

Low-Rank Bilinear Classification: Efficient Convex Optimization and Extensions

Takumi Kobayashi

International Journal of Computer Vision

ISSN 0920-5691

Int J Comput Vis

DOI 10.1007/s11263-014-0709-5



Your article is protected by copyright and all rights are held exclusively by Springer Science +Business Media New York. This e-offprint is for personal use only and shall not be self-archived in electronic repositories. If you wish to self-archive your article, please use the accepted manuscript version for posting on your own website. You may further deposit the accepted manuscript version in any repository, provided it is only made publicly available 12 months after official publication or later and provided acknowledgement is given to the original source of publication and a link is inserted to the published article on Springer's website. The link must be accompanied by the following text: "The final publication is available at link.springer.com".

Low-Rank Bilinear Classification: Efficient Convex Optimization and Extensions

Takumi Kobayashi

Received: 9 June 2013 / Accepted: 25 February 2014
© Springer Science+Business Media New York 2014

Abstract In pattern classification, it is needed to efficiently treat not only feature vectors but also feature matrices defined as two-way data, while preserving the two-way structure such as spatio-temporal relationships. The classifier for the feature matrix is generally formulated in a bilinear form composed of row and column weights which jointly result in a matrix weight. The rank of the matrix should be low from the viewpoint of generalization performance and computational cost. For that purpose, we propose a low-rank bilinear classifier based on the efficient convex optimization. In the proposed method, the classifier is optimized by minimizing the trace norm of the classifier (matrix) to reduce the rank without any hard constraint on it. We formulate the optimization problem in a tractable convex form and provide the procedure to solve it efficiently with the global optimum. In addition, we propose two novel extensions of the bilinear classifier in terms of multiple kernel learning and cross-modal learning. Through kernelizing the bilinear method, we naturally induce a novel multiple kernel learning. The method integrates both the inter kernels between heterogeneous reproducing kernel Hilbert spaces (RKHSs) and the ordinary kernels within respective RKHSs into a new discriminative kernel in a unified manner using the bilinear model. Besides, for cross-modal learning, we consider to map into the common space the multi-modal features which are subsequently classified in that space. We show that the projection and the classification

are jointly represented by the bilinear model, and then propose the method to optimize both of them simultaneously in the bilinear framework. In the experiments on various visual classification tasks, the proposed methods exhibit favorable performances compared to the other methods.

Keywords Bilinear classifier · Low-rank matrix · Convex optimization · Multiple kernel learning · Cross modal learning

1 Introduction

Various applications related to pattern classification are built upon classifying the features which are extracted from the target domains such as images (Dalal and Triggs 2005; Csurka et al. 2004) and motion sequences (Kobayashi and Otsu 2009, 2012b). Those features are usually represented in a *vector* form (one-way) such as by concatenation, even though they are inherently defined in a *matrix* form (two-way). For example, the matrix forms are typically found in image pixels, arrays of (local) feature vectors extracted at spatio-temporal (grid) points, and co-occurrence features (Fig. 1). The dimensionality of the concatenated feature vector corresponds to the product of two-way's dimensions, resulting in high dimensionality, and the inherent structure of the two-way features, e.g., spatial relationship, is unfortunately collapsed (Tenenbaum and Freeman 2000).

There are some works to directly deal with the features in a matrix form; 2D-PCA (Yang et al. 2004) and 2D-LDA (Ye et al. 2005) are extended from principal component analysis (PCA) and linear discriminant analysis (LDA) to matrix-based formulations for dimensionality reduction, and the factorization methods of the feature matrix are also proposed by Lee and Seung (1999) and Eriksson and van den

Communicated by Carlo Colombo.

This article is an extended version of the ECCV 2012 conference paper (Kobayashi and Otsu 2012a)..

T. Kobayashi (✉)
National Institute of Advanced Industrial Science and Technology,
Tsukuba, Japan
e-mail: takumi.kobayashi@aist.go.jp

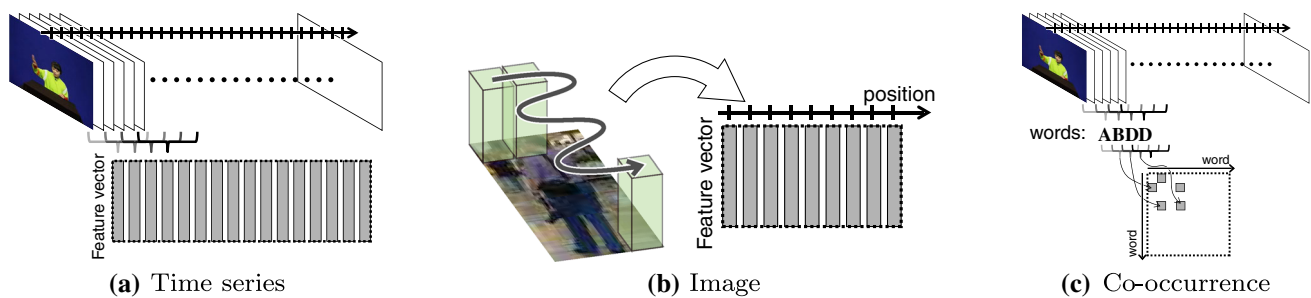


Fig. 1 Examples of feature matrices. Matrix forms are found in arrays of feature vectors extracted from **a** time-series and **b** image data, and in **c** co-occurrence features

Hengel (2010). Those methods are formulated mainly to learn effective image representations in a matrix form and are not explicitly intended for classification problems in a supervised framework.

As to classification, significant research efforts have been made on training linear classifiers for feature vectors; e.g., SVM (Vapnik 1998) and L_1 -SVM (Graepel et al. 1999) in the maximum-margin framework (Smola et al. 2000), which can be further extended to kernel-based methods (Schölkopf and Smola 2001). On the other hand, the classifier for feature matrices is naturally defined as a *bilinear* model (Tenenbaum and Freeman 2000) comprising two kinds of weights for row and column which jointly form a matrix weight in general. Along with the advances of the linear classifiers, some bilinear classifiers have been recently proposed by Pirsiavash et al. (2009) and Wolf et al. (2007). The main concern in the bilinear methods is to construct the *low-rank* bilinear (matrix) classifier in a manner similar to the maximum-margin framework since the VC-dimension of the low-rank bilinear model is proven to be less than that of the concatenated linear models (Wolf et al. 2007). The bilinear model has also attracted attention in the field of the collaborative filtering (Rennie and Srebro 2005; Loeff and Farhadi 2008) and it is defined in a way similar to the above-mentioned classifier. Those bilinear-related methods formulate the optimization problems in a biconvex (non-convex) form and semi-definite programming (SDP) which is computationally less efficient, and in order to cope with the non-convexity and computationally inefficiency, the approximated optimization approaches are employed, though resulting in local minima. In addition, the hard constraint regarding the rank is explicitly introduced as a free parameter; that is, users are required to determine the classifier rank in advance.

In this paper, we propose a novel method to efficiently optimize the bilinear classifier. Without approximations nor hard constraints on the rank, the method automatically produces the optimal low-rank classifier by minimizing the trace norm of the classifier matrix, while reducing the classification errors on training samples. Our contributions are to formulate a tractable convex optimization problem for learning the

low-rank bilinear classifier and to present an optimization procedure to computationally efficiently provide the global minimum. The SDP has so far been mentioned as a convex formulation for the bilinear model, although it suffers from large computational burden. The proposed convex formulation is defined in a different way from the ordinary SDP and it is much faster than the SDP.

In addition, we also propose two novel extensions of the above-mentioned bilinear model in terms of multiple kernel learning (MKL) (Lanckriet et al. 2004) and cross-modal learning (Kan et al. 2012). First, by introducing multiple types of non-linear kernels into the bilinear classifier, we naturally induce a novel MKL method, *heterogeneous MKL*. The MKL methods have been successfully applied to exploit the discriminative power in the form of the composite kernel to which the multiple kernels are (linearly) integrated (Lanckriet et al. 2004; Rakotomamonjy et al. 2008; Varma and Ray 2007). In the proposed heterogeneous MKL, the feature matrix used in the bilinear model is formulated based on the multiple types of reproducing kernel Hilbert space (RKHS). Thereby, we introduce a novel concept of the inter kernels between the heterogeneous types of RKHS features, while the ordinary kernels are also incorporated as intra kernels. The proposed method integrates both the inter and intra kernels into a new discriminative kernel via the bilinear model.

Second, for cross-modal learning, we deal with the multi-modal features which are projected into the common space shared across the multiple modalities (Kan et al. 2012; Sharma and Jacobs 2011). In the common space, we can leverage the knowledge (training samples) transferred from all the modalities; we assume the linear classification in that space. In the case of the multi-modal feature vectors in the multi-class setting, it is shown that both the projections into the common space and the linear classifier are jointly represented by the bilinear model. Accordingly, we propose a method to simultaneously learn both of them in the bilinear framework.

The rest of this paper is organized as follows. In Sect. 2, we define the model of bilinear classifier and briefly review the previous methods related to the optimization for the bilinear

model, and then Sect.3 details the proposed convex formulation for learning the bilinear classifier and its optimization approach. In the subsequent two sections, we present two extensions of the bilinear model. By introducing the non-linear kernel functions into the bilinear classifier, the heterogeneous multiple kernel learning is proposed in Sec.4. The cross-modal learning is casted into the bilinear framework and thereby we propose in Sect.5 the method to classify multi-class multi-modal feature vectors via the bilinear model. In Sect.6, we show the experimental results by applying the proposed methods to various visual classification tasks, and finally conclude this paper in Sect.7.

This paper contains the substantial improvements over the ECCV 2012 conference article (Kobayashi and Otsu 2012a) in the following points; (1) the bilinear model is extended to a new cross-modal learning method for dealing with multi-modal features in the multi-class setting, (2) the experiments on cross-modal learning are conducted to show the capability of that new method, and (3) some technical contents, such as the optimization procedure in Sect.3.2, are presented in more detail.

2 Bilinear Classifier

Let X be a feature matrix whose dimensions are denoted by h and w ($X \in \mathbb{R}^{h \times w}$). For example, X is regarded as the array of the h -dimensional feature vectors extracted at w points, such as in xy -coordinates for images or along t -axis for time-series signals, as shown in Fig. 1. To deal with the feature matrix, a bilinear classifier is simply formulated as $\hat{y} = \mathbf{w}_h^\top X \mathbf{w}_w + b$ where $\mathbf{w}_h \in \mathbb{R}^h$, $\mathbf{w}_w \in \mathbb{R}^w$. This is regarded as a '1-rank' classifier, and by integrating multiple such classifiers, the multi-rank bilinear classifier is generally defined by

$$\hat{y} = \text{tr}(\mathbf{W}_h^\top X \mathbf{W}_w) + b = \text{tr}(\mathbf{W}^\top X) + b, \quad (1)$$

where $\text{tr}(\cdot)$ denotes the trace of a matrix, b is the bias and $\mathbf{W} = \mathbf{W}_h \mathbf{W}_w^\top \in \mathbb{R}^{h \times w}$ is the classifier matrix ($\mathbf{W}_h \in \mathbb{R}^{h \times r}$, $\mathbf{W}_w \in \mathbb{R}^{w \times r}$ where the rank $r \leq \min[w, h]$). For simplicity, we consider a two-class classification problem, given n samples $\{X_i, y_i\}_{i=1, \dots, n}$ where y_i is the class label ($y_i \in \{+1, -1\}$) of the i -th sample. This is straightforwardly extended to multi-class classifiers by using a one-vs-rest approach.

For training the bilinear classifier, as in the maximum-margin framework (Smola et al. 2000), we measure the margin of the bilinear classifier (1) by the matrix trace norm, *i.e.*, sum of singular values, to minimize the matrix rank, which renders the following optimization problem:

$$\min_{\mathbf{W}, b} \|\mathbf{W}\|_\Sigma + C \sum_{i=1}^n \max[0, 1 - y_i \{\text{tr}(\mathbf{W}^\top X_i) + b\}], \quad (2)$$

where $\|\cdot\|_\Sigma$ indicates the trace norm of a matrix and C is the balancing parameter between the margin and the classification errors.

Let the singular values of \mathbf{W} be denoted by $\sigma \in \mathbb{R}^r$. The trace norm is represented by $\|\mathbf{W}\|_\Sigma = \|\sigma\|_1$, while the rank is measured as $\text{rank}(\mathbf{W}) = \|\sigma\|_0$. Thereby, in the formulation (2), the L_1 norm (trace norm) in the objective cost is regarded as a relaxation of the L_0 norm which directly minimizes the rank. Generally speaking, such L_1 -norm minimization induces sparsity (Graepel et al. 1999) in the singular values, minimizing the rank accordingly.

In the following, we briefly review the previous methods regarding the optimization of the bilinear classifier; they modify the form of the margin since it is difficult to directly treat the trace norm in the optimization.

2.1 SDP Formulation for Bilinear Model

Srebro et al. (2005) showed the way to reformulate (2) into SDP. By introducing the following augmented variables,

$$\begin{aligned} \tilde{X}_i &= \begin{bmatrix} \mathbf{0} & X_i \\ X_i^\top & \mathbf{0} \end{bmatrix}, \tilde{\mathbf{W}} = \begin{bmatrix} \mathbf{W}_h & \\ & \mathbf{W}_w \end{bmatrix} \begin{bmatrix} \mathbf{W}_h \\ \mathbf{W}_w \end{bmatrix}^\top \\ &= \begin{bmatrix} \mathbf{W}_h \mathbf{W}_h^\top & \mathbf{W} \\ \mathbf{W}^\top & \mathbf{W}_w \mathbf{W}_w^\top \end{bmatrix}, \end{aligned} \quad (3)$$

the formulation (2) results in

$$\min_{\tilde{\mathbf{W}} \succ_{0,b}} \frac{1}{2} \text{tr}(\tilde{\mathbf{W}}) + C \sum_{i=1}^n \max\left[0, 1 - y_i \left\{ \frac{1}{2} \text{tr}(\tilde{\mathbf{W}}^\top \tilde{X}_i) + b \right\}\right]. \quad (4)$$

The trace norm $\|\mathbf{W}\|_\Sigma$ is replaced with $\frac{1}{2} \text{tr}(\tilde{\mathbf{W}})$ since $\|\mathbf{W}\|_\Sigma \leq \frac{1}{2} \text{tr}(\tilde{\mathbf{W}})$ and the equality holds at the optimum (Srebro et al. 2005). In this convex problem (4), the global optimum is obtained by SDP (Boyd and Vandenberghe 2004), but with a high computational cost, which makes it infeasible for large-scaled samples.

This kind of bilinear model is also addressed in the literature of the collaborative filtering by Rennie and Srebro (2005) and Loeff and Farhadi (2008) who employ the trace norm $\|\mathbf{W}\|_\Sigma$ as in (2). Those authors also mentioned that the resulting SDP is exhaustive and difficult to solve for large-scale samples, and thus they apply the approximated optimization approaches, though resulting in local minima.

2.2 Bilinear SVM

Pirsiavash et al. (2009) formulate the optimization problem slightly differently from (2) by

$$\min_{\mathbf{W}_h, \mathbf{W}_w, b} \frac{1}{2} \text{tr}(\mathbf{W}_h \mathbf{W}_w^\top \mathbf{W}_w \mathbf{W}_h^\top) + C \sum_i \max[0, 1 - y_i \{\text{tr}(\mathbf{W}_w \mathbf{W}_h^\top \mathbf{X}_i) + b\}]. \tag{5}$$

In this formulation, the margin is measured by the Frobenius norm, $\|\mathbf{W}_h \mathbf{W}_w^\top\|_F^2 = \text{tr}(\mathbf{W}_h \mathbf{W}_w^\top \mathbf{W}_w \mathbf{W}_h^\top)$, instead of the trace norm. Due to employing the Frobenius norm, (5) is quite similar to the linear SVM (Vapnik 1998); in case that we introduce $\mathbf{W} = \mathbf{W}_h \mathbf{W}_w^\top$ as the classifier weight to be optimized without decomposition, (5) results in the same formulation as the SVM by unfolding matrices to vectors. The Frobenius norm corresponds to the L_2 norm of the singular values σ and the optimized classifier tends to have dense singular values, resulting in the *full-rank* classifier, unlike the trace norm (the L_1 norm of σ). Therefore, to achieve low-rank SVM, Wolf et al. (2007) and Pirsivash et al. (2009) additionally introduce a hard constraint on the rank, $\text{rank}(\mathbf{W}) \leq k$ or correspondingly the explicit representation of $\mathbf{W}_h \in \mathbb{R}^{h \times k}$, $\mathbf{W}_w \in \mathbb{R}^{w \times k}$ in the formulation (5). It, however, usually produces the classifier of the full (maximum) rank under that constraint, *i.e.*, $\text{rank}(\mathbf{W}) = k$, since the objective cost including the Frobenius norm $\|\mathbf{W}\|_F^2$ favors dense singular values as discussed above, and the optimal rank k is generally difficult to determine in advance; it depends on the recognition task. Besides, it should be noted that (5) is not convex but biconvex with respect to \mathbf{W}_h and \mathbf{W}_w , and those two types of weights are alternatively optimized, converging to the local minima.

3 Efficient Convex Optimization

3.1 Proposed Formulation

By using $\mathbf{W}_h, \mathbf{W}_w$ as in (5), we can simply rewrite (4) to

$$\min_{\mathbf{W}_w} \left[\frac{1}{2} \text{tr}(\mathbf{W}_w \mathbf{W}_w^\top) + \min_{\mathbf{W}_h, b} L(\mathbf{W}_h, b; \mathbf{W}_w) \right], \tag{6}$$

$$L(\mathbf{W}_h, b; \mathbf{W}_w) \triangleq \frac{1}{2} \text{tr}(\mathbf{W}_h \mathbf{W}_h^\top) + C \sum_i \max[0, 1 - y_i \{\text{tr}(\mathbf{W}_h^\top \mathbf{X}_i \mathbf{W}_w) + b\}]. \tag{7}$$

The formulation (6) is biconvex or bilevel (Dempe 2002), and the iterative approach which alternately optimizes either of \mathbf{W}_h or \mathbf{W}_w is applicable as described in Sect.2.2. Such an approach is tractable in contrast to SDP (4), but it results in local minima. In this study, we further reformulate (6) to a tractable *convex* problem and propose a procedure to provide the global optimum efficiently.

Here, we suppose the column size is smaller than the row size, $h > w$, without loss of generality. The inner optimization, $\max_{\mathbf{W}_h, b} L(\mathbf{W}_h, b; \mathbf{W}_w)$ in (6), is regarded as the SVM

quadratic programming (QP) with respect to \mathbf{W}_h, b , and thus it has the following dual:

$$\max_{\alpha \in \Omega} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K_{ij}, \tag{8}$$

$$\text{where } \Omega = \{\alpha \mid \forall i, 0 \leq \alpha_i \leq C, \sum_{i=1}^n y_i \alpha_i = 0\}, \tag{9}$$

$$K_{ij} = \text{tr}\{(\mathbf{X}_i \mathbf{W}_w)^\top (\mathbf{X}_j \mathbf{W}_w)\} = \text{tr}(\mathbf{W}_w \mathbf{W}_w^\top \mathbf{X}_i^\top \mathbf{X}_j). \tag{10}$$

Thereby, given the optimum \mathbf{W}_w^* , we get the optimum bilinear classifier as

$$\mathbf{W}_h^* = \sum_i \alpha_i^* y_i \mathbf{X}_i \mathbf{W}_w^*, \quad \hat{y} = \text{tr}(\mathbf{W}_w^* \mathbf{W}_w^{*\top} \sum_i \alpha_i^* y_i \mathbf{X}_i^\top \mathbf{X}), \tag{11}$$

where α_i^* are the optimizers in (8) using \mathbf{W}_w^* . In the forms (6,10,11), we can see that the key variable is $\Sigma_w \triangleq \mathbf{W}_w \mathbf{W}_w^\top \succcurlyeq 0$ rather than \mathbf{W}_w itself. And, since the inner optimization in (6) can be replaced with its dual (8) due to the string duality (Rakotomamonjy et al. 2008; Varma and Ray 2007), the optimization (6) is reformulated into

$$\min_{\Sigma_w \succcurlyeq 0} \left[\frac{1}{2} \text{tr}(\Sigma_w) + \max_{\alpha \in \Omega} \left\{ \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K_{ij}(\Sigma_w) \right\} \right], \tag{12}$$

$$\text{where } K_{ij}(\Sigma_w) = \text{tr}(\Sigma_w \mathbf{X}_i^\top \mathbf{X}_j). \tag{13}$$

This is still one form of bilevel optimization (Dempe 2002), but by using the *unique* optimizer α^* in (8), we finally obtain our proposed formulation for optimizing the bilinear classifier by

$$\min_{\Sigma_w \succcurlyeq 0} \left[J(\Sigma_w) \triangleq \frac{1}{2} \text{tr}(\Sigma_w) + \sum_i \alpha_i^*(\Sigma_w) - \frac{1}{2} \sum_{i,j} \alpha_i^*(\Sigma_w) \alpha_j^*(\Sigma_w) y_i y_j K_{ij}(\Sigma_w) \right], \tag{14}$$

$$\text{where } \alpha^*(\Sigma_w) = \arg \max_{\alpha \in \Omega} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K_{ij}(\Sigma_w). \tag{15}$$

By folding the inner optimization into $\alpha^*(\Sigma_w)$ (the function of Σ_w) via (15), the optimization problem (14) is a single-level form with respect to Σ_w and it is *convex* (see Appendix for the proof). The introduction of the essential variable Σ_w removes the ambiguity regarding the rotations of the classifier weights; even though any rotation matrix $\Theta \in \mathbb{R}^{r \times r}$ ($\Theta \Theta^\top = \mathbf{I}$) is applied to the row and column weights by $\mathbf{W}_h \Theta$ and $\mathbf{W}_w \Theta$, neither the bilinear classifier nor the margin is changed: $\hat{y} = \text{tr}(\Theta^\top \mathbf{W}_h^\top \mathbf{X} \mathbf{W}_w \Theta) = \text{tr}(\mathbf{W}_h^\top \mathbf{X} \mathbf{W}_w)$ in the classifier and $\text{tr}(\mathbf{W}_h \Theta \Theta^\top \mathbf{W}_h^\top) = \text{tr}(\mathbf{W}_h \mathbf{W}_h^\top)$, $\text{tr}(\mathbf{W}_w \Theta \Theta^\top \mathbf{W}_w^\top) = \text{tr}(\mathbf{W}_w \mathbf{W}_w^\top)$ in the margin.

The essential rank r of the classifier $\mathbf{W} = \mathbf{W}_h \mathbf{W}_w^\top$ is usually less than $\min[w, h] = w$ (full rank), and the redundant ranks are eventually eliminated by assigning zero singular values through the optimization (14) which minimizes the trace norm $\|\mathbf{W}_h \mathbf{W}_w^\top\|_\Sigma$. The proposed method, *i.e.*, the optimization (14), produces the bilinear classifier of the optimal low rank automatically without any hard constraint on the rank. In summary, our contribution is to formulate the bilinear classification problem in the tractable convex form (14) by introducing Σ_w .

We finally obtain the global optimum Σ_w^* , instead of \mathbf{W}_w^* , and then the two types of classifier weights are retrieved through the eigen-decomposition of Σ_w^* by

$$\Sigma_w^* = \mathbf{V}^* \Lambda^* \mathbf{V}^{*\top}, \tag{16}$$

$$\mathbf{W}_w^* = \mathbf{V}^* \Lambda^{*\frac{1}{2}}, \quad \mathbf{W}_h^* = \sum_i \alpha^*(\Sigma_w^*)_{y_i} \mathbf{X}_i \mathbf{V}^* \Lambda^{*\frac{1}{2}}, \tag{17}$$

and the classifier weight is given by

$$\mathbf{W}^* = \sum_i \alpha^*(\Sigma_w^*)_{y_i} \mathbf{X}_i \Sigma_w^*. \tag{18}$$

3.2 Optimization by Gradient Descent

In this study, the cost function J in (14) is minimized by means of the following gradient-descent approach.

3.2.1 Initialization

We simply start with $\Sigma_w = \mathbf{I}$ (identity matrix) since the redundant rank is automatically eliminated by assigning the zero singular values through the optimization as mentioned above. Note that this initialization provides $\alpha^*(\mathbf{I})$ that corresponds to the solution of the linear SVM using $K_{ij} = \text{tr}(\mathbf{X}_i^\top \mathbf{X}_j)$. It is the optimum solution for the full-rank classifier and is regarded as a good initial point for fast convergence.

3.2.2 Sub-Problem QP

We apply the off-the-shelf SVM solver to optimize the sub-problem (15). The SMO-based SVM solver such as libsvm (Chang and Lin 2001) effectively works on it, while the recently developed linear SVM solvers such as liblinear (Fan et al. 2008) are also applicable by decomposing Σ_w into $\mathbf{W}_w \mathbf{W}_w^\top$ to provide the linear kernel in (13).

3.2.3 Gradient Descent

The derivative of J is given, as if $\alpha^*(\Sigma_w)$ do not depend on Σ_w (see Appendix), by

$$\nabla J = \frac{1}{2} \left\{ \mathbf{I} - \sum_{ij} \alpha_i^*(\Sigma_w) \alpha_j^*(\Sigma_w) y_i y_j \mathbf{X}_i^\top \mathbf{X}_j \right\}. \tag{19}$$

In order to ensure the positive semi-definiteness $\Sigma_w \succcurlyeq 0$, we apply the projected gradient descent (Rakotomamonjy et al. 2008; Varma and Ray 2007) via the eigen decomposition of $\Sigma_w - \eta \nabla J$ and cutting off both the negative eigenvalues and their eigenvectors:

$$\Sigma_w^{old} - \eta \nabla J = \mathbf{V} \Lambda \mathbf{V}^\top = \sum_{i=1}^w \lambda_i \mathbf{v}_i \mathbf{v}_i^\top, \tag{20}$$

$$\Sigma_w^{new} \leftarrow \sum_{i|\lambda_i > 0} \lambda_i \mathbf{v}_i \mathbf{v}_i^\top = \mathbf{V}_+ \Lambda_+ \mathbf{V}_+^\top, \tag{21}$$

where η is the step size determined by the line search, say Armijo rule (Nocedal and Wright 1999), and λ_i, \mathbf{v}_i are the i -th eigenvalue and eigenvector, respectively.

In the proposed formulation (14), the dimensionality of the variable Σ_w , $O(w^2)$, is much smaller than that of $\tilde{\mathbf{W}}$, $O((h+w)^2)$, in the SDP (4). The computational complexity of the above optimization procedure is solely dependent on that of the sub-problem QP (15). The computationally exhaustive step other than the QP is the eigen decomposition in (20). In practice, however, either the dimensions h or w is low; *e.g.*, high-dimensional features are extracted at a few points ($h \gg w$), or either of the dimensionalities can be reduced such as by applying PCA in advance; the PCA projection vectors $\mathbf{U} \in \mathbb{R}^{w \times w'}$ transform the feature matrix \mathbf{X} into $\mathbf{X}\mathbf{U} \in \mathbb{R}^{h \times w'}$. Thus, the computational costs of the eigen decomposition are negligible compared to those of QP in most cases.

3.3 Smoothing Regularization

Either or both of the bilinear weights, $\mathbf{W}_h, \mathbf{W}_w$, are occasionally connected to physical properties; for example, \mathbf{W}_w works as weights on spatio-temporal positions, while \mathbf{W}_h is for the feature vector. In such cases, it is useful to take into account the physical relationships between the weight components as regularization, which would improve the generalization performance. For that purpose, we introduce the smoothing regularization. In the case of time-series (Fig. 1a), the extracted features are not independently drawn but naturally have *continuity* between adjacent features, expecting the smooth weights \mathbf{W}_w . The smoothing regularization is expressed by using the quadratic form derived from the Laplacian of the weights, and the formulation results in

$$\begin{aligned} \min_{\mathbf{W}_w, \mathbf{W}_h, b} & \frac{1}{2} \text{tr}(\mathbf{W}_w \mathbf{W}_w^\top) + \frac{1}{2} \text{tr}(\mathbf{W}_h \mathbf{W}_h^\top) \\ & + \frac{1}{2} C_w \text{tr}(\mathbf{W}_w^\top \mathbf{L}_w \mathbf{W}_w) + \frac{1}{2} C_h \text{tr}(\mathbf{W}_h^\top \mathbf{L}_h \mathbf{W}_h) \\ & + C \sum_i \max[0, 1 - y_i \{\text{tr}(\mathbf{W}_h^\top \mathbf{X}_i \mathbf{W}_w) + b\}], \end{aligned} \quad (22)$$

where $\mathbf{L}_w, \mathbf{L}_h$ are the matrices to measure the smoothness and C_w, C_h are regularization parameters. For example, in time-series (one-way), the matrix \mathbf{L}_w is determined based on $\text{tr}(\mathbf{W}_w^\top \mathbf{L}_w \mathbf{W}_w) = \sum_{i,r} \| -\mathbf{w}_{i-1,r} + 2\mathbf{w}_{i,r} - \mathbf{w}_{i+1,r} \|_2^2$. In this study, the regularization parameter C_w is set so as to equally balance the spectral norms of the two matrices \mathbf{I} and \mathbf{L}_w by $C_w = 1/\|\mathbf{L}_w\|_s$, ($C_h = 1/\|\mathbf{L}_h\|_s$ for \mathbf{L}_h), where $\|\cdot\|_s$ denotes the spectral norm (the maximum singular value) of a matrix. Then, the above formulation is rewritten to

$$\begin{aligned} \min_{\bar{\mathbf{W}}_w, \bar{\mathbf{W}}_h, b} & \frac{1}{2} \text{tr}(\bar{\mathbf{W}}_w \bar{\mathbf{W}}_w^\top) + \frac{1}{2} \text{tr}(\bar{\mathbf{W}}_h \bar{\mathbf{W}}_h^\top) \\ & + C \sum_i \max[0, 1 - y_i \{\text{tr}(\bar{\mathbf{W}}_h^\top \bar{\mathbf{X}}_i \bar{\mathbf{W}}_w) + b\}], \end{aligned} \quad (23)$$

$$\text{where } \bar{\mathbf{X}}_i = (\mathbf{I} + C_h \mathbf{L}_h)^{-\frac{1}{2}} \mathbf{X}_i (\mathbf{I} + C_w \mathbf{L}_w)^{-\frac{1}{2}}. \quad (24)$$

This is the same as (6) except for the feature matrices $\bar{\mathbf{X}}$, and the optimization procedure described in Sect.3.2 is directly applicable to it. We finally obtain the smoothed classifier weights by $\mathbf{W}_h = (\mathbf{I} + C_h \mathbf{L}_h)^{-\frac{1}{2}} \bar{\mathbf{W}}_h$, $\mathbf{W}_w = (\mathbf{I} + C_h \mathbf{L}_w)^{-\frac{1}{2}} \bar{\mathbf{W}}_w$.

This smoothing regularization is somewhat a naive extension of the bilinear model. In the following sections (Sect.4 and Sect.5), we propose the two noteworthy extensions of bilinear models by utilizing the proposed method (Sect.3.1, 3.2) as a basic optimization tool.

4 Heterogeneous Multiple Kernel Learning

By considering a kernel-based extension of the bilinear classifier, we naturally induce a novel multiple kernel learning (MKL), called *heterogeneous multiple kernel learning*, which effectively integrates the inter and intra kernels among various types of RKHS.

4.1 Feature Matrix for Kernelization

To kernelize the bilinear model, we introduce the kernels for $\mathbf{R}^{ij} \triangleq \mathbf{X}_i^\top \mathbf{X}_j \in \mathbb{R}^{w \times w}$ in (13). The optimized Σ_w works as weights to integrate the multiple ($w \times w$ types) kernels of \mathbf{R}^{ij} into a new composite kernel which is fed into (13), as in MKL (Lanckriet et al. 2004). In what follows, we consider the kernel feature vector ϕ_{ic} ($c \in \{1, \dots, w\}$) in the c -th type of RKHS \mathcal{K}_c which is derived from the feature \mathbf{x}_{ic} and is

endowed with the kernel function $k_c: k_c(\mathbf{x}_{ic}, \mathbf{x}_{jc}) = \phi_{ic}^\top \phi_{jc}$. Those vectors ϕ_{ic} form the feature matrix along its (block-)diagonal as

$$\mathbf{X}_i^\phi = \begin{bmatrix} \phi_{i1} & & \\ & \ddots & \\ & & \phi_{iw} \end{bmatrix} = \text{diag}(\phi_{i1}, \dots, \phi_{iw}), \quad (25)$$

whose column size is w . If we simply use $\mathbf{X}_i = \mathbf{X}_i^\phi$, the kernelized \mathbf{R}^{ij} results in the diagonal matrix $\mathbf{R}^{ij} = \text{diag}\{k_1(\mathbf{x}_{i1}, \mathbf{x}_{j1}), \dots, k_w(\mathbf{x}_{iw}, \mathbf{x}_{jw})\}$. In this case, we take into account only the respective types of kernels in disregard of the inter connections among those kernels, and the proposed bilinear method (14) using this diagonal matrix \mathbf{R}^{ij} reduces to the MKL method by Varma and Ray (2007).

In order to exploit the relationships between the multiple types of RKHSs, we define the feature matrix \mathbf{X} by

$$\mathbf{X} = \mathbf{Z} \mathbf{X}^\phi, \quad (26)$$

where \mathbf{Z} is a transformation matrix. For example, in case that the RKHSs are all homogeneous with the identical kernel function $k_c = k \forall c$, the kernelized \mathbf{R} can be simply obtained as the dense matrix $\mathbf{R}^{ij} = \{\phi_{ic}^\top \phi_{jd} = k(\mathbf{x}_{ic}, \mathbf{x}_{jd})\}_{c=1, \dots, w}^{d=1, \dots, w}$ via $\mathbf{Z} = [\mathbf{I}, \dots, \mathbf{I}]$. In this study, by effectively determining \mathbf{Z} , we establish the *densely* kernelized \mathbf{R} even for multiple types of RKHSs, namely multiple kernel functions, to incorporate not only the intra kernels (in diagonal) between the homogeneous RKHSs but also the inter kernels (in off-diagonal) among the heterogeneous RKHSs, which induces heterogeneous MKL (hMKL) as follows.

4.2 Heterogeneous Kernel Integration

The main concern in the hMKL is to construct kernels, especially for the off-diagonal elements of \mathbf{R}^{ij} ; those are inter kernels between the heterogeneous RKHSs, which is a novel concept in this paper. We determine the transformation \mathbf{Z} by

$$\mathbf{Z} = [\mathbf{U}_1 \mathbf{V}_1^\top, \dots, \mathbf{U}_w \mathbf{V}_w^\top] = [\mathbf{K}_1^{-\frac{1}{2}} \Phi_1^\top, \dots, \mathbf{K}_w^{-\frac{1}{2}} \Phi_w^\top], \quad (27)$$

where $\Phi_c = [\phi_{1c}, \dots, \phi_{nc}] = \mathbf{V}_c \Lambda_c \mathbf{U}_c^\top$ (SVD). It leads to the following form of the kernelized \mathbf{R}^{ij} ,

$$\mathbf{R}_{cd}^{ij} = \phi_{ic}^\top \mathbf{V}_c \mathbf{U}_c^\top \mathbf{U}_d \mathbf{V}_d^\top \phi_{jd} = \mathbf{k}_{ic}^\top \mathbf{K}_c^{-\frac{1}{2}} \mathbf{K}_d^{-\frac{1}{2}} \mathbf{k}_{id}, \quad (28)$$

where \mathbf{K}_c is the c -th kernel Gram matrix using the kernel function k_c ;

$$\mathbf{K}_c = \{k_c(\mathbf{x}_{ic}, \mathbf{x}_{jc})\}_{i=1, \dots, n}^{j=1, \dots, n} = \Phi_c^\top \Phi_c = \mathbf{U}_c \Lambda_c^2 \mathbf{U}_c^\top \in \mathbb{R}^{n \times n}, \quad (29)$$

$$\mathbf{K}_c^{-\frac{1}{2}} = \mathbf{U}_c \Lambda_c^{-1} \mathbf{U}_c^\top, \quad (30)$$

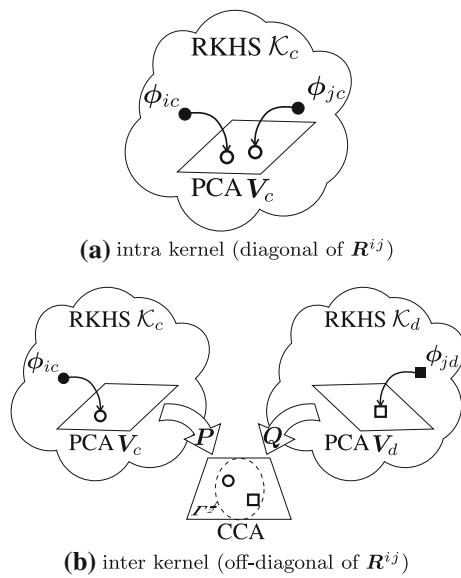


Fig. 2 Interpretation in the proposed kernel (28)

and the kernel vector k_{ic} corresponds to the i -th column vector of K_c as

$$k_{ic} = [k_c(x_{ic}, x_{1c}), \dots, k_c(x_{ic}, x_{nc})]^\top = \Phi_c^\top \phi_{ic} \in \mathbb{R}^n. \tag{31}$$

The proposed kernel formulated in (28) is composed of two parts, *intra* kernel (diagonal, $c = d$) and *inter* kernel (off-diagonal, $c \neq d$), to which we can give interpretations as follows (Fig. 2).

4.2.1 Intra Kernel

Due to $U_c^\top U_c = I$, the intra kernels in the diagonal components of R^{ij} result in

$$R_{cc}^{ij} = k_{ic}^\top K_c^{-\frac{1}{2}} K_c^{-\frac{1}{2}} k_{jc} = \phi_{ic}^\top V_c V_c^\top \phi_{jc}, \tag{32}$$

where V_c is regarded as the (kernel) PCA projection vectors (Schölkopf and Smola 2001) in the c -th RKHS \mathcal{K}_c . Thus, this is simply interpreted as the inner product on the PCA subspace in \mathcal{K}_c (Fig. 2a). Especially, on the training samples, the diagonal components of R^{ij} are identical to the original kernels; $R_{cc}^{ij} = k_c(x_{ic}, x_{jc})$.

4.2.2 Inter Kernel

The formulation (28) with $c \neq d$ is closely related to (kernel) canonical correlation analysis (CCA) (Akaho 2001). The CCA provides the projections A, B for the two types of features Φ_c, Φ_d in $\mathcal{K}_c, \mathcal{K}_d$ so as to maximize the correlation coefficient, by solving the following eigenvalue problem:

$$\begin{bmatrix} \mathbf{0} & \Phi_c \Phi_d^\top \\ \Phi_d \Phi_c^\top & \mathbf{0} \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} \Phi_c \Phi_c^\top & \mathbf{0} \\ \mathbf{0} & \Phi_d \Phi_d^\top \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} \Gamma, \tag{33}$$

$$\therefore A = V_c \Lambda_c^{-1} P, B = V_d \Lambda_d^{-1} Q, \tag{34}$$

where $U_c^\top U_d = P \Gamma Q^\top$ (SVD). In the CCA, the feature vectors ϕ_{ic}, ϕ_{jd} are first whitened by $V_c \Lambda_c^{-1}, V_d \Lambda_d^{-1}$ via PCA, and then the PCA axes are rotated by P, Q so as to ensure the consistency, maximizing the correlation coefficient. By using these notations, (28) is further rewritten to

$$R_{cd}^{ij} = \phi_{ic}^\top V_c P \Gamma^{\frac{1}{2}} \Gamma^{\frac{1}{2}} Q^\top V_d^\top \phi_{jd}. \tag{35}$$

This is a quite similar form to the CCA projection; the feature vectors ϕ_{ic}, ϕ_{jd} are first projected by PCA vectors V_c, V_d , and then they are rotated by the CCA rotation matrices P, Q with weighting the CCA axes by the correlation coefficients $\Gamma^{\frac{1}{2}}$ (Fig. 2b). The differences from the CCA projections are that (1) we use the orthogonal PCA projection, not whitening, to preserve the magnitude (norm) of ϕ_{ic}, ϕ_{jd} and then (2) we employ the weighting by the correlation coefficients Γ which measure consistency along respective CCA axes. That is, the inner product in the CCA space is enhanced along the highly consistent axis of higher Γ .

In summary, the proposed kernel (28) is based on the inner product by applying the PCA projections to the kernel feature vectors in RKHSs. For the intra kernel in homogeneous RKHSs which are intrinsically consistent, the inner product is directly employed (Fig. 2a). Whereas, the inter kernel between heterogeneous RKHSs requires additionally CCA rotations to ensure the consistency so that the (reasonable) inner product is computed (Fig. 2b).

By using the transformation (27), the feature matrix that we deal with in the proposed hMKL is explicitly represented by

$$X = Z X^\phi = [K_1^{-\frac{1}{2}} k_1, \dots, K_w^{-\frac{1}{2}} k_w] \in \mathbb{R}^{n \times w}, \tag{36}$$

$$k_c = [k_c(x_c, x_{1c}), \dots, k_c(x_c, x_{nc})]^\top \in \mathbb{R}^n. \tag{37}$$

Therefore, the bilinear classifier in the hMKL is also described by the bilinear form (1) and the proposed bilinear optimization (Sec.3) is directly applicable even to this hMKL. It should be noted again that $\Sigma_w = W_w W_w^\top$ is regarded as the weights on both the intra and inter kernels; $K_{ij} = \text{tr}(\Sigma_w R^{ij})$ in (13).

5 Cross-Modal Learning in Bilinear Framework

The bilinear formulation is also applicable to cope with multi-modal features in cross-modal learning. The objects to be classified are occasionally represented in multiple

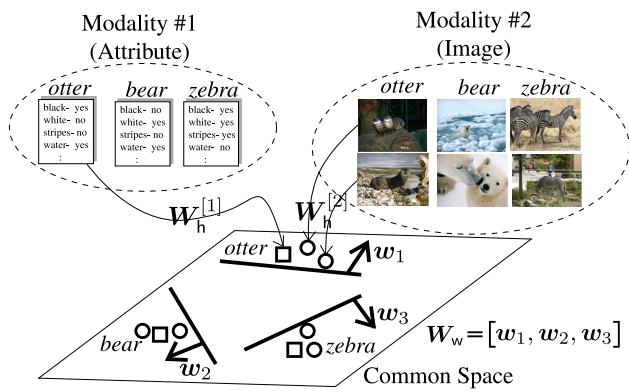


Fig. 3 Classification for multi-modal features in the multi-class setting

modalities; for example, faces are depicted by *photos* and *sketches* (Wang and Tang 2009), and animals are described by *images* and *attributes* (Lampert et al. 2009) (Fig. 3). We first describe in Sect.5.1 the formulation for classifying the multi-modal feature matrices in two classes and then show how the multi-modal feature vectors in the *multi-class* setting are casted into that bilinear formulation in Sect.5.2.

5.1 Bilinear Formulation

Suppose we have M types of modalities from which the features matrices $X_i^{[m]} \in \mathbb{R}^{h^{[m]} \times w}$ with labels $y_i^{[m]} \in \{+1, -1\}$ ($m = 1, \dots, M, i = 1, \dots, n^{[m]}$) are derived, and thereby the m -th modal classifier is defined by $\hat{y}^{[m]} = \text{tr}(\mathbf{W}^{[m]} X_i^{[m]} + b^{[m]})$. Here, the classifier weight $\mathbf{W}^{[m]} \in \mathbb{R}^{h^{[m]} \times w}$ is decomposed into $\mathbf{W}^{[m]} = \mathbf{W}_h^{[m]} \mathbf{W}_w^\top$ and the column weight \mathbf{W}_w is assumed to be shared across all of the M modalities (classifiers). In this formulation, the classification consists of the following two procedures; the m -th modal feature $X^{[m]}$ is first projected into *common space* via the transformation by $\mathbf{W}_h^{[m]} \in \mathbb{R}^{h^{[m]} \times r}$ and the projected feature is then classified by applying $\mathbf{W}_w \in \mathbb{R}^{w \times r}$ in disregard of the modalities. We can transfer the knowledge of the m -th modality into learning the common classifier \mathbf{W}_w via $\mathbf{W}_h^{[m]}$.

The dimensionality of the common space is desired to be low for reducing the complexity, which naturally induces the low-rank (bilinear) classifier $\mathbf{W}^{[m]}$ as follows. The multi-modal classifiers $\mathbf{W}^{[m]}$ are vertically concatenated into the large matrix which is decomposed to

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}^{[1]} \\ \vdots \\ \mathbf{W}^{[M]} \end{bmatrix} = \begin{bmatrix} \mathbf{W}_h^{[1]} \\ \vdots \\ \mathbf{W}_h^{[M]} \end{bmatrix} \mathbf{W}_w^\top \in \mathbb{R}^{h \times w}, \quad (38)$$

where $h = \sum_{m=1}^M h^{[m]}$ and \mathbf{W}_w is shared across M classifiers. Thus, the minimization of the ranks of $\mathbf{W}^{[m]}$ ($m = 1, \dots, M$) corresponds to minimize the rank of \mathbf{W} , which defines the following optimization problem as in Sect.3.1:

$$\min_{\{\mathbf{W}^{[m]}, b^{[m]}\}_m} \|\mathbf{W}^{[1]\top}, \dots, \mathbf{W}^{[M]\top}\|_\Sigma \quad (39)$$

$$+ C \sum_{m=1}^M \sum_{i=1}^{n^{[m]}} \max\left[0, 1 - y_i^{[m]} \{\text{tr}(\mathbf{W}^{[m]\top} X_i^{[m]} + b^{[m]})\}\right]$$

is reformulated to

$$\min_{\{\mathbf{W}_h^{[m]}, b^{[m]}\}_m, \mathbf{W}_w} \frac{1}{2} \text{tr}(\mathbf{W}_w \mathbf{W}_w^\top) + \frac{1}{2} \sum_m \text{tr}(\mathbf{W}_h^{[m]\top} \mathbf{W}_h^{[m]})$$

$$+ C \sum_{m=1}^M \sum_{i=1}^{n^{[m]}} \max\left[1 - y_i^{[m]} \{\text{tr}(\mathbf{W}_h^{[m]\top} X_i^{[m]} \mathbf{W}_w) + b^{[m]}\}, 0\right], \quad (40)$$

and we finally obtain

$$\min_{\Sigma_w \succ 0} \frac{1}{2} \text{tr}(\Sigma_w) + \sum_{m=1}^M L^{[m]}(\alpha^{[m]*}(\Sigma_w); \Sigma_w), \quad (41)$$

where

$$L^{[m]}(\alpha; \Sigma_w) = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i^{[m]} y_j^{[m]} K_{ij}^{[m]}(\Sigma_w), \quad (42)$$

$$\alpha^{[m]*}(\Sigma_w) = \arg \max_{\alpha \in \Omega^{[m]}} L_m(\alpha; \Sigma_w), \quad (43)$$

$$\Omega^{[m]} = \{\alpha \mid \forall i, 0 \leq \alpha_i \leq C, \sum_{i=1}^{n^{[m]}} y_i^{[m]} \alpha_i = 0\}, \quad (44)$$

and $K_{ij}^{[m]}(\Sigma_w) = \text{tr}(\Sigma_w X_i^{[m]\top} X_j^{[m]})$ is the i, j -th component of the m -th modal Gram matrix. This form is the same as (14) except that the sub-problems (SVM-QP) are solved for respective modalities. This is a general formulation using multi-modal feature matrices, and in the next section we address the problem for classifying multi-modal *multi-class* feature vectors which results in the same formulation.

5.2 Multi-Class Feature Vectors

Suppose in the m -th modality the feature vectors $x_i^{[m]} \in \mathbb{R}^{h^{[m]}}$ are assigned with the label $l_i^{[m]} \in \{1, \dots, w\}$ of w classes. We first project the feature vectors $x^{[m]}$ into the common space shared across the multiple modalities by using the modality-sensitive affine transformation $\mathbf{W}_h^{[m]\top} x^{[m]} + b^{[m]}$ where $\mathbf{W}_h^{[m]} \in \mathbb{R}^{h^{[m]} \times r}$ is the transformation matrix into the r -dimensional common space and $b^{[m]} \in \mathbb{R}^r$ is the bias vector¹. Then, the projected vectors are finally classified by applying the c -th classifier weight $w_c \in \mathbb{R}^r$ (see Fig. 3) as

¹ The biases in the affine model work so as to coordinate the origins of the projections from the M -modal feature spaces. In practice, we centerize the feature vectors by subtracting the means in order to project them simply by $\mathbf{W}_h^{[m]\top} x^{[m]}$.

$$\hat{l}^{[m]} = \arg \max_c \mathbf{w}_c^\top (\mathbf{W}_h^{[m]\top} \mathbf{x}^{[m]} + \mathbf{b}^{[m]}) + b_c \quad (45)$$

$$= \arg \max_c \mathbf{w}_c^\top (\mathbf{W}_h^{[m]\top} \mathbf{x}^{[m]}) + b_c^{[m]}, \quad (46)$$

where $b_c^{[m]} = \mathbf{w}_c^\top \mathbf{b}^{[m]} + b_c$ as a bias for the c -th class in the m -th modality.

The classifier weights \mathbf{w}_c are horizontally concatenated into $\mathbf{W}_w = [\mathbf{w}_1, \dots, \mathbf{w}_w]^\top \in \mathbb{R}^{w \times r}$, while the feature vector $\mathbf{x}^{[m]}$ is augmented into the feature matrix $\mathbf{X}_c^{[m]} \in \mathbb{R}^{h^{[m]} \times w}$ in which only the c -th column contains $\mathbf{x}^{[m]}$ and the other columns are zeros. By using these notations, we can rewrite the c -th classifier for the m -th modality into

$$\mathbf{w}_c^\top (\mathbf{W}_h^{[m]\top} \mathbf{x}^{[m]}) + b_c^{[m]} = \text{tr}(\mathbf{W}_h^{[m]\top} \mathbf{X}_c^{[m]} \mathbf{W}_w) + b_c^{[m]}. \quad (47)$$

This corresponds to the bilinear model (1) except that the feature matrix $\mathbf{X}_c^{[m]}$ is augmented from the feature vector $\mathbf{x}^{[m]}$ for respective classes to cope with w class problems. As described in the previous section, the knowledge can be transferred via the modality-sensitive transformation $\mathbf{W}_h^{[m]}$ from the m -th modality into the common space for training the classifier \mathbf{W}_w .

The dimensionality of the common space into which the feature vector is first projected is required to be small for adequately reducing the complexity of the classifier. In addition, as is the case with the conventional multi-class SVM, we train the multi-class classifiers based on the one-vs-rest approach. These enable us to formulate the following optimization problem for training the classifier \mathbf{W}_w as well as the transformations $\mathbf{W}_h^{[m]}$ in a manner similar to (40):

$$\begin{aligned} & \min_{\{\mathbf{W}_h^{[m]}\}_m, \mathbf{W}_w, \{b_c^{[m]}\}_m} \frac{1}{2} \text{tr}(\mathbf{W}_w \mathbf{W}_w^\top) + \frac{1}{2} \sum_m \text{tr}(\mathbf{W}_h^{[m]\top} \mathbf{W}_h^{[m]}) \\ & + C \sum_{m=1}^M \sum_{c=1}^w \sum_{i=1}^{n^{[m]}} \max[1 - y_{i,c}^{[m]} \{\text{tr}(\mathbf{W}_h^{[m]\top} \mathbf{X}_{i,c}^{[m]} \mathbf{W}_w) + b_c^{[m]}\}, 0], \end{aligned} \quad (48)$$

$$\text{where } y_{i,c}^{[m]} = \begin{cases} +1 & (c = l_i^{[m]}) \\ -1 & (c \neq l_i^{[m]}) \end{cases}, \quad (49)$$

and the feature vector $\mathbf{x}_i^{[m]}$ appears w times in the form of $\mathbf{X}_{i,c}^{[m]}$ due to the one-vs-rest approach for the multi-class problem. By introducing $\Sigma_w = \mathbf{W}_w \mathbf{W}_w^\top$, the above formulation finally results in

$$\min_{\Sigma_w \succcurlyeq 2} \frac{1}{2} \text{tr}(\Sigma_w) + \sum_{m=1}^M L^{[m]}(\boldsymbol{\alpha}^{[m]*}(\Sigma_w); \Sigma_w), \quad (50)$$

where

$$L^{[m]}(\boldsymbol{\alpha}; \Sigma_w) = \sum_{c=1}^w \sum_{i=1}^{n^{[m]}} \alpha_{i,c} \quad (51)$$

$$- \frac{1}{2} \sum_{c,d} \sum_{i,j} \alpha_{i,c} \alpha_{j,d} y_{i,c}^{[m]} y_{j,d}^{[m]} K_{ij,cd}^{[m]}(\Sigma_w),$$

$$\boldsymbol{\alpha}^{[m]*}(\Sigma_w) = \arg \max_{\boldsymbol{\alpha} \in \Omega^{[m]}} L^{[m]}(\boldsymbol{\alpha}; \Sigma_w), \quad (52)$$

$$\Omega^{[m]} = \{\boldsymbol{\alpha} \mid \forall i, c, \quad 0 \leq \alpha_{i,c} \leq C, \quad \forall c, \sum_i y_{i,c}^{[m]} \alpha_{i,c} = 0\}, \quad (53)$$

and $K_{ij,cd}^{[m]}(\Sigma_w)$ is the i, j -th component of the Gram matrix $\mathbf{K}_{cd}^{[m]}(\Sigma_w) = [\Sigma_w]_{cd} \mathbf{K}^{[m]}$, and $\mathbf{K}^{[m]}$ is the kernel Gram matrix of the m -th modal feature vectors and $[\Sigma_w]_{cd}$ is the c, d -th component of the matrix Σ_w . Note that via this optimization, the dimensionality of the common space, *i.e.*, the rank r , is automatically determined so as to be favorably low for classification, and it is less than the number of classes w as in Fisher discriminant analysis (Duda et al. 2001).

By projecting the feature vectors into the common space, the characteristics specific to the modalities are suppressed, while enhancing the discriminative information. Thus, the proposed cross-modal method is applicable not only to the classification for w classes but also to the comparison across the modalities in the common space, which is empirically shown in the experiments (Sect.6.3); namely, the features from multiple modalities in the same class would be projected on close positions in that common space as shown in Fig. 3.

5.2.1 Optimization of Sub-Problem

The sub-problem (52) is slightly different from the previous one (15) in that it includes multiple equation constraints regarding respective classes in (53). It can also be efficiently optimized by applying the off-the-shelf SVM-SMO solver in the following iterative manner. We decompose the QP (52) into *class-wise* blocks²; we focus on the class-wise (dual) variables and labels $\{\alpha_{i,c}, y_{i,c}\}_{i=1, \dots, n}$, and the optimization with respect to the c -th class while fixing the others is given by

$$\boldsymbol{\alpha}_c = \arg \max_{\boldsymbol{\alpha} \in \Omega} L_c(\boldsymbol{\alpha}; \Sigma_w), \quad (54)$$

$$\begin{aligned} L_c(\boldsymbol{\alpha}; \Sigma_w) = & \sum_i \alpha_i \{1 - y_{i,c} \sum_{d \neq c} \sum_j [\Sigma_w]_{cd} K_{ij} y_{j,d} \alpha_{j,d}\} \\ & - \frac{1}{2} \sum_{ij} \alpha_i \alpha_j y_{i,c} y_{j,c} [\Sigma_w]_{cc} K_{ij}, \end{aligned} \quad (55)$$

$$\Omega_c = \{\boldsymbol{\alpha} \mid \forall i, \quad 0 \leq \alpha_i \leq C, \quad \sum_i y_{i,c} \alpha_i = 0\}. \quad (56)$$

² Hereafter, we omit the superscript $^{[m]}$ for simplicity.

The off-the-shelf SMO³ is applicable to iteratively optimize w -class variables $\{\alpha_c\}_{c=1,\dots,w}$ until convergence; it is regarded as sequential minimal optimization with respect to *class-wise blocks*. Thus, it should be noted that the following gap derived from KKT (Fan et al. 2005) is employed for the stopping criterion and the selection of the block to be optimized in (54);

$$\Delta_c = \max_{i \in I_+(\alpha)} \{-y_{i,c} \nabla_{\alpha_i} L_c(\alpha)\} - \min_{i \in I_-(\alpha)} \{-y_{i,c} \nabla_{\alpha_i} L_c(\alpha)\} \tag{57}$$

where $I_+(\alpha) = \{i | \alpha_i < C, y_{i,c} = +1 \vee \alpha_i > 0, y_{i,c} = -1\}$,
 $I_-(\alpha) = \{i | \alpha_i < C, y_{i,c} = -1 \vee \alpha_i > 0, y_{i,c} = 1\}$.

The optimization converges if $\max_c \Delta_c < \epsilon$, and the class-wise block to be updated by (54) is selected as $c = \arg \max_d \Delta_d$ to efficiently minimize the gap Δ_c , $c = \{1, \dots, w\}$.

6 Experimental Results

We applied the proposed methods to a variety of visual classification problems using feature arrays and co-occurrence features for bilinear classification (Sect. 3) as well as using multiple kernels for heterogeneous MKL (Sect. 4) and cross-modal features for bilinear cross-modal learning (Sect. 5).

6.1 Bilinear Classification

For evaluating the performances, we compared the proposed bilinear classifier (Sect. 3) with linear SVM (Vapnik 1998) applied to the concatenated feature vectors and bilinear SVM (Pirsiavash et al. 2009) directly dealing with the feature matrices. The bilinear SVM proposed by Pirsiavash et al. (2009) includes a rank parameter for $\text{rank}(\mathbf{W}) \leq k$ which is determined based on two-fold cross validations from $k \in \{5, 10\}$ as in the paper (Pirsiavash et al. 2009). In the case of multi-class problems, the one-vs-rest approach is employed. All the methods were implemented by MATLAB with libsvm (Chang and Lin 2001) on 3.33 GHz PC.

6.1.1 Toy Example

First, by using toy data, we intuitively demonstrate how the proposed method works on feature matrices. Two types of binary images (100×100) are provided for two-class feature matrices ($\mathbf{X} \in \mathbb{R}^{100 \times 100}$), as shown in Fig. 4; the images have basically one rank with salt and pepper noise of size 5×5 , and there are 100 samples in each type (class). The

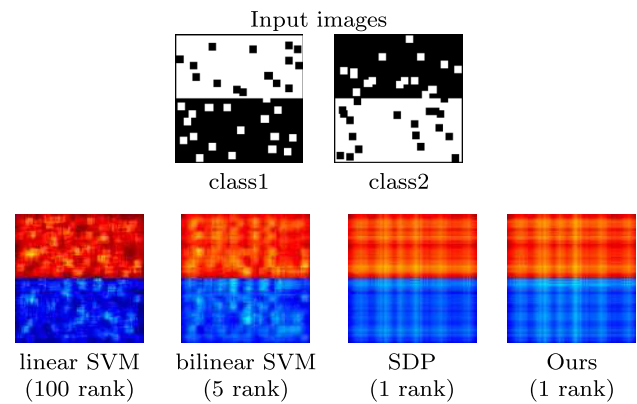


Fig. 4 Classifier weights on toy data. Negative and positive weights are shown by pseudo colors from blue to red. The numbers of the classifier ranks are shown in the parentheses. This figure is best viewed in color


obtained classifier (\mathbf{W}) is shown in Fig. 4, compared to those by the other methods including SDP (4) which is feasible in such a small dataset by using SeDuMi solver. In the linear SVM, the concatenated linear classifier (vector) is folded into the intrinsic matrix form and its rank is also measured. The proposed method favorably produces the one-rank classifier which is the same as the global optimum one by SDP, while the classifiers by the linear SVM (Vapnik 1998) and the bilinear SVM (Pirsiavash et al. 2009) are overly fitted to the data with higher (or even full) rank. Without any hard constraint on the rank, the proposed method recovers the essential rank in the data with a low computational cost (0.4sec) which is much faster than the exhaustive SDP method (14.0sec). The computation time comparison to the bilinear SVM (Pirsiavash et al. 2009) is conducted in Sect. 6.1.4 using the larger-scale datasets than this toy set in order to show the substantial improvements.

6.1.2 Feature Array

Next, we conducted the practical experiments on motion classification using RWC gesture dataset (Kobayashi and Otsu 2009) and image classification using INRIA person dataset (Dalal and Triggs 2005). In these experiments, the array of the (column) feature vectors extracted at temporal/spatial points is formed into the feature matrix \mathbf{X} as shown in Fig. 1ab.

RWC Gesture Dataset: It contains 17 types of human gesture, each of which is performed four times by 48 subjects (23 men and 25 women), as shown in Fig. 5. We extract 751-dimensional CHLAC motion feature vector (Kobayashi and Otsu 2009) at every frame with multiple correlation intervals $\Delta r \in \{1, 3, 5\}$; for details of this feature, refer to (Kobayashi and Otsu 2009). Since the numbers of frames are different across the motion image sequences, we subsample those frame-based features by bilinear interpolation into 50-frame

³ The linear term w.r.t α in the SVM-QP is required to be slightly modified for (55).



Class	17 types of gestures
Subject	48 persons (23 men and 25 women)
Sample	3,264 videos (= 17 × 48 × 4)
Feature matrix	$\mathbb{R}^{751 \times 50}$
Evaluation	3-fold CV

Fig. 5 RWC gesture dataset

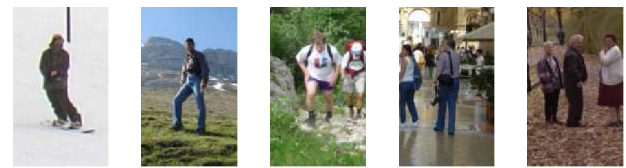
Table 1 Classification performances on RWC dataset

Method	Rank	Err. (%)
SVM	50	2.11
bilinear SVM	5.59	1.41
Ours	3.27	0.98
Ours (smooth)	3.25	1.01
Kobayashi and Otsu (2012b)	–	1.9
Kobayashi and Otsu (2009)	–	4.14

The performance results of the proposed methods are highlighted by bold font

feature vectors, resulting in the feature matrix $X \in \mathbb{R}^{751 \times 50}$ of feature-vs-time (Fig. 1a). The performance is evaluated by three-fold cross validation, and both the error rates and the ranks of the classifiers averaged across classes are shown in Table 1. The proposed method produces the favorable performance compared to the other methods of linear SVM and bilinear SVM (Pirsiavash et al. 2009) and even to the prior works (Kobayashi and Otsu 2009, 2012b); we show the performances reported in those papers (Kobayashi and Otsu 2009, 2012b). While the rank of the SVM classifier is 50, the proposed classifier has around only three rank, improving the efficiency of information compression as well as the generalization performance. In this case, the smoothed classifier (Sect.3.3) that imposes the smoothing regularization on the temporal weight W_w produces slightly inferior performance. This is because the feature matrices are already smoothed by the subsampling procedure and such regularization leads to over smoothing slightly degrading the performance.

INRIA Person Dataset: We used 2,416 person and 12,180 person-free images (64×128) for training, and 1,132 person and 13,590 person-free images for test as shown in Fig. 6. The image is divided into 4×8 subregions and 324-dimensional GLAC feature vectors proposed by Kobayashi and Otsu (2008) are extracted at each region with the same parameter settings as in that paper. Thereby, we obtain the feature matrix $X \in \mathbb{R}^{324 \times 32}$ of feature-vs-space (positions) (Fig. 1b). The performance results are shown in Table 2. In the proposed method, the smoothed classifier that



Class	2 categories (person vs. non-person)
Sample	training 2,416 images for person 12,180 images for non-person
	test 1,132 images for person 13,590 images for non-person
Feature matrix	$\mathbb{R}^{324 \times 32}$

Fig. 6 INRIA person dataset

Table 2 Classification performances on INRIA dataset

Method	Rank	EER (%)
SVM	32	0.55
bilinear SVM	10	0.71
Ours	12	0.62
Ours (smooth)	12	0.53
Kobayashi and Otsu (2008)	–	0.58
Dalal and Triggs (2005)	–	2.25

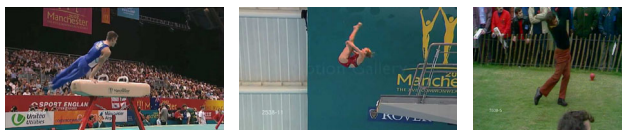
The performance results of the proposed methods are highlighted by bold font

imposes the regularization on the (two-way) spatial positions for W_w slightly improves the performance. The performance is comparable to SVM and is superior to the prior works (Kobayashi and Otsu 2008; Dalal and Triggs 2005). Note that the proposed classifier of low rank (= 12) reduces the computational cost in the detection stage since our low-rank classifier is decomposed into a few separable filters of $O(\text{rank}(W) \times \max(h, w))$, compared to $O(hw)$ in the full rank SVM classifier.

6.1.3 Co-occurrence Feature

The bilinear classifier can directly deal with the co-occurrence features which are inherently formed as a matrix (Fig. 1c). In the framework of bag-of-features (Csurka et al. 2004), the local features, such as SIFT descriptors (Lowe 2004), are assigned with (visual) words via clustering, and the simple occurrence of the words are counted to produce the final histogram-based feature vector. The occurrence features are extended to the co-occurrence features of the neighboring words in the bag-of-features framework (Ling and Soatto 2007).

For motion recognition, we follow the framework employed by Kobayashi and Otsu (2012b) which extracts frame-based motion features at every 10 frames and applies k -means to cluster the features into motion words in Fisher discriminant space; refer to (Kobayashi and Otsu 2012b) for more details. In these experiments, we count the



Class	9 types of actions
Subject	about 17 players in each action
Sample	300 videos (= 150 × 2 (mirrored))
Feature matrix	$\mathbb{R}^{540 \times 270}$
Evaluation	3-fold CV

Fig. 7 UCF sport action dataset

co-occurrence of the motion words along the time axis, as follows. Given w motion words, the frame-based feature at time t with the correlation scale $\Delta r \in \{3, 6, 9\}$ is assigned with multiple motion words, the (voting) weights of which form a vector $f(t) \in \mathbb{R}^{3w}$ in total (Kobayashi and Otsu 2012b). The co-occurrence features are extracted by

$$F(\Delta t) = \sum_t f(t)f(t + \Delta t)^\top, \tag{58}$$

where Δt denotes the interval along the t axis, say $\Delta t \in \{0, 20\}$ in these experiments, and those features are concatenated into the final feature matrix $X = [F(0), F(20)]^\top \in \mathbb{R}^{6w \times 3w}$. The number of words is simply determined by $w = 10 \times \text{\#class}$.

We conducted the motion classification experiments by using UCF sport action dataset (Rodriguez et al. 2008) and Cambridge hand gesture dataset (Kim et al. 2007). For the smoothed classifier (Sect.3.3), the regularization matrices L_h, L_w are set as the graph Laplacian (Belkin and Niyogi 2003) for which the pair-wise similarities between words are simply measured based on the 10 nearest neighbors of word centers.

UCF Sport Action Dataset: This dataset contains nine types of sport actions, each of which is performed by about 17 players, and the total number of sequence is 150, as shown in Fig. 7. To enlarge the training size, the number of training samples are doubled by adding horizontally mirrored video clips. For evaluation, three-fold cross validation is applied and the averaged error rates across action classes are reported in Table 3. The proposed methods exhibit superior performances to the other methods, and in particular, the smoothed bilinear classifier improves the performance with quite low rank. Then, we evaluate the robustness of the methods against the numbers of the (motion) words, to which the size of the feature matrix is proportional. Figure 8 shows the performances by increasing the number of words. In the proposed method, due to the appropriate low rank, the performances are stably high even for larger number of words, while the other methods degrade their performances.

Cambridge Hand Gesture Dataset: There are nine types of hand gestures defined by three primitive hand shapes and

Table 3 Classification performances on UCF dataset

Method	Rank	Err. (%)
SVM	246.67	28.80
bilinear SVM	9.63	27.87
Ours	1.15	24.19
Ours (smooth)	1.11	23.73

The performance results of the proposed methods are highlighted by bold font

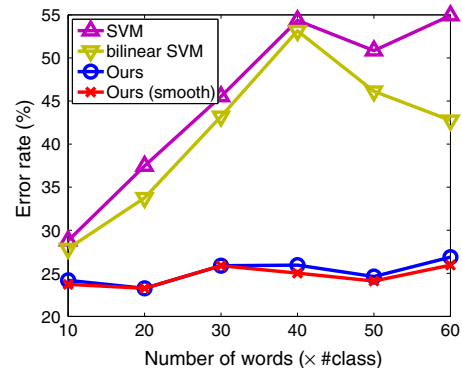



Fig. 8 Performances for various numbers of words on UCF dataset



Class	9 types of gestures
Sample	training 180 videos (= 9 × 2 × 1 × 10) test 720 videos (= 9 × 2 × 4 × 10)
Feature matrix	$\mathbb{R}^{540 \times 270}$

Fig. 9 Cambridge hand gesture dataset

three primitive motions, which are performed ten times by two subjects under five different illumination conditions, as shown in Fig. 9. We used the sequences acquired under the plain illumination condition for training and those under the remaining four conditions for test. The averaged error rates across all gesture classes over the four test conditions are reported in Table 4. The proposed method produces superior performances to the others including the prior works (Kobayashi and Otsu 2012b; Kim et al. 2007); the performances reported in those papers are shown for comparison. Especially, the smoothed classifier is the most favorable. Figure 10 also shows the performances on various numbers of words, demonstrating the robustness of the proposed method as is the case with the UCF dataset.

These experimental results show that the proposed method robustly produces high performances, requiring users only to set sufficiently large number of words without carefully tuning it nor the rank of the classifier for classifying the co-occurrence features.

Table 4 Classification performances on Cambridge dataset

Method	Rank	Err. (%)
SVM	251	14.03
bilinear SVM	10	14.17
Ours	1.22	9.17
Ours (smooth)	1.33	9.03
Kobayashi and Otsu (2012b)	–	11
Kim et al. (2007)	–	18

The performance results of the proposed methods are highlighted by bold font

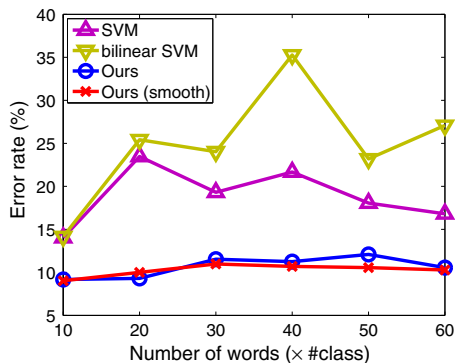


Fig. 10 Performances for various numbers of words on Cambridge dataset

Table 5 Computation time (sec) in training bilinear classifiers for all classes on various datasets

Dataset	RWC	INRIA	UCF	Cambridge
Ours	226	1659	29	35
Ours (smooth)	227	1913	20	39
Bilinear SVM	563	3565	203	52

The performance results of the proposed methods are highlighted by bold font

6.1.4 Computation Time

Finally, we compared the proposed method to the bilinear SVM (Pirsiavash et al. 2009) in terms of the computation time for training the bilinear classifiers on all classes. The comparison results on the datasets that we used in the above experiments are shown in Table 5. The proposed method exhibits faster computation time (roughly twice) than the bilinear SVM, demonstrating that the proposed optimization approach in Sect.3.2 efficiently works.

6.2 Heterogeneous Multiple Kernel Learning

Next, we evaluate the extension of the bilinear model, heterogeneous multiple kernel learning (Sect.4), with comparison to the other MKL methods: simpleMKL (Rakotomamonjy et al. 2008) and the method by Varma and

Ray (2007). It should be noted again that the proposed method dealing with only diagonal matrix of $R^{ij} = \text{diag}\{k_1(\mathbf{x}_{i1}, \mathbf{x}_{j1}), \dots, k_w(\mathbf{x}_{iw}, \mathbf{x}_{jw})\}$ ($Z = I$ in (26)) reduces to the MKL method of Varma and Ray (2007) and it is denoted by “diagonal”.

To demonstrate the effectiveness of the proposed kernel (28), we conducted the comparative experiment using PASCAL VOC 2007 dataset (Everingham et al. 2007) which contains 5,011 images for training and 4,952 images for test in 20 object categories. We used 15 types of pre-computed features provided on the web⁴ of Guillaumin et al. (2010) (for details of the features, refer to that paper), and applied the RBF kernels to those features as $k_c(\mathbf{x}_{ic}, \mathbf{x}_{jc}) = \exp(-\|\mathbf{x}_{ic} - \mathbf{x}_{jc}\|^2/\gamma)$ where γ is the mean of pairwise distances.

The following alternative forms to the proposed kernel (28) are conceivable:

$$\begin{aligned}
 \text{Product: } Z &= [\Phi_1^\top, \dots, \Phi_w^\top] \\
 &\Rightarrow R_{cd}^{ij} = \mathbf{k}_{ic}^\top \mathbf{k}_{jd}, \tag{59} \\
 \text{PCA: } Z &= [V_1^\top, \dots, V_w^\top] \\
 &\Rightarrow R_{cd}^{ij} = \mathbf{k}_{ic}^\top U_c \Lambda_c^{-1} \Lambda_d^{-1} U_d^\top \mathbf{k}_{jd}, \tag{60} \\
 \text{Inverse: } Z &= [U_1 \Lambda_1^{-1} V_1^\top, \dots, U_w \Lambda_w^{-1} V_w^\top] \\
 &\Rightarrow R_{cd}^{ij} = \mathbf{k}_{ic}^\top \mathbf{K}_c^{-1} \mathbf{K}_d^{-1} \mathbf{k}_{jd}. \tag{61}
 \end{aligned}$$

Compared to (27, 28), the PCA kernel (60) loses the CCA rotation matrix derived from U_c , while the inverse kernel (61) additionally introduces the whitening by Λ_c^{-1} to normalize the magnitudes (norms) of features.

Table 6a shows the mean of average precision rates (mAP) across the categories. The proposed kernel (28) is superior to the other types of kernels. The inverse kernel is the worst since it cancels out the kernel function by applying the inverse of the Gram matrix. The difference between the proposed kernel (28) and the inverse kernel (61) is small in terms of only Λ_c , but the effectiveness of the kernel is completely different; the proposed kernel significantly outperforms. The PCA kernel is somewhat effective due to that it contains the same diagonal component (intra kernel) as in the proposed kernel, though the performance is inferior to the proposed kernel. These results demonstrate that both the rotation matrix by CCA and the magnitude preservation by PCA in the proposed kernel are effective for classifications. Table 6b shows that the proposed method is favorably competitive with the other MKL methods. The “diagonal” method uses only the intra kernels which are the same as the diagonal in the PCA kernel, and the performance is similar to that of PCA kernel in Table 6a.

⁴ <http://lear.inrialpes.fr/people/guillaumin/data.php>.

Table 6 Classification performances on PASCAL VOC 2007 dataset

	mAP (%)
(a) Kernel types	
Product	42.30
PCA	45.60
Inverse	37.88
Proposed	48.64
(b) Comparison	
simpleMKL	48.04
diagonal	45.83
Ours	48.64

The performance results of the proposed methods are highlighted by bold font



Class	17 flower categories
Sample	1,360 images (= 17 × 80)
Kernel	7 types
Evaluation	3-fold CV

Fig. 11 Oxford flower dataset

We further conducted the MKL experiments using Oxford flower dataset (Nilsback and Zisserman 2006), Butterfly dataset (Lazebnik et al. 2004) and Bird dataset (Lazebnik et al. 2005). We used the RBF kernel of respective types of the features (distances) in the same manner as described above.

Oxford Flower Dataset: It is composed of 80 images of 17 flower categories as shown in Fig. 11. We used seven types of precomputed pairwise distances provided in the web⁵ of Nilsback and Zisserman (2008); for the details of the distances, refer to (Nilsback and Zisserman 2006, 2008). Table 7 shows the performance results by three-fold cross validations using the same predefined splits as in (Nilsback and Zisserman 2006). For comparison, the performance reported by Gehler and Nowozin (2009) who use the same features and cross validation splits is also shown.

Butterfly Dataset: There are 619 images of seven butterfly classes as shown in Fig. 12. We used the three types of precomputed pairwise distances provided on the web⁶ of Christoudias et al. (2010); for the details of the distances, refer to that paper. The classification accuracies are evaluated by three-fold cross validations and are shown in Table 8.

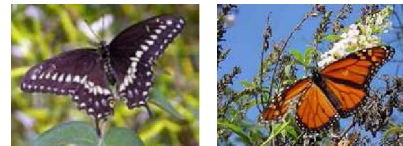
⁵ <http://www.robots.ox.ac.uk/~vgg/research/flowers/index.html>.

⁶ <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/shape/msorec/>.

Table 7 Classification performances on Oxford flower dataset

Method	Acc. (%)
simpleMKL	81.08
simpleMKL (multiclass)	82.55
diagonal	85.10
Ours	86.47
Gehler and Nowozin (2009)	85.5

The performance results of the proposed methods are highlighted by bold font



Class	7 butterfly categories
Sample	619 images
Kernel	3 types
Evaluation	3-fold CV

Fig. 12 Butterfly dataset

Table 8 Classification performances on Butterfly dataset

Method	Acc. (%)
simpleMKL	69.60
simpleMKL (multiclass)	67.95
diagonal	74.42
Ours	76.54


The performance results of the proposed methods are highlighted by bold font

Bird Dataset: The dataset contains six bird classes with 100 images per class as shown in Fig. 13. For multiple kernels, we also used the three types of precomputed pairwise distances which are the same as in Butterfly dataset. Table 9 shows the performances evaluated by three-fold cross validations.

In those three datasets, the proposed method produces superior performances to the others. Especially, in comparison to “diagonal”, we can see that the inter kernels between heterogeneous RKHSs effectively contribute to improve the performances. As a result, the proposed heterogeneous MKL method, which combines both intra kernels (diagonal) and inter kernels (off-diagonal) via the bilinear model, is competitive to the other MKL methods (Rakotomamonjy et al. 2008; Varma and Ray 2007).

6.3 Bilinear Cross-Modal Learning

At the last, we test the bilinear cross-modal learning (Sect.5) which is the extension of the bilinear model. In this



Class	6 bird categories
Sample	600 images ($= 6 \times 100$)
Kernel	3 types
Evaluation	3-fold CV

Fig. 13 Bird dataset

Table 9 Classification performances on Bird dataset

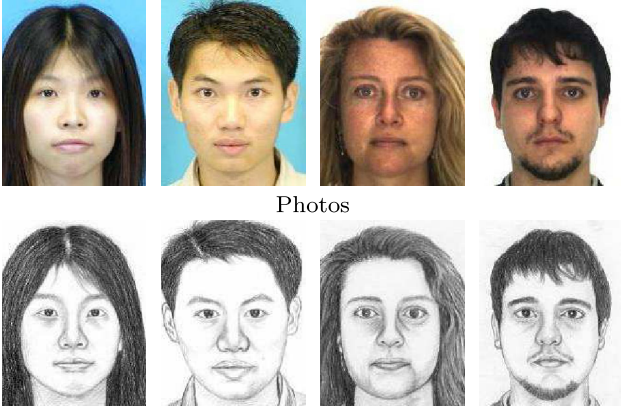
Method	Acc. (%)
simpleMKL	69.21
simpleMKL (multiclass)	68.71
diagonal	72.85
Ours	74.50

The performance results of the proposed methods are highlighted by bold font

cross-modal learning, the way for evaluating classification performances is different from those employed in the previous sections. The common space that multi-modal features are mapped into is first learned by using training samples. Then, the multi-modal test samples of the novel classes unknown in the training are classified by 1-NN based on the distances between samples in the common space. For the NN-based classification, we partition into gallery and probe sets the modalities from which the test samples are drawn; namely, the samples from the probe modalities are classified according to the nearest samples of the gallery modalities in the common space. Thus, note that the training samples are used only for constructing/learning the common space, not for the classification. According to such standard evaluation protocol, we utilize only the transformation $W_h^{[m]}$ optimized by the proposed method (Sect.5.2) for mapping the feature vectors into the common space, without applying the classifier W_w . We evaluate the performances of the proposed method on two datasets of CUHK Face Sketch dataset (Wang and Tang 2009) and Animals with Attributes (AWA) dataset (Lampert et al. 2009).

6.3.1 CUHK Face Sketch Dataset

This dataset includes face images of 188 subjects from CUHK student as well as 123 subjects from the AR dataset (Martínez and Benavente 1998). Those faces are depicted by *photos* and *sketches* drawn by an artist based on a photo taken in a frontal pose as shown in Fig. 14. Those two types of images are cropped and resized into 62×80 pixels which are directly employed for the two-modal image feature vectors via L_2 normalization. We randomly pick



Modality	2 types (photo and sketch)	
Class	training	150 subjects
	test	161 subjects
Sample	training	150 photos and 150 sketches
	test	161 photos and 161 sketches
Evaluation	3 random splits	

Fig. 14 CUHK Face Sketch dataset

up the training 150 subjects to provide 300 training samples of 150 classes in two modalities (photo and sketch) which are fed into the cross-modal learning. The remaining 322 samples of 161 subjects which are unknown during the training are classified in the learnt common space; 161 photo images (probe set) are classified by applying 1-NN over 161 sketch images (gallery set), and vice versa. This is repeated three times and the averaged performances are reported.

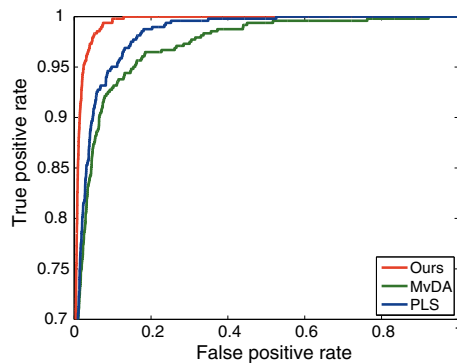
The proposed method is compared to the PLS-based method proposed by Sharma and Jacobs (2011) as well as the multi-view discriminant analysis (MvDA) proposed by Kan et al. (2012). Note that the PLS method relies on the one-to-one correspondence between the two modalities of photos and sketches, while the MvDA exploits the class information as in the proposed method. In those methods, we set the dimensionality of the common space as 49 dimensions for PLS method and 53 dimensions for MvDA, which are determined empirically so as to produce the best results. On the other hand, the proposed method automatically renders the optimal dimensions of the common space, and in this experiment we obtain 44-dimensional space on an average.

The performance results are shown in Table 10. While the PLS method produces the superior performance in the classification of photos, the proposed method is superior in the sketch classification and produces favorable averaged performance. Those classifications are performed based on the pair-wise distances between the photo and sketch samples, forming the 161×161 distance matrix in the test. Those pair-wise distances are also assigned with the labels that indicate

Table 10 1-Rank classification accuracies (%) on CUHK dataset

Method	Gallery: <i>sketch</i> Probe: <i>photo</i>	Gallery: <i>photo</i> Probe: <i>sketch</i>	Mean
PLS	90.06	81.37	85.72
MvDA	82.40	80.95	81.68
Ours	85.30	86.34	85.82

The performance results of the proposed methods are highlighted by bold font

**Fig. 15** ROC for pair-wise distances between *photo* and *sketch* samples on CUHK dataset

whether the pair of the samples belongs to the same class (+1) or not (-1). Thereby, we can show the ROC of those distances for analyzing the further details of the learnt common space. The ROC shown in Fig. 15 indicates that the proposed method provides discriminative metrics; in the common space optimized by the proposed method, the samples of the same class (subject) are mapped into the close points, while the samples in the different classes are far away. We also measured the k -rank performances⁷ as shown in Fig. 16, whereas the performances in Table 10 are evaluated in 1-rank classification. The proposed method is superior on an average to the other methods in accordance with the ROC (Fig. 15).

6.3.2 Animals with Attributes (AWA) Dataset

It is composed of 30,475 images in 50 animal classes, and each animal class is also described by using 85 attributes, resulting in the 50×85 class-attribute matrix. Consequently, there are two modalities of *images* and *attributes*; each animal category contains only *one* attribute vector and a number of images (Fig. 17).

In this experiment, we apply the non-linear kernel $\mathbf{K}^{[m]}$ in (51). For describing the images, we used the precomputed six types of image features provided on the web⁸ of Lampert et

al. (2009) and employed sum of χ^2 -kernels computed from respective types of features; for the details, refer to (Lampert et al. 2009). On the other hand, the attributes are given by Osherson et al. (1991) who collected judgments from human subjects on the *relative strength of association* between the 85 attributes and the animals. They are basically represented by quantitative values, and we normalize those attribute values in unit L_1 -norm, though Lampert et al. (2009) transform them into binary values to which the (kernel) SVM can be applied for estimating the presence of each attribute in the image. In this study, we can deal with the quantitative attributes themselves, since the proposed method accepts any types of feature representation, regarding even those attributes as the feature vectors. The Gaussian kernel is then applied to those normalized attribute feature vectors. As a result, the proposed method optimizes the common space by using those two modal features represented in the kernels.

For training, we pick up randomly 10,000 image samples from the predefined 40 training classes, while using 40 attribute vectors correspondingly. Note that in this experiment, two modalities have different number of samples and different feature (kernel) representation. By using those two modal samples, we learn the transformation $\mathbf{W}_h^{[m]}$ ($m = 1, 2$) as well as the classifier \mathbf{W}_w by (50), though \mathbf{W}_w is not actually used for classification. For test on image classification, the 10 attribute samples corresponding to the 10 test classes are mapped into the common space by $\mathbf{W}_h^{[2]}$ to form the gallery samples. On the other hand, the 6,180 image samples belonging to the predefined 10 test classes are also mapped into the common space via $\mathbf{W}_h^{[1]}$ and then are classified based on the gallery samples; the image sample is classified by 1-NN on the gallery (attribute) samples. The above procedure is repeated three times and the average classification accuracy is reported. As in CUHK dataset, the attribute classification is also conducted by alternating the modalities for gallery and probe.

For comparison, we apply multi-view DA (MvDA) (Kan et al. 2012) in the same setting as the proposed method, and the method proposed by Lampert et al. (2009) using binary attributes and the same kernel-based image features.

Table 11 shows 1-rank classification performances. As is the case with CUHK dataset, the k -rank performances are also shown in Fig. 18a for image classification and Fig. 18b for attribute classification. These results show that the proposed method is superior to the other methods on both two types of classifications. The classification using the gallery of attributes is similar to the standard image classification, since each class has only one gallery sample; but the test classes are not presented in the training, making the standard classification methods inapplicable to this task. On the other hand, the attribute classification utilizing the gallery of images is close to the task of image retrieval that picks up the closest image to the query represented by the attribute.

⁷ The accuracies are measured as follows; the sample of which k -NN contain the gallery sample belonging to the same class is regarded to be correctly classified.

⁸ <http://attributes.kyb.tuebingen.mpg.de/>.

Fig. 16 k -rank classification accuracies on CUHK dataset

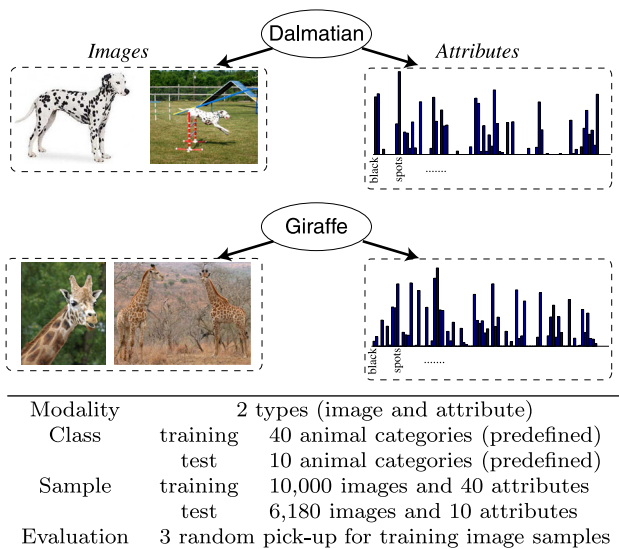
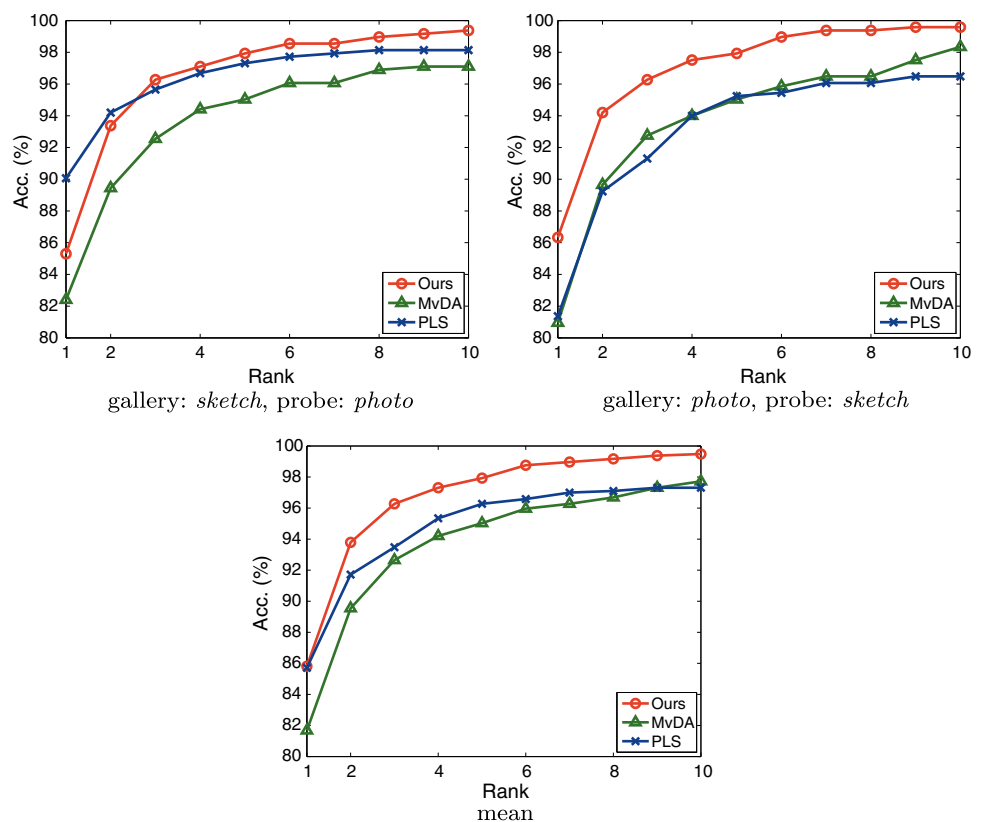


Fig. 17 Animals with Attributes (AWA) dataset

These two kinds of tasks are seamlessly performed by using the common space learned by the proposed method. The dimensionality of the common space is determined automatically in the proposed method; we obtain 40-dimensional space on an average. In contrast, the MvDA requires the dimensionality to be determined in advance and the performance is dependent on it as shown in Fig. 19; in this exper-

Table 11 1-Rank classification accuracies (%) on AWA dataset

Method	Gallery: <i>attribute</i> Probe: <i>image</i>	Gallery: <i>image</i> Probe: <i>attribute</i>
Lampert et al. (2009)	36.76	70.00
MvDA	31.92	60.00
Ours	41.43	70.00

The performance results of the proposed methods are highlighted by bold font

iment, we employ 8-dimensional space to produce the best performance.

7 Conclusions

We have proposed the method to optimize a low-rank bilinear classifier for feature matrices without any hard constraint on the rank. The classifier is optimized by minimizing the trace norm of the classifier matrix, which contributes to the rank reduction. The optimization is formulated in a tractable convex form and it is computationally efficiently optimized by the gradient descent approach. In addition, we propose the two notable extensions of the bilinear model, regarding multiple kernel learning and cross-modal learning.

By introducing non-linear kernel functions into the bilinear method, heterogeneous multiple kernel learning (hMKL) is proposed. The hMKL combines not only the ordinary

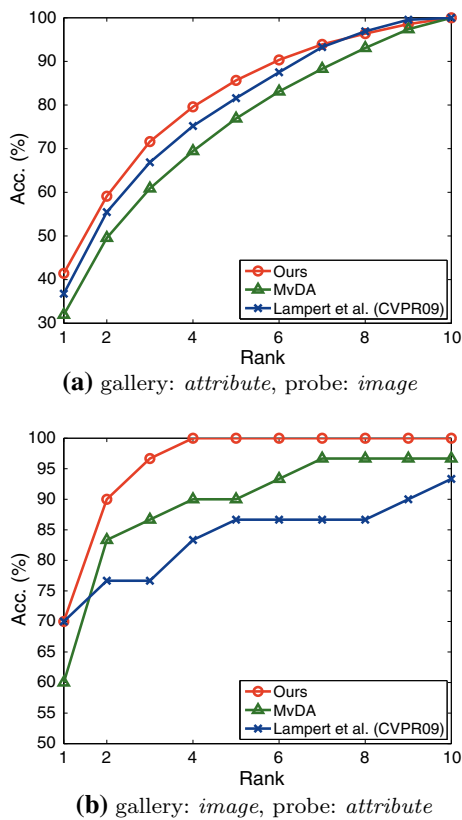


Fig. 18 k -rank classification accuracies on AWA dataset

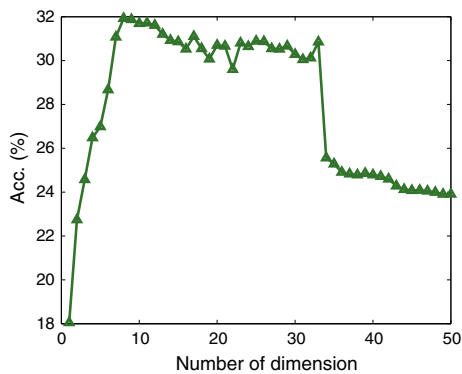


Fig. 19 Image classification accuracies on AWA dataset by various numbers of dimensionality in MvDA (Kan et al. 2012)

(intra) kernels in the homogeneous RKHS features but also the inter kernels between heterogeneous RKHS features into a new composite kernel through the bilinear model.

The cross-modal learning is related to the classification of the features mapped from multiple modalities into the common space shared across those modalities. We formulate in the bilinear model both the mapping and the multi-class classifiers in the common space jointly, and thus apply the proposed bilinear optimization to simultaneously learn both of them.

To evaluate the proposed methods, we conducted the experiments on various visual classification tasks, which are mainly categorized into (1) bilinear classification for feature arrays and co-occurrence feature matrices, (2) multiple kernel learning using multiple kernel functions, and (3) cross-modal learning using multi-modal features in a multi-class setting. The proposed methods exhibited favorable performances compared to the other methods in those tasks.

Appendix: Proof of Convexity in (14)

We prove the convexity of the proposed formulation (14). Since the constraint $\Sigma_w \succcurlyeq 0$, i.e., positive semidefinite cone, is a convex set, the convexity of the proposed optimization problem will be established if the objective cost function J is proven to be convex. Here, we use the following notations:

$$\alpha^*(\Sigma_w) \in \mathbb{R}^n := [\alpha_1^*(\Sigma_w), \dots, \alpha_n^*(\Sigma_w)]^\top, \quad (62)$$

$$\bar{K}(\Sigma_w) \in \mathbb{R}^{n \times n} : \bar{K}_{ij}(\Sigma_w) = y_i y_j K_{ij}(\Sigma_w), \quad (63)$$

and then the objective cost function is rewritten by

$$J(\Sigma_w) = \frac{1}{2} \text{tr}(\Sigma_w) + \mathbf{1}^\top \alpha^*(\Sigma_w) - \frac{1}{2} \alpha^*(\Sigma_w)^\top \bar{K}(\Sigma_w) \alpha^*(\Sigma_w). \quad (64)$$

[Derivative of J] Before proceeding to the proof of the convexity, we first show the form of the derivative of J with respect to Σ_w based on the Lemma 2 in (Chapelle et al. 2002).

The derivative of J with respect to $[\Sigma_w]_{cd}$, the c, d -th component of the matrix Σ_w , is simply obtained by

$$\begin{aligned} \frac{\partial J}{\partial [\Sigma_w]_{cd}} &= \frac{1}{2} [\Sigma_w]_{cd} + \mathbf{1}^\top \frac{\partial \alpha^*(\Sigma_w)}{\partial [\Sigma_w]_{cd}} \\ &\quad - \frac{1}{2} \alpha^*(\Sigma_w)^\top \frac{\partial \bar{K}(\Sigma_w)}{\partial [\Sigma_w]_{cd}} \alpha^*(\Sigma_w) - \alpha^*(\Sigma_w)^\top \bar{K}(\Sigma_w) \frac{\partial \alpha^*(\Sigma_w)}{\partial [\Sigma_w]_{cd}}, \end{aligned} \quad (65)$$

where δ_{cd} is the Kronecker delta and note that $\alpha^*(\Sigma_w)$ is the function of Σ_w .

The Lagrangian function for (15) is given by

$$\begin{aligned} L(\alpha, \lambda, \beta, \gamma) &= \mathbf{1}^\top \alpha - \frac{1}{2} \alpha^\top \bar{K}(\Sigma_w) \alpha \\ &\quad - \lambda \mathbf{y}^\top \alpha + \beta^\top \alpha + \gamma^\top (C\mathbf{1} - \alpha), \end{aligned} \quad (66)$$

where $\lambda \in \mathbb{R}$, $\beta \in \mathbb{R}_+^n$ and $\gamma \in \mathbb{R}_+^n$ are the Lagrange multipliers for the constraints Ω in (9). Thus, the unique

optimizer $\alpha^*(\Sigma_w)$ in (15) satisfies the following conditions:

$$\begin{aligned} \mathbf{1} - \bar{\mathbf{K}}(\Sigma_w)\alpha^*(\Sigma_w) - \lambda^*(\Sigma_w)\mathbf{y} + \beta^*(\Sigma_w) - \gamma^*(\Sigma_w) &= \mathbf{0} \\ \Rightarrow \mathbf{1} - \bar{\mathbf{K}}(\Sigma_w)\alpha^*(\Sigma_w) &= \lambda^*(\Sigma_w)\mathbf{y} - \beta^*(\Sigma_w) + \gamma^*(\Sigma_w), \end{aligned} \quad (67)$$

$$\mathbf{y}^\top \alpha^*(\Sigma_w) = 0 \Rightarrow \mathbf{y}^\top \frac{\partial \alpha^*(\Sigma_w)}{\partial [\Sigma_w]_{cd}} = \mathbf{0}, \quad (68)$$

$$\begin{aligned} \forall i, \beta_i^*(\Sigma_w)\alpha_i^*(\Sigma_w) &= 0 \\ \Rightarrow \beta_i^*(\Sigma_w) \frac{\partial \alpha_i^*(\Sigma_w)}{\partial [\Sigma_w]_{cd}} + \frac{\partial \beta_i^*(\Sigma_w)}{\partial [\Sigma_w]_{cd}} \alpha_i^*(\Sigma_w) &= 0, \end{aligned} \quad (69)$$

$$\begin{aligned} \forall i, \gamma_i^*(\Sigma_w)\{C - \alpha_i^*(\Sigma_w)\} &= 0 \\ \Rightarrow -\gamma_i^*(\Sigma_w) \frac{\partial \alpha_i^*(\Sigma_w)}{\partial [\Sigma_w]_{cd}} + \frac{\partial \gamma_i^*(\Sigma_w)}{\partial [\Sigma_w]_{cd}} \{C - \alpha_i^*(\Sigma_w)\} &= 0, \end{aligned} \quad (70)$$

where $\lambda^*(\Sigma_w)$, $\beta^*(\Sigma_w)$ and $\gamma^*(\Sigma_w)$ are the optimum Lagrange multipliers, depending on the variable Σ_w . Based on the fact that either $\beta_i^*(\Sigma_w) = 0$ or $\alpha_i^*(\Sigma_w) = 0$ in (69) and either $\gamma_i^*(\Sigma_w) = 0$ or $\alpha_i^*(\Sigma_w) = C$ in (70), we can further obtain the following conditions from (69) and (70):

$$\beta_i^*(\Sigma_w) \frac{\partial \alpha_i^*(\Sigma_w)}{\partial [\Sigma_w]_{cd}} = -\frac{\partial \beta_i^*(\Sigma_w)}{\partial [\Sigma_w]_{cd}} \alpha_i^*(\Sigma_w) = 0, \quad (71)$$

$$\gamma_i^*(\Sigma_w) \frac{\partial \alpha_i^*(\Sigma_w)}{\partial [\Sigma_w]_{cd}} = \frac{\partial \gamma_i^*(\Sigma_w)}{\partial [\Sigma_w]_{cd}} (C - \alpha_i^*(\Sigma_w)) = 0. \quad (72)$$

Thus, the derivative (65) results in

$$\begin{aligned} \frac{\partial J}{\partial [\Sigma_w]_{cd}} &= \frac{1}{2} \delta_{ij} - \frac{1}{2} \alpha^*(\Sigma_w)^\top \frac{\partial \bar{\mathbf{K}}(\Sigma_w)}{\partial [\Sigma_w]_{cd}} \alpha^*(\Sigma_w) \\ &+ \left\{ \frac{\partial \alpha^*(\Sigma_w)}{\partial [\Sigma_w]_{cd}} \right\}^\top \{ \mathbf{1} - \bar{\mathbf{K}}(\Sigma_w)\alpha^*(\Sigma_w) \} \end{aligned} \quad (73)$$

$$\begin{aligned} &= \frac{1}{2} \delta_{ij} - \frac{1}{2} \alpha^*(\Sigma_w)^\top \frac{\partial \bar{\mathbf{K}}(\Sigma_w)}{\partial [\Sigma_w]_{cd}} \alpha^*(\Sigma_w) \\ &+ \left\{ \frac{\partial \alpha^*(\Sigma_w)}{\partial [\Sigma_w]_{cd}} \right\}^\top \{ \lambda^*(\Sigma_w)\mathbf{y} - \beta^*(\Sigma_w) + \gamma^*(\Sigma_w) \} \end{aligned} \quad (74)$$

$$= \frac{1}{2} \delta_{ij} - \frac{1}{2} \alpha^*(\Sigma_w)^\top \frac{\partial \bar{\mathbf{K}}(\Sigma_w)}{\partial [\Sigma_w]_{cd}} \alpha^*(\Sigma_w), \quad (75)$$

where we use (67) for transforming (73) into (74), and use (68), (71) and (72) to get (75) from (74).

Finally, the derivative of J with respect to Σ_w is given by

$$\begin{aligned} \nabla J &= \frac{1}{2} \mathbf{I} - \frac{1}{2} \sum_{i,j} \alpha_i^*(\Sigma_w)\alpha_j^*(\Sigma_w) \frac{\partial \bar{K}_{ij}(\Sigma_w)}{\partial \Sigma_w} \\ &= \frac{1}{2} \left\{ \mathbf{I} - \sum_{i,j} \alpha_i^*(\Sigma_w)\alpha_j^*(\Sigma_w) y_i y_j \mathbf{X}_i^\top \mathbf{X}_j \right\}. \end{aligned} \quad (76)$$

[Convexity of J] Then, we prove the convexity of J by verifying the following first order condition:

$$J(\mathbf{B}) \geq J(\mathbf{A}) + \text{tr}\{(\mathbf{B} - \mathbf{A})\nabla J(\mathbf{A})\}, \quad (77)$$

where $\mathbf{A} \succcurlyeq 0$, $\mathbf{B} \succcurlyeq 0$.

Proof From (64), the left-hand side in (77) is written by

$$J(\mathbf{B}) = \frac{1}{2} \text{tr}(\mathbf{B}) + \mathbf{1}^\top \alpha^*(\mathbf{B}) - \frac{1}{2} \alpha^*(\mathbf{B})^\top \bar{\mathbf{K}}(\mathbf{B}) \alpha^*(\mathbf{B}), \quad (78)$$

while by using (76), the right-hand side in (77) results in

$$\begin{aligned} J(\mathbf{A}) + \text{tr}\{(\mathbf{B} - \mathbf{A})\nabla J(\mathbf{A})\} &= \frac{1}{2} \text{tr}(\mathbf{A}) + \mathbf{1}^\top \alpha^*(\mathbf{A}) - \frac{1}{2} \alpha^*(\mathbf{A})^\top \bar{\mathbf{K}}(\mathbf{A}) \alpha^*(\mathbf{A}) \\ &+ \frac{1}{2} \text{tr}(\mathbf{B} - \mathbf{A}) - \frac{1}{2} \sum_{ij} \alpha_i^*(\mathbf{A}) \alpha_j^*(\mathbf{A}) y_i y_j \text{tr}\{(\mathbf{B} - \mathbf{A}) \mathbf{X}_i^\top \mathbf{X}_j\} \\ &= \frac{1}{2} \text{tr}(\mathbf{B}) + \mathbf{1}^\top \alpha^*(\mathbf{A}) - \frac{1}{2} \alpha^*(\mathbf{A})^\top \bar{\mathbf{K}}(\mathbf{B}) \alpha^*(\mathbf{A}). \end{aligned} \quad (79)$$

Since $\alpha^*(\mathbf{B}) = \arg \max_{\alpha \in \Omega} \{ \mathbf{1}^\top \alpha - \frac{1}{2} \alpha^\top \bar{\mathbf{K}}(\mathbf{B}) \alpha \}$ and $\alpha^*(\mathbf{A}) \in \Omega$, it is shown that

$$\begin{aligned} \mathbf{1}^\top \alpha^*(\mathbf{B}) - \frac{1}{2} \alpha^*(\mathbf{B})^\top \bar{\mathbf{K}}(\mathbf{B}) \alpha^*(\mathbf{B}) &\geq \mathbf{1}^\top \alpha^*(\mathbf{A}) - \frac{1}{2} \alpha^*(\mathbf{A})^\top \bar{\mathbf{K}}(\mathbf{B}) \alpha^*(\mathbf{A}). \end{aligned} \quad (80)$$

From (78–80), the inequality (77) holds. \square

References

- Akaho, S. (2001). A kernel method for canonical correlation analysis. In *international meeting on psychometric society (IMPS2001)*.
- Belkin, M., & Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6), 1373–1396.
- Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge: Cambridge University Press.
- Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3), 27:1–27:27.
- Chapelle, O., Vapnik, V., Bousquet, O., & Mukherjee, S. (2002). Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1), 131–159.
- Christoudias, C.M., Urtasun, R., Salzmann, M., Darrell, T. (2010). Learning to recognize objects from unseen modalities. In *European conference on computer vision (ECCV)* (pp. 677–691).
- Csurka, G., Dance, C.R., Fan, L., Willamowski, J., Bray, C. (2004). Visual categorization with bags of keypoints. In: *ECCV workshop on statistical learning in computer vision*, (pp. 1–22).
- Dalal, N., Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 886–893).
- Dempe, S. (2002). *Foundations of bilevel programming*. Dordrecht: Kluwer Academic Publishers.
- Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification* (2nd ed.). New York: Wiley-Interscience.

- Eriksson, A., van den Hengel, A. (2010). Efficient computation of robust low-rank matrix approximations in the presence of missing data using the l_1 norm. In *IEEE conference on computer vision and pattern recognition (CVPR)*, (pp. 771–778).
- Everingham, M., Gool, L. V., Williams, C., Winn, J., & Zisserman, A. (2007). *The PASCAL visual object classes challenge 2007 (VOC2007) results*.
- Fan, R. E., Chen, P. H., & Lin, C. J. (2005). Working set selection using second order information for training support vector machines. *Journal of Machine Learning Research*, 6, 1889–1918.
- Fan, R. E., Chang, K. W., Hsieh, C. J., Wang, X. R., & Lin, C. J. (2008). Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9, 1871–1874.
- Gehler, P., Nowozin, S. (2009). On feature combination for multiclass object classification. In *International conference on computer vision (ICCV)* (pp. 221–228).
- Graepel, T., Herbrich, R., Schölkopf, B., Smola, A., Bartlett, P., Müller, K.R., Obermayer, K., Williamson, R. (1999). Classification on proximity data with lp-machines. In *international conference on artificial neural networks (ICANN)* (pp. 304–309).
- Guillaumin, M., Verbeek, J., Schmid, C. (2010). Multimodal semi-supervised learning for image classification. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 902–909).
- Kan, M., Shan, S., Zhang, H., Lao, S., Chen, X. (2012). Multi-view discriminant analysis. In *European conference on computer vision (ECCV)* (pp. 808–821).
- Kim, T.K., Wong, S.F., Cipolla, R. (2007). Tensor canonical correlation analysis for action classification. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1–8).
- Kobayashi, T., Otsu, N. (2008). Image feature extraction using gradient local auto-correlations. In *European conference on computer vision (ECCV)* (pp. 346–358).
- Kobayashi, T., & Otsu, N. (2009). A three-way auto-correlation based approach to motion recognition. *Pattern Recognition Letters*, 30(3), 185–192.
- Kobayashi, T., Otsu, N. (2012a). Efficient optimization for low-rank integrated bilinear classifiers. In *European conference on computer vision (ECCV)* (pp. 474–487).
- Kobayashi, T., & Otsu, N. (2012b). Motion recognition using local auto-correlation of space-time gradients. *Pattern Recognition Letters*, 33(9), 1188–1195.
- Lampert, C.H., Nickisch, H., Harmeling, S. (2009) Learning to detect unseen object classes by between-class attribute transfer. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 951–958).
- Lanckriet, G. R., Cristianini, N., Bartlett, P., Ghaoui, L. E., & Jordan, M. I. (2004). Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5, 27–72.
- Lazebnik, S., Schmid, C., Ponce, J. (2004). Semi-local affine parts for object recognition. In *British machine vision conference (BMVC)* (pp. 779–788).
- Lazebnik, S., Schmid, C., Ponce, J. (2005). A maximum entropy framework for part-based texture and object recognition. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 832–838).
- Lee, D., & Seung, H. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401, 788–791.
- Ling, H., Soatto, S. (2007). Proximity distribution kernels for geometric context in category recognition. In *International conference on computer vision (ICCV)* (pp. 1–8).
- Loeff, N., Farhadi, A. (2008). Scene discovery by matrix factorization. In *European conference on computer vision (ECCV)* (pp. 451–464).
- Lowe, D. G. (2004). Distinctive image features from scale invariant features. *International Journal of Computer Vision*, 60(2), 91–110.
- Martínez, A., Benavente, R. (1998). *The AR Face Database*. Tech. Rep. 24, Computer Vision Center, Bellatera.
- Nilsback, M.E., Zisserman, A. (2006). A visual vocabulary for flower classification. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1447–1454).
- Nilsback, M.E., Zisserman, A. (2008). Automated flower classification over a large number of classes. In *Indian conference on computer vision, graphics and image processing (ICVGIP)* (pp. 722–729).
- Nocedal, J., & Wright, S. J. (1999). *Numerical optimization*. New York: Springer.
- Osherson, D. N., Stern, J., Wilkie, O., Stob, M., & Smith, E. E. (1991). Default probability. *Cognitive Science*, 15(2), 251–269.
- Pirsiavash, H., Ramanan, D., Fowlkes, C. (2009). Bilinear classifiers for visual recognition. In *Advances in neural information processing systems 22* (pp. 1482–1490).
- Rakotomamonjy, A., Bach, F. R., Canu, S., & Grandvalet, Y. (2008). SimpleMKL. *Journal of Machine Learning Research*, 9, 2491–2521.
- Rennie, J.D., Srebro, N. (2005). Fast maximum margin matrix factorization for collaborative prediction. In *international conference on machine learning (ICML)* (pp. 713–719).
- Rodriguez, M., Ahmed, J., Shah, M. (2008). Action MACH: a spatio-temporal maximum average correlation height filter for action recognition. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1–8).
- Schölkopf, B., & Smola, A. J. (2001). *Learning with kernels*. Cambridge: MIT Press.
- Sharma, A., Jacobs, D.W. (2011). Bypassing synthesis: PLS for face recognition with pose, low-resolution and sketch. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 593–600).
- Smola, A. J., Bartlett, P., Schölkopf, B., & Schuurmans, D. (2000). *Advances in large-margin classifiers*. Cambridge: MIT Press.
- Srebro, N., Rennie, J. D. M., & Jaakkola, T. S. (2005). Maximum-margin matrix factorization. In L. K. Saul, Y. Weiss, & L. Bottou (Eds.), *Advances in neural information processing systems 17* (pp. 1329–1336). Cambridge: MIT Press.
- Tenenbaum, J. B., & Freeman, W. T. (2000). Separating style and content with bilinear models. *Neural Computation*, 12(6), 1247–1283.
- Vapnik, V. N. (1998). *Statistical learning theory*. New York: Wiley.
- Varma, M., Ray, D. (2007). Learning the discriminative power-invariance trade-off. In *international conference on computer vision (ICCV)* (pp. 1–8).
- Wang, X., & Tang, X. (2009). Face photo-sketch synthesis and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11), 1955–1967.
- Wolf, L., Jhuang, H., Hazan, T. (2007). Modeling appearances with low-rank svm. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1–6).
- Yang, J., Zhang, D., Frangi, A. F., & Yang, J. Y. (2004). Two-dimensional PCA: A new approach to appearance-based face representation and recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 26(1), 131–137.
- Ye, J., Janardan, R., & Li, Q. (2005). Two-dimensional linear discriminant analysis. In L. K. Saul, Y. Weiss, & L. Bottou (Eds.), *Advances in neural information processing systems 17* (pp. 1569–1576). Cambridge: MIT Press.