Multi-Output Regularized Feature Projection

Shipeng Yu, Kai Yu, Volker Tresp, and Hans-Peter Kriegel

Abstract—Dimensionality reduction by feature projection is widely used in pattern recognition, information retrieval, and statistics. When there are some outputs available (e.g., regression values or classification results), it is often beneficial to consider *supervised projection*, which is based not only on the inputs, but also on the target values. While this applies to a single-output setting, we are more interested in applications with multiple outputs, where several tasks need to be learned simultaneously. In this paper, we introduce a novel projection approach called *Multi-Output Regularized feature Projection* (MORP), which preserves the information of input features and, meanwhile, captures the correlations between inputs/outputs and (if applicable) between multiple outputs. This is done by introducing a latent variable model on the joint input-output space and minimizing the reconstruction errors for both inputs and outputs. It turns out that the mappings can be found by solving a generalized eigenvalue problem and are ready to extend to nonlinear mappings. Prediction accuracy can be greatly improved by using the new features since the structure of outputs is explored. We validate our approach in two applications. In the first setting, we predict users' preferences for a set of paintings. The second is concerned with image and text categorization where each image (or document) may belong to multiple categories. The proposed algorithm produces very encouraging results in both settings.

Index Terms—Dimensionality reduction, supervised projection, feature transformation.

1 Introduction

Consider the pattern recognition task of predicting an output quantity y given an input feature vector $\mathbf{x}$. If the input space is high-dimensional and contains irrelevant features, the design of an appropriate pattern recognition system becomes a nontrivial problem. Thus, it is desirable to employ a preprocessing step in which input features are first projected into a new feature space that is compact, noise-free, and highly indicative. Projection methods such as principal component analysis (PCA) and linear discriminant analysis (LDA) (see [1]) have been applied intensively in various applications, like face recognition [2] and text retrieval [3], [4].

In contrast to unsupervised approaches, where projections are calculated based solely on the inputs, we study *supervised projection* methods where the feature projections are calculated based on both inputs and outputs. In general, this leads to an *informed* or *biased* feature projection, which will be more relevant to the particular supervised learning problem. In the case where we have only one output dimension, i.e., a regression or classification task, it is ideal to have the projection function informed by the *dependency* between inputs and outputs. More generally, if we have multiple output dimensions, i.e., for an input $\mathbf{x}$ the corresponding output is no longer a scalar but a vector $\mathbf{y} = [y_1, \ldots, y_L]^\top$, the *intracorrelation* between different dimensions of output should also be taken into account. In this paper, we consider the general case and call it a *multi-output* setting, and name the corresponding projection algorithm *Multi-Output Regularized feature Projection* (MORP).

The general multi-output setting is very common in real-world applications. One working example in this paper is to predict people's preferences on a set of paintings, which is a typical multi-output problem since, for each painting, many people's preferences have to be estimated. One may treat each person separately, but a notable fact is that people's tastes are usually correlated. One technology, known as *collaborative filtering* [5], explores people's like-mindedness to make predictions. Another example is the problem of multilabel text or image categorization, where each document/image is allowed to be associated with more than one category and where categories often have semantic correlations. For example, a document talking about category *car* must also belong to the category *vehicle*; an image in category *ski* is likely to be associated with category *snow*.

In this paper, we introduce a very general MORP algorithm to obtain supervised feature projection functions. In the algorithm, we try to minimize a *supervised cost function* for feature projections which includes both the *intercorrelation* between inputs and outputs and the *intracorrelation* between different output dimensions if we have at least two of them. We derive an analytical solution to the optimization problem, which turns out to be a generalized eigenvalue problem. A nonlinear projection is also easy to obtain if we introduce regularization to the optimization problem. The computational complexity of MORP is similar to a normal PCA problem, and empirical studies on preference prediction and multilabel classification verify the effectiveness of the algorithm.

1.1 Notations

We consider a set of N objects (e.g., documents). For $i = 1, \ldots, N$, each object i is described by an M-dimensional feature vector $\mathbf{x}_i \in \mathcal{X}$ and is associated (in general) with an L-dimensional output vector $\mathbf{y}_i \in \mathcal{Y}$. We denote the input data as a matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^\top \in \mathbb{R}^{N \times M}$, and the output data as $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]^\top \in \mathbb{R}^{N \times L}$, where $[\cdot]^\top$ denotes matrix

- *S. Yu and H.-P. Kriegel are with the Institute for Computer Science, University of Munich, Oettingenstrasse 67, D-80538 Munich, Germany. E-mail: {spyu, kriegel}@dbs.ifi.lmu.de.*
- *K. Yu and V. Tresp are with Siemens Corporate Technology, CT IC 4, Otto-Hahn-Ring 6, D-81730 Munich, Germany. E-mail: {kai.yu, volker.tresp}@siemens.com.*

Manuscript received 17 Nov. 2005; revised 6 July 2006; accepted 13 July 2006; published online 18 Oct. 2006.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-0515-1105.

transpose. In this paper, we aim to derive a mapping $\Psi : \mathcal{X} \mapsto \mathcal{V}$ that projects the input features into a K-dimensional latent space $\mathcal{V}$.

In the following, lowercase bold Roman letters denote column vectors and uppercase ones denote matrices. For a specific example, $\mathbf{v}$ is normally used for eigenvectors and $\mathbf{V}$ denotes the matrix $[\mathbf{v}_1, \ldots, \mathbf{v}_K]$. Eigenvalues are denoted as λ and it should be clear from the context which matrix they correspond to. Finally, $\|\cdot\|$ denotes the Frobenius norm for matrices and 2-norm for vectors and $\mathtt{tr}(\cdot)$ denotes trace for matrices.

1.2 Paper Organization

The paper is organized as follows: In Section 2, we formulate the supervised dimensionality reduction problem with a supervised latent variable model, and provide an analytical solution to this optimization problem. Then, we formally introduce the MORP algorithm in Section 3, in which both the primal form (linear projection) and the dual form (nonlinear projection) are discussed. In Section 4, we point out its connections to kernel PCA and some supervised projection methods and, in Section 5, we report some experimental results on preference prediction and multilabel classification. Finally, Section 6 concludes the paper.

2 Supervised Latent Variable Model

The supervised latent variable model is motivated from the unsupervised latent variable model for PCA, which we briefly review at first.

2.1 Unsupervised Latent Variable Model

In unsupervised linear projection, we aim at finding a linear mapping from the input space $\mathcal{X}$ to some low-dimensional latent space $\mathcal{V}$, while most of the structure in the data can be explained and recovered. In this sense, we can turn this linear projection problem to an optimization problem, where we are trying to minimize the *reconstruction error*:

$$\begin{aligned} \min_{\mathbf{A},\mathbf{V}} \quad & \|\mathbf{X} - \mathbf{V}\mathbf{A}\|^2 \\ \text{subject to :} \quad & \mathbf{V}^\top \mathbf{V} = \mathbf{I}, \end{aligned} \tag{1}$$

where $\mathbf{V} \in \mathbb{R}^{N \times K}$ gives the K-dimensional *projections* of objects and $\mathbf{A} \in \mathbb{R}^{K \times M}$ is the *loading matrix*. By constraining $\mathbf{V}^\top \mathbf{V} = \mathbf{I}$, we restrict the K latent variables to be linearly independent, i.e., to have diagonal covariance matrix in latent space.

Since matrix product $\mathbf{V}\mathbf{A}$ has rank K, in (1), we are indeed seeking a low-rank approximation to the data matrix $\mathbf{X}$. It can be shown that (1) has strong connections to PCA and kernel PCA (see Section 4.1).

The derived projection explains the covariance of input data, while it is not necessarily relevant to the outputs. Thus, unsupervised feature projections may or may not be beneficial to supervised learning problems. Generally speaking, it is more desirable to consider the *correlation* between input $\mathbf{X}$ and output $\mathbf{Y}$ and the *intracorrelation* between the L dimensions of $\mathbf{Y}$ if $L > 1$. Therefore, we turn to *supervised projection* in the next section, incorporating both input $\mathbf{X}$ and output $\mathbf{Y}$.

2.2 A Supervised Latent Variable Model

The unsupervised projection (1) explicitly represents the projections of input data $\mathbf{X}$ in matrix $\mathbf{V}$. To consider the output information, we can enforce the projections $\mathbf{V}$ in (1) to be sensitive to $\mathbf{Y}$ as well. Thus, in supervised projection, we solve the following optimization problem:

$$\begin{aligned} \min_{\mathbf{A},\mathbf{B},\mathbf{V}} \quad & (1-\beta)\|\mathbf{X} - \mathbf{V}\mathbf{A}\|^2 + \beta\|\mathbf{Y} - \mathbf{V}\mathbf{B}\|^2 \\ \text{subject to :} \quad & \mathbf{V}^\top \mathbf{V} = \mathbf{I}, \end{aligned} \tag{2}$$

where $\mathbf{V} \in \mathbb{R}^{N \times K}$ gives the K-dimensional projections of objects, for features of both $\mathbf{X}$ and $\mathbf{Y}$. $\mathbf{A} \in \mathbb{R}^{K \times M}$, $\mathbf{B} \in \mathbb{R}^{K \times L}$ are the loading matrices. For input $\mathbf{X}$ and output $\mathbf{Y}$, respectively, $0 \leq \beta \leq 1$ is a tuning parameter determining how much the projections should be biased by the outputs. As before, $\mathbf{V}^\top \mathbf{V} = \mathbf{I}$ restricts the K latent variables to be linearly independent. Clearly, the cost function is a trade-off between the *reconstruction error* of both $\mathbf{X}$ and $\mathbf{Y}$. We wish to find the optimal projections that give the minimum reconstruction error.

To see the optimization problem more clearly, we rewrite the cost in (2) as

$$(1-\beta)\|\mathbf{X} - \mathbf{V}\mathbf{A}\|^2 + \beta \sum_{l=1}^{L} \sum_{i=1}^{N} \left((\mathbf{y}_i)_l - \sum_{k=1}^{K} \mathbf{V}_{i,k} \mathbf{B}_{k,l} \right)^2,$$

where $\mathbf{V}_{i,k}$ denotes the (i,k)th entry in $\mathbf{V}$. Then, we have the following observations:

- When $L = 1$, the second part of the cost constrains that the output $\mathbf{Y}$ can be *linearly reconstructed* from the latent projection $\mathbf{V}$. Therefore, in the whole optimization problem, we are minimizing the *correlation* between $\mathbf{X}$ and $\mathbf{Y}$.
- When $L > 1$, all of the columns of $\mathbf{Y}$ are constrained to be linearly reconstructable from $\mathbf{V}$. Therefore, they are not considered independently, but *jointly*. In other words, we are minimizing the *intracorrelation* between columns of $\mathbf{Y}$.

We would mention here that one can very easily generalize this cost function to have a different weight β_l for the lth output dimension while maintaining a proper normalization for all β_ls. In this way, one can constrain each output dimension differently, which leads to a more flexible cost with, however, more free parameters. In the following, we stick to the simpler setting since all the learning algorithms can be easily generalized to cover this case.

Remark 1. When $L = 1$, finding the mapping from $\mathbf{X}$ to $\mathbf{Y}$ is known as a *regression* problem if $\mathbf{y}_i \in \mathbb{R}$, or *multiclass classification* if $\mathbf{y}_i$ is chosen from a finite set of integers. In the general case when $L > 1$, the former is called *multivariate regression*, while the latter is called *multilabel classification*. Therefore, our notations consider the most general setting, which we call a *multi-output* problem.[1]

1. In this paper, we explicitly distinguish between multiclass classification and multilabel classification, both of which classify objects into multiple categories. In "multiclass classification," an object can *only* belong to one category, while, in "multilabel classification," one object can belong to several categories simultaneously. Therefore, the former is a special case of the latter. In this paper, multiclass classification is viewed as a *single-output* problem since we can label the multiple categories as integers and assign one integer to one object. The more general multilabel classification is considered a *multi-output* setting, where each output could be a binary or multiclass classifier. Our setting covers both cases and is thus very general.

