

Canonical Correlation Analysis for Multilabel Classification: A Least-Squares Formulation, Extensions, and Analysis

Liang Sun, Shuiwang Ji, *Student Member, IEEE*, and Jieping Ye, *Member, IEEE*

Abstract—Canonical Correlation Analysis (CCA) is a well-known technique for finding the correlations between two sets of multidimensional variables. It projects both sets of variables onto a lower-dimensional space in which they are maximally correlated. CCA is commonly applied for supervised dimensionality reduction in which the two sets of variables are derived from the data and the class labels, respectively. It is well-known that CCA can be formulated as a least-squares problem in the binary class case. However, the extension to the more general setting remains unclear. In this paper, we show that under a mild condition which tends to hold for high-dimensional data, CCA in the multilabel case can be formulated as a least-squares problem. Based on this equivalence relationship, efficient algorithms for solving least-squares problems can be applied to scale CCA to very large data sets. In addition, we propose several CCA extensions, including the sparse CCA formulation based on the 1-norm regularization. We further extend the least-squares formulation to partial least squares. In addition, we show that the CCA projection for one set of variables is independent of the regularization on the other set of multidimensional variables, providing new insights on the effect of regularization on CCA. We have conducted experiments using benchmark data sets. Experiments on multilabel data sets confirm the established equivalence relationships. Results also demonstrate the effectiveness and efficiency of the proposed CCA extensions.

Index Terms—Canonical correlation analysis, least squares, multilabel learning, partial least squares, regularization



1 INTRODUCTION

CANONICAL Correlation Analysis (CCA) [1] is a well-known technique for finding the correlations between two sets of multidimensional variables. It makes use of two views of the same set of objects and projects them onto a lower-dimensional space in which they are maximally correlated. CCA has been applied successfully in various applications [2], [3]. One popular use of CCA is for supervised learning, in which one view is derived from the data and the other view is derived from the class labels. In this setting, the data can be projected onto a lower-dimensional space directed by the label information. Such a formulation is particularly appealing in the context of dimensionality reduction for multilabel data [4].

Multivariate Linear Regression (MLR) that minimizes the sum-of-squares cost function is a well-studied technique for regression problems. It can also be applied to classification problems by defining an appropriate class indicator matrix [5], [6]. The solution to MLR based on least squares can be obtained by solving a linear system of equations. A number of algorithms, including the conjugate gradient algorithm, can be applied to solve it efficiently [7]. Furthermore, the least-squares formulation can be readily extended using the regularization technique. For example, 1-norm regularization

can be incorporated into the least-squares formulation to control model complexity and improve sparseness [8]. Sparseness often leads to easy interpretation and a good generalization ability. It has been used successfully in several algorithms, including Principal Component Analysis [9] and Support Vector Machines [10].

In contrast to least squares, CCA involves a generalized eigenvalue problem, which is computationally more expensive to solve [11]. Furthermore, it is challenging to derive sparse CCA as it involves a difficult sparse generalized eigenvalue problem. Convex relaxation of sparse CCA has been studied in [12], where the exact sparse CCA formulation has been relaxed in several steps. On the other hand, an interesting connection between least squares and CCA has been established in the literature. In particular, CCA has been shown to be equivalent to Fisher Linear Discriminant Analysis (LDA) for binary-class problems [13]. Meanwhile, it is well known that LDA is also equivalent to least squares in this case [5], [6]. Thus, CCA can be formulated as a least-squares problem for binary-class problems. In practice, the multilabel problems are very prevalent. It is therefore tempting to investigate their relationship in the more general setting.

In this paper, we study the relationship between CCA and least squares for multilabel problems. We show that, under a mild condition which tends to hold for high-dimensional data, CCA can be formulated as a least-squares problem by constructing a specific class indicator matrix. Based on this equivalence relationship, we propose several CCA extensions, including sparse

• The authors are with the Department of Computer Science and Engineering and the Center for Evolutionary Medicine and Informatics (CEMI) of The Biodesign Institute, Arizona State University, Tempe, AZ 85287. E-mail: {sun.liang, shuiwang.ji, jieping.ye}@asu.edu

CCA using the 1-norm regularization. We show that the least-squares formulation of CCA and its extensions can be solved efficiently. For example, the equivalent least-squares formulation and the 2-norm regularized extension can be computed using the iterative conjugate gradient algorithm LSQR [14], which can handle very large-scale problems. We extend the least-squares formulation to Orthonormalized Partial Least Squares (OPLS), a variant of Partial Least Squares (PLS), by establishing the equivalence relationship between CCA and OPLS. In addition, we analyze the effect of regularization on CCA. In particular, we show that the CCA projection for one set of variables is independent of the regularization on the other set of multidimensional variables, elucidating the effect of regularization on CCA. Furthermore, it can be shown that our analysis can be extended to kernel-induced feature space. More details are provided in the supplementary file, which can be found in the Computer Society Digital Library at <http://doi.ieeecomputersociety.org/10.1109/TPAMI.2010.160>.

Notations. The number of training samples, the data dimensionality, and the number of labels are denoted by n , d , and k , respectively. $x_i \in \mathbb{R}^d$ denotes the i th observation and $y_i \in \mathbb{R}^k$ encodes the corresponding label information. Let $X = [x_1, \dots, x_n] \in \mathbb{R}^{d \times n}$ be the data matrix and $Y = [y_1, \dots, y_n] \in \mathbb{R}^{k \times n}$ be the class label matrix. We assume that both $\{x_i\}_{i=1}^n$ and $\{y_i\}_{i=1}^n$ are centered, i.e., $\sum_{i=1}^n x_i = 0$, and $\sum_{i=1}^n y_i = 0$. $\|A\|_F$ denotes the Frobenius norm of matrix A . I is the identity matrix, and e is a vector of all ones.

2 BACKGROUND AND RELATED WORK

In this section we review CCA, least squares, and some related work.

2.1 Canonical Correlation Analysis

In CCA two different representations of the same set of objects are given, and a projection is computed for each representation such that they are maximally correlated in the dimensionality-reduced space. Formally, CCA computes two projection vectors, $w_x \in \mathbb{R}^d$ and $w_y \in \mathbb{R}^k$, such that the correlation coefficient

$$\rho = \frac{w_x^T X Y^T w_y}{\sqrt{(w_x^T X X^T w_x)(w_y^T Y Y^T w_y)}} \quad (1)$$

is maximized. Since ρ is invariant to the scaling of w_x and w_y , CCA can be formulated equivalently as

$$\begin{aligned} \max_{w_x, w_y} \quad & w_x^T X Y^T w_y \\ \text{subject to} \quad & w_x^T X X^T w_x = 1, \\ & w_y^T Y Y^T w_y = 1. \end{aligned} \quad (2)$$

In the following, we assume that $Y Y^T$ is nonsingular. It can be shown that w_x can be obtained by solving the

following optimization problem:

$$\begin{aligned} \max_{w_x} \quad & w_x^T X Y^T (Y Y^T)^{-1} Y X^T w_x \\ \text{subject to} \quad & w_x^T X X^T w_x = 1. \end{aligned} \quad (3)$$

Both formulations in Eqs. (2) and (3) attempt to find the eigenvectors corresponding to top eigenvalues of the following generalized eigenvalue problem:

$$X Y^T (Y Y^T)^{-1} Y X^T w_x = \eta X X^T w_x, \quad (4)$$

where η is the eigenvalue corresponding to the eigenvector w_x . It has also been shown that multiple projection vectors under certain orthonormality constraints consists of the top ℓ eigenvectors of the generalized eigenvalue problem in Eq. (4) [2].

In regularized CCA (rCCA), two regularization terms, $\lambda_x I$ and $\lambda_y I$, with $\lambda_x > 0$, $\lambda_y > 0$, are added in Eq. (2) to prevent the overfitting and avoid the singularity of $X X^T$ and $Y Y^T$ [2], [15]. Specifically, rCCA solves the following generalized eigenvalue problem:

$$X Y^T (Y Y^T + \lambda_y I)^{-1} Y X^T w_x = \eta (X X^T + \lambda_x I) w_x. \quad (5)$$

2.2 Least Squares for Regression and Classification

In regression, we are given a training set $\{(x_i, t_i)\}_{i=1}^n$, where $x_i \in \mathbb{R}^d$ is the observation and $t_i \in \mathbb{R}^k$ is the corresponding target. We assume that both the observations and the targets are centered. As a result, the intercept in regression can be eliminated. In this case, the least-squares method can be applied to compute the projection matrix W by minimizing the following sum-of-squares cost function:

$$\min_W f(W) = \sum_{i=1}^n \|W^T x_i - t_i\|_2^2 = \|W^T X - T\|_F^2, \quad (6)$$

where $T = [t_1, \dots, t_n] \in \mathbb{R}^{k \times n}$. It is well known that the optimal projection matrix is given by [5], [6]

$$W_{LS} = (X X^T)^\dagger X T^T, \quad (7)$$

where $(X X^T)^\dagger$ denotes the pseudo-inverse of $X X^T$.

