Co-GCN for Multi-View Semi-Supervised Learning

Shu Li, Wen-Tao Li, Wei Wang*
National Key Laboratory for Novel Software Technology
Nanjing University, Nanjing 210023, China
{lis, liwt, wangw}@lamda.nju.edu.cn

Abstract

In many real-world applications, the data have several disjoint sets of features and each set is called as a view. Researchers have developed many multi-view learning methods in the past decade. In this paper, we bring Graph Convolutional Network (GCN) into multi-view learning and propose a novel multi-view semi-supervised learning method Co-GCN by adaptively exploiting the graph information from the multiple views with combined Laplacians. Experimental results on real-world data sets verify that Co-GCN can achieve better performance compared with state-of-the-art multi-view semi-supervised methods.

Introduction

Learning from labeled data is well-established in the machine learning community. However, providing labels to the data requires human labor, and is time-consuming and expensive. In many real-world applications, unlabeled data can often be obtained abundantly and cheaply, so there has been substantive interest in semi-supervised learning that exploits a large amount of unlabeled data together with labeled data to achieve better performance. In general, semi-supervised learning (Olivier, Schölkopf, and Alexander 2006; Zhu 2005) can be categorized into four classes: generative methods that use a generative model for the classifier and employ EM to model the label or parameter estimation process (Dempster, Laird, and Rubin 1977; Miller and Uyar 1996; Nigam et al. 2000); semi-supervised support vector machine methods (S^3VMs) that use unlabeled data to guide the decision boundary away from dense regions (Bennett and Demiriz 1999); graph-based methods that regularize the learning process by enforcing the label smoothness over the graph as a regularization term (Belkin, Matveeva, and Niyogi 2004; Zhou, Schölkopf, and Hofmann 2004; Zhu, Ghahramani, and Lafferty 2003); disagreement-based methods that train different learners and then let them label unlabeled data to boost the learning performance (Blum and Mitchell 1998; Goldman and Zhou 2000; Zhou and Li 2005a; 2005b). In many years, these semi-supervised learning methods were developed in parallel threads, then Wang and Zhou (2010) presented a theoretical graph-based explanation of co-training (a representative method of disagreement-based learning), and provided a possibility of bringing the graph-based and disagreement-based methods together.

Co-training (Blum and Mitchell 1998) is a representative method of disagreement-based learning. When co-training was proposed, it assumed that the data have two disjoint sets of features and each set is called as a view. In real-world applications, many data have more than one view. For example, the webpage classification data have two views, i.e., the text appearing on the page itself and the anchor text attached to the hyperlink pointing to this page (Blum and Mitchell 1998; Jing et al. 2017); the multi-media data have at least two views, i.e., image and text (Yan and Mikolajczyk 2015). Previous studies surveyed in Xu, Tao, and Xu (2013) have shown that multi-view learning methods could achieve better performance than traditional single-view learning methods.

In recent years, Deep Neural Networks (DNNs) have witnessed great successes in many applications. With the fast development of deep learning, some deep semi-supervised learning methods were developed. When the data have multiple views or multiple feature sets, deep neural network and co-training are brought together to improve the performance (Ardehaly and Culotta 2017; Cheng et al. 2016). These methods utilized the two views (i.e., RGB and depth in Cheng et al. (2016); image and text in Ardehaly and Culotta (2017)) to learn two DNNs and let them label unlabeled data to augment the training set for each other to boost the performance. Then Chen et al. (2018) presented tri-net which extends to the case with more than two DNNs. However, these methods all try to iteratively augment the labeled data set, which is time-consuming for retraining and can be easily corrupted by the misleading pseudo labels.

Inspired by the theoretical graph-based explanation of co-training (Wang and Zhou 2010) which provides the possibility of bringing graph-based methods and co-training together, in this paper we simultaneously consider the spectral graph information, multiple views and the expressive power

*Corresponding author.

of DNN, and develop a novel multi-view semi-supervised learning method Co-GCN, which unifies Graph Convolutional Network (GCN) and co-training into one framework. In Co-GCN, the spectral graph information among the multiple views are adaptively exploited with the combined Laplacians. We conduct experiments on several real-world data sets and the results verify that the proposed Co-GCN can achieve better performance compared with state-of-the-art multi-view semi-supervised methods.

The rest of this paper is organized as follows. After introducing some related works, we present our method and conduct the experiments. Finally, we make a conclusion.

Related Works

There are many existing multi-view and semi-supervised learning methods, we only briefly introduce the most related ones herein.

Multi-View Learning

Multi-view learning methods are generally categorized into three classes (Xu, Tao, and Xu 2013; Ye et al. 2015), i.e., fusion methods, subspace learning, and co-training.

The fusion methods include pre-fusion and post-fusion methods. Pre-fusion methods (e.g., multiple kernel learning) were originally developed to control the search space capacity of possible kernel matrices to get good generalization on data with single view. Lanckriet et al. (2004) formulated multiple kernel learning as a semi-definite programming problem and Bach, Lanckriet, and Jordan (2004) treated multiple kernel learning as a second-order cone problem. Multiple kernel learning is widely used in multi-view learning because multiple kernels naturally correspond to multiple views. Some other pre-fusion methods were also developed, e.g., MLAN (Nie, Cai, and Li 2017) adaptively learns an optimal graph for spectral clustering and semi-supervised classification. The post-fusion methods (Ye et al. 2012) mainly focus on how to aggregate different classifiers trained on each view. In recent years, hybrid-fusion methods (Ye et al. 2015) were proposed in a privacy-preserving way, which use rank consistency to communicate across views, and adaptively aggregate the classifiers on multiple views in the training process.

