

Multi-View Local Learning

Dan Zhang¹, Fei Wang¹, Changshui Zhang², Tao Li³

^{1,2}State Key Laboratory on Intelligent Technology and Systems,
Tsinghua National Laboratory for Information Science and Technology (TNList)
Department of Automation, Tsinghua University, Beijing, 100084, China.

¹{dan-zhang05, feiwang03}@mails.thu.edu.cn, ²zcs@mail.thu.edu.cn

³School of Computer Science, Florida International University, Miami, FL 33199, U.S.A.
taoli@cs.fiu.edu

Abstract

The idea of local learning, *i.e.*, classifying a particular example based on its neighbors, has been successfully applied to many semi-supervised and clustering problems recently. However, the local learning methods developed so far are all devised for single-view problems. In fact, in many real-world applications, examples are represented by multiple sets of features. In this paper, we extend the idea of local learning to multi-view problem, design a multi-view local model for each example, and propose a Multi-View Local Learning Regularization (MVLL-Reg) matrix. Both its linear and kernel version are given. Experiments are conducted to demonstrate the superiority of the proposed method over several state-of-the-art ones.

Introduction

In many real-world applications, examples are represented by multiple sets of features (views). For example, in web mining problems, each web-page has disparate descriptions: textual content, in-bound and out-bound links, etc. Since different sets of features could have different statistical properties, it is a challenging problem to utilize them together in machine learning.

A very common method to deal with the multi-view problem is to define a kernel for each view, and convexly combine them (Joachims, Cristianini, & Shawe-Taylor 2001) (Zhang, Popescul, & Dom 2006). Then, a kernel machine can be adopted for classification based on such a combined kernel. Recently, another type of methods, which is based on data graphs have aroused considerable interests in the machine learning and data mining community. When these methods are concerned, a natural approach is to convexly combine the graph Laplacians on different views (Sindhwani, Niyogi, & Belkin 2005) (Argyriou, Herbster, & Pontil 2005) (Tsuda, Shin, & Schölkopf 2005), since the pseudo-inverse of the graph Laplacian can be deemed as a kernel (Smola & Kondor 2003). In (Zhou & Burges 2007), the authors consider the spectral clustering and transductive learning problems on multiple directed graphs, with the undirected graph as its special case. The mincut criterion seems natural for multi-view directed graphs. However, when it

comes to undirected graphs, their method also equals to convexly combining the graph Laplacian on each view. Although the above algorithms are quite reasonable, they are all global ones. As pointed out by Vapnik (Vapnik 1999), it is usually not easy to find a good predictor in the whole input space.

In fact, in (Bottou & Vapnik 1992), the authors have pointed out that the local learning algorithms often outperform global ones. This is because nearby examples are more likely generated by the same data generation mechanism, while far away examples tend to differ in it. Furthermore, in (Yu & Shi 2003), it is proposed that locality is very crucial for capacity control. Inspired by these works, the idea of local learning has been employed widely in semi-supervised learning (Wu & Schölkopf 2007), clustering (Wang, Zhang, & Li 2007) (Wu & Schölkopf 2006), dimensionality reduction (Wu *et al.* 2007), etc.

In this paper, we will show that the idea of local learning can also be utilized to improve the performances of multi-view semi-supervised learning and multi-view clustering. To achieve this goal, we design a local multi-view model for each example, and use these local models to classify the unlabeled examples. We will demonstrate that this is equivalent to designing a new regularization matrix that can not be simply considered by the convex combination of multiple Laplacian matrix on each view. We name it Multi-View Local Learning Regularization (MVLL-Reg) matrix.

The rest of the paper is organized as follows: In Section 2, we introduce the problem statement and notations. The proposed algorithm will be elaborated in Section 3. In Section 4, the experimental results are presented. In the end, conclusions will be drawn in Section 5.

Problem Statements and Notations

Without loss of generality, in this paper, we consider only the two-view problem. For the semi-supervised classification task, we are given l labeled examples: $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l)$, and u ($l \ll u$) unlabeled ones: $\mathbf{x}_{l+1}, \dots, \mathbf{x}_{l+u}$. Each example $\mathbf{x}_i = (\mathbf{x}_{i(1)}, \mathbf{x}_{i(2)})$ is seen in two views with $\mathbf{x}_{i(1)} \in \mathbb{X}_{(1)}$ and $\mathbf{x}_{i(2)} \in \mathbb{X}_{(2)}$. y_i is the class label and can be taken from c classes, *i.e.*, $y_i \in \{1, 2, \dots, c\}$. The goal is to derive the labels on these unlabeled examples.

For the multi-view clustering task, we are given a set of n two-view examples: $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$. The goal is to par-

l	The number of labeled examples.
u	The number of unlabeled examples.
n	The total number of examples $n = l + u$
$\mathcal{N}(\mathbf{x}_{i(k)})$	The neighbors of $\mathbf{x}_{i(k)}$ on the k -th view
$n_{i(k)}$	The cardinality of $\mathcal{N}(\mathbf{x}_{i(k)})$

Table 1: Frequently used notations

tition this given dataset into c clusters, such that different clusters are in some sense "distinct" from each other.

Table 1 shows some symbols and notations that will be frequently used throughout this paper.

The Proposed Method

In a traditional supervised single-view classification problem, the final classifier is trained by all the labeled examples, and this classifier is used to classify the unlabeled examples. Unlike those traditional methods, under the local learning setting, for each example, we need to train a local model by the examples that lie within its local region¹. Each unlabeled example should be classified by one of these local models. So, the classification of a specific unlabeled example will be only related to the nearby examples and will not be affected by the examples that lie far away.

But in order to design a multi-view local learning method, we need to solve two problems: 1. how to define the local region under the multi-view setting. 2. how to handle so many multi-view local models, and use them to classify the unlabeled examples. Next, we will elaborate our method.

The Local Region

For a multi-view local learning method, there is a natural problem: for an example \mathbf{x}_i , since it has two views and on each view the neighborhood situation is most likely to be different, how to define its nearby examples? To solve this problem, we give the following empirical definition:

Definition: For a multi-view example \mathbf{x}_i , its neighbors are defined as the union of the neighbors on each independent view, *i.e.*, $\mathcal{N}(\mathbf{x}_i) = \mathcal{N}(\mathbf{x}_{i(1)}) \cup \mathcal{N}(\mathbf{x}_{i(2)})$ and the local region for \mathbf{x}_i is the region spanned by $\mathcal{N}(\mathbf{x}_i)$.

The Local Model

In this section, for simplicity, we focus on the regression problem, where a real valued f_i , $1 \leq i \leq l + u$, is assigned to each data point \mathbf{x}_i .

For an example \mathbf{x}_i , suppose the output of its corresponding local model on the k th view takes the form as follows:

$$o_{i(k)}(\mathbf{x}) = \mathbf{w}_{i(k)}^T(\mathbf{x}_{(k)} - \mathbf{x}_{i(k)}) + b_{i(k)}, \quad (1)$$

where, the subscript i indicates that this local learning model is trained by the examples that lie in the local region of \mathbf{x}_i , *i.e.*, $\mathbf{x}_i \in \mathcal{N}(\mathbf{x}_i)$. Then, by utilizing the soft labels of these examples, how can we train this local linear model?

¹In a single-view local learning problem, the local region often refers to the region spanned by the several nearest neighbors of the example.

In the local region for \mathbf{x}_i , we devise a local regressor that minimize the error on the examples that lie within this local region and the disagreement between the output of different views on \mathbf{x}_i (In this local model, \mathbf{x}_i is the only unlabeled example). Then, this local model can be trained by solving the following optimization problem:

$$\min_{\mathbf{w}_{i(1)}, \mathbf{w}_{i(2)}, b_{i(1)}, b_{i(2)}} G(\mathbf{w}_{i(1)}, \mathbf{w}_{i(2)}, b_{i(1)}, b_{i(2)}) \quad (2)$$

where,

$$\begin{aligned} & G(\mathbf{w}_{i(1)}, \mathbf{w}_{i(2)}, b_{i(1)}, b_{i(2)}) \\ &= \lambda_1 \|\mathbf{w}_{i(1)}\|^2 \\ &+ \sum_{\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i)} (\mathbf{w}_{i(1)}^T(\mathbf{x}_{j(1)} - \mathbf{x}_{i(1)}) + b_{i(1)} - f_j)^2 \\ &+ \lambda_2 \|\mathbf{w}_{i(2)}\|^2 \\ &+ \sum_{\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i)} (\mathbf{w}_{i(2)}^T(\mathbf{x}_{j(2)} - \mathbf{x}_{i(2)}) + b_{i(2)} - f_j)^2 \\ &+ \lambda_3 (b_{i(1)} - b_{i(2)})^2 \end{aligned}$$

In this optimization problem, the first four terms can be deemed as the objective function for regularized least squares defined on two views, and the last term encodes the requirement that the output of different views should not deviate too much on the unlabeled example \mathbf{x}_i (Note that $o_{i(k)}(\mathbf{x}_i)$ equals the bias $b_{i(k)}$). In fact, this formulation embodies two multi-view assumptions, which are presented in (Blum & Mitchell 1998): a) the assumption that different views are independent given the labels (independence assumption), b) the assumption that the output on each view should not deviate too much on most of the examples (comparability assumption). In fact, the same motivation has been employed in (Brefeld *et al.* 2006). But their method is a global one.

Furthermore, it should be noticed that one of the theoretical results in (Rosenberg & Bartlett 2007) is that, for the co-regularized kernel class \mathcal{J} , the empirical Rademacher complexity are bounded and this bound can be reduced with the help of the unlabeled data. The form of our local optimization problem is exactly the same as in that paper. Then, since, in our proposed method, when designing a local multi-view function for each example \mathbf{x} and its nearest neighbors, the only unlabeled example is \mathbf{x} , the last term in Eq.(2) that requires different views to agree on this example can also reduce the Rademacher complexity on this multi-view local model.

Eq.(2) is a convex optimization problem with respect to $\mathbf{w}_{i(1)}$, $\mathbf{w}_{i(2)}$, $b_{i(1)}$ and $b_{i(2)}$. By taking the derivative of $G(\mathbf{w}_{i(1)}, b_{i(1)}, \mathbf{w}_{i(2)}, b_{i(2)})$ with respect to $\mathbf{w}_{i(1)}$, $\mathbf{w}_{i(2)}$, $b_{i(1)}$, $b_{i(2)}$, and let them be zero, we get:

$$b_{i(1)} = \frac{\mathbf{c}_{11}\mathbf{f}_i + \lambda_3 b_{i(2)}}{c_{12}}, \quad b_{i(2)} = \frac{\mathbf{c}_{21}\mathbf{f}_i + \lambda_3 b_{i(1)}}{c_{22}} \quad (3)$$

where,

$$\begin{aligned}
\mathbf{c}_{11} &= \mathbf{1}^T - \mathbf{1}^T \mathbf{X}_{i(1)}^T (\lambda_1 \mathbf{I} + \mathbf{X}_{i(1)} \mathbf{X}_{i(1)}^T)^{-1} \mathbf{X}_{i(1)} \\
&= \mathbf{1}^T - \mathbf{1}^T \mathbf{X}_{i(1)}^T \mathbf{X}_{i(1)} (\lambda_1 \mathbf{I} + \mathbf{X}_{i(1)}^T \mathbf{X}_{i(1)})^{-1} \\
\mathbf{c}_{12} &= n_i + \lambda_3 - \mathbf{1}^T \mathbf{X}_{i(1)}^T (\lambda_1 \mathbf{I} + \mathbf{X}_{i(1)} \mathbf{X}_{i(1)}^T)^{-1} \mathbf{X}_{i(1)} \mathbf{1} \\
&= n_i + \lambda_3 - \mathbf{1}^T \mathbf{X}_{i(1)}^T \mathbf{X}_{i(1)} (\lambda_1 \mathbf{I} + \mathbf{X}_{i(1)}^T \mathbf{X}_{i(1)})^{-1} \mathbf{1} \\
\mathbf{c}_{21} &= \mathbf{1}^T - \mathbf{1}^T \mathbf{X}_{i(2)}^T (\lambda_2 \mathbf{I} + \mathbf{X}_{i(2)} \mathbf{X}_{i(2)}^T)^{-1} \mathbf{X}_{i(2)} \\
&= \mathbf{1}^T - \mathbf{1}^T \mathbf{X}_{i(2)}^T \mathbf{X}_{i(2)} (\lambda_2 \mathbf{I} + \mathbf{X}_{i(2)}^T \mathbf{X}_{i(2)})^{-1} \\
\mathbf{c}_{22} &= n_i + \lambda_3 - \mathbf{1}^T \mathbf{X}_{i(2)}^T (\lambda_2 \mathbf{I} + \mathbf{X}_{i(2)} \mathbf{X}_{i(2)}^T)^{-1} \mathbf{X}_{i(2)} \mathbf{1} \\
&= n_i + \lambda_3 - \mathbf{1}^T \mathbf{X}_{i(2)}^T \mathbf{X}_{i(2)} (\lambda_2 \mathbf{I} + \mathbf{X}_{i(2)}^T \mathbf{X}_{i(2)})^{-1} \mathbf{1}
\end{aligned} \tag{4}$$

where, n_i is the cardinality of $\mathcal{N}(\mathbf{x}_i)$. \mathbf{I} is the identity matrix and $\mathbf{f}_i \in \mathbb{R}^{n_i}$ is the vector $[f_j]^T$ for $\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i)$. $\mathbf{1}$ is the column vector of all 1's, $\mathbf{X}_{i(1)} \in \mathbb{R}^{d \times n_i}$ denotes the matrix $[\mathbf{x}_{j(1)} - \mathbf{x}_{i(1)}]$ for $\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i)$, and $\mathbf{X}_{i(2)}$ refers to $[\mathbf{x}_{j(2)} - \mathbf{x}_{i(2)}]$ for $\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i)$, accordingly. By solving Eq.(3), the output for \mathbf{x}_i can be determined by:

$$\begin{aligned}
o_i(\mathbf{x}_i) &= \frac{b_{i(1)} + b_{i(2)}}{2} \\
&= \frac{\mathbf{c}_{11} \mathbf{c}_{22} + \lambda_3 \mathbf{c}_{21} + \mathbf{c}_{21} \mathbf{c}_{12} + \lambda_3 \mathbf{c}_{11}}{2(\mathbf{c}_{12} \mathbf{c}_{22} - \lambda_3^2)} \mathbf{f}_i \\
&= \alpha_i \mathbf{f}_i
\end{aligned} \tag{5}$$

Multi-View Local Learning Regularization Matrix

In the previous section, we have elaborated how to train a multi-view local model for \mathbf{x}_i , and approximate the output of \mathbf{x}_i by this local model. We require that this approximation of the soft label should not deviate too much from its actual soft label, *i.e.*, f_i . By taking the quadratic loss, the regularizer term takes the form:

$$\begin{aligned}
\sum_{i=1}^{l+u} (f_i - o_i(\mathbf{x}_i))^2 &= \|\mathbf{f} - \mathbf{o}\|^2 = \|\mathbf{f} - \mathbf{A}\mathbf{f}\|^2 \\
&= \mathbf{f}^T (\mathbf{I} - \mathbf{A})^T (\mathbf{I} - \mathbf{A}) \mathbf{f} \\
&= \mathbf{f}^T \mathbf{L}_{MVL} \mathbf{f},
\end{aligned} \tag{6}$$

where, \mathbf{A} is an $n \times n$ matrix, with its element a_{ij} equals the corresponding element of α_i if $\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i)$, otherwise, a_{ij} equals zero. $\mathbf{o} = [o_1(\mathbf{x}_1), \dots, o_n(\mathbf{x}_n)]^T$ and $\mathbf{f} = [f_1, f_2, \dots, f_n]^T$. \mathbf{L}_{MVL} is the proposed Multi-View Local Learning Regularization (MVL-Reg) matrix. It should be noted that the calculation of \mathbf{A} in Eq. (6) needs calculating α_i in Eq.(5) for each example. The time complexity for calculating α_i is $O(2n_i^3)$. Then, the total time complexity for calculating \mathbf{A} can be determined by $O(\sum_{i=1}^n 2n_i^3)$.

So far, we have obtained the Multi-View Local Learning Regularization matrix. It is very convenient to apply this regularization matrix to semi-supervised learning and clustering problems.

In this paper, for semi-supervised multi-class learning, we employ the following framework:

$$\min_{\mathbf{F} \in \mathbb{R}^{l+u}} \text{tr} (\mathbf{F}^T \mathbf{L}_{MVL} \mathbf{F} + (\mathbf{F} - \mathbf{Y})^T \mathbf{C} (\mathbf{F} - \mathbf{Y})), \tag{7}$$

where, \mathbf{Y} is an $n \times c$ matrix, with \mathbf{Y}_{ik} equals 1 if \mathbf{x}_i is labeled and belongs to the k -th class, \mathbf{Y}_{ik} equals -1 if \mathbf{x}_i is labeled and does not belong to the k -th class, \mathbf{Y}_{ik} equals zero if \mathbf{x}_i is unlabeled. $\mathbf{C} \in \mathbb{R}^{n \times n}$ is a diagonal matrix, with its i -th diagonal element c_i being computed as: $c_i = C_l > 0$ for $1 \leq i \leq l$, and $c_i = C_u \geq 0$ for $l+1 \leq i \leq l+u$, where C_l and C_u are two parameters that control the penalty imposed on the labeled and unlabeled examples, respectively. In most cases, C_u equals zero. $\text{tr}(\cdot)$ stands for the trace of a matrix. \mathbf{F} is the estimated real valued label matrix, and the final classification result can be obtained by $\arg \max_j (\mathbf{F}_{ij}), l+1 \leq i \leq l+u$. The optimal solution for \mathbf{F} can be obtained by:

$$\mathbf{F}^* = (\mathbf{L}_{MVL} + \mathbf{C})^{-1} \mathbf{C} \mathbf{Y} \tag{8}$$

Note that since the estimated output for each example only rely on the several examples within its local region, \mathbf{A} will be sparse, and so will be \mathbf{L}_{MVL} . This means we can solve Eq.(8) more efficiently by using some algebraic methods (*e.g.*, the Lanczos iteration).

As for the clustering problem, it can be transformed to the following optimization problem (Yu & Shi 2003) (Chan, Schlag, & Zien 1994):

$$\begin{aligned}
\min_{\mathbf{H} \in \mathbb{R}^{n \times c}} \text{tr} (\mathbf{H}^T \mathbf{L}_{MVL} \mathbf{H}) \\
\text{subject to } \mathbf{H}^T \mathbf{H} = \mathbf{I},
\end{aligned} \tag{9}$$

where, $\mathbf{H} \in \mathbb{R}^{n \times c}$ is a continuous relaxation of the partition matrix. From the *Ky Fan* theorem (Zha *et al.* 2001), we know the optimal value of the above problem is

$$\mathbf{H}^* = [\mathbf{h}_1^*, \mathbf{h}_2^*, \dots, \mathbf{h}_c^*] \mathbf{R}, \tag{10}$$

where \mathbf{h}_k^* ($1 \leq k \leq c$) is the eigenvector corresponds to the k -th smallest eigenvalue of matrix \mathbf{L}_{MVL} , and \mathbf{R} is an arbitrary $c \times c$ matrix. Since the values of the entries in \mathbf{H}^* is continuous, we need to further discretize \mathbf{H}^* to get the cluster assignments of all the examples. There are mainly two approaches to achieve this goal:

1. Note that the optimal \mathbf{H}^* is not unique (because of the existence of an arbitrary matrix \mathbf{R}). Thus, we can pursue an optimal \mathbf{R} that will rotate \mathbf{H}^* to an indication matrix². The detailed algorithm can be referred to (Yu & Shi 2003).
2. As in (Ng, Jordan, & Weiss 2001), we can treat the i -th row of \mathbf{H} as the embedding of \mathbf{x}_i in a c -dimensional space, and apply some traditional clustering methods like kmeans to clustering these embeddings into c clusters.

A Kernel Version

The previous analysis is based on the assumption that the local models are linear. We can also design, on each view, a local kernel ridge function for each example. For \mathbf{x}_i , on the k -th view, the output of this local model can be defined as:

$$o_{i(k)}(\mathbf{x}_i) = \sum_{\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i)} \beta_{ij(k)} K(\mathbf{x}_{i(k)}, \mathbf{x}_{j(k)}), \tag{11}$$

²An indication matrix \mathbf{T} is a $n \times c$ matrix with its (i, j) -th entry $\mathbf{T}_{ij} \in \{0, 1\}$ such that for each row of \mathbf{T}^* there is only one 1. In this way, \mathbf{x}_i can be assigned to the j -th cluster such that $j = \arg_j \mathbf{T}_{ij}^* = 1$.

where, $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is a positive definite kernel function (Scholkopf & Smola 2002). $\beta_{i,j(k)}$ is the corresponding expansion coefficients on the k -th view. Then, for each example \mathbf{x}_i , its local model can be trained by solving the following optimization problem:

$$\min_{\beta_{i(1)}, \beta_{i(2)}} G(\beta_{i(1)}, \beta_{i(2)}) \quad (12)$$

where

$$\begin{aligned} G(\beta_{i(1)}, \beta_{i(2)}) &= \lambda_1 \beta_{i(1)}^T \mathbf{K}_{i(1)} \beta_{i(1)} + \|\mathbf{K}_{i(1)} \beta_{i(1)} - \mathbf{f}_i\|^2 \\ &+ \lambda_2 \beta_{i(2)}^T \mathbf{K}_{i(2)} \beta_{i(2)} + \|\mathbf{K}_{i(2)} \beta_{i(2)} - \mathbf{f}_i\|^2 \\ &+ \lambda_3 (\mathbf{K}_{i(1)}^i \beta_{i(1)} - \mathbf{K}_{i(2)}^i \beta_{i(2)})^2 \\ &= \lambda_1 \beta_{i(1)}^T \mathbf{K}_{i(1)} \beta_{i(1)} + (\mathbf{K}_{i(1)} \beta_{i(1)} - \mathbf{f}_i)^T (\mathbf{K}_{i(1)} \beta_{i(1)} - \mathbf{f}_i) \\ &+ \lambda_2 \beta_{i(2)}^T \mathbf{K}_{i(2)} \beta_{i(2)} + (\mathbf{K}_{i(2)} \beta_{i(2)} - \mathbf{f}_i)^T (\mathbf{K}_{i(2)} \beta_{i(2)} - \mathbf{f}_i) \\ &+ \lambda_3 (\mathbf{K}_{i(1)}^i \beta_{i(1)} - \mathbf{K}_{i(2)}^i \beta_{i(2)})^2 \end{aligned}$$

In this formulation, $\beta_{i(k)}$ is a n_i -dimensional vector with the j -th element being $\beta_{i,j(k)}$. $\mathbf{K}_{i(1)}$ is a $n_i \times n_i$ matrix, with $\mathbf{K}_{i(1)}^{(m,n)}$ equals $K(\mathbf{x}_{m(1)}, \mathbf{x}_{n(1)})$, $\mathbf{x}_m, \mathbf{x}_n \in \mathcal{N}(\mathbf{x}_i)$, and $\mathbf{K}_{i(1)}^i$ is a row vector with its j -th value being $K(\mathbf{x}_{i(1)}, \mathbf{x}_{j(1)})$, $\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i)$. $K_{i(2)}$ and $K_{i(2)}^i$ are also defined likewise. Like its linear version, the first four terms are the regularized least squares defined on two views, and the last term prevents the outputs of the unlabeled example \mathbf{x}_i on different views from deviating too much. The optimal parameters $\beta_{i(1)}, \beta_{i(2)}$ for the local model can be acquired by taking the derivative of $G(\beta_{i(1)}, \beta_{i(2)})$ with respect to $\beta_{i(1)}$ and $\beta_{i(2)}$, and let them be zero. The optimal solutions are expressed as:

$$\begin{aligned} \beta_{i(1)} &= (\mathbf{I} - \mathbf{c}_{12} \mathbf{c}_{22})^{-1} (\mathbf{c}_{11} + \mathbf{c}_{12} \mathbf{c}_{21}) \mathbf{f}_i \\ \beta_{i(2)} &= (\mathbf{I} - \mathbf{c}_{22} \mathbf{c}_{12})^{-1} (\mathbf{c}_{21} + \mathbf{c}_{22} \mathbf{c}_{11}) \mathbf{f}_i \end{aligned}$$

where,

$$\begin{aligned} \mathbf{c}_{11} &= (\lambda_1 \mathbf{K}_{i(1)} + \mathbf{K}_{i(1)} \mathbf{K}_{i(1)} + \lambda_3 \mathbf{K}_{i(1)}^i \mathbf{K}_{i(1)}^i)^{-1} \mathbf{K}_{i(1)} \\ \mathbf{c}_{12} &= \lambda_3 (\lambda_1 \mathbf{K}_{i(1)} + \mathbf{K}_{i(1)} \mathbf{K}_{i(1)} + \lambda_3 \mathbf{K}_{i(1)}^i \mathbf{K}_{i(1)}^i)^{-1} \\ &\quad \times \mathbf{K}_{i(1)}^i \mathbf{K}_{i(2)}^i \\ \mathbf{c}_{21} &= (\lambda_2 \mathbf{K}_{i(2)} + \mathbf{K}_{i(2)} \mathbf{K}_{i(2)} + \lambda_3 \mathbf{K}_{i(2)}^i \mathbf{K}_{i(2)}^i)^{-1} \mathbf{K}_{i(2)} \\ \mathbf{c}_{22} &= \lambda_3 (\lambda_2 \mathbf{K}_{i(2)} + \mathbf{K}_{i(2)} \mathbf{K}_{i(2)} + \lambda_3 \mathbf{K}_{i(2)}^i \mathbf{K}_{i(2)}^i)^{-1} \\ &\quad \times \mathbf{K}_{i(2)}^i \mathbf{K}_{i(1)}^i \end{aligned}$$

The output for \mathbf{x}_i can be obtained by:

$$\begin{aligned} o_i(\mathbf{x}_i) &= \frac{o_{i(1)}(\mathbf{x}_i) + o_{i(2)}(\mathbf{x}_i)}{2} = \frac{\mathbf{K}_{i(1)}^i \beta_{i(1)} + \mathbf{K}_{i(2)}^i \beta_{i(2)}}{2} \\ &= \frac{\mathbf{K}_{i(1)}^i (\mathbf{I} - \mathbf{c}_{12} \mathbf{c}_{22})^{-1} (\mathbf{c}_{11} + \mathbf{c}_{12} \mathbf{c}_{21}) \mathbf{f}_i}{2} \\ &\quad + \frac{\mathbf{K}_{i(2)}^i (\mathbf{I} - \mathbf{c}_{22} \mathbf{c}_{12})^{-1} (\mathbf{c}_{21} + \mathbf{c}_{22} \mathbf{c}_{11}) \mathbf{f}_i}{2} \\ &= \alpha_i \mathbf{f}_i \end{aligned} \quad (13)$$

Still, the form of α_i is independent of \mathbf{f}_i and we can also get a concrete form:

$$\mathbf{o} = \mathbf{A} \mathbf{f} \quad (14)$$

The definition of \mathbf{A} is the same as that in Eq.(6). Like its linear version, \mathbf{L}_{MVL} is also defined as $(\mathbf{I} - \mathbf{A})^T (\mathbf{I} - \mathbf{A})$.

Experiments

Dataset Description

We use two real world datasets to evaluate the performances of the proposed method. Table 2 summarizes the characteristics of these datasets.

The WebKB dataset³ consists of about 6000 web pages from computer science department in four Universities, *i.e.*, Cornell, Texas, Washington, and Wisconsin. These pages are categorized into seven categories. The Cora dataset (McCallum *et al.* 2000) consists of the abstracts and references of around 34,000 computer science papers. Part of them are categorized into one of subfields of Data Structure (DS), Hardware and Architecture (HA), Machine Learning (ML), Operation Systems (OS) and Programming Language (PL). Since the main objective of our paper is not to investigate how to utilize the information hidden in the link structures (Zhu *et al.* 2007), we treat links as the features of each document, *i.e.*, for a feature vector, its i -th feature is *link-to-page_i*. In this way, the features of the examples can be split into two views: the content features (View1) and link features (View2).

Datasets		Sizes	Classes	View1	View2
WebKB	Cornell	827	7	4134	827
	Washington	1166	7	4165	1166
	Wisconsin	1210	7	4189	1210
	Texas	814	7	4029	814
Cora	DS	751	9	6234	751
	HA	400	7	3989	400
	ML	1617	7	8329	1617
	OS	1246	4	6737	1246
	PL	1575	9	7949	1575

Table 2: The detailed description of the datasets. View1 is the content dimension, while View2 stands for the link dimension.

Classification

Methods We compare the classification performances of MVLL Regularization (MVLL-Reg) matrix with Laplacian Regularization (Lap-Reg) matrix (Zhu, Ghahramani, & Lafferty 2003), Normalized Laplacian Regularization (NLap-Reg) matrix (Zhou *et al.* 2003), LLE Regularization (LLE-Reg) matrix (Wang & Zhang 2006) and the linear version Local Learning Regularization (LL-Reg) matrix (Wu & Schölkopf 2007). Except the MVLL-Reg, the other regularization matrices are not specifically designed for the multi-view learning. Therefore, we adopt the most commonly used strategy, as mentioned in the introduction, *i.e.*, convexly

³CMU world wide knowledge base (WebKB) project. Available at <http://www.cs.cmu.edu/ WebKB/>

	Cornel	Washington	Wisconsin	Texas	DS	HA	ML	OS	PL
MVLL-Reg	91.25 ± 2.02	92.52 ± 1.86	88.15 ± 1.49	94.85 ± 2.36	45.00 ± 2.13	58.27 ± 2.40	54.26 ± 1.44	61.32 ± 2.73	45.84 ± 1.57
Lap-Reg-content	70.68 ± 2.39	76.65 ± 8.49	73.94 ± 0.30	66.09 ± 11.32	23.03 ± 1.94	24.09 ± 2.37	23.78 ± 2.76	43.82 ± 1.08	24.14 ± 5.27
Lap-Reg-link	86.98 ± 1.98	83.32 ± 4.52	80.61 ± 4.71	81.50 ± 4.68	35.11 ± 2.79	34.95 ± 2.02	43.43 ± 1.45	51.47 ± 2.12	37.95 ± 1.89
Lap-Reg-content+link	87.15 ± 3.55	86.37 ± 2.74	82.97 ± 0.30	84.99 ± 5.98	32.52 ± 1.85	52.39 ± 2.21	40.73 ± 1.09	51.42 ± 0.89	36.16 ± 1.36
NLap-Reg-content	71.86 ± 1.96	77.55 ± 1.96	74.12 ± 0.53	69.04 ± 7.28	25.31 ± 1.89	25.64 ± 2.39	31.63 ± 2.30	44.59 ± 2.78	17.05 ± 4.53
NLap-Reg-link	86.95 ± 2.30	85.53 ± 2.85	80.71 ± 3.23	78.65 ± 7.71	40.82 ± 2.35	34.27 ± 2.88	43.70 ± 3.54	55.58 ± 1.67	34.99 ± 3.03
Nlap-Reg-content+link	87.29 ± 5.16	86.70 ± 3.11	83.22 ± 0.61	79.15 ± 6.14	43.67 ± 2.40	53.34 ± 1.96	58.65 ± 2.46	59.25 ± 1.85	48.94 ± 3.20
LLE-Reg-content	76.35 ± 2.23	82.72 ± 2.11	79.58 ± 1.92	72.00 ± 3.02	26.82 ± 1.51	27.78 ± 1.62	32.87 ± 0.98	44.98 ± 0.98	28.10 ± 0.95
LLE-Reg-link	71.52 ± 6.70	64.89 ± 8.17	59.39 ± 5.18	73.32 ± 4.84	41.29 ± 2.13	39.32 ± 2.49	50.20 ± 1.39	56.26 ± 2.05	42.89 ± 1.49
LLE-Reg-content+link	87.90 ± 1.59	86.60 ± 1.46	84.97 ± 2.52	85.69 ± 3.41	41.81 ± 1.66	53.59 ± 1.75	51.01 ± 1.65	56.15 ± 2.10	43.79 ± 1.69
LL-Reg-content	76.19 ± 2.11	82.79 ± 2.20	79.64 ± 1.83	73.12 ± 2.63	27.74 ± 1.45	28.50 ± 1.87	32.88 ± 1.14	45.01 ± 1.00	28.08 ± 0.95
LL-Reg-link	87.41 ± 2.42	84.74 ± 3.15	81.30 ± 2.39	81.76 ± 4.30	41.80 ± 2.04	39.74 ± 2.24	50.59 ± 1.48	56.79 ± 1.69	43.32 ± 1.37
LL-Reg-content+link	88.39 ± 1.16	87.02 ± 2.18	84.29 ± 2.64	86.70 ± 3.60	43.15 ± 3.00	56.26 ± 2.44	53.17 ± 2.30	57.44 ± 2.07	43.75 ± 1.77

Table 3: Average accuracies (%) and the standard deviations (%) on the WebKB and Cora dataset.

combine the regularization matrices on each view. Among all our experiments, the combination coefficient are tuned using the grid search method with five-fold cross validation. We also compare the performances when different kinds of regularization matrices are employed in each view. We have introduced two kinds of MVLL-Reg matrices, *i.e.*, the linear version and the kernel version. For the classification tasks, we adopt the linear version for convenience.

For the WebKB dataset, we randomly choose 5% examples as the labeled data and the others are left as the unlabeled set. On the Cora data set, 30% examples are randomly selected as the labeled set, and the others are left as the unlabeled ones. All the parameters are determined by 5-fold cross validation. We measure the results by the classification accuracy, *i.e.*, the percentage of the number of correctly classified documents. The final results are averaged over 50 independent runs and the standard deviation are also given.

Classification Results Among all the experiments, Lap-Reg, NLap-Reg, LLE-Reg, LL-Reg and MVLL-Reg only differ in the design of the regularization matrix. Eq.(7) is used as the basic classification framework.

As can be seen from the classification results, MVLL-Reg performs better than the other methods in most cases. This shows that our method is more adapted to the multi-view problem, and the idea of local learning is beneficial in designing multi-view learning methods.

Clustering

Methods For the clustering tasks, we adopt the kernel version of MVLL-Reg for convenience. We compare the proposed algorithm with K-means, Lap-Reg, NLap-Reg (Yu & Shi 2003), and the kernel version LL-Reg (Wu & Schölkopf 2006). Still, except MVLL-Reg, the other methods are not specifically designed for multi-view learning. For K-means, the content features and link features are concatenated together and K-means is performed on these concatenated features. For Lap-Reg, NLap-Reg, and the kernel version LL-Reg, the Laplacian matrix are obtained by combining the graph Laplacians on each view with equal weights.

The clustering experiments are conducted on the WebKB data set. For all these methods, the weights on data graph edges are computed by Gaussian functions, the variance of which is determined by local scaling (Zelnik-Manor & Perona 2004). The parameters λ_1 , λ_2 are determined by searching the grid $\{0.1, 1, 10\}$ and the neighborhood size

for each view are also set to the same by searching the grid $\{10, 20, 40\}$. The parameter λ_3 is set by searching $\{0.001, 0.01, 0.1, 1, 10\}$. For MVLL-Reg, Lap-Reg, NLap-Reg and LL-Reg, we adopt the same discretization method as in (Yu & Shi 2003) since it shows better empirical results.

We set the number of clusters equal to the true number of classes c for all the clustering algorithms. To evaluate their performances, we compare the clusters generated by these algorithms with the true classes by computing two evaluation metrics: Clustering Accuracy (Acc)⁴ and Normalized Mutual Information (NMI)(Strehl & Ghosh 2002).

	Cornell	Washington	Wisconsin	Texas
MVLL-Reg	0.5362	0.5142	0.3385	0.5851
K-means	0.1872	0.1535	0.1166	0.2365
Lap-Reg	0.3644	0.3004	0.1593	0.4078
Nlap-Reg	0.4479	0.3797	0.2934	0.3463
LL-Reg	0.5272	0.3951	0.3722	0.5283

Table 4: Normalized Mutual Information (NMI) results on WebKB dataset

	Cornel	Washington	Wisconsin	Texas
MVLL-Reg	0.8065	0.8593	0.6094	0.7912
K-means	0.5538	0.5978	0.4851	0.5909
Lap-Reg	0.6868	0.7882	0.4628	0.7334
Nlap-Reg	0.7291	0.7487	0.5347	0.6327
LL-Reg	0.7793	0.8002	0.6083	0.7623

Table 5: Clustering Accuracy (Acc) results on WebKB dataset

⁴This performance measure discovers one-to-one relationship between clusters and true classes and measures the extent to which cluster contained examples from the corresponding category. It sums up the whole matching degree between all pair clusters. The greater clustering accuracy means, the better clustering performance. It can be evaluated as:

$$Acc = \frac{1}{n} \sum_{i=1}^n \delta(y_i, map(c_i))$$

where, $map(\cdot)$ is a function that maps each cluster index to a class label, which can be found by the Hungarian algorithm (Papadimitriou & Steiglitz 1998). c_i and y_i are the cluster index of x_i and the true class label. $\delta(a, b)$ is a function that equals 1 when a equals b , and 0 otherwise.

Clustering Results We report the clustering results on the WebKB data set in Table 4 and 5, from which we can see that, in most cases, MVLL-Reg outperforms the other clustering methods, which supports the assertion that the idea of local learning can be utilized to improve the performance of multi-view clustering.

Conclusions

In this paper, we put forward a novel multi-view local learning regularization matrix for semi-supervised learning and clustering. Unlike previous multi-view methods, our method employs the idea of local learning. Both the linear and kernel version of this regularization matrix are given. In the experiment part, we give some empirical experiments on both the WebKB and Cora dataset, which demonstrate the superior of the proposed method over several state-of-the-art ones. In the future, we will consider whether the idea of local learning can be employed in some other machine learning problems.

Acknowledgements

This work was supported by by NSFC (Grant No. 60721003, 60675009). We would like to thank Feiping Nie, Yangqiu Song for their help with this work. We would also thank the anonymous reviewers for their valuable comments.

References

- Argyriou, A.; Herbster, M.; and Pontil, M. 2005. Combining graph laplacians for semi-supervised learning. In *NIPS*.
- Blum, A., and Mitchell, T. 1998. Combining labeled and unlabeled data with co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers*, 92–100.
- Bottou, L., and Vapnik, V. 1992. Local learning algorithms. *Neural Computation* 4(6):888–900.
- Brefeld, U.; Gärtner, T.; Scheffer, T.; and Wrobel, S. 2006. Efficient co-regularised least squares regression. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, 137–144. New York, NY, USA: ACM Press.
- Chan, P. K.; Schlag, M. D. F.; and Zien, J. Y. 1994. Spectral k-way ratio-cut partitioning and clustering. *IEEE Trans. on CAD of Integrated Circuits and Systems* 13(9):1088–1096.
- Joachims, T.; Cristianini, N.; and Shawe-Taylor, J. 2001. Composite kernels for hypertext categorisation. In Brodley, C., and Danyluk, A., eds., *Proceedings of ICML-01, 18th International Conference on Machine Learning*, 250–257. San Francisco: Morgan Kaufmann Publishers.
- McCallum, A. K.; Nigam, K.; Rennie, J.; and Seymore, K. 2000. Automating the construction of internet portals with machine learning. *Information Retrieval* 3(2):127–163.
- Ng, A. Y.; Jordan, M. I.; and Weiss, Y. 2001. On spectral clustering: Analysis and an algorithm. In *NIPS*, 849–856.
- Papadimitriou, C. H., and Steiglitz, K. 1998. *Combinatorial Optimization : Algorithms and Complexity*. Dover Publications.
- Rosenberg, D. S., and Bartlett, P. L. 2007. The rademacher complexity of co-regularized kernel classes. In *AISTATS*.
- Scholkopf, B., and Smola, A. 2002. *Learning with Kernels. Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press.
- Sindhwani, V.; Niyogi, P.; and Belkin, M. 2005. A co-regularization approach to semi-supervised learning with multiple views. In *Proceedings of the Workshop on Learning with Multiple Views, 22nd ICML*.
- Smola, A., and Kondor, R. 2003. Kernels and regularization on graphs. In *Proc. 16th Annual Conference on Learning Theory*.
- Strehl, A., and Ghosh, J. 2002. Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *Journal on Machine Learning Research (JMLR)* 3:583–617.
- Tsuda, K.; Shin, H.; and Schölkopf, B. 2005. Fast protein classification with multiple networks. *Bioinformatics* 21(2):59–65.
- Vapnik, V. N. 1999. *The Nature of Statistical Learning Theory (Information Science and Statistics)*. Springer.
- Wang, F., and Zhang, C. 2006. Label propagation through linear neighborhoods. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, 985–992. New York, NY, USA: ACM Press.
- Wang, F.; Zhang, C.; and Li, T. 2007. Clustering with local and global regularization. In *AAAI*, 657–662.
- Wu, M., and Schölkopf, B. 2006. A local learning approach for clustering. In *Advances in Neural Information Processing Systems: NIPS 2006*, 1–8.
- Wu, M., and Schölkopf, B. 2007. Transductive classification via local learning regularization. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, 624–631.
- Wu, M.; Yu, K.; Yu, S.; and Schölkopf, B. 2007. Local learning projections. In *ICML*, 1039–1046.
- Yu, S. X., and Shi, J. 2003. Multiclass spectral clustering. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, 313. Washington, DC, USA: IEEE Computer Society.
- Zelnik-Manor, L., and Perona, P. 2004. Self-tuning spectral clustering. In *NIPS*.
- Zha, H.; He, X.; Ding, C. H. Q.; Gu, M.; and Simon, H. D. 2001. Spectral relaxation for k-means clustering. In *NIPS*, 1057–1064.
- Zhang, T.; Popescul, A.; and Dom, B. 2006. Linear prediction models with graph regularization for web-page categorization. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 821–826. New York, NY, USA: ACM Press.
- Zhou, D., and Burges, C. J. C. 2007. Spectral clustering and transductive learning with multiple views. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, 1159–1166. New York, NY, USA: ACM Press.
- Zhou, D.; Bousquet, O.; Lal, T.; Weston, J.; and Schölkopf, B. 2003. Learning with local and global consistency. In *In 18th Annual Conf. on Neural Information Processing Systems*.
- Zhu, S.; Yu, K.; Chi, Y.; and Gong, Y. 2007. Combining content and link for classification using matrix factorization. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 487–494. New York, NY, USA: ACM Press.
- Zhu, X.; Ghahramani, Z.; and Lafferty, J. D. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, 912–919.