The least-squares formulation can also be applied for classification problems. In the general multiclass case, we are given a data set consisting of n samples $\{(x_i, y_i)\}_{i=1}^n$, where $x_i \in \mathbb{R}^d$, and $y_i \in \{1, 2, \dots, k\}$ denotes the class label of the i th sample, and $k > 2$. To apply the least-squares formulation to the multiclass case, the 1-of- k binary coding scheme is usually employed to apply a vector-valued class code to each data point [5]. The solution depends on the choice of class indicator matrix. Several class indicator matrices have been proposed in the literature [6].

2.3 Related Work

The inherent relationship between least squares and several other models has been established in the past.

In particular, it is a classical result that LDA and least-squares problems are equivalent for binary-class problems [5]. Recently, this equivalence relationship was extended to the multiclass case by defining a specific class indicator matrix [16]. CCA has been shown to be equivalent to LDA for multiclass problems [13]. Thus, CCA is equivalent to least squares in the multiclass case. We show in the next section that under a mild condition, CCA can be formulated as a least-squares problem for the more general settings, i.e., multilabel problems when one of the views used in CCA is derived from the labels.

3 RELATIONSHIP BETWEEN CCA AND LEAST SQUARES FOR MULTILABEL CLASSIFICATION

In this section we investigate the relationship between CCA and least squares in the multilabel case. Due to space constraints, all proofs are provided in the supplementary file, which can be found on the Computer Society Digital Library at <http://doi.ieeecomputersociety.org/10.1109/TPAMI.2010.160>.

We first define four matrices for our derivation:

$$H = Y^T(Y Y^T)^{-\frac{1}{2}} \in \mathbb{R}^{n \times k}, \quad (8)$$

$$C_{XX} = X X^T \in \mathbb{R}^{d \times d}, \quad (9)$$

$$C_{HH} = X H H^T X^T \in \mathbb{R}^{d \times d}, \quad (10)$$

$$C_{DD} = C_{XX} - C_{HH} \in \mathbb{R}^{d \times d}. \quad (11)$$

Note that we assume $n \gg k$ and $\text{rank}(Y) = k$ for multilabel problems. Thus, $(Y Y^T)^{-\frac{1}{2}}$ is well-defined. It follows from the definitions above that the solution to CCA can be expressed as the eigenvectors corresponding to top eigenvalues of the matrix $C_{XX}^\dagger C_{HH}$.

3.1 Basic Matrix Properties

In this section, we study the basic properties of the matrices involved in the following discussion. Following the definition of H in Eq. (8), we have:

Lemma 3.1: Let H be defined as in Eq. (8) and let $\{y_i\}_{i=1}^n$ be centered, i.e., $\sum_{i=1}^n y_i = 0$. Then, we have

- H has orthonormal columns, i.e., $H^T H = I_k$;
- $H^T e = 0$.

Given $H \in \mathbb{R}^{n \times k}$ with orthonormal columns, there exists $D \in \mathbb{R}^{n \times (n-k)}$ such that $[H, D] \in \mathbb{R}^{n \times n}$ is an orthogonal matrix [7], that is

$$I_n = [H, D][H, D]^T = H H^T + D D^T.$$

It follows that $C_{DD} = C_{XX} - C_{HH} = X D D^T X^T$. Let the Singular Value Decomposition (SVD) of X be

$$X = U \Sigma V^T = [U_1, U_2] \text{diag}(\Sigma_r, 0) [V_1, V_2]^T = U_1 \Sigma_r V_1^T,$$

where $r = \text{rank}(X)$, U and V are orthogonal matrices, $\Sigma \in \mathbb{R}^{d \times n}$, $U_1 \in \mathbb{R}^{d \times r}$, $U_2 \in \mathbb{R}^{d \times (d-r)}$, $V_1 \in \mathbb{R}^{n \times r}$, $V_2 \in \mathbb{R}^{n \times (n-r)}$, and $\Sigma_r \in \mathbb{R}^{r \times r}$. It is clear that U_2 lies in the null space of X^T , that is,

$$X^T U_2 = 0. \quad (12)$$

3.2 Computing CCA via Eigendecomposition

Recall that the solution to CCA consists of the top ℓ eigenvectors of the matrix $C_{XX}^\dagger C_{HH}$. We next show how to compute these eigenvectors. Define the matrix $A \in \mathbb{R}^{r \times k}$ by

$$A = \Sigma_r^{-1} U_1^T X H = V_1^T H. \quad (13)$$

Let the SVD of A be $A = P \Sigma_A Q^T$, where $P \in \mathbb{R}^{r \times r}$ and $Q \in \mathbb{R}^{k \times k}$ are orthogonal and $\Sigma_A \in \mathbb{R}^{r \times k}$ is diagonal. Then,

$$A A^T = P \Sigma_A \Sigma_A^T P^T. \quad (14)$$

The eigendecomposition of the matrix $C_{XX}^\dagger C_{HH}$ is summarized in the following theorem:

Theorem 3.1: There are k nonzero eigenvalues for the matrix $C_{XX}^\dagger C_{HH}$. Specifically, the solution to CCA, which consists of the eigenvectors corresponding to the top ℓ ($\ell \leq k$) eigenvalues of $C_{XX}^\dagger C_{HH}$, is given by

$$W_{CCA} = U_1 \Sigma_r^{-1} P_\ell, \quad (15)$$

where P_ℓ contains the first ℓ columns of P .

3.3 Equivalence of CCA and Least Squares

Consider the class indicator matrix \tilde{T} defined as follows:

$$\tilde{T} = (Y Y^T)^{-\frac{1}{2}} Y = H^T. \quad (16)$$

It follows from Eq. (7) that the solution to the least-squares problem for the given \tilde{T} is

$$W_{LS} = (X X^T)^\dagger X H = U_1 \Sigma_r^{-1} P \Sigma_A Q^T. \quad (17)$$

It is clear from Eqs. (15) and (17) that the difference between CCA and least squares lies in Σ_A and Q^T .

We next show that all the diagonal elements of Σ_A are ones under a mild condition, that is, $\text{rank}(X) = n-1$ and $\text{rank}(Y) = k$. Note that the first condition is equivalent to requiring that the original data points are linearly independent before centering, which tends to hold for high-dimensional data. Before presenting the main result summarized in Theorem 3.2 below, we have the following lemma:

Lemma 3.2: Assume

$$\text{rank}(C_{XX}) + s = \text{rank}(C_{HH}) + \text{rank}(C_{DD}),$$

for some non-negative integer s . Then for the matrix $\hat{\Sigma}_A = \Sigma_A \Sigma_A^T = \text{diag}(a_1, a_2, \dots, a_r) \in \mathbb{R}^{r \times r}$, we have

$$1 = \dots = a_{f-s} > a_{f-s+1} \geq \dots \geq a_f > a_{f+1} = \dots = 0.$$

where $f = \text{rank}(\Sigma_A)$.

Theorem 3.2: Assume that $\text{rank}(X) = n-1$ and $\text{rank}(Y) = k$ for multilabel problems. Then, we have $\text{rank}(C_{XX}) = n-1$, $\text{rank}(C_{HH}) = k$, and $\text{rank}(C_{DD}) = n-k-1$. Thus, s defined in Lemma 3.2 equals to zero and

$$1 = a_1 = \dots = a_k > a_{k+1} = \dots = a_r = 0.$$

This implies that all diagonal elements of Σ_A are ones.

Since $\text{rank}(\Sigma_A) = k$, $C_{XX}^\dagger C_{HH}$ contains k nonzero eigenvalues. If we choose $\ell = k$, then

$$W_{CCA} = U_1 \Sigma_r^{-1} P_k. \quad (18)$$

The only difference between W_{LS} and W_{CCA} lies in the orthogonal matrix Q^T in W_{LS} .

In practice, we can use both W_{CCA} and W_{LS} to project the original data onto a lower-dimensional space before classification. For classifiers based on the Euclidean distance, the orthogonal transformation Q^T will not affect the classification performance since the Euclidean distance is invariant to any orthogonal transformation. Some well-known algorithms satisfying this property include the k -Nearest-Neighbor (k NN) algorithm [6] based on the Euclidean distance and the linear Support Vector Machines (SVM) [17]. In the following, the equivalent least-squares CCA formulation is called ‘‘LS-CCA’’.

4 LEAST-SQUARES EXTENSIONS OF CCA

Based on the equivalence relationship established in the last section, the classical CCA formulation can be extended using the regularization technique, which is commonly used to control the complexity and improve the generalization performance of models. Similar to ridge regression [6], we obtain the 2-norm regularized least-squares CCA formulation (called ‘‘LS-CCA₂’’) which minimizes the following objective function by using the target matrix \tilde{T} in Eq. (16):

$$L_2(W, \lambda) = \sum_{j=1}^k \left(\sum_{i=1}^n (x_i^T w_j - \tilde{T}_{ij})^2 + \lambda \|w_j\|_2^2 \right),$$

where $W = [w_1, \dots, w_k]$, and $\lambda > 0$ is the regularization parameter.