Subspace learning assumes that the features from different views are generated from a common latent subspace. Canonical Correlation Analysis (CCA) (Hotelling 1936) is the first subspace learning method, which explores basis vectors for the examples in the two views by mutually maximizing the correlation between the projections onto these basis vectors. Later, KCCA (Akaho 2006) incorporates kernel tricks for better generalization. The DNN-based methods such as DCCA (Andrew et al. 2013) and DCCAE (Wang et al. 2015) have also been developed to deal with large amounts of training data.

Co-training was proposed by Blum and Mitchell (1998). It learns two classifiers with initial labeled data on the two views respectively and lets them label unlabeled data for each other to augment the training data. Several successful variants of co-training have been proposed by using two different learning algorithms instead of two views, e.g., Zhou and Li (2005a) using two regressors with different parameter configurations. Some other variants also considered co-training from the view of Bayesian, e.g., Krishnapuram et al. (2004)) used the graph information as prior, and Yu et al. (2011) used a co-training kernel for gaussian process classifier. There is another method which is similar to co-training for multi-view learning, i.e., co-regularization (Sindhwani, Niyogi, and Belkin 2005). It optimizes the empirical loss on labeled data and the disagreement of two views over unlabeled data. Co-LapSVM and Co-LapRLS were proposed by using the classical regularization framework in Reproducing Kernel Hilbert Spaces (Sindhwani and Rosenberg 2008). Some recent works (Wang, Bian, and Tao 2013; Wu et al. 2019) also considered the disagreement among the multiple views for multi-view semi-supervised feature learning.

Spectral Graph Convolutional Networks

Spectral graph convolutional neural networks were introduced in Bruna et al. (2014), they considered a possible generalization of CNNs for non-Euclidean data. Later, Defferrard, Bresson, and Vandergheynst (2016) extended this framework by considering the fast localized convolutions. Graph Convolutional Network (GCN) (Kipf and Welling 2017) is a spectral convolution method that restricts the spectral filters to operate in 1-step neighborhood around each node, which can improve scalability and classification performance in large network. FastGCN (Chen, Ma, and Xiao 2018) and GraphSAGE (Hamilton, Ying, and Leskovec 2017) are two variants of GCN, which aim to improve the propagation rule by using novel sampling strategy in the process of feature aggregation. Graph attention network (Velickovic et al. 2018) was proposed by adding masked self-attentional layers to attend over the neighborhoods' features. An analysis of GCN (Li, Han, and Wu 2018) brought deeper insight and addressed that GCN is actually a special form of Laplacian smoothing and the vertices in the same cluster tend to be densely connected.

Preliminaries

In the multi-view setting, the examples are described with several disjoint sets of features. For simplicity, we first consider the two-view setting and then discuss how to extend it to the setting with more than two views. Suppose the data have two views, we can denote the data matrices of the two views as $\mathbf{X}_1$ and $\mathbf{X}_2$, respectively. For the v-th view ($v \in \{1, 2\}$), we have $\mathbf{X}_v \in \mathbb{R}^{n \times d_v}$, where n is the number of data and d_v is the feature dimension. We let l denote the number of labeled data and let u denote the number of unlabeled data (i.e., $n = l + u$). For the labeled data, let $\mathbf{Y}_l = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_l]^T$ be the ground-truth label, where $\mathbf{y}_i \in \mathbb{R}^{C \times 1}$ is one-hot indicator vector for the i-th example and C is the number of classes. For the j-th element of $\mathbf{y}_i$, $y_{ij} = 1$ means the i-th example belongs to the j-th class.

We now briefly introduce the architecture of Graph Convolutional Network (GCN) (Kipf and Welling 2017). Let $\mathbf{A}$ denote the adjacency matrix of graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The graph Laplacian can be denoted as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where

$\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$ and $\mathbf{A}_{ij}$ represents the (i,j)-th element in $\mathbf{A}$. Motivated by a first-order approximation of localized spectral filter on graph, the propagation rule of GCN layer is designed as

$$\mathbf{H}^{(k+1)} = \sigma(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(k)}\mathbf{W}^{(k)}), \tag{1}$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_n$ is the adjacency matrix of the undirected graph $\mathcal{G}$ with added self-connections, $\mathbf{I}_n$ is the identity matrix, and $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$. $\mathbf{W}^{(k)}$ is a layer-specific trainable weight matrix, and $\sigma(\cdot)$ denotes the activation function. $\mathbf{H}^{(0)} = \mathbf{X}$ is the feature matrix, and the loss of GCN is evaluated with the cross-entropy error over all labeled examples

$$\mathcal{L} = -\sum_{i=1}^{l}\sum_{j=1}^{C} y_{ij} \ln \mathbf{H}_{ij}^{(K)}, \tag{2}$$

where K is the number of layers in GCN.

Our Method

In order to deal with data that have more than one view, a simple intuition is to apply the post-fusion technique to GCN trained on each view. With co-training (Blum and Mitchell 1998), another basic idea is training a GCN in each view and letting them label the unlabeled data to augment the training set iteratively. However, this iterative method has two problems: (1) the process of re-training is time-consuming and therefore is not suitable for deep models; (2) the misleading pseudo-labels could corrupt the learners severely.

