LEARNING WITH HETEROGENOUS DATA SETS BY WEIGHTED MULTIPLE KERNEL CANONICAL CORRELATION ANALYSIS

Shi Yu   Bart De Moor   Yves Moreau

Katholieke Universiteit Leuven
Department of Electrical Engineering, ESAT-SCD-SISTA
Kasteelpark Arenberg 10, B-3001 Leuven(Heverlee), Belgium
{shi.yu, bart.demoor, yves.moreau}@esat.kuleuven.be

ABSTRACT

A new formulation of weighted multiple kernel based canonical correlation analysis (WMKCCA) is proposed in this paper. Computational issues are also considered in the proposed method to make it feasible on large data sets. This method uses incomplete Cholesky decomposition (ICD) and singular value decomposition (SVD) to approximate the original eigenvalue problem for low rank. For the weighted extension, an incremental eigenvalue decomposition method(EVD) is proposed to avoid redoing EVD each time weights are changed. Based on WMKCCA we proposed, a machine learning framework to extract common information among heterogeneous data sets is purposed and experimental results on two UCI data sets are reported.

1. INTRODUCTION

The goal of canonical correlation analysis (taking two data sets for example) is to identify canonical variables that minimize or maximize the linear correlations between the transformed variables [1]. Traditionally, canonical correlation analysis (CCA) is mainly employed on two data sets in observation space. Extension of CCA to multiple sets leads to different criteria of selecting the canonical variables, which are summarized as 5 different models: sum of correlation model, sum of squared correlation model, maximum variance model, minimal variance model and generalized variance model [2]. Kernel CCA is a generalization of CCA using the kernel trick to find canonical variables of data sets in kernel space [3] and its extension to multiple sets was given in [4]. Kernel CCA on multiple sets (MKCCA) was proposed as an independence measure to find uncorrelated variables in kernel space created by RBF kernels [4].

In this paper, we will show that MKCCA can also be regarded as a method to extract common information through maximization of the pairwise correlations among multiple data sets. A weighted extension of MKCCA can be easily derived with a natural link to the weighted objective function of MKCCA. The weighted MKCCA method can also be extended to out-of-sample points, which becomes important for model selection. Another important issue for MKCCA is that the problem scales up exponentially with the number of incorporated data sets and the number of samples. To make this method applicable on machines with standard CPU and memory, low rank approximation techniques based on Incomplete Cholesky Decomposition (ICD) and Singular Value Decomposition (SVD) are introduced in this paper. Moreover, for the weighted extension of MKCCA, a incremental algorithm is proposed to avoid recomputing eigenvalue decomposition each time weights of MKCCA are updated. To our knowledge, the weighted version of Kernel CCA and the incremental EVD algorithm for Kernel CCA have not been reported before.

The paper is organized as follows: Section 2 derives the mathematical formulation of WMKCCA. Section 3 discusses the computational issue of the low rank approximation of the MKCCA problem proposed in [4] and a novel incremental algorithm for WMKCCA. Section 4 presents a framework of plugging WMKCCA into common machine learning applications with a novelty of learning with common information among heterogeneous data sets. In Section 5 we report the experimental results of visualization and classification of 2 UCI pattern recognition data sets using WMKCCA. The computational savings of the incremental algorithm is also discussed. In Section 6 a conclusion is made.

2. FORMULATION OF WMKCCA

2.1. Linear CCA on multiple sets

The problem of CCA consists in finding linear relations between two sets of variables [1]. For the problem of two variables x_1 and $x_2$with zero means, the objective is to identify vectors w_1 and w_2 such that the correlation between the projected variables $w_1^T x_1$ and $w_2^T x_2$ is maximized:

$$\max_{w_1,w_2} \rho = \frac{w_1^T C_{x_1x_2} w_2}{\sqrt{w_1^T C_{x_1x_1} w_2}\sqrt{w_2^T C_{x_2x_2} w_2}} \tag{1}$$

where $C_{x_1x_1} = \mathcal{E}[x_1x_1^T], C_{x_2x_2} = \mathcal{E}[x_2x_2^T], C_{x_1x_2} = \mathcal{E}[x_1x_2^T]$.