It is well known that sparseness can often be achieved by penalizing the 1-norm of the variables [8]. It has been introduced into the least-squares formulation and the resulting model is called lasso [8]. Based on the established equivalence relationship between CCA and least squares, we derive the 1-norm regularized least-squares CCA formulation (called ‘‘LS-CCA₁’’) which minimizes the following objective function:

$$L_1(W, \lambda) = \sum_{j=1}^k \left(\sum_{i=1}^n (x_i^T w_j - \tilde{T}_{ij})^2 + \lambda \|w_j\|_1 \right).$$

LS-CCA₁ can be solved efficiently using state-of-the-art algorithms [18], [19]. Furthermore, the entire solution path for all values of τ can be computed by the Least Angle Regression algorithm [20].

5 EFFICIENT IMPLEMENTATIONS OF CCA

Recall that we deal with the generalized eigenvalue problem in Eq. (4) to solve CCA, although, in our theoretical derivation, an equivalent eigenvalue problem is used instead. Large-scale generalized eigenvalue problems are known to be much harder than regular

Algorithm 1 Efficient Implementation of CCA via LSQR

Input: X, Y

Compute matrix $H = Y^T (Y Y^T)^{-\frac{1}{2}}$ based on the SVD of Y .

Regress X on $\tilde{T} = H^T$ using LSQR.

eigenvalue problems [11], [21]. There are two options to transform the problem in Eq. (4) into a standard eigenvalue problem [21]: (1) factor XX^T ; and (2) employ the standard Lanczos algorithm for the matrix $(XX^T)^{-1} X H H^T X^T$ using the XX^T inner product. The second option has its own issue for singular matrices, which is the case for high-dimensional problems with a small regularization. Thus, in this paper, we factor XX^T and solve a symmetric eigenvalue problem using the Lanczos algorithm.

The equivalent least-squares formulation leads to an efficient implementation. The pseudocode of the algorithm is given in Algorithm 1. The complexity of the first step is $O(nk^2)$. In the second step, we solve k least-squares problems. In our implementation, we use the LSQR algorithm proposed in [14], which is an implementation of a conjugate gradient type method for solving sparse least-squares problems. Note that the original matrix $X \in \mathbb{R}^{d \times n}$ is sparse in many applications such as text document modeling. However, after centering, X is no longer sparse. In order to keep the sparseness of X , the vector x_i is augmented by an additional component as $\tilde{x}_i^T = [1, x_i^T]$. This new component acts as the intercept for least squares. The extended X is denoted as $\tilde{X} \in \mathbb{R}^{(d+1) \times n}$, and the revised least-squares problem is expressed as

$$\min_{\tilde{W}} \|\tilde{W}^T \tilde{X} - \tilde{T}\|_F^2, \quad (19)$$

where $\tilde{W} \in \mathbb{R}^{(d+1) \times k}$. For a new data point $x \in \mathbb{R}^d$, its projection is given by $\tilde{W}^T [1; x]$.

For a dense data matrix, the computational cost involved in each iteration of LSQR is $O(3n + 5d + 2dn)$ [14]. Since the least-squares problems are solved k times, the overall cost of LSQR is $O(Nk(3n + 5d + 2dn))$, where N is the total number of iterations. When the matrix \tilde{X} is sparse, the cost is notably reduced. Suppose the number of nonzero elements in \tilde{X} is z . The overall cost of LSQR is reduced to $O(Nk(3n + 5d + 2z))$. In summary, the total time complexity for solving the least-squares formulation via LSQR is $O(nk^2 + Nk(3n + 5d + 2z))$ when \tilde{X} is sparse.

6 EXTENSIONS OF THE LEAST-SQUARES FORMULATION

Recall that CCA seeks a pair of linear transformations, one for each set of variables, such that the data are maximally correlated in the transformed space. In contrast, Partial Least Squares (PLS) finds the directions of maximum covariance. Covariance and correlation are

two different statistical measures for quantifying how variables covary. It has been shown that there is a close connection between PLS and CCA in discrimination [22]. In [23] and [24], a unified framework for PLS and CCA is developed, and CCA and Orthonormalized Partial Least Squares (OPLS) [25], a variant of PLS, can be considered as special cases of the unified framework by choosing different values of regularization parameters. However, the intrinsic equivalence relationship between CCA and OPLS has not been studied yet. In this section, we show the equivalence relationship between CCA and OPLS, thus extending the least-squares formulation to OPLS. The following optimization problem is considered in OPLS:

$$\begin{aligned} \max_W \quad & \text{tr}(W^T XY^T YX^T W) \\ \text{subject to} \quad & W^T XX^T W = I. \end{aligned} \quad (20)$$

The optimal W is given by the eigenvectors of the following generalized eigenvalue problem:

$$XH_{pls}H_{pls}^T X^T w = \eta XX^T w, \quad (21)$$

where the matrix H_{pls} is defined as:

$$H_{pls} = Y^T \in \mathbb{R}^{n \times k}. \quad (22)$$

Recall that in CCA, the matrix $A = V_1^T H$ is defined in Eq. (13) and its SVD is given by $A = P \Sigma_A Q^T$. Similarly, we define $A_{pls} = V_1^T H_{pls}$. Let the thin SVD of A_{pls} be $A_{pls} = P_{pls} \Sigma_{pls} Q_{pls}^T$, where $P_{pls} \in \mathbb{R}^{r \times k}$, $\Sigma_{pls} \in \mathbb{R}^{k \times k}$, $Q_{pls} \in \mathbb{R}^{k \times k}$. We have the following result on the range space of $V_1^T H_{pls}$:

Lemma 6.1: Let $A = V_1^T H$ be defined in Eq. (13), and $A_{pls} = V_1^T H_{pls} \in \mathbb{R}^{r \times k}$. Then, $\mathcal{R}(A) = \mathcal{R}(A_{pls})$, where $\mathcal{R}(A)$ and $\mathcal{R}(A_{pls})$ are the range spaces of A and A_{pls} , respectively. Furthermore, there exists an orthogonal matrix R such that $P_{pls} = P_k R$, where P_k consists of the first k columns of P .

The main result of this section is summarized in the following theorem:

Theorem 6.1: Let W_{pls} be the optimal solution to the optimization problem in Eq. (20) and let W_{CCA} be the optimal CCA transformation defined in Eq. (18). Then, $W_{pls} = W_{CCA} R$ for an orthogonal matrix R .

It follows from Theorem 6.1 that OPLS can be readily reformulated as an equivalent least-squares problem using the same class indicator matrix defined in Eq. (16).

7 ANALYSIS OF REGULARIZATION ON CCA

In this section, we investigate the effect of regularization on CCA. The least-squares CCA formulation established in this paper assumes that no regularization is applied. However, regularization is commonly used to control the complexity of a learning model and it has been applied in various machine learning algorithms. The use of regularization in CCA has natural statistical interpretations [15], [26]. In practice, regularization is commonly enforced for both sets of multidimensional variables in

CCA, as it is generally believed that the CCA solution is dependent on the regularization on both variables. Following the derivations from previous sections, we show that the CCA projection for one set of variables is independent of the regularization on the other set of multidimensional variables, providing new insights on the effect of regularization on CCA.

7.1 Regularization on Y

The use of regularization on Y in CCA leads to the following generalized eigenvalue problem:

$$XY^T(YY^T + \lambda_y I)^{-1} YX^T w = \eta(XX^T)w, \quad (23)$$

where $\lambda_y > 0$ is the regularization parameter. The generalized eigenvalue problem in Eq. (23) can be expressed as:

$$XH_r H_r^T X^T w = \eta(XX^T)w, \quad (24)$$

where the matrix $H_r \in \mathbb{R}^{n \times k}$ for regularized CCA is defined as:

$$H_r = Y^T(YY^T + \lambda_y I)^{-1/2}. \quad (25)$$

The main result is summarized in the following theorem:

Theorem 7.1: Let W_{rCCA} be the matrix consisting of the principal eigenvectors of the generalized eigenvalue problem in Eq. (24) corresponding to the nonzero eigenvalues. Then, $W_{rCCA} = W_{CCA} R$ for an orthogonal matrix R .

It is easy to check that the range spaces of H in Eq. (8), and H_r in Eq. (25) coincide. The proof follows the same arguments in Lemma 6.1 and Theorem 6.1.

Theorem 7.1 shows that the CCA formulation can be formulated equivalently as a least-squares problem when the regularization on Y is considered. Note that Y can be an arbitrary matrix (not necessarily the class label matrix). An important consequence from the equivalence relationship is that the projection of CCA for one view is independent of the regularization on the other view. A similar result can be obtained for kernel CCA.

