Knowledge-Based Systems 195 (2020) 105698

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Multi-view semi-supervised learning for classification on dynamic networks☆

Chuan Chen [a,b], Yuzheng Li [a], Hui Qian [a], Zibin Zheng [a,b,*], Yanqing Hu [a]

[a] School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China
[b] National Engineering Research Center of Digital Life, Sun Yat-sen University, Guangzhou, China

ARTICLE INFO

Article history:
Received 17 October 2019
Received in revised form 20 February 2020
Accepted 21 February 2020
Available online 27 February 2020

Keywords:
Semi-supervised learning
Multi-view learning
Dynamic networks
Total variation

ABSTRACT

In recent decades, the task of graph-based multi-view learning has become a fundamental research problem, which could integrate data from multiple sources to improve performance. The dynamic networks could be treated as one kind of multi-view network, but it is continually evolving and leads to entirely different observations at multiple epochs. In this paper, we treat these observations as multiple views and seek a semi-supervised multi-view approach to address the classification problem. Therefore, we propose Multi-view Semi-supervised learning for Classification on Dynamic networks (*MSCD*). With the aid of total variation regularization, *MSCD* can obtain a sparse and smooth combination of the views and a better classification result. From the theoretical point of view, the *MSCD* model is decomposed into simpler sub-problems, which can be effectively solved under the Alternating Direction Method of Multipliers (ADMM) framework. Extensive experiments on both synthetic and real-world datasets show that our model can outperform the state-of-the-art approaches.

1. Introduction

Graph-based learning is a fundamental research field of machine learning, which leverages structural information to improve the performance of learning tasks. In graph-based learning, the instances are taken as nodes of a graph, and the edges are generally calculated as the similarity among instances. For example, in social networks, users are represented as nodes, and relationships among users such as shared interests and common friends are represented as edges. One of the advantages of graph-based learning is that structural information could be used to boost the learning process. To utilize this, Semi-Supervised Learning (SSL) [1–5] has been highly developed in graph-based learning, which integrates labeled and unlabeled data together to achieve some learning tasks, such as classification and community detection. The key idea of SSL is so-called *manifold assumption*, in which nodes connected by edges with a large weight on the graph are required to have similar labels. One of the fundamental research of graph-based learning is spectral clustering [6], which addressed graph cut problems and introduced corresponding vector–matrix form objectives. Inspired by spectral clustering, many improvements [7–9] have been developed in various disciplines [10–12], due to their flexibility, easiness of implementation and excellent efficiency regarding both computational storage and cost.

To reduce the effect of misleading information, it is meaningful to integrate data from multiple sources, which is called views as well. For example, the relationship of social networks' users could be constructed based on different social platforms, and properly integrating them could improve the performance of classification tasks among users. In other words, the data from multiple sources can be integrated to identify groups of objects in a more reliable manner. To achieve this goal, multi-view learning [13] was incorporated into graph-based SSL. By treating input networks as views, the multi-view graph-based SSL seeks to combine all the views, to improve the classification performance. According to the significance of each view, we can differentiate the critical or irrelevant views, which is called view selection. During this combination, the algorithm could discard noisy data and adequately combine the partial information of different views to complement each other. Many practical applications [14–18] have proved that multi-view learning could effectively improve the performance of graph-based unsupervised and SSL tasks.

In a dynamic scenario, the internal relationships of networks are continually evolving. For example, interactions happen all the time on social platforms, and the properties of affected users might change as well. By observing one dynamic network at different epochs, we would obtain a series of network snapshots, which are so-called time-varying networks (Fig. 1). Moreover,

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.knosys.2020.105698.

* Correspondence to: Room D203, South laboratory building, Sun Yat-sen University, Guangzhou, Guangdong, 511400, China.
E-mail address: zhzibin@mail.sysu.edu.cn (Z. Zheng).

https://doi.org/10.1016/j.knosys.2020.105698
0950-7051/

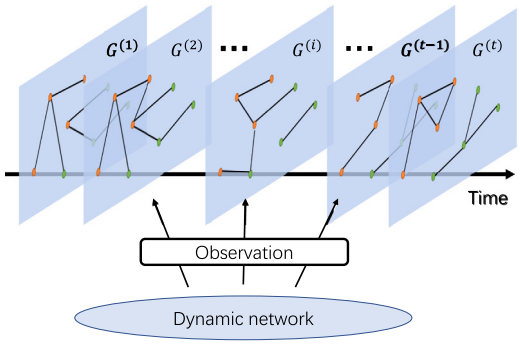


Fig. 1. Time-varying networks observed from an underlying dynamic network.

we can obtain time-varying networks from any evolving system. Some of them evolve rapidly (like computer networks), and the observation at any epoch may involve random noise and misleading information, which implies that it is crucial to obtain a complementary combination of multiple observations. By treating each observation as one view, this problem becomes a multi-view learning problem. However, the existing multi-view approaches neglect the smoothness of the dynamic networks [14,15], i.e., the views corresponding to consecutive epochs can be interpreted to interact actively and consistently. On the other hand, the label rate is an essential prerequisite for semi-supervised learning. In extreme cases with few labeled training data, many existing approaches may degrade or fail to classify nodes.

Intuitively, we propose a transductive graph-based SSL approach to address classification on dynamic networks, i.e., Multi-view learning Classification for Dynamic Networks (*MSCD*). To classify the instances of a dynamic network, given a series of observations (views) sorted in temporal order, *MSCD* could find an optimal combination of views to achieve the task. To utilize the temporal information, we introduce total variation regularization [19,20], which encourages the weight of views to be locally consistent. This constraint fits well with our assumption that information gaps between any adjacent epochs are generally small. Noteworthy, with the aid of total variation regularization, this approach could effectively handle the scenarios with a limited amount of labels.

In the optimization part, *MSCD* is theoretically decomposed into two sub-problems, i.e., one network-weighting sub-problem and one node label indicator sub-problem, both of which can be analytically solved. The label indicator can be obtained by solving a linear sub-problem, and the network weights can be determined by aggregating the contribution of each network in the node label indicator sub-problem, both of which can be efficiently solved under alternating direction method of multipliers (ADMM) framework [21,22]. We should note that even though the network weights are required to be locally smooth under total variation regularization, it can also be globally sparse according to their degrees of contribution in the transfer of dynamic network knowledge.

The rest of this paper is organized as follows. In Section 2, we provide a brief review of the related work. In Section 3, we give some symbol definitions and present the proposed model and algorithm for *MSCD*. In Section 4, experimental results for synthetic and real-world datasets are given to demonstrate the superior performance of the proposed method to the other conventional methods. Finally, some concluding remarks are given in Section 5.

2. Related works

In this section, we will review several related works, and point out the main difference between our work compared to the related ones.