With the theoretical graph-based explanation of co-training (Wang and Zhou 2010), it is revealed that co-training can be represented as the label propagation process on a combinative graph, where the number of connected components is reduced. Inspired by this, we could construct a combinative graph

$$\mathbf{A}_c = \max(\mathbf{A}_1, \mathbf{A}_2)$$

from the multiple views and then run GCN with this combinative graph. Here $\mathbf{A}_v$ is the adjacency matrix in the v-th view, $\max(\cdot)$ is an element-wise maximum function, and $v \in \{1,2\}$. The combinative graph $\mathbf{A}_c$ can preserve the adjacency relationships among the examples in the two views. Thus, we can apply GCN with the combinative graph $\mathbf{A}_c$ to utilize the structural information in each view. Let $G_v(\mathbf{X}_v, \mathbf{A}_c)$ denote the GCN model trained with the combinative graph $\mathbf{A}_c$ in the v-th view, the two GCN models can be aggregated as follows

$$\tilde{\mathbf{H}} = \max(\mathbf{H}_1^{(K_1)}, \mathbf{H}_2^{(K_2)}), \tag{3}$$

where K_v is the number of layer in $G_v(\mathbf{X}_v, \mathbf{A}_c)$ and the output layer uses the softmax activation function. Then the label can be inferred with

$$c = \underset{j\in\{1,2,\ldots,C\}}{\operatorname{argmax}} \tilde{\mathbf{H}}_{ij}. \tag{4}$$

Co-GCN with Combined Laplacians

The method discussed above that runs GCN with the combinative graph $\mathbf{A}_c$ in each view and then aggregates the two GCNs to infer the label attempts to exploit the structural information from the multiple views. However, the multiple views have different sets of features, this method which runs GCN with the same combinative graph $\mathbf{A}_c$ for all views may be not a good solution for multi-view learning. A better idea is that we should construct the specific structural information for each view. In order to tackle this, inspired by some multiple graph methods (Argyriou, Herbster, and Pontil 2005; Wang et al. 2009), we give trainable weights to the Laplacians from each view. We suppose that a good graph for the v-th view ($v \in \{1,2\}$) is a weighted combination of multiple graph Laplacian matrices in the form of

$$\sum_{w=1}^{V} \pi_{vw}\mathbf{L}_w, \tag{5}$$

where $\sum_{w=1}^{V} \pi_{vw} = 1$ as that in Wang et al. (2009). Here, $\mathbf{L}_w$ is the graph Laplacian matrix in the w-th view, π_{vw} is a weight parameter, V is the number of views, and $w \in \{1,2\}$. If $\mathbf{L}_w$ could provide diverse information for the v-th view, the weight parameter π_{vw} should be large in order to exploit the complementary information from $\mathbf{L}_w$; if $\mathbf{L}_w$ only provides similar information for the v-th view, the weight parameter π_{vw} should be small. In original GCN (Kipf and Welling 2017), it uses a first-order approximation of Chebyshev polynomials

$$g_\theta \star x \approx \theta_0 x + \theta_1(\mathbf{L} - \mathbf{I}_n)x \tag{6}$$

with two free parameters θ_0 and θ_1. To derive the formulation of our co-training style GCN, we first replace the Laplacian $\mathbf{L}$ with the weighted combination of Laplacians for the v-th view, i.e., $\pi_{v1}\mathbf{L}_1 + \pi_{v2}\mathbf{L}_2$. We constrain the number of parameters by assuming $\theta = \theta_0 = -\theta_1$ as that in original GCN (Kipf and Welling 2017) and obtain

$$g_\theta \star x \approx \theta(\mathbf{I}_n + \pi_{v1}\mathbf{D}_1^{-\frac{1}{2}}\mathbf{A}_1\mathbf{D}_1^{-\frac{1}{2}} + \pi_{v2}\mathbf{D}_2^{-\frac{1}{2}}\mathbf{A}_2\mathbf{D}_2^{-\frac{1}{2}})x$$

in the v-th view. By using the renormalization trick

$$\mathbf{I}_n + \mathbf{D}_v^{-\frac{1}{2}}\mathbf{A}_v\mathbf{D}_v^{-\frac{1}{2}} \rightarrow \tilde{\mathbf{D}}_v^{-\frac{1}{2}}\tilde{\mathbf{A}}_v\tilde{\mathbf{D}}_v^{-\frac{1}{2}}$$

and defining $\tilde{\mathbf{A}}_v = \mathbf{A}_v + \mathbf{I}_n$ and $\tilde{\mathbf{D}}_{v(ii)} = \sum_j \tilde{\mathbf{A}}_{v(ij)}$, we have the propagation rule for GCN layer in the v-th view:

$$\begin{aligned} \mathbf{M}_v^{(k)} &= \tilde{\mathbf{L}}_v^{(k)}\mathbf{H}_v^{(k)}\mathbf{W}_v^{(k)}, \\ \mathbf{H}_v^{(k+1)} &= \sigma(\mathbf{M}_v^{(k)}), \end{aligned} \tag{7}$$

where $\sigma(\cdot)$ is the activation function, $\mathbf{H}_v^{(k)}$ is the v-th view's representation in the k-th layer and specifically $\mathbf{H}_v^{(0)} = \mathbf{X}_v$. For $\tilde{\mathbf{L}}_v^{(k)}$, it is formulated as

$$\tilde{\mathbf{L}}_v^{(k)} = \pi_{v1}^{(k)}\tilde{\mathbf{D}}_1^{-\frac{1}{2}}\tilde{\mathbf{A}}_1\tilde{\mathbf{D}}_1^{-\frac{1}{2}} + \pi_{v2}^{(k)}\tilde{\mathbf{D}}_2^{-\frac{1}{2}}\tilde{\mathbf{A}}_2\tilde{\mathbf{D}}_2^{-\frac{1}{2}}, \tag{8}$$

where $\pi_{vw}^{(k)}$ ($w \in \{1,2\}$) is the trainable parameter representing the weight of each Laplacian in the k-th layer for the v-th view ($v \in \{1,2\}$). With the propagation rule above, the model in each view can learn with the dynamically weighted Laplacians, and a combination of these models is used to make predictions. We denote this method as Co-GCN and it

Algorithm 1 Co-GCN

Input:
$\mathbf{X}_1$, $\mathbf{X}_2$, $\mathbf{Y}$, and adjacency matrices $\mathbf{A}_1$ and $\mathbf{A}_2$.
Parameter:
T_1, T_2, α_π and $\alpha_{\mathbf{W}}$.

```
for v = 1 to 2 do
    for t = 1 to T_v do
        Calculate the gradient of G_v with the loss function in Equation (2);
        Fix {π_vw^(k)}, update {W_v^(k)} in G_v with the learning rate α_W;
        Calculate the gradient of G_v with the loss function in Equation (2);
        Fix {W_v^(k)}, update {π_vw^(k)} in G_v according to Equation (11) with the learning rate α_π.
    end for
end for
```

Output:
Aggregation of G_1 and G_2 according to Equation (3).

can be optimized in an alternating way to learn the adaptive weights.

Optimization. Different from GCN (Kipf and Welling 2017), we have an extra set of weights $\{\pi_{vw}^{(k)}\}$ $(v, w \in \{1, 2\})$ to update in Co-GCN. In the optimization process, we alternately optimize the two sets of parameters $\{\pi_{vw}^{(k)}\}$ and $\{\mathbf{W}_v^{(k)}\}$. The parameter $\pi_{vw}^{(k)}$ could be updated separately according to its gradient. Thus, the optimization process can be illustrated as the following two steps:

Update $\{\mathbf{W}_v^{(k)}\}$: Fix $\{\pi_{vw}^{(k)}\}$, then update $\{\mathbf{W}_v^{(k)}\}$ by using the optimization method like gradient descent;

Update $\{\pi_{vw}^{(k)}\}$: Fix $\{\mathbf{W}_v^{(k)}\}$, then calculate the gradient of $\{\pi_{vw}^{(k)}\}$:

$$\frac{\partial \mathcal{L}}{\partial \pi_{vw}^{(k)}} = \mathrm{Tr}\left[(\frac{\partial \mathcal{L}}{\partial \mathbf{M}_v^{(k)}})^\top \frac{\partial \mathbf{M}_v^{(k)}}{\partial \pi_{vw}^{(k)}} \right] \tag{9}$$

here $\mathrm{Tr}(\cdot)$ is the matrix trace operator and

$$\frac{\partial \mathbf{M}_v^{(k)}}{\partial \pi_{vw}^{(k)}} = \tilde{\mathbf{D}}_w^{-\frac{1}{2}} \tilde{\mathbf{A}}_w \tilde{\mathbf{D}}_w^{-\frac{1}{2}} \mathbf{H}_v^{(k)} \mathbf{W}_v^{(k)}, \tag{10}$$

Finally, we update $\pi_{vw}^{(k)}$ according to the gradient

$$\pi_{vw}^{(k)} \leftarrow \pi_{vw}^{(k)} - \alpha_\pi \frac{\partial \mathcal{L}}{\partial \pi_{vw}^{(k)}}, \tag{11}$$

where α_π is the learning rate when updating $\{\pi_{vw}^{(k)}\}$. After updating $\{\pi_{vw}^{(k)}\}$, we apply softmax normalization in each layer to meet the constraint $\pi_{v1}^{(k)} + \pi_{v2}^{(k)} = 1$. According to Equation (10), the computational complexity of updating $\{\pi_{vw}^{(k)}\}$ is also linear in the number of graph edges. We summarize the process of our Co-GCN in Algorithm 1.

Extension to More-than-Two-View Setting

In some real-world applications, the data may have more than two views. We denote the data with V views ($V \geq 3$) as $\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_V$. In this section, we discuss how to generalize Co-GCN to the setting with more than two views. The Equation (8) can be extended as follows:

$$\tilde{\mathbf{L}}_v^{(k)} = \sum_{w=1}^{V} \pi_{vw}^{(k)} \tilde{\mathbf{D}}_w^{-\frac{1}{2}} \tilde{\mathbf{A}}_w \tilde{\mathbf{D}}_w^{-\frac{1}{2}}, \tag{12}$$

where $\pi_{vw}^{(k)}$ is the trainable parameter and can be updated according to Equation (11) separately. To satisfy the constraint $\sum_{w=1}^{V} \pi_{vw}^{(k)} = 1$, the softmax normalization is applied to each hidden layer at the end of each epoch as

$$\pi_{vw}^{(k)} \leftarrow \frac{\exp(\pi_{vw}^{(k)})}{\sum_{w=1}^{V} \exp(\pi_{vw}^{(k)})}, \tag{13}$$

for $\pi_{vw}^{(k)}$ $(v, w \in \{1, 2, ..., V\})$.

Experiments

In this section, we evaluate the performance of the proposed method on several real-world data sets.

Data Sets

In the experiments, we use three two-view data sets (i.e., Course, Cora, and Citeseer) and five more-than-two-view data sets (i.e., Ads, HW, Reuters, Cal7 and Cal20). Table 1 briefly summarizes the statistics of these data sets.

Table 1: Statistics of data sets. n is the number of examples, C is the number of classes, V is the number of views, and d_v is the number of features of each view.

data set	n	C	V	$d_v(v = 1, 2, ..., V)$
Course	1051	2	2	3447, 427
Cora	2708	7	2	1433, 2708
Citeseer	3264	6	2	3703, 3264
Ads	983	2	5	457, 495, 472, 111, 19
Reuters	1200	6	5	2000, 2000, 2000, 2000, 2000
HW	2000	10	6	240, 76, 216, 47, 64, 6
Cal7	1474	7	6	48, 40, 254, 1984, 512, 928
Cal20	2386	20	6	48, 40, 254, 1984, 512, 928

- The Course data set contains 1,051 pages collected from web sites of Computer Science departments of several universities, and has two views. These pages are manually labeled as course or non-course, each with a *fulltext* view and a *inlinks* view.
- The Cora data set contains 2708 documents over 7 labels, where the documents are described by 1433 words in the *content* view, and by the links between them in the *cites* view.
- The Citeseer data set has the same structure as Cora and contains 3312 documents over 6 labels. Following the same strategy as Cora, the *content* view and the *cites* view are used in experiments.
- The Advertise data set contains 983 images and has 5 views, i.e., *caption*, *alt* features in html description together with *base url*, *destination url* and *image url*. Each

Table 2: Accuracy (%) of the methods with γ ($\gamma = 1\%, 5\%, 10\%$) labeled data.