Extending this objective function to multiple sets of variables $x_1, \ldots, x_m$ one obtains the form as

$$\max_{w_1 \cdots w_m} \rho = \frac{\mathcal{O}[x_1, \ldots, x_m]}{\prod_{i=m}^{m} \sqrt{w_i^T C_{x_i x_i} w_i}}, \tag{2}$$

where $\mathcal{O}[x_1, \ldots, x_m]$ is the objective function of correlations among multiple sets as the optimization criterion. To keep the problem analogous as the two-set one, we use the sum of correlation criterion and rewrite (2) as

$$\max_{w_i, 1 \le u < v \le m} \rho = \frac{\sum_{u,v} w_u^T C_{x_u x_v} w_v}{\prod_{i=1}^{m} \sqrt{w_i^T C_{x_i x_i} w_i}}, \tag{3}$$

which leads to the generalized eigenvalue problem:

$$\begin{bmatrix} 0 & \ldots & C_{x_1 x_m} \\ \vdots & \ddots & \vdots \\ C_{x_m x_1} & \ldots & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix} = \rho \begin{bmatrix} C_{x_1 x_1} & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & C_{x_m x_m} \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix} \tag{4}$$

where ρ is the correlation coefficient.

2.2. Kernel CCA on multiple sets

Kernel CCA is a nonlinear extension of CCA using kernel methods. The data is first mapped into a high dimensional Hilbert space induced by a kernel and then linear CCA is applied. In this way, linear correlation discovered in the Hilbert space corresponds to nonlinear correlation concealed in the observation space. Let $\{x_1^k, \ldots, x_m^k\}_{k=1}^N$ denote N observations of data set $x_1, \ldots, x_m$ respectively and $\phi_1(\cdot), \ldots, \phi_m(\cdot)$ as the feature maps from input spaces to the high dimensional Hilbert spaces for the different data sets. The centered kernel matrices of the m data sets becomes

$$\begin{aligned} \Phi_1 &= [\phi_1(x_1^{(1)})^T - \hat{\mu}_{\phi_1}; \ldots; \phi_1(x_1^{(N)})^T - \hat{\mu}_{\phi_1}] \\ & \qquad \cdots \\ \Phi_m &= [\phi_m(x_m^{(1)})^T - \hat{\mu}_{\phi_m}; \ldots; \phi_m(x_m^{(N)})^T - \hat{\mu}_{\phi_m}] \end{aligned} \tag{5}$$

the projection vectors $w_1, \ldots, w_m$ lie in the span of the mapped data

$$P_1 = \Phi_1 w_1, \ldots, P_m = \Phi_m w_m. \tag{6}$$

The resulted problem of kernel CCA can be deduced as the analogue of linear CCA problem on the projected data sets $P_1, \ldots, P_m$ in Hilbert space:

$$\max_{w_i, 1 \le u < v \le m} \rho = \frac{\sum_{u,v} w_u^T C_{\Phi_u \Phi_v} w_v}{\prod_{i=1}^{m} \sqrt{w_i^T C_{\Phi_i \Phi_i} w_i}}, \tag{7}$$

which leads to the generalized eigenvalue problem:

$$\begin{bmatrix} 0 & \ldots & \Omega_1 \Omega_m \\ \vdots & \ddots & \vdots \\ \Omega_m \Omega_1 & \ldots & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix} = \rho \begin{bmatrix} \Omega_1 \Omega_1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & \Omega_m \Omega_m \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix} \tag{8}$$

where Ω_i denotes the centered kernel matrix of i-th data set where Mercer's condition is applied within $\Omega = \Phi\Phi^T \in R^{N \times N}$:

$$\Omega_{(ij)} = \phi(x_i)^T \phi(x_j) = K(x_i, x_j) \tag{9}$$

The problem in (8) is ill-conditioned and the non-zero solutions of generalized eigenvalue problem are $\rho = \pm 1$. Hence it needs to be regularized to obtain meaningful estimation of canonical correlation in Hilbert space [4, 5, 6]. This paper employed the regularization method proposed in [5] which results in the following regularized general eigenvalue problem:

$$\begin{bmatrix} 0 & \ldots \Omega_1 \Omega_m \\ \vdots \quad \ddots & \vdots \\ \Omega_m \Omega_1 \ldots & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix} = \rho \begin{bmatrix} (\Omega_1 + \kappa I)^2 \ldots & 0 \\ \vdots \quad \ddots & \vdots \\ 0 & \ldots (\Omega_m + \kappa I)^2 \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix} \tag{10}$$

where κ is a small positive regularization constant.

2.3. Weighted MKCCA

Starting from the objective function in (7), the weighted extension of Multiple Kernel CCA can be formulated by employing additional weights $\xi_{u,v}$ on pairwise correlations:

$$\max_{w_i, 1 \le u < v \le m} \rho = \frac{\sum_{u,v} \xi_{u,v} w_u^T C_{\Phi_u \Phi_v} w_v}{\prod_{i=1}^{m} \sqrt{w_i^T C_{\Phi_i \Phi_i} w_i}}, \tag{11}$$

where $\xi_{u,v}$ is the scalar weight of correlation between e_u and e_v. If we denote the generalized eigenvalue problem in (10) as the form of $\Omega\alpha = \lambda\Omega_{\mathcal{R}}\alpha$, the weights of Kernel CCA can be decomposed as an additional positive definite matrix $\mathcal{W}$ multiplying at the left and right side of the matrix Ω:

$$\mathcal{W}\Omega\mathcal{W}\alpha = \lambda\Omega_{\mathcal{R}}\alpha \tag{12}$$

where

$$\mathcal{W} = \begin{bmatrix} \zeta_1 I & 0 & \ldots & 0 \\ 0 & \zeta_2 I & \ldots & 0 \\ \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & \ldots & \zeta_m I \end{bmatrix},$$

$$\Omega = \begin{bmatrix} 0 & \Omega_1 \Omega_2 & \ldots & \Omega_1 \Omega_m \\ \Omega_2 \Omega_1 & 0 & \ldots & \Omega_2 \Omega_m \\ \vdots & \vdots & \ddots & \vdots \\ \Omega_m \Omega_1 & \Omega_{m-1} \Omega_1 & \ldots & 0 \end{bmatrix},$$

$$\Omega_{\mathcal{R}} = \begin{bmatrix} (\Omega_1 + \kappa I)^2 & 0 & \ldots & 0 \\ 0 & (\Omega_2 + \kappa I)^2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & (\Omega_m + \kappa I)^2 \end{bmatrix},$$

$$\sum_{k=1}^{m} \zeta_k = k,$$

$$\xi_{u,v} = \psi \zeta_u \zeta_v,$$

$$\psi = \frac{1}{\sum_{1 \le i < j \le m} \zeta_i \zeta_j}.$$

Through this formulation, the weights of pairwise correlation ξ in objective function (11) are decomposed as the weights ζ on data sets. The sum of ζ is constrained to keep the mean as 1. ψ is a normalization parameter to make the sum of ξ equal to 1. This normalization constant ψ only affects the solution of eigenvalue but does not affect the eigenvector solution.

3. COMPUTATIONAL ISSUE

3.1. Standard Eigenvalue Problem for WMKCCA

Similar to the transformation presented in [4], the generalized eigenvalue problem in (12) can be written in following form:

$$[\mathcal{W}\Omega\mathcal{W} + \Omega_{\mathcal{R}}]\alpha = (\lambda + 1)\Omega_{\mathcal{R}}\alpha \tag{13}$$

The problem of finding maximal generalized eigenvalue in (12) is equivalent to finding the minimal generalized eigenvalue in (13) because if the generalized eigenvalues in (12) are $\{\lambda_1, -\lambda_1, \ldots, \lambda_p, \lambda_p, 0, \ldots, 0\}$, then correspondingly the generalized eigenvalues in (13) are $\{1+\lambda_1, 1-\lambda_1, \ldots, 1+\lambda_p, 1-\lambda_p, 1, \ldots, 1\}$. Since $\Omega_{\mathcal{R}}$ is regularized and can be decomposed as $\Omega_{\mathcal{R}} = \mathcal{C}^T\mathcal{C}$, defining $\beta = \mathcal{C}\alpha$,$K_\kappa = \mathcal{W}\Omega\mathcal{W} + \Omega_{\mathcal{R}}$ the problem can be transformed as the following:

$$\begin{aligned} \mathcal{K}_\kappa\alpha &= \lambda^\sharp\Omega_{\mathcal{R}}\alpha \\ \mathcal{K}_\kappa\alpha &= \lambda^\sharp\mathcal{C}^T\mathcal{C}\alpha \\ \mathcal{C}^{-T}\mathcal{K}_\kappa\mathcal{C}^{-1}\beta &= \lambda^\sharp\beta \end{aligned} \tag{14}$$

Since $\Omega_{\mathcal{R}}$ is a positive definite matrix in a diagonal form, we have

$$\mathcal{C}^T = \mathcal{C} = \Omega_{\mathcal{R}}^{1/2} = \begin{bmatrix} \Omega_1 + \kappa I \ldots & 0 \\ \vdots \quad \ddots & \vdots \\ 0 \quad \ldots & \Omega_m + \kappa I \end{bmatrix}. \tag{15}$$

Replacing (14) with (15), the problem is written in the form of standard eigenvalue problem:

$$\begin{bmatrix} I & \ldots & \zeta_1\zeta_m I r_\kappa(\Omega_1) r_\kappa(\Omega_m) \\ \vdots & \ddots & \vdots \\ \zeta_1\zeta_m I r_\kappa(\Omega_m) r_\kappa(\Omega_1) & \ldots & I \end{bmatrix} \beta = \lambda^\sharp\beta, \tag{16}$$

where $r_\kappa(\Omega_i) = \Omega_i(\Omega_i + \kappa I)^{-1} = (\Omega_i + \kappa I)^{-1}\Omega_i$

If eigenvalues $\lambda^\sharp$ and eigenvectors β are solved from (16), the eigenvalues and eigenvectors of problem (12) is $\lambda^\sharp$ and $\mathcal{C}^{-1}\beta$. More formally, the eigenvectors α_i of problem (10) are equal to

$$\alpha_i = (\Omega_i + \kappa I)^{-1}\beta_i. \tag{17}$$

3.2. Incomplete Cholesky Decomposition

According to incomplete Cholesky decomposition, full rank (N) centered kernel matrix Ω_i can be factorized as $\Omega_i \approx G_iG_i^T$, where G_i is in low rank M_i ($M_i \leq N$). Apply singular value decomposition on G_i to obtain $N \times M_i$ matrix U_i with orthogonal columns and $M_i \times M_i$ diagonal matrix Λ_i such that:

$$\Omega_i \approx G_iG_i^T = U_i\Lambda_iV_i^T(U_i\Lambda_iV_i^T)^T = U_i\Lambda_i^2U_i^T. \tag{18}$$

Denoting E_i as the orthogonal complement of U_i such that (U_iE_i) is a full rank $N \times N$ matrix, one obtains:

$$\Omega_i \approx U_i\Lambda_i^2U_i^T == (U_iE_i)\begin{bmatrix} \hat{\Lambda}_i & 0 \\ 0 & 0 \end{bmatrix}(U_iE_i)^T \tag{19}$$

For regularized matrices in (10), one obtains:

$$r_\kappa(\Omega_i) \approx (U_iE_i)\begin{bmatrix} R_i & 0 \\ 0 & 0 \end{bmatrix}(U_iE_i)^T = U_iR_iU_i^T \tag{20}$$

where R_i is the diagonal matrix obtained from the diagonal matrix $\hat{\Lambda}_i$ by transformation $R_i^j = \frac{\hat{\Lambda}_i^j}{\hat{\Lambda}_i^j + \kappa}$ to its elements. Replacing (16) with (20), decomposing (16) as

$$UR_\kappa U^T\beta = \lambda^\sharp\beta, \tag{21}$$

where

$$U = \begin{bmatrix} U_1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & U_m \end{bmatrix}$$

$$R_\kappa = \begin{bmatrix} I & \ldots & \zeta_1\zeta_m I R_1 U_1^T U_m R_m \\ \vdots & \ddots & \vdots \\ \zeta_1\zeta_m I R_m U_m^T U_1 R_1 & \ldots & I \end{bmatrix} \tag{22}$$

Since R_k is deduced from a similar matrix transformation, the eigenvalues in are equivalent, moreover, the eigenvectors of the low rank approximation is related to the full rank problem by the following transformation:

$$\begin{aligned} UR_kU^T\beta &= \lambda^\sharp\beta \\ R_kU^T\beta &= \lambda^\sharp U^T\beta \\ R_k\gamma &= \lambda^\sharp\gamma \end{aligned} \tag{23}$$

Hence, once we obtained the eigenvector γ_i in low rank approximation problem (23) it can be restored to full rank problem in (16) through $\beta_i = U_i\gamma_i$. Furthermore, the generalized eigenvector α_i of the original problem can be calculated as formula (17), hence we have:

$$\alpha_i \approx (\Omega_i + \kappa I)^{-1}U_i\gamma_i \tag{24}$$

We have several parameters involved in MKCCA computation: κ the regularization parameter, η the precision parameter for incomplete Cholesky decomposition, τ the cut value of eigenvalues determining the size of U_i and λ_i in singular value decomposition of Ω_i.