In multi-view learning, many methods assume that the information collected in different views are for the same set of instances, and all views share one label prediction. In the unsupervised setting, Kumar et al. [23] proposed to use the idea of co-regularized for multi-view spectral clustering and proposed the idea of co-training in [8]. These works aimed to optimize the clustering structure on each view, which can be naturally combined with the label indicator of spectral clustering. Lihi et al. [9] proposed a self-tuning spectral clustering method, which can infer the number of groups automatically. Zhao et al. [24] introduced a clustering algorithm based on matrix factorization. Tao et al. [25] employed the low-rank sparse decomposition to consider the similarity between different views explicitly and detected the noises in each view at the same time. Auto-weighted is also an effective method to solve multi-view clustering problems by allocating ideal weight for each view automatically [26–28]. Wang et al. [17] explored the impact of different graph metrics on the multi-view clustering performance. Yi et al. [18] tackled the multi-task multi-view clustering problems in heterogeneous situations. They proposed an approach to transform the sample space onto multi-view space, then on multi-task space for clustering. Zhou et al. [29] modeled the dynamic community detection tasks [30,31] as multi-objective problems, and proposed a discrete bat algorithm to capture the structural information of graphs. Graph data is often unbalanced, so in recent studies, there are methods [32] focusing on data enhancement and generation.

In the semi-supervised [1] setting, Muslea et al. [33] further combined active learning in co-training progress and proposed robust semi-supervised learning. Zhu et al. proposed a graph-based semi-supervised learning [34], which is fundamental research about the label propagation algorithm. Chen et al. [16] considered that the sets of instances in views might be different, and proposed a multi-domain semi-supervised classification approach to address this situation. Muslea et al. [35] combined active learning and semi-supervised classification, proposed a robust multi-view learning approach. Besides, Karasuyama et al. [14] proposed a sparse multi-view learning approach, and Wang et al. [15] combined multiple views in a unified model to address video annotation problem. Xiao et al. [36] utilized the graph inference to seek the relationship between miRNA data and diseases. Ibrahim et al. [37] presented a method for link prediction in dynamic networks by integrating temporal information, community structure, and node centrality in the network. Lei et al. [38] built weighted dynamic PPI (protein–protein interaction) networks and predict the protein complex with a moth–flame optimization-based algorithm. Jing et al. [39] studied the attribute reduction problem on dynamic networks. Huang et al. [40] concentrated on information fusing of multi-source interval-valued data with the dynamic updating of data sources.

Compared with the related works, our work pays more attention to dynamic networks, and the nodes classification task on them. By utilizing the continuity of time slices, both views weighting and nodes classification are well studied in a unified model. In the next section, we will introduce the proposed approach in detail.

3. Method

3.1. Preliminary

We are given a limited number of labeled instances to classify the rest. The connectivity among n instances of tth view is represented by the affinity matrix $\mathbf{W}^{(t)}$, where (i, j)-element $W_{ij}^{(t)}$ is the similarity between instances x_i and x_j. We should note that proper similarity measurements such as *Gaussian kernel* and *vector inner*

product are all accepted to calculate the affinity matrix, and the selection depends more on the characteristic of the dataset, so we omit the discussion of measurements in our model. The known labels are stored in an $n \times c$ membership matrix $\mathbf{Y}^{(t)}$, where c is number of classes. Without loss of generality, we suppose that the first ℓ instances are labeled, therefore, $Y_{ij}^{(t)} = 1$ if $i \leq \ell$ and x_i belongs to class j. Otherwise, $Y_{ij}^{(t)} = 0$. In our problem settings, the label rate should be extremely low, that is $\ell \ll n$.

Firstly, we consider the single-view SSL classification model. The label propagation algorithm [41] estimates the labels based on the smoothness assumption on the networks, which assumes that instances share a label if they are close to each other. Concretely, on binary classification problem, the algorithm estimates the label indicators $\{f_i\}_{i=1}^n$ on tth view by solving the following regularized least squares problem:

$$\min_{\mathbf{f}} \sum_{i,j}^{n} W_{ij}^{(t)} \left(\frac{f_i}{\sqrt{D_{ii}^{(t)}}} - \frac{f_j}{\sqrt{D_{jj}^{(t)}}}\right)^2 + \alpha \sum_{i=1}^{n} (y_i - f_i)^2, \tag{1}$$

where $(D)_{ii}^{(t)} = \sum_{j=1}^{n} W_{ij}^{(t)}$, and α is a trade-off parameter. This objective constrains the smoothness of label predictions on the graph, and leverages the semi-supervised knowledge. Under the multiple classes settings, we rewrite the above problem as:

$$\min_{\mathbf{F}} \mathrm{Tr}(\mathbf{F}^{\intercal}\mathbf{L}^{(t)}\mathbf{F}) + \alpha \|\mathbf{F} - \mathbf{Y}\|_{\mathrm{fro}}^2, \tag{2}$$

where $\mathbf{F} \in \mathbb{R}^{n \times c}$ is the label indicator, and the normalized Laplacian matrix $\mathbf{L}$ is defined by $\mathbf{L} = \mathbf{I} - (\mathbf{D}^{(t)})^{-1/2}\mathbf{W}^{(t)}(\mathbf{D}^{(t)})^{-1/2}$. Based on this symbols definition, we raise our model in next sub-section.

3.2. Model statement

In many real-life scenarios, we are confronted with dynamic networks whose relationship among nodes keep constantly evolving. The existing multi-view classification algorithms cannot capture the evolving information well since the smoothness of time-varying data is neglected. Therefore, we consider the continuity of time, innovatively combine T dynamic networks with smooth weights in a unified model:

$$\begin{aligned} \min_{\mathbf{F},\mathbf{w}} \ \mu \,\mathrm{Tr}(\mathbf{F}^{\intercal}\mathbf{L}^{(t')}\mathbf{F}) + \sum_{t=1,t\neq t'}^{T} w^{(t)} \mathrm{Tr}(\mathbf{F}^{\intercal}\mathbf{L}^{(t)}\mathbf{F}) \\ + \alpha \|\mathbf{F} - \mathbf{Y}\|_{\mathrm{fro}}^2 + \beta \|\mathbf{D}\mathbf{w}\|_1, \\ s.t.\ \mathbf{w}^{\intercal}\mathbf{1} = 1, \mathbf{w} \geq 0, \end{aligned} \tag{3}$$

where μ, α and β are trade-off parameters, and we highlight the *target view* (i.e. the t'th view) with the first term. The model is expected to automatically identify those views relevant to the target view by assigning them higher weight $w^{(t)}$ than other irrelevant views. In real-life applications, we generally set the latest view as target. To leverage the rest views as auxiliary, we are inspired by label propagation and linearly combine their local smooth graph structures in the second term. The weight of this combination is defined by $\mathbf{w} \in \mathbb{R}^{T-1}$. As aforementioned, for the sake of data consecutiveness, total variation regularization is introduced as the last term, in which the time dimension factor is constrained to be smooth and sparse, and therefore gaining a better generalization of the model. In details, $\mathbf{D} \in \mathbb{R}^{(T-2)\times(T-1)}$ is a first-order differential matrix where $\mathbf{D}_{i,i} = 1$ and $\mathbf{D}_{i,i+1} = -1$ for $i = 1, 2, \ldots, T-2$ and other entries are zeros:

$$\|\mathbf{D}\mathbf{w}\|_1 = \sum_{i=1}^{T-2} |w_i - w_{i+1}|. \tag{4}$$

Algorithm 1 Iterative solution for label indicator

Input:
Laplacian matrix $\mathbf{L}'$, initial indicator $\mathbf{F}^0$, semi-supervised label $\mathbf{Y}$, parameter α.