7.2 Regularization on X

Since the regularization on Y does not affect the projection of X , we next consider the regularization on X separately. The resulting generalized eigenvalue problem in CCA can be formulated as follows:

$$(XHH^T X^T)w = \eta(XX^T + \lambda_x I)w, \quad (26)$$

where $\lambda_x > 0$ is the regularization parameter for X . Similarly, we can derive the eigendecomposition of the matrix $(XX^T + \lambda_x I)^{-1}(XHH^T X^T)$, and the result is summarized in the following lemma:

Lemma 7.1: Define the matrix $B \in \mathbb{R}^{r \times k}$ as

$$B = (\Sigma_1^2 + \lambda_x I)^{-1/2} \Sigma_1 V_1^T H \quad (27)$$

and denote its SVD as $B = P_B \Sigma_B Q_B^T$, where $P_B \in \mathbb{R}^{r \times r}$ and $Q_B \in \mathbb{R}^{k \times k}$ are orthogonal, and $\Sigma_B \in \mathbb{R}^{r \times k}$ is

diagonal. Then, the eigenvectors corresponding to the top ℓ eigenvalues of matrix $(XX^T + \lambda_x I)^{-1}(XHH^T X^T)$ are given by

$$W = U_1(\Sigma_1^2 + \lambda_x I)^{-1/2} P_{B\ell}, \quad (28)$$

where $P_{B\ell}$ consists of the first ℓ ($\ell \leq \text{rank}(B)$) columns of P_B .

It can be observed that the range space of B is different from that of A ; thus the equivalence relationship between CCA and least squares does not hold when the regularization on X is considered. However, the equivalence relationship between CCA and OPLS still holds when the regularization on X is applied. The main results are summarized in Theorem 7.2 below (proofs follows similar arguments in Lemma 6.1):

Theorem 7.2: Let $B_{pls} = (\Sigma_1^2 + \lambda_x I)^{-1/2} \Sigma_1 V_1^T H_{pls}$. Let the thin SVD of B and B_{pls} be

$$B = P^B \Sigma^B (Q^B)^T, B_{pls} = P_{pls}^B \Sigma_{pls}^B (Q_{pls}^B)^T,$$

where $P^B, P_{pls}^B \in \mathbb{R}^{r \times r_B}$, and $r_B = \text{rank}(B) = \text{rank}(B_{pls})$. Then the range spaces of B and B_{pls} coincide. Furthermore, there exists an orthogonal matrix $R^B \in \mathbb{R}^{r_B \times r_B}$ such that $P^B = P_{pls}^B R^B$. Therefore, CCA and OPLS are equivalent for any $\lambda_x > 0$.

Recall that the CCA formulation reduces to a generalized eigenvalue problem as in Eq. (5), which requires the computation of the inverse of the matrix $YY^T \in \mathbb{R}^{k \times k}$. Computing the inverse may be computationally expensive, when the dimensionality k of the data in Y is large. This is the case in content-based image retrieval [27], where the two views correspond to the text and image data that are both of high-dimensionality. An important consequence of the established equivalence relationship between CCA and OPLS is that the inverse of a large matrix can be effectively avoided for the computation of the projection for one view.

8 EXPERIMENTS

In this section, we use a collection of multilabel data sets to verify the results established in this paper. We also evaluate the effectiveness and efficiency of LS-CCA and its extensions. The effect of regularization and the relationship between CCA and OPLS are also investigated. All algorithms were implemented in Matlab, and the source codes are available at <http://www.public.asu.edu/~jye02/Software/CCA/>.

8.1 Experimental Setup

We use three types of data in the experiments. The gene expression pattern image data¹ describe the gene expression patterns of *Drosophila* during development [28]. Each image is annotated with a variable number of textual terms (labels) from a controlled vocabulary. We apply Gabor filters to extract a 384-dimensional feature

1. All images were extracted from the BDGP database at <http://www.fruitfly.org>.

TABLE 1
Summary of statistics of the data sets

Data Set	n_{tot}	d	k
Gene Image 1	863	384	10
Gene Image 2	1041	384	15
Gene Image 3	1138	384	20
Gene Image 4	1222	384	25
Gene Image 5	1349	384	30
Scene	2407	294	6
Yahoo\Arts&Humanities	3712	23146	26

n_{tot} is number of data points, d is the dimensionality, and k is the number of labels.

vector from each image. We use five data sets with different numbers of terms (class labels). We also evaluate the proposed methods on the the scene data set [29], which is commonly used as a benchmark data set for multilabel learning. To study the scalability of the proposed least-squares formulation, a text document data set with high dimensionality from Yahoo! [30] is used. The statistics of these data sets are summarized in Table 1.

For all data sets, ten random partitions of data into training and test sets are generated and the averaged performance is reported. For the high-dimensional text document data set, we follow the feature selection methods studied in [31] for text documents and extract different numbers of terms (features) to investigate the performance of algorithms. Five algorithms are compared, including CCA and its regularized version (denoted as rCCA) as in Eq. (5), the proposed least-squares CCA formulation (denoted as LS-CCA) and its 2-norm and 1-norm regularized versions (denoted as LS-CCA₂ and LS-CCA₁, respectively). All methods are used to project the data onto a lower-dimensional space in which a linear SVM is applied for classification for each label. The Receiver Operating Characteristic (ROC) score is computed for each label and the averaged performance over all labels and all splittings is reported.

8.2 Evaluation of Equivalence Relationship and Performance Comparison

We first evaluate the equivalence relationship between CCA and least squares. We observe that when the data dimensionality d is much larger than the sample size n , the condition in Theorem 3.2 tends to hold. It follows from Theorem 3.2 that $\text{rank}(C_{XX})$ equals $\text{rank}(C_{HH}) + \text{rank}(C_{DD})$ and all diagonal elements of Σ_A are ones, which is consistent with the observations in the experiments.

In Table 2, we report the mean ROC scores over all labels and all splittings for each data set. The main observations include: 1) CCA and LS-CCA achieve the same performance for all data sets, which is consistent with our theoretical results; 2) The regularized CCA extensions including rCCA, LS-CCA₂, and LS-CCA₁ perform much better than their counterparts CCA and LS-CCA without the regularization; 3) LS-CCA₂ is comparable to rCCA in all data sets, while LS-CCA₁ achieves the best

performance for all gene image data sets. These observations demonstrate the effectiveness of the proposed least-squares extensions using the regularization technique.

TABLE 2
Comparison of different CCA formulations in terms of mean ROC scores

Data set	n	CCA	LS-CCA	rCCA	LS-CCA ₂	LS-CCA ₁
Gene 1	368	0.542	0.542	0.617	0.619	0.722
Gene 2	362	0.534	0.534	0.602	0.603	0.707
Gene 3	372	0.538	0.538	0.609	0.610	0.714
Gene 4	369	0.540	0.540	0.603	0.605	0.704
Gene 5	354	0.548	0.548	0.606	0.608	0.709
Scene	198	0.710	0.710	0.864	0.900	0.900
Yahoo	2000	0.521	0.521	0.799	0.801	0.784

n is the size of training set. For all data sets, we run the algorithms on ten different partitions of the data into training and test sets. For the regularized algorithms, the parameter value is chosen via cross-validation.

8.3 Sensitivity Study

In this experiment, we investigate the performance of LS-CCA and its variants in comparison with CCA when the condition in Theorem 3.2 does not hold, which is the case in many real-world applications. Specifically, we use a gene data set *Gene Image 2* with the dimensionality fixed at $d = 384$ and $k = 15$ labels, while the size of the training set varies from 100 to 900 with a step size about 100.

The performance of different linear algorithms as the size of training set increases is presented in Figure 1(A). We can observe that, in general, the performance of all algorithms increases as the training size increases. When n is small, the condition in Theorem 3.2 holds, thus CCA and LS-CCA are equivalent, and they achieve the same performance. When n further increases, CCA and LS-CCA achieve different ROC scores, although the difference is always very small in our experiment. Similar to the last experiment, we can observe from the figure that the regularized methods perform much better than CCA and LS-CCA, and LS-CCA₂ is comparable to rCCA. The sparse formulation LS-CCA₁ performs the best for this data set.

The sensitivity experiment is also performed on the scene data set. The results are summarized in Figure 1(B), from which similar observations can be made.

8.4 Scalability Study

In this experiment we study the scalability of the least-squares formulation in comparison with the original CCA formulation. Since regularized algorithms are preferred in practice, we compare the regularized CCA formulation (rCCA) and the 2-norm regularized least-squares formulation (LS-CCA₂). The least-squares problem is solved by the LSQR algorithm [14].

Figure 2 (left) shows the computation time of the two formulations on the high-dimensional text document

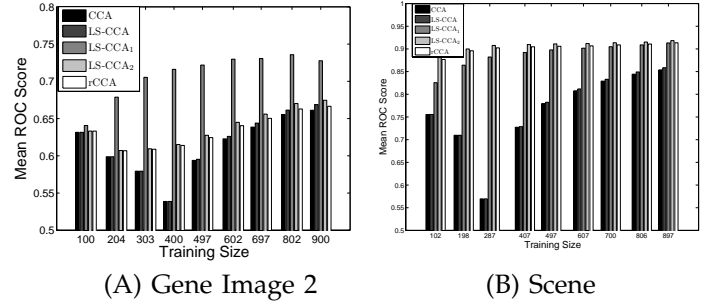


Fig. 1. Comparison of all linear methods on the Gene Image 2 (left) and the scene (right) data sets in terms of mean ROC scores as the size of training set increases.

data set Yahoo\Arts&Humanities as the data dimensionality increases with the size of training set fixed at 1,000. It can be observed that the computation time of both algorithms increases steadily as the data dimensionality increases. However, the computation time of the least-squares formulation (LS-CCA₂) is substantially less than that of the original formulation (rCCA). In fact, the computation time of LS-CCA₂ is less than 5 second for all tested data dimensionality. We also evaluate the scalability of two formulations in terms of the training sample size. Figure 2 (right) plots the computation time of the two formulations on the text document data set when the training sample size increases with the data dimensionality fixed at 2,000, and a similar observation can be made. The size of the training set is not further increased due to the high computational cost of the original eigenvalue problem. From Figure 2 we conclude that the least-squares formulation is more scalable than the original CCA formulation.

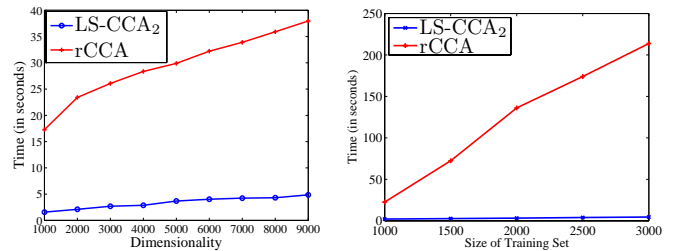


Fig. 2. Computation time (in seconds) of the regularized CCA (rCCA) and the equivalent least-squares formulation with 2-norm regularization (LS-CCA₂) on the Yahoo\Arts&Humanities data set as the dimensionality (left) or the size of training set (right) increases. The x -axis is the dimensionality (left) or the size of training set (right) and the y -axis is the computation time (in seconds).

8.5 Regularization Analysis

In this experiment, we study the effect of regularization for CCA. In addition, we compare the performance of CCA and OPLS under different regularization parameter values. Specifically, we randomly choose 700 samples from the scene data set for training, and vary the regularization parameter values from $1e-6$ to $1e4$.

First, we consider the regularization on X only. The performance of CCA and OPLS on the scene data set as λ_x varies is summarized in Figure 3 (left). We can

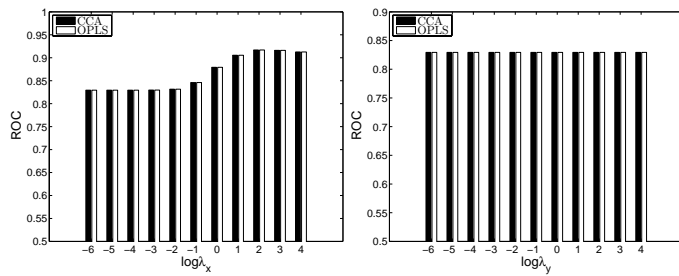


Fig. 3. Comparison of CCA and OPLS in terms of the mean ROC scores on the scene data set as the regularization parameter λ_x and λ_y vary. In the left figure, λ_x increases from $1e-6$ to $1e4$; in the right figure, λ_y varies from $1e-6$ to $1e4$.

observe from the figure that under all values of λ_x , the performance of CCA and OPLS is identical. This confirms the equivalence relationship between CCA and OPLS established in Theorem 7.2. We also observe that the performance of CCA and OPLS can be improved significantly by using an appropriate regularization parameter, which justifies the use of regularization on X .

Next, we consider the regularization on Y only. The performance of CCA and OPLS with different values of λ_y is summarized in Figure 3 (right). We can observe that the performance of CCA remains the same as λ_y varies, verifying that the regularization on Y does not affect its performance. In addition, we observe that the performance of both methods is identical in all cases, which is consistent with our theoretical analysis.

9 CONCLUSIONS

In this paper, we establish an equivalent least-squares formulation for CCA under a mild condition, which tends to hold for high-dimensional data. Based on the equivalence relationship established in this paper, we propose several CCA extensions including sparse CCA. An efficient algorithm is presented to scale the CCA formulation to very large data sets. We further extend the equivalence relationship to orthonormalized partial least squares. In addition, we show that the CCA projection for one view is independent of the regularization on the other view. We have conducted experiments on a collection of multilabel data sets. Our experiments show that the performance of the proposed least-squares CCA formulation and the original CCA formulation is very close even when the condition is violated.

ACKNOWLEDGMENTS

This research is sponsored in part by the Arizona State University, by the National Science Foundation under Grant No. IIS-0612069, and by the National Geo-spatial Intelligence Agency under Grant No. HM1582-08-1-0016.

REFERENCES

- [1] H. Hotelling, "Relations between two sets of variables," *Biometrika*, vol. 28, pp. 312–377, 1936.
- [2] D. Hardoon, S. Szedmak, and J. Shawe-taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Computation*, vol. 16, no. 12, 2004.

- [3] J.-P. Vert and M. Kanehisa, "Graph-driven feature extraction from microarray data using diffusion kernels and kernel CCA," in *NIPS 15*, 2003, pp. 1425–1432.
- [4] S. Yu, K. Yu, V. Tresp, and H.-P. Kriegel, "Multi-output regularized feature projection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 12, pp. 1600–1613, 2006.
- [5] C. Bishop, *Pattern Recognition and Machine Learning*. New York, NY: Springer, 2006.
- [6] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY: Springer, 2001.
- [7] G. Golub and C. V. Loan, *Matrix Computations*. Baltimore, MD: Johns Hopkins Press, 1996.
- [8] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B*, vol. 58, no. 1, pp. 267–288, 1996.
- [9] A. d'Aspremont, L. Ghaoui, M. Jordan, and G. Lanckriet, "A direct formulation for sparse PCA using semidefinite programming," in *NIPS 16*, 2004, pp. 41–48.
- [10] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani, "1-norm support vector machines," in *NIPS 15*, 2003, pp. 49–56.
- [11] D. Watkins, *Fundamentals of matrix computations*. New York, NY: John Wiley & Sons, Inc., 1991.
- [12] B. Sriperumbudur, D. Torres, and G. Lanckriet, "Sparse eigen methods by D.C. programming," in *ICML*, 2007, pp. 831–838.
- [13] T. Hastie, A. Buja, and R. Tibshirani, "Penalized discriminant analysis," *Annals of Statistics*, vol. 23, pp. 73–102, 1995.
- [14] C. Paige and M. Saunders, "LSQR: An algorithm for sparse linear equations and sparse least squares," *ACM Transactions on Mathematical Software*, vol. 8, no. 1, pp. 43–71, 1982.
- [15] F. Bach and M. Jordan, "Kernel independent component analysis," *Journal of Machine Learning Research*, vol. 3, pp. 1–48, 2003.
- [16] J. Ye, "Least squares linear discriminant analysis," in *ICML*, 2007, pp. 1087–1094.
- [17] B. Schölkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. Cambridge, MA: MIT Press, 2002.
- [18] J. Liu, S. Ji, and J. Ye, *SLEP: Sparse Learning with Efficient Projections*, Arizona State University, 2009. [Online]. Available: <http://www.public.asu.edu/~jye02/Software/SLEP>
- [19] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani, "Pathwise coordinate optimization," *Annals of Applied Statistics*, no. 2, pp. 302–332, 2007.
- [20] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Annals of Statistics*, vol. 32, p. 407.
- [21] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*. New York, NY: Halsted Press, 1992.
- [22] M. Barker and W. Rayens, "Partial least squares for discrimination," *Journal of Chemometrics*, vol. 17, no. 3, pp. 166–173, 2003.
- [23] D. Hardoon, "Semantic models for machine learning," Ph.D. dissertation, University of Southampton, 2006.
- [24] R. Rosipal and N. Krämer, "Overview and recent advances in partial least squares," in *Subspace, Latent Structure and Feature Selection Techniques, Lecture Notes in Computer Science*. Springer, 2006, pp. 34–51.
- [25] K. Worsley, J.-B. Poline, K. Friston, and A. Evans, "Characterizing the response of PET and fMRI data using multivariate linear models," *Neuroimage*, vol. 6, no. 4, pp. 305–319, 1997.
- [26] F. Bach and M. Jordan, "A probabilistic interpretation of canonical correlation analysis," University of California, Berkeley, Tech. Rep., 2005.
- [27] D. Hardoon and J. Shawe-Taylor, "KCCA for different level precision in content-based image retrieval," in *Third International Workshop on Content-Based Multimedia Indexing*, 2003.
- [28] P. Tomancak and *et al.*, "Systematic determination of patterns of gene expression during *Drosophila* embryogenesis," *Genome Biology*, vol. 3, no. 12, 2002.
- [29] M. Boutell, J. Luo, X. Shen, and C. Brown, "Learning multilabel scene classification," *Pattern Recognition*, vol. 37, no. 9, pp. 1757–1771, 2004.
- [30] H. Kazawa, T. Izumitani, H. Taira, and E. Maeda, "Maximal margin labeling for multi-topic text categorization," in *NIPS 17*, 2005, pp. 649–656.
- [31] Y. Yang and J. Pedersen, "A comparative study on feature selection in text categorization," in *ICML*, 1997, pp. 412–420.
- [32] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. New York, NY: Cambridge University Press, 2004.

SUPPLEMENTARY FILE

Proofs

Proof of Theorem 3.1

Proof: The matrix $C_{XX}^\dagger C_{HH}$ can be diagonalized as follows:

$$\begin{aligned} C_{XX}^\dagger C_{HH} &= U_1 \Sigma_r^{-2} U_1^T X H H^T X^T \\ &= U_1 \Sigma_r^{-1} A H^T X^T U U^T \\ &= U \begin{bmatrix} I_r \\ 0 \end{bmatrix} \Sigma_r^{-1} A [H^T X^T U_1, H^T X^T U_2] U^T \\ &= U \begin{bmatrix} \Sigma_r^{-1} A A^T \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} U^T \\ &= U \begin{bmatrix} \Sigma_r^{-1} P & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \Sigma_A \Sigma_A^T & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} P^T \Sigma_r & 0 \\ 0 & I \end{bmatrix} U^T \end{aligned}$$

where the second equality follows since U is orthogonal, the fourth equality follows since $X^T U_2 = 0$ as shown in Eq. (12), and the last equality follows from Eq. (14). \square

Proof of Lemma 3.2

Proof: Define the matrix $J \in \mathbb{R}^{d \times d}$ as follows:

$$J = U \begin{bmatrix} \Sigma_r^{-1} P & 0 \\ 0 & I_{d-r} \end{bmatrix}. \quad (29)$$

It follows from the definitions of C_{XX} , C_{HH} , and C_{DD} in Eqs. (9)-(11) that

$$\begin{aligned} J^T C_{XX} J &= \text{diag}(I_r, 0), \\ J^T C_{HH} J &= \text{diag}(\Sigma_A \Sigma_A^T, 0) \\ &= \text{diag}(a_1, \dots, a_r, 0, \dots, 0), \\ J^T C_{DD} J &= J^T C_{XX} J - J^T C_{HH} J \\ &= \text{diag}(b_1, \dots, b_r, 0, \dots, 0), \end{aligned} \quad (30)$$

where $b_i = 1 - a_i$, for $i = 1, \dots, r$. Note that since J is nonsingular, we have

$$\text{rank}(C_{XX}) = \text{rank}(J^T C_{XX} J) = r.$$

It follows from our assumption that

$$\text{rank}(J^T C_{HH} J) + \text{rank}(J^T C_{DD} J) = r + s. \quad (31)$$

Since both $J^T C_{HH} J$ and $J^T C_{DD} J$ are diagonal, there are a total of $r + s$ nonzero elements in $J^T C_{HH} J$ and $J^T C_{DD} J$. Note that $f = \text{rank}(\Sigma_A) = \text{rank}(\hat{\Sigma}_A)$. Thus $a_1 \geq \dots \geq a_f > 0 = a_{f+1} = \dots = a_r$. It follows from Eq. (30) that

$$a_i + b_i = 1, b_r \geq \dots \geq b_1 \geq 0. \quad (32)$$

This implies that at least one of a_i or b_i is positive for $1 \leq i \leq r$. To satisfy the rank equality in Eq. (31), we therefore must have

$$\begin{aligned} 1 &= a_1 = a_2 = \dots = a_{f-s} > a_{f-s+1} \geq \dots \geq a_f \\ &> a_{f+1} = \dots = a_r = 0, \\ 0 &= b_1 = b_2 = \dots = b_{f-s} < b_{f-s+1} \leq \dots \leq b_f \\ &< b_{f+1} = \dots = b_r = 1. \end{aligned}$$

This completes the proof of the lemma. \square

Proof of Theorem 3.2

Proof: Denote $H = [h_1, \dots, h_k]$, and $D = [h_{k+1}, \dots, h_n]$. Note that X is column centered, i.e., $\sum_{i=1}^n x_i = 0$. It follows from Lemma 3.1 that $H^T e = 0$, that is,

$$h_i^T e = 0, \text{ for } 1 \leq i \leq k. \quad (33)$$

Since $[H, D]$ is an orthogonal matrix, $\{h_1, \dots, h_n\}$ form a basis for \mathbb{R}^n . Thus we can represent $e \in \mathbb{R}^n$ as

$$e = \sum_{i=1}^n \mu_i h_i, \text{ where } \mu_i \in \mathbb{R}. \quad (34)$$

It follows from the orthogonality of $[H, D]$ and Eq. (33) that e can be expressed as $e = \sum_{i=k+1}^n \mu_i h_i$, and

$$0 = X e = X \left(\sum_{i=k+1}^n \mu_i h_i \right) = \sum_{i=k+1}^n \mu_i (X h_i). \quad (35)$$

Since not all μ_i 's are zero, the $n - k$ columns of $X D$ are linearly dependent, thus $\text{rank}(X D) \leq n - k - 1$. According to the property of matrix rank, we have

$$\begin{aligned} \text{rank}(X D) &\geq \text{rank}(X) + \text{rank}(D) - n \\ &= (n - 1) + (n - k) - n = n - k - 1. \end{aligned} \quad (36)$$

Thus, $\text{rank}(X D) = n - k - 1$ holds.

For matrix $X H$, we have

$$\begin{aligned} \text{rank}(X) &= \text{rank}(X [H, D]) \leq \text{rank}(X H) + \text{rank}(X D) \\ &\Leftrightarrow n - 1 \leq \text{rank}(X H) + n - k - 1 \\ &\Leftrightarrow \text{rank}(X H) \geq k. \end{aligned}$$

On the other hand, since $X H \in \mathbb{R}^{d \times k}$, $\text{rank}(X H) \leq k$. Thus we have $\text{rank}(X H) = k$ and

$$\begin{aligned} \text{rank}(C_{XX}) &= \text{rank}(X) = n - 1, \\ \text{rank}(C_{HH}) &= \text{rank}(X H) = k, \\ \text{rank}(C_{DD}) &= \text{rank}(X D) = n - k - 1. \end{aligned}$$

It follows that $s = 0$. On the other hand,

$$f = \text{rank}(A) = \text{rank}(\Sigma_r^{-1} U_1^T X H) = \text{rank}(X H) = k.$$

Hence,

$$1 = a_1 = a_2 = \dots = a_k > 0 = a_{k+1} = \dots = a_r,$$

and all diagonal elements of Σ_A are one. This completes the proof of the theorem. \square

Proof of Lemma 6.1

Proof: Let the SVD of Y be

$$Y = U_y \Sigma_y V_y^T, \quad (37)$$

where $U_y \in \mathbb{R}^{k \times k}$, $V_y \in \mathbb{R}^{n \times k}$, and $\Sigma_y \in \mathbb{R}^{k \times k}$ is diagonal. Since Y is assumed to have full column rank, all the diagonal elements of Σ_y are positive. Thus,

$$\begin{aligned} A &= V_1^T H = V_1^T V_y U_y^T \\ A_{pls} &= V_1^T H_{pls} = V_1^T V_y \Sigma_y U_y^T. \end{aligned}$$

It follows that $A = A_{pls}U_y\Sigma_y^{-1}U_y^T$ and $A_{pls} = AU_y\Sigma_yU_y^T$. Therefore, $\mathcal{R}(A) = \mathcal{R}(A_{pls})$.