3.3. Incremental EVD solution for WMKCCA

Starting from the weighted problem expressed in (12), the update of weights in WMKCCA can be expressed as an additional update matrix $\mathcal{V}$ multiplied at the left and right sides of the WMKCCA formulation:

$$\mathcal{V}\mathcal{W}\Omega\mathcal{W}\mathcal{V}\alpha = \lambda\Omega_{\mathcal{R}}\alpha \tag{25}$$

where

$$\mathcal{V} = \begin{bmatrix} v_1 & 0 & \ldots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \ldots & 0 & v_m \end{bmatrix} \tag{26}$$

$v_1, \ldots, v_m$ are the update ratios of weights corresponding to $\zeta_1, \ldots, \zeta_m$.
Following the analog steps from (13) to (16), the standard eigenvalue problem with updated weights is in the form of:

$$\begin{bmatrix} I & \ldots & v_1v_m\zeta_1\zeta_m I R_1 U_1^T U_m R_m \\ \vdots & \ddots & \vdots \\ v_1v_m\zeta_1\zeta_m I R_m U_m^T U_1 R_1 & \ldots & I \end{bmatrix}\gamma = \lambda^\sharp\gamma \tag{27}$$

For simplicity we denote the matrix of eigenvalue problem before weight updating is $\mathcal{R}_k$, the one after updating is $\mathcal{R}_{new}$, we define $E = \mathcal{R}_{new} - \mathcal{R}_k$. For the weights updated problem, we need to solve $\mathcal{R}_{new}\gamma = \lambda^\sharp\gamma$. Obviously, we could approximate the solution of the new problem on the basis of the previous solution of $\mathcal{R}_k\gamma = \lambda^\sharp\gamma$ without redoing the eigenvalue decomposition from the scratch. From the previous solution we have:

$$\gamma_k\Lambda_k\gamma_k^T = \mathcal{R}_k \tag{28}$$

The updated problem is equal to adding E on both side of equation:

$$\begin{aligned}\gamma_k\Lambda_k\gamma_k^T + E &= \mathcal{R}_k + E\\ \gamma_k(\Lambda_k + \gamma_k^T E\gamma_k)\gamma_k^T &= \mathcal{R}_{new}\\ \gamma_k T\gamma_k^T &= \mathcal{R}_{new}\end{aligned} \tag{29}$$

Since in (27) weight updating only affect the off-diagonal elements of the matrix. Moreover, due to the constraints of weights matrix in (12) where the mean value of ζ_k is 1, the update parameters is also constrained within a certain scope. Usually, for small updates these values are close to 1. The matrix E is in the form of:

$$E = \begin{bmatrix} 0 & E_{1,2} & \dots E_{1,m} \\ \vdots & \ddots & \vdots \\ E_{m,1} & E_{m,2} & \dots \quad 0 \end{bmatrix} \tag{30}$$

where

$$E_{i,j} = (v_i v_j - 1)\zeta_i\zeta_j IR_i U_i^T U_j R_j \tag{31}$$

which only have non-zero values at off-diagonal positions and most of the elements are close to 0. Hence, $\gamma_k^T E\gamma_k$ is also a sparse matrix with most of the off diagonal elements are close to 0. So, the matrix T in (29) is a nearly diagonal matrix thus can be solved more efficiently by iterative eigenvalue decomposition algorithms. Hence, instead of doing EVD each time with updated weights, we stored the previous EVD solution and computed the EVD solution of T incrementally.

4. WMKCCA FOR MACHINE LEARNING

WMKCCA can extract common information among multiple heterogeneous data sets. Given a group of objects, usually multiple observations were obtained by different methods and conditions, however, the inter-relationships among these objects follow a intrinsic pattern. By WMKCCA, the relationships are investigated in a Hilbert space and patterns of these relationships from multiple observations are compared. When more than two observations are presented, the advantage of weighted extension of kernel CCA is the flexibility to bias the model towards several important observations without ignoring the information of the others. These relationships and patterns are useful for machine learning applications. Hence an integrative framework for WMKCCA based machine learning is presented in Figure 1. The framework integrates WMKCCA with supervised machine learning where the validation data and test data are projected onto the embedding of the training data through out-of-sample projection [7]. The model for WMKCCA is selected by evaluating the machine learning performance on the validation set so that the parameters of kernel function and the weights assigned on correlations are optimized.

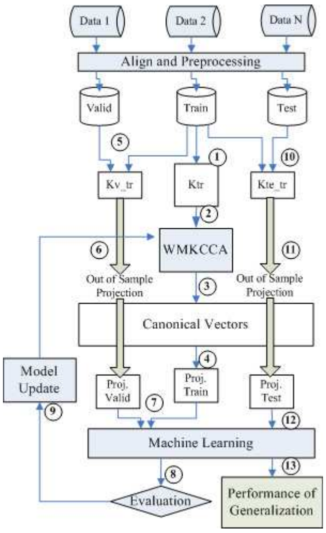

Fig. 1. A framework for learning using WMKCCA with heterogeneous data sources

5. RESULTS ON EXPERIMENTAL DATA SETS

5.1. Data sets and kernel functions

We adopted two pattern recognition data sets, Pen-Based Recognition of Handwritten Digits and Optical Recognition of Handwritten Digits, from the UCI machine learning data archives. For abbreviation, we denote them as PenData and OptData respectively. Both data sets have 10 labels corresponding to digits from 0 to 9. PenData has 16 input attributes measured from 0 to 100 and OptData has 64 attributes measured from 0 to 16. We extracted 3750 samples (375 samples for each digit) from the training part of both data sets, 80% of them used for training and 20% used for validation. We adopted their original test data as test set(3498 for PenData, 1797 for OptData). We applied the RBF kernel to both data sets and the kernel width was selected as the mean of covariance (for PenData $\sigma = 97$, for OptData $\sigma = 13$). Moreover, we transformed the class information of data into another kernel matrix of labels. Firstly, the vector of class labels is coded into an $N \times 10$ matrix L where the *i*-th column represents the label of $i - 1$ digit. For example, for digit ”6” the *7*-th column is assigned to 1 and other columns are 0. Then, the label matrix is transformed into a kernel matrix by the linear kernel $L * L^T$. So, in our training step we produced three 3000×3000 kernel matrices, $K_{pen}, K_{opt}, K_{label}$.

5.2. Visualization of canonical projections

Similar to the kernel CCA visualization method presented in [8] on single data sets, we visualized PenData and OptData simultaneously in lower dimensional space. In Figure 2 we presented a series of figures visualizing all 3000 training points in the space spanned by the 1st and 2nd canonical variate obtained by KCCA and WMKCCA. By adjusting the weights, we are able to discover the difference and transformative pattern of integrating two heterogenous data sets. The 1st row shows the projections of two set KCCA on K_{pen}, K_{label} and K_{opt}, K_{label} respectively. The next three rows show the projections produced by WMKCCA on three sets with different weights. When we assign a large weight on PenData (1.99) and small weight on OptData (0.01), the projection of PenData is quite similar to the result of KCCA, however, the projection of OptData is quite different. Similarly, large weight on OptData makes WMKCCA a similar result with KCCA on OptData but not for PenData. When equal weights are assigned on all three sets, not only the correlations between observations and label but also the correlation between K_{pen}, K_{opt} is maximized.

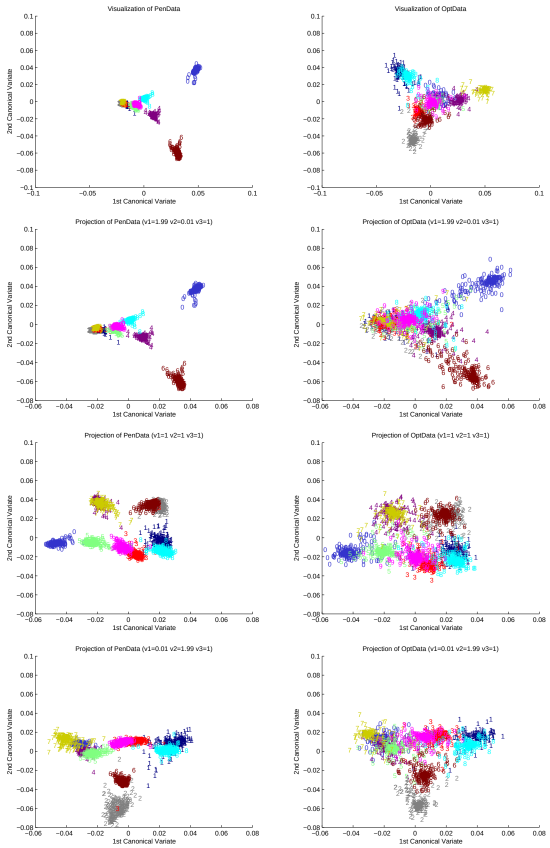

Fig. 2. Visualization of PenData and OptData by KCCA and WMKCCA in the space spanned by first 2 canonical vectors

5.3. WMKCCA based Classification on canonical spaces

We applied a centroid classification approach on the projected data in canonical spaces which treats each set of digits as a cluster and calculates the centroid as a function of the mean across all dimensions in canonical space. When new points are presented for classification, the Euclidean distance from the new point to each cluster centroid is calculated. We compared the distances of new data to all 10 clusters centroids and label the data with the one has the shortest distance. The validation data and test data were firstly projected into the canonical spaces of training data by out-of-sample projection, then classified by centroid method according to their distance to the cluster centroids of training data. The accuracy of classification was evaluated calculating the percentage of correctly classified data of all labels.

We benchmarked the accuracy of validation sets with different weights, using the incremental EVD method discussed in Section 2. The weights on $K_{pen}, K_{opt}, K_{label}$, denoted as $\zeta_{pen}, \zeta_{opt}, \zeta_{label}$ are benchmarked by grid search from 0.1 to 2.9 with step 0.1 with the constraint that $\zeta_{pen} + \zeta_{opt} + \zeta_{label} = 3$. We also compared the performance of classification in canonical subspaces of different sizes. For each validation, the data was projected to the subspace spanned by 10, 100 and 500 canonical vectors respectively. The optimal weights and subspace were selected by the average of classification accuracies on two validation data sets. Then the test data was fed into the WMKCCA model with the selected weights and projected to the selected subspace hence the accuracy on test data is obtained. We compared the accuracy obtained on WMKCCA model with the results from other methods mentioned in the literature in Table 1. Results of KCCA, MKCCA and WMKCCA were obtained by methods mentioned in this paper. Results of other methods were referenced from the literature. The best result of WMKCCA on PenData (0.9794) and on OptData (0.9716) was obtained when $\zeta_{pen} = 1.3, \zeta_{opt} = 1.3, \zeta_{label} = 0.4$ and the projection space set to 500 vectors. As it is shown, the result of WMKCCA is better than the one of MKCCA with equal weights. It is also better than the results of KCCA that applied on observations and labels of individual data set using all 3750 samples for training. It also seems that the result of WMKCCA is comparable to the best results of other methods reported by far. The result on OptData is only slightly worse than the result produced by the hierarchical combining of 45 SVMs [9].

5.4. Efficiency of Incremental EVD solution

We benchmarked the direct EVD algorithm and the incremental EVD algorithm in WMKCCA experiments with different matrix sizes and update scales. The benchmark did not consider previous ICD and SVD but only compare the

Table 1. Classification accuracy on Test Data

METHODS	PenData	OptData	Notes
WMKCCA	0.9794	0.9716	See text
MKCCA	0.9766	0.9688	Equal weights
KCCA	0.9783	0.9711	3750 training
Linear SVM	0.9494	0.9602	
RBF DDA SVM [10]	0.9708	0.9722	
SVM Ensemble [9]	N/A	0.9783	45 SVMs
MLP [10]	0.9703	0.9517	
kNN [10]	0.9771	0.9649	k=3
Bayes+PCA [11]	0.9763	0.9694	

CPU time of solving the eigenvalue problem in (23) by direct method with solving the transformed problem in (29) by incremental method. For the incremental method, the CPU cost of calculating E, matrix multiplication of calculating T, EVD of T and matrix multiplication of calculating canonical vectors are taken into account. We adjusted the scale of the problem by increasing the size of training set. We selected 100, 500, 1000, 1500 and 2000 data samples from two data sets and fed them into the WMKCCA algorithm ($\eta = 0.9$, $\kappa = 0.1$, $\tau = 0.9$). After ICD and SVD, the matrix R_k had size 160, 701, 1323, 1911 and 2483 respectively. We also adjusted the scale of the weight update parameter, which is denoted as the ratio between new weight and old weight in weight update matrix $\mathcal{V}$. As we discussed before, when that ratio is close to 1, the matrix E becomes sparse off-diagonal. When that ratio is too big, E is not necessarily a sparse off-diagonal matrix, thus the assumption of incremental EVD does not hold. We tried three scales 1.1, 2 and 10 which represents weak update, moderate update and strong update of WMKCCA model respectively. For the problem of 3 data sets, when the update scale is set to 2 and 10, the mean value of ζ does not necessarily equal to 1 hence the constraint in (12) does not hold. The benchmark was run on a PC with Intel Core 2 CPU of 1.86GHz and 2G physical memory. The software package for simulation was MATLAB 2006a. The EVD algorithm used the eig function in MATLAB, which is based on QR factorization. From Figure 3, we can see that the incremental algorithm significantly saves CPU time when the update is small and moderate. However, when the update is large, the matrix T is no longer nearly diagonal so the incremental algorithm paid additional cost for matrix multiplication.

6. CONCLUSIONS

A new weighted formulation of kernel CCA on multiple sets is proposed. Using low rank approximation and incremental eigenvalue algorithm, WMKCCA is applicable to machine learning problems as a flexible model for common information extraction among multiple data sources. The paper presents an experiments of applying WMKCCA to extract and visualize correlations between observations and its class labels among two heterogeneous OCR data sources. Furthermore, projections of data in canonical spaces obtained by WMKCCA are fed into a simple clustering centroid classification algorithms and it shows comparable classification accuracy with respect to the best reported results in the literature.

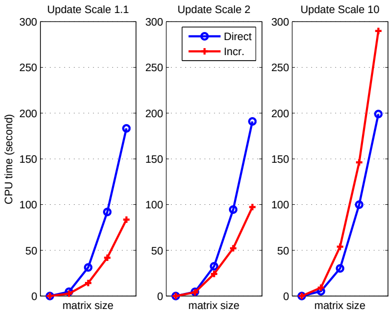

Fig. 3. Comparison of CPU time between direct EVD method and Incremental EVD method

7. REFERENCES

[1] H. Hotelling, "Relations between two sets of variates," *Biometrika*, vol. 28, pp. 321–377, 1936.

[2] J.R. Kettenring, "Canonical analysis of several sets of variables," *Biometrika*, vol. 58, pp. 433–451, 1971.

[3] S. Akaho, "A kernelmethod for canonical correlation analysis," in *International Meeting of Psychometric Society (IMPS2001)*, 2001.

[4] Francis R. Bach and Michael I. Jordan, "Kernel independent component analysis," *J. Mach. Learn. Res.*, vol. 3, pp. 1–48, 2003.

[5] David R. Hardoon, Sandor R. Szedmak, and John R. Shawe-taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Comput.*, vol. 16, no. 12, pp. 2639–2664, 2004.

[6] Arthur Gretton, Ralf Herbrich, Alexander Smola, Olivier Bousquet, and Bernhard Scholkopf, "Kernel methods for measuring independence," *J. Mach. Learn. Res.*, vol. 6, pp. 2075–2129, 2005.

[7] Y. Bengio, O. Delalleau, N. Le Roux, J.-F. Paiement, P. Vincent, and M. Ouimet, "Learning eigenfunctions links spectral embedding and kernel PCA," *Neural Computation*, vol. 16, no. 10, pp. 2197–2219, 2004.

[8] Yuan chin Ivan Chang, Yuh-Jye Lee, Hsing-Kuo Pao, Meihsien Lee, and Su-Yun Huang, "Data visualization via kernel machines," Tech. Rep., Institute of Statistical Science, Academia Sinica, Taiwan, China, 2004.

[9] H.C. Kim, S. Pang, H.M. Je, D. Kim, and S.Y. Bang, "Pattern classification using support vector machine ensemble," *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 2, pp. 160–163, 2002.

[10] A. L. I. Oliveira, F. B. L. Neto, and S. R. L. Meira, "Improving rbf-dda performance on optical character recognition through parameter selection," in *ICPR '04*, Washington, DC, USA, 2004, pp. 625–628, IEEE Computer Society.

[11] Lasse Holmstrom and Fabian Hoti, "Application of semiparametric density estimation to classification," in *ICPR '04*, Washington, DC, USA, 2004, pp. 371–374, IEEE Computer Society.