Output:
Optimal label indicator $\mathbf{F}$.

1: Update $\mathbf{F}$ by:

$$\mathbf{F}^{t+1} = \frac{1}{1+\alpha}(\mathbf{I} - \mathbf{L}')\mathbf{F}^t + \frac{\alpha}{1+\alpha}\mathbf{Y};$$

2: Let $t = t + 1$, repeat step 1 until convergence.

We utilize the total variation regularization to constrain the time dimension factor to obtain a smooth but sparse weight and gain better generalization of our model. With a smooth combination of views, Eq. (3) keeps a reasonable *view-agreement* assumption by sharing one label indicator $\mathbf{F}$. This assumption held by much multi-view learning literature is that all the views should share a similar underlying clustering structure. We will optimize this model in the next sub-section.

3.3. Optimization

Recall that Eq. (3) is convex respecting to each variable. Therefore, an iterative optimization approach is needful to solve it. We choose an effective solving framework ADMM and introduce an auxiliary variable $\mathbf{z} = \mathbf{D}\mathbf{w}$, then we consider the optimization as follows:

$$\begin{aligned} \min_{\mathbf{F},\mathbf{w}} \ \mu \,\mathrm{Tr}(\mathbf{F}^{\intercal}\mathbf{L}^{(t')}\mathbf{F}) + \sum_{t=1,t\neq t'}^{T} w^{(t)} \mathrm{Tr}(\mathbf{F}^{\intercal}\mathbf{L}^{(t)}\mathbf{F}) \\ + \alpha \|\mathbf{F} - \mathbf{Y}\|_{\mathrm{fro}}^2 + \beta \|\mathbf{z}\|_1, \\ s.t.\ \mathbf{w}^{\intercal}\mathbf{1} = 1, \mathbf{w} \geq 0, \mathbf{z} = \mathbf{D}\mathbf{w}, \end{aligned} \tag{5}$$

whose solution for $(\mathbf{w}, \mathbf{F})$ coincides with the solution of Eq. (3). Though a new constraint is introduced together, it will be solved effective shortly. To achieve this, we construct the augmented Lagrangian function as follow:

$$\begin{aligned} \min_{\mathbf{F},\mathbf{w},\mathbf{z}} \ \mu \,\mathrm{Tr}(\mathbf{F}^{\intercal}\mathbf{L}^{(t')}\mathbf{F}) + \sum_{t=1,t\neq t'}^{T} w^{(t)} \mathrm{Tr}(\mathbf{F}^{\intercal}\mathbf{L}^{(t)}\mathbf{F}) \\ + \alpha \|\mathbf{F} - \mathbf{Y}\|_{\mathrm{fro}}^2 + \beta \|\mathbf{z}\|_1 \\ + \boldsymbol{\lambda}^{\intercal}(\mathbf{D}\mathbf{w} - \mathbf{z}) + \frac{\rho}{2}\|\mathbf{D}\mathbf{w} - \mathbf{z}\|_2^2, \\ s.t.\ \mathbf{w}^{\intercal}\mathbf{1} = 1, \mathbf{w} \geq 0, \end{aligned} \tag{6}$$

where $\boldsymbol{\lambda}$ is the Lagrangian multiplier and ρ is a hyper-parameter (i.e. the learning rate). To solve the problem given by Eq. (6), we alternately minimize the objective function with respect to $\mathbf{F}, \mathbf{w}$ and $\mathbf{z}$. Noteworthy, since each sub-problem is convex, the existence of global optimal in each iteration is guaranteed.

By fixing $\mathbf{w}$ and $\mathbf{z}$, an analytical solution of the $\mathbf{F}$ sub-problem is obtained as:

$$\mathbf{F} = \alpha(\alpha\mathbf{I} + \mathbf{L}')^{-1}\mathbf{Y}, \tag{7}$$

where $\mathbf{L}' = \mu\mathbf{L}^{(t')} + \sum_{t=1,t\neq t'}^{T} w^{(t)}\mathbf{L}^{(t)}$. And furthermore, the computational complexity can be reduced by using an iterative approach discussed in [15], as shown in Algorithm 1.

For the $\mathbf{w}$ sub-problem, it becomes a Quadratic Programming (QP) with other irrelevant terms omitted. There are lots of methods [42,43] to solve QP, and we will not discuss them here. To be

Algorithm 2 Optimization for *MSCD*

Input:
Laplacian matrices $\{\mathbf{L}^{(t)}\}_{t=1}^{T}$, known label $\mathbf{Y}$, parameters α, β, and μ, learning rate ρ
Output:
Label indicator $\mathbf{F}$ and views weight $\mathbf{w}$.
1: Initialize $\mathbf{F}$ and $\mathbf{w}$;
2: Update $\mathbf{F}$ by solving Eq. (7) with Algorithm 1;
3: Update $\mathbf{w}$ by solving Eq. (8);
4: Update $\mathbf{z}$ according to Eq. (10);
5: Update Lagrangian multiplier according to Eq. (12);
6: Repeat steps 2–5 until convergence.

clear, we rewrite the sub-problem in an obvious QP form:

$$\min_{\mathbf{w}} \quad \frac{\rho}{2}\mathbf{w}^{\top}\mathbf{D}^{\top}\mathbf{D}\mathbf{w} + \langle \mathbf{V} + \boldsymbol{\lambda}^{\top}\mathbf{D} - \rho\mathbf{z}^{\top}\mathbf{D}, \mathbf{w}\rangle, \quad s.t. \quad \mathbf{w}^{\top}\mathbf{1} = 1, \mathbf{w} \geq 0, \tag{8}$$

where $\mathbf{V} = (\mathrm{tr}(\mathbf{F}^{\top}\mathbf{L}^{(1)}\mathbf{F}), \ldots, \mathrm{tr}(\mathbf{F}^{\top}\mathbf{L}^{(T)}\mathbf{F}))^{\top}$ but excludes t'th term (the target view).