It is clear from the definition that $P_kP_k^T$ and $P_{pls}P_{pls}^T$ are the orthogonal projections onto the range spaces of A and A_{pls} , respectively. Since $\mathcal{R}(A) = \mathcal{R}(A_{pls})$, both $P_kP_k^T$ and $P_{pls}P_{pls}^T$ are orthogonal projections onto the same subspace. Note that the orthogonal projection onto a subspace is unique [7], then we have

$$P_kP_k^T = P_{pls}P_{pls}^T. \quad (38)$$

Therefore,

$$P_{pls} = P_{pls}P_{pls}^TP_{pls} = P_kP_k^TP_{pls} = P_kR,$$

where $R = P_k^TP_{pls}$ is a square matrix. It is easy to verify that $RR^T = R^TR = I$. This completes the proof of the lemma. \square

Proof of Theorem 6.1

Proof: Similar to the derivation of the eigendecomposition of CCA, we can obtain the eigenvectors corresponding to nonzero eigenvalues of the generalized eigenvalue problem in Eq. (21):

$$W_{pls} = U_1\Sigma_r^{-1}P_{pls} = U_1\Sigma_r^{-1}P_kR = W_{CCAR}, \quad (39)$$

where the second equality follows from Lemma 6.1, and the third equality follows from Eq. (18). This completes the proof of the theorem. \square

Proof of Lemma 7.1

Proof: We can decompose $(XX^T + \lambda_x I)^{-1}(XHH^TX^T)$ as follows:

$$\begin{aligned} & (XX^T + \lambda_x I)^{-1}(XHH^TX^T) \\ &= U_1(\Sigma_1^2 + \lambda_x I)^{-1}\Sigma_1V_1^THH^TV_1\Sigma_1U_1^T \\ &= U_1(\Sigma_1^2 + \lambda_x I)^{-1/2}(\Sigma_1^2 + \lambda_x I)^{-1/2}\Sigma_1V_1^THH^TV_1\Sigma_1(\Sigma_1^2 + \lambda_x I)^{-1/2}(\Sigma_1^2 + \lambda_x I)^{1/2}U_1^T \\ &= U_1(\Sigma_1^2 + \lambda_x I)^{-1/2}BB^T(\Sigma_1^2 + \lambda_x I)^{1/2}U_1^T \\ &= U \begin{bmatrix} I_r \\ 0 \end{bmatrix} (\Sigma_1^2 + \lambda_x I)^{-1/2}BB^T(\Sigma_1^2 + \lambda_x I)^{1/2} \begin{bmatrix} I_r & 0 \end{bmatrix} U^T \\ &= U \begin{bmatrix} (\Sigma_1^2 + \lambda_x I)^{-1/2}BB^T(\Sigma_1^2 + \lambda_x I)^{1/2} & 0 \\ 0 & 0 \end{bmatrix} U^T \\ &= U \begin{bmatrix} (\Sigma_1^2 + \lambda_x I)^{-1/2}P_B & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \Sigma_B\Sigma_B^T & 0 \\ 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} P_B^T(\Sigma_1^2 + \lambda_x I)^{1/2} & 0 \\ 0 & I \end{bmatrix} U^T. \end{aligned}$$

Thus, the eigenvectors corresponding to the top ℓ eigenvalues of $(XX^T + \lambda_x I)^{-1}(XHH^TX^T)$ are given by $U_1(\Sigma_1^2 + \lambda_x I)^{-1/2}P_{B\ell}$. \square

Kernel CCA and Kernel Least Squares

Kernel Methods

Both CCA and least squares can be extended to the kernel-induced feature space using the kernel trick. Kernel methods [17], [32] work by first mapping the data into a higher-dimensional Hilbert space (feature space) \mathcal{F} through a nonlinear mapping Φ as $\Phi : \mathbb{R}^d \rightarrow \mathcal{F}$. The nonlinear mapping can be implicitly defined by a symmetric kernel function κ , which computes the inner product of the images of each data pair in \mathcal{F} as

$$\kappa(x_i, x_j) = \Phi(x_i)^T\Phi(x_j).$$

A kernel function κ satisfies the finitely positive semidefinite property: for any $x_1, \dots, x_n \in \mathbb{R}^d$, the so-called kernel Gram matrix K , defined as $K_{ij} = \kappa(x_i, x_j)$, is symmetric and positive semidefinite [17].

Note that in multi-label learning the view Y is derived from the label information, thus only the feature mapping for the view X is considered. It follows from the representer theorem [17] that the projection vector w_x can be written as:

$$w_x = \Phi(X)\alpha, \quad (40)$$

for some vector $\alpha \in \mathbb{R}^n$. Thus, the resulting correlation coefficient ρ in the feature space can be expressed as

$$\begin{aligned} \rho &= \frac{w_x^T\Phi(X)Y^T w_y}{\sqrt{(w_x^T\Phi(X)\Phi(X)^T w_x)(w_y^T Y Y^T w_y)}} \\ &= \frac{\alpha^T K_x Y^T w_y}{\sqrt{(\alpha^T K_x^2 \alpha)(w_y^T Y Y^T w_y)}}, \end{aligned}$$

where $K_x = \Phi(X)^T\Phi(X)$ is the kernel matrix. Similar to the linear case, it can be formulated equivalently as the following optimization problem:

$$\begin{aligned} \max_{\alpha, w_y} & \alpha^T K_x Y^T w_y \\ \text{subject to} & \alpha^T K_x^2 \alpha = 1, \\ & w_y^T Y Y^T w_y = 1. \end{aligned} \quad (41)$$

Assume that $Y Y^T$ is nonsingular. It can be shown that α can be obtained by solving the following optimization problem:

$$\begin{aligned} \max_{\alpha} & \alpha^T K_x Y^T (Y Y^T)^{-1} Y K_x \alpha \\ \text{subject to} & \alpha^T K_x^2 \alpha = 1. \end{aligned} \quad (42)$$

It can be verified that the optimal α is the eigenvector corresponding to the largest eigenvalue of the following generalized eigenvalue problem:

$$K_x Y^T (Y Y^T)^{-1} Y K_x \alpha = \eta K_x^2 \alpha. \quad (43)$$

Multiple projection vectors can be obtained simultaneously by computing the top ℓ eigenvectors of the generalized eigenvalue problem in Eq. (43).

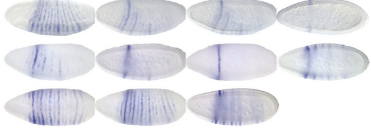
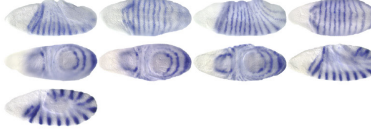
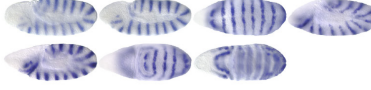

Stage range		BDGP terms
4-6		dorsal ectoderm anlage in statu nascendi mesectoderm anlage in statu nascendi segmentally repeated trunk mesoderm anlage in statu nascendi ventral ectoderm anlage in statu nascendi
7-8		dorsal ectoderm primordium hindgut anlage mesectoderm primordium procephalic ectoderm anlage trunk mesoderm primordium P2 ventral ectoderm primordium P2
9-10		inclusive hindgut primordium mesectoderm primordium procephalic ectoderm primordium trunk mesoderm primordium ventral ectoderm primordium
11-12		atrium primordium brain primordium clypeo-labral primordium dorsal epidermis primordium gnathal primordium head epidermis primordium P1 hindgut proper primordium midline primordium ventral epidermis primordium ventral nerve cord primordium

Fig. 4. Sample image groups and their associated terms (class labels) in the BDGP database (<http://www.fruitfly.org>) for the segmentation gene *engrailed* in 4 stage ranges.

The regularization technique can also be applied to kernel CCA, resulting in the regularized kernel CCA (kr-CCA), which solves the following optimization problem:

$$\begin{aligned} \max_G \quad & \text{trace} \left(G^T K_x Y^T (Y Y^T)^{-1} Y K_x G \right) \\ \text{subject to} \quad & G^T (K_x^2 + \lambda K_x) G = I_\ell. \end{aligned} \quad (44)$$

Similarly, the least squares problem can also be formulated in the kernel-induced feature space. Following the representer theorem [17], the weight matrix W in the feature space can be expressed as

$$W = \Phi(X)G, \quad (45)$$

for some $G \in \mathbb{R}^{n \times \ell}$. Substituting Eq. (45) into the cost function, we have

$$f = \|W^T \Phi(X) - T\|_F^2 = \|G K_x - T\|_F^2.$$

It follows that the optimal G is given by

$$G = K_x^\dagger T^T. \quad (46)$$

Relationship between Kernel CCA and Kernel Least Squares

In this section, we extend the equivalence relationship between CCA and least squares to the kernel-induced feature space. Assume that $\Phi(X)$ is centered in the feature space, that is $\Phi(X)e = 0$. Since the kernel matrix $K_x = \Phi(X)^T \Phi(X)$ is symmetric, we have

$$K_x e = 0, \text{ and } e^T K_x = 0. \quad (47)$$

We use superscript Φ to denote quantities in the feature space transformed by the mapping Φ . Similar to the linear case, we define the following matrices:

$$C_{XX}^\Phi = K_x^2 \in \mathbb{R}^{n \times n}, \quad (48)$$

$$C_{HH}^\Phi = K_x H H^T K_x \in \mathbb{R}^{n \times n}, \quad (49)$$

$$C_{DD}^\Phi = C_{XX}^\Phi - C_{HH}^\Phi = K_x D D^T K_x \in \mathbb{R}^{n \times n}. \quad (50)$$

Assume $r_K = \text{rank}(K_x)$, and let the SVD of K_x be

$$\begin{aligned} K_x &= U_K \Sigma_K U_K^T \\ &= [U_{K1}, U_{K2}] \text{diag}(\Sigma_{K1}, 0) [U_{K1}, U_{K2}]^T \\ &= U_{K1} \Sigma_{K1} U_{K1}^T, \end{aligned} \quad (51)$$

where U_K is orthogonal, $\Sigma_K \in \mathbb{R}^{n \times n}$, $U_{K1} \in \mathbb{R}^{n \times r_K}$, $U_{K2} \in \mathbb{R}^{n \times (n-r_K)}$ and $\Sigma_{K1} \in \mathbb{R}^{r_K \times r_K}$. Note that U_{K2} lies in the null space of K_x , that is $K_x U_{K2} = 0$.

