:1–48.
10. Vert JP, Kanehisa M. Graph-driven feature extraction from microarray data using diffusion kernels and kernel cca. In: Advances in Neural Information Processing Systems. Vancouver: NIPS Foundation; 2002. p. 1425–432.
11. Yamanishi Y, Vert JP, Nakaya A, Kanehisa M. Extraction of correlated gene clusters from multiple genomic data by generalized kernel canonical correlation analysis. Bioinformatics. 2003;19(suppl 1):323–30.
12. Balakrishnan S, Puniyani K, Lafferty JD. Sparse additive functional and kernel CCA. In: Proceedings of the 29th International Conference on

Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012. The International Machine Learning Society; 2012.
13. Ravikumar P, Lafferty J, Liu H, Wasserman L. Sparse additive models. J R Stat Soc Ser B Stat Methodol. 2009;71(5):1009–30.
14. Gretton A, Bousquet O, Smola AJ, Schölkopf B. Measuring statistical dependence with hilbert-schmidt norms. In: Algorithmic Learning Theory, 16th International Conference, ALT 2005, Singapore, October 8-11, 2005, Proceedings. Springer; 2005. p. 63–77.
15. Cristianini N, Shawe-taylor J, Elisseeff A, Kandola J. On kernel-target alignment. In: Advances in Neural Information Processing Systems 14. Vancouver; 2001. Citeseer.
16. Chang B, Krüger U, Kustra R, Zhang J. Canonical correlation analysis based on hilbert-schmidt independence criterion and centered kernel target alignment. In: Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013. The International Machine Learning Society; 2013. p. 316–24.
17. Lanckriet GR, Cristianini N, Bartlett P, Ghaoui LE, Jordan MI. Learning the kernel matrix with semidefinite programming. J Mach Learn Res. 2004;5: 27–72.
18. Bach FR, Lanckriet GR, Jordan MI. Multiple kernel learning, conic duality, and the smo algorithm. In: Proceedings of the Twenty-first International Conference on Machine Learning. ACM; 2004. p. 6.
19. Cortes C, Mohri M, Rostamizadeh A. Two-stage learning kernel algorithms. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel. The International Machine Learning Society; 2010. p. 239–46.
20. Yamada M, Jitkrittum W, Sigal L, Xing EP, Sugiyama M. High-dimensional feature selection by feature-wise kernelized lasso. Neural Comput. 2014;26(1):185–207.
21. Kloft M, Brefeld U, Sonnenburg S, Zien A. Lp-norm multiple kernel learning. J Mach Learn Res. 2011;12:953–97.
22. Martin P, Guillou H, Lasserre F, Déjean S, Lan A, Pascussi J, Sancristobal M, Legrand P, Besse P, Pineau T. Novel aspects of pparalpha-mediated regulation of lipid and xenobiotic metabolism revealed through a nutrigenomic study. Hepatol (Baltim Md.) 2007;45(3):767–77.
23. González I, Déjean S, Martin PG, Baccini A, et al. Cca: An r package to extend canonical correlation analysis. J Stat Softw. 2008;23(12):1–14.
24. Leurgans SE, Moyeed RA, Silverman BW. Canonical correlation analysis when the data are curves. R Stat Soc Ser B Methodol, J. 1993;725–40.
25. Yuan M, Lin Y. Model selection and estimation in regression with grouped variables. J R Stat Soc Ser B Methodol. 2006;68(1):49–67.
26. Bach FR. Consistency of the group lasso and multiple kernel learning. J Mach Learn Res. 2008;9:1179–225.
27. Lin D, Zhang J, Li J, Calhoun VD, Deng HW, Wang YP. Group sparse canonical correlation analysis for genomic data integration. BMC Bioinforma. 2013;14(1):245.