The $\mathbf{z}$ sub-problem contains an ℓ_1-norm, therefore, we cannot easily obtain the derivative respecting to $\mathbf{z}$. We have the optimization problem as following:

$$\min_{\mathbf{z}} \ \beta\|\mathbf{z}\|_1 + \boldsymbol{\lambda}^{\top}(\mathbf{D}\mathbf{w} - \mathbf{z}) + \frac{\rho}{2}\|\mathbf{D}\mathbf{w} - \mathbf{z}\|_2^2. \tag{9}$$

The closed form solution is given by:

$$\mathbf{z} := \frac{1}{\rho}\mathcal{T}(\boldsymbol{\lambda} + \mathbf{D}\mathbf{w}, \beta), \tag{10}$$

where $\mathcal{T}(v, \eta)$ is soft-shrinkage operator [44] acting on each element of the given vector, whose main idea is keeping $\partial(|z_i|)$ indeterminate and discussing it on three cases of the sign of z_i. It is given by:

$$\mathcal{T}(v, \eta) = \mathrm{sign}(v)\max\{(|v| - \eta), 0\}. \tag{11}$$

Finally, the Lagrangian multiplier $\boldsymbol{\lambda}$ is updated as:

$$\boldsymbol{\lambda} := \boldsymbol{\lambda} + \rho(\mathbf{D}\mathbf{w} - \mathbf{z}). \tag{12}$$

The multiplier $\boldsymbol{\lambda}$ accumulates the error of the subtraction (with learning rate ρ), which encourages to satisfy the constraint iteratively, that is to achieve the mission of the auxiliary variable.

In practice, we could easily check the variation of each variable at new iteration to prove the convergence. In particular, we further define $r^k = \|\mathbf{D}\mathbf{w} - \mathbf{z}\|_2$ and $s^k = \rho\mathbf{D}^{\top}(z^k - z^{k-1})$ at the kth iteration to see whether they are both small enough for the convergence checking. We summarize this process in Algorithm 2.

3.4. Time complexity

Here we provide a brief complexity analysis for Algorithm 2. For the $\mathbf{F}$ sub-problem, we utilize an iterative algorithm to deal with the inverse operation, which reduces the complexity to $\mathcal{O}(n^2k)$ under the settings of n samples and k classes. For the $\mathbf{w}$ sub-problem, solving a QP brings an $\mathcal{O}(T^3L)$ complexity, where $T \ll n$ denotes the number of views, and L is the length of a binary coding of $\mathbf{w}$ [45]. For the $\mathbf{z}$ sub-problem, the time cost only lies on matrix multiplication, therefore, the time complexity is $\mathcal{O}((T-1)(T-2)) = \mathcal{O}(T^2)$. Assume that Algorithm 2 requires t iterations for convergence, then its overall time complexity is of order $\mathcal{O}(t(n^2k + T^3L + T^2))$. Apparently, the proposed learning algorithm is efficient to optimize the objective function.

4. Experimental result

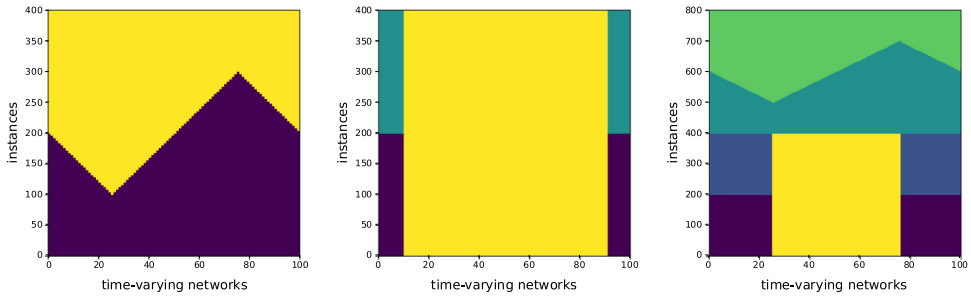
In this section, we evaluate our approach on some synthetic and real-world datasets. Here we raise four related approaches to compare with:

- Single-view Semi-supervised Classification (*SSC*) [41]. This approach is an extension of spectral clustering and spread the information from labeled nodes to their neighbors. This approach is often used as an illustration to examine whether multi-view can enhance classification performance. In a real-world scenario, it is random to have an observation of dynamic networks. Therefore, this approach may suffer a lot from noise and misleading information.
- Optimized Multiple Graph-based Semi-Supervised Learning (*OMG-SSL*) [15]. This approach could leverage multi-graph information but has a hyper-parameter representing the significance of each view. The parameter needs to be tuned manually for every view. The optimization is done by using an EM-style iterative algorithm, which updates the label indicator and views weight alternately.
- Sparse Multiple Graph Integration(*SMGI*) [14]. This approach combines the structural information of the views, and assume they share one label indicator equally. Especially, this approach aims to obtain a sparse weight of the views, which assumes that only a few views are needed to reconstruct the complete structural information.
- Semi-supervised Time Series Classification (*STSC*) [46]. This approach aims to address the SSL of time-series and train a classifier with labeled nodes based on Euclidean distance. Then it uses the classifier to label the rest nodes. This procedure is efficient but needs a sufficient amount of labeled instances to train a better classifier. We will only report the experimental result of *STSC* on the final experiment, which is testing on time-series datasets.

Next, we evaluate the above approaches and ours under three different datasets. The parameters of each approach are tuned for optimal performance following their literature or multiple tests. For the single-view methods, we perform them overall views and report the average results. For every approach, several label rates are tested for ten runs each, and the average performances are reported. We should note that, in transductive learning, the label rates are generally low. Therefore, we perform these experiments at four levels of label rate: 1%, 3%, 5%, and 10%. Meanwhile, for the sake of unbiased comparison, we repeat ten runs for each experiment to reduce the bias of randomly choosing initial label nodes.

4.1. Synthetic dataset experiment

We first generate three synthetic datasets with a simulating algorithm [47], which could generate simple dynamic networks based on stochastic block models. This algorithm simulates several dynamic communities, and the nodes of them will move from one community to another over time with some presupposed pattern. Then, it constructs the communities as undirected, unweighted graphs, that nodes inside the same community are linked with a probability p_{in}, and linked with a probability p_{out} to nodes of other communities. Based on this algorithm, we generated three datasets with three patterns, as shown in Fig. 2. The pattern of Fig. 2(a) is *grow-shrink*, whose nodes will move from one community to another over time, and leading to the size of communities changing. Fig. 2(b) is *merge-split*, the communities are alternately merging and splitting. The generating parameters of it are the same as above. Fig. 2(c) is *mixed*, which contains four communities, and half instances follow the behavior pattern of



(a) grow-shrink (b) merge-split (c) mixed

Fig. 2. Visualization of synthetic datasets. The *x*-axis represents the snapshots of time-varying networks along the time. The colors represent classes of instances. According to different behavior patterns, instances change their classes over time. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Performance comparison on synthetic datasets. The results are the average accuracy rate (and standard derivation) over 10 runs.

Dataset	Label Rate	SSC	OMG-SSL	SMGI	MSCD
Grow	1%	0.5188(0.0099)	0.5580(0.0315)	0.5495(0.0300)	**0.6145(0.0335)**
	3%	0.5621(0.0131)	0.5638(0.0206)	0.5603(0.0171)	**0.6308(0.0272)**
	5%	0.5984(0.0164)	0.6065(0.0318)	0.5908(0.0247)	**0.6625(0.0263)**
	10%	0.6606(0.0112)	0.6630(0.0233)	0.6500(0.0217)	**0.7260(0.0173)**
Merge	1%	0.5124(0.0081)	0.5205(0.0123)	0.5210(0.0172)	**0.5863(0.0344)**
	3%	0.5435(0.0177)	0.5445(0.0193)	0.5397(0.0175)	**0.6125(0.0236)**
	5%	0.5794(0.0296)	0.5880(0.0324)	0.5775(0.0317)	**0.6565(0.0186)**
	10%	0.6114(0.0244)	0.6263(0.0243)	0.6470(0.0294)	**0.7165(0.0283)**
Mixed	1%	0.4190(0.0258)	0.4143(0.0251)	0.4096(0.0266)	**0.4578(0.0203)**
	3%	0.4843(0.0156)	0.4688(0.0185)	0.4594(0.0170)	**0.5656(0.0162)**
	5%	0.5262(0.0146)	0.5039(0.0149)	0.4961(0.0142)	**0.6190(0.0281)**
	10%	0.6095(0.0111)	0.6040(0.0169)	0.5941(0.0164)	**0.7040(0.0164)**

grow-shrink, while others follow the pattern of *merge-split*. Each dataset contains 100 views and 200 instances per class. During the experiment, we set the last view as *target view* and set the label rate at four different levels: 1%, 3%, 5%, and 10%.

As we can see in Table 1, all of the multi-view approaches outperform the single-view *SSC*, which indicates that the role of cross-view relationship in enhancing the accuracy of classification results. Besides, by fitting well with the characteristics of dynamic networks, *MSCD* obtains a sparse and smooth weight (as shown in Fig. 3) and outperforms other multi-view approaches. Furthermore, as the label rate grows, the performance of *MSCD* increases sharply as well. Paying attention to the weight, we can see *SMGI* produces a sparse solution with more than half of the weights are assigned to 0. *OMG-SSL* can map out trends in time, but the weights are too dense to ignore misleading information. We should note that the total variation term of *MSCD* constrains the weight to be locally smooth and globally sparse, which could fit the involving pattern of dynamic networks (as shown in Fig. 2(a)), and assign the relative views higher weights than other irrelative ones.

4.2. Daily and sports activities dataset experiment

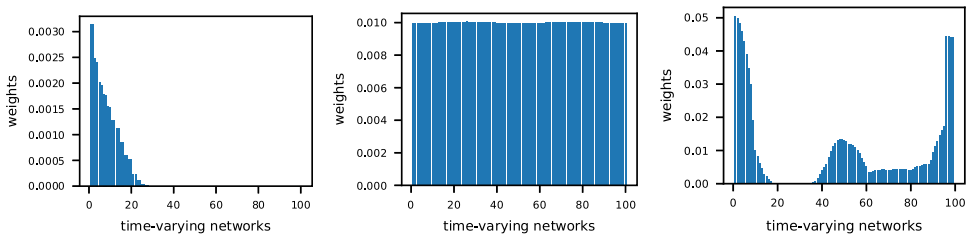
In this section, we employ a real-world dataset named Daily and Sports Activities Dataset [48]. This dataset contains records of human activities, which are recorded by motion sensors on eight subjects. Each record has 45 dimensions representing 45 sensors on one subject and has 5-min length. Sensor units are calibrated to acquire data at 25 Hz sampling frequency. The 5-min signals are divided into 5-s segments so that 480 ($= 60 \times 8$) signal segments are obtained for each activity. We should note that the subjects are asked to perform the activities in their style and were not restricted to how the activities should be performed. By treating signal segments as nodes, we can construct 125 ($=$ 25 Hz $\times$ 5-s) dynamic networks, which represent the action process of activities.

In the experiment, we split the 19 activities into three groups and construct three sets of time-varying networks (shown in Fig. 4). Each group contains 6 or 7 classes, and we randomly select 200 records from each and calculate their similarity with Euclidean measurement. As aforementioned, we preserve 25 time-varying networks from 125 ones by taking one every five, which enlarges the information gap between adjacency networks. We note that static-motion activities are easily distinguishable; on the contrary, it is hard to see the patterns at first glance among dynamic-motion activities.

With similar settings, the results are reported in Table 2. Concretely, for multi-view approaches, label prediction and network weighting are conducted in a unified framework. As we can see, *MSCD* outperforms other methods, even if the label rate is too low to handle (i.e., dataset 7–12). Under the datasets of multiple classes and dynamic networks, *MSCD* can utilize smoothness to find an appropriate weights allocation, as shown in Fig. 5.

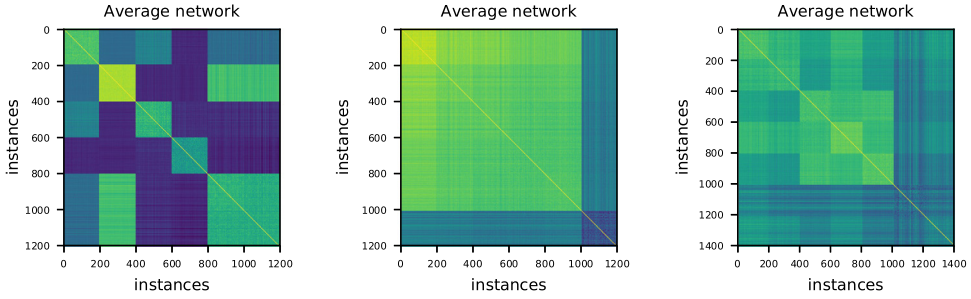
4.3. Time series data experiment

In this section, we further test methods on several time-series datasets [49]. Some of them are recorded by sensors (such as ECG recode). And some are serialized from images of an object (such as handwritten fonts, people practicing yoga, etc.) and stored in structural order. We split each sequence into T sub-segments (typical length is 10–20) and construct T dynamic networks. We obtain the tth network $\mathbf{G}^{(t)}$ by treating the tth sub-segments of all sequences as nodes and calculate the weight of edges with the Euclidean distance measurement. It is noteworthy that $\{\mathbf{G}^{(t)}\}_{t=1}^{T}$ is still arranged in chronological or structural order. We construct networks on three time-series datasets [49]: (1) ECGFiveDays, which contains 884 instances, 2 classes, and 10 networks (views),



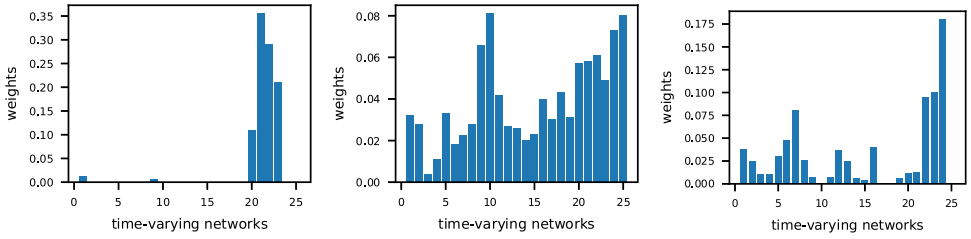
(a) SMGI (b) OMG-SSL (c) MSCD

Fig. 3. The average weights distribution obtained by three methods among 10 runs on grow-shrink datasets, and the label rates were set to 10%. The target view is the last one, therefore, other views similar to it should obtain higher weights.



(a) Categories 1-6 (b) Categories 7-12 (c) Categories 13-19

Fig. 4. Average similarity matrices of three subsets constructed from Daily and Sports Activities Dataset. Lighter points represent higher similarity.



(a) Categories 1-6 (b) Categories 7-12 (c) Categories 13-19

Fig. 5. The average weights distribution obtained by MSCD among 10 runs on Daily and Sports Activities Datasets, and the label rates were set to 10%. The weights are locally smooth and globally sparse.

Table 2
Performance comparison on daily and sports activities datasets. The results are the average accuracy rate (and standard derivation) over 10 runs.

Dataset	Label Rate	SSC	OMG-SSL	SMGI	MSCD
1–6	1%	0.6727(0.0119)	0.6702(0.0005)	0.6730(0.0053)	**0.7107(0.0277)**
	3%	0.6805(0.0096)	0.6767(0.0000)	0.6777(0.0014)	**0.7145(0.0095)**
	5%	0.6851(0.0039)	0.6833(0.0000)	0.6841(0.0009)	**0.7217(0.0019)**
	10%	0.7032(0.0020)	0.7000(0.0000)	0.7003(0.0005)	**0.7361(0.0014)**
7–12	1%	0.2065(0.0042)	0.1750(0.0000)	0.1753(0.0005)	**0.2631(0.0222)**
	3%	0.2033(0.0030)	0.1917(0.0000)	0.1918(0.0002)	**0.2418(0.0250)**
	5%	0.2135(0.0014)	0.2083(0.0000)	0.2083(0.0000)	**0.2689(0.0115)**
	10%	0.2519(0.0006)	0.2500(0.0000)	0.2500(0.0000)	**0.3018(0.0117)**
13–19	1%	0.4026(0.0080)	0.5311(0.0633)	0.6529(0.0471)	**0.6584(0.0394)**
	3%	0.4229(0.0138)	0.4966(0.0623)	**0.6981(0.0197)**	**0.6981(0.0203)**
	5%	0.4388(0.0109)	0.5133(0.0382)	0.7113(0.0161)	**0.7149(0.0147)**
	10%	0.4731(0.0105)	0.5419(0.0361)	0.7311(0.0117)	**0.7312(0.0111)**

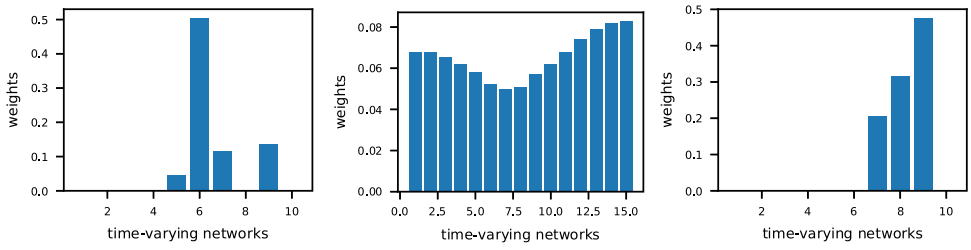
(2) Yoga, which contains 3300 instances, 2 classes, and 15 networks, and (3) ECG5000, which contains 5000 instances, 5 classes, and 10 networks.

In this task, we also perform the time-series-based method *STSC* on binary classification datasets. During the experiment, we will perform it on each network (as the pre-calculated distance matrix), and report the average performance. The settings for other methods are the same as before.

The mean accuracy and standard of 10 times repeated experiments under each label rate are shown in Table 3. STSC can only process binary classification tasks, hence that we report no result about it on multiple classes dataset. We should note that *MSCD* results in better performance with low label rates (1% and 3%).

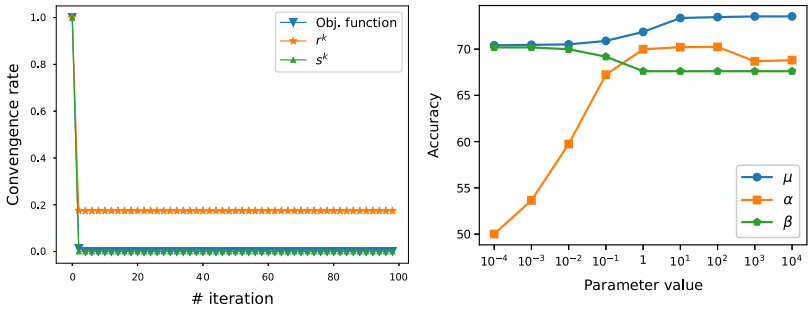
Table 3
Performance comparison on time series datasets. The results are the average accuracy rate (and standard derivation) over 10 runs.

Dataset	Label Rate	SSC	OMG-SSL	SMGI	STSC	MSCD
ECGFiveDay	1%	0.5905(0.0264)	0.6699(0.0812)	0.6749(0.0799)	0.4714(0.0319)	**0.7982(0.0395)**
	3%	0.6180(0.0159)	0.7672(0.0701)	0.7854(0.0611)	0.5033(0.0319)	**0.8295(0.0518)**
	5%	0.6247(0.0209)	0.7905(0.0555)	0.8202(0.0370)	0.5082(0.0319)	**0.8442(0.0630)**
	10%	0.6514(0.0147)	0.8439(0.0583)	**0.8828(0.0452)**	0.5362(0.0319)	0.8644(0.0252)
Yoga	1%	0.5200(0.0297)	0.5168(0.0181)	0.5191(0.0184)	0.5253(0.0202)	**0.5361(0.0183)**
	3%	0.5450(0.0262)	0.5450(0.0391)	0.5452(0.0372)	0.5294(0.0183)	**0.5551(0.0394)**
	5%	0.5579(0.0254)	0.5939(0.0252)	**0.6005(0.0476)**	0.5370(0.0144)	0.5825(0.0297)
	10%	0.5222(0.0295)	0.5967(0.0403)	**0.6126(0.0381)**	0.5253(0.0202)	0.6077(0.0379)
ECG5000	1%	0.5882(0.0000)	0.7612(0.1706)	0.7428(0.1644)	–	**0.7615(0.1706)**
	3%	0.5966(0.0000)	0.8344(0.1747)	0.8208(0.1537)	–	**0.8360(0.1648)**
	5%	0.6050(0.0000)	0.8473(0.1202)	0.8517(0.0768)	–	**0.8528(0.0838)**
	10%	0.6258(0.0000)	0.8473(0.1202)	0.8517(0.0768)	–	**0.9132(0.0066)**



(a) ECGFiveDays (b) Yoga (c) ECG5000

Fig. 6. The average weights distribution obtained by MSCD among 10 runs on three time series datasets, and the label rates were set to 10%. It is possible to obtain smooth but sparse weights.



(a) Convergence rate analysis (b) The effect of parameters

Fig. 7. The results of convergence rate and parameter sensitivity experiment.

Meanwhile, the performances of other methods increase with the incrementation of label rates. In Fig. 6, we report the weight distribution of *MSCD* results for each dataset. In general, *MSCD* can combine the information of multiple dynamic networks and obtain an optimal weight ratio according to the time smoothness.

4.4. Convergence rate and parameter sensitivity experiment

In this section, we further investigate how fast the optimization of *MSCD* would converge. We perform *MSCD* on the synthetic dataset *grow* with 10% label rate. To verify the convergence speed, we fixed the parameters to an optimal combination, and show the curves of converging rates in Fig. 7(a). In particular, these curves are normalized, and the 0th iteration is initialization. We can find out that *MSCD* can achieve fast convergence within a few iterations. The tendencies of r^k and s^k are identical to the objective function. Therefore, we can use these two variables to verify convergence and reduce computational consumption.

To evaluate the sensitivity of the parameters, μ, α and β are tuned in the range of $\{10^{-4}, 10^{-3}, \ldots, 1, 10^{1}, \ldots, 10^{4}\}$. We report the performance results in Fig. 7(b), which indicates that the optimal ranges of parameters are board, and the parameters tuning in real-world applications could be an easy job.

5. Conclusion

Nodes classification of dynamic networks remains a challenging problem. We find out the complementary relationships among multiple observations of a dynamic network at different epochs, and propose a multi-view learning approach to leverage it, i.e., Multi-view Semi-supervised learning for Classification on Dynamic networks (*MSCD*). To fit the smoothness of continuous observations, we introduce the total variation regularization to ensure. Furthermore, we apply the ADMM framework to decompose the model into several sub-problem, both of which are easy to implement. Extensive experimental results demonstrate that under low label rates, our model can classify nodes more accurately than the baseline methods.

CRediT authorship contribution statement

Chuan Chen: Conceptualization, Methodology, Resources. **Yuzheng Li:** Data curation, Writing - original draft, Software. **Hui Qian:** Validation, Visualization, Investigation. **Zibin Zheng:** Supervision, Writing - review & editing. **Yanqing Hu:** Validation, Writing - review & editing.

Acknowledgments

The work described in this paper was supported by the National Key Research and Development Program, China (2016YFB1000101), the National Natural Science Foundation of China (11801595, 61722214, 61773412), Guangzhou Science and Technology Program key Project, China (201804010473), Guangdong Research and Development Program in Key Fields, China (2019B020214002), the Natural Science Foundation of Guangdong, China (2018A030310076) and the CCF-Tencent Open Fund WeBank Special Funding, China.

References

[1] O. Chapelle, J. Weston, B. Schölkopf, Cluster kernels for semi-supervised learning, in: Advances in Neural Information Processing Systems, Vol. 15, Neural Information Processing Systems, NIPS 2002, December 9-14, 2002, Vancouver, British Columbia, Canada, 2002, pp. 585–592, URL http://papers.nips.cc/paper/2257-cluster-kernels-for-semi-supervised-learning.

[2] O. Chapelle, B. Scholkopf, E. A. Zien, Semi-supervised learning (Chapelle, o. et al., eds. 2006) [book reviews], IEEE Trans. Neural Netw. 20 (3) (2009) 542, http://dx.doi.org/10.1109/tnn.2009.2015974.

[3] F. Ye, C. Chen, J. Zhang, J. Wu, Z. Zheng, An adaptive semi-local algorithm for node ranking in large complex networks, in: Service-Oriented Computing - 16th International Conference, ICSOC 2018, Hangzhou, China, November 12-15, 2018, Proceedings, 2018, pp. 505–514, http://dx.doi.org/10.1007/978-3-030-03596-9_36.

[4] W. Hu, C. Chen, F. Ye, Z. Zheng, G. Ling, Nonnegative spectral clustering for large-scale semi-supervised learning, in: Database Systems for Advanced Applications - 24th International Conference, DASFAA 2019, Chiang Mai, Thailand, April 22-25, 2019, Proceedings, Part III, and DASFAA 2019 International Workshops: BDMS, BDQM, and GDMA, Chiang Mai, Thailand, April 22-25, 2019, Proceedings, 2019, pp. 287–291, http://dx.doi.org/10.1007/978-3-030-18590-9_30.

[5] Y. Li, C. Chen, F. Ye, Z. Zheng, G. Ling, A semi-supervised classification approach for multiple time-varying networks with total variation, in: Database Systems for Advanced Applications - 24th International Conference, DASFAA 2019, Chiang Mai, Thailand, April 22-25, 2019, Proceedings, Part III, and DASFAA 2019 International Workshops: BDMS, BDQM, and GDMA, Chiang Mai, Thailand, April 22-25, 2019, Proceedings, 2019, pp. 334–337, http://dx.doi.org/10.1007/978-3-030-18590-9_40.

[6] U. Von Luxburg, A tutorial on spectral clustering, Stat. Comput. 17 (4) (2007) 395–416.

[7] S. Fortunato, Community detection in graphs, Phys. Rep. 486 (3–5) (2010) 75–174.

[8] A. Kumar, H. Daumé, A co-training approach for multi-view spectral clustering, in: Proceedings of the 28th International Conference on Machine Learning, ICML-11, 2011, pp. 393–400.

[9] L. Zelnik-Manor, P. Perona, Self-tuning spectral clustering, in: Advances in Neural Information Processing Systems, 2005, pp. 1601–1608.

[10] R.A.R. Ashfaq, X.-Z. Wang, J.Z. Huang, H. Abbas, Y.-L. He, Fuzziness based semi-supervised learning approach for intrusion detection system, Inform. Sci. 378 (2017) 484–497, http://dx.doi.org/10.1016/j.ins.2016.04.019.

[11] M. Herbster, M. Pontil, L. Wainer, Online Learning Over Graphs, ICML-05, ACM Press, 2005, pp. 305–312, http://dx.doi.org/10.1145/1102351.1102390.

[12] T. Joachims, Transductive Learning Via Spectral Graph Partitioning, ICML-03, 2003, pp. 290–297, URL http://www.aaai.org/Papers/ICML/2003/ICML03-040.pdf.

[13] C. Xu, D. Tao, C. Xu, A survey on multi-view learning, 37, 2013, pp. 2531–2544, http://dx.doi.org/10.1109/tpami.2015.2417578, arXiv preprint arXiv:1304.5634.

[14] M. Karasuyama, H. Mamitsuka, Multiple graph label propagation by sparse integration, IEEE Trans. Neural Netw. Learn. Syst. 24 (12) (2013) 1999–2012, http://dx.doi.org/10.1109/tnnls.2013.2271327.

[15] M. Wang, X.-S. Hua, R. Hong, J. Tang, G.-J. Qi, Y. Song, Unified video annotation via multigraph learning, IEEE Trans. Circuits Syst. Video Technol. 19 (5) (2009) 733–746, http://dx.doi.org/10.1109/tcsvt.2009.2017400.

[16] C. Chen, J. Xin, Y. Wang, L. Chen, M.K. Ng, A semisupervised classification approach for multidomain networks with domain selection, IEEE Trans. Neural Netw. Learn. Syst. (99) (2018) 1–15, http://dx.doi.org/10.1109/tnnls.2018.2837166.

[17] H. Wang, Y. Yang, B. Liu, H. Fujita, A study of graph-based system for multi-view clustering, Knowl.-Based Syst. 163 (2019) 1009–1019, http://dx.doi.org/10.1016/j.knosys.2018.10.022.

[18] Z. Yi, Y. Yang, T. Li, H. Fujita, A multitask multiview clustering algorithm in heterogeneous situations based on LLE and LE, Knowl.-Based Syst. 163 (2019) 776–786, http://dx.doi.org/10.1016/j.knosys.2018.10.001.

[19] L.I. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, Physica D 60 (1–4) (1992) 259–268, http://dx.doi.org/10.1016/0167-2789(92)90242-f.

[20] Z. Xu, R. Jin, J. Zhu, I. King, M.R. Lyu, Z. Yang, Adaptive regularization for transductive support vector machine, in: Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a Meeting Held 7-10 December 2009, Vancouver, British Columbia, Canada, 2009 pp. 2125–2133.

[21] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al., Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends® Mach. Learn. 3 (1) (2011) 1–122, http://dx.doi.org/10.1561/2200000016.

[22] B. Wahlberg, S. Boyd, M. Annergren, Y. Wang, An ADMM algorithm for a class of total variation regularized estimation problems, 45, 2012, pp. 83–88, http://dx.doi.org/10.3182/20120711-3-be-2027.00310, arXiv preprint arXiv:1203.1828.

[23] A. Kumar, P. Rai, H. Daume, Co-regularized multi-view spectral clustering, in: Advances in Neural Information Processing Systems, 2011, pp. 1413–1421.

[24] H. Zhao, Z. Ding, Y. Fu, Multi-view clustering via deep matrix factorization, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[25] Z. Tao, H. Liu, S. Li, Z. Ding, Y. Fu, From ensemble clustering to multi-view clustering, in: IJCAI, 2017.

[26] S. Huang, Z. Kang, Z. Xu, Self-weighted multi-view clustering with soft capped norm, Knowl.-Based Syst. 158 (2018) 1–8, http://dx.doi.org/10.1016/j.knosys.2018.05.017.

[27] S. Huang, Z. Kang, I.W. Tsang, Z. Xu, Auto-weighted multi-view clustering via kernelized graph learning, Pattern Recognit. 88 (2019) 174–184, http://dx.doi.org/10.1016/j.patcog.2018.11.007.

[28] C. Chen, H. Qian, W. Chen, Z. Zheng, H. Zhu, Auto-weighted multi-view constrained spectral clustering, Neurocomputing 366 (2019) 1–11, http://dx.doi.org/10.1016/j.neucom.2019.06.098.

[29] X. Zhou, X. Zhao, Y. Liu, A multiobjective discrete bat algorithm for community detection in dynamic networks, Appl. Intell. 48 (9) (2018) 3081–3093, http://dx.doi.org/10.1007/s10489-017-1135-5.

[30] S. Fortunato, Community detection in graphs, 2009, arXiv:0906.0612, CoRR abs/0906.0612, URL http://arxiv.org/abs/0906.0612.

[31] F. Ye, C. Chen, Z. Wen, Z. Zheng, W. Chen, Y. Zhou, Homophily preserving community detection, IEEE Trans. Neural Netw. Learn. Syst. (2019) 1–13, http://dx.doi.org/10.1109/TNNLS.2019.2933850.

[32] F. Zhou, S. Yang, H. Fujita, D. Chen, C. Wen, Deep learning fault diagnosis method based on global optimization GAN for unbalanced data, Knowl.-Based Syst. 187 (2020) http://dx.doi.org/10.1016/j.knosys.2019.07.008.

[33] I. Muslea, S. Minton, C.A. Knoblock, Active learning with multiple views, J. Artificial Intelligence Res. 27 (2006) 203–233.

[34] X. Zhu, Z. Ghahramani, Learning from Labeled and Unlabeled Data with Label Propagation, Tech. Rep., Citeseer, 2002, http://dx.doi.org/10.21236/ada565197.

[35] I. Muslea, S. Minton, C.A. Knoblock, Active+ semi-supervised learning= robust multi-view learning, in: ICML, Vol. 2, 2002, pp. 435–442.

[36] Q. Xiao, J. Dai, J. Luo, H. Fujita, Multi-view manifold regularized learning-based method for prioritizing candidate disease miRNAs, Knowl.-Based Syst. 175 (2019) 118–129, http://dx.doi.org/10.1016/j.knosys.2019.03.023.

[37] N.M.A. Ibrahim, L. Chen, Link prediction in dynamic social networks by integrating different types of information, Appl. Intell. 42 (4) (2015) 738–750, http://dx.doi.org/10.1007/s10489-014-0631-0.

[38] X. Lei, M. Fang, H. Fujita, Moth-flame optimization-based algorithm with synthetic dynamic PPI networks for discovering protein complexes, Knowl.-Based Syst. 172 (2019) 76–85, http://dx.doi.org/10.1016/j.knosys.2019.02.011.

[39] Y. Jing, T. Li, H. Fujita, B. Wang, N. Cheng, An incremental attribute reduction method for dynamic data mining, Inform. Sci. 465 (2018) 202–218, http://dx.doi.org/10.1016/j.ins.2018.07.001.

[40] Y. Huang, T. Li, C. Luo, H. Fujita, S. Horng, Dynamic fusion of multisource interval-valued data by fuzzy granulation, IEEE Trans. Fuzzy Syst. 26 (6) (2018) 3403–3417, http://dx.doi.org/10.1109/TFUZZ.2018.2832608.

[41] D. Zhou, O. Bousