Multi-Label Informed Latent Semantic Indexing

Kai Yu[†], Shipeng Yu[‡], Volker Tresp[†]
[†]Siemens AG, Corporate Technology, Information and Communications, Munich, Germany
[‡]Institute for Computer Science, University of Munich, Germany
kai.yu@siemens.com, volker.tresp@siemens.com,
spyu@dbs.informatik.uni-muenchen.de

ABSTRACT

Latent semantic indexing (LSI) is a well-known unsupervised approach for dimensionality reduction in information retrieval. However if the output information (i.e. category labels) is available, it is often beneficial to derive the indexing not only based on the inputs but also on the target values in the training data set. This is of particular importance in applications with *multiple labels*, in which each document can belong to several categories simultaneously. In this paper we introduce the multi-label informed latent semantic indexing (MLSI) algorithm which preserves the information of inputs and meanwhile captures the correlations between the multiple outputs. The recovered "latent semantics" thus incorporate the human-annotated category information and can be used to greatly improve the prediction accuracy. Empirical study based on two data sets, Reuters-21578 and RCV1, demonstrates very encouraging results.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Indexing methods*

General Terms

Algorithms, Theory, Measurement, Performance

Keywords

Latent Semantic Indexing, Dimensionality Reduction, Supervised Projection, Multi-label Classification

1. INTRODUCTION

Information retrieval and pattern recognition often suffer from the problem of high dimensionality of the data, for the reason of learnability or computational efficiency. Therefore dimensionality reduction in terms of *semantic indexing* or *feature projection* is of great importance and is commonly applied to solve real world problems [2, 1, 5].

Among various methods, latent semantic indexing (LSI) turns out to be a successful approach and is widely applied to document analysis and information retrieval [2]. To apply LSI, documents are represented in a vector space model, and singular value decomposition (SVD) is performed to find the sub-eigenspace with large eigenvalues. It is shown that LSI can find the best subspace in terms of Frobenius norm of matrix. Thus the technology behind LSI is also called principal component analysis (PCA) in the sense that each "latent semantic" can be viewed as a "component" to represent the data (see, e.g. [4]).

LSI is purely *unsupervised* and is not capable to incorporate some additional knowledge. There are at least two reasons for further improvements on this issue. First, considerable information about the content of documents is reflected by document's labels, which is often annotated by human experts. This is particularly the case in the multi-label setting where each document is assigned to multiple categories. The semantic correlations of assignments for variant categories and the hierarchical structure of categories expresses the semantic relationships between documents. Therefore, it is desired to have a LSI technique that can be informed by this additional knowledge and produce semantically more meaningful latent factors.

Second, the unsupervision of LSI leads to results that may be or may not be useful in discriminative analysis like automatic text categorization. However in one specific classification or regression problem, output information is in general very important and should be incorporated into the feature mapping or selection process. In particular we consider problems with *multiple labels*: For an input $\mathbf{x}$ the corresponding output is no longer a scalar but a vector $\mathbf{y} = [y_1, \ldots, y_L]^T$. Thus the text categorization system solves many related tasks at the same time. In this setting the dependencies between multiple labels are worth considering for multivariate data analysis, and can be used to improve the indexing for these specific tasks. Furthermore, training a system with multiple labels might lead to smaller parameter variance and the prediction for a particular label is improved if the labels are correlated.

This setting is very common in real-world applications. One example is the problem of multi-label document categorization, where each document is allowed to be associated with more than one category and where categories often have semantic correlations [8]. The well-known text data set Reuters-21578 contains such documents, and the new text data corpus RCV1 has additionally a topic hierarchy [7]. These two data sets will be used in the experiments.

SIGIR'05, August 15–19, 2005, Salvador, Brazil.
Copyright 2005 ACM 1-59593-034-5/05/0008 ...$5.00.

In this paper we introduce a supervised LSI called multi-label informed latent semantic indexing (MLSI). MLSI maps the input features into a new feature space that retains the information of original inputs and meanwhile captures the dependency of output dimensions. The mapping is derived by solving an optimization problem for linear projections, and can be easily extended for nonlinear mappings with kernels. We use this method as a preprocessing step and achieve encouraging results on the multi-label text classification problems.

1.1 Notations

We consider a set of N documents. For $i = 1, \dots, N$, each document i is described by an M-dimensional feature vector $\mathbf{x}_i \in \mathcal{X}$, and is associated with an L-dimensional output vector $\mathbf{y}_i \in \mathcal{Y}$. We denote the input data as a matrix $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]^T \in \mathbb{R}^{N \times M}$, and the output data as $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_N]^T \in \mathbb{R}^{N \times L}$, where $[\cdot]^T$ denotes matrix transpose. We aim to derive a mapping $\Psi : \mathcal{X} \mapsto \mathcal{V}$ that projects the input features into a K-dimensional latent space.

In the following, lower-case bold Roman letters denote column vectors, and upper-case ones denote matrices. In particular, $\mathbf{I}$ is reserved for identity matrix. Eigenvalues are usually denoted as λ and it should be clear from the context which matrix they are corresponding to. $\|\cdot\|$ denotes Frobenius norm for matrices and 2-norm for vectors, and $\mathrm{Tr}\,[\cdot]$ denotes trace for square matrices.

1.2 Paper Organization

The paper is organized as follows. In Section 2 we formulate the data projection as an optimization problem in the linear case and then propose a regularized version to prevent overfitting, which is generalized to nonlinear mapping by using kernels. Then we point out its connections to related work in Section 3 and report the experimental results in Section 4. In Section 5 we conclude the paper.

2. THE MLSI ALGORITHM

We begin by introducing an optimization explanation for LSI, and then take into account the output information.