The following proposition states the interdependency between $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{V}$ at the optimum.

Proposition 2. *If $\mathbf{V}, \mathbf{A}$ and $\mathbf{B}$ are the optimal solutions to (2), and $\mathbf{G} = (1-\beta)\mathbf{X}\mathbf{X}^\top + \beta\mathbf{Y}\mathbf{Y}^\top$, then:*

1. $\mathbf{A} = \mathbf{V}^\top\mathbf{X}$, $\mathbf{B} = \mathbf{V}^\top\mathbf{Y}$.
2. *At the optimum, the objective function in (2) is equal to $\mathtt{tr}(\mathbf{G}) - \mathtt{tr}\big(\mathbf{V}^\top\mathbf{G}\mathbf{V}\big)$.*

To improve readability, we put all proofs into the Appendix. Since $\mathtt{tr}(\mathbf{G})$ is fixed, Proposition 2 suggests that (2) can be considered to be an optimization problem only with respect to $\mathbf{V}$:

$$\begin{aligned} \max_{\mathbf{V}\in\mathbb{R}^{N\times K}} \quad & \mathtt{tr}\Big(\mathbf{V}^\top\mathbf{G}\mathbf{V}\Big) \\ \text{subject to :} \quad & \mathbf{V}^\top\mathbf{V} = \mathbf{I}. \end{aligned} \tag{3}$$

Note that an ambiguity arises in (2) and (3): If $\mathbf{V}$ is the solution, then $\tilde{\mathbf{V}} = \mathbf{V}\mathbf{R}$ is also a solution, given an arbitrary rotation matrix $\mathbf{R}$. The following theorem summarizes the situation.

Theorem 3. *Suppose that $[\mathbf{v}_1, \ldots, \mathbf{v}_N]$ are the eigenvectors of matrix $\mathbf{G}$ and $\lambda_1 \geq \ldots \geq \lambda_N$ are the corresponding eigenvalues. If $\tilde{\mathbf{V}}$ solves (3), then*

1. *$\tilde{\mathbf{V}} = [\mathbf{v}_1, \ldots, \mathbf{v}_K]\mathbf{R}$, where $\mathbf{R}$ is an arbitrary $K \times K$ orthogonal rotation matrix.*
2. *The maximum of the objective function (3) is $\sum_{i=1}^{K}\lambda_i$.*

The theorem states that the eigenvectors of $\mathbf{G}$ form a solution to (2) and any arbitrary rotation does not change the optimum. However, to remove the ambiguity, we are focusing on the solutions given by the eigenvectors of $\mathbf{G}$, i.e., $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_K]$. Therefore, (2) can be equivalently achieved by solving:[2]

$$\begin{aligned} \max_{\mathbf{v}\in\mathbb{R}^{N}} \quad & \mathbf{v}^\top\mathbf{G}\mathbf{v} \\ \text{subject to :} \quad & \mathbf{v}^\top\mathbf{v} = 1. \end{aligned} \tag{4}$$

Setting the Lagrange derivative to be zero, we have the eigenvalue problem $\mathbf{G}\mathbf{v} = \lambda\mathbf{v}$. Let $\mathbf{v}_1, \ldots, \mathbf{v}_N$ be the eigenvectors of $\mathbf{G}$ with the eigenvalues sorted in a *nonincreasing* order. Using the first K eigenvectors, we solve (2) as $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_K]$, $\mathbf{A} = \mathbf{V}^\top\mathbf{X}$, and $\mathbf{B} = \mathbf{V}^\top\mathbf{Y}$.

3 Multi-Output Regularized Feature Projection

The solution to the supervised latent variable model is elegant, but only applicable to training data since, for test data, we do not have any output information. Therefore, to complete the MORP algorithm, we need to refine the original problem.

1. *Linear Constraint.* It is easy to see that solving (4) only gives the projections for training data with both features inputs in $\mathbf{X}$ and complete outputs in $\mathbf{Y}$. We wish to construct a mapping $\Psi : \mathcal{X} \mapsto \mathcal{V}$ that is able to handle the input features of any new objects; thus, we restrict the latent variables as *linear mappings* of $\mathbf{X}$:

$$\mathbf{V} = \mathbf{X}\mathbf{W}.$$

This is one key step for the proposed supervised projection algorithm. With this constraint, we turn the original problem to an optimization problem with respect to the linear weight matrix $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_K] \in \mathbb{R}^{M\times K}$ and, by definition, we have $\mathbf{v}_k = \mathbf{X}\mathbf{w}_k$, for $k = 1, \ldots, K$. Plugging $\mathbf{v} = \mathbf{X}\mathbf{w}$ into (4), we have an optimization problem with respect to $\mathbf{w}$ only:

$$\begin{aligned} \max_{\mathbf{w}\in\mathbb{R}^{M}} \quad & \mathbf{w}^\top\mathbf{X}^\top\mathbf{G}\mathbf{X}\mathbf{w} \\ \text{subject to :} \quad & \mathbf{w}^\top\mathbf{X}^\top\mathbf{X}\mathbf{w} = 1. \end{aligned} \tag{5}$$

2. *Regularization.* Similarly to other linear systems, the learned mappings can be unstable when the $\mathtt{span}\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ has a lower rank than $\mathbb{R}^M$ due to the small size of the training set or dependence between input features. As a result, a disturbance of $\mathbf{w}$ with an arbitrary $\mathbf{w}^* \perp \mathtt{span}\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ does not change the objective function of optimization since $(\mathbf{w} + \mathbf{w}^*)^\top\mathbf{x}_i = \mathbf{w}^\top\mathbf{x}_i$, but may dramatically change the projections of unseen test points which are not in the spanned space. To improve the stability, we have to constraint $\mathbf{w}$ in some way.

Suppose $\mathtt{rank}(\mathbf{G}) = N$, then (4) is equivalent to minimizing $\mathbf{v}^\top\mathbf{G}^{-1}\mathbf{v}$.[3] We introduce the *Tikhonov regularization* [6] into (5) as the following:

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{R}^{M}} \quad & \mathbf{w}^\top\mathbf{X}^\top\mathbf{G}^{-1}\mathbf{X}\mathbf{w} + \gamma\|\mathbf{w}\|^2 \\ \text{subject to :} \quad & \mathbf{w}^\top\mathbf{X}^\top\mathbf{X}\mathbf{w} = 1, \end{aligned} \tag{6}$$

where $\|\mathbf{w}\|^2 = \mathbf{w}^\top\mathbf{w}$ is a penalty term which has been used in the ridge regression (see [1]) and $\gamma > o$ is a tuning parameter. Setting the derivative of its Lagrange formulism with respect to $\mathbf{w}$ to be zero, we reach a generalized eigenvector problem:

$$\big[\mathbf{X}^\top\mathbf{G}^{-1}\mathbf{X} + \gamma\mathbf{I}\big]\mathbf{w} = \tilde{\lambda}\mathbf{X}^\top\mathbf{X}\mathbf{w}, \tag{7}$$

which gives generalized eigenvectors $\mathbf{w}_1, \ldots, \mathbf{w}_M$ with eigenvalues $\tilde{\lambda}_1 \leq \ldots \leq \tilde{\lambda}_M$. Note that we sort eigenvalues in an nondecreasing order since we take the K eigenvectors with the smallest eigenvalues to form the mapping.

2. Solving (4) itself only gives the first eigenvector $\mathbf{v}_1$ of $\mathbf{G}$. The full optimization problem should be recursively computing $\mathbf{v}_j$ by maximizing $\mathbf{v}^\top\mathbf{G}\mathbf{v}$ with the constraint $\mathbf{v}^\top\mathbf{v} = 1$ and $\mathbf{v} \perp \mathtt{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_{j-1}\}$. Here, we state the problem as (4) for simplicity and also because its Lagrange formulism directly leads to the eigenvalue problem.

3. One can also minimize $\mathbf{v}^\top(-\mathbf{G})\mathbf{v}$ which is also equivalent, but then we lose nonnegativity of its eigenvalues, which may cause problems later on in solving the generalized eigenvalue problem. For the invertibility of $\mathbf{G}$, it is easy to show that $\mathbf{G}$ is at least positive semidefinite since we have $\mathbf{u}^\top\mathbf{G}\mathbf{u} = (1-\beta)\mathbf{u}^\top\mathbf{X}\mathbf{X}^\top\mathbf{u} + \beta\mathbf{u}^\top\mathbf{Y}\mathbf{Y}^\top\mathbf{u} = (1-\beta)\|\mathbf{X}^\top\mathbf{u}\|^2 + \beta\|\mathbf{Y}^\top\mathbf{u}\|^2 \geq 0$, $\forall\mathbf{u} \in \mathbb{R}^N$. In case the $\mathbf{G}$ is not positive definite, it suffices to use pseudoinverse instead or makes it so by adding a tiny positive scalar to diagonal elements. In the dual form in Section 3.2, $\mathbf{G}$ is, in most cases, positive definite since $\mathbf{G}_x$ is normally positive definite (with, e.g., RBF kernel) and $\mathbf{G}_y$ is at least positive semidefinite.

The following theorem shows that the regularization term $\|\mathbf{w}\|^2$ removes the ambiguity of mapping functions by restricting $\mathbf{w}$ in the span of $\mathbf{x}_i$, $i = 1, \ldots, N$, and, thus, improves the stability of mapping functions.

Theorem 4. *If* $\mathbf{w}$ *is an eigenvector of the generalized eigenvalue problem (7), then* $\mathbf{w}$ *must be a linear combination of* $\mathbf{x}_i$, $i = 1, \ldots, N$, *namely,*

$$\mathbf{w} = \mathbf{X}^\top \boldsymbol{\alpha} = \sum_{i=1}^{N} (\boldsymbol{\alpha})_i \mathbf{x}_i,$$

where $\boldsymbol{\alpha} \in \mathbb{R}^N$.

3.1 MORP-Primal Form

In (7), we are interested in the eigenvectors with the smallest eigenvalues, whose computation is, however, the most unstable part in solving an eigenvalue problem. Thus, we let $\lambda = 1/\tilde{\lambda}$ and turn the problem into an equivalent one as

$$\mathbf{X}^\top \mathbf{X} \mathbf{w} = \lambda \big[\mathbf{X}^\top \mathbf{G}^{-1} \mathbf{X} + \gamma \mathbf{I}\big] \mathbf{w}, \tag{8}$$

where we are seeking the K eigenvectors with the *largest* eigenvalues. To solve this problem, we note that matrix $\mathbf{Q} = \mathbf{X}^\top \mathbf{G}^{-1} \mathbf{X} + \gamma \mathbf{I}$ is symmetric and positive definite, so there exists a symmetric and positive definite matrix $\mathbf{L}$ such that $\mathbf{Q} = \mathbf{L}^2$. Then, we can change the problem to an equivalent one as $\mathbf{L}^{-1} \mathbf{X}^\top \mathbf{X} \mathbf{L}^{-1} \mathbf{L} \mathbf{w} = \lambda \mathbf{L} \mathbf{w}$, in which we can solve an eigenvalue problem for matrix $\mathbf{L}^{-1} \mathbf{X}^\top \mathbf{X} \mathbf{L}^{-1}$ with eigenvectors given as $\mathbf{z} = \mathbf{L}\mathbf{w}$. After that, we can recover $\mathbf{w}$ as $\mathbf{w} = \mathbf{L}^{-1}\mathbf{z}$. Note that the solution satisfies $\mathbf{w}^\top \mathbf{Q} \mathbf{w} = 1$. This leads to an additional scaling factor for the mapping.

As can be seen from the optimization problem in (2), MORP assumes that the projections of all the data points in $\mathbf{X}$ have $\mathbf{I}$ as the covariance matrix. This means all the scaling factors (i.e., the variance on each projection direction) are not maintained in the projection values. This will cause problems since the pairwise distances are changed. Therefore, we add these scaling factors back after we find the projection directions, which will recover PCA if no output information is available.

This primal form of the MORP algorithm is summarized in Algorithm 1. To only extract the projection dimensions which represent the intrinsic structure of the data, we centralize the data before performing the algorithm, i.e., subtract the sample mean from each data point.

Algorithm 1 MORP in Primal Form

Require: A set of N data points with M-dimensional input features $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^\top \in \mathbb{R}^{N \times M}$ and L-dimensional outputs $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]^\top \in \mathbb{R}^{N \times L}$.

Require: Projection dimension $K > 0$. $0 \leq \beta \leq 1$, $\gamma \geq 0$.

1: Centralize data: $\mathbf{x}_i \Leftarrow \mathbf{x}_i - \bar{\mathbf{x}}$, $\mathbf{y}_i \Leftarrow \mathbf{y}_i - \bar{\mathbf{y}}$ where $\bar{\mathbf{x}} = \frac{1}{N}\sum_i \mathbf{x}_i$, $\bar{\mathbf{y}} = \frac{1}{N}\sum_i \mathbf{y}_i$.

2: Calculate $\mathbf{G} = (1-\beta)\mathbf{X}\mathbf{X}^\top + \beta \mathbf{Y}\mathbf{Y}^\top$.

3: Set $\mathbf{P} = \mathbf{X}^\top \mathbf{X}$ and $\mathbf{Q} = [\mathbf{X}^\top \mathbf{G}^{-1} \mathbf{X} + \gamma \mathbf{I}]$. Solve the generalized eigenvalue problem: $\mathbf{P}\mathbf{w} = \lambda \mathbf{Q}\mathbf{w}$, obtain eigenvectors $\mathbf{w}_1, \ldots, \mathbf{w}_K$ with largest K eigenvalues $\lambda_1 \geq \ldots \geq \lambda_K$ such that $\mathbf{w}^\top \mathbf{Q} \mathbf{w} = 1$.

Output: Projection function for the kth dimension as $\psi_k(\mathbf{x}) = \mathbf{w}_k^\top \mathbf{x}$, $k = 1, \ldots, K$.

3.2 MORP-Dual Form

So far, we have considered linear mappings that project inputs $\mathbf{x}$ into latent space $\mathcal{V}$. However, Theorem 4 implies that we can also derive a nonlinear mapping Ψ.

Let a *kernel* function $\kappa_x(\cdot, \cdot)$ be the inner product in $\mathcal{X}$, i.e., $\kappa_x(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle = \mathbf{x}_i^\top \mathbf{x}_j$, then, from Theorem 4,

$$\mathbf{v} = \mathbf{X}\mathbf{w} = \mathbf{X}\mathbf{X}^\top \boldsymbol{\alpha} = \mathbf{G}_x \boldsymbol{\alpha},$$

where $\mathbf{G}_x$ is the $N \times N$ kernel matrix satisfying $(\mathbf{G}_x)_{i,j} = \kappa_x(\mathbf{x}_i, \mathbf{x}_j)$. $\|\mathbf{w}\|^2$ can also be calculated with kernel $\mathbf{G}_x$:

$$\|\mathbf{w}\|^2 = \mathbf{w}^\top \mathbf{w} = \boldsymbol{\alpha}^\top \mathbf{X}\mathbf{X}^\top \boldsymbol{\alpha} = \boldsymbol{\alpha}^\top \mathbf{G}_x \boldsymbol{\alpha}.$$

Similarly, we can define a kernel function $\kappa_y(\cdot, \cdot)$ for inner product in $\mathcal{Y}$ and obtain a kernel matrix $\mathbf{G}_y = \mathbf{Y}\mathbf{Y}^\top$. Then, we can calculate the matrix $\mathbf{G}$ using kernels

$$\mathbf{G} = (1-\beta)\mathbf{G}_x + \beta \mathbf{G}_y \tag{9}$$

and express the *dual formalism* of (6) with respect to coefficients $\boldsymbol{\alpha}$ as

$$\begin{aligned} \min_{\boldsymbol{\alpha} \in \mathbb{R}^N} \quad & \boldsymbol{\alpha}^\top \mathbf{G}_x \mathbf{G}^{-1} \mathbf{G}_x \boldsymbol{\alpha} + \gamma \boldsymbol{\alpha}^\top \mathbf{G}_x \boldsymbol{\alpha} \\ \text{subject to :} \quad & \boldsymbol{\alpha}^\top \mathbf{G}_x^2 \boldsymbol{\alpha} = 1. \end{aligned} \tag{10}$$

The Lagrange solution of this problem leads to a generalized eigenvalue problem

$$\big[\mathbf{G}_x \mathbf{G}^{-1} \mathbf{G}_x + \gamma \mathbf{G}_x\big] \boldsymbol{\alpha} = \tilde{\lambda} \mathbf{G}_x^2 \boldsymbol{\alpha}. \tag{11}$$

We obtain the generalized eigenvectors $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_N$, with $\tilde{\lambda}_1 \leq \ldots \leq \tilde{\lambda}_N$. The first K eigenvectors are applied to form the mappings. The kth mapping function, $k = 1, \ldots, K$, is given by

$$\psi_k(\mathbf{x}) = \mathbf{w}_k^\top \mathbf{x} = \sum_{i=1}^{N} (\boldsymbol{\alpha}_k)_i \kappa_x(\mathbf{x}_i, \mathbf{x}).$$

As before, we define $\lambda = 1/\tilde{\lambda}$ and change (11) to the following equivalent form:

$$\mathbf{G}_x^2 \boldsymbol{\alpha} = \lambda \big[\mathbf{G}_x \mathbf{G}^{-1} \mathbf{G}_x + \gamma \mathbf{G}_x\big] \boldsymbol{\alpha} \tag{12}$$

and, hence, we can choose the K eigenvectors with the largest eigenvalues.

The algorithm is ready to deal with *nonlinear* mappings. For this, we consider a nonlinear mapping $\phi : \mathbf{x} \in \mathcal{X} \mapsto \phi(\mathbf{x}) \in \mathcal{F}$, which maps $\mathbf{x}$ into a high-dimensional or even infinite-dimensional feature space $\mathcal{F}$ and changes $\mathbf{X}$ to be $[\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_N)]^\top$. Then, the kernel function is accordingly defined as

$$\kappa_x(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle,$$

where we still have $\mathbf{G}_x = \mathbf{X}\mathbf{X}^\top$. Therefore, we can directly work with kernels, e.g., RBF kernel

$$\kappa_x(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2),$$

without knowing $\phi(\cdot)$ explicitly.

Similarly, we can define a nonlinear mapping for $\mathcal{Y}$ and directly work on the corresponding kernel matrix $\mathbf{G}_y$.

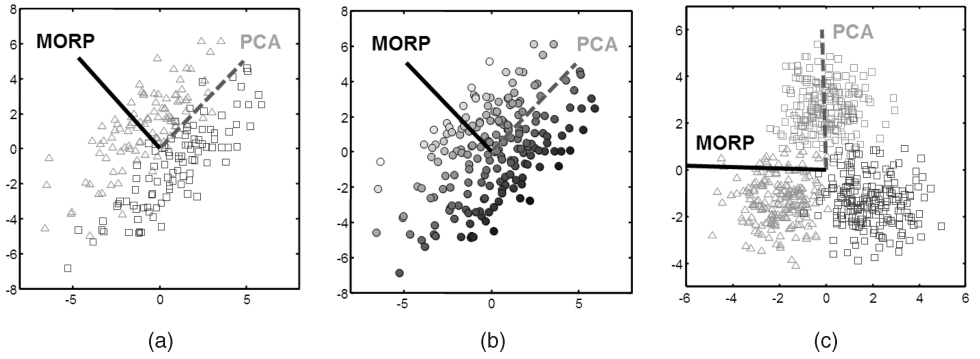

Fig. 1. The first projection directions of MORP (black thick line) and PCA (gray dashed line) for three toy data sets with different output types: (a) Binary classification ($+1$ for triangles and -1 for squares), (b) regression values (-1 to $+1$ from top-left to bottom-right, shown with different gray scales), and (c) multilabel classification. In (c), the first classifier has label $+1$ for points in the the bottom-left cluster and the top cluster, and -1 for points in the bottom-right cluster; the second classifier has label $+1$ for squares (the bottom-right cluster and the top cluster) and -1 for triangles (the bottom-left cluster). Note that (c) is not a multiclass problem. For all three data sets, β is set to 0.5 for MORP.

Although this paper mainly considers the linear kernel to explore the linear correlation of multivariate outputs, the formulism implies that the method can generally handle more complex outputs by using some other suitable kernels.

For centering of the data, we can achieve this in the new feature space $\mathcal{F}$ without knowing the explicit mapping $\phi(\cdot)$ ([7]):

$$\mathbf{G} \Leftarrow \mathbf{G} - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^\top\mathbf{G} - \frac{1}{N}\mathbf{G}\mathbf{1}_N\mathbf{1}_N^\top + \frac{1}{N^2}\mathbf{1}_N\mathbf{1}_N^\top\mathbf{G}\mathbf{1}_N\mathbf{1}_N^\top, \quad (13)$$

where $\mathbf{1}_N$ denotes the all one column vector $[1,\ldots,1]^\top$ of length N. For the kernel vector $\mathbf{k}(\mathbf{X},\mathbf{x})$ given test data $\mathbf{x}$, it can also be centered by

$$\mathbf{k} \Leftarrow \mathbf{k} - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^\top\mathbf{k} - \frac{1}{N}\mathbf{G}\mathbf{1}_N + \frac{1}{N^2}\mathbf{1}_N\mathbf{1}_N^\top\mathbf{G}\mathbf{1}_N. \quad (14)$$

The final dual form of the algorithm is summarized in Algorithm 2.

Algorithm 2 MORP in Dual Form

Require: A set of N data points with M-dimensional input features $\mathbf{X} = [\mathbf{x}_1,\ldots,\mathbf{x}_N]^\top \in \mathbb{R}^{N\times M}$ and L-dimensional outputs $\mathbf{Y} = [\mathbf{y}_1,\ldots,\mathbf{y}_N]^\top \in \mathbb{R}^{N\times L}$.

Require: Kernel functions $\kappa_x(\cdot,\cdot)$ and $\kappa_y(\cdot,\cdot)$ for input space $\mathcal{X}$ and output space $\mathcal{Y}$. Projection dimension $K > 0$. $0 \le \beta \le 1$, $\gamma \ge 0$.

1: Calculate two $N \times N$ matrices $(\mathbf{G}_x)_{ij} = \kappa_x(\mathbf{x}_i,\mathbf{x}_j)$, $(\mathbf{G}_y)_{ij} = \kappa_y(\mathbf{y}_i,\mathbf{y}_j)$.

2: Centralize the kernel matrices $\mathbf{G}_x$ and $\mathbf{G}_y$ using (13).

3: Calculate $\mathbf{G} = (1-\beta)\mathbf{G}_x + \beta\mathbf{G}_y$.

4: Set $\mathbf{P} = \mathbf{G}_x^2$ and $\mathbf{Q} = \mathbf{G}_x\mathbf{G}^{-1}\mathbf{G}_x + \gamma\mathbf{G}_x$. Solve the generalized eigenvalue problem: $\mathbf{P}\boldsymbol{\alpha} = \lambda\mathbf{Q}\boldsymbol{\alpha}$, obtain eigenvectors $\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_K$ with largest eigenvalues $\lambda_1 \ge \ldots \ge \lambda_K$ such that $\boldsymbol{\alpha}^\top\mathbf{Q}\boldsymbol{\alpha} = 1$.

Output: Projection function for the kth dimension as $\psi_k(\mathbf{x}) = \vec{k}(\mathbf{X},\mathbf{x})^\top\boldsymbol{\alpha}_k$, $k = 1,\ldots,K$, where $\vec{k}(\mathbf{X},\mathbf{x}) := [\kappa_x(\mathbf{x}_1,\mathbf{x}),\ldots,\kappa_x(\mathbf{x}_N,\mathbf{x})]^\top$ and centralized via (14).

3.3 Discussions

In Fig. 1, we show the projection directions of MORP and PCA for three toy data. Fig. 1a shows binary classification, Fig. 1b shows regression values, and Fig. 1c shows multilabel classification, respectively. In all the cases, MORP can find the most informative directions for the specific supervised learning problems and deviates from the PCA directions dramatically. This means various outputs can significantly bias the extracted features. PCA just finds the direction which has the largest data variance in all the cases.

MORP defines a general solution for supervised projection, i.e., *minimization of an output-regularized cost function*. In general, one can go beyond the Frobenius norm and consider a more general cost for $\mathbf{X}$ and $\mathbf{Y}$:

$$(1-\beta)f(\mathbf{X},\mathbf{V}) + \beta g(\mathbf{Y},\mathbf{V}),$$

where f and g define the *input-specific cost* and *output-specific cost*, respectively, with respect to the observation ($\mathbf{X}$ or $\mathbf{Y}$) and the projection $\mathbf{V}$. There may be some parameters involved (like $\mathbf{A}$ and $\mathbf{B}$ in the Frobenius norm case) and, in general, there is no analytical solution to this optimization problem. For instance, f could be matrix 1-norm (like the case for sparse PCA [8]) and g could be hinge-lose for the binary classification problem [9]. For simplicity and tractability, we stick to the Frobenius norm in this paper.

As a natural extension of (2), we can have another output set, say $\mathbf{Z}$, associated with all input data and add the reconstruction error of $\mathbf{Z}$ to the cost function. Then, the cost function looks like

$$(1-\beta_1-\beta_2)\|\mathbf{X}-\mathbf{V}\mathbf{A}\|^2 + \beta_1\|\mathbf{Y}-\mathbf{V}\mathbf{B}\|^2 + \beta_2\|\mathbf{Z}-\mathbf{V}\mathbf{C}\|^2$$

and, potentially, $\mathbf{Z}$ could have different intracorrelations compared to $\mathbf{Y}$. Both of these two output sets can be incorporated into MORP by defining possibly different kernels for $\mathbf{Y}$ and $\mathbf{Z}$ and including them into the matrix $\mathbf{G}$. Therefore, MORP introduces an elegant way to take into account various supervised information and allows great flexibility and generalization ability.

MORP solves a generalized eigenvalue problem for $M \times M$ matrices in the primal form and for $N \times N$ matrices in the dual form, which, in computational complexity, is similar to unsupervised projection PCA and kernel PCA (see [10] for details of generalized eigenvalue problems). The time complexity of the primal and dual solutions is, respectively, $\mathcal{O}(mM^2K)$ and $\mathcal{O}(mN^2K)$ if we use the power

method to solve the eigenvalue problem, where m is the number of iterations. For implementation, it is very easy and just takes several lines with Matlab. The calculations of kernels and matrix multiplications are the most time-consuming parts of the algorithm, as well as the matrix inversion in kernel form.

4 Connections to Related Works

The proposed algorithm MORP is a *supervised* projection from the input space to the latent space and aims at minimizing the reconstruction errors of both input data $\mathbf{X}$ and output data $\mathbf{Y}$. The algorithm is naturally generalizable to nonlinear mappings and can explore the intracorrelation of multiple outputs.

In the literature, there are some other well-known unsupervised and supervised projection methods, such as principal components analysis (PCA) ([11], [12]), linear discriminant analysis (LDA) ([1], [13]), canonical correlation analysis (CCA) ([14], [15]), partial least squares (PLS) ([16], [17]), and kernel dependency estimation (KDE) [18]. In this section, we briefly review these methods and point out the substantial differences as well as possible connections between MORP and these methods. Other recent works include kernel dimensionality reduction [19], multitask learning ([20], [21], [22]) and will also be briefly discussed.

4.1 Kernel Principal Component Analysis (Kernel PCA)

PCA is shown to be a robust unsupervised method for linear projection and has been intensively applied to regression and classification applications [11]. Kernel PCA releases the linear limitation of PCA and is actually performing a linear PCA in a kernel induced feature space, i.e., a reproducing kernel Hilbert space ([7], [12], also see the discussions in [23]). Let $\bar{\phi} : \mathbf{x} \in \mathcal{X} \mapsto \bar{\phi}(\mathbf{x}) \in \mathcal{F}$ denote the nonlinear mapping and let $\bar{\mathbf{G}}$ denote the corresponding kernel matrix, i.e., $\bar{\mathbf{G}}_{ij} = \langle \bar{\phi}(\mathbf{x}_i), \bar{\phi}(\mathbf{x}_j) \rangle_{\mathcal{F}}$. It turns out that kernel PCA is readily performed by solving the following eigenvalue problem:

$$\bar{\mathbf{G}}\boldsymbol{\alpha} = \lambda\boldsymbol{\alpha}.$$

After the eigenvectors $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_K$ with largest eigenvalues $\lambda_1 \geq \ldots \geq \lambda_K$ are obtained, the nonlinear mappings $\psi_j(\mathbf{x}) = \sum_{i=1}^{N} (\boldsymbol{\alpha}_j)_i \bar{k}(\mathbf{x}_i, \mathbf{x})$, $j = 1, \ldots, K$, project the input data $\mathbf{x}$ to a K-dimensional latent space, where kernel function $\bar{k}(\mathbf{x}_i, \mathbf{x}) = \langle \bar{\phi}(\mathbf{x}_i), \bar{\phi}(\mathbf{x}) \rangle_{\mathcal{F}}$.

The proposed method MORP is motivated from (2) and is also performing an unsupervised projection when $\beta = 0$, which is identical to (1). In this case, we have $\mathbf{G} = \mathbf{X}\mathbf{X}^\top$, which is just the kernel matrix $\mathbf{G}_x$ for $\mathbf{X}$ in dual form, as revealed by (9). Then, from Theorem 3 and remarks after (4), it is clear that, when $\beta = 0$, MORP is also solving the eigenvalue problem for $\mathbf{G}_x$ and, thus, is *identical* to kernel PCA. This also clarifies the connection between kernel PCA and (1): The optimal solution $\mathbf{V}$ to (1) corresponds to, up to a rotation factor, the K nonlinear principal components of kernel PCA in columns.

When $\beta = 0$, the equivalence of MORP and kernel PCA can also be shown from (10), which changes to

$$\begin{aligned} \min_{\boldsymbol{\alpha} \in \mathbb{R}^N} \quad & (1+\gamma)\boldsymbol{\alpha}^\top \mathbf{G}_x \boldsymbol{\alpha} \\ \text{subject to :} \quad & \boldsymbol{\alpha}^\top \mathbf{G}_x^2 \boldsymbol{\alpha} = 1, \end{aligned}$$

since $\mathbf{G} = \mathbf{G}_x$ holds. Under this situation, the regularization term controlled by γ is just a rescaling of the cost function and therefore does not change the cost function at all. Hence, γ is just a nuisance parameter. On the other hand, if we let $\gamma \to \infty$, the regularization term in (10) dominates the cost function and MORP tends to be kernel PCA, whatever β is.

When $\beta > 0$, MORP actually performs *output regularized* kernel PCA or *supervised* kernel PCA since it can be viewed as directly modifying the kernel matrix $\mathbf{G}$ with output information. With moderate β, the mapping takes into account the kernel of $\mathbf{Y}$, but is meanwhile restricted to the input space $\mathcal{X}$. No information on $\mathbf{y}$ is required for calculating MORP projection of a new data point $\mathbf{x}$.

4.2 Linear Discriminant Analysis (LDA)

LDA or Fisher Discriminant Analysis (FDA) is a canonical supervised projection for input data $\mathbf{X}$ and conceptually can only handle binary classification problems (see [1], [24]). It chooses a projection direction $\mathbf{w}$ that maximizes the *interdistance* of projected means and, meanwhile, minimizes the *intravariances* of both classes. Therefore, it focuses on the single classification problem where the output is one-dimensional, while, in contrast, MORP considers predictions with multivariate outputs and is thus more general.

A recently proposed approach, Kernel Dimensionality Reduction (KDR) [19], is also a supervised method and aims to find a low-dimensional effective subspace that retains the statistical relationship between input data and output data. However, it has similar limitations and can only handle one-dimensional output.

4.3 Canonical Correlation Analysis (CCA) and Partial Least Squares (PLS)

CCA has a long history in the statistics community (back to [14]) and aims at discovering the correlations between two representatives of the same objects (e.g., inputs $\mathbf{X}$ and outputs $\mathbf{Y}$ in our setting). The optimization problem solving CCA can be written as:

$$\begin{aligned} \max_{\mathbf{v}_x, \mathbf{v}_y \in \mathbb{R}^N} \quad & \mathrm{Corr}(\mathbf{v}_x, \mathbf{v}_y) \\ \text{subject to :} \quad & \mathbf{v}_x = \mathbf{X}\mathbf{w}_x, \mathbf{v}_y = \mathbf{Y}\mathbf{w}_y, \end{aligned}$$

which is equivalent to minimization of $\|\mathbf{v}_x - \mathbf{v}_y\|^2$ when both $\mathbf{v}_x$ and $\mathbf{v}_y$ have norm 1 (see a recent discussion in [15]). In this sense, CCA is a certain kind of supervised projection, but it does not require the projections $\mathbf{v}_x$ and $\mathbf{v}_y$ to guarantee a low-reconstruction error of $\mathbf{X}$ and $\mathbf{Y}$. Therefore, CCA only considers the *inter*correlation between $\mathbf{v}_x$ and $\mathbf{v}_y$, but ignores the *intra*correlation of either (especially $\mathbf{y}$). Instead, MORP takes into account all the inter- and intradependencies since the projections minimize the reconstruction error of inputs and outputs simultaneously.

Another related approach is PLS, which was originally developed for regression problems in chemometrics ([16], [25]). PLS aims at finding orthogonal projection directions for inputs $\mathbf{X}$, each of which maximizes the covariance between the outputs $\mathbf{Y}$ and a linear combination of $\mathbf{X}$:

$$\max_{\mathbf{v}_x \in \mathbb{R}^N} \quad \mathrm{Cov}(\mathbf{v}_x, \mathbf{Y})$$
$$\text{subject to :} \quad \mathbf{v}_x = \mathbf{X}\mathbf{w}_x, \mathbf{w}_x^\top \mathbf{w}_x = 1.$$

PLS can be seen as a penalized CCA since covariance is simply correlation weighted by the square root of variance. Tikhonov and Arsenin [6] pointed out that PLS cannot find a space of larger dimensionality than that of $\mathbf{Y}$; thus, its generalization performance on new dimensions of outputs is restricted. Instead, our method can find, in principle, N orthogonal dimensions (if $\mathbf{G}_x$ is positive definite).

4.4 Kernel Dependency Estimation (KDE)

MORP is also related to kernel dependency estimation, a two-stage method for discovering dependency between possibly nonlinear mappings of inputs and outputs [18]. In the first step, KDE performs kernel PCA on output $\mathbf{Y}$ and obtains some principal components $\mathbf{v}_y$; then, a regression model with kernels (kernel ridge regression in [18]) is applied to $\mathbf{X}$ for projections of $\mathbf{Y}$ to each $\mathbf{v}_y$. No explicit mappings for $\mathbf{X}$ are available and a certain cost function has to be defined and minimized to find the output for a text point the (so-called "pre-image" problem).

If applied to regression, MORP has similar behavior if $\beta = 1$: We are looking for a projection of $\mathbf{X}$ that seems to be enforced to entirely explain the dependency of outputs, as can be seen from (2). However, it turns out to be not true if we introduce the regularization to prevent overfitting since the Lagrange formalism of minimizing the regularizer $\boldsymbol{\alpha}^\top \mathbf{G}_x \boldsymbol{\alpha}$, under the constraint $\boldsymbol{\alpha}^\top \mathbf{G}_x^2 \boldsymbol{\alpha} = 1$, tends to a kernel PCA of input features. To see it more clearly, recall that $\mathbf{v} = \mathbf{G}_x \boldsymbol{\alpha}$ and, thus, we can write the cost function in (10) equivalently as

$$\max_{\mathbf{v} \in \mathbb{R}^N} \quad \mathbf{v}^\top \left(\mathbf{G}_y^{-1} + \gamma \mathbf{G}_x^{-1} \right)^{-1} \mathbf{v}, \tag{15}$$

where $\mathbf{G} = \mathbf{G}_y$ holds when $\beta = 1$ and we change the minimization to maximization by adding inversion to the matrix sum. Geometrically, (15) forces $\mathbf{v}$ to be close to the eigenvector of $\mathbf{G}_y$ as well as that of $\mathbf{G}_x$, both with the largest eigenvalue. Therefore, in this special case, MORP is performing *input regularized kernel PCA* for output $\mathbf{Y}$, while finally obtaining a mapping for $\mathbf{X}$ explicitly. Compared to the two-step approach taken by KDE, MORP can be a *feature mapping* step for regression models and provides a more elegant and direct way for multi-output regression.

4.5 Multitask Learning

The work is also related to the recent research on multitask learning (e.g., [20], [21], [22]), which learns many related predictive tasks together by exploring their dependency. We can first use the proposed algorithm to map the input features into a new space and then treat each task independently using the new representatives as input features. This two-stage solution can more easily deal with new tasks, while multitask learning has to retrain all the tasks once new tasks need to be handled.

5 Empirical Study

In this section, we evaluate the proposed MORP algorithm based on two settings. The first is *prediction of user preferences*, in which we predict users' preferences on some data based on both the content features and rankings of other users. If we take each user as one output for all the data, we can think of this setting as a natural multi-output problem, where common interests of users stand for the intracorrelation among outputs. The second is to perform *multilabel classification* on the projected space, taking MORP as a preprocessing or feature transformation step. In this setting, we allow one data object to belong to multiple categories and, therefore, different classification problems could have correlations between each other. This information will be utilized in MORP for deriving the mapping.

5.1 Prediction of User Preferences

Our first experiment is performed on a painting database which contains 642 paintings from 47 artists. A Web-based online survey is built to gather user ratings. For all the paintings, we extract and combine *color histogram* (216-dim.), *correlagram* (256-dim.), *first and second color moments* (9-dim.), and *Pyramid wavelet texture* (10-dim.) to form 491-dimensional input features to represent the images. All the features are then centralized and standardized with deviation 1. For the online survey, each user gave ratings, i.e., "like" or "dislike," to a randomly selected subset of paintings. Finally, we obtained a total of $L = 190$ users' ratings encoded as $+1$ and -1. On average, each user has rated 89 paintings, and each painting was rated by 30 users.

5.1.1 Experimental Settings

In the experiment, a set of users are selected as *test users*, and 10-fold cross-validation is performed for each test user with one fold training and nine folds testing. An SVM using RBF kernel with all 491 image features can be trained for each test user, and this is denoted as ORIGINAL FEATURES and serves as the baseline. We will basically compare three projection methods. KERNEL PCA performs unsupervised projection and maps the input data into a low-dimensional space. The two supervised methods, MORP and KERNEL CCA, additionally make use of the rating information of the other users. All three of the competing methods use the same RBF kernel as in ORIGINAL FEATURES and same dimensionality. The new features given by these methods are then fed into a linear SVM for classification.

We choose all the parameters for these algorithms as follows: The RBF kernel width is $\sigma = 25$, which gives ORIGINAL FEATURES the best performance and is then fixed for all the projection methods. Different values for dimensionality K yield similar comparison results between these projection methods and, for simplicity, we fix $K = 50$. In MORP, β is simply chosen as 0.5 to give equivalent weights to $\mathbf{G}_x$ and $\mathbf{G}_y$, after we scale $\mathbf{G}_x$ and $\mathbf{G}_y$ to ensure they have equal traces for balance. γ is insensitive to the result and is simply fixed as 1. For KERNEL CCA, we tune the regularization parameter for best performance and set it to be 0.9.

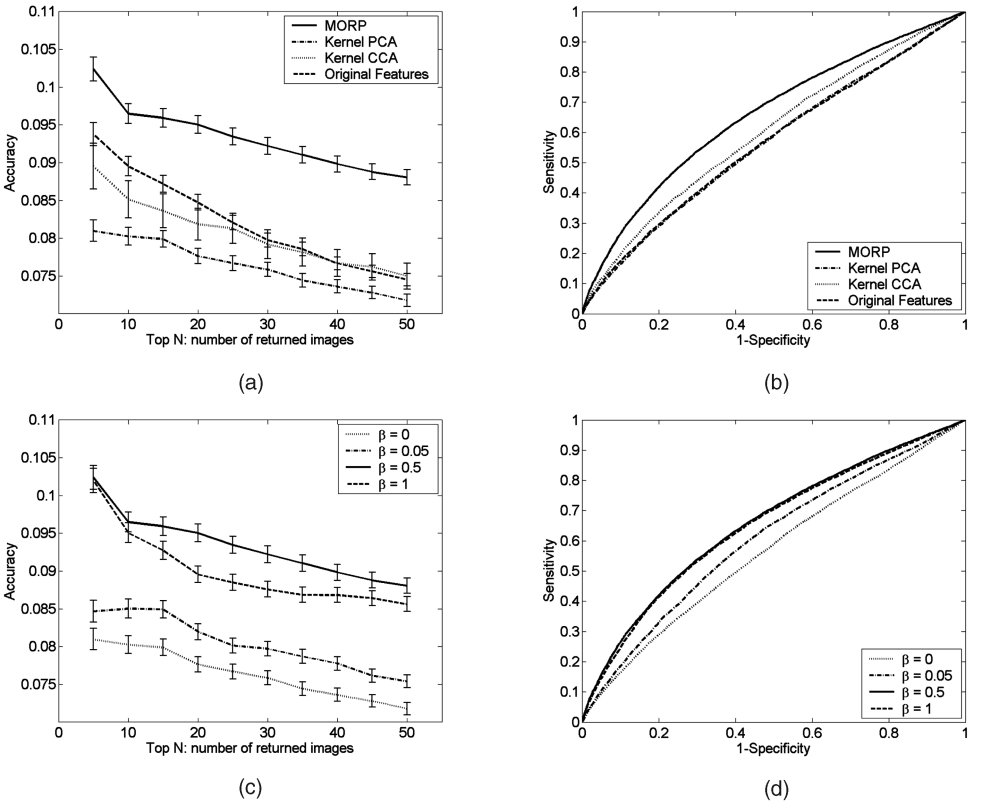

Fig. 2. Comparison of algorithms for predicting user preferences. (a), (c) show the mean and standard deviation of prediction accuracy at different top number of returned images, and (b), (d) show the corresponding ROC curve, i.e., Sensitivity versus 1-Specificity. The upper row compares four methods: MORP ($\beta = 0.5$, $\gamma = 1$), KERNEL PCA, KERNEL CCA (regularization parameter 0.9), and ORIGINAL FEATURES (with SVM). The RBF kernel is used with $\sigma = 25$ for all kernel methods. All the projection methods use latent space with $K = 50$. The lower row compares MORP algorithms with different β values, where we have scaled $\mathbf{G}_x$ and $\mathbf{G}_y$ to ensure they have equal traces for balance. γ and K are chosen the same as in the upper row.

5.1.2 Comparison Metrics

These algorithms are evaluated using two metrics. One is *Top-N accuracy*, i.e., the proportion of truly liked paintings among the N top-ranked paintings. Since normal users only care about the quality of first returned items, this quantity reflects the *subjective* quality of an information filtering system. The other is the *ROC* (receiver operating characteristics) *curve*, which plots *sensitivity* versus 1-*specificity*. Sensitivity is defined as the probability that a good painting is recommended by the system, and specificity is the probability that a disliked painting is rejected by the system. By changing the cut point (e.g., return top 10 or 20 paintings), a curve can be plotted. The area under the curve (AUC) measures the *objective* quality of ranking. A higher AUC indicates a better ranking.

5.1.3 Results

The performances of the four algorithms are shown in Fig. 2a and 2b, which clearly indicate that MORP significantly outperforms the rest in terms of both Top-N accuracy and ROC curve. The unsupervised methods ORIGINAL FEATURES and KERNEL PCA give unsatisfying results due to their ignorance of the correlation between user ratings. ORIGINAL FEATURES performs better than KERNEL PCA because it considers all the features for paintings. The other supervised method, KERNEL CCA, is unsuccessful in this setting, but obtains slightly better results than unsupervised methods in terms of ROC curve.

Our method can be seen as a way to combine content-based filters and collaborative filters. The two-stage treatment first learns a feature mapping based on many users' ratings and then uses the new features to feed content-based filters. The parameter β controls the trade-off between the *content-based kernel* $\mathbf{G}_x$ and the *preference kernel* $\mathbf{G}_y$. In the second experiment, we study the impact of β in the performance of preference prediction, as shown in Fig. 2c and Fig. 2d (as before, we scale $\mathbf{G}_x$ and $\mathbf{G}_y$ to ensure they have equal traces for balance). With $\beta = 0$, MORP is indeed kernel PCA and making an unsupervised projection, which gives a bad performance. As β increases gradually, the performance improves significantly, as shown here when $\beta = 0.05$. This clearly shows that the quality of projection has been improved by exploring the correlation among users. $\beta = 0.5$ gives the best results, corresponding to an even balance between the eigenspaces of the two matrices. When β increases further, the performance drops down and overfitting occurs if no information of input is considered.

5.1.4 Visualization

In the last experiment, we visualize the projections of paintings in the first two dimensions and see if we observe interesting distributions. As shown in Fig. 3, we visualize four artists' paintings, Dali, Van Gogh, Monet, and Asian (an anonymous Asian painter). We denote points with different shapes and colors for paintings of different

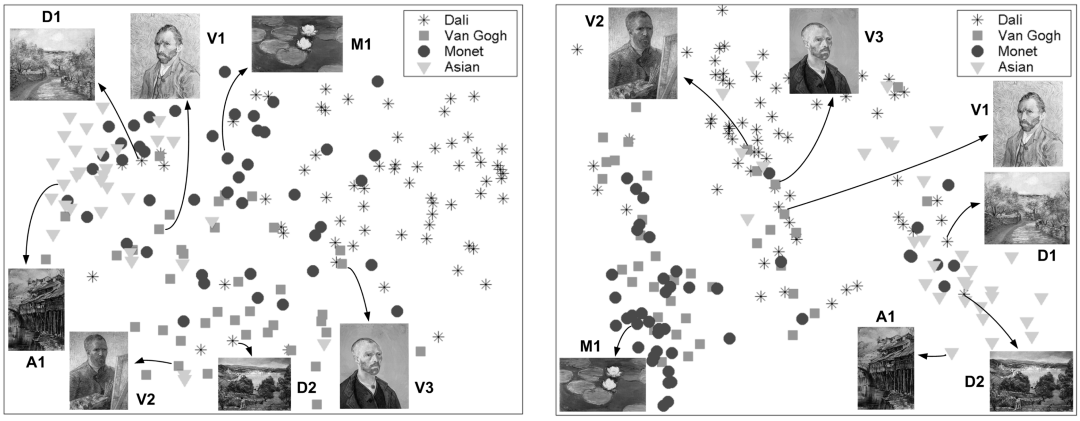

Fig. 3. Visualization of paintings in the first two projected dimensions for (a) KERNEL PCA and (b) MORP. Four painters are compared and several images are shown with annotations. ("D" for Dali, "V" for Van Gogh, "M" for Monet, and "A" for the Asian painter. Numbers show indices for each painter.) Parameter settings are the same as in Fig. 2a.

painters and illustrate some images for clear explanation. The annotation beside each image clarifies the corresponding painter. For KERNEL PCA (Fig. 3a), projections are built based on the low-level features of images, and paintings of different painters are somehow separted from each other (e.g., Dali's on the right and the Asian's on the left). This recovers the unsupervised characteristic of KERNEL PCA. However, for the specific task of user preferences prediction, this is not sufficient. It is more interesting to investigate the patterns found by MORP (Fig. 3b), which forces the projection to reflect user correlations. Let's take a close look at the differences between Fig. 3a and Fig. 3b. Roughly speaking, there are three groups, left, middle, and right, in Fig. 3b. First of all, it appears that the paintings of Van Gogh and Monet frequently stay close (in the left group), indicating that people often have similar preferences for these two artists' works, i.e., a user either likes both or dislikes both. Second, Van Gogh's self-portraits (annotated as V1, V2, and V3) stay very close to users' preferences (middle group), but it is interesting that they seem to be outliers in his paintings, and people's preferences for them are more correlated to the opinions of Dali's works. Furthermore, Dali's paintings in the early years (e.g., the two marked out as D1 and D2, painted in 1922) substantially differ from the majority of his works in style. Instead, D1 and D2 stay close to the Asian's paintings, which are mainly about houses and buildings in the countryside. Though a rigorous interpretation of the visualized distribution is lacking, we can still conclude that MORP maps paintings into a very meaningful space which will be beneficial for predicting interests for new users.

5.2 Multilabel Classification

The experiment in this section is based on two text and one image data sets. The first text data is taken from Reuters-21578, which contains all the documents with multiple categories. Eliminating those minor categories that contain less than 50 documents, we have 47 categories to work with. Picking up all the words that occur at least in five documents, we finally obtain 1,600 documents with 6,076 words. On average, each document is assigned to 2.48 categories and each category has 85 positive documents.

The second text data is a subset of the RCV1-v2 data set, provided by Reuters and corrected by Lewis et al. [26]. Since it is common that one document is assigned to multiple topics, this is an ideal data set for multilabel classification. After the same preprocessing, we finally obtain 3,588 documents with 5,496 words and have 79 topics left. On average, each topic contains 180 positive documents, and each document belongs to 3.96 topics. Standard TFIDF features are then computed for these two text data sets.

Our last data is a subset of the Corel image database, which contains 1,021 images. We manually labeled them into 37 categories. On average, each image belongs to 3.6 categories and each category contains 98 images. As in the previous painting case, we extract the same 491-dimensional features as the input features for images and centralize them.

In the following, we denote "Reuters," "RCV1," and "Corel" for these data sets, respectively.

5.2.1 Experimental Settings

In the first setting (I), we randomly pick up 70 percent of the categories for classification and employ 5-fold cross-validation with one fold training and four folds testing. This is a standard classification setting, and our goal is to evaluate whether the feature mappings are generalizable to new data points. We will test the four algorithms described in the previous section, i.e., ORIGINAL FEATURES, KERNEL PCA, MORP, and KERNEL CCA. ORIGINAL FEATURES still serves as the baseline, KERNEL PCA defines unsupervised mappings, and the latter two give supervised mappings. Note that this setting is actually a *batch* version of many single-output binary classification tasks, where the performance is averaged over all tasks.

We also have a second setting (II), which aims to test the generalization ability of the projection methods on new categorization tasks. For this, we consider the classification problems for the remaining 30 percent of categories. To make a fair comparison, we perform 5-fold cross-validation on previous unseen data, using the feature mappings derived from setting (I). We will also compare all four methods in this setting.

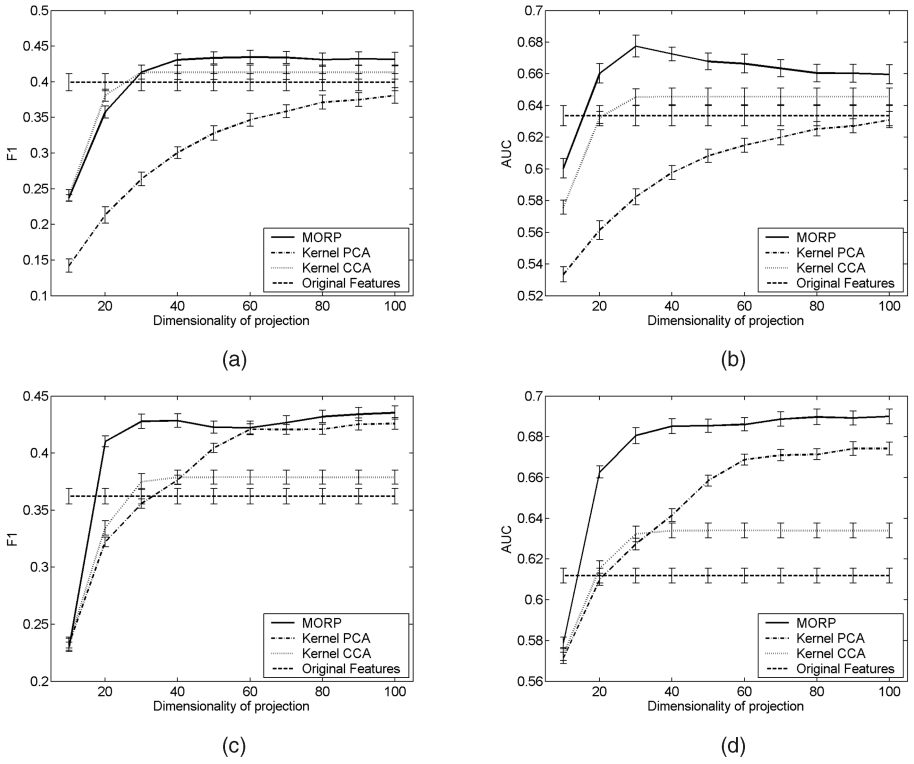

Fig. 4. Classification performance on Reuters. (a), (b) show results with setting (I), and (c), (d) show results with setting (II).

We use the RBF kernel with width $\sigma = 25$ for the Corel data (which gives ORIGINAL FEATURES the best performance) and use linear kernels for the text data sets. For MORP, we set the parameter β to 0.5 after rescaling $\mathbf{G}_x$ and $\mathbf{G}_y$ and fix γ as 0. The regularization parameter for KERNEL CCA is tuned for each data set and set to 0.9, 0.3, and 0.3 for Reuters, RCV1, and Corel, respectively. For both settings, we repeat the experiments 10 times with randomization, and the performance versus dimensionality of projection is shown with means and standard deviations in Fig. 4, Fig. 5, Fig. 6 for Reuters, RCV1, and Corel, respectively.

5.2.2 Comparison Metrics

The classification performance is compared using F_1 measure and AUC score. The F_1 measure defines a trade-off between precision and recall and is known to be a good metric for classification evaluation. Alternatively, the AUC score measures the quality of ranking for specific classification problems. Both of these scores are averaged over all the output dimensions. We also tried classification accuracy, but didn't get the informative comparison because most of the classification problems are very unbalanced (more than 90 percent of data are negative examples).

5.2.3 Results

The first observation from these figures is that MORP outperforms KERNEL PCA in almost all the cases. This indicates that the mapping functions in MORP are generalizable to new test data for setting (I) and also generalizable to new related prediction tasks, as seen in setting (II). The difference is especially big for setting (I), where the predictions are made for the known categories. By incorporating the output information for the training data, MORP can obtain more informative mappings for these specific tasks.

KERNEL CCA also performs a supervised projection and, in general, it obtains worse but comparable results as MORP in setting (I). However, the performance is quite bad for setting (II) and, in most cases, it is even worse than KERNEL PCA. This indicates that KERNEL CCA suffers from overfitting and is not generalizable to new prediction tasks. It can also be seen that KERNEL CCA approaches a constant performance after a small number of dimensions. The reason is that KERNEL CCA could only extract pairs of mappings (one for $\mathbf{X}$ and the other for $\mathbf{Y}$) and, thus, could not obtain more dimensions than the number of outputs for training. This is very limited when we want the mappings generalizable to new outputs. In contrast, MORP does not have this problem and in general could extract N directions.

Another observation from these figures is that projected data can lead to better classification performance than ORIGINAL FEATURES that simply uses all the original features. This is especially the case in setting (II), where a large gap can be observed for all projection methods, even for the unsupervised method KERNEL PCA. This suggests that projecting input data into a low-dimensional space can not only accelerate the classification tasks, but also improve the performance. Therefore, it is of great importance to derive a good projection method for supervised learning. MORP is seen to outperform all the other methods in setting (II) and, thus, is a very good choice.

5.2.4 Parameter Sensitivity

MORP has two tunable parameters, β and γ, that control the kernel combination weights and the strength of regulariza-

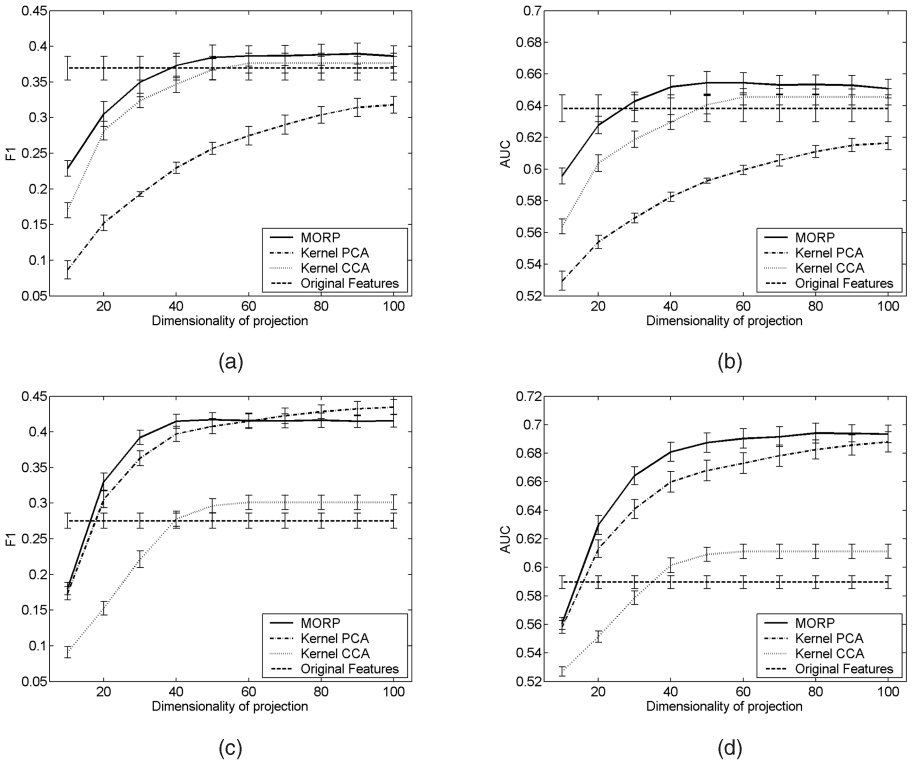

Fig. 5. Classification performance on RCV1. (a), (b) show results with setting (I), and (c), (d) show results with setting (II).

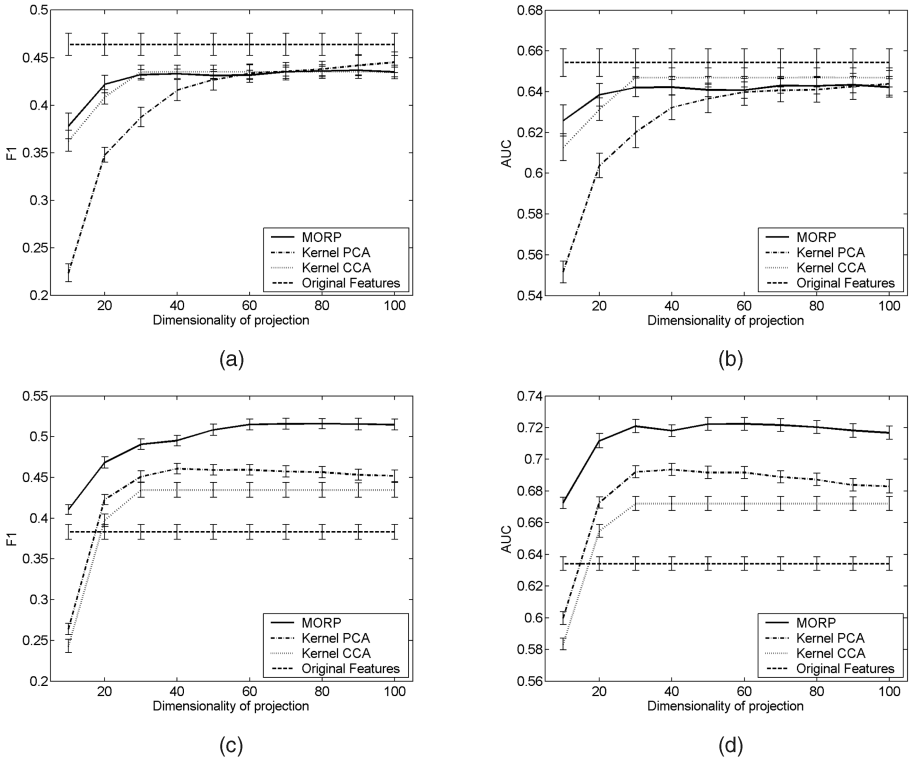

Fig. 6. Classification performance on Corel. (a), (b) show results with setting (I), and (c), (d) show results with setting (II).

tion, respectively. For previous figures, it is assumed fixed and, in this last experiment, we study the classification performance when they are varied. Since we can see similar results for the three data sets, we only show in Fig. 7 the illustrations for Reuters with AUC scores. Figures for β are shown with dimensionality K fixed as 50 since it is insensitive to the results.

A first impression from Fig. 7 is that the curves are rather smooth (except when β approaches 1 in setting (II)). This indicates that the performance is not very sensitive to small

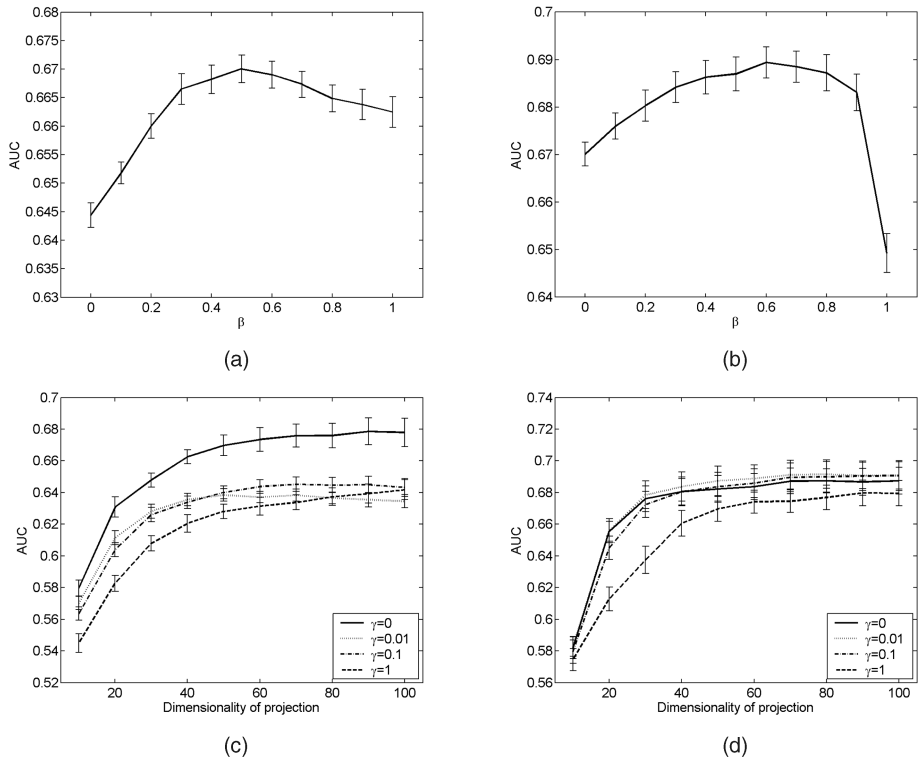

Fig. 7. AUC performance of MORP with respect to β (upper row) and γ (lower row) for the Reuters data set. (a), (c) show results with setting (I) and (b), (d) show results with setting (II). All the β values are chosen after we scale $\mathbf{G}_x$ and $\mathbf{G}_y$ to have equal traces.

changes of β value. When β increases from 0 to 1, it is seen that all the curves first increase and then decrease, indicating that a good trade-off should be identified for best performance. When β approaches 0, MLSI tends to be LSI and, thus, unsupervised. Outputs are ignored in this case and poor performance is observed for both settings. On the other hand, when β approaches 1, the mappings tend to solely explain outputs $\mathbf{Y}$, ignoring the intrinsic structure of inputs $\mathbf{X}$. This also leads to poor performance, especially for setting (II), because the mappings are not good to generalize to new outputs. Overfitting occurs in this case, where a sharp decrease can be observed with even a much worse performance than LSI ($\beta = 0$). Finally, $\beta = 0.5$ is seen to be a good trade-off for both settings. From our experience, a slightly larger β (e.g., 0.6) is better for setting (I) and a slightly smaller β (e.g., 0.4) is more stable for setting (II).

For γ, we have the observation that small γ leads to better performance for setting (I), while an appropriately chosen γ is necessary for setting (II). This reflects its regularization effect since, for setting (II), new categories are considered and setting $\gamma = 0$ will lead to overfitting.

6 Summary and Conclusions

In this paper, we propose a novel feature mapping algorithm MORP for multi-output regularized feature projection. The projections are supervised and retain the statistical information of not only input features but also the (possibly multivariate) outputs. We present both the primal and the dual formalisms for the linear mappings such that nonlinear mappings can be derived by using reproducing kernels. The final solution ends up as a simple generalized eigenvalue problem that can be easily solved. The algorithm is applied for user preference prediction and multilabel classification, both with very encouraging results. Currently, we are mainly exploiting linear dependency of outputs. In the near future, we plan to apply other types of kernels to explore richer structured outputs.

Appendix A

Proof of Proposition 2

Let $J(\mathbf{A}, \mathbf{B}, \mathbf{V}) := (1-\beta)\|\mathbf{X} - \mathbf{V}\mathbf{A}\|^2 + \beta\|\mathbf{Y} - \mathbf{V}\mathbf{B}\|^2$. Setting the derivative of J with respect to $\mathbf{A}$ and $\mathbf{B}$ to be zero and applying $\|\mathbf{C}\|^2 = \mathtt{tr}(\mathbf{C}\mathbf{C}^\top)$ for any matrix $\mathbf{C}$, we obtain

$$\frac{\partial J}{\partial \mathbf{A}} = 2(1-\beta)(\mathbf{V}^\top\mathbf{X} - \mathbf{V}^\top\mathbf{V}\mathbf{A}) = 0 \Rightarrow \mathbf{A} = \mathbf{V}^\top\mathbf{X},$$
$$\frac{\partial J}{\partial \mathbf{B}} = 2\beta(\mathbf{V}^\top\mathbf{Y} - \mathbf{V}^\top\mathbf{V}\mathbf{B}) = 0 \Rightarrow \mathbf{B} = \mathbf{V}^\top\mathbf{Y},$$

which proves conclusion 1. Then, we use the results 1 to replace $\mathbf{A}$ and $\mathbf{B}$ in J and obtain conclusion 2. □

Appendix B

Proof of Theorem 3

Denote $\tilde{\mathbf{V}} = [\tilde{\mathbf{v}}_1, \ldots, \tilde{\mathbf{v}}_K]$. The Lagrange formalism of (3) is

$$L(\tilde{\mathbf{V}}, \tilde{\mathbf{\Lambda}}) = \sum_{i=1}^{K} \tilde{\mathbf{v}}_i^\top \mathbf{K}\tilde{\mathbf{v}}_i - \sum_{i=1}^{K} \tilde{\lambda}_{i,i}(\tilde{\mathbf{v}}_i^\top\tilde{\mathbf{v}}_i - 1) - 2\sum_{i>j} \tilde{\lambda}_{i,j}\tilde{\mathbf{v}}_i^\top\tilde{\mathbf{v}}_j,$$

where $(\tilde{\mathbf{\Lambda}})_{i,j} = \tilde{\lambda}_{i,j}$ is a symmetric matrix if we define $\tilde{\lambda}_{i,j} = \tilde{\lambda}_{j,i}$ for $i < j$. Setting its derivative with respect to $\tilde{\mathbf{v}}_i$ to be zero, we obtain

$$\frac{\partial L}{\partial \tilde{\mathbf{v}}_i} = 2\mathbf{K}\tilde{\mathbf{v}}_i - 2\sum_{j=1}^{K} \tilde{\lambda}_{i,j}\tilde{\mathbf{v}}_j = 0, \quad i = 1, \ldots, K,$$

which can be rewritten as $\mathbf{K}\tilde{\mathbf{V}} = \tilde{\mathbf{V}}\tilde{\mathbf{\Lambda}}$. Since $\tilde{\mathbf{\Lambda}}$ is a symmetric matrix, we have $\tilde{\mathbf{\Lambda}} = \mathbf{R}^\top \mathbf{\Lambda} \mathbf{R}$, where $\mathbf{\Lambda}$ is a diagonal matrix and $\mathbf{R} \in \mathbb{R}^{K \times K}$ is an orthogonal rotation matrix satisfying $\mathbf{R}\mathbf{R}^\top = \mathbf{R}^\top\mathbf{R} = \mathbf{I}$. Then,

$$\mathbf{K}\tilde{\mathbf{V}} = \tilde{\mathbf{V}}\mathbf{R}^\top\mathbf{\Lambda}\mathbf{R} \quad \Rightarrow \quad \mathbf{K}\tilde{\mathbf{V}}\mathbf{R}^\top = \tilde{\mathbf{V}}\mathbf{R}^\top\mathbf{\Lambda}.$$

Since $\mathbf{\Lambda}$ is diagonal, it is easy to see that the columns of $\mathbf{V} = \tilde{\mathbf{V}}\mathbf{R}^\top$ are the eigenvectors of $\mathbf{K}$. Thus, the optimal $\tilde{\mathbf{V}}$ is formed by an arbitrary rotation of $\mathbf{K}$'s eigenvectors, i.e., $\tilde{\mathbf{V}} = \mathbf{V}\mathbf{R}$. Inserting $\tilde{\mathbf{V}}$ back into the objective function, we have the value of the objective function as $\mathtt{tr}(\mathbf{\Lambda})$, i.e., sum of the K corresponding eigenvalues of $\mathbf{K}$. It is easy to see that the maximal $\mathtt{tr}(\mathbf{\Lambda})$ is the sum of the K largest eigenvalues, which proves conclusion 2. In this case, $\tilde{\mathbf{V}}$ is an arbitrary rotation of the K largest eigenvectors, thus conclusion 1 holds. □

APPENDIX C
PROOF OF THEOREM 4

Let $J(\mathbf{w})$ denote the cost function in (6), i.e.,

$$J(\mathbf{w}) := \mathbf{w}^\top\mathbf{X}^\top\mathbf{K}^{-1}\mathbf{X}\mathbf{w} + \gamma\|\mathbf{w}\|^2.$$

Obviously, $J(\mathbf{w})$ achieves the minimum at the first eigenvector $\mathbf{w}$ of the generalized eigenvalue problem (7). Denote $\mathbf{w}_\parallel$ as the projection of $\mathbf{w}$ on the subspace

$$\mathtt{span}\{\mathbf{x}_1, \ldots, \mathbf{x}_N\},$$

then we can write $\mathbf{w} = \mathbf{w}_\parallel + \mathbf{w}_\perp$, where $\mathbf{w}_\perp$ is orthogonal to the subspace. Compare $J(\mathbf{w}_\parallel)$ with $J(\mathbf{w})$. We have

$$\mathbf{w}^\top\mathbf{x}_i = \mathbf{w}_\parallel^\top\mathbf{x}_i + \mathbf{w}_\perp^\top\mathbf{x}_i = \mathbf{w}_\parallel^\top\mathbf{x}_i,$$

so $\mathbf{X}\mathbf{w}_\parallel = \mathbf{X}\mathbf{w}$, which means $J(\mathbf{w}_\parallel)$ and $J(\mathbf{w})$ agree on the first term. Since $\|\mathbf{w}\|^2 = \|\mathbf{w}_\parallel\|^2 + \|\mathbf{w}_\perp\|^2 \geq \|\mathbf{w}_\parallel\|^2$, $J(\mathbf{w}) \geq J(\mathbf{w}_\parallel)$ holds. However, this must be an equation since $J(\mathbf{w})$ achieves the minimum. Therefore, we have $\|\mathbf{w}_\perp\| = 0$ and, hence $\mathbf{w}_\perp = 0$, which means $\mathbf{w}$ is actually a linear combination of $\mathbf{x}_i$, $i = 1, \ldots, N$.

So far, we have proved the theorem for the first eigenvector (with the smallest eigenvalue). Given eigenvectors $\mathbf{w}_j$, $j = 1, \ldots, n-1$, it is known that the nth eigenvector is obtained by first deflating the matrix $\mathbf{K}^{-1}$ with $\mathbf{K}^\dagger = \mathbf{K}^{-1} - \sum_{j=1}^{n-1} \lambda_j \mathbf{X}\mathbf{w}_j\mathbf{w}_j^\top\mathbf{X}^\top$ and then solving the following problem:

$$\begin{aligned} \min_{\mathbf{w} \in \mathbb{R}^M} \quad & \mathbf{w}^\top\mathbf{X}^\top\mathbf{K}^\dagger\mathbf{X}\mathbf{w} + \gamma\|\mathbf{w}\|^2 \\ \text{subject to :} \quad & \mathbf{w}^\top\mathbf{X}^\top\mathbf{X}\mathbf{w} = 1. \end{aligned}$$

Following the same procedure as before, we can prove that the eigenvector $\mathbf{w}_n$ also lies in the span of $\mathbf{x}_i$, $i = 1, \ldots, N$. □

ACKNOWLEDGMENTS

Shipeng Yu gratefully acknowledges the support of a Siemens scholarship. The authors thank all of the anonymous reviewers for their valuable comments and suggestions which improved the quality of this paper.

REFERENCES

[1] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning.* Springer Verlag, 2001.
[2] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 7, pp. 711-720, July 1997.
[3] S.C. Deerwester, S.T. Dumais, T.K. Landauer, G.W. Furnas, and R.A. Harshman, "Indexing by Latent Semantic Analysis," *J. Am. Soc. Information Science,* vol. 41, no. 6, pp. 391-407, 1990.
[4] M.W. Berry, S.T. Dumais, and G.W. O'Brien, "Using Linear Algebra for Intelligent Information Retrieval," *SIAM Rev.,* vol. 37, no. 4, pp. 573-595, 1995.
[5] C. Basu, H. Hirsh, and W.W. Cohen, "Recommendation as Classification: Using Social and Content-Based Information in Recommendation," *Proc. 15th Nat'l Conf. Artificial Intelligence AAAI/IAAI,* pp. 714-720, 1998.
[6] A.N. Tikhonov and V.Y. Arsenin, *Solutions of Ill-Posed Problems.* New York: Wiley, 1977.
[7] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," *Neural Computation,* vol. 10, pp. 1299-1319, 1998.
[8] H. Zou, T. Hastie, and R. Tibshirani, "Sparse Principal Component Analysis," technical report, Statistics Dept., Stanford Univ., 2005.
[9] V. Vapnik, *The Nature of Statistical Learning Theory.* New York: Springer, 1995.
[10] G. H. Golub and C.F. Van Loan, *Matrix Computations.* The Johns Hopkins Univ. Press, 1996.
[11] K. Mardia, J. Kent, and J. Bibby, *Multivariate Analysis.* Academic Press, 1979.
[12] B. Schölkopf, A.J. Smola, and K.-R. Müller, "Kernel Principal Component Analysis," *Advances in Kernel Methods—Support Vector Learning,* pp. 327-352, 1999.
[13] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis.* Cambridge Univ. Press, 2004.
[14] H. Hotelling, "Relations between Two Sets of Variables," *Biometrika,* vol. 28, pp. 321-377, 1936.
[15] D.R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical Correlation Analysis; An Overview with Application to Learning Methods," technical report, Royal Holloway Univ. of London, 2003.
[16] H. Wold, "Soft Modeling by Latent Variables; The Nonlinear Iterative Partial Least Squares Approach," *Perspectives in Probability and Statistics, Papers in Honour of M.S. Bartlett,* 1975.
[17] R. Rosipal and L.J. Trejo, "Kernel Partial Least Squares Regression in Reproducing Kernel Hilbert Space," *J. Machine Learning Research,* vol. 2, no. 12, pp. 97-123, 2001.
[18] J. Weston, O. Chapelle, A. Elisseeff, B. Schölkopf, and V. Vapnik, "Kernel Dependency Estimation," *Advances in Neural Information Processing Systems 15,* S. Thrun, S. Becker, and K. Obermayer, eds., MIT Press, 2003.
[19] K. Fukumizu, F.R. Bach, and M.I. Jordan, "Dimensionality Reduction for Supervised Learning with Reproducing Kernel Hilbert Spaces," *J. Machine Learning Research,* vol. 5, pp. 73-99, Jan. 2004.
[20] T. Evgeniou and M. Pontil, "Regularized Multi-Task Learning," *Proc. ACM SIGKDD,* 2004.
[21] K. Yu, V. Tresp, and S. Yu, "A Nonparametric Hierarchical Bayesian Framework for Information Filtering," *Proc. 27th Ann. Int'l ACM SIGIR Conf.,* 2004.
[22] A. Schwaighofer, V. Tresp, and K. Yu, "Hierarchical Bayesian Modelling with Gaussian Processes," *Advances in Neural Information Processing Systems 17,* MIT Press, 2005.
[23] B. Schölkopf and A.J. Smola, *Learning with Kernels.* MIT Press, 2002.
[24] K. Fukunaga, *Statistical Pattern Recognition,* second ed. Academic Press, 1990.

[25] H. Wold, "Partial Least Squares," *Encyclopedia of the Statistical Sciences,* pp. 581-591, 1985.
[26] D.D. Lewis, Y. Yang, T. Rose, and F. Li, "RCV1: A New Benchmark Collection for Text Categorization Research," *J. Machine Learning Research,* vol. 5, pp. 361-397, 2004.

Shipeng Yu received the BSc and MSc degrees in mathematics in 2000 and 2003, respectively, from Peking University, China. He is a PhD student at the Institute for Computer Science at the University of Munich. His research work is supported through a scholarship from Siemens Corporate Technology in Munich. He has been working in information retrieval, data mining, Web-based CRP systems, and network security. Currently, his research interests mainly focus on statistical machine learning and its applications in data mining, information and image retrieval, and user modeling.

Kai Yu received the BSc and MSc degrees in 1998 and 2000, respectively, from Nanjing University, China, and the PhD degree from the Computer Science Department at the University of Munich, Germany, in 2004. He joined Siemens AG in January 2004, where he is currently a senior research scientist. His major research interests are probabilistic machine learning models and their applications to information retrieval, user modeling, Web mining, and data mining. He has played key roles in several Siemens industrial R&D projects in the fields of Web mining, personalization, telecommunication, and clinical data analysis. He has published scientific research papers in many leading international conferences and journals, including the *IEEE Transactions on Knowledge and Data Engineering,* NIPS, ICML, UAI, ECML, SIGKDD, SIGIR, CIKM, CVPR, and *ACM Multimedia*.

Volker Tresp received the Diploma degree in physics from the University of Göttingen, Germany, in 1984, and the MSc and PhD degrees from Yale University, New Haven, Connecticut, in 1986 and 1989, respectively. He joined the central research and development unit of Siemens AG in 1989, where he has been the head of various research teams. In 1994, he was a visiting scientist at the Massachusetts Institute of Technology's Center for Biological and Computational Learning.

Hans-Peter Kriegel received the MS and PhD degrees in 1973 and 1976, respectively, from the University of Karlsruhe, Germany. He is a full professor of database systems at the Institute for Computer Science at the University of Munich and has been the director of the institute since 2003. His research interests are in spatial and multimedia database systems, particularly in query processing, performance issues, similarity search, and parallel systems. Data exploration using visualization led him to the area of knowledge discovery and data mining. Dr. Kriegel has been a chairman and program committee member for many international database conferences. He has published more than 200 refereed conference and journal papers. In 1997, he received the internationally prestigious SIGMOD Best Paper Award for the publication and prototype implementation, "Fast Parallel Similarity Search in Multimedia Databases," together with four members of his research team.