Int J Comput Vis (2015) 113:176–192
DOI 10.1007/s11263-014-0783-8

Relatively-Paired Space Analysis: Learning a Latent Common Space From Relatively-Paired Observations

Zhanghui Kuang · Kwan-Yee K. Wong

Received: 29 April 2014 / Accepted: 16 October 2014 / Published online: 12 November 2014

Abstract Discovering a latent common space between different modalities plays an important role in cross-modality pattern recognition. Existing techniques often require absolutely-paired observations as training data, and are incapable of capturing more general semantic relationships between cross-modality observations. This greatly limits their applications. In this paper, we propose a general framework for learning a latent common space from relatively-paired observations (i.e., two observations from different modalities are more-likely-paired than another two). Relative-pairing information is encoded using relative proximities of observations in the latent common space. By building a discriminative model and maximizing a distance margin, a projection function that maps observations into the latent common space is learned for each modality. Cross-modality pattern recognition can then be carried out in the latent common space. To speed up the learning procedure for large scale training data, the problem is reformulated into learning a structural model, which is efficiently solved by the cutting plane algorithm. To evaluate the performance of the proposed framework, it has been applied to feature fusion, cross-pose face recognition, text-image retrieval and attribute-image retrieval. Experimental results demonstrate that the proposed framework outperforms other state-of-the-art approaches.

Keywords Multi-modality analysis · Multi-view analysis · Absolutely-paired observations · Relatively-paired observations · Structural learning · Cutting plane algorithm

1 Introduction

It is very common that an object can have very different representations in different modalities. For instance, printed and hand-written forms of the same character can look very different, so are face photo and face sketch of the same person. Humans have little problem in recognizing objects across different modalities (e.g., matching face sketches to face photos). In contrast, conventional machine learning methods, such as k-NN classifiers, perform poorly in cross-modality pattern recognition since they assume both the training data and test patterns are randomly sampled from the same distribution (which is not the case in cross-modality pattern recognition) (Tenenbaum and Freeman 2000).

There exist a number of research studies in the literature targeting at cross-modality pattern recognition, which can be roughly classified into one of the three main approaches. The first approach consists of transforming one modality into another in a preprocessing step (Zhou et al. 2012; Blanz et al. 2005). The second approach is by extracting modality-invariant features to represent an object (Lowe 2004; Zhang et al. 2011). A major limitation of these two approaches is that methods based on these approaches are usually tailor-made for each different modality pair involved in different recognition tasks. The third approach is to find an underlying latent common space shared between different modalities (Tenenbaum and Freeman 2000; Knutsson et al. 1997; Lin and Tang 2006; Sun et al. 2008; Prince et al. 2008). Unlike the first two approaches, the third approach does not depend on task-dependent knowledge. Methods based on the third approach

Communicated by Tilo Burghardt , Majid Mirmehdi, Walterio Mayol and Dima Damen.

Z. Kuang (✉)· K.-Y. K. Wong
Department of Computer Science, The University of Hong Kong,
Pokfulam, Hong Kong
e-mail: kuangzhh@gmail.com

K.-Y. K. Wong
e-mail: kykwong@cs.hku.hk

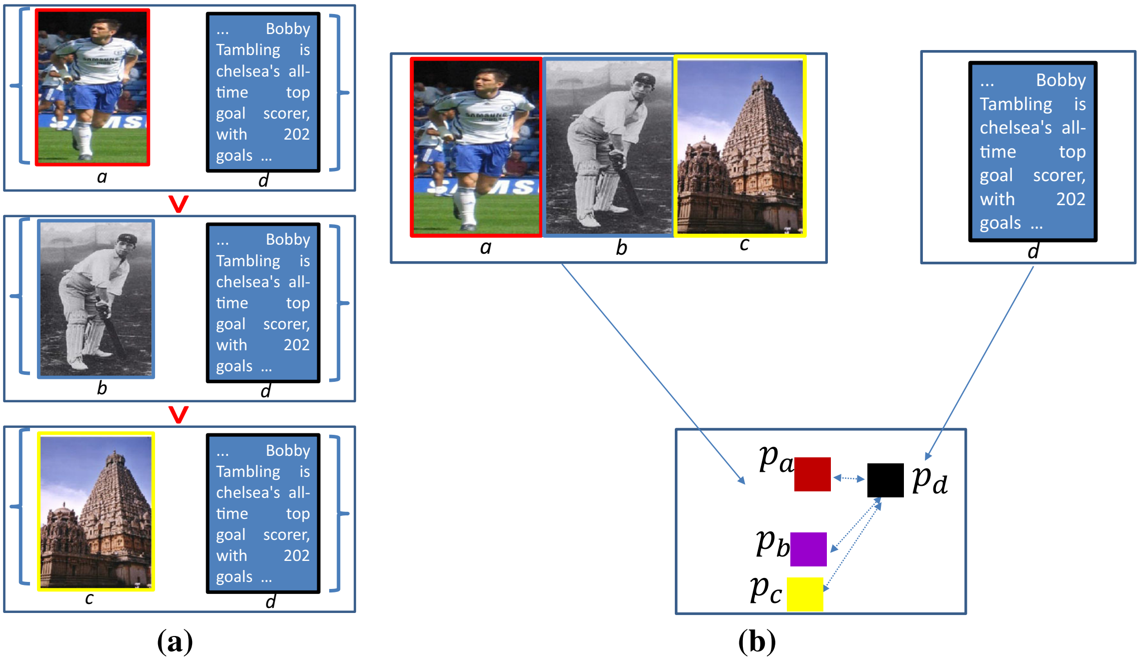
... Bobby Tambling is chelsea's all-time top goal scorer, with 202 goals ...
a
d
∨
b
d
∨
c
d
(a)
a
b
c
d
p_a
p_d
p_b
p_c
(b)

Fig. 1 Illustration of the proposed method. **a** shows the relative-pairing relationships between observations from image and text modalities. ∨ indicates being more likely paired. **b** shows the distances between the projections of observations in the latent common space

are therefore general frameworks that can be applied to different applications. Existing methods of the third approach often require absolutely-paired observations as training data. We refer to them as *Absolutely-Paired Space Analysis* (APSA). These methods assume the projections of paired observations being dependent in the latent space, and can only represent a binary relationship between observations (i.e., either paired observations or non-paired observations).

In many application scenarios, however, it is more suitable to consider relatively-paired observations (i.e., two observations from different modalities are more-likely-paired than another two) than absolutely-paired observations. For instance, given an input text query, an image search-engine (such as Google) will return a list of most probable images. The images selected by the user are not absolutely-paired with the input text, but instead are more-likely-paired with the input text than other returned images. In fact, relative-pairing is a general pairing relationship that also covers absolute-pairing. One can safely consider two observations that are absolutely-paired being more-likely-paired than other non-paired observations. Another advantage of considering relatively-paired training data is that label information of the observations can be easily integrated to boost recognition performance. It is reasonable to assume observations with the same label being more-likely-paired than those with different labels. This strategy can be used to reduce within-class scatter while maximizing between-class scatter in the latent common space, as well as increase the minimum distance between observations with different labels in the latent common space.

In this paper, we introduce a general framework named *Relatively-Paired Space Analysis* (RPSA) which works on relatively-paired observations. Note that RPSA is *not* a trivial extension of APSA as they are based on completely different models. APSA methods are often based on generative models (Knutsson et al. 1997; Bach and Jordan 2005; Prince et al. 2008) which either explicitly or implicitly assume the distributions of model parameters and noise (e.g., Gaussian distribution). The final estimation will be unreliable when real data do not fit the assumption. As opposed to APSA, our method is based on a discriminative model that has no distribution assumption. Besides, APSA methods learn a projection function for each modality by exploring the statistics dependence of the projections of absolutely-paired observations in the latent common space. This one-to-one absolute-pairing requirement makes them not suitable for relatively-paired observations. In our proposed framework, we compute the projection functions by preserving the relative proximities of observations in the latent common space.

Figure 1 illustrates the principle of the proposed method based on the data set Wiki Text-Image (Rasiwasia et al. 2010) used in our experiments. The data set has two modalities, namely image modality and text modality. We select three images a, b and c and one text article d from it. a, b and c show a soccer player, a baseball player, and a building respectively while d describes a soccer team "Chelsea" and their team members. Obviously, d is highly relevant to a, slightly relevant to b (since both b and d have the concept of sports), and little relevant to c. Therefore, a is more likely paired with d than b, and b is more likely paired with d than c. Assume p_a, p_b, p_c and p_d are projections of a, b, c and d in the latent common space respectively. The proposed method attempts to learn one projection function for each modality so that the distance between p_a and p_d is shorter than that between p_b and p_d which is shorter than that between p_c and p_d.

We first learn the model parameters of RPSA via alternating variable method (Shen et al. 2011), and find that the training time increases dramatically as the number of training triplets increases. To this end, we reformulate the RPSA problem into learning a structural model (Tsochantaridis et al. 2004), and a scalable approach based on the cutting plane algorithm is proposed to solve this problem.

We validate our RPSA framework by applying it to feature fusion, cross-pose face recognition, text-image retrieval and attribute-image retrieval. Experimental results demonstrate that our proposed framework outperforms other state-of-the-art approaches. The main contributions of this paper are

1. We propose a general framework called Relatively-Paired Space Analysis (RPSA) for automatically learning a latent common space between different modalities from relatively-paired observations, which, to the best of our knowledge, has not been explored before.
2. We propose a scalable optimization approach based on the cutting plane algorithm to learn the model parameters of RPSA.
3. We apply our proposed RPSA framework to feature fusion, cross-pose face recognition, text-image retrieval and attribute-image retrieval. RPSA achieves significant improvement in recognition and retrieval performance compared with other state-of-the-art methods.

Preliminary results of this work had been published in the proceedings of the British Machine Vision Conference 2013 in Bristol, UK (Kuang and Wong 2013). The differences between this version and the previous one are as follows:

1. A more detailed and up-to-date survey of multi-modality analysis is included in Sect. 2.
2. A term which measures the sum of the distances between points and their corresponding target neighbors is added in the proposed objective energy function to boost the performance of RPSA.
3. The RPSA problem is reformulated as a structural learning model and a scalable approach based on the cutting plane algorithm is proposed to solve it.
4. New experiments on Wiki Text-Image data set (Rasiwasia et al. 2010) and Public Figures Face Database (Parikh and Grauman 2011; Kumar et al. 2009) have been carried out for testing and the results are compared with state-of-the-art techniques.

2 Related Work

There exist a large number of research studies on cross-modality pattern recognition in the literature. Due to page limitation, however, we focus our discussion only on those most relevant work that automatically learn a latent common space between different modalities. Knutsson et al. (1997) proposed the Canonical Correlation Analysis (CCA) which finds a latent common space by maximizing the correlation of the projections of cross-modality observations. Sun et al. (2008) extended CCA by maximizing the within-class correlations and minimizing between-class correlations. Torre and Black (2001) developed the Asymmetric Coupled Component Analysis (ACCA) to explicitly learn the dependence of projections in a latent common space. Similarly, Lin and Tang (2005) explored the coupled space by alternatively maximizing the correlation of projections of cross-modality observations and finding the relations between these projections. Different from CCA, Partial Least Square (PLS) (Prince et al. 2008; Rosipal and Krämer 2006) chooses linear mappings such that the covariance between projections of cross-modality observations in the latent common space is maximized. Bilinear Model (BLM) (Tenenbaum and Freeman 2000) was proposed to separate style and content. Observations with different styles (from different modalities) for an object are encouraged to map to the same content in a latent common space by solving two-factor tasks. Recently, Sharma and Kumar (2012) proposed a General Multi-view Analysis (GMA) approach which learns a latent common space by solving a generalized eigenvalue problem. Kan et al. (2012) introduced a Multi-view Discriminant Analysis (MvDA) method to seek for a projection function for each modality by optimizing a generalized Rayleigh quotient. Besides, researchers have proposed advanced nonlinear methods based on the Gaussian Process Latent Variable Model (GPLVM) (Shon et al. 2006; Navaratnam et al. 2007; Ek et al. 2008). All the above methods require absolutely-paired observations as training data. Recently, Lampert and Krömer (2010) learned a latent space based on weakly-paired data (i.e., subsets of observations of one modality are paired with those of another modality) by alternatively finding element pairs and maximizing covariance of projections of cross-modality observations. Different from previous work, our proposed framework depends on neither prior distribution assumptions nor statistics computations, and learns a latent common space by preserving relative proximities of the relatively-paired training data in the latent common space.

Metric learning can be interpreted as finding a latent space for a single-modality observation space by linear projection. Xing et al. (2002) proposed to minimize the distances between samples from a similar set while keeping the distances of those from a dissimilar set above a threshold. Goldberger et al. (2004) directly maximized a stochastic variant of the leave-one-out k-NN score on the training set. Since then, many other methods (Weinberger et al. 2006; Davis et al. 2007; Shen et al. 2009; Zheng et al. 2013) were proposed to achieve a similar goal. Specifically, Zheng et al. (2013) proposed a metric learning approach named Relative Distance Comparison (RDC) to solve reidentification. They formulated RDC to maximize the likelihood of a pair of true matches having a relatively smaller distance than that of a wrong match pair in a soft discriminant manner. However, these methods only focus on a single modality. For cross-modality pattern recognition problems as studied in this paper, observations from different modalities are heterogeneous, and metric learning approaches cannot get good results (Knutsson et al. 1997; Sun et al. 2008; Torre and Black 2001; Wu et al. 2010). Our experiments on cross-pose face recognition support this conclusion. In some cases, observations from different modalities have different numbers of dimension (our experiments on feature fusion and image-text retrieval are examples). Metric learning approaches cannot be used to do cross-modality pattern recognition since metrics such as Mahalanobis distance, require the same dimension number for different observations (which is not the case in this task). Our work is not a trivial extension of metric learning. First, the relative-pairing information which encodes the relationship between observations from different modalities is novel. Second, the scalable optimization method based on structural learning to speed up multi-modality analysis was not explored before. Recently, Quadrianto and Lampert (2011) extended metric learning to multiple modalities by explicitly modeling linear projections. Their objective function is non-convex and thus the final optimum obtained depends on initialization. Moreover, their method requires the dimension of the latent common space to be known a priori. As opposed to their method, our model is convex which guarantees a global optimum, and can find a latent common space with any dimension in a single optimization.

Exploiting latent spaces can also be found in related research studies, such as local metric learning (Andrea et al. 2007), hashing (Bronstein and Bronstein 2010), multitask learning (Parameswaran and Weinberger 2010), domain adaption (Saenko et al. 2010) and ranking (Wang et al. 2009). However, their goals are very different from the one in this paper.

3 Relatively-Paired Space Analysis

In this section, we describe our RPSA framework for learning a latent common space from relatively-paired observations. The goal is to find linear mappings that project observations from different modalities into a latent common space in which the relative proximities of the relatively-paired observations are preserved.

3.1 Preliminaries

Let us define some notation first. We use boldface uppercase, lowercase and calligraphic letters (e.g., $\mathbf{X}$, $\mathbf{x}$ and $\mathcal{X}$) to denote matrices, vectors and sets, respectively. X_{ij} denotes the (i, j)th entry of $\mathbf{X}$, x_i denotes the ith entry of $\mathbf{x}$, and x_{ij} denotes the jth entry of $\mathbf{x}_i$. $\mathbf{X} \succeq 0$ denotes $\mathbf{X}$ being a positive semi-definite matrix. Let $\mathrm{Tr}(\mathbf{X})$ denote the trace of $\mathbf{X}$ and $\mathbf{X}^{\mathrm{T}}$ its transpose, and the inner product of two matrices $\langle \mathbf{X}, \mathbf{Y} \rangle$ can then be represented by $\mathrm{Tr}(\mathbf{X}^{\mathrm{T}}\mathbf{Y})$. For a symmetric matrix $\mathbf{X}$, its eigenvalue decomposition is given by $\mathbf{X} = \mathbf{U}\varLambda\mathbf{U}^{\mathrm{T}}$ with $\mathbf{U}$ being an orthogonal matrix. The positive part of the matrix $\mathbf{X}$ is defined as

$$(\mathbf{X})_+ = \mathbf{U}\max(\varLambda, \mathbf{0})\mathbf{U}^{\mathrm{T}}, \tag{1}$$

and the negative part as

$$(\mathbf{X})_- = \mathbf{U}\min(\varLambda, \mathbf{0})\mathbf{U}^{\mathrm{T}}. \tag{2}$$

Clearly, $\mathbf{X} = (\mathbf{X})_+ + (\mathbf{X})_-$ always holds true.

3.2 The RPSA Model

Consider a set of M modalities $\{\varOmega_1, \varOmega_2, \ldots, \varOmega_M\}$ with dimensions $\{d_1, d_2, \ldots d_M\}$ respectively, and a training data set of N observations $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ with a corresponding flag set $\{t_1, t_2, \ldots, t_N\}$ such that $t_i \in \{1, \ldots, M\}$ indicates that $\mathbf{x}_i$ comes from $\varOmega_{t_i}$. Let the relative-pairing knowledge of the observations be represented by a set of triplets $\mathcal{T} = \{(i, j, k)\}$, where each triplet (i, j, k) encodes that $\mathbf{x}_i$ and $\mathbf{x}_j$ are more-likely-paired than $\mathbf{x}_i$ and $\mathbf{x}_k$. Note that $\mathbf{x}_i$, $\mathbf{x}_j$ and $\mathbf{x}_k$ can come from either the same or different modalities. When they are from the same modality, "being more-likely-paired" means "being more similar".

To learn a latent common space Z with dimension d_z, we seek a $d_z \times d_m$ linear projection matrix $\mathbf{W}_{\varOmega_m}$ for each modality $\varOmega_m$ such that the relative proximities of the projections of the relatively-paired observations are preserved in Z, i.e.,

$$d(i, j) \leq d(i, k) \quad \forall (i, j, k) \in \mathcal{T}, \tag{3}$$

where

$$d(i, j) = \|\mathbf{W}_{\varOmega_{t_i}}\mathbf{x}_i - \mathbf{W}_{\varOmega_{t_j}}\mathbf{x}_j\|^2 \tag{4}$$

denotes the squared Euclidean distance between the projections of $\mathbf{x}_i$ and $\mathbf{x}_j$ in Z. Let $\mathbf{W} = [\mathbf{W}_1 \ \ldots \mathbf{W}_M]$ and $\mathbf{S}_{\varOmega_m}$ be a $(\sum d_n) \times d_m$ matrix with all elements being zero except for row $(\sum_{n<m} d_n) + 1$ to row $\sum_{n \le m} d_n$ being an identity matrix, such that $\mathbf{W}_{\Omega_m} = \mathbf{W}\mathbf{S}_{\Omega_m}$. Substituting this into (4) gives

$$\begin{aligned} d(i,j) &= (\mathbf{S}_{\Omega_{t_i}}\mathbf{x}_i - \mathbf{S}_{\Omega_{t_j}}\mathbf{x}_j)^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\mathbf{W}(\mathbf{S}_{\Omega_{t_i}}\mathbf{x}_i - \mathbf{S}_{\Omega_{t_j}}\mathbf{x}_j) \\ &= \mathrm{Tr}(\mathbf{A}\mathbf{C}_{i,j}), \end{aligned} \qquad (5)$$

where $\mathbf{A} = \mathbf{W}^{\mathrm{T}}\mathbf{W}$ and

$$\mathbf{C}_{i,j} = (\mathbf{S}_{\Omega_{t_i}}\mathbf{x}_i - \mathbf{S}_{\Omega_{t_j}}\mathbf{x}_j)(\mathbf{S}_{\Omega_{t_i}}\mathbf{x}_i - \mathbf{S}_{\Omega_{t_j}}\mathbf{x}_j)^{\mathrm{T}}. \qquad (6)$$

Substituting (5) into (3) gives

$$\mathrm{Tr}(\mathbf{A}\mathbf{C}_{i,k}) - \mathrm{Tr}(\mathbf{A}\mathbf{C}_{i,j}) \ge 0 \quad \forall (i,j,k) \in \mathcal{T}. \qquad (7)$$

(7) defines the relative proximity constraints on $\mathbf{A}$ which encodes $\mathbf{W}$ (i.e., the set of projection matrices). Since t_i, t_j, $t_k \in \{1, \ldots, M\}$, there are M^3 possible modality configurations for a triplet (i,j,k). When $t_i = t_j = t_k$, $\mathbf{x}_i$, $\mathbf{x}_j$ and $\mathbf{x}_k$ are from the same modality, and (7) provides constraints in one modality which is the same as metric learning. Now to learn the latent common space, we find a positive-semidefinite matrix $\mathbf{A}$ (i.e., $\mathbf{A} \succeq 0$.) which fulfills (7). Note that if $\mathbf{A}^*$ is a solution, multiplying $\mathbf{A}^*$ by any arbitrary positive scalar will also give a solution. To specify a unique solution, we let $\mathbf{C}_{i,j,k} = \mathbf{C}_{i,k} - \mathbf{C}_{i,j}$ and optimize an SVM style energy function, given by

$$\begin{aligned} \min \quad & \frac{1}{2}\|\mathbf{A}\|_{\mathrm{F}}^2 + \gamma_1 \sum \xi_{i,j,k} + \gamma_2 \sum \mathrm{Tr}(\mathbf{A}\mathbf{C}_{\bar{i},\bar{j}}) \\ s.t. \quad & \mathrm{Tr}(\mathbf{A}\mathbf{C}_{i,j,k}) \ge 1 - \xi_{i,j,k}, \quad \mathbf{A} \succeq 0 \ \text{ and} \\ & \quad \xi_{i,j,k} \ge 0, \quad \forall (i,j,k) \in \mathcal{T}, \quad \forall (\bar{i},\bar{j}) \in \mathcal{P}, \end{aligned} \qquad (8)$$

where both γ_1 and γ_2 are non-negative weights, and $\mathcal{P}$ is a set of pairs $(\bar{i},\bar{j})$ which indicates $\mathbf{x}_{\bar{j}}$ is a target neighbor of $\mathbf{x}_{\bar{i}}$. We will discuss how to set γ_1, γ_2, $\mathcal{T}$ and $\mathcal{P}$ in Sect. 5.2. The first term in (8) is a regularization term which controls the complexity of the model we learn. The second term is the standard hinge loss term which gives a penalty for any violated constraint defined in (7). Minimizing the hinge loss term is equivalent to maximizing a distance margin, which makes the learned model robust against noise. The third term encourages the Euclidean distance between the projections of $\mathbf{x}_{\bar{i}}$ and $\mathbf{x}_{\bar{j}}$ in the latent common space (i.e., $d(\bar{i},\bar{j})$) to be as short as possible. The effects of optimizing (8) is illustrated in Fig. 2.

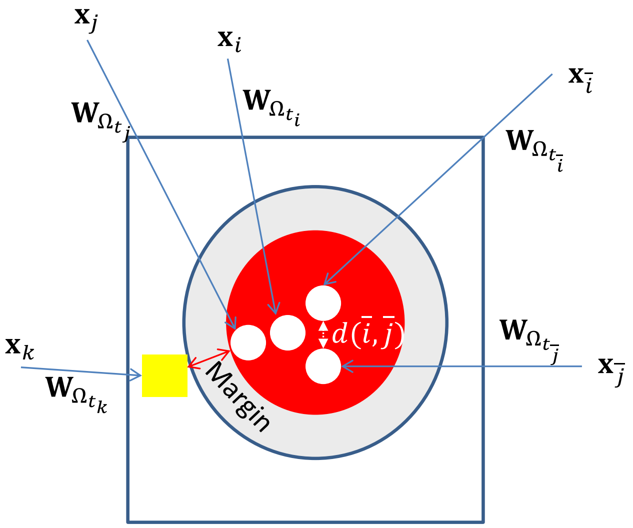
$\mathbf{x}_j$
$\mathbf{x}_i$
$\mathbf{x}_{\bar{i}}$
$\mathbf{W}_{\Omega_{t_j}}$
$\mathbf{W}_{\Omega_{t_i}}$
$\mathbf{W}_{\Omega_{t_{\bar{i}}}}$
$d(\bar{i},\bar{j})$
$\mathbf{W}_{\Omega_{t_{\bar{j}}}}$
$\mathbf{x}_k$
$\mathbf{x}_{\bar{j}}$
$\mathbf{W}_{\Omega_{t_k}}$
Margin

Fig. 2 Illustration of the effects of optimizing (8). Given $\mathbf{x}_h$ from different modalities, where $h \in \{i, j, k, \bar{i}, \bar{j}\}$ with $(i,j,k) \in \mathcal{T}$ and $(\bar{i},\bar{j}) \in \mathcal{P}$, we attempt to learn projection matrices $\mathbf{W}_{\Omega_{t_h}}$ so that a distance margin $d(i,k) - d(i,j)$ is maximized while the distance between points in target neighborhood $d(\bar{i},\bar{j})$ is minimized

3.3 Optimization

We consider the Lagrangian of (8):

$$\begin{aligned} \mathcal{L}(\mathbf{A}, \xi, \mathbf{X}, \mathbf{u}, \mathbf{p}) &= \frac{1}{2}\|\mathbf{A}\|_{\mathrm{F}}^2 + \gamma_1 \sum \xi_{i,j,k} \\ &+ \gamma_2 \sum \mathrm{Tr}(\mathbf{A}\mathbf{C}_{\bar{i},\bar{j}}) - \sum u_{i,j,k}\mathrm{Tr}(\mathbf{A}\mathbf{C}_{i,j,k}) \\ &+ \sum u_{i,j,k} - \sum u_{i,j,k}\xi_{i,j,k} - \mathbf{p}^{\mathrm{T}}\xi - \mathrm{Tr}(\mathbf{A}\mathbf{X}) \\ & s.t. \ \ \mathbf{X} \succeq 0, \ \text{ and } \ u_{i,j,k} \ge 0 \ \text{ and } \ p_{i,j,k} \ge 0, \\ & \forall (i,j,k) \in \mathcal{T}, \forall (\bar{i},\bar{j}) \in \mathcal{P}, \end{aligned} \qquad (9)$$

where $\mathbf{X}$, $u_{i,j,k}$ and $\mathbf{p}$ are the Lagrangian multipliers for the primal variable $\mathbf{A}$, the constraint corresponding to the training triplet (i,j,k) in (8), and ξ respectively. Setting the gradient of (9) with respect to the primal variables $\mathbf{A}$ and ξ to $\mathbf{0}$ gives

$$\mathbf{A}^* = \mathbf{X}^* + \sum u^*_{i,j,k}\mathbf{C}_{i,j,k} - \gamma_2 \sum \mathbf{C}_{\bar{i},\bar{j}}, \qquad (10)$$

and $u^*_{i,j,k} = \gamma_1 - p_{i,j,k}$. Substituting the above expressions into the Lagrangian (9) gives the negative of the dual problem:

$$\begin{aligned} \min \quad & \frac{1}{2}\left\|\mathbf{X} - \hat{\mathbf{C}}\right\|_{\mathrm{F}}^2 - \sum u_{i,j,k} \\ s.t. \quad & \mathbf{X} \succeq 0 \ \text{ and } \ \gamma_1 \ge u_{i,j,k} \ge 0, \\ & \quad \forall (i,j,k) \in \mathcal{T}, \end{aligned} \qquad (11)$$

where $\hat{\mathbf{C}} = -\sum u_{i,j,k}\mathbf{C}_{i,j,k} + \mathbf{B}$ with the matrix $\mathbf{B}$ being $\gamma_2 \sum \mathbf{C}_{\bar{i},\bar{j}}$.

(11) has two variables, namely $\mathbf{X}$ and $\mathbf{u}$. It is optimized by alternating variable method, where one variable is optimized while another is fixed at one time. $\mathbf{X}$ is first optimized while

Algorithm 1 Algorithm of RPSA

```
1: Input: $\{\mathbf{x}_i\}$, $\{t_i\}$, $\mathcal{T}$, $\mathcal{P}$, $\lambda_1$ and $\lambda_2$
2: Output: $\mathbf{A}^*$
3: Initialize $\mathbf{u}$;
4: while not converge do
5:     Compute $\hat{\mathbf{C}}$ according to the current $\mathbf{u}$;
6:     Compute $(\hat{\mathbf{C}})_+$ and $(\hat{\mathbf{C}})_-$ by performing the eigenvalue decomposition;
7:     Compute the first derivative of (13) by (14);
8:     Compute the objective value of (11) by $\frac{1}{2}\mathrm{Tr}((\hat{\mathbf{C}})_-(\hat{\mathbf{C}})_-) - \sum u_{i,j,k}$;
9:     Update $\mathbf{u}$ and its approximated Hessian;
10: end while
11: Let $\mathbf{A}^* = (\hat{\mathbf{C}})_+ - \hat{\mathbf{C}} = -(\hat{\mathbf{C}})_-$;
```

fixing $\mathbf{u}$, and $\mathbf{u}$ is then optimized while fixing $\mathbf{X}$ in each iteration. Specifically, while fixing $\mathbf{u}$, (11) fortunately has a close-form optimal solution:

$$\mathbf{X}^* = (\hat{\mathbf{C}})_+. \tag{12}$$

Fixing $\mathbf{X}$ leads to the following box constraints quadratic programming (QP) over $\mathbf{u}$:

$$\begin{aligned} \min \quad & \frac{1}{2}\left\|\mathbf{X}^* - \mathbf{B} + \sum u_{i,j,k}\mathbf{C}_{i,j,k}\right\|_{\mathrm{F}}^2 - \sum u_{i,j,k} \\ s.t. \quad & \gamma_1 \geq u_{i,j,k} \geq 0, \quad \forall (i,j,k) \in \mathcal{T}. \end{aligned} \tag{13}$$

The off-the-shelf first order Newton algorithm L-BFGS-B (Liu and Nocedal 1989) is employed to solve this QP problem. L-BFGS-B is an iterative algorithm, in each iteration of which, $\mathbf{u}$ is updated till the algorithm converges.

In normal alternating variable method framework, one variable (e.g., $\mathbf{X}$) is updated after another variable (e.g., $\mathbf{u}$) stop changing. For fast convergence, $\mathbf{X}$ is updated once $\mathbf{u}$ is changed in each iteration of L-BFGS-B. Therefore, the gradient of (13) is given by

$$\begin{aligned} G(u_{i,j,k}) &= \mathrm{Tr}((\mathbf{X} - \mathbf{B} + \sum u_{i,j,k}\mathbf{C}_{i,j,k})\mathbf{C}_{i,j,k}) - 1 \\ &= \mathrm{Tr}(-(\hat{\mathbf{C}})_-\mathbf{C}_{i,j,k}) - 1. \end{aligned} \tag{14}$$

The overall optimization procedure is summarized in Algorithm 1. Its main body is the off-the-shelf algorithm L-BFGS-B. The code for computing objective value and gradient, and updating $\mathbf{X}$ (Line 5–8) is implemented by callback functions. The code for updating $\mathbf{u}$ and its approximated Hessian (Line 9) is provided internally in L-BFGS-B. Therefore, there is no need to implement it.

After getting the optimum $\mathbf{A}^*$, we obtain $\mathbf{W}$ by minimizing $\left\|\mathbf{A}^* - \mathbf{W}^{\mathrm{T}}\mathbf{W}\right\|_{\mathrm{F}}$. Suppose the rows of $\mathbf{W}$ are orthogonal to each other, $\mathbf{W}^{\mathrm{T}}\mathbf{W}$ will then be a positive-semidefinite matrix with rank d_z (i.e., the dimension of the latent common space Z). According to Eckart–Young theorem (Stewart 1993), $\mathbf{W}^{\mathrm{T}}\mathbf{W}$ will be the rank-d_z approximation of $\mathbf{A}^*$. We perform eigenvalue decomposition over the positive-semidefinite matrix $\mathbf{A}^*$, getting $\mathbf{A}^* = \mathbf{U}\Lambda\mathbf{U}^{\mathrm{T}}$ with $\mathbf{U}$ being an orthogonal matrix and Λ a real diagonal matrix with decreasing singular values $\sigma_1 \geq \cdots \geq \sigma_{\sum d_m}$. We obtain $\mathbf{W} = \Lambda'\mathbf{U}^{\mathrm{T}}$ with Λ' being a diagonal matrix with decreasing diagonal values $\sqrt{\sigma_1}, \sqrt{\sigma_2}, \ldots, \sqrt{\sigma_{d_z}}, 0, \ldots, 0$. Linear projections $\mathbf{W}_{\Omega_m}$ for different dimensions of Z can be obtained after optimizing (8) and one eigenvalue decomposition. Note that the appropriate latent common space dimension d_z is application dependent, and is determined by cross validation in this paper.

3.4 Time Complexity

In this section, we discuss the time complexity of Algorithm 1. In each iteration, the time complexity for computing $\hat{\mathbf{C}}$ is $\mathcal{O}(Ks^2)$ where K is the number of training triplets (i.e., $|\mathcal{T}|$) and s is the averaged sparsity of $\mathbf{S}_{\Omega_{t_i}}\mathbf{x}_i - \mathbf{S}_{\Omega_{t_j}}\mathbf{x}_j$ or that of $\mathbf{S}_{\Omega_{t_i}}\mathbf{x}_i - \mathbf{S}_{\Omega_{t_k}}\mathbf{x}_k$, whichever is bigger (Line 5). The eigenvalue decomposition of $\hat{\mathbf{C}}$ has $\mathcal{O}(D^3)$ time complexity with $D = \sum d_m$ (Line 6). Computing gradient has the time complexity of $\mathcal{O}(Ks^2)$ (Line 7) while $\mathcal{O}(D^2 + K)$ for computing objective values (Line 8). The cost for updating $\mathbf{u}$ and approximation Hessian (Shen et al. 2011) is $\mathcal{O}(rK)$ with r being a constant (Line 9).

If K is not much greater than D, the eigenvalue decomposition of $\hat{\mathbf{C}}$ dominates the computation complexity in each iteration and the optimization algorithm can converge in a small number of iterations. In this case, the overall time complexity is $\mathcal{O}(T_1 D^3)$ with $D = \sum d_m$ and T_1 being the iteration number.

However, in real applications, for learning stable models, one prefers collecting large scale data set so that the distribution of training data can converge the true, underlying data distribution. In general, one has $\mathcal{O}(N^3)$ training triplets for the data set with size N if enumerating all possible combinations. Although one can cut down on the number of training triplets with heuristics as in the work of Rakotomamonjy (2004), the number is still very large. For instance, the number of triplets in our experiments on the Wiki data set in Sect. 5 is as large as 1 million. In this case, K is much greater than D, and the time spent on the eigenvalue decomposition can be ignored. The overall time complexity is $\mathcal{O}(T_1 r_1 K)$ with r_1 being a constant.

To validate the above discussion, an experiment was conducted on the Wiki data to show the relationship between the training time and the number of training triplets (Fig. 3). It has been observed the time of eigenvalue decomposition (Line 6) dominates the total training time when the number of training triplets is small while that of computing gradient (Line 7) dominates when the number of training triplets is large (see Fig. 3a). Figure 3b shows that the training time of Algorithm 1 is a linear function w.r.t. the number of train-

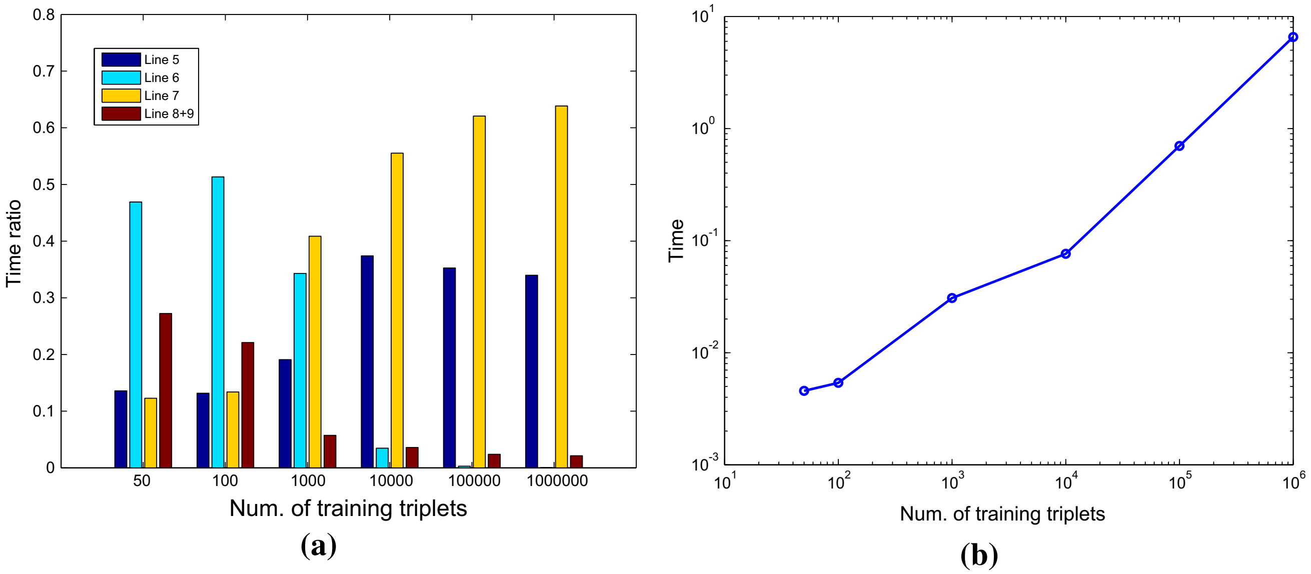

Fig. 3 Training time of RPSA against the number of training triplets. **a** shows the training time ratio of each step in Algorithm 1 as the number of training triplets increases. **b** shows the training time of Algorithm 1 as the number of training triplets increases

ing triplets when the number is large. These observations are consistent with our previous discussion.

4 Efficient Relatively-Paired Space Analysis

As discussed in the previous section, the optimization procedure of (8) slows down dramatically as the number of training triplets increases. The underlying reason is that large scale training triplets would lead to a long vector **u** and thus a large scale QP problem (13). One possible way to speed up the optimization procedure is to reduce the number of training triplets involved. In the literature, Stochastic Gradient Descent (SGD) (Bottou 2010) is usually employed to train models over large scale training samples by randomly selecting a mini-batch of them each time. Although it is successfully used in many applications such as training SVM (Shalev-Shwartz et al. 2007), however, there is no theoretical guarantee that SGD converges to optimal solutions and thus its usefulness heavily dependents on users' parameter tuning experience. Structural learning (Taskar 2004) can also be used to speed up training models by selecting the most violated constraint (training sample) in each iteration. Joachims (2006) reformulated a linear SVM model into a structural SVM model which is solved by the cutting plane algorithm. The reformulation model is proved to be equivalent to the original SVM model. Making things interesting, the structural learning based approach is several orders of magnitude faster than decomposition methods when feature vector is highly sparse. Our model solved in this paper is different from SVM. However, both of them involve optimizations with constraints. It is not clear whether structural learning can speed up our multi-modality analysis model or not.

To this end, we first reformulate the problem of relatively-paired space analysis into learning a structure model. The cutting plane algorithm (Tsochantaridis et al. 2004) is then used to solve this problem, in each iteration of which only a few training triplets are involved. The efficiency of Algorithm 1 and structural learning based approach is finally compared in terms of time complexity and empirical training time.

4.1 Reformulating RPSA into Structural Learning

By introducing a binary variable $c_{i,j,k} \in \{0, 1\}$ for each triplet (i, j, k), the RPSA model can be reformulated into a structural learning problem which can be learned by solving the following optimization problem:

$$\begin{aligned} \min \quad & \frac{1}{2}\|\mathbf{A}\|_{\mathrm{F}}^2 + \gamma_1 K\xi + \gamma_2 \sum \mathrm{Tr}(\mathbf{A}\mathbf{C}_{\bar{i},\bar{j}}) \\ s.t. \quad & \frac{1}{K}\sum c_{i,j,k}\mathrm{Tr}(\mathbf{A}\mathbf{C}_{i,j,k}) \geq \frac{1}{K}\sum c_{i,j,k} - \xi, \\ & \mathbf{A} \succeq 0 \ \text{ and } \ \xi \geq 0, \forall \mathbf{c} \in \{0, 1\}^K. \end{aligned} \tag{15}$$

While (15) has 2^K constraints, one for each possible vector $\mathbf{c} \in \{0, 1\}^K$, it has only one slack variable variable ξ which is shared across all constraints. Interestingly, (15) and (8) are equivalent.

Theorem 1 *Any solution* $\mathbf{A}^*$ *of* (15) *is also the solution of* (8) *(and vice verse) with* $\xi^* = \frac{1}{K}\sum \xi^*_{i,j,k}$.

Proof The following derivation will show for any **A**, (8) and (15) have the same objective value. Given **A**, $\xi_{i,j,k}$ can be optimized individually such that the objective value of (8) is as small as possible. i.e., $\xi^*_{i,j,k} = \max(0, 1 - \mathrm{Tr}(\mathbf{A}\mathbf{C}_{i,j,k}))$. For (15), we have

$$\begin{aligned}
\gamma_1 K \xi^* &= \gamma_1 K \max(0, \max_{\mathbf{c}}(\frac{1}{K} \sum c_{i,j,k} \\
&\quad - \frac{1}{K} \sum c_{i,j,k} \mathrm{Tr}(\mathbf{A}\mathbf{C}_{i,j,k}))) && \text{(16a)} \\
&= \gamma_1 \max(0, \max_{\mathbf{c}}(\sum c_{i,j,k} \\
&\quad - \sum c_{i,j,k} \mathrm{Tr}(\mathbf{A}\mathbf{C}_{i,j,k}))) && \text{(16b)} \\
&= \gamma_1 \max(0, \sum \max_{c_{i,j,k}}(c_{i,j,k} \\
&\quad - c_{i,j,k} \mathrm{Tr}(\mathbf{A}\mathbf{C}_{i,j,k}))) && \text{(16c)} \\
&= \gamma_1 \sum \max(0, 1 - \mathrm{Tr}(\mathbf{A}\mathbf{C}_{i,j,k})) && \text{(16d)} \\
&= \gamma_1 \sum \xi^*_{i,j,k} && \text{(16e)}
\end{aligned}$$

(16a) follows directly from the definition of ξ^*; (16c) holds because each element of $\mathbf{c}$ is independent and can be optimized individually. (16c) and (16d) are equivalent because $c_{i,j,k} \in \{0, 1\}$; again, (16e) follows directly from the definition of $\xi^*_{i,j,k}$. The above equations prove that the objective values of (8) and (15) are the same for any given $\mathbf{A}$. Additionally, (8) and (15) have identical solution space. □

4.2 Optimization

Theorem 1 guarantees that (8) and (15) have the same optima. One may spot that (15) has even more constraints (2^K) than (8) (K), and wonder what one may benefit from this kind of reformulation. Note that there is only one slack variable which is an upper bound for the penalty of all possible constraints in (15). It suggests that only a small subset of constraints are informative. Ignoring other non-informative constraints would lead to a simple reduced problem which can be efficiently solved as only a few constraints are involved in its primal problem or a few dual variables in its dual problem. Joachims (2006) employed the cutting-plane algorithm to find a small set of most violated constraints to speed up the training procedure of linear support vector machine. It has been proved that one can approximate the SVM problem by a reduced problem with a small constant number of constraints. Surprisingly, the number of constraints in the reduced problem is independent of that in the original SVM.

The cutting-plane algorithm is employed to solve (15). In each iteration, the most-violated constraint is first found:

$$\mathbf{c}^* = \arg\max_{\mathbf{c}} \sum c_{i,j,k} - \sum c_{i,j,k} \mathrm{Tr}(\mathbf{A}\mathbf{C}_{i,j,k}), \tag{17}$$

and put it into the most-violated constraint set Γ. i.e., $\Gamma = \Gamma \cup \mathbf{c}^*$. $c_{i,j,k}$ are independent and thus can be optimized individually. One has $c^*_{i,j,k} = 1$ if $\mathrm{Tr}(\mathbf{A}\mathbf{C}_{i,j,k}) < 1$, otherwise 0. $\mathbf{A}$ is then updated by optimizing the following reduced problem:

Algorithm 2 Algorithm of efficient relatively-paired space analysis

```
1: Input: $\{\mathbf{x}_i\}$, $\{t_i\}$, $\mathcal{T}$, $\mathcal{P}$, $\lambda_1$, and $\lambda_2$
2: Output: $\mathbf{A}^*$
3: Initialize $\mathbf{A} = \mathbf{I}$, $\Gamma = \emptyset$;
4: while not converge do
5:   Compute the most violated constraint $\mathbf{c}^*$ by (17), $\Gamma = \Gamma \bigcup \mathbf{c}^*$;
6:   while not converge do
7:     Update $\overline{\mathbf{u}}$ by (21);
8:     Update $\overline{\mathbf{X}}$ by $\overline{\mathbf{X}} = (\hat{\overline{\mathbf{C}}})_+$, and $\mathbf{A}$ by $\mathbf{A} = -(\hat{\overline{\mathbf{C}}})_-$;
9:   end while
10: end while
11: Let $\mathbf{A}^* = \mathbf{A}$;
```

$$\begin{aligned}
\min \quad & \frac{1}{2} \|\mathbf{A}\|_F^2 + \gamma_1 K \xi + \gamma_2 \sum \mathrm{Tr}(\mathbf{A}\mathbf{C}_{i,j}) \\
s.t. \quad & \frac{1}{K} \sum c_{i,j,k} \mathrm{Tr}(\mathbf{A}\mathbf{C}_{i,j,k}) \geq \frac{1}{K} \sum c_{i,j,k} - \xi, \\
& \mathbf{A} \succeq 0 \quad \text{and} \quad \xi \geq 0, \forall \mathbf{c} \in \Gamma.
\end{aligned} \tag{18}$$

Obviously, the number of constraints is $|\Gamma|$, which usually is a small number. Let $\overline{\mathbf{C}}_{\mathbf{c}} = \frac{1}{K} \sum c_{i,j,k} \mathbf{C}_{i,j,k}$ and $\overline{w}_{\mathbf{c}} = \frac{1}{K} \sum c_{i,j,k}$. The negative dual problem of (18) is given by:

$$\begin{aligned}
\min \quad & \frac{1}{2} \left\| \overline{\mathbf{X}} - \hat{\overline{\mathbf{C}}} \right\|_F^2 - \sum \overline{w}_{\mathbf{c}} \overline{u}_{\mathbf{c}} \\
s.t. \quad & \sum \overline{u}_{\mathbf{c}} \leq K \gamma_1, \quad \text{and} \quad \overline{u}_{\mathbf{c}} \geq 0, \quad \forall \mathbf{c} \in \Gamma,
\end{aligned} \tag{19}$$

where $\overline{\mathbf{X}}$ and $\overline{u}_{\mathbf{c}}$ are the Lagrangian multiplier of $\mathbf{A}$ and the constraint corresponding to $\mathbf{c}$ respectively. $\hat{\overline{\mathbf{C}}} = -\sum \overline{u}_{\mathbf{c}} \overline{\mathbf{C}}_{\mathbf{c}} + \mathbf{B}$. Similar to (11), (19) also has two variables and can be optimized by alternating variable method. Again, $\overline{\mathbf{X}}$ has a close-form optimal solution while fixing $\overline{\mathbf{u}}$. i.e.,

$$\overline{\mathbf{X}}^* = (\hat{\overline{\mathbf{C}}})_+, \tag{20}$$

Fixing $\overline{\mathbf{X}}$ and optimizing $\overline{u}_{\mathbf{c}}$ gives a quadratic programming (QP) with a sum constraint:

$$\begin{aligned}
\min \quad & \frac{1}{2} \left\| \overline{\mathbf{X}}^* - \mathbf{B} + \sum \overline{u}_{\mathbf{c}} \overline{\mathbf{C}}_{\mathbf{c}} \right\|_F^2 - \sum \overline{w}_{\mathbf{c}} \overline{u}_{\mathbf{c}} \\
s.t. \quad & \overline{\mathbf{X}} \succeq 0 \quad \text{and} \quad \sum \overline{u}_{\mathbf{c}} \leq K \gamma_1, \quad \overline{u}_{\mathbf{c}} \geq 0, \quad \forall \mathbf{c} \in \Gamma.
\end{aligned} \tag{21}$$

Since L-BFGS-B cannot solve QP problems with a sum constraint, the quadprog function with interior points option[1] in Matlab is employed to optimize it efficiently.

The optimization procedure of (15) is summarized in Algorithm 2. In Line 3, $\mathbf{A}$ is initialized to an identity matrix $\mathbf{I}$. In Line 7, we initialize the problem (21) using previous $\overline{\mathbf{u}}$ to speed up convergence.

[1] The number of variables is very small.

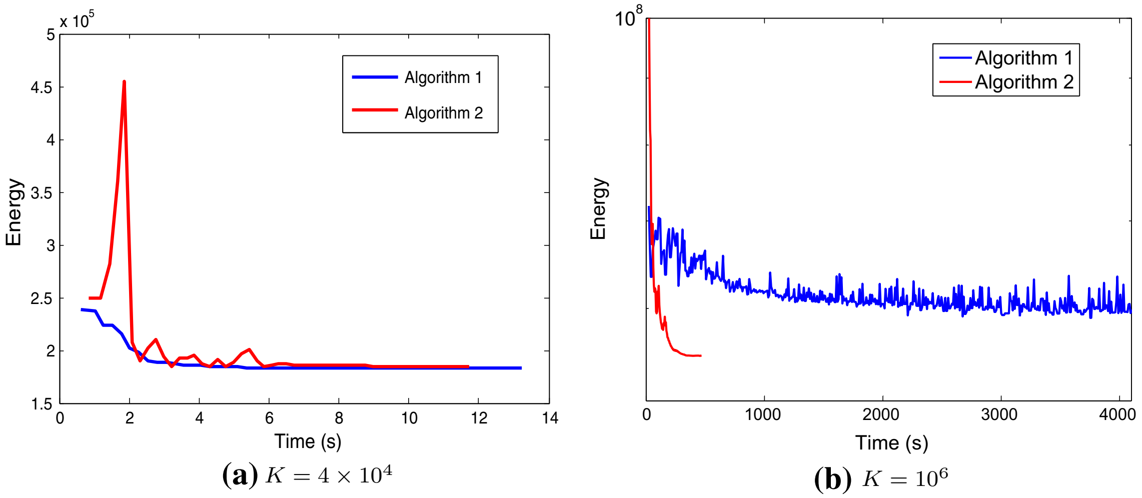

Fig. 4 Efficiency comparison between Algorithm 1 and 2 on the Wiki Text-Image data set with different numbers of training triplets

4.3 Time Complexity

In each outer iteration of Algorithm 2, the time complexity of computing the most violated constraints (Line 5) is $\mathcal{O}(Ks^2)$. Line 6–9 solve the reduced problem (18) which is actually an RPSA problem and can be optimized by Algorithm 1. Since it has very limited constraints (i.e., $|\Gamma|$ is small and thus $\bar{\mathbf{u}}$ is a short vector), it can be solved with time complexity $\mathcal{O}(T_1 D^3)$ as discussed in Sect. 3.4. For large scale training triplets, the time of computing the most violated constraints dominates the total time of each outer iteration while that of solving the reduced problem is negligible. Therefore, the time complexity of Algorithm 2 is $\mathcal{O}(T_2 K s^2)$ with the constant T_2 being the outer iteration number.

Both Algorithm 1 and 2 have a linear time complexity for each outer iteration when a huge number of training triplets are available. However, Algorithm 2 empirically converges much faster than Algorithm 1. Theoretically, its outer iteration number T_2 does not depend on the number of training examples K (Joachims et al. 2009).

4.4 Efficiency Comparison

Algorithm 1 and 2 both converge to an identical solution which is guaranteed by Theorem 1. One concerns only their efficiency. We conducted two experiments on the Wiki text-image data set to compare them. The number of training triplets is set to 4×10^4 in the first experiment while 10^6 in the second one (other settings can be found in Sect. 5.5).

Figure 4 plots the energy of (8) against training time. Figure 4a shows that Algorithm 1 converges faster than Algorithm 2 when the number of triplets (K) is small. This is reasonable since Algorithm 1 invokes less eigenvalue decomposition which dominates computation time in this case than Algorithm 2. Moreover, Algorithm 1 has more stable energy decreasing procedure. The underlying reason is that Algorithm 2 finds a most violated constraint in each of its outer iterations. Figure 4b shows that Algorithm 2 is typically several orders of magnitude faster than Algorithm 1 when the number of triplets (K) is huge.

4.5 Discussion

In analogy to previous work (Tsochantaridis et al. 2004; Joachims 2006), Algorithm 2 also uses a 1-slack energy function. However, there are two significant differences. First, Algorithm 2 is a reformulation of a semi-definite programming problem while Tsochantaridis et al. (2004)'s work is a general framework for structural learning and Joachims (2006)'s work is a reformulation of a linear SVM. Second, Algorithm 2 has very different property from that of Joachims (2006)'s work. Algorithm 2's reformulation has advantage when K is huge compared with the original formulation while Joachims (2006)'s reformulation has advantage when feature vectors are highly sparse. From above discussion, our main technical contribution is to seamlessly integrate semi-definite programing with the cutting plane algorithm. Another technical contribution is a detailed analysis of time complexity of Algorithm 1 and 2 and their empirical comparison.

5 Experiments

The performance of our proposed RPSA framework was evaluated by applying it to feature fusion, cross-pose face recognition, text-image retrieval and attribute-image retrieval.

Table 1 Four types of triplets defined for describing relative-pairing information of a given pair of observations $(\mathbf{x}_p, \mathbf{x}_q)$

Type	Form	Num.	Remark
1	(p, q, q_1)	n_1	$\mathbf{x}_{q_1}$ is the kth $(k \leq n_1)$ nearest neighbor of $\mathbf{x}_q$ s.t. $t_p \neq t_q \wedge t_q = t_{q_1} \wedge l_p = l_q \wedge l_q \neq l_{q_1}$
2	(q, p, p_1)	n_2	$\mathbf{x}_{p_1}$ is the kth $(k \leq n_2)$ nearest neighbor of $\mathbf{x}_p$ s.t. $t_q \neq t_p \wedge t_p = t_{p_1} \wedge l_q = l_p \wedge l_p \neq l_{p_1}$
3	(p, p_1, p_2)	n_3	$\mathbf{x}_{p_1}$ is the kth $(k \leq n_3)$ nearest neighbor of $\mathbf{x}_p$ s.t. $t_p = t_{p_1} \wedge l_p = l_{p_1}$ $\mathbf{x}_{p_2}$ is the kth $(k \leq n_3)$ nearest neighbor of $\mathbf{x}_p$ s.t. $t_p = t_{p_2} \wedge l_p \neq l_{p_2}$
4	(q, q_1, q_2)	n_4	$\mathbf{x}_{q_1}$ is the kth $(k \leq n_4)$ nearest neighbor of $\mathbf{x}_q$ s.t. $t_q = t_{q_1} \wedge l_q = l_{q_1}$ $\mathbf{x}_{q_2}$ is the kth $(k \leq n_4)$ nearest neighbor of $\mathbf{x}_q$ s.t. $t_q = t_{q_2} \wedge l_q \neq l_{q_2}$

5.1 Training Triplets and Pairs

Training triplets $(i, j, k) \in \mathcal{T}$ can be generated in an unsupervised or supervised fashion. Relatively-paired data can be collected from clickthrough data of search engines or priori knowledge about relative-pairing. This kind of data is naturally gotten. It can also be generated from category labels based on the principle that observations with the same label are expected to be more-likely-paired than those with different labels. Let l_i denote the label of an observation $\mathbf{x}_i$. Given a pair of cross-modality observations $(\mathbf{x}_p, \mathbf{x}_q)$ (where $t_p \neq t_q$) for an object, we define four types of triplets to describe the relative-pairing knowledge (see Table 1). Each triplet (i, j, k) suggests that $\mathbf{x}_i$ is more-likely-paired with $\mathbf{x}_j$ than with $\mathbf{x}_k$. Euclidean distance between two observations is used in defining nearest neighbor in Table 1. Figure 5 gives a graphical illustration for these four types of triplets. If the numbers of these four types of triplets are n_1, n_2, n_3 and n_4, respectively, for each given pair $(\mathbf{x}_p, \mathbf{x}_q)$, we say that the training triplets have a structure of (n_1, n_2, n_3, n_4). The total number of triplets is therefore $(n_1 + n_2 + n_3 + n_4) \times N_p$, where N_p is the number of pairs.

Similarly, we generate training pair set $\mathcal{P}$ from labels. Given a pair of cross-modality observations $(\mathbf{x}_p, \mathbf{x}_q)$ (where $t_p \neq t_q$) for an object, three types of pairs are defined to describe target neighborhood (see Table 2). Each pair $(\bar{i}, \bar{j})$ suggests that $\mathbf{x}_{\bar{i}}$ is a target neighbor of $\mathbf{x}_{\bar{j}}$ and vise versa. Again, if the number of these three types of pairs are $\overline{n}_1$, $\overline{n}_2$ and $\overline{n}_3$, respectively, for each given pair $(\mathbf{x}_p, \mathbf{x}_q)$, we say that the training pairs have a structure of $(\overline{n}_1, \overline{n}_2, \overline{n}_3)$. Therefore, we have $(\overline{n}_1 + \overline{n}_2 + \overline{n}_3) \times N_p$ training pairs in total. Note that $\overline{n}_1$ has only two choices 0 or 1.

5.2 Parameter Settings

There are two weights, namely γ_1 and γ_2 in our energy function (8). In our experiments, we found that γ_1 is not sensitive to other settings, such as the number of training triplets. The underlying reason is that the hinge loss term only penalize violated constraints in (7) no matter how many training triplets we have. We therefore fixed it to 1 in all our experiments. Because γ_2 is the weight of the sum of distances between points in neighborhood, it is affected by the scale of feature vectors. Hence, we individually tuned γ_2 for different data sets using validation data. Detailed analyses can be found in each corresponding sections.

Theoretically, the more training triplets we use, the more constraint information and better performance we get. This has been confirmed in our experimental results (see Fig. 6). Therefore, we used as many as possible triplets in our experiments. Because the numbers of training data of each category in different data sets are different, we have different training triplets in different tasks. Detailed triplet structures for different tasks can be found in their corresponding sections.

The parameters regarding training pairs $\mathcal{P}$ are insensitive to other settings. If one expects the distance between the projections of $\mathbf{x}_p$ and $\mathbf{x}_q$ to be short, then $\overline{n}_1$ should be set to 1, otherwise 0. The second and third kind of training pairs encourage small distances between projections of observations with the same label in each modality. We found that

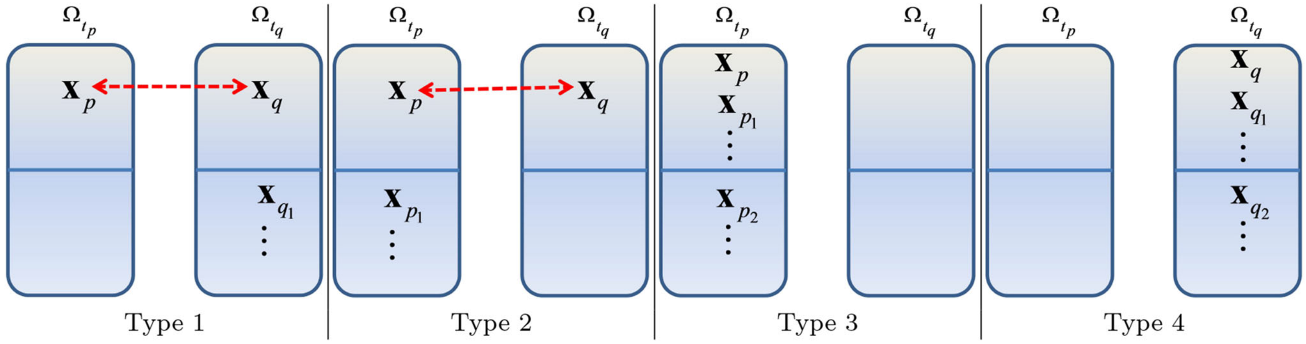

Fig. 5 Four types of triplets defined for describing relative-pairing information of a given pair of observations $(\mathbf{x}_p, \mathbf{x}_q)$. $\mathbf{x}_p \leftarrow\text{- - - -}\rightarrow \mathbf{x}_q$ means $\mathbf{x}_p$ and $\mathbf{x}_q$ are paired observations from different modalities. Grids on the same *horizontal line* contain cross-modality observations with the same label

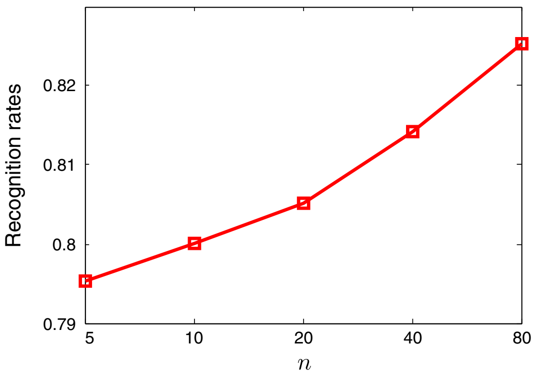

Fig. 6 Performance of RPSA on validation data with different numbers of training triplets. RPSA is used to fuse the feature pair (Zer, Mor). n_1, n_2, n_3 and n_4 are set to n while other parameters are tuned to maximize the performance for each specific n

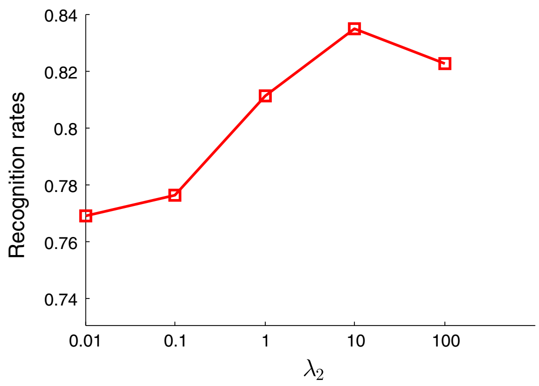

Fig. 7 Performance of RPSA on validation data when fusing the modality pair (Zer Mor) with different γ_2

Table 2 Three types of pairs defined for describing target neighborhood of a given pair of observations $(\mathbf{x}_p, \mathbf{x}_q)$

Type	Form	Num.	Remark
1	(p, q)	$\overline{n}_1$	$\overline{n}_1 = 0$ or 1
2	(p, p_1)	$\overline{n}_2$	$\mathbf{x}_{p_1}$ is the kth $(k \leq \overline{n}_2)$ nearest neighbor of $\mathbf{x}_p$ s.t. $t_p = t_{p_1} \wedge l_p = l_{p_1}$
3	(q, q_1)	$\overline{n}_3$	$\mathbf{x}_{q_1}$ is the kth $(k \leq \overline{n}_3)$ nearest neighbor of $\mathbf{x}_q$ s.t. $t_q = t_{q_1} \wedge l_q = l_{q_1}$

small target neighborhood (i.e., $\overline{n}_2$ and $\overline{n}_3$ is set to a small number, e.g., 5) works well in our experiments. For feature fusion, diversity of projections of exactly-paired observations from different modalities and small distances between projections of observations with the same label in target neighborhood in each modality are desirable. Therefore, we set $\overline{n}_1 = 0$ and $\overline{n}_2 = \overline{n}_3 = 5$. For cross-modality pattern recognition tasks, namely, cross-pose face recognition, text-image retrieval and attribute-image retrieval, similar projections of exactly-paired observations are desirable, and thus we set $\overline{n}_1 = 1$ and $\overline{n}_2 = \overline{n}_3 = 0$.

To summarize, only γ_2 should be tuned for each data set while other parameters are fixed in advance.

5.3 Feature Fusion

For classifying patterns with different kinds of features stemming from different sources, a critical issue is to efficiently utilize these cross-modality features. A common solution is feature fusion by first projecting cross-modality features into a latent common space to reduce dimension and suppress noise, and then adding the paired projections together as a final feature vector. The fused feature for two modalities (Sun et al. 2008; Zhang and Zhang 2011) is usually given by

$$\mathbf{y} = \mathbf{W}_{\Omega_{t_i}} \mathbf{x}_i + \mathbf{W}_{\Omega_{t_j}} \mathbf{x}_j, \tag{22}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are two feature vectors for different modalities of an object (i.e., $t_i \neq t_j$). The proposed method was used to fuse features of UCI Multiple Features data set.[2] This data set consists of 2,000 instances of ten hand-written numerals ('0'–'9'). Each instance has six features, namely Fou, Fac, Kar, Pix, Zer and Mor, with dimensions 76, 216, 64, 240, 47, and 6 respectively. We considered each feature as one modality. In our experiment, any two kinds of features were selected to fuse, and we had $C_2^6 = 15$ combination pairs. In the training phase, for each feature pair, the number of training data for each digit (N_t) was set to 100. The latent common space had a dimension of 25, except for feature pairs involving Mor where it had a dimension of 6. In the testing phase, we find the nearest training fused feature with label for each testing fused feature. The experiment was repeated 10 times by randomly selecting fixed number of training data (i.e., $N_t \times 10$, here 10 is the number of digit categories). We evaluated our method by mean recognition rates.

To determine γ_2, we used $\frac{1}{5}$ of training data as validation data and the rest as "training data". The structure of training triplets are fixed to be (79,79,79,79) (given a digit $\mathbf{x}_p$, the number of digits with the same label as $\mathbf{x}_p$ is 79 in the "training data"). The pair of modalities (Zer, Mor) are fused with different γ_2. The recognition rates are shown in Fig. 7. It has been observed that the proposed method gets the best result with $\gamma_2 = 10$. Therefore, we fixed γ_2 to 10 in this experiment. After parameter tuning, RPSA was trained with all training data with fixed parameters.

The proposed method was compared with Canonical Correlation Analysis (CCA) (Hardoon et al. 2004), Discriminative Canonical Correlation Analysis (DCCA) (Sun et al. 2008), Partial Least Squares (PLS) (Prince et al. 2008; Rosipal and Krämer 2006), bagging CCA (bgCCA), bagging

[2] http://archive.ics.uci.edu/ml/datasets/Multiple+Features.

Table 3 Recognition rates on multiple features data set

Pair		Method								
		CCA	DCCA	bgCCA	bgDCCA	bsCCA	bsDCCA	PLS	RCE	RPSA
Fac	Fou	0.86	0.89	0.86	0.89	0.84	0.88	0.94	0.95	**0.98**
Fac	Kar	0.95	0.98	0.95	**0.98**	0.93	**0.98**	0.94	**0.98**	**0.98**
Fac	Pix	0.86	0.97	0.86	0.97	0.86	0.97	0.94	0.95	**0.98**
Fac	Zer	0.85	0.88	0.86	0.88	0.84	0.87	0.96	**0.97**	**0.97**
Fac	Mor	0.73	0.82	0.75	0.82	0.74	0.81	0.88	0.88	**0.97**
Fou	Kar	0.90	0.90	0.90	0.90	0.88	0.89	**0.97**	0.96	**0.97**
Fou	Pix	0.76	0.89	0.77	0.89	0.74	0.87	**0.98**	0.95	**0.98**
Fou	Zer	0.82	0.83	0.82	0.83	0.80	0.81	0.81	0.85	**0.86**
Fou	Mor	0.75	0.77	0.75	0.77	0.74	0.76	0.44	0.80	**0.84**
Kar	Pix	0.94	0.95	0.94	0.95	0.93	0.94	**0.98**	0.96	**0.98**
Kar	Zer	0.90	0.88	0.90	0.88	0.89	0.86	0.83	**0.96**	**0.96**
Kar	Mor	0.75	0.80	0.77	0.80	0.76	0.79	0.62	0.86	**0.97**
Pix	Zer	0.83	0.87	0.83	0.87	0.80	0.86	0.84	0.94	**0.97**
Pix	Mor	0.72	0.79	0.73	0.79	0.71	0.77	0.71	0.84	**0.98**
Zer	Mor	0.68	0.75	0.72	0.75	0.70	0.74	0.72	0.77	**0.84**

The best performance for each experimental settings are in bold

DCCA (bgDCCA), boosting CCA (bsCCA), boosting DCCA (bsDCCA) (Zhang and Zhang 2011) and Random Correlation Ensemble (RCE) (Zhang and Zhang 2011). For fair comparison, all the methods employ nearest neighbor method as the classifier. The results of competitors are from Table 2 in Zhang and Zhang (2011). From Table 3, we see that RPSA is clearly superior to CCA, DCCA, bgCCA, bgDCCA, bsCCA, bsDCCA and PLS. RPSA achieves better accuracy than RCE for 12 pairs, and identical accuracy for the remaining 3 pairs. Note that RCE is a sophisticated method which first finds random cross-view correlations between within-class examples and then boosts performance by ensemble learning.

Our proposed RPSA is a general framework of multimodality analysis. It is natural to extend RPSA to fuse features from three modalities. We conducted feature fusion over all possible three features and compared with the result with that of Multi-View CCA (Rupnik and Shawe-Taylor 2010; Gong et al. 2014). Therefore, we have $C_3^6 = 20$ configurations. Due to space limitation, we only reported the average of the mean recognition rates over 20 configurations. The average of the mean recognition rates of RPSA over three features is 0.97 while that of Multi-View CCA is 0.93. The average of the mean recognition rates of RPSA over two features is 0.95, which suggests that fusing more features can produce better recognition results.

5.4 Cross-Pose Face Recognition

Faces observed under a particular pose can be considered as being sampled from one modality, and therefore faces observed under different poses correspond to different modalities. RPSA can be used to recognize faces under different poses, in which gallery faces are in one pose while probe faces are in another pose. Note that our method requires knowing the rough pose of each photo (i.e., to which modality it belongs) as in the work of Sharma and Jacobs (2011). CMU PIE face database[3] was used in our experiments. This data set consists of 68 subjects, each of which has face photos in 13 different poses (indexed by c27/05/29/37/11/07/09/02/14/22/34 /25/31). Photos in the same pose were aligned by the eyes and mouth. All photos were cropped and down-sampled to 48 × 40. Each photo was then reshaped into a column vector giving an observation $\mathbf{x}_i$. In our experiments, subject 1 to 34 were used as training data, while the rest were used as testing data. In the training phase, we learned one projection matrix for each modality. The learned latent common space had a dimension of 25. In the testing phase, the nearest gallery face of each probe face was found in the learned latent common space, and the recognition rates were reported.

To tune γ_2, we randomly selected $\frac{1}{5}$ of training data as validation data and the rest as "training data". The structure of training triplets are fixed to be (26,26,0,0) (given a face $\mathbf{x}_p$, the number of faces with different labels is 26 while the number of faces with same label as $\mathbf{x}_p$ is 0). RPSA is used to do cross-pose face recognition with c22 as gallery and c07 as query. Figure 8 plots the recognition rates with different γ_2. It has been shown that RPSA is not sensitive to γ_2 as long as

[3] http://vasc.ri.cmu.edu/idb/html/face/.

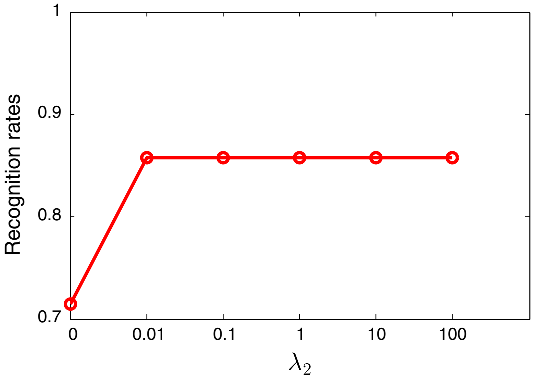

Fig. 8 Performance of RPSA over the validation set with different γ_2

$\gamma_2 > 0.01$. Therefore, we fixed $\gamma_2 = 10$. and retrained our model over all training data.

In Table 4, we compare our method with those using frontal faces (photos indexed by c27 in CMU PIE data set) as gallery, in terms of mean recognition rates over different subsets of probe poses. The subsets of probe poses are set to be the same as those in Sharma and Jacobs (2011). The results of competitors are from Table 3 in Sharma and Jacobs (2011). It can be seen that RPSA outperforms all competitors. Note that TFA requires 14 user-elaborately-clicked points for photo alignment and Gabor filter for extracting complex features, whereas our method only needs 3 points (eyes and mouth) for photo alignment and directly employs the face image as a feature vector.

We also compare our method with PLS (Sharma and Jacobs 2011) and Multi-view Discriminant Analysis (MvDA) (Kan et al. 2012) which, to the best of our knowledge, report the best performance in the recent literature. Two arbitrary poses were selected as a gallery-probe pair, and we therefore had $P_2^{13} = 156$ configurations. The results are shown in Table 5. The result of PLS is from Table 1 in Sharma and Jacobs (2011) and that of MvDA is collected by running its publicly available code.[4] It can be seen that the proposed RPSA is much better than PLS and MvDA. RPSA achieves the best average recognition rates for all different galleries. It gets the best results in 140 configurations and the second best results in 16 configurations. RPSA is slightly worse than MvDA for the configurations (c22,c07) and (c07, c22) (c22 is a side view while c07 is a frontal view). This might be due to big pose difference between c22 and c07.

The overall accuracy of the proposed RPSA for all gallery-probe pairs is 0.957 while those of PLS and MvDA are 0.901 and 0.922 respectively. Our method improves the state-of-the-art result by 3.8 %.

In this experiment, observations from different modalities have the same number of dimension. Therefore, metric learning methods designed for one modality can be used to do cross-pose face recognition without considering modality difference. We evaluated the performance of the state-of-the-art metric learning method Information Theoretic Metric Learning (ITML) on this task. The average accuracies are 0.409, 0.390, 0.547, 0.532, 0.542, 0.471, 0.583, 0.446, 0.449, 0.569, 0.463, 0.529, and 0.392 when c34, c31, c14, c11, c29, c09, c27, c07, c05, c37, c25, c02 and c22 are gallery respectively. Its overall accuracy is only 0.486. It has been observed that multi-modality analysis methods greatly outperform metric learning methods designed for one modality on cross-modality pattern recognition problems.

5.5 Text-Image Retrieval

Text and image are two different modalities. Using text query to retrieve images or image query to retrieve texts are cross-modality problems, which requires common representations. The proposed RPSA was validated by text-image retrieval on Wiki Text-Image data set (Rasiwasia et al. 2010). The data set consists of 2,173 training and 693 testing image-text pairs with 10 different categories. The images are represented by 128-dimensional SIFT feature vectors while texts are encoded by 10-dimensional latent Dirichlet allocation model-based feature vectors (Blei et al. 2003). In the training phase, we learned one projection matrix for each modality. In the testing phase, queries and probes were projected into the learned latent space with the dimension of 10, and then text-image retrieval was conducted by finding the nearest neighbors of the projections of queries. It is considered to be correct if the retrieved image (or text) has the same label as the query text (or image). As in Sharma and Kumar (2012), the precisions of retrieval are evaluated at 11 different recall levels {0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1}; the Mean Average Precision (mAP) given by $\frac{1}{11}\sum_{i=1}^{11} precision_i$ is finally reported.

To tune γ_2, $\frac{1}{8}$ of training data were selected as validation data and the rest as “ training set”. The structure of training triplets were fixed to be (119,119,119,119) (given an image $\mathbf{x}_p$, the number of images with the same label as $\mathbf{x}_p$ is 119). We evaluated RPSA with text as query on validation data. We found that the performance was best when $\gamma_2 = 100$. Therefore, we fixed $\gamma_2 = 100$. RPSA was retrained over all training data with fixed parameters.

In our experiments, we found that overweighting the first type of training triplets (i.e., when t_p is image modality) will greatly boost the performance of RPSA with image as query while the second type training triplets (i.e., when t_p is text modality) with text as query. This might be because the model we learned is too simple (the dimension of the latent common space is too low) to satisfy all training triplet constraints at the same time. The results reported in this Section were obtained by overweighting the first and the second type of training

[4] http://vipl.ict.ac.cn/members/mnkan.

Table 4 Mean recognition rates for frontal faces (c27) gallery

Gallery	Probe	Method	Accuracy	Method	Accuracy
c27	c05/37/25/22/29/11/14/34	PGFR (Liu and Chen 2005)	0.86	RPSA	0.98
c27	c05/22	TFA (Prince et al. 2008)	0.95	RPSA	0.96
c27	c05/29/37/11/07/09	LLR (Chai et al. 2007)	0.95	RPSA	1.00
c27	c05/29/37/11/07/09	ELF (Gross et al. 2004)	0.90	RPSA	1.00

Table 5 Recognition rates for different gallery-probe pose pairs on CMU PIE

Gallery	Probe													
		c34	c31	c14	c11	c29	c09	c27	c07	c05	c37	c25	c02	c22	Avg.
c34	RPSA	–	**0.97**	**1**	**0.94**	**0.94**	**0.91**	**0.94**	0.91	**0.88**	**0.91**	**0.76**	**0.91**	**0.85**	**0.912**
	MvDA	–	0.91	0.97	**0.94**	0.85	0.82	0.85	0.91	0.71	**0.91**	0.62	0.82	0.82	0.846
	PLS	–	0.88	0.94	**0.94**	0.91	0.88	0.91	**0.97**	0.85	0.88	0.70	0.85	0.61	0.862
c31	RPSA	**0.97**	–	**1**	**1**	**1**	0.94	**0.94**	**1**	**0.97**	**1**	**0.91**	**0.97**	**0.91**	**0.968**
	MvDA	0.94	–	**1**	**1**	**1**	**1**	**0.94**	**1**	0.94	**1**	0.79	0.85	0.74	0.934
	PLS	0.85	–	**1**	**1**	**1**	0.88	0.85	0.91	0.85	0.88	0.76	0.85	0.76	0.884
c14	RPSA	0.94	**1**	–	**1**	**1**	**0.97**	**1**	**1**	**0.94**	**1**	**0.91**	**1**	**0.94**	**0.98**
	MvDA	0.94	**1**	–	0.97	**1**	**0.97**	**1**	**1**	0.79	**1**	0.76	0.82	0.79	0.922
	PLS	**0.97**	**1**	–	**1**	0.97	0.91	0.97	**1**	0.91	**1**	0.82	0.91	0.67	0.928
c11	RPSA	**0.97**	**1**	**1**	–	**1**	**0.94**	0.97	**1**	**1**	**1**	**0.85**	**0.97**	**0.82**	**0.961**
	MvDA	0.94	**1**	0.97	–	**1**	0.91	**1**	**1**	0.94	0.97	0.76	**0.97**	0.79	0.939
	PLS	0.79	0.97	**1**	–	**1**	0.88	**1**	**1**	0.97	0.97	**0.85**	0.88	0.67	0.916
c29	RPSA	**0.91**	**0.94**	0.97	**1**	–	**1**	**1**	**1**	**1**	**1**	0.82	**0.97**	**0.91**	**0.961**
	MvDA	0.88	**0.94**	0.97	**1**	–	0.97	**1**	**1**	**1**	0.97	0.76	0.88	0.79	0.931
	PLS	0.76	**0.94**	**1**	**1**	–	**1**	**1**	**1**	**1**	**1**	**0.85**	0.91	0.73	0.933
c09	RPSA	**0.97**	0.97	**0.97**	**1**	**1**	–	**1**	**1**	**1**	**1**	**0.94**	**0.97**	**0.91**	**0.978**
	MvDA	0.88	**1**	**0.97**	**1**	0.97	–	**1**	0.97	**1**	0.97	0.91	0.82	0.85	0.946
	PLS	0.76	0.88	0.91	0.94	0.94	–	0.97	0.94	0.91	0.88	0.82	0.79	0.70	0.872
c27	RPSA	**0.94**	**0.94**	**1**	**1**	**1**	**1**	–	**1**	**1**	**1**	**0.97**	**1**	**0.91**	**0.980**
	MvDA	0.88	**0.94**	0.97	**1**	**1**	0.94	–	**1**	**1**	0.97	0.82	**1**	**0.91**	0.953
	PLS	0.85	0.91	0.97	**1**	**1**	**1**	–	**1**	**1**	**1**	0.85	0.88	0.79	0.939
c07	RPSA	**0.91**	**1**	**1**	**1**	**1**	**1**	**1**	–	**1**	**1**	0.85	**0.97**	0.82	**0.963**
	MvDA	0.85	0.91	0.97	**1**	**1**	0.94	**1**	–	**1**	0.97	**0.94**	**0.97**	**0.88**	0.953
	PLS	0.79	0.91	0.97	**1**	**1**	0.97	**1**	–	**1**	0.97	0.85	0.91	0.76	0.929
c05	RPSA	**0.88**	**1**	**1**	0.97	**1**	**1**	**1**	**1**	–	**1**	**0.97**	**1**	**0.94**	**0.980**
	MvDA	0.85	0.97	0.88	**1**	**1**	**1**	**1**	**1**	–	**1**	0.94	**1**	0.91	0.963
	PLS	0.79	0.97	0.97	0.94	**1**	0.94	**1**	**1**	–	0.97	0.91	0.91	0.82	0.936
c37	RPSA	0.88	**0.94**	**1**	**1**	**1**	**0.97**	**1**	**1**	**1**	–	0.97	**1**	**0.94**	**0.976**
	MvDA	**0.91**	**0.94**	**1**	0.97	**1**	**0.97**	0.94	0.97	0.94	–	0.94	**1**	0.88	0.956
	PLS	0.79	0.94	**1**	0.94	0.94	0.88	0.94	0.94	0.97	–	**1**	**1**	**0.94**	0.941
c25	RPSA	**0.76**	**0.91**	**0.88**	**0.88**	**0.88**	**0.91**	**0.91**	0.88	0.91	**0.97**	–	**0.97**	**0.85**	**0.895**
	MvDA	0.68	0.79	0.85	**0.88**	0.82	**0.91**	0.85	**0.91**	0.91	0.94	–	0.91	0.82	0.858
	PLS	0.67	0.82	0.76	0.79	**0.88**	0.88	0.88	**0.91**	**0.94**	**0.97**	–	**0.97**	0.76	0.855
c02	RPSA	**0.85**	**0.94**	**0.94**	**0.97**	**1**	**0.97**	**1**	**1**	**1**	**1**	**1**	–	**1**	**0.973**
	MvDA	0.74	0.82	0.79	0.94	0.97	0.85	0.97	0.97	**1**	**1**	0.97	–	0.97	0.917
	PLS	0.76	0.88	0.88	0.94	0.94	0.88	0.97	0.94	**1**	**1**	**1**	–	0.97	0.931
c22	RPSA	0.82	**0.94**	**0.94**	**0.88**	**0.91**	**0.94**	**0.91**	0.85	**0.97**	**0.94**	**0.91**	**0.97**	–	**0.918**
	MvDA	**0.85**	0.79	0.85	0.79	0.82	0.85	**0.91**	**0.91**	0.82	0.88	**0.91**	0.94	–	0.863
	PLS	0.64	0.70	0.64	0.79	0.76	0.67	0.82	0.82	0.85	0.91	0.85	0.91	–	0.784

The best performance for each experimental settings are in bold

Table 6 mAP on Wiki Text-Image data set

Query	Method							
	PLS	BLM	CCA	SM	SCM	GMMFA	GMLDA	RPSA
Image	0.207	0.237	0.182	0.225	0.277	0.264	0.272	0.280
Text	0.192	0.144	0.209	0.223	0.226	0.231	0.232	0.249
Avg.	0.199	0.191	0.196	0.224	0.252	0.248	0.253	0.265

triplets by 20 times when image and text are used as queries respectively.

The performance of RPSA is compared with those of the state-of-the-art multi-modality analysis techniques: CCA, PLS, BLM, Semantic Matching (SM) (Rasiwasia et al. 2010), Semantic Correlation Matching (SCM) (Rasiwasia et al. 2010), Generalized Multi-view Marginal Fisher Analysis (GMMFA) (Sharma and Kumar 2012) and Generalized Multi-view LDA (GMLDA) (Sharma and Kumar 2012) in Table 6. The results of competitors are from Table 3 in Sharma and Kumar (2012). It has been shown that the proposed method achieves the best performance in terms of mAP with image query, that with text query, and the average mAP. Specifically, RPSA improves the state of the art result by 4.7 %. Figure 9 shows recall precision curves of RPSA compared with others. It has been shown that RPSA performs better than CCA and PLS with an obvious margin. Figure 10 shows mAP of each category with text query obtained by CCA, PLS and RPSA. It has been shown that RPSA consistently performs better than its competitors for each category except sport.

5.6 Attribute-Image Retrieval

In the previous experiments, all the training triplets are generated with category labels as shown in Table 1. In this Section, we would like to evaluate PRSA using natural relative-pairing information. The data set we used is Public Figures Face Database (Kumar et al. 2009). In Parikh and Grauman (2011), a subset consisting of 241 training faces and 531 test faces are selected to study relative attributes. These faces belong to 8 persons. They have 11 attributes, namely, “Male”, “White”, “Yong”, “Smiling”, “Chubby”, “Visible Forehead”, “Bush Eyebrows”, “Narrow Eyes”, “Pointy Nose”, “Big Lips” and “Round Face”. Each face is encoded by a 542-d feature vector based on Gist and color histogram extracted from its image. It is also encoded by 11-d binary attribute vector. e.g., the

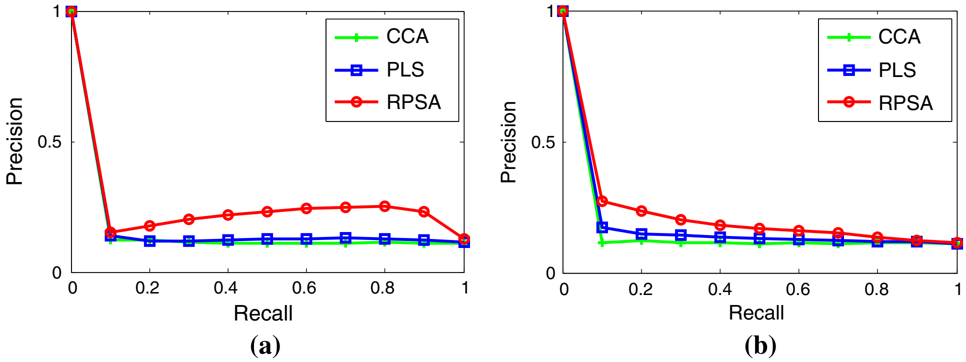

Fig. 9 Comparison between recall precision curves of CCA, PLS, and RPSA. **a** shows recall precision curves with image query. **b** shows recall precision curves with text query

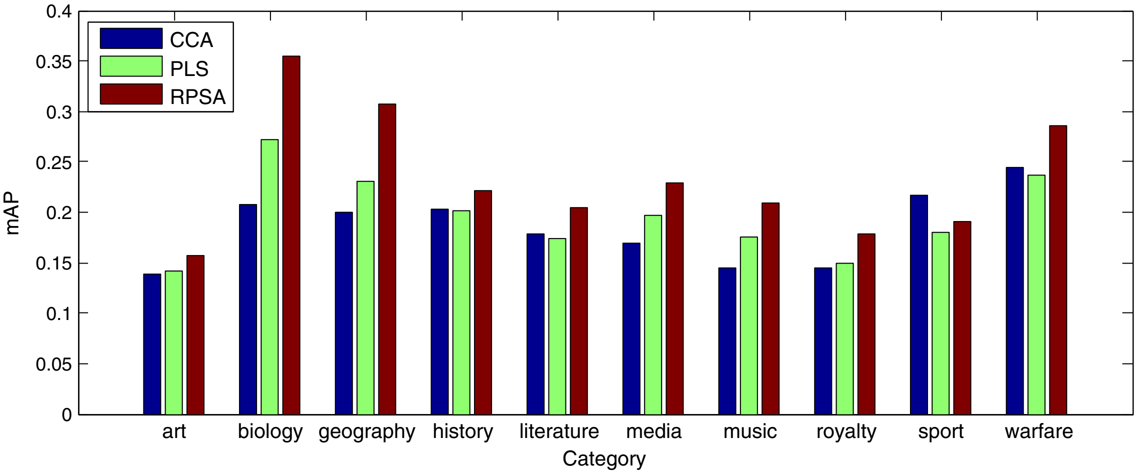

Fig. 10 Comparisons between mAP of each category obtained by CCA, PLS and RPSA

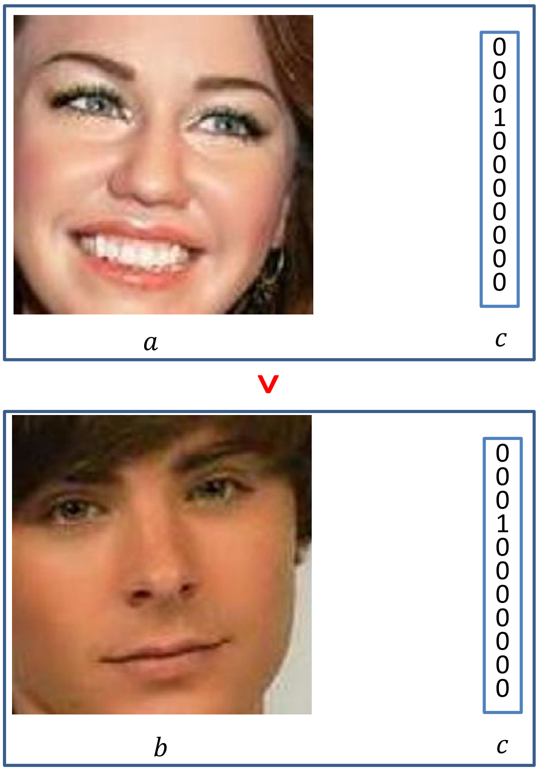

Fig. 11 Illustration of training triplets generated with relative attribute. ∨ indicates being more likely paired. Face *a* is more smiling than face *b*. Therefore, *a* is more likely paired with the smiling (the *fourth*) attribute code *c* than *b* with *c*

attribute vector (1,1,0,0,0,0,0,0,0,0,0) indicates a white male face. In this experiment, face image feature is considered as image modality while attribute code as attribute modality. We learn a 11-d latent common space between image modality and attribute modality, and then retrieve image (attribute) with attribute (image) as query. The evaluation measure is the same as that in Image-Text retrieval in Sect. 5.5.

Different from text-image retrieval experiment, we used natural training triplets instead of those generated with category labels. In Parikh and Grauman (2011), each image is assigned a relative strength for each attribute. If face image *a* has higher strength for the *k* th attribute than image *b*, then image *a* is more-likely paired with the attribute code *c* than image *b*, where *c* being a 11-d zero vector except the *k*th element being 1. Figure 11 shows one example. We used 73,470 training triplets by enumerating all attribute comparison given in the data set.

We set $\overline{n}_1 = 1$ and $\overline{n}_2 = \overline{n}_3 = 0$ as attribute-image retrieval is a cross-modality retrieval problem. To tune γ_2, $\frac{1}{8}$ of training data were selected as validation data and the rest as " training set". We found that RPSA performed best with γ_2 being 100. Therefore, γ_2 was fixed to 100. In order to evaluate RPSA trained only on natural relative-pairing information, we also reported the performance of RPSA without training pairs (named by RPSA$_n$). i.e., $\mathcal{P} = \emptyset$.

Table 7 mAP on public figures face database

Query	Method			
	CCA	PLS	RPSA$_n$	RPSA
Image	0.323	0.323	0.589	0.668
Attribute	0.215	0.323	0.329	0.568
Avg.	0.269	0.323	0.459	0.618

RPSA is compared with CCA and PLS. The results are listed in Table 7. It has been shown that RPSA outperforms its competitors with a large margin. RPSA improves the results of CCA and PLS by 129.7 and 91.3 % in terms of the average mAP. RPSA$_n$ is inferior to RPSA. The reason is that training pairs used by RPSA can encourage the projections of exactly-paired observations from different modalities to be identical, which is important in cross-modality pattern recognition or retrieval as discussed in Sect. 5.2. Nevertheless, RPSA$_n$ is still superior to CCA and PLS. Note that both RPSA and RPSA$_n$ are trained without category labels.

6 Conclusion and Future work

In this paper, we have proposed a framework called Relatively-Paired Space Analysis (RPSA) which can automatically learn a latent common space between multiple modalities from relatively-paired observations. Relative-pairing can explore more general semantic relationships between observations than absolute-pairing, and allows easy integration of label information. Theoretically, RPSA is a discriminative model which does not assume any parameter or noise distribution, and is a general framework which can be used in any cross-modality pattern recognition. We have evaluated the performance of RPSA by applying it to feature fusion, cross-pose face recognition, text-image retrieval and attribute-image retrieval. Experimental results show that RPSA outperforms other state-of-the-art techniques, some of which are tailored for the particular problems. In future work, we would like to extend RPSA to a nonlinear version.

References

Andrea, F., Yoram, S., Sha, F., & Jitendra, M. (2007). Learning globally-consistent local distance functions for shape-based image retrieval. In: *ICCV* (pp. 1–8).

Bach, F., & Jordan, M. (2005). *A probabilistic interpretation of canonical correlation analysis*. Technical Report: Department of Statistics, University of California, Berkeley.

Blanz, V., Grother, P., Phillips, P., & Vetter, T. (2005). Face recognition based on frontal views generated from non-frontal images. In: *CVPR* (pp. 454–461).

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *JMLR*, *3*, 993–1022.

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In: *COMPSTAT* (pp. 177–187).

Bronstein, M., & Bronstein, A. (2010). Data fusion through cross-modality metric learning using similarity-sensitive hashing. In: *CVPR* (pp. 3594–3601).

Chai, X., Shan, S., Chen, X., & Gao, W. (2007). Locally linear regression for pose-invariant face recognition. *TIP*, *16*(7), 1716–1725.

Davis, J.V., Kulis, B., Jain, P., Sra, S., Dhillon, I. (2007). Information theoretic metric learning. In: *ICML* (pp. 209–216).

Ek, C.H., Rihan, J., Torr, P.H.S., Rogez, G., & Lawrence, N.D. (2008). Ambiguity modeling in latent spaces. In: *MLMI* (pp. 62–73).

Goldberger, J., Roweis, S., Hinton, G., & Salakhutdinov, R. (2004). Neighbourhood components analysis. In: *NIPS* (pp. 513–520).

Gong, Y., Ke, Q., Isard, M., & Lazebnik, S. (2014). A multi-view embedding space for modeling internet images, tags, and their semantics. *IJCV*, *106*(2), 210–233.

Gross, R., Matthews, I., & Baker, S. (2004). Appearance-based face recognition and light-fields. *PAMI*, *26*(4), 449–465.

Hardoon, D. R., Szedmak, S., & Shawe-Taylor, J. (2004). Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, *16*(12), 2639–2664.

Joachims, T. (2006). Training linear SVMs in linear time. In: KDD, pp 217–226.

Joachims, T., Finley, T., & Yu, C. N. J. (2009). Cutting-plane training of structural SVMs. *Machine Learning*, *77*(1), 27–59.

Kan, M., Shan, S., & Zhang, H. (2012). Multi-view discriminant analysis. In: *ECCV* (pp. 808–821).

Knutsson, H., Borga, M., & Tomas, L. (1997). Learning canonical correlations. In: *SCIA, Computer Vision Laboratory, vol 1*.

Kuang, Z., & Wong, K.Y.K. (2013). Relatively-paired space analysis. In: *BMVC*.

Kumar, N., Berg, A.C., Belhumeur, P.N., & Nayar, S.K. (2009). Attribute and simile classifiers for face verification. In: *ICCV*.

Lampert, C., & Krömer, O. (2010). Weakly-paired maximum covariance analysis for multimodal dimensionality reduction and transfer learning. In: *ECCV* (pp. 566–579).

Lin, D., & Tang, X. (2005). Coupled space learning of image style transformation. In: *ICCV* (pp. 1699–1706).

Lin, D., & Tang, X. (2006). Inter-modality face recognition. In: *ECCV* (pp. 13–26).

Liu, D. C., & Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, *45*(1), 503–528.

Liu, X., & Chen, T. (2005). Pose-robust face recognition using geometry assisted probabilistic modeling. *CVPR*, *1*, 502–509.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *IJCV*, *60*(2), 91–110.

Navaratnam, R., Fitzgibbon, A.W., & Cipolla, R. (2007). The joint manifold model for semi-supervised multi-valued regression. In: *ICCV* (pp. 1–8).

Parameswaran, S., & Weinberger, K. (2010). Large margin multi-task metric learning. In: *NIPS* (pp. 1–9).

Parikh, D., & Grauman, K. (2011). Relative attributes. In: *ICCV*.

Prince, S., Warrell, J., Elder, J., & Felisberti, F. (2008). Tied factor analysis for face recognition across large pose differences. *PAMI*, *30*(6), 970–984.

Quadrianto, N., & Lampert, C. (2011). Learning multi-view neighborhood preserving projections. In: *ICML* (pp. 425–432).

Rakotomamonjy, A. (2004). *Support vector machines and area under ROC curves*. PSI-INSA de Rouen: Technical Report.

Rasiwasia, N., Pereira, J.C., Coviello, E., Doyle, G., Lanckriet, G.R.G., Levy, R., & Vasconcelos, N. (2010). A new approach to cross-modal multimedia retrieval. In: *ACM MM* (pp. 251–260).

Rosipal, R., & Krämer, N. (2006). Overview and recent advances in partial least squares. *Subspace, latent structure and feature selection* (pp. 34–51). Berlin: Springer.

Rupnik, J., & Shawe-Taylor, J. (2010). Multi-view canonical correlation analysis. In: *SiKDD*.

Saenko, K., Kulis, B., Fritz, M., & Darrell, T. (2010). Adapting visual category models to new domains. In: *ECCV* (pp. 1–14).

Shalev-Shwartz, Singer, Y., & Srebro, N. (2007). Pegasos: Primal estimated sub-GrAdient SOlver for SVM. In: *ICML*.

Sharma, A., & Jacobs, D.W. (2011). Bypassing synthesis: PLS for face recognition with pose, low-resolution and sketch. In: *CVPR* (pp. 593–600).

Sharma, A., & Kumar, A. (2012). Generalized multiview analysis: A discriminative latent space. In: *CVPR* (pp. 2160–2167).

Shen, C., Kim, J., Wang, L., & Hengel, A. (2009). Positive semidefinite metric learning with boosting. In: *NIPS* (pp. 1651–1659).

Shen, C., Kim, J., & Wang, L. (2011). A scalable dual approach to semidefinite metric learning. In: *CVPR* (pp. 2601–2608).

Shon, A.P., Grochow, K., Hertzmann, A., & Rao, R.P.N. (2006). Learning shared latent structure for image synthesis and robotic imitation. In: *NIPS* (pp. 1233–1240).

Stewart, G. (1993). On the early history of the singular value decomposition. In: *SIAM* (pp. 551–566).

Sun, T., Chen, S., Yang, J., & Shi, P. (2008). A novel method of combined feature extraction for recognition. In: *ICDM* (pp. 1043–1048).

Taskar, B. (2004). Learning structured prediction models: A large margin apporach. PhD thesis, Stanford University.

Tenenbaum, J., & Freeman, W. (2000). Separating style and content with bilinear models. *Neural Computation*, *12*(6), 1247–1283.

Torre, F., & Black, M. (2001). Dynamic coupled component analysis. *CVPR*, *2*, 643–650.

Tsochantaridis, I., Hofmann, T., Joachims, T., & Altun, Y. (2004). Support vector machine learning for interdependent and structured output spaces. In: *ICML* (pp. 104–112).

Wang, B., Tang, J., Fan, W., Chen, S., Yang, Z., & Liu, Y. (2009). Heterogeneous cross domain ranking in latent space categories and subject descriptors. In: *CIKM*.

Weinberger, K.Q., Blitzer, J., & Saul, L.K. (2006). Distance metric learning for large margin nearest neighbor classification. In: *NIPS*.

Wu, W., Xu, J., & Li, H. (2010). Learning similarity function between objects in heterogeneous spaces. Tech. Rep. MSR-TR-2010-86.

Xing, E. P., Ng, A. Y., Jordan, M. I., & Russell, S. (2002). Distance metric learning, with application to clustering with side-information. *NIPS*, *15*, 505–512.

Zhang, J., & Zhang, D. (2011). A novel ensemble construction method for multi-view data using random cross-view correlation between within-class examples. *Pattern Recognition*, *44*(6), 1162–1171.

Zhang, W., Wang, X., & Tang, X. (2011). Coupled information-theoretic encoding for face photo-sketch recognition. In: *CVPR* (pp. 513–520).

Zheng, W., Gong, S., & Tao, X. (2013). Re-identification by relative distance comparison. *PAMI*, *35*(3), 653–668.

Zhou, H., Kuang, Z., & Wong, K.Y.K. (2012). Markov Weight Fields for face sketch synthesis. In: *CVPR* (pp. 1091–1097).