2.1 Optimization Problem for LSI

In LSI, 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. We can achieve this by taking a *latent variable model* and solving the following optimization problem which minimizes the *reconstruction error* (see, e.g., [4]):

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

where $\mathbf{V} \in \mathbb{R}^{N \times K}$ and $\mathbf{A} \in \mathbb{R}^{K \times M}$, given $K \leq M$. Each column of $\mathbf{V}$ corresponds to one *latent variable* or *latent semantic*, and by $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ we constrain that they are uncorrelated and each has unit variance[1]. For each document in $\mathcal{X}$ (represented as one row in $\mathbf{X}$), the corresponding row in $\mathbf{V}$ explicitly gives its *projection* in $\mathcal{V}$. $\mathbf{A}$ is sometimes called *factor loadings* and gives the mapping from latent space $\mathcal{V}$ to input space $\mathcal{X}$. At the optimum, $\mathbf{VA}$ leads to the best K-rank approximation of the observations $\mathbf{X}$.

The derived indexing explains the covariance of input data, which is however not necessarily relevant to the output quantities. Thus LSI 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 *intra-correlation* within $\mathbf{Y}$ (if multiple labels). Therefore, we turn to *supervised indexing* in the next subsection, incorporating both input $\mathbf{X}$ and output $\mathbf{Y}$.

2.2 A Supervised LSI

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

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

where $\mathbf{V} \in \mathbb{R}^{N \times K}$ gives the K-dimensional projections of documents, *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 factor loadings for $\mathbf{X}$ and $\mathbf{Y}$, respectively. $0 \leq \beta \leq 1$ is a tuning parameter determining how much the indexing should be biased by the outputs. As before, $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ restricts the K latent variables to be uncorrelated and have unit variance. 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 indexing that gives the minimum reconstruction error. The second part in the objective function of problem (2) enforces the latent semantics to explain the dependency structure of multiple labels. The following theorem states the interdependency between $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{V}$ at the optimum.

THEOREM 1. *Denote* $\mathbf{C} = (1-\beta)\mathbf{X}\mathbf{X}^T + \beta\mathbf{Y}\mathbf{Y}^T$*, and let* $\lambda_1 \geq \ldots \geq \lambda_N$ *be eigenvalues of* $\mathbf{C}$ *with corresponding eigenvectors* $\mathbf{v}_1, \dots, \mathbf{v}_N$*. If* $\mathbf{V}, \mathbf{A}$ *and* $\mathbf{B}$ *are the optimal solutions to problem* (2)*, then:*

(a) $\mathbf{A} = \mathbf{V}^T\mathbf{X}$, $\mathbf{B} = \mathbf{V}^T\mathbf{Y}$*;*

(b) $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_K]\mathbf{R}$*, where* $\mathbf{R}$ *is an arbitrary* $K \times K$ *orthogonal rotation matrix;*

(c) At the optimum, the objective function in (2) *equals to* $\mathrm{Tr}\,[\mathbf{C}] - \mathrm{Tr}\,\left[\mathbf{V}^T\mathbf{C}\mathbf{V}\right]$*, or equivalently,* $\sum_{i=K+1}^{N} \lambda_i$*.*

To improve readability, we put all proofs into Appendix. Theorem 1 states that the leading eigenvectors of $\mathbf{C}$ form a solution for matrix $\mathbf{V}$, and any arbitrary rotation for $\mathbf{V}$ does not change the optimum. Therefore to remove the ambiguity, we focus on the solution given by the leading eigenvectors of $\mathbf{C}$, i.e., $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_K]$. Problem (2) can thus be achieved by solving the eigenvalue problem $\mathbf{C}\mathbf{v} = \lambda\mathbf{v}$ for the

[1] An equivalent version of (1) has the same objective function but instead constraining $\mathbf{A}\mathbf{A}^T = \mathbf{I}$. The difference between the obtained projections and the computed $\mathbf{V}$ in (1) is just a linear scaling caused by the top K singular values of $\mathbf{X}$. Here we consider the form (1) for the convenience of deriving the extensions in next section.

first K leading eigenvectors, which is equivalent to solving[2]:

$$\max_{\mathbf{v}\in\mathbb{R}^N} \quad \mathbf{v}^T\mathbf{C}\mathbf{v} \tag{3}$$
$$\text{subject to}: \quad \mathbf{v}^T\mathbf{v} = 1.$$

Then $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_K]$, $\mathbf{A} = \mathbf{V}^T\mathbf{X}$, and $\mathbf{B} = \mathbf{V}^T\mathbf{Y}$ gives the optimal solution for problem (2).

2.3 MLSI - Primal Form

To complete the MLSI algorithm, we still need to consider two things. Firstly, the indexing should not rely on the labels, since for new documents we have no target information yet. Secondly, the stability of indexing should be taken into account, because otherwise overfitting is likely to occur.

2.3.1 Linear Constraint

It is not hard to see that solving problem (3) only gives the projections for training data with both features in $\mathbf{X}$ and $\mathbf{Y}$. We wish to construct a mapping $\Psi : \mathcal{X} \mapsto \mathcal{V}$ that is able to handle the input features of any new documents, thus we add a linear constraint to problem (2) and restrict the latent variables as *linear mappings* of $\mathbf{X}$, i.e.,

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

Therefore we have $\mathbf{v}_i = \mathbf{X}\mathbf{w}_i$, for $i = 1, \ldots, K$, if we denote $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_K] \in \mathbb{R}^{M\times K}$. Plugging $\mathbf{v} = \mathbf{X}\mathbf{w}$ into (3), we have an optimization problem with respect to $\mathbf{w}$:

$$\max_{\mathbf{w}\in\mathbb{R}^M} \quad \mathbf{w}^T\mathbf{X}^T\mathbf{C}\mathbf{X}\mathbf{w} \tag{4}$$
$$\text{subject to}: \quad \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} = 1.$$

2.3.2 Overfitting and Regularization

Similar to other linear systems, the learned mappings can be unstable when the span$\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ has a lower rank than M, due to the small size of training set or dependence between input features[3]. As a result, a disturbance of $\mathbf{w}$ with an arbitrary $\mathbf{w}^* \perp$ span$\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ does not change the objective function of optimization since $(\mathbf{w} + \mathbf{w}^*)^T\mathbf{x}_i = \mathbf{w}^T\mathbf{x}_i$, but may dramatically change the projections of unseen test documents which are not in the spanned space. To improve the stability, we have to constrain $\mathbf{w}$ in some way.

Suppose rank$(\mathbf{C}) = N$, then maximizing (3) is equivalent to minimizing $\mathbf{v}^T\mathbf{C}^{-1}\mathbf{v}$.[4] We introduce the *Tikhonov regularization* [14] into problem (4) as the following

$$\min_{\mathbf{w}\in\mathbb{R}^M} \quad \mathbf{w}^T\mathbf{X}^T\mathbf{C}^{-1}\mathbf{X}\mathbf{w} + \gamma\|\mathbf{w}\|^2 \tag{5}$$
$$\text{subject to}: \quad \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} = 1,$$

where $\|\mathbf{w}\|^2 = \mathbf{w}^T\mathbf{w}$ is a penalty term and γ is a tuning parameter. 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 2. *If $\mathbf{w}$ is an eigenvector of the generalized eigenvalue problem* (5), *then $\mathbf{w}$ must be a linear combination of $\mathbf{x}_i, i = 1, \ldots, N$, namely*

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

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

Problem (5) is easily solvable by setting the derivative of its Lagrange formulism with respect to $\mathbf{w}$ to be zero. Then we obtain a generalized eigenvalue problem

$$\left[\mathbf{X}^T\mathbf{C}^{-1}\mathbf{X} + \gamma\mathbf{I}\right]\mathbf{w} = \tilde{\lambda}\mathbf{X}^T\mathbf{X}\mathbf{w}, \tag{6}$$

which gives generalized eigenvectors $\mathbf{w}_1, \ldots, \mathbf{w}_M$ with eigenvalues $\tilde{\lambda}_1 \leq \ldots \leq \tilde{\lambda}_M$. Note we sort eigenvalues in a non-decreasing order, since we take the K eigenvectors with the smallest eigenvalues to form the mapping. The first K eigenvectors are used to form the mapping functions as the following

$$\psi_j(\mathbf{x}) = \mathbf{w}_j^T\mathbf{x}, \quad j = 1, \ldots, K, \tag{7}$$

where in this paper we focus on the projection directions and ignore possible scaling factors. As the main results we obtain $\Psi(\mathbf{x}) = [\psi_1(\mathbf{x}), \ldots, \psi_K(\mathbf{x})]^T$ which maps $\mathbf{x}$ into a K-dimensional space.

In problem (6) 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:

$$\mathbf{X}^T\mathbf{X}\mathbf{w} = \lambda\left[\mathbf{X}^T\mathbf{C}^{-1}\mathbf{X} + \gamma\mathbf{I}\right]\mathbf{w}, \tag{8}$$

where we are seeking the K eigenvectors with the *largest* eigenvalues. This gives the MLSI algorithm in primal form, as summarized in Table 1.

Table 1: MLSI in primal form

Input	$\mathbf{X} \in \mathbb{R}^{N\times M}, \mathbf{Y} \in \mathbb{R}^{N\times L}, 0 \leq \beta \leq 1, \gamma \geq 0, K > 0$
Steps	(i) Calculate $\mathbf{C} = (1-\beta)\mathbf{X}\mathbf{X}^T + \beta\mathbf{Y}\mathbf{Y}^T$; (ii) Solve the generalized eigenvalue problem: $\mathbf{X}^T\mathbf{X}\mathbf{w} = \lambda\left[\mathbf{X}^T\mathbf{C}^{-1}\mathbf{X} + \gamma\mathbf{I}\right]\mathbf{w}$, obtain eigenvectors $\mathbf{w}_1, \ldots, \mathbf{w}_K$ with largest K eigenvalues $\lambda_1 \geq \ldots \geq \lambda_K$.
Output	indexing function $\psi_j(\mathbf{x}) = \mathbf{w}_j^T\mathbf{x}, j = 1, \ldots, K$

2.4 MLSI - Dual Form

So far we have considered linear mappings that project inputs $\mathbf{x}$ into a meaningful space $\mathcal{V}$. However, Theorem 2 implies that we can also derive a nonlinear mapping Ψ.

[2] Solving problem (3) itself only gives the first eigenvector $\mathbf{v}_1$ of $\mathbf{C}$. The full optimization problem should be recursively computing $\mathbf{v}_j$ by maximizing $\mathbf{v}^T\mathbf{C}\mathbf{v}$ with the constraint $\mathbf{v}^T\mathbf{v} = 1$ and $\mathbf{v} \perp$ span$\{\mathbf{v}_1, \ldots, \mathbf{v}_{j-1}\}$. Here we state the problem as (3) for simplicity and also because its Lagrange formulism directly leads to the eigenvalue problem.

[3] This will be a crucial problem when we consider nonlinear mapping in the dual form (cf. Section 2.4), since the dimensionality of data point $\mathbf{x}$ in the reproducing kernel Hilbert space (RKHS) could be very high, or even infinite (e.g., in case of RBF kernel). See, e.g., [12].

[4] This equivalence holds whenever $\mathbf{C}$ is positive definite and thus invertible. It is easy to show that matrix $\mathbf{C}$ is at least positive semi-definite, since we have $\mathbf{u}^T\mathbf{C}\mathbf{u} = (1-\beta)\mathbf{u}^T\mathbf{X}\mathbf{X}^T\mathbf{u} + \beta\mathbf{u}^T\mathbf{Y}\mathbf{Y}^T\mathbf{u} = (1-\beta)\|\mathbf{X}^T\mathbf{u}\|^2 + \beta\|\mathbf{Y}^T\mathbf{u}\|^2 \geq 0, \forall\mathbf{u} \in \mathbb{R}^N$. In case that $\mathbf{C}$ is not positive definite, it suffices to use pseudo-inverse instead, or makes it so by adding a tiny positive scalar to diagonal entries.

Let a *kernel* function $k_x(\cdot,\cdot)$ be the inner product in $\mathcal{X}$, i.e., $k_x(\mathbf{x}_i,\mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle = \mathbf{x}_i^T \mathbf{x}_j$, then from Theorem 2,

$$\mathbf{v} = \mathbf{X}\mathbf{w} = \mathbf{X}\mathbf{X}^T\boldsymbol{\alpha} = \mathbf{K}_x\boldsymbol{\alpha},$$

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

$$\|\mathbf{w}\|^2 = \mathbf{w}^T\mathbf{w} = \boldsymbol{\alpha}^T\mathbf{X}\mathbf{X}^T\boldsymbol{\alpha} = \boldsymbol{\alpha}^T\mathbf{K}_x\boldsymbol{\alpha}.$$

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

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

and express the *dual formalism* of problem (5) with respect to coefficients $\boldsymbol{\alpha}$ as

$$\begin{aligned} \min_{\boldsymbol{\alpha}\in\mathbb{R}^N} \quad & \boldsymbol{\alpha}^T\mathbf{K}_x\mathbf{C}^{-1}\mathbf{K}_x\boldsymbol{\alpha} + \gamma\boldsymbol{\alpha}^T\mathbf{K}_x\boldsymbol{\alpha} \qquad (10)\\ \text{subject to :} \quad & \boldsymbol{\alpha}^T\mathbf{K}_x^2\boldsymbol{\alpha} = 1, \end{aligned}$$

which gives rise to again a generalized eigenvalue problem

$$\left[\mathbf{K}_x\mathbf{C}^{-1}\mathbf{K}_x + \gamma\mathbf{K}_x\right]\boldsymbol{\alpha} = \tilde{\lambda}\mathbf{K}_x^2\boldsymbol{\alpha}. \tag{11}$$

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 j-th mapping function, $j = 1,\ldots,K$, is given by

$$\psi_j(\mathbf{x}) = \mathbf{w}_j^T\mathbf{x} = \sum_{i=1}^{N}(\boldsymbol{\alpha}_j)_i k_x(\mathbf{x}_i,\mathbf{x}).$$

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

$$\mathbf{K}_x^2\boldsymbol{\alpha} = \lambda\left[\mathbf{K}_x\mathbf{C}^{-1}\mathbf{K}_x + \gamma\mathbf{K}_x\right]\boldsymbol{\alpha}, \tag{12}$$

and hence we can choose the K eigenvectors with the largest eigenvalues. The MLSI algorithm in dual form is summarized in Table 2.

Table 2: MLSI in dual form

Input	$\mathbf{X}\in\mathbb{R}^{N\times M}, \mathbf{Y}\in\mathbb{R}^{N\times L}, 0\leq\beta\leq 1, \gamma\geq 0, K>0$
Steps	(i) $(\mathbf{K}_x)_{i,j} = k_x(\mathbf{x}_i,\mathbf{x}_j)$, $(\mathbf{K}_y)_{i,j} = k_y(\mathbf{y}_i,\mathbf{y}_j)$, $\mathbf{C} = (1-\beta)\mathbf{K}_x + \beta\mathbf{K}_y$; (ii) Solve the generalized eigenvalue problem: $\mathbf{K}_x^2\boldsymbol{\alpha} = \lambda\left[\mathbf{K}_x\mathbf{C}^{-1}\mathbf{K}_x + \gamma\mathbf{K}_x\right]\boldsymbol{\alpha}$, obtain eigenvectors $\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_K$ with largest eigenvalues $\lambda_1 \geq \ldots \geq \lambda_K$.
Output	indexing function $\psi_j(\mathbf{x}) = \sum_{i=1}^{N}(\boldsymbol{\alpha}_j)_i k_x(\mathbf{x}_i,\mathbf{x})$, $j = 1,\ldots,K$

Several advantages of dual MLSI can be seen from Table 2. First of all, in contrast of solving a generalized eigenvalue problem for $M \times M$ matrices in primal MLSI, in dual MLSI we only need to solve a similar problem for $N \times N$ matrices. In a general indexing problem, the input dimension M (i.e., number of words) is much larger than the number of documents N, and therefore working in dual form is more efficient. In the experiments we will use the dual form for indexing. Second, MLSI in dual form 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 change $\mathbf{X}$ to be $[\phi(\mathbf{x}_1),\ldots,\phi(\mathbf{x}_N)]^T$. Then the kernel function is accordingly defined as

$$k_x(\mathbf{x}_i,\mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j)\rangle_{\mathcal{F}},$$

where we still have $\mathbf{K}_x = \mathbf{X}\mathbf{X}^T$. Therefore, we can directly work with kernels (e.g., RBF kernel $k_x(\mathbf{x}_i,\mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2)$), without knowing $\phi(\cdot)$ explicitly. Similarly, we can define a nonlinear mapping for $\mathcal{Y}$ and directly work on the corresponding kernel matrix $\mathbf{K}_y$. Although this paper mainly considers the linear kernel to explore the linear correlation of inputs and multivariate labels, the formulism implies that the method can generally handle more complex inputs and outputs (e.g., images) by using some other suitable kernels.

3. CONNECTIONS TO RELATED WORK

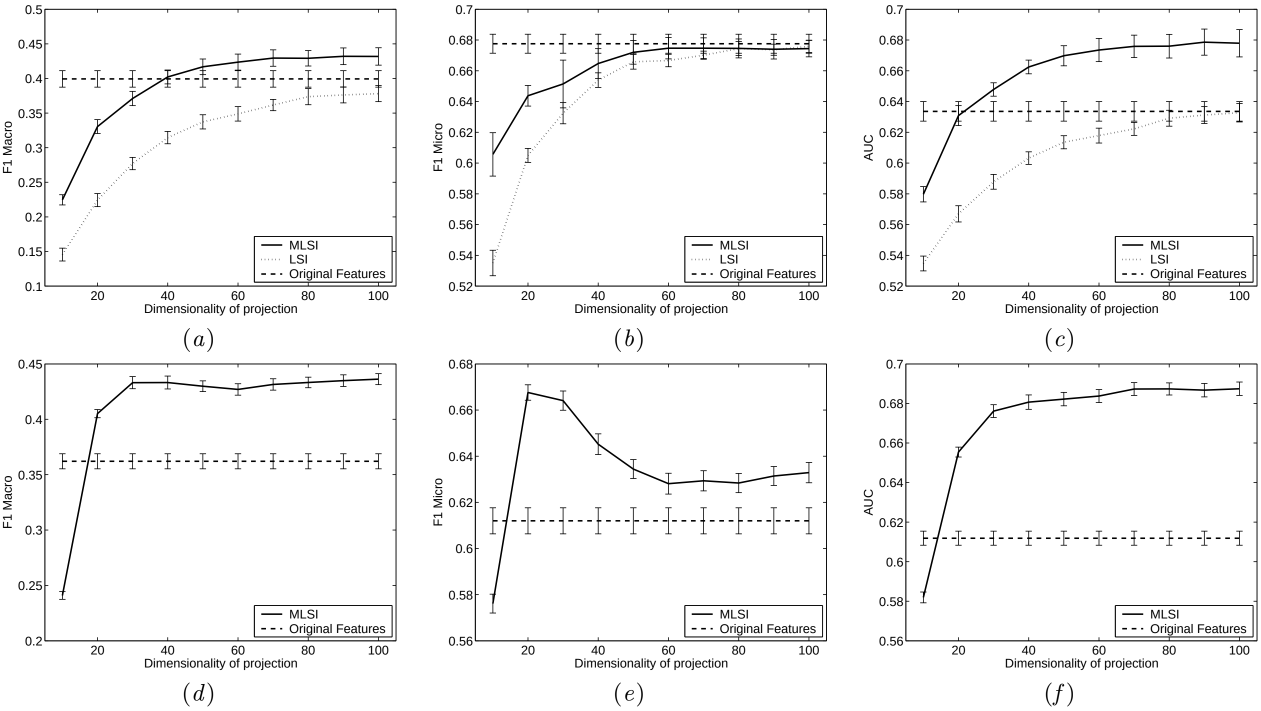
The proposed algorithm MLSI is seen to solve the same optimization problem as LSI when $\beta = 0$, as seen in (1) and (2). Therefore MLSI takes as special case the unsupervised LSI, or more specifically, kernel PCA [10, 11]. Kernel PCA is the dual form of PCA and turns out to solve the eigenvalue problem $\mathbf{K}_x\boldsymbol{\alpha} = \lambda\boldsymbol{\alpha}$ with kernel matrix $(\mathbf{K}_x)_{i,j} = k_x(\mathbf{x}_i,\mathbf{x}_j)$. To build this connection, we see from (9) that $\mathbf{C} = \mathbf{K}_x$ holds when $\beta = 0$ in MLSI. Therefore from Table 2 it is easy to check that MLSI solves the generalized eigenvalue problem

$$\mathbf{K}_x^2\boldsymbol{\alpha} = \lambda(1+\gamma)\mathbf{K}_x\boldsymbol{\alpha},$$

which is identical to kernel PCA since $\mathbf{K}_x$ is invertible. Under this situation, the regularization term controlled by γ is just a rescaling of the cost function, as can be seen in (10). Hence γ is just a nuisance parameter and we obtain rescaled eigenvalues compared to kernel PCA. From this perspective, MLSI in general performs *label informed* kernel PCA or *supervised* kernel PCA, since it can be viewed as directly modifying the kernel matrix $\mathbf{C}$ with label information.

In the literature there are some other well-known supervised projection methods, like linear discriminant analysis (LDA) (e.g., [13]), canonical correlation analysis (CCA) (e.g., [6, 3]) and partial least squares (PLS) [15, 9]. MLSI substantially differs from them. LDA is focusing on single classification problem where the output is one-dimensional, while in contrast MLSI considers predictions with multivariate labels and is thus more general. CCA finds the correlations between two representatives of the same documents (e.g., inputs $\mathbf{X}$ and outputs $\mathbf{Y}$ in our setting) by minimizing $\|\mathbf{v}_x - \mathbf{v}_y\|^2$ subject to both $\mathbf{v}_x$ and $\mathbf{v}_y$ being unitary and linear mappings of $\mathbf{x}_i$ and $\mathbf{y}_i$ (see a recent discussion in [3]). However, it does not require the projections $\mathbf{v}_x$ and $\mathbf{v}_y$ to promise low-reconstruction error of $\mathbf{x}$ and $\mathbf{y}$ and thus ignores the *intra* correlation of either (especially $\mathbf{y}$). Instead, MLSI takes into account all the inter and intra dependencies, since the projections minimize the reconstruction error of inputs and outputs simultaneously. PLS can be seen as a penalized CCA, but it cannot find a space of larger dimensionality than that of $\mathbf{Y}$, thus its generalization performance on new dimensions of outputs is restricted (see discussions in [14]). Instead, MLSI can find in principle N orthogonal dimensions (if $\mathbf{K}_x$ is positive definite).

4. EMPIRICAL STUDY



(a) (b) (c)

(d) (e) (f)

Figure 1: Classification performance on Reuters data set. Upper rows ((a),(b),(c)) show results with setting (I), and lower rows ((d),(e),(f)) show results with setting (II).

In this section we evaluate the proposed MLSI algorithm based on the task of *multi-label text classification*, in which we allow one document to be assigned to multiple labels. One can treat each classification problem separately, but these problems could have correlations between each other and could be solved simultaneously. We solve this problem by applying MLSI and *encoding* the labelling information into the mapping, and then each classification problem is solved independently using the projected features. By incorporating the output information that may be difficult to reveal from inputs, the indexing is *biased* by the specific classification tasks and is thus more suitable for discriminate analysis.

We compare the classification performance using features learned by MLSI and normal LSI, where in the latter case no labelling information is used in indexing. Experiments are performed on two text data sets taken from Reuters-21578 and RCV1, respectively, followed by detailed discussions.

4.1 Data Sets and Preparation

Our first data set is a text corpus which contains all the documents in Reuters-21578 that are associated with multiple categories. Eliminating those minor categories that contain less then 50 documents, we have 47 categories to work with. Picking up all the words that occur at least in 5 documents, we finally obtain 1600 documents with 6076 words that are used in computing TFIDF feature vectors. In average, each document is assigned to 2.48 categories, and each category has 85 positive documents.

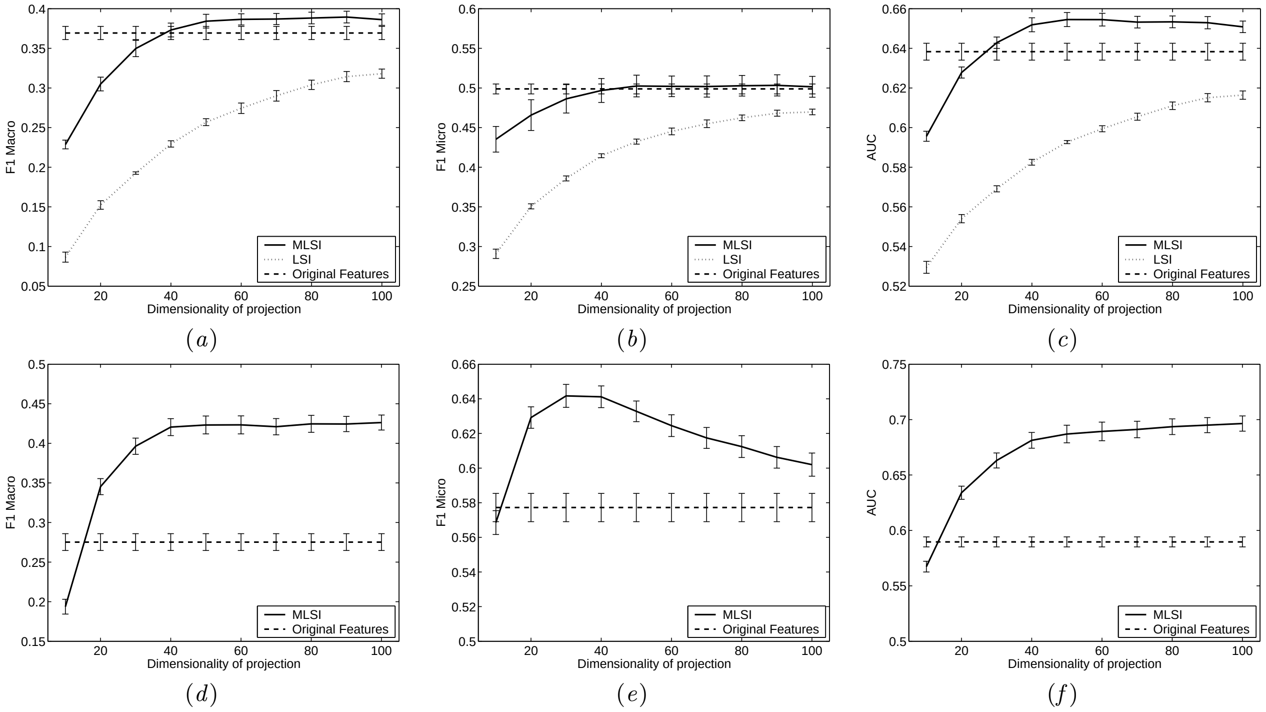
The other data set is a subset of the RCV1-v2 text data set, provided by Reuters and corrected by Lewis et al. [7]. The data set contains topics, regions and industries information to each document and a hierarchical structure for topics and industries. Since it is common that one document is assigned to multiple topics, this is an ideal data set for multi-label text classification. We use topics as the classification tasks and simply ignore the topic hierarchy. A small part of the data set is chosen, and similar preprocessing as for Reuters-21578 is done by picking up words with more than 5 occurrences and topics with more than 50 positive assignments. We end up with 3588 documents with 5496 words, and have 79 topics left. In average, each topic contains 180 positive documents, and each document belongs to 3.96 topics. In the following we denote “Reuters” and “RCV1” for these two data sets respectively.

4.2 Experimental Design

We have two settings in this experiment. In the first setting (I), we randomly pick up 70% categories for classification and employ 5-fold cross-validation with one fold training and 4 folds testing. This is a standard classification setting, and our goal is to evaluate whether the feature mappings are generalizable to new data points. The second setting (II) aims to test the generalization performance of the projection methods on new categorization tasks. For this we consider the classification problems for the rest 30% categories. To make a fair comparison, we perform 5-fold cross-validation on previous unseen data (with the same size as training data), using the feature mappings derived from setting (I).

We will compare the following three methods in our experiment:

1. Original Features: A linear SVM with all the text features is trained for each category, and this serves as the baseline for comparison.



(*a*) (*b*) (*c*)

(*d*) (*e*) (*f*)

Figure 2: Classification performance on RCV1 data set. Upper rows ((a),(b),(c)) show results with setting (I), and lower rows ((d),(e),(f)) show results with setting (II).

2. LSI: Standard unsupervised projection is performed which maps the input data into a low-dimensional space. Then a linear SVM is trained on this projected space.

3. MLSI: Additional label information for training data is used for making a supervised mapping. Then the same SVM is trained on this projected space.

In both of the projection methods LSI and MLSI, we use the dual form in this experiment simply because this gives much improved efficiency. In case of linear kernels, this will give the same results as that in primal form.

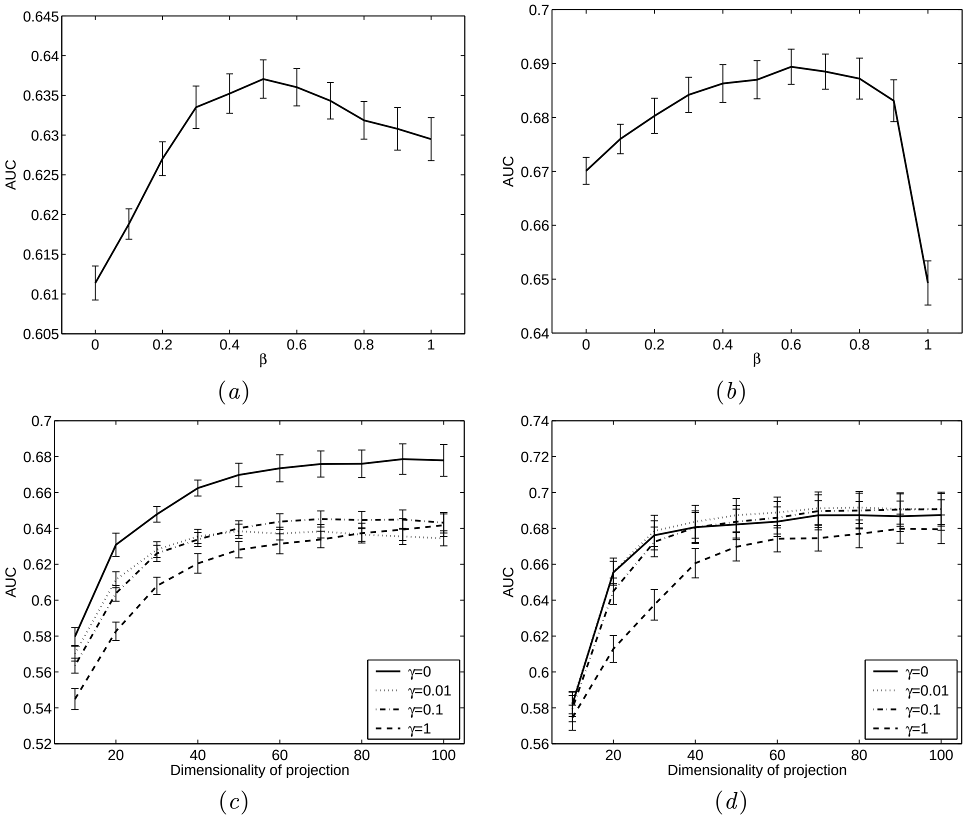
The classification performance is compared using F_1 Macro, Micro and AUC (Area Under Curve) score. F_1-measure defines a trade-off between precision and recall, and is known to be a good metric for classification evaluation. In case of multiple outputs, F_1 Macro is just the arithmetic average of F_1 measures of all output dimensions, and F_1 Micro can be seen as a weighted average. Alternatively, AUC score is the area under the ROC (receiver operating characteristics) curve, which plots *sensitivity* versus 1-*specificity.* It is known to measure the *objective* quality of ranking for specific classification problems. A higher AUC indicates a better ranking. It is also averaged over all the output dimensions. We also tried classification accuracy, but didn't get informative comparison because most of the classification problems are very unbalanced (more than 90% of data are negative examples).

We choose all the parameters for these algorithms as follows. We use LIBSVM with linear kernel and fix $C = 100$, which gives ORIGINAL FEATURES the best performance and is then fixed for the other two methods. For MLSI we set the parameter β to 0.5 after we scale $\mathbf{K}_x$ and $\mathbf{K}_y$ to ensure they have equal traces for balance. γ is simply fixed as 0 to give the best performance. For both settings we repeat the experiments 50 times with randomization, and the performance versus dimensionality of projection is shown with means and standard deviations in Figure 1 and Figure 2 for Reuters and RCV1, respectively.

The first observation from these figures is that MLSI outperforms LSI in all the cases for setting (I). This indicates that the mapping functions in MLSI are generalizable to new test data, by incorporating the output information for the training data.

Another encouraging observation is that MLSI in most cases can even lead to better classification performance than ORIGINAL FEATURES, which uses at least 50 times more features. MLSI in this case can not only greatly accelerate the classification tasks, but also improve the performance. This is especially true for F_1 Macro and AUC score, where a large gap can be observed for all the figures. For F_1 Micro the effect of MLSI is mixed, and an interesting decrease can be observed in Figure 1(e) and Figure 2(e). Consider the difference between F_1 Macro and F_1 Micro measures, these results indicate that MLSI is particularly useful for classification problems with small positive training examples, since by randomly choosing training data we are more likely to choose small positive examples for them. For large classes that have lots of training data, SVMs with full features can already do a very good job.

MLSI has two tunable parameters β and γ that controls the kernel combination weights and the strength of regularization, respectively. For previous figures it is assumed fixed, and in this last experiments we study the classification performance when they varied. Since we can see similar results for both data sets on all the evaluation measures, we



(a) (b) (c) (d)

Figure 3: Performance of MLSI with respect to β ((a),(b)) and γ ((c),(d)) for Reuters data set. (a),(c) show results with setting (I), and (b),(d) show results with setting (II).

only show in Figure 3 the illustrations for Reuters with AUC score. Figures for β are shown with dimensionality K fixed as 50 since it is insensitive to the results.

A first impression from Figure 3 is that the curves are rather smooth (except when β approaching 1 in setting (II)). This indicates that the performance is not very sensitive to small 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 experiences, 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.

5. CONCLUSIONS

In this paper we propose a novel indexing algorithm MLSI for multi-label informed latent semantic indexing. The mappings are supervised and retain the statistical information of not only input features but also the multivariate outputs. We present both the primal and the dual formalisms for the linear mappings, and nonlinear mappings can also 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 multi-label text classification with very encouraging results. Currently we are mainly exploiting linear dependency of inputs as well as outputs. In the near future we plan to apply the algorithm to other types of objects like images with suitable kernels (e.g., RBF kernels), and define kernels to explore richer structured outputs.

6. REFERENCES

[1] R. K. Ando. Latent semantic-space: iterative scaling improves precision of inter-document similarity measurement. In *Proceedings of the 23rd Annual International ACM SIGIR Conference*, pages 216–223, 2000.

[2] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[3] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor.

Canonical correlation analysis; an overview with application to learning methods. Technical Report CSD-TR-03-02, Royal Holloway University of London, 2003.

[4] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Satatistical Learning*. Springer Verlag, 2001.

[5] X. He, D. Cai, H. Liu, and W.-Y. Ma. Locality preserving indexing for document representation. In *Proceedings of the 27th Annual International ACM SIGIR Conference*, pages 96–103, 2004.

[6] H. Hotelling. Relations between two sets of variables. *Biometrika*, 28:321–377, 1936.

[7] D. D. Lewis, Y. Yang, T. Rose, and F. Li. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2005.

[8] A. McCallum. Multi-label text classification with a mixture model trained by EM. In *AAAI'99 Workshop on Text Learning*, 1999.

[9] R. Rosipal and L. J. Trejo. Kernel partial least squares regression in reproducing kernel hilbert space. *Journal of Machine Learning Research*, 2(12):97–123, 2001.

[10] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[11] B. Schölkopf, A. Smola, and K.-R. Müller. Kernel principal component analysis. In *Advances in Kernel Methods - Support Vector Learning*, pages 327–352, 1999.

[12] B. Schölkopf and A. J. Smola. *Learning with Kernels.* MIT Press, 2002.

[13] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge Univeristy Press, 2004.

[14] A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-Posed Problems.* Wiley, New York, 1977.

[15] 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.

APPENDIX

A. PROOFS

A.1 Proof of Theorem 1

Applying the rule $\|\mathbf{C}\|^2 = \mathrm{Tr}\left[\mathbf{C}\mathbf{C}^T\right]$ for an arbitrary matrix $\mathbf{C}$, we obtain

$$\begin{aligned} J(\mathbf{A},\mathbf{B},\mathbf{V}) :&= (1-\beta)\|\mathbf{X}-\mathbf{V}\mathbf{A}\|^2 + \beta\|\mathbf{Y}-\mathbf{V}\mathbf{B}\|^2 \\ &= (1-\beta)\mathrm{Tr}\left[\mathbf{X}\mathbf{X}^T - 2\mathbf{V}\mathbf{A}\mathbf{X}^T + \mathbf{V}\mathbf{A}\mathbf{A}^T\mathbf{V}^T\right] \\ &\quad + \beta\mathrm{Tr}\left[\mathbf{Y}\mathbf{Y}^T - 2\mathbf{V}\mathbf{B}\mathbf{Y}^T + \mathbf{V}\mathbf{B}\mathbf{B}^T\mathbf{V}^T\right]. \end{aligned}$$

Let the derivative of J with respect to $\mathbf{A}$ and $\mathbf{B}$ be zero, we have

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

which proves (a). Then we use the results (a) to replace $\mathbf{A}$ and $\mathbf{B}$ in J and obtain $J_{\mathrm{opt}} = \mathrm{Tr}\left[\mathbf{C}\right] - \mathrm{Tr}\left[\mathbf{V}^T\mathbf{C}\mathbf{V}\right]$, which is first part of (c).

Since $\mathrm{Tr}\left[\mathbf{C}\right]$ is fixed, this suggests that problem (2) can be considered to be an optimization problem only with respect to $\mathbf{V}$:

$$\begin{aligned} \max_{\mathbf{V}\in\mathbb{R}^{N\times K}} \quad & \mathrm{Tr}\left[\mathbf{V}^T\mathbf{C}\mathbf{V}\right] \\ \text{subject to:} \quad & \mathbf{V}^T\mathbf{V} = \mathbf{I}. \end{aligned} \tag{13}$$

For notation simplicity, we denote the optimal solution for $\mathbf{V}$ as $\tilde{\mathbf{V}} = [\tilde{\mathbf{v}}_1, \ldots, \tilde{\mathbf{v}}_K]$ for a moment. The Lagrange formalism of problem (13) is

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

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

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

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

$$\mathbf{C}\tilde{\mathbf{V}} = \tilde{\mathbf{V}}\mathbf{R}^T\mathbf{\Lambda}\mathbf{R} \quad \Rightarrow \quad \mathbf{C}\tilde{\mathbf{V}}\mathbf{R}^T = \tilde{\mathbf{V}}\mathbf{R}^T\mathbf{\Lambda}$$

Since $\mathbf{\Lambda}$ is diagonal, it is easy to see that the columns of $\overline{\mathbf{V}} = \tilde{\mathbf{V}}\mathbf{R}^T$ are the eigenvectors of $\mathbf{C}$. Thus the optimal $\tilde{\mathbf{V}}$ is formed by an arbitrary rotation of $\mathbf{C}$'s eigenvectors, i.e. $\tilde{\mathbf{V}} = \overline{\mathbf{V}}\mathbf{R}$. Inserting $\tilde{\mathbf{V}}$ back into the objective function, we have the value of objective function as $\mathrm{Tr}\left[\mathbf{\Lambda}\right]$, i.e., sum of the K corresponding eigenvalues of $\mathbf{C}$. It is easy to see that the maximal $\mathrm{Tr}\left[\mathbf{\Lambda}\right]$ is the sum of the K largest eigenvalues, which proves second part of (c). In this case, $\tilde{\mathbf{V}}$ is an arbitrary rotation of the K largest eigenvectors, thus conclusion (b) holds. □

A.2 Proof of Theorem 2

Let $J(\mathbf{w})$ denote the cost function in (5), i.e.,

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

Obviously $J(\mathbf{w})$ achieves the minimum at the first eigenvector $\mathbf{w}$ of the generalized eigenvalue problem (6). Denote $\mathbf{w}_{\|}$ as the projection of $\mathbf{w}$ on the subspace

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

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

$$\mathbf{w}^T\mathbf{x}_i = \mathbf{w}_{\|}^T\mathbf{x}_i + \mathbf{w}_{\perp}^T\mathbf{x}_i = \mathbf{w}_{\|}^T\mathbf{x}_i,$$

so $\mathbf{X}\mathbf{w}_{\|} = \mathbf{X}\mathbf{w}$, which means $J(\mathbf{w}_{\|})$ and $J(\mathbf{w})$ agree on the first term. Since $\|\mathbf{w}\|^2 = \|\mathbf{w}_{\|}\|^2 + \|\mathbf{w}_{\perp}\|^2 \geq \|\mathbf{w}_{\|}\|^2$, $J(\mathbf{w}) \geq J(\mathbf{w}_{\|})$ 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 n-th eigenvector is obtained by first deflating the matrix $\mathbf{C}^{-1}$ with $\mathbf{C}^{\dagger} = \mathbf{C}^{-1} - \sum_{j=1}^{n-1} \lambda_j \mathbf{X}\mathbf{w}_j\mathbf{w}_j^T\mathbf{X}^T$, and then solving the following problem

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

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