Local Relevance Weighted Maximum Margin Criterion for Text Classification

Quanquan Gu* Jie Zhou*

Abstract

Text classification is a very important task in information retrieval and data mining. In vector space model (VSM), document is represented as a high dimensional vector, and a feature extraction phase is usually needed to reduce the dimensionality of the document. In this paper, we propose a feature extraction method, named Local Relevance Weighted Maximum Margin Criterion (LRWMMC). It aims to learn a subspace in which the documents in the same class are as near as possible while the documents in the different classes are as far as possible in the local region of each document. Furthermore, the relevance is taken into account as a weight to determine the extent to which the documents will be projected. LRWMMC is able to find the low dimensional manifold embedded in the high dimensional ambient space. In addition, We generalize LRWMMC to Reproducing Kernel Hilbert Space (RKHS), which can resolve the nonlinearity of the input space. We also generalize LRWMMC to tensor space which is suitable for a new document representation, named tensor space model (TSM). On the other hand, in order to utilize the large amount of unlabeled documents, we also present a Semi-Supervised LRWMMC, which aims to find a projection inferred from the labeled samples, as well as the unlabeled samples. Finally, we present a fast algorithm based on QR-decomposition to make the methods proposed in this paper apply for large scale data set. Encouraging experimental results on benchmark text classification data sets indicate that the proposed methods outperform many existing feature extraction methods for text classification.

Keywords

Local Relevance Weighted Maximum Margin Criterion, Feature Extraction, Text Classification, Semi-Supervised Learning, Kernel, Tensor

1 Introduction

Text classification [1] is one of the core issues in information retrieval and data mining. In text classification, an initial data set of pre-classified documents is partitioned into a training set and a testing set that are subsequently used to construct and evaluate classifiers. For high dimensional text classification problems, a dimensionality reduction phase is often applied so as to reduce the size of the document representations. This has both the effect of reducing over fitting, and learning a semantic latent subspace. Dimensionality reduction techniques include two types: (1) feature selection: to select a subset of most representative features from the input feature set [2] [3], and (2) feature extraction: to transform the input space to a smaller feature space. Compared with feature selection, feature extraction can not only reduce the dimensionality of the input space, but also exploit the latent semantic subspace of the input space.

Many feature extraction methods [4] [5] [6] [7] [8] [9] [10] [11] have been proposed in the past decades, however, most of these methods assume that the documents are sampled from a Euclidean space. Recent studies suggest that the documents are actually sampled from a nonlinear low-dimensional manifold which is embedded in the high-dimensional ambient space [12] [13].

In this paper, we proposed a feature extraction method, named *Local Relevance Weighted Maximum Margin Criterion* (LRWMMC) for text classification. It aims to learn a subspace in which the documents in the same class are as near as possible while the documents in the different classes are as far as possible in the local region of each document. Furthermore, the relevance is taken into account as a weight to determine the extent to which the documents will be projected. LRWMMC not only inherits the good property of *Maximum Margin Criterion* [11], but also is able to find the low dimensional manifold embedded in the high dimensional ambient space.

Due to the nonlinearity of the input space, linear method usually cannot map the documents to a subspace, such that the documents in the same class are near enough while the documents in the different classes are far enough. Kernel method [14] alleviates this problem by first mapping the input space to a high dimensional feature space, and then finding a subspace of the feature space. In [15], a kernel LSI was proposed, called *Latent Semantic Kernel* (LSK). In this paper, we generalize LRWMMC to *Reproducing Kernel Hilbert Space* (RKHS) [14], named Kernel LRWMMC, which can resolve the nonlinearity of the input space.

All the methods mentioned above are based on the document representation, named *Vector Space Model* (VSM). Recently, a novel document representation,

*State Key Laboratory on Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology (TNList), Department of Automation, Tsinghua University, Beijing, China 100084, gqq03@mails.tsinghua.edu.cn, jzhou@tsinghua.edu.cn.

named *Tensor Space Model* (TSM) [16], was proposed, which can exploit the high order correlation between words and may be a potential direction of text representation. Hence, we also generalize LRWMMC to tensor space, named Tensor LRWMMC.

On the other hand, while text classification frees organizations from the need of manually organizing document databases, it still needs professionals to label a large enough training data set for learning a classifier, which requires much expensive human labor and much time. Furthermore, compared with the large amount of documents increasing every day, the labeled samples are always insufficient. To address this problem, semi-supervised learning [17], which aims to learn from partially labeled data, provides a solution. In this paper, we present a Semi-Supervised LRWMMC, which aims to learn a subspace, inferred from the labeled samples, as well as the unlabeled samples.

Finally, a fast algorithm based on QR-decomposition is presented to make the methods proposed in this paper apply for large scale data set.

Encouraging experimental results on benchmark text classification data sets indicate that the proposed methods outperform many existing feature extraction methods for text classification.

The remainder of this paper is organized as follows. In Section 2, we will review some methods closely related to our method. In Section 3 we will propose a feature extraction method named *Local Maximum Margin Criterion* for text classification. We generalize LRWMMC to RKHS and tensor space in Section 4 and Section 5 respectively. In Section 6, we present Semi-Supervised LRWMMC. Finally, we present a QR-decomposition based fast algorithm in Section 7. The experiments on standard text classification datasets are demonstrated in Section 8. Finally, we draw a conclusion in Section 9.

2 Related Works

In this section, we will briefly review the methods mostly related with ours.

The *Vector Space Model* (VSM) is widely used for document representation. In VSM, each document is represented as a bag of words. Let $\mathcal{W} = \{w_1, w_2, \ldots, w_d\}$ be the complete vocabulary set of the document corpus after the stop words removal and words stemming operations. The term vector $\mathbf{x}_i$ of document d_i is defined as

$$\begin{aligned} \mathbf{x}_i &= [x_{1i}, x_{2i}, \ldots, x_{di}]^T \\ x_{ji} &= t_{ji} \log(\frac{n}{idf_j}), \end{aligned} \tag{2.1}$$

where t_{ji} denotes the term frequency of word $w_j \in \mathcal{W}$ in document d_i, idf_j denotes the number of documents containing word w_j, and n denotes the total number of documents in the corpus. In addition, $\mathbf{x}_i$ is normalized to unit length. Using $\mathbf{x}_i$ as the ith column, we construct the $d \times n$ term-document matrix $\mathbf{X}$. This matrix will be used to conduct text classification.

In text classification literature, the most popular feature extraction method is *Latent Semantic Indexing* (LSI) [4]. It aims to learn a subspace by minimizing the mean squared error in which sense it is equivalent to *Principal Component Analysis* (PCA). However, LSI does not take into account the class information, it is an unsupervised method and it does not always perform well, sometimes even worse than original term vector [18]. To address this problem, several variants of LSI integrating the class information were proposed. [5] first proposed the concept of local LSI, which performed SVD on a local region of each class so that the most important local structure, which is crucial in separating relevant documents from irrelevant documents, could be captured. A drawback is that the local region is defined by only relevant/positive documents which contain no discriminative information, which makes the improvement of classification performance very limited. [6] extended the local region by introducing some irrelevant/negative documents which are the most difficult to be distinguished from the relevant documents and is found to be more effective than using only relevant documents. [7] proposed a Local Relevance Weighted LSI (LRW-LSI) method which gives different weight to each document in the local region according to its relevance. It should be noted that these local LSI methods mentioned above have to perform a separate SVD in the local region of each class, thus they use different projections for different testing documents. [8] proposed a clustered LSI (CLSI) based on low rank matrix approximation. Based on CLSI, [9] proposed a Centroid Representatives (CM) method for text classification.