γ		Course	Cora	Citeseer	Ads	Reuters	HW	Cal7	Cal20
1%	CoTrade	80.72	32.25	22.74	-	-	-	-	-
	CoLap-SVM	78.45	39.29	44.20	-	-	-	-	-
	DCCAE	82.40	48.18	35.34	-	-	-	-	-
	MKL	81.85	42.25	43.71	86.04	40.23	81.27	86.24	73.01
	MLAN	78.23	22.84	25.04	86.10	22.15	**95.02**	85.69	65.96
	RANC	84.10	49.32	56.27	60.33	47.16	84.93	86.74	**77.45**
	GCN fusion	82.60	67.56	60.08	88.02	31.21	81.25	80.13	63.70
	GCN with $\mathbf{A}_c$	93.16	**73.69**	**62.94**	89.16	44.11	85.91	83.15	69.72
	Co-GCN	**93.59**	73.11	61.91	**91.45**	**48.41**	92.00	**87.69**	73.11
5%	CoTrade	80.51	48.86	35.45	-	-	-	-	-
	CoLap-SVM	79.96	61.06	62.47	-	-	-	-	-
	DCCAE	91.46	57.88	49.11	-	-	-	-	-
	MKL	91.07	61.50	61.10	91.04	57.98	95.25	92.53	81.79
	MLAN	78.33	47.12	49.83	86.15	27.15	**97.27**	89.54	77.64
	RANC	91.57	65.75	66.18	93.18	59.64	94.16	90.93	**84.16**
	GCN fusion	95.88	74.70	70.23	90.84	60.30	94.97	87.78	79.37
	GCN with $\mathbf{A}_c$	96.77	**78.93**	71.64	**94.41**	61.01	90.99	89.21	82.50
	Co-GCN	**97.77**	78.37	**71.95**	94.09	**62.28**	96.83	**93.62**	84.15
10%	CoTrade	81.88	58.94	47.84	-	-	-	-	-
	CoLap-SVM	85.64	69.21	67.13	-	-	-	-	-
	DCCAE	92.64	63.29	54.26	-	-	-	-	-
	MKL	93.86	71.58	67.20	92.44	65.10	96.67	94.87	86.09
	MLAN	78.67	57.91	63.76	86.33	51.24	97.59	91.90	81.05
	RANC	91.69	69.50	68.03	91.92	66.90	96.10	91.86	87.29
	GCN fusion	96.83	77.94	71.68	92.09	65.63	96.91	91.04	82.73
	GCN with $\mathbf{A}_c$	98.59	82.41	71.30	**95.60**	67.07	89.69	91.04	85.16
	Co-GCN	**99.06**	**82.59**	**71.79**	94.85	**67.21**	**97.65**	**94.96**	**87.59**

example describes an image on the web, and the images are manually labeled as ads or non-ads.

- The Reuters data set (Bisson and Grimal 2012) is constructed from the Reuters RCV1/RCV2 Multilingual test collection. Its multi-view information is created from different languages, i.e., *English*, *French*, *German*, *Italian* and *Spanish*.

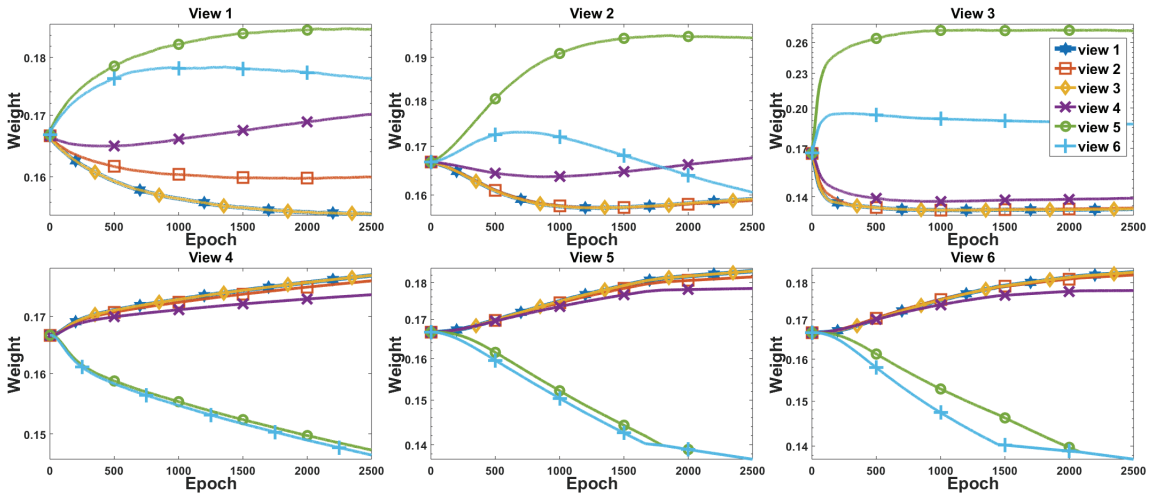
- The HW data set has 10 classes digits, each class has 200 different HW digits, and there are 2000 data points. The first view is the 216-D *profile-correlation* features, the second is the 76-D *Fourier-coefficient* features, the third is the 64-D *Karhunen-Loeve-coefficient* features, the fourth is the 240-D *intensity-averaged* features in 2×3 windows, the fifth is the 47-D *Zernike moment* features, and the sixth is the 6-D *morphological* features.