The eigenvalue problem in kernel CCA can be reformulated as

$$(C_{XX}^\Phi)^\dagger C_{HH}^\Phi \alpha = \eta \alpha.$$

Similarly, we define

$$A^\Phi = \Sigma_{K1}^{-1} U_{K1}^T K_x H = U_{K1}^T H$$

and let the SVD of A^Φ be $A^\Phi = P^\Phi \Sigma_A^\Phi (Q^\Phi)^T$. Then the eigendecomposition of the matrix $(C_{XX}^\Phi)^\dagger C_{HH}^\Phi$ can be

TABLE 3

Comparison of different CCA formulations in terms of mean ROC scores. n is the size of training set. Ten different partitions of the data into training and test sets are applied for each data set. For the regularized algorithms, the parameter value is chosen via cross-validation.

Data set	n	CCA	LS-CCA	rCCA	LS-CCA ₂	LS-CCA ₁	kCCA	kLS-CCA	krCCA	kLS-CCA ₂
Gene 1	368	0.542	0.542	0.617	0.619	0.722	0.677	0.677	0.727	0.733
Gene 2	362	0.534	0.534	0.602	0.603	0.707	0.659	0.659	0.712	0.721
Gene 3	372	0.538	0.538	0.609	0.610	0.714	0.666	0.666	0.719	0.730
Gene 4	369	0.540	0.540	0.603	0.605	0.704	0.660	0.660	0.713	0.724
Gene 5	354	0.548	0.548	0.606	0.608	0.709	0.667	0.667	0.716	0.729
Scene	198	0.710	0.710	0.864	0.900	0.900	0.878	0.878	0.919	0.921
Yahoo	2000	0.521	0.521	0.799	0.801	0.784	0.705	0.705	0.809	0.810

derived as follows:

$$\begin{aligned}
(C_{XX}^\Phi)^\dagger C_{HH}^\Phi &= U_{K1} \Sigma_{K1}^{-2} U_{K1}^T K_x H H^T K_x^T \\
&= U_{K1} \Sigma_{K1}^{-1} A^\Phi H^T K_x U_K U_K^T \\
&= U_K \begin{bmatrix} I_{r_K} \\ 0 \end{bmatrix} \Sigma_{K1}^{-1} A^\Phi [H^T K_x U_{K1}, H^T K_x U_{K2}] U_K^T \\
&= U_K \begin{bmatrix} \Sigma_{K1}^{-1} A^\Phi (A^\Phi)^T \Sigma_{K1} & 0 \\ 0 & 0 \end{bmatrix} U_K^T \\
&= U_K \begin{bmatrix} \Sigma_{K1}^{-1} P^\Phi & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \Sigma_A^\Phi (\Sigma_A^\Phi)^T & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} (P^\Phi)^T \Sigma_{K1} & 0 \\ 0 & I \end{bmatrix} U_K^T.
\end{aligned}$$

Thus the optimal solution, which consists of top ℓ eigenvectors of matrix $(C_{XX}^\Phi)^\dagger C_{HH}^\Phi$, is given by

$$G_{CCA}^\Phi = U_{K1} \Sigma_{K1}^{-1} P_\ell^\Phi, \quad (52)$$

where P_ℓ^Φ contains the first ℓ columns of P^Φ .

Using the class indicator matrix defined in Eq. (16), the solution to kernel least squares is given by

$$\begin{aligned}
G_{LS}^\Phi &= K_x^\dagger T^T \\
&= U_{K1} \Sigma_{K1}^{-1} U_{K1}^T H^T \\
&= U_{K1} \Sigma_{K1}^{-1} A^\Phi \\
&= U_{K1} \Sigma_{K1}^{-1} P^\Phi \Sigma_A^\Phi (Q^\Phi)^T. \quad (54)
\end{aligned}$$

Similar to the linear case, we can show that under a mild condition, all the diagonal entries of Σ_A^Φ are ones, as summarized in the following theorem:

Theorem 9.1: Assume that $\text{rank}(K_x) = n - 1$ and $\text{rank}(Y) = k$ for multi-label problems. Then all diagonal elements of Σ_A^Φ are ones.

It follows from Theorem 9.1 that

$$G_{LS}^\Phi = U_{K1} \Sigma_{K1}^{-1} P_k^\Phi (Q^\Phi)^T. \quad (55)$$

Note that $\text{rank}(\Sigma_A^\Phi) = k$, thus $(C_{XX}^\Phi)^\dagger C_{HH}^\Phi$ contains k nonzero eigenvalues. When ℓ is chosen to be k , we have

$$G_{CCA}^\Phi = U_{K1} \Sigma_{K1}^{-1} P_k^\Phi. \quad (56)$$

The only difference between G_{CCA}^Φ and G_{LS}^Φ in the feature space is the orthogonal matrix $(Q^\Phi)^T$. We denote the least squares formulation for kernel CCA as “kLS-CCA”.

Based on this equivalence relationship between kernel CCA and kernel least squares, the kernel CCA can also be extended using the regularization techniques discussed in Section 4. In particular, we denote the 2-norm regularized kernel least squares CCA formulation as “kLS-CCA₂”.

Description of the Gene Image Data Sets

In our experiments, the Drosophila gene expression pattern images from the Berkeley Drosophila Genome Project (BDGP) database (<http://www.fruitfly.org>) are studied. The Drosophila gene expression pattern images document the spatial and temporal dynamics of gene expression and they are valuable tools for explicating the gene functions, interaction, and networks during Drosophila embryogenesis. To provide text-based pattern searching, the gene expression pattern images in the BDGP high-throughput study are annotated with anatomical and developmental ontology terms using a controlled vocabulary (CV). Currently, the annotation is performed manually by human curators. The annotated terms from the controlled vocabulary can be considered as labels in multi-label classification. As the number of available images is now rapidly increasing, it is therefore of great importance and interest to predict the terms to annotate gene expression pattern images automatically using multi-label learning. Figure 4 shows sample Drosophila gene expression pattern images and their corresponding annotated terms.

Empirical Studies of Kernel Methods

We study the empirical performance of kernel algorithms in comparison with linear algorithms, including:

- kernel CCA (denoted as kCCA) and its regularized version (denoted as krCCA);
- The proposed kernel least squares CCA (denoted as kLS-CCA) and its regularized version (denoted as kLS-CCA₂).

For all algorithms, we use the prefix “k” to denote the kernel algorithms. For example, “krCCA” denotes the regularized kernel CCA. For the kernel methods, the

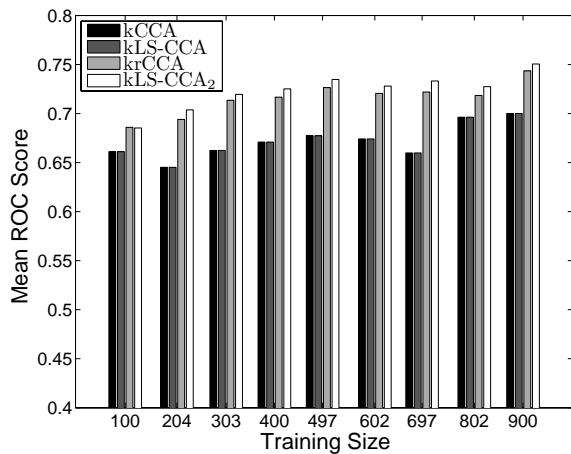


Fig. 5. Comparison of all kernel methods on the Gene Image 2 data set in terms of mean ROC scores as the size of training set increases.

Gaussian kernel is applied. The mean ROC scores over all labels and all partitions of training and test sets are reported.

We evaluate the equivalence relationship between kernel CCA (kCCA) and kernel least squares (kLS-CCA) on all data sets in Table 1. It is observed that the equivalence relationship always holds in our experiments given that $\text{rank}(K_x) = n - 1$.

We also test the performance of kernel algorithms on all data sets in Table 1, and the results are summarized in Table 3. It can be observed that kCCA and kLS-CCA also achieve the same performance, which validates the equivalence relationship between them. Kernel methods with regularization outperforms their counterparts without regularization. Most importantly, the kernel algorithms perform better than their linear counterparts, and kLS-CCA₂ achieves the best performance for all data sets.

Recall that the equivalence relationship tends to hold for kernel methods even when the size of training set is small. A similar sensitivity experiment is performed for these kernel methods on the Gene Image 2 data set. We increase the size of training set from 100 to 900, and mean ROC scores computed by different kernel algorithms are plotted in Figure 5. It can be observed that kCCA and kLS-CCA achieve the same performance in all cases, and the performance of regularized algorithms is much better than their counterparts without regularization.