Linear Discriminant Analysis (LDA) is a famous feature extraction method. It aims to learn a subspace, in which the between class variance is maximum while the within class variance is minimum, i.e.

$$\max \operatorname{tr}((\mathbf{A}^T\mathbf{S}_w\mathbf{A})^{-1}(\mathbf{A}^T\mathbf{S}_b\mathbf{A})), \tag{2.2}$$

where $\mathbf{S}_b = \sum_{l=1}^{c} n_l(\mathbf{m}_l - \mathbf{m})(\mathbf{m}_l - \mathbf{m})^T$ is called between-class scatter matrix, $\mathbf{m}_l$ and n_l are mean vector and size of class l respectively, $\mathbf{m} = \sum_{l=1}^{c} n_l\mathbf{m}_l$ is the overall mean vector, $\mathbf{S}_w = \sum_{l=1}^{c} \mathbf{S}_l$ is the within-class scatter matrix, $\mathbf{S}_l$ is the covariance matrix of class l. The solution of LDA are composed of the eigenvectors of the matrix $\mathbf{S}_w^{-1}\mathbf{S}_b$ corresponding to the largest eigenvalues. There is little work [18] using LDA for document classification. The reason is that LDA

involves the inverse and eigen-decomposition of large and dense matrices. Most recently, [10] proposed a spectral regression method to train LDA in linear time, which makes applying LDA for document classification practical. However, LDA still suffers several other drawbacks: (1) *Small Sample Size* (SSS) problem: when the size of the data set is smaller than the dimension of the feature space, the within-class scatter matrix $\mathbf{S}_w$ will be singular which makes the generalized eigen-problem unsolvable; (2) it is only optimal in the case that the data distribution of each class satisfies Gaussian assumption with an identical covariance matrix; (3) it can only extract at most $c-1$ features where c is the number of classes.

The most related method to our work is *Maximum Margin Criterion* (MMC) [11]. MMC has been shown to be more effective than LDA. It aims to learn a subspace, in which the sample is close to those in the same class but far from those in the different classes, i.e.

$$\max \mathrm{tr}(\mathbf{A}^T(\mathbf{S}_b - \mathbf{S}_w)\mathbf{A}), \tag{2.3}$$

where $\mathrm{tr}(\cdot)$ denotes the matrix trace and $\mathbf{A}^T\mathbf{A} = \mathbf{I}$. We can see that there is no need for computing any matrix inversion when optimizing the above criterion.

3 Local Relevance Weighted Maximum Margin Criterion

3.1 Quantitative Measure of Relevance Although we have mentioned "relevant" many times, we have not given a quantitative measure of relevance. In this subsection, we give several candidate measures of relevance.

The simplest measure of relevance between documents can be calculated by the cosine distance as

$$r(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i^T \mathbf{x}_j}{||\mathbf{x}_i|| \cdot ||\mathbf{x}_j||}. \tag{3.4}$$

We can see that the range of relevance is $[0, 1]$. $r(\mathbf{x}_i, \mathbf{x}_j) = 1$ if and only if $\mathbf{x}_i = \mathbf{x}_j$, that means $\mathbf{x}_i$ and $\mathbf{x}_j$ refer to the same document. And $r(\mathbf{x}_i, \mathbf{x}_j) = 0$ if and only if $\mathbf{x}_i \perp \mathbf{x}_j$, that means there is no common word shared by these two documents.

Since $\mathbf{x}_i$ is normalized to unit length, the cosine distance degenerates to inner product

$$r(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j. \tag{3.5}$$

The other relevance measures include Information Gain (IG), χ^2 statistic (CHI), Mutual Information (MI) and so on. For details of these relevance measures, please refer to [2][3].

In our study, we use Eq.(3.5) as the relevance measure.

3.2 Local Relevant Region Definition MMC assumes that the data points are sampled from a Euclidean space, and treats the data points as a whole. When the data points are sampled from a nonlinear low-dimensional manifold embedded in the high-dimensional ambient space, which is usually the case in document classification [12] [13], MMC may fail to find the correct latent subspace. A natural treatment for the data sampled from a manifold is to deal with the data region by region, based on the assumption that the manifold is locally homeomorphism to a Euclidean space. Rather than considering the local region of each class [5] [6] [7], we consider the local region of each document. Similar settings can be found in [19] [20]. In detail, for each document, we define two kinds of local relevant region.

DEFINITION 3.1. *Within Class Local Relevant Region: For each document $\mathbf{x}_i$, its within class local relevant region is the set of its k most relevant documents which are in the same class. Denoted by $\mathcal{N}_w(\mathbf{x}_i) = \{\mathbf{x}_j | y_j = y_i, 1 \le j \le k\}$*

DEFINITION 3.2. *Between Class Local Relevant Region: For each document $\mathbf{x}_i$, its between class local relevant region is the set of its k most relevant documents which are in the different classes. Denoted by $\mathcal{N}_b(\mathbf{x}_i) = \{\mathbf{x}_j | y_j \neq y_i, 1 \le j \le k\}$.*

3.3 LRWMMC We aim to learn a subspace in which the documents in the same class are as near as possible while the documents in the different classes are as far as possible in the local region of each document. With the definition of local relevant region, the idea mentioned above can be formulated in a more concise way. That is, we aim to find a subspace in which the documents in the *Within Class Local Relevant Region* are as near as possible, while the documents in the *Between Class Local Relevant Region* are as far as possible. Furthermore, the relevance is taken into account as a weight to determine the extent to which the documents will be projected. More concretely, in the *Within Class Local Relevant Region*, the less relevant two documents are, the more attention we pay for them, pooling them as near as possible in the subspace, while in *Between Class Local Relevant Region*, the more relevant two documents are, the more attention we pay for them, pooling them as far as possible in the subspace. This idea is formulated by *Maximum Margin*

Criterion for each document as follows,

$$\sum_i \sum_{x_j \in \mathcal{N}_b(\mathbf{x}_i)} r(\mathbf{x}_i, \mathbf{x}_j) ||\mathbf{A}^T\mathbf{x}_i - \mathbf{A}^T\mathbf{x}_j||^2 - \sum_i \sum_{x_j \in \mathcal{N}_w(\mathbf{x}_i)} (1 - r(\mathbf{x}_i, \mathbf{x}_j)) ||\mathbf{A}^T\mathbf{x}_i - \mathbf{A}^T\mathbf{x}_j||^2. \tag{3.6}$$

As we have mentioned above, the range of $r(\mathbf{x}_i, \mathbf{x}_j)$ is $[0, 1]$. As a result, the weight $1 - r(\mathbf{x}_i, \mathbf{x}_j)$ in *Within Class Local Relevant Region* and the weight $r(\mathbf{x}_i, \mathbf{x}_j)$ in *Between Class Local Relevant Region* are both non-negative. This makes the *Maximum Margin Criterion* correct and reasonable. Since in our setting, the *Maximum Margin Criterion* weighted by relevance is conducted on each document, we call Eq.(3.6) *Local Relevance Weighted Maximum Margin Criterion* (LRWMMC). The rationale of LRWMMC is shown in Figure 1.

By defining an adjacency matrix $\mathbf{W}$

$$W_{ij} = \begin{cases} r(\mathbf{x}_i, \mathbf{x}_j), & \text{if } \mathbf{x}_j \in N_b(\mathbf{x}_i) \text{ or } \mathbf{x}_i \in N_b(\mathbf{x}_j) \\ r(\mathbf{x}_i, \mathbf{x}_j) - 1, & \text{if } \mathbf{x}_j \in N_w(\mathbf{x}_i) \text{ or } \mathbf{x}_i \in N_w(\mathbf{x}_j) \\ 0, & \text{otherwise.} \end{cases} \tag{3.7}$$

Eq.(3.6) can be formulated as

$$\begin{aligned} & \sum_{i,j} ||\mathbf{A}^T\mathbf{x}_i - \mathbf{A}^T\mathbf{x}_j||^2 W_{ij} \\ = \; & 2\sum_{i,j} (\mathbf{x}_i^T\mathbf{A}\mathbf{A}^T\mathbf{x}_i) W_{ij} - 2\sum_{i,j} (\mathbf{x}_i^T\mathbf{A}\mathbf{A}^T\mathbf{x}_j) W_{ij} \\ = \; & 2\sum_{i,j} \mathrm{tr}(\mathbf{A}^T\mathbf{x}_i W_{ij} \mathbf{x}_i^T\mathbf{A}) - 2\sum_{i,j} \mathrm{tr}(\mathbf{A}^T\mathbf{x}_i W_{ij} \mathbf{x}_j^T\mathbf{A}) \\ = \; & 2\sum_{i} \mathrm{tr}(\mathbf{A}^T\mathbf{x}_i D_{ii} \mathbf{x}_i^T\mathbf{A}) - 2\sum_{i,j} \mathrm{tr}(\mathbf{A}^T\mathbf{x}_i W_{ij} \mathbf{x}_j^T\mathbf{A}) \\ = \; & 2\mathrm{tr}(\mathbf{A}^T\mathbf{X}(\mathbf{D} - \mathbf{W})\mathbf{X}^T\mathbf{A}) \\ = \; & 2\mathrm{tr}(\mathbf{A}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{A}) \end{aligned} \tag{3.8}$$

where $D_{ii} = \sum_j W_{ij}$ is called degree matrix, and $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is called graph Laplacian [19].

Now we rewrite the *Local Relevance Weighted Maximum Margin Criterion* (LRWMMC) as

$$\begin{aligned} \max \quad & \mathrm{tr}(\mathbf{A}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{A}) \\ s.t. \quad & \mathbf{A}^T\mathbf{A} = \mathbf{I} \end{aligned} \tag{3.9}$$

If we expand $\mathbf{A}$ as $\mathbf{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_m)$, then Eq.(3.9) is equivalent to

$$\begin{aligned} \max \quad & \sum_{i=1}^{m} \mathbf{a}_i^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{a}_i \\ s.t. \quad & \mathbf{a}_i^T\mathbf{a}_i = 1, \mathbf{a}_i^T\mathbf{a}_j = 0 (i \neq j). \end{aligned} \tag{3.10}$$

Using the Lagrangian method, we can easily find that the optimal $\mathbf{A}$ in Eq.(3.10) is composed of the m eigenvectors corresponding to the largest m eigenvalues of $\mathbf{X}\mathbf{L}\mathbf{X}^T$.

We summarize the LRWMMC method in Algorithm 1.

Algorithm 1 Local Relevance Weighted Maximum Margin Criterion

Input: Training set $\{\mathbf{x}_i, y_i\}_{i=1}^n$, Local Relevant Region size k, desired dimensionality m;
Output: $\mathbf{A} \in \mathbb{R}^{d \times m}$;
1. Construct the with-class local relevant region and between class local relevant region for each $\mathbf{x}_i$;
2. Calculate the adjacent matrix $\mathbf{W}$ by Eq.(3.7), and $\mathbf{L} = \mathbf{D} - \mathbf{W}$;
3. Calculate the projection $\mathbf{A}$ as the m eigenvectors corresponding to the largest m eigenvalues of $\mathbf{X}\mathbf{L}\mathbf{X}^T$.

3.4 Discussion The advantages of LRWMMC include four-fold:

1. It inherits the properties of MMC, hence it does not suffer from *SSS* problem, and it can extract more than $c - 1$ features, etc.
2. It exploits the local class information, which is more discriminative than global class information;
3. Since LRWMMC considers the local region of each document and deals with the data region by region, it is able to find nonlinear low-dimensional manifold which is embedded in the high-dimensional ambient space;
4. It takes into account the relevance weight in the local relevant region, hence it is more reasonable and robust, which will be illustrated in detail in the following;

It is worthwhile noting that, [21] proposed a Discriminant Neighborhood Embedding (DNE), which is in fact also a variant of MMC. Although in their setting, the authors constructed two graphs, the essence of DNE is also a local MMC with the adjacency matrix being defined as

$$W_{ij} = \begin{cases} 1, & \text{if } \mathbf{x}_j \in N_b(\mathbf{x}_i) \text{ or } \mathbf{x}_i \in N_b(\mathbf{x}_j) \\ -1, & \text{if } \mathbf{x}_j \in N_w(\mathbf{x}_i) \text{ or } \mathbf{x}_i \in N_w(\mathbf{x}_j) \\ 0, & \text{otherwise.} \end{cases} \tag{3.11}$$

Our method is different from theirs. We can see that our method pays different attentions to the data points in the local relevant region by relevance weight, while

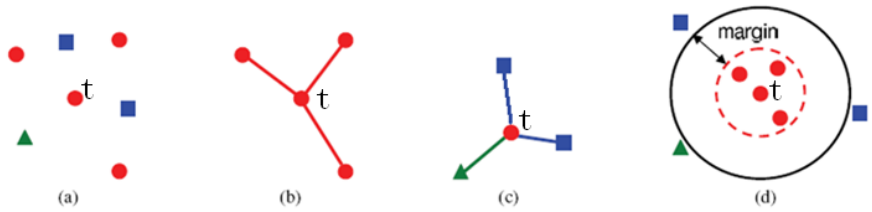

Figure 1: The rationale of LRWMMC: (a) The original local relevant region of document t (the red circle in the center) (b) Within class relevant region of document t with $k = 3$; (c) Between class relevant region of document t with $k = 3$; (d) The resulted subspace after LRWMMC projection.

DNE generally pays equal attention to them in both *Within Class Local Relevant Region* and *Between Class Local Relevant Region*. DNE fails to exploit the concise relevance information in the local region, which may benefit the classification performance a lot.

Furthermore, when the data distribution is sparse in the input space, which is usually the case in text classification, paying equal attention to the data points in the local region may even ruin the feature extraction algorithm. For example, in the *Between Class Local Relevant Region*, the least relevant data point may be much more irrelevant than the most relevant data point. If we pay as much attention to the least relevant data point as to the most relevant data point, it may prevent us from pooling the data points in the *Within Class Local Relevant Region* near enough.

In our experiment, we will see that the performance of DNE is encouraging only when the size of local relevant region is very small, e.g. $k = 1, 3$. In contrast, our method performs very well when k varied in a very wide range, e.g. $k \in [1, 30]$.

4 Kernel LRWMMC

Kernel methods have been widely used in non-linear dimensionality reduction, and also been successfully used for text classification [15]. To this end, we will address kernelization of LRWMMC.

Due to the nonlinearity of the input space, linear method usually cannot map the documents to a subspace, such that the documents in the same class are near enough while the documents in the different classes are far enough. Kernel method [14] alleviates this problem by first mapping the input space to a high dimensional feature space, and then finding a subspace of the feature space. In the following, we utilize the kernel trick [14] to generalize LRWMMC to *Reproducing Kernel Hilbert Space* (RKHS), namely Kernel LRWMMC.

We consider the problem in a feature space $\mathcal{F}$ induced by some nonlinear mapping $\phi : \mathbb{R}^d \to \mathcal{F}$. For a proper chosen ϕ, the inner product $\langle , \rangle$ in $\mathcal{F}$ is defined as

$$\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle = K(\mathbf{x}, \mathbf{y}), \tag{4.12}$$

where $K(,)$ is a positive semi-definite kernel function. The mostly used kernel functions include:

1. **Polynomial Kernel**:$K(\mathbf{x}, \mathbf{y}) = (1 + \langle \mathbf{x}, \mathbf{y} \rangle)^d$;
2. **Gaussian Kernel**:$K(\mathbf{x}, \mathbf{y}) = \exp(-\frac{||\mathbf{x}-\mathbf{y}||^2}{\sigma^2})$.

Let $\Phi = [\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \ldots, \phi(\mathbf{x}_n)]$ denote the data matrix in RKHS, then Eq.(3.10) can be written as follows:

$$\max \sum_{i=1}^{m} \mathbf{a}_i^T \Phi \mathbf{L} \Phi^T \mathbf{a}_i. \tag{4.13}$$

According to *Representor Theorem* [14], $\mathbf{a}_i$ are linear combinations of $\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_n)$. There exist coefficients $\alpha_i^j, j = 1, 2, \ldots, n$, such that

$$\mathbf{a}_i = \sum_{j=1}^{n} \alpha_i^j \phi(\mathbf{x}_j) = \Phi \boldsymbol{\alpha}_i, \tag{4.14}$$

where $\boldsymbol{\alpha}_i = (\alpha_i^1, \alpha_i^2, \ldots, \alpha_i^n)^T$.

Submit Eq.(4.14) into Eq.(4.13), we obtain

$$\begin{aligned} & \max \sum_{i=1}^{m} \mathbf{a}_i^T \Phi \mathbf{L} \Phi^T \mathbf{a}_i \\ = \; & \max \sum_{i=1}^{m} \boldsymbol{\alpha}_i^T \Phi^T \Phi \mathbf{L} \Phi^T \Phi \boldsymbol{\alpha}_i \\ = \; & \max \sum_{i=1}^{m} \boldsymbol{\alpha}_i^T \mathbf{KLK} \boldsymbol{\alpha}_i, \end{aligned} \tag{4.15}$$

where $\mathbf{K}$ is the kernel matrix with element $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$. The optimal $\boldsymbol{\alpha}_i, 1 \le i \le m$ in Eq.(4.15) is the m eigenvectors corresponding to the m largest eigenvalues of $\mathbf{KLK}$.

We summarize the Kernel LRWMMC method in Algorithm 2.

Algorithm 2 Kernel Local Relevance Weighted Maximum Margin Criterion

Input: Training set $\{\mathbf{x}_i, y_i\}_{i=1}^n$, Local Relevant Region size k, desired dimensionality m, Kernel type, Kernel Parameters;
Output: $\mathbf{A} \in \mathbb{R}^{d\times m}$;
1. Construct the kernel matrix $\mathbf{K}$ on the training set;
2. Construct the with-class local relevant region and between class local relevant region for each $\phi(\mathbf{x}_i)$;
3. Calculate the adjacent matrix $\mathbf{W}$ by Eq.(3.7), and $\mathbf{L} = \mathbf{D} - \mathbf{W}$;
4. Calculate the projection $\boldsymbol{\alpha}_i, 1 \leq i \leq m$ as the m eigenvectors corresponding to the largest m eigenvalues of $\mathbf{KLK}$.

5 Tensor LRWMMC

All the methods mentioned above are based on *Vector Space Model* (VSM). Recently, a novel document representation, named *Tensor Space Model* (TSM) [16], was proposed, which can exploit the high order correlation between words and may be a potential direction of text representation. Hence, we will discuss tensorization of LRWMMC.

Several tensor-based methods [22][23][24] have been proposed in machine learning and data mining literature. Here, we generalize LRWMMC to tensor space. Since the proposed approach is mostly based on tensor algebra (or multi-linear algebra), we first introduce the notation and basic definition of tensor algebra. For more details about tensor algebra, please refer to [25].

5.1 Tensor Algebra Scalars are denoted by lower case letters $(a, b, \ldots)$, vectors by bold lower case letters $(\mathbf{a}, \mathbf{b}, \ldots)$, matrices by bold upper case letters $(\mathbf{A}, \mathbf{B}, \ldots)$, and high-order tensors by calligraphic upper case letters $(\mathcal{A}, \mathcal{B}, \ldots)$.

5.1.1 Notation and Terminology A tensor is a higher order generalization of a vector (first order tensor) and a matrix (second order tensor). From a multi-linear algebra view, tensor is a multi-linear mapping over a set of vector spaces. The order of tensor $\mathcal{A} \in \mathbb{R}^{I_1\times\ldots I_n\ldots I_N}$ is N, where I_n is the dimensionality of the nth order. Elements of $\mathcal{A}$ are denoted as $\mathcal{A}_{i_1\ldots i_n\ldots i_N}, 1 \leq i_n \leq I_n$.

5.1.2 Mode-n Flattening The mode-n vectors of a Nth order tensor $\mathcal{A}$ are the I_n dimensional vectors obtained from $\mathcal{A}$ by varying index i_n while keeping the other indices fixed. The mode-n vectors are the column vectors of mode-n flattening matrix $A_{(n)} \in \mathbb{R}^{I_n\times(I_1\ldots I_{n-1}I_{n+1}\ldots I_N)}$ that results by mode-n flattening the tensor $\mathcal{A}$. For example, matrix column vectors are referred to as mode-1 vectors and matrix row vectors are referred to as mode-2 vectors.

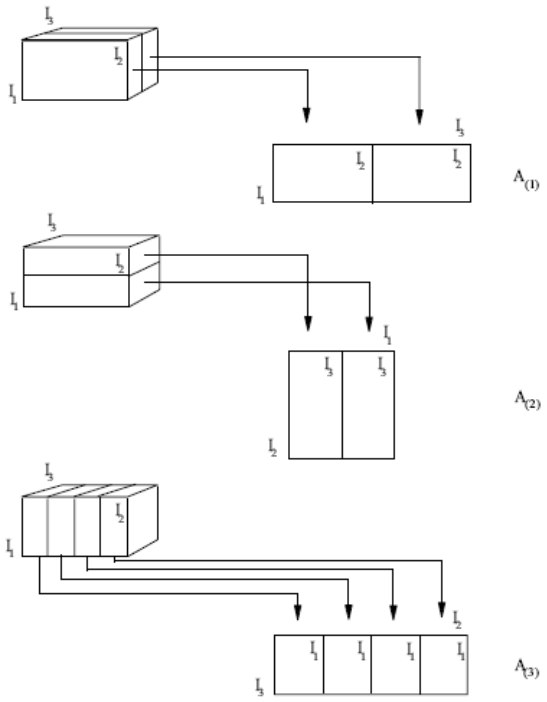

Figure 2: Flattening a 3th order tensor, which is flattened in 3 ways to obtain matrices comprising its mode-1, mode-2 and mode-3 vectors.

5.1.3 Mode-n Product A generalization of the product of two matrices is the product of a tensor and a matrix. The mode-n product of a tensor $\mathcal{A} \in \mathbb{R}^{I_1\times\ldots I_n\ldots I_N}$ by a matrix $\mathbf{U} \in \mathbb{R}^{I_n\times J_n}$, denoted by $\mathcal{A} \times_n \mathbf{U}$, is a tensor $\mathcal{B} \in \mathbb{R}^{I_1\times\ldots\times I_{n-1}\times J_n\times I_{n+1}\times\ldots\times I_N}$ whose entries are

(5.16)
$$(\mathcal{A}\times_n \mathbf{U})_{i_1\ldots i_{n-1}j_n i_{n+1}\ldots i_N} = \sum_{i_n} \mathcal{A}_{i_1\ldots i_{n-1}i_{n+1}\ldots i_N}\mathbf{U}_{i_n j_n}.$$

In general, a tensor $\mathcal{X} \in \mathbb{R}^{I_1\times I_2\ldots I_N}$ can multiply a sequence of matrices $\{\mathbf{U}_i\}_{i=1}^N \in \mathbb{R}^{I_i\times J_i}$ as $\mathcal{X} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \ldots \times_N \mathbf{U}_N$, which can be written as $\mathcal{X}\prod_{i=1}^N \times_i \mathbf{U}_i$ for clarity. From the definition above, we can easily find that the mode-n product $\mathcal{B} = \mathcal{A}\times_n \mathbf{U}$ can be computed via the matrix multiplication $\mathbf{B}_{(n)} = \mathbf{U}^T\mathbf{A}_{(n)}$, followed by a re-tensorization to undo the mode-n flattening.

5.1.4 Scalar Product The scalar product of two tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1\ldots I_n\ldots I_N}$, is defined as $\langle\mathcal{A}, \mathcal{B}\rangle = \sum_{i_1}\ldots\sum_{i_N}\mathcal{A}_{i_1\ldots i_N}\mathcal{B}_{i_1\ldots i_N}$. The Frobenius norm of a

tensor $\mathcal{A}$ is $||\mathcal{A}|| = \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}$.

5.2 Tensor LRWMMC Given n tensors $\{\mathcal{X}_1, \ldots, \mathcal{X}_n\} \in \mathbb{R}^{I_1 \times \ldots I_n \ldots I_N}$, Tensor LRWMMC aims to find a sequence of projections $\mathbf{U}_k \in \mathbb{R}^{I_k \times J_k}$, which maximizes the LRWMMC in the tensor metric induced by Frobenius norm of a tensor:

$$\sum_{i,j} ||\mathcal{X}_i \prod_{k=1}^{N} \times_k \mathbf{U}_k - \mathcal{X}_j \prod_{k=1}^{N} \times_k \mathbf{U}_k||^2 W_{ij}. \tag{5.17}$$

Such type of optimization can be solved approximately by employing an iterative scheme [22]. In the following, we will adopt such an iterative scheme to solve the optimization problem.

Given $\mathbf{U}_1, \ldots, \mathbf{U}_{k-1}, \mathbf{U}_{k+1}, \ldots, \mathbf{U}_N$, denote $\mathcal{Y}_i^{\setminus k}$ as

$$\mathcal{Y}_i^{\setminus k} = \mathcal{X}_i \times_1 \mathbf{U}_1 \ldots \times_{k-1} \mathbf{U}_{k-1} \times_{k+1} \mathbf{U}_{k+1} \ldots \mathbf{U}_N \tag{5.18}$$

Then Eq.(5.17) can be written as

$$\begin{aligned}
& \sum_{i,j} ||\mathcal{X}_i \prod_{k=1}^{N} \times_k \mathbf{U}_k - \mathcal{X}_j \prod_{k=1}^{N} \times_k \mathbf{U}_k||^2 W_{ij} \\
= & \sum_{i,j} ||\mathcal{Y}_i^{\setminus k} \times_k \mathbf{U}_k - \mathcal{Y}_j^{\setminus k} \times_k \mathbf{U}_k||^2 W_{ij} \\
= & \sum_{i,j} ||\mathbf{U}_k^T \mathbf{Y}_{i(k)}^{\setminus k} - \mathbf{U}_k^T \mathbf{Y}_{j(k)}^{\setminus k}||^2 W_{ij} \\
= & 2\mathrm{tr}(\mathbf{U}_k^T \sum_{i,j} W_{ij} (\mathbf{Y}_{i(k)}^{\setminus k} - \mathbf{Y}_{j(k)}^{\setminus k})(\mathbf{Y}_{i(k)}^{\setminus k} - \mathbf{Y}_{j(k)}^{\setminus k})^T \mathbf{U}_k) \\
= & 2\mathrm{tr}(\mathbf{U}_k^T \mathbf{T}_k \mathbf{U}_k)
\end{aligned} \tag{5.19}$$

where $\mathbf{T}_k = \sum_{i,j} W_{ij} (\mathbf{Y}_{i(k)}^{\setminus k} - \mathbf{Y}_{j(k)}^{\setminus k})(\mathbf{Y}_{i(k)}^{\setminus k} - \mathbf{Y}_{j(k)}^{\setminus k})^T$. The optimal $\mathbf{U}_k$ in Eq.(5.19) is composed of the J_k eigenvectors corresponding to the largest J_k eigenvalues of $\mathbf{T}_k$.

We summarize the Tensor LRWMMC method in Algorithm 3.

Algorithm 3 Tensor Local Relevance Weighted Maximum Margin Criterion

Input: Training set $\{\mathcal{X}_i, y_i\}_{i=1}^n$, Local Relevant Region size k, desired dimensionality $J_1, J_2 \ldots, J_N$;
Output: A sequence of projections $\mathbf{U}_k \in \mathbb{R}^{I_k \times J_k}$;
1. Construct the within class local relevant region and between class local relevant region for each $\mathcal{X}_i$;
2. Calculate the adjacent matrix $\mathbf{W}$ by Eq.(3.7);
3. Initialize $\mathbf{U}_k = \mathbf{I}_k$ where $\mathbf{I}_k$ is any $I_k \times J_k$ orthogonal matrix;
for $t = 1$ to T_{max} **do**
  for $k = 1$ to N **do**
    (a)Calculate $\mathcal{Y}_i^{\setminus k}$ by Eq.(5.18);
    (b)Calculate $\mathbf{Y}_{i(k)}^{\setminus k}$ by mode-k flattening of $\mathcal{Y}_i^{\setminus k}$;
    (c)Calculate $\mathbf{T}_k = \sum_{i,j} W_{ij} (\mathbf{Y}_{i(k)}^{\setminus k} - \mathbf{Y}_{j(k)}^{\setminus k})(\mathbf{Y}_{i(k)}^{\setminus k} - \mathbf{Y}_{j(k)}^{\setminus k})^T$;
    (d)Calculate the projection $\mathbf{U}_k$ as the J_k eigenvectors corresponding to the largest J_k eigenvalues of $\mathbf{T}_k$;
  end for
end for

6 Semi-Supervised LRWMMC

Semi-supervised learning, which can exploit the large amount of unlabeled samples to improve classification, is successfully used in text classification [26]. In this section, we will present a semi-supervised version of LRWMMC.

LRWMMC considers finding the optimal projection purely on the labeled training set. While text classification frees organizations from the need of manually organizing document bases, it still needs professionals to label a large enough training data set for learning a classifier, which requires much expensive human labor and much time. Furthermore, compared with the large amount of documents increasing every day, the labeled samples are always insufficient. That is, in reality, it is possible to acquire a large set of unlabeled data rather than labeled data. To address this problem, semi-supervised learning [17], which aims to learn from partially labeled data, provides a solution. Thus, we extend LRWMMC to incorporate the unlabeled data. In general, semi-supervised learning can be categorized into two classes: (1) transductive learning: to estimate the labels of the given unlabeled data; and (2) inductive learning: to induce a decision function which has a low error rate on the whole sample space. Our method belongs to inductive learning. Other semi-supervised inductive methods in text classification include Co-EM [26] and Semi-Supervised Discriminant Analysis (SDA) [27]. Although there are many other semi-supervised methods such as [28] [29], they are transdutive learning methods, which are not very practical in large scale text classification.

Semi-supervised learning is formulated as follows. Given a point set $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_l, \mathbf{x}_{l+1}, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^d$, and a label set $\mathcal{L} = \{1, \ldots, c\}$, the first l points $\mathbf{x}_i, 1 \le i \le l$ are labeled as $y_i \in \mathcal{L}$ and the remaining points $\mathbf{x}_u, l+1 \le u \le n$ are unlabeled.

We define another adjacent matrix $\mathbf{W}^{(2)}$ for both

labeled and unlabeled samples as

$$W_{ij}^{(2)} = \begin{cases} r(\mathbf{x}_i, \mathbf{x}_j), & \text{if } \mathbf{x}_j \in N(\mathbf{x}_i) \text{ or } \mathbf{x}_i \in N(\mathbf{x}_j) \\ 0, & \text{otherwise.} \end{cases} \tag{6.20}$$

where $N(\mathbf{x}_i)$ denotes the k nearest neighbor of $\mathbf{x}_i$. $\mathbf{W}^{(2)}$ reflects the pairwise relevance of all the documents.

In dimensionality reduction, there is an assumption that nearby points are likely to have the same embedding. Thus, a natural regularizer can be defined as

$$\sum_{i,j} || \frac{1}{\sqrt{D_{ii}^{(2)}}} \mathbf{A}^T \mathbf{x}_i - \frac{1}{\sqrt{D_{jj}^{(2)}}} \mathbf{A}^T \mathbf{x}_j ||^2 W_{ij}^{(2)}. \tag{6.21}$$

where $\mathbf{D}^{(2)}$ is the degree matrix corresponding to the adjacency matrix $\mathbf{W}^{(2)}$. This regularization incurs a heavy penalty if relevant documents $\mathbf{x}_i$ and $\mathbf{x}_j$ are mapped far apart. Therefore, minimizing it is an attempt to ensure that if $\mathbf{x}_i$ and $\mathbf{x}_j$ are relevant, then $\mathbf{A}^T\mathbf{x}_i$ and $\mathbf{A}^T\mathbf{x}_j$ are near as well. By similar derivation of Eq.(3.8), Eq.(6.21) is equivalent to

$$2\text{tr}(\mathbf{A}^T \mathbf{X} \mathbf{L}^{(2)} \mathbf{X}^T \mathbf{A}), \tag{6.22}$$

where $\mathbf{L}^{(2)} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}^{(2)}\mathbf{D}^{-1/2}$ is the normalized graph Laplacian corresponding to $\mathbf{W}^{(2)}$.

Define an adjacency matrix $\mathbf{W}^{(1)}$ for labeled samples

$$\mathbf{W}^{(1)} = \begin{bmatrix} \mathbf{W}_{l\times l} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \tag{6.23}$$

where $\mathbf{W}_{l\times l}$ is defined in Eq.(3.7) for l labeled samples.

Then the objective function of Semi-Supervised Local Maximum Margin Criterion is

$$\begin{aligned} & \max(\text{tr}(\mathbf{A}^T \mathbf{X} \mathbf{L}^{(1)} \mathbf{X}^T \mathbf{A}) - \lambda \text{tr}(\mathbf{A}^T \mathbf{X} \mathbf{L}^{(2)} \mathbf{X}^T \mathbf{A})) \\ & \max \text{tr}(\mathbf{A}^T \mathbf{X} (\mathbf{L}^{(1)} - \lambda \mathbf{L}^{(2)}) \mathbf{X}^T \mathbf{A}), \end{aligned} \tag{6.24}$$

where λ is a positive regularization parameter. The optimal $\mathbf{A}$ in Eq.(6.24) is composed of the m eigenvectors corresponding to the largest m eigenvalues of $\mathbf{X}(\mathbf{L}^{(1)} - \lambda\mathbf{L}^{(2)})\mathbf{X}^T$.

We summarize the Semi-Supervised LRWMMC method in Algorithm 4.

Algorithm 4 Semi-Supervised Local Relevance Weighted Maximum Margin Criterion

Input: Training set $\{\mathbf{x}_i, y_i\}_{i=1}^n$, Local Relevant Region size k, desired dimensionality m;
Output: $\mathbf{A} \in \mathbb{R}^{d\times m}$;
1. Construct the with-class local relevant region and between class local relevant region for each labeled samples $\mathbf{x}_i$;
2. Calculate the adjacent matrix $\mathbf{W}^{(1)}$ by Eq.(6.23);
3. Calculate the adjacent matrix $\mathbf{W}^{(2)}$ for both labeled and unlabeled samples by Eq.(6.20);
4. Calculate the projection $\mathbf{A}$ as the m eigenvectors corresponding to the largest m eigenvalues of $\mathbf{X}(\mathbf{L}^{(1)} - \lambda\mathbf{L}^{(2)})\mathbf{X}^T$.

7 Fast Algorithm

The computational complexity of the proposed methods in this paper (except Tensor LWRMMC, since the dimensionality of TSM is usually not very high, e.g. 27 in our experiment, the original algorithm in Algorithm 3 is sufficiently efficient) is high, due to that it involves in the eigen-decomposition of a large matrix, e.g. $\mathbf{XLX}^T$ in Algorithm 1. More concretely, the size of $\mathbf{XLX}^T$ is $d \times d$, where d is the dimensionality of the data. And the time complexity of the eigen-decomposition is $O(d^3)$. As a result, when d is very high, e.g. document, the eigen-decomposition is very time consuming. To this end, a fast algorithm is urgently required. To alleviate computational complexity, we adopt QR [30] decomposition, which is also adopted in [31] [32]. In the following, we will present a QR-based fast algorithm.

It is worthwhile noting that although the following derivation is based on Algorithm 1, the QR-based fast algorithm also works for the other methods presented in this paper, e.g. Kernel LRWMMC in Section 4 and Semi-Supervised LRWMMC in Section 6.

Let $\mathbf{X} = \mathbf{QR}$ be the QR-decomposition of $\mathbf{X}$, where $\mathbf{Q} \in \mathbb{R}^{d\times t}$ has orthonormal columns, i.e. $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$, $\mathbf{R} \in \mathbb{R}^{t\times n}$ is an upper triangular matrix, and $t = \text{rank}(\mathbf{X})$ is the rank of $\mathbf{X}$. Then the projection $\mathbf{A} \in \mathbb{R}^{d\times m}$ can be expressed as $\mathbf{A} = \mathbf{QV}$, where $\mathbf{V} \in \mathbb{R}^{t\times m}$ is arbitrary orthogonal matrix. Substitute $\mathbf{A} = \mathbf{QV}$ and $\mathbf{X} = \mathbf{QR}$ into Eq.(3.9), then Eq.(3.9) can be rewritten as

$$\begin{aligned} & \max \text{tr}(\mathbf{V}^T\mathbf{Q}^T\mathbf{QRLR}^T\mathbf{Q}^T\mathbf{QV}) \\ = \; & \max \text{tr}(\mathbf{V}^T\mathbf{RLR}^T\mathbf{V}) \end{aligned} \tag{7.25}$$

where $\mathbf{V}^T\mathbf{V} = \mathbf{I}$. Hence, the optimal $\mathbf{V}$ is composed of the the m eigenvectors corresponding to the largest m eigenvalues of $\mathbf{RLR}^T$. After we obtain the optimal $\mathbf{V}$, the optimal $\mathbf{A}$ can be computed as $\mathbf{A} = \mathbf{QV}$.

It should be noted that $\mathbf{RLR}^T$ is of size $t \times t$, which is much smaller than $\mathbf{XLX}^T$, since $t \ll d$. The time complexity of eigen-decomposition for $\mathbf{RLR}^T$ is $O(t^3)$. And the time complexity of OR-decomposition for $\mathbf{X}$ is $O(t^2n)$. As a result, the total time complexity of the fast algorithm is $O(t^3+t^2n)$, while the total time complexity of Algorithm 1 is $O(d^3)$. So the fast algorithm is much more efficient than the original algorithm.

8 Experiments

In this section, we compare our method with the state of the art methods on three benchmark text classification data sets. We choose Nearest Neighbor as the classification algorithm. All of our experiments have been performed on an AMD Athlon 64 X2 dual 4400+ 2.4GHz Windows XP machine with 2GB memory.

8.1 Data Sets We use three publicly available datasets:

1. **Reuters21578**[1] We use the ModLewis split of the database into training and testing parts. Since the aim is classification, in which each document belongs to an exclusive category, we discard documents with no label or with multiple labels. Furthermore, those rare classes that do not occur at least once in both training and testing set are discarded. The resulting training set contains 6535 documents, and the test set 2570 documents with 52 document classes.

2. **TREC-9**[2] This is the dataset used for the filtering track in TREC-9, the 2000 Text Retrieval Conference. We use a subset of the dataset, namely Ohsumed set. It was pointed out that this set is much more challenging than the Reuters set. By similar preprocessing of Retuters, we get 634 training and 3038 test documents from 63 categories.

3. **20Newsgroup**[3] We use the 20 Newsgroups sorted by date version. There are 18846 documents, from 20 different newsgroups, with 11314 (60%) training and 7532 (40%) testing.

8.2 Evaluation Metrics To evaluate the effectiveness of classification, we firstly calculate the average precision and recall by micro-averaging and macro-averaging [1] in Table 1.

Table 1: Average precision and recall by micro-averaging and macro-averaging

	Microaveraging	Macroaveraging
Precesion(p)	$p = \frac{\sum_{i=1}^{c} TP_i}{\sum_{i=1}^{c} TP_i + FP_i}$	$p = \frac{\sum_{i=1}^{c} \frac{TP_i}{TP_i + FP_i}}{c}$
Recall(r)	$r = \frac{\sum_{i=1}^{c} TP_i}{\sum_{i=1}^{c} TP_i + FN_i}$	$r = \frac{\sum_{i=1}^{c} \frac{TP_i}{TP_i + FN_i}}{c}$

And we use the single number metric F1 score [1] which is defined as the harmonic mean of the precision and recall to measure the classification performance

$$F1 = \frac{2rp}{r+p}. \tag{8.26}$$

8.3 Experiment 1 In this experiment, we compare LRWMMC with some state of the art methods, e.g. LSI [4], CLSI [8], CM [9] and DNE [21]. The implementation of LSI, CLSI and CM are based on the toolbox [33]. The implementation of LDA is based on [10]. The implementation of LRWMMC is based on QR-decomposition based fast algorithm. For LRWMMC and DNE, the size of local relevant region k is set by grid search $\{1, 3, 5, 10, 20, 30, 40, 50\}$, and the best classification result is selected.

The text classification results are shown in Table 2.

We can find that LRWMMC outperforms other methods on all the data sets. The outstanding performance of LRWMMC mainly owes to the utilization of discriminate information in the local relevant region of each documents, rather than the discriminate information in the local region of each topic (class). And LRWMMC outperforms DNE due to it takes into account the relevance weight in the local relevant region.

Furthermore, we investigate the performance of LRWMMC and DNE with respect to the size of the local relevant region, i.e. k. The results are shown in Figure 3. We can see that the performance of DNE is encouraging only when the size of local relevant region is very small, e.g. $k = 1, 3$. As k increases, the performance of DNE degenerates very much. In contrast, our method performs very good when k varied in a very wide range, e.g. $k \in [1, 30]$. This advantage mainly owes to the local relevance weight, which makes our method more robust for text classification,

8.4 Experiment 2 In this experiment, we compare Kernel LRWMMC with LSK [15] and Kernel Discriminant Analysis (KDA) [34], which are state of the art kernel methods. The implementation of KDA is based on [34] which is very efficient. The implementation of Kernel LRWMMC is based on QR-decomposition based fast algorithm. We use Gaussian kernel for all the kernel based methods. The hyper-parameter σ in Gaussian kernel is tuned by 5-fold cross validation on the training set, and the best σ for each method is chosen in the testing.

The text classification results are shown in Table 3.

We can find that Kernel LRWMMC outperforms LSK and KDA on all the data sets. It should be noted that Kernel LRWMMC also outperforms LRWMMC, the reason is that by nonlinear mapping, Kernel LRWMMC can find even better projection which projects the documents in the same class as near as possible,

[1] http://www.daviddlewis.com/resources/testcollections/
[2] http://trec.nist.gov/data/t9_filtering.html
[3] http://people.csail.mit.edu/jrennie/20Newsgroups/

Table 2: Classification results on Reuters

Method	Reuters21578		TREC-9		20Newsgroup	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
LSI	0.9018	0.7319	0.8421	0.7714	0.4987	0.4901
CLSI	0.9371	0.7422	0.8723	0.8222	0.5175	0.5088
CM	0.9156	0.7275	0.8776	0.8315	0.5319	0.5274
LDA	0.9267	0.7305	0.8575	0.7828	0.5625	0.5571
DNE	0.9421	0.7875	0.9098	0.8504	0.6432	0.6396
LRWMMC	0.9552	0.8097	0.9115	0.8886	0.6616	0.6604

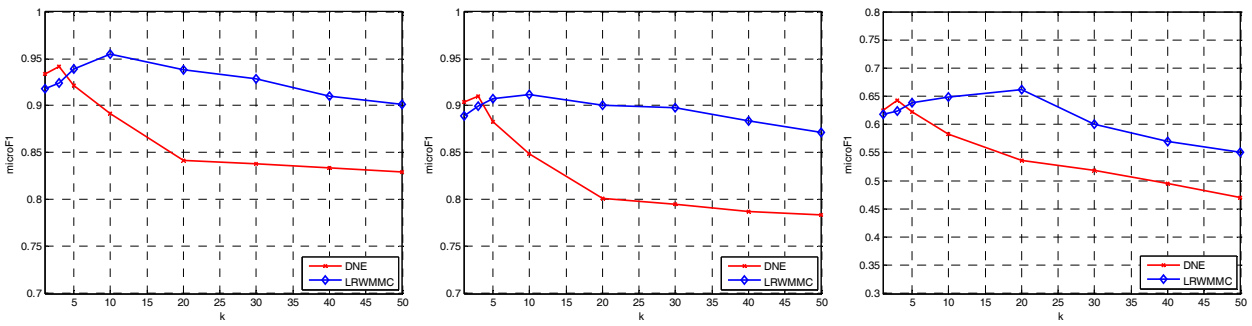

Figure 3: Classification results (microF1) with respect to the size of local relevant region, i.e. k.

Table 3: Classification results of kernel methods

Method	Reuters21578		TREC-9		20Newsgroup	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
LSK	0.9159	0.7892	0.8575	0.7875	0.5174	0.5119
KDA	0.9349	0.8067	0.8821	0.8016	0.5951	0.5872
Kernel LRWMMC	0.9656	0.8129	0.9271	0.9043	0.7127	0.7109

while the documents in the different classes as far as possible, in the local relevant region of each document.

8.5 Experiment 3 In this experiment, we compare Tensor LRWMMC with High Order SVD (HOSVD) [16]. HOSVD is the generalization of SVD to tensor space. As a result, HOSVD can be seen as generalized LSI for TSM. The construction of the Tensor Space Model (TSM) is according to [16]. More concretely, we use a 3-order tensor to represent each document and index this document by the 26 English letters. All the other characters except the 26 English letters are treated as the same symbol, denoted as $*$. The character string in each document is separated 3 characters by 3 characters with overlap. For example, we separate the following sentence,

It said it plans ...

as

It*,t*s,*sa,sai,aid,id*,d*i,*it,it*,...

Thus, each document is a $27\times27\times27$ 3-order tensor, and is normalized to unit length. And the corpus is a $27 \times 27 \times 27 \times n$ 4-order tensor, where n is the number of documents in the corpus.

The text classification results are shown in Table 4.

We can find that Tensor LRWMMC outperforms HOSVD on all the data sets. However, when we compare TSM with VSM in Table 2, we find that TSM does not always outperform VSM as we expected. The reason may be that the construction of the TSM is just a preliminary strategy [16]. In the future work, we will investigate other kinds of strategies for constructing the TSM.

8.6 Experiment 4 In this experiment, we compare Semi-Supervised LRWMMC with Co-EM [26] and Semi-Supervised Discriminant Analysis (SDA) [27]. Co-EM is the most popular inductive semi-supervised learning algorithm for text classification.

Table 4: Classification results with tensor representation

Method	Reuters21578		TREC-9		20Newsgroup	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
HOSVD	0.8514	0.6917	0.8271	0.7319	0.5172	0.5087
Tensor LRWMMC	0.9133	0.7649	0.8810	0.8275	0.6739	0.6529

The implementation of Semi-Supervised LRWMMC is based on QR-decomposition based fast algorithm. We randomly split the training set into labeled and "unlabeled" set for semi-supervised learning. And use the testing set for inductive classification. For Reuters21578, we vary the number of labeled samples by $\{200, 500, 1000, 2000, 3000, 4000, 5000\}$; for TREC-9, we vary the number of labeled samples by $\{200, 300, 400, 500, 600\}$; for 20Newsgroup, we vary the number of labeled samples by $\{2000, 4000, 6000, 8000, 10000\}$, and the rest samples are treated as "unlabeled". The regularizer λ in Semi-Supervised LRWMMC is set by searching the grid $\{4^{-3}, 4^{-2}, 4^{-1}, 4^0, 4^1, 4^2, 4^3\}$. The LRWMMC trained on the labeled samples is used as the baseline. Since the labeled samples are randomly chosen, we repeat this experiment 10 times and calculate the average result.

The results of inductive classification are shown in Figure 4.

We can find that Semi-Supervised LRWMMC outperforms the Co-EM and SDA on all the data sets. It should be noted that as the number of labeled samples increases, the performance of LRWMMC increases dramatically. For Reuters dataset, when the number of labeled samples increases to 4000, and for TREC9 dataset, when that increases to 500, LRWMMC even outperforms Co-EM and SDA. This strengthens the effectiveness of LRWMMC again.

9 Conclusion

The contributions of this paper include five folds: (1) We proposed a feature extraction method, named Local Relevance Weighted Maximum Margin Criterion (LRWMMC) for text classification; (2) We presented a Kernel LRWMMC to resolve the nonlinearity of the input space; (3) We presented a Tensor LRWMMC for TSM; (4) We presented a Semi-Supervised LRWMMC, which utilizes both the labeled and unlabeled samples. (5) We presented a fast algorithm for the methods proposed in this paper. Encouraging experimental results on benchmark text classification data sets indicate that the proposed methods are superior to many existing feature extraction methods for text classification.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (No.60721003, No.60673106 and No.60573062) and the Specialized Research Fund for the Doctoral Program of Higher Education. We thank the anonymous reviewers for their helpful comments.

References

[1] Fabrizio Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, 2002.

[2] Yiming Yang and Jan O. Pedersen, "A comparative study on feature selection in text categorization," in *ICML*, 1997, pp. 412–420.

[3] Tao Liu, Shengping Liu, Zheng Chen, and Wei-Ying Ma, "An evaluation on feature selection for text clustering," in *ICML*, 2003, pp. 488–495.

[4] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman, "Indexing by latent semantic analysis," *JASIS*, vol. 41, no. 6, pp. 391–407, 1990.

[5] David A. Hull, "Improving text retrieval for the routing problem using latent semantic indexing," in *SIGIR*, 1994, pp. 282–291.

[6] Hinrich Schütze, David A. Hull, and Jan O. Pedersen, "A comparison of classifiers and document representations for the routing problem," in *SIGIR*, 1995, pp. 229–237.

[7] Tao Liu, Zheng Chen, Benyu Zhang, Wei-Ying Ma, and Gongyi Wu, "Improving text classification using local latent semantic indexing," in *ICDM*, 2004, pp. 162–169.

[8] Efstratios Gallopoulos and Dimitrios Zeimpekis, "Clsi: A flexible approximation scheme from clustered term-document matrices," in *SDM*, 2005.

[9] Dimitrios Zeimpekis and Efstratios Gallopoulos, "Linear and non-linear dimensional reduction via class representatives for text classification," in *ICDM*, 2006, pp. 1172–1177.

[10] Deng Cai, Xiaofei He, and Jiawei Han, "Training linear discriminant analysis in linear time," in *ICDE*, 2008, pp. 209–217.

[11] Haifeng Li, Tao Jiang, and Keshu Zhang, "Efficient and robust feature extraction by maximum margin criterion," in *NIPS*, 2003.

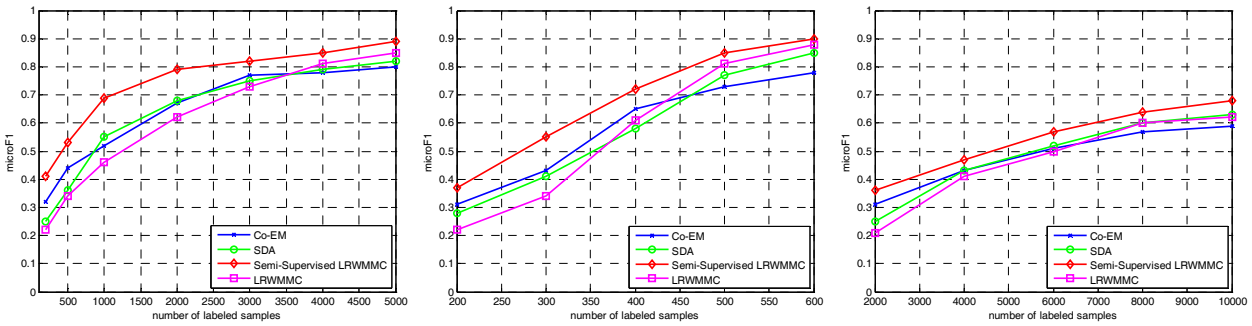

Figure 4: Inductive classification results (microF1) with respect to the number of labeled samples. The horizontal axis represents the number of randomly labeled data in the training set.

[12] Xiaofei He, Deng Cai, Haifeng Liu, and Wei-Ying Ma, "Locality preserving indexing for document representation," in *SIGIR*, 2004, pp. 96–103.

[13] Dell Zhang, Xi Chen, and Wee Sun Lee, "Text classification with kernels on the multinomial manifold," in *SIGIR*, 2005, pp. 266–273.

[14] B. Sch?lkopf and A.J. Smola, *Learning with Kernels*, MIT Press, Cambridge, MA, USA, 2002.

[15] Nello Cristianini, John Shawe-Taylor, and Huma Lodhi, "Latent semantic kernels," in *ICML*, 2001, pp. 66–73.

[16] Ning Liu, Benyu Zhang, Jun Yan, Zheng Chen, Wenyin Liu, Fengshan Bai, and Leefeng Chien, "Text representation: From vector to tensor," in *ICDM*, 2005, pp. 725–728.

[17] Xiaojin Zhu, "Semi-supervised learning literature survey," Tech. Rep., Computer Sciences, University of Wisconsin-Madison, 2005.

[18] Kari Torkkola, "Linear discriminant analysis in document classification," in *In IEEE ICDM Workshop on Text Mining*, 2001.

[19] Shuicheng Yan, Dong Xu, Benyu Zhang, HongJiang Zhang, Qiang Yang, and Stephen Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, 2007.

[20] Deng Cai, Xiaofei He, Kun Zhou, Jiawei Han, and Hujun Bao, "Locality sensitive discriminant analysis," in *IJCAI*, 2007, pp. 708–713.

[21] Wei Zhang, Xiangyang Xue, Zichen Sun, Yue-Fei Guo, and Hong Lu, "Optimal dimensionality of metric space for classification," in *ICML*, 2007, pp. 1135–1142.

[22] Shuicheng Yan, Dong Xu, Qiang Yang, Lei Zhang, Xiaoou Tang, and HongJiang Zhang, "Discriminant analysis with tensor representation," in *CVPR (1)*, 2005, pp. 526–532.

[23] Jimeng Sun, Spiros Papadimitriou, and Philip S. Yu, "Window-based tensor analysis on high-dimensional and multi-aspect streams," in *ICDM*, 2006, pp. 1076–1080.

[24] Jimeng Sun, Dacheng Tao, and Christos Faloutsos, "Beyond streams and graphs: dynamic tensor analysis," in *KDD*, 2006, pp. 374–383.

[25] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle, "On the best rank-1 and rank-(r1,r2,. . .,rn) approximation of higher-order tensors," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1324–1342, 2000.

[26] Kamal Nigam, Andrew McCallum, Sebastian Thrun, and Tom M. Mitchell, "Text classification from labeled and unlabeled documents using em," *Machine Learning*, vol. 39, no. 2/3, pp. 103–134, 2000.

[27] Deng Cai, Xiaofei He, and Jiawei Han, "Semi-supervised discriminant analysis," in *IEEE 11th International Conference on Computer Vision*, 2007, pp. 1–7.

[28] Xiaojin Zhu, Zoubin Ghahramani, and John D. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *ICML*, 2003, pp. 912–919.

[29] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf, "Learning with local and global consistency," in *NIPS*, 2003.

[30] Gene H. Golub and Charles F. Van Loan, *Matrix Computation*, Johns Hopkins University Press, Baltimore, Maryland, third edition, 1996.

[31] Jieping Ye, Qi Li, Hui Xiong, Haesun Park, Ravi Janardan, and Vipin Kumar, "Idr/qr: an incremental dimension reduction algorithm via qr decomposition," in *KDD*, 2004, pp. 364–373.

[32] Haixian Wang, Wenming Zheng, Zilan Hu, and Sibao Chen, "Local and weighted maximum margin discriminant analysis," in *CVPR*, 2007.

[33] D. Zeimpekis and E. Gallopoulos, "Tmg: A matlab toolbox for generating term-document matrices from text collections," *Grouping Multidimensional Data: Recent Advances in Clustering*, pp. 187–210, 2006.

[34] Deng Cai, Xiaofei He, and Jiawei Han, "Efficient kernel discriminant analysis via spectral regression," in *ICDM*, 2007, pp. 427–432.