- Caltech-101 image data set consists of 101 categories of images for object recognition. We follow previous work (Li et al. 2015) and select the widely used 7 classes to get 1474 images, which we call Cal7. We also select a larger set named Cal20 which contains totally 2386 images of 20 classes. Five sets of features are extracted from all the images, i.e., 48 dimension *Gabor* features, 40 dimension *wavelet moments (WM)*, 254 dimension *CENTRIST* features, 1984 dimension *HOG* features, 512 dimension *GIST* features, and 928 dimension *LBP* features.

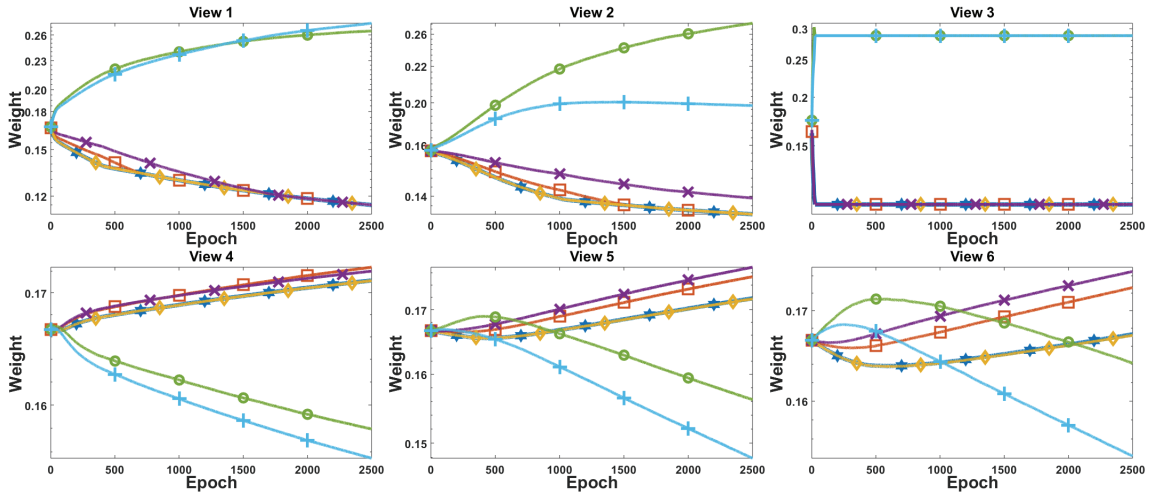
Setting

We randomly sample 10% data as the validation set, and then randomly sample γ ($\gamma = 1\%, 5\%, 10\%$) of the remaining data as the labeled data, and the remainder of the data are used as the unlabeled data. For each method, 10 trials are performed and the average accuracy is reported. The hyper-parameters are chosen according to the validation performance in the first trial, and then are fixed. We construct k-nearest-neighbor graph for each view with different distance metric $\exp(-\frac{d(x_{v(i)},x_{v(j)})}{\sigma^2})$ ($k \in \{1,3,5,7,9\}$, $d(x_{v(i)}, x_{v(j)})$ is Euclidean and cosine distance function, and $\sigma \in \{10^{-2}, 10^{-1}, 1\}$). In the experiments, we find that Co-GCN is relatively robust to the parameter k, since it not only utilizes the spectral information from the graph but also learns from the features. For each data set, we study the accuracy with different labeled ratios and different k under cosine distance and euclidean distance, and find that for all data sets the 1-NN graph performs worse than other graphs and the performances of 7-NN and 9-NN are comparable, which reminds us that we can choose a relatively large k in real-world applications.

In the experiments, we use GCN with two layers similar to the setting in Kipf and Welling (2017). We use dropout ($p = 0.3$) after each layer, use ReLU as the activation function in the hidden layer, and use softmax activation function in the output layer. The sizes of the hidden layer are varied according to the dimension of features in the view. If the view has more than 500 attributes, we use a hid-



(a) Weights in the hidden layer



(b) Weights in the output layer

Figure 1: Weights of Laplacians in Co-GCN. A uniform legend applied to (a) and (b) is given in view 3 of (a).

den layer with 256 units, otherwise 128 units. We train all models for a maximum of 2500 epochs (training iterations) using Adam (Kingma and Ba 2015) with a learning rate of $\alpha_{\mathbf{W}} = 10^{-3}$ and early stopping with a window size of 50, i.e., we stop the training process if the validation accuracy does not increase for 50 consecutive epochs. When updating $\pi_{vw}^{(k)}$, the learning rate α_π is set to be 10^{-2}. Parameters in $\left\{\pi_{vw}^{(l)}\right\}$ are first initialized with $1/V$ before training.

Baselines

We use the method that runs GCN with the combinative graph $\mathbf{A}_c = \max(\mathbf{A}_1, \ldots, \mathbf{A}_V)$ in each view and then aggregates them to make predictions according to Equation (3) as a baseline denoted as *GCN with* $\mathbf{A}_c$ in Table 2. We also compare our Co-GCN with other state-of-the-art multi-view learning methods. For co-training style methods, we use *CoLap-SVM* (Sindhwani, Niyogi, and Belkin 2005) and an improved version of co-training named *CoTrade* (Zhang and Zhou 2011). For subspace methods, we use *DCCAE* (Wang et al. 2015). For fusion methods, we use *MKL* (Bach, Lanckriet, and Jordan 2004), *MLAN* (Nie, Cai, and Li 2017) and *RANC* (Ye et al. 2015). When data have multiple views, each view can run GCN with its own graph and then aggregates these GCNs to make predictions according to Equation (3), we denote this method as *GCN fusion* in Table 2. Among these methods, some are limited to the two-view setting, i.e., *CoLap-SVM*, *CoTrade* and *DCCAE*.

We consider the linear, polynomial and RBF kernel during the training process for *CoLap-SVM*. Both encoder and decoder networks have one hidden layer in *DCCAE*, the number of units in the hidden layer is chosen from $\{2^5, 2^6, 2^7\}$, and the dimension of subspace after encoding is chosen from $\{10, 20, 30\}$. Hyper-parameter C is chosen from $\{10^{-2}, 10^{-1}, 1\}$ for *MKL*. We test several kernel types in *MKL*, such as linear kernel, polynomial kernel with the degree chosen from $\{2, 3, 4\}$ and RBF kernel with the parameter chosen from $\{10^{-1}, 10^0, 10^1\}$. For

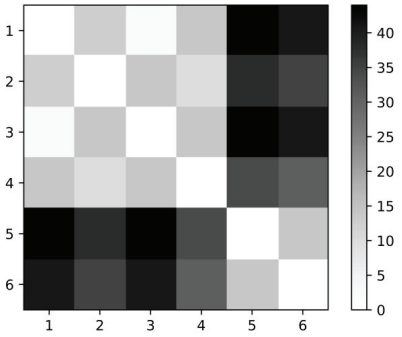


Figure 2: Color map of the pairwise distance matrix for Laplacians of six views.

RANC, two regularization parameters are selected from $\{10^{-4}, 10^{-3}, \ldots, 10^{3}, 10^{4}\}$, three different solutions, i.e., *APG*, *ADMM* and *accelerate*, are used for training. For all baselines, we also use the validation set to choose the best hyper-parameters for them.

Results

The results are described in Table 2 for different labeled data ratios with $\gamma \in \{1\%, 5\%, 10\%\}$. From Table 2, it can be found that Co-GCN and *GCN with* $\mathbf{A}_c$ perform much better than *GCN fusion*. Furthermore, it also shows that Co-GCN is better than *GCN with* $\mathbf{A}_c$ on most data sets while comparable with *GCN with* $\mathbf{A}_c$ on others, which verifies the usefulness of our adaptively weighted Laplacians.

Secondly, we compare Co-GCN with state-of-the-art multi-view learning methods. From Table 2 we can find that, in general, Co-GCN can achieve better performance than other methods under different sizes of labeled examples, which verifies the superiority of the proposed method. When the labeled ratio is relatively small, i.e., $\gamma = 1\%$, *MLAN* and *MKL* perform better on some data sets, i.e., HW, Cal20. The reason is that these two data sets consist of multiple classes (see Table 1), and the labeled data for each class is not enough to train good models. When the labeled ratio increases, i.e., $\gamma = 10\%$, Co-GCN outperforms almost all baselines.

Discussions

In Co-GCN, combined Laplacians are adopted to consider the information from different views, and the weights $\{\pi_{vw}^{(k)}\}$ are updated and normalized in each epoch. To illustrate why Co-GCN could perform better, we depict the weights of Laplacians on Cal20 in Figure 1. To validate this, we also draw a color map (see Figure 2) of the pairwise distance matrix of the six Laplacians, where the pairwise distance matrix $\mathbf{P}$ is defined as

$$\mathbf{P}_{ij} = \|\mathbf{D}_i^{-\frac{1}{2}}\mathbf{A}_i\mathbf{D}_i^{-\frac{1}{2}} - \mathbf{D}_j^{-\frac{1}{2}}\mathbf{A}_j\mathbf{D}_j^{-\frac{1}{2}}\|_F.$$

In Figure 2, the (i, j)-th block characterizes the difference between the i-th and the j-th Laplacians (the lighter the color is, the smaller the difference is). One can readily check that the Laplacians of the 1-st view and the 3-rd view are almost the same (the Frobenius norm of their difference is relatively small), which echoes with the indistinguishable two lines in Figure 1. Furthermore, one can observe from Figure 1 that the weights of views 5 and 6's Laplacians are rather large in Co-GCN networks of views 1, 2 and 3. One possible explanation is that the colors of the $(5, 1)$-th, $(5, 2)$-th, $(5, 3)$-th, $(6, 1)$-th, $(6, 2)$-th, and $(6, 3)$-th blocks are dark in Figure 2, which implies that views 1, 2, and 3 can provide complementary information for views 5 and 6.

Conclusion

In this paper, we propose Co-GCN for multi-view semi-supervised learning which unifies co-training, spectral graph information and the expressive power of neural network into one framework, in which we use combined Laplacian to exploit the graph information from the multiple views. The experimental results demonstrate that the proposed method is superior to state-of-the-art multi-view semi-supervised learning, and also empirically show that the Co-GCN network in each view can adaptively learn the spectral information from other complementary views.

Acknowledgements

The authors would like to thank Ching-Yun Ko for helpful discussions. This work is supported by the National Key R&D Program of China (2017YFB1002201), the National Science Foundation of China (61673202), and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

Akaho, S. 2006. A kernel method for canonical correlation analysis. *CoRR* abs/cs/0609071.

Andrew, G.; Arora, R.; Bilmes, J. A.; and Livescu, K. 2013. Deep canonical correlation analysis. In *ICML*.

Ardehaly, E. M., and Culotta, A. 2017. Co-training for demographic classification using deep learning from label proportions. In *ICDM Workshops*.

Argyriou, A.; Herbster, M.; and Pontil, M. 2005. Combining graph laplacians for semi-supervised learning. In *NIPS*.

Bach, F. R.; Lanckriet, G. R. G.; and Jordan, M. I. 2004. Multiple kernel learning, conic duality, and the SMO algorithm. In *ICML*.

Belkin, M.; Matveeva, I.; and Niyogi, P. 2004. Regularization and semi-supervised learning on large graphs. In *COLT*.

Bennett, K. P., and Demiriz, A. 1999. Semi-supervised support vector machines. In *NIPS*.

Bisson, G., and Grimal, C. 2012. Co-clustering of multi-view datasets: A parallelizable approach. In *ICDM*.

Blum, A., and Mitchell, T. M. 1998. Combining labeled and unlabeled data with co-training. In *COLT*.

Bruna, J.; Zaremba, W.; Szlam, A.; and LeCun, Y. 2014. Spectral networks and locally connected networks on graphs. In *ICLR*.

Chen, D.; Wang, W.; Gao, W.; and Zhou, Z. 2018. Tri-net for semi-supervised deep learning. In *IJCAI*.

Chen, J.; Ma, T.; and Xiao, C. 2018. Fastgcn: Fast learning with graph convolutional networks via importance sampling. In *ICLR*.

Cheng, Y.; Zhao, X.; Cai, R.; Li, Z.; Huang, K.; and Rui, Y. 2016. Semi-supervised multimodal deep learning for RGB-D object recognition. In *IJCAI*.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS*.

Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)* 39(1):1–22.

Goldman, S. A., and Zhou, Y. 2000. Enhancing supervised learning with unlabeled data. In *ICML*.

Hamilton, W. L.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. In *NIPS*.

Hotelling, H. 1936. Relations between two sets of variates. *Biometrika* 28(3/4):321–377.

Jing, X.; Wu, F.; Dong, X.; Shan, S.; and Chen, S. 2017. Semi-supervised multi-view correlation feature learning with application to webpage classification. In *AAAI*.

Kingma, D. P., and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Kipf, T. N., and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Krishnapuram, B.; Williams, D.; Xue, Y.; Hartemink, A. J.; Carin, L.; and Figueiredo, M. A. T. 2004. On semi-supervised classification. In *NIPS*.

Lanckriet, G. R. G.; Cristianini, N.; Bartlett, P. L.; Ghaoui, L. E.; and Jordan, M. I. 2004. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research* 5:27–72.

Li, Y.; Nie, F.; Huang, H.; and Huang, J. 2015. Large-scale multi-view spectral clustering via bipartite graph. In *AAAI*.

Li, Q.; Han, Z.; and Wu, X. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI*.

Miller, D. J., and Uyar, H. S. 1996. A mixture of experts classifier with learning based on both labelled and unlabelled data. In *NIPS*.

Nie, F.; Cai, G.; and Li, X. 2017. Multi-view clustering and semi-supervised classification with adaptive neighbours. In *AAAI*.

Nigam, K.; McCallum, A. K.; Thrun, S.; and Mitchell, T. 2000. Text classification from labeled and unlabeled documents using em. *Machine learning* 39(2-3):103–134.

Olivier, C.; Schölkopf, B.; and Alexander, Z. 2006. *Semi-Supervised Learning*. MIT Press.

Sindhwani, V., and Rosenberg, D. S. 2008. An RKHS for multi-view learning and manifold co-regularization. In *ICML*.

Sindhwani, V.; Niyogi, P.; and Belkin, M. 2005. A co-regularization approach to semi-supervised learning with multiple views. In *ICML workshops*.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph attention networks. In *ICLR*.

Wang, W., and Zhou, Z. 2010. A new analysis of co-training. In *ICML*.

Wang, M.; Hua, X.-S.; Hong, R.; Tang, J.; Qi, G.-J.; and Song, Y. 2009. Unified video annotation via multigraph learning. *IEEE Transactions on Circuits and Systems for Video Technology* 19(5):733–746.

Wang, W.; Arora, R.; Livescu, K.; and Bilmes, J. A. 2015. On deep multi-view representation learning. In *ICML*.

Wang, X.; Bian, W.; and Tao, D. 2013. Grassmannian regularized structured multi-view embedding for image classification. *IEEE Transaction on Image Processing* 22(7):2646–2660.

Wu, F.; Jing, X.; Zhou, J.; Ji, Y.; Lan, C.; Huang, Q.; and Wang, R. 2019. Semi-supervised multi-view individual and sharable feature learning for webpage classification. In *WWW*.

Xu, C.; Tao, D.; and Xu, C. 2013. A survey on multi-view learning. *CoRR* abs/1304.5634.

Yan, F., and Mikolajczyk, K. 2015. Deep correlation for matching images and text. In *CVPR*.

Ye, G.; Liu, D.; Jhuo, I.; and Chang, S. 2012. Robust late fusion with rank minimization. In *CVPR*.

Ye, H.; Zhan, D.; Miao, Y.; Jiang, Y.; and Zhou, Z. 2015. Rank consistency based multi-view learning: A privacy-preserving approach. In *CIKM*.

Yu, S.; Krishnapuram, B.; Rosales, R.; and Rao, R. B. 2011. Bayesian co-training. *Journal of Machine Learning Research* 12:2649–2680.

Zhang, M., and Zhou, Z. 2011. Cotrade: Confident co-training with data editing. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 41(6):1612–1626.

Zhou, Z., and Li, M. 2005a. Semi-supervised regression with co-training. In *IJCAI*.

Zhou, Z.-H., and Li, M. 2005b. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering* 17(11):1529–1541.

Zhou, D.; Schölkopf, B.; and Hofmann, T. 2004. Semi-supervised learning on directed graphs. In *NIPS*.

Zhu, X.; Ghahramani, Z.; and Lafferty, J. D. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*.

Zhu, X. 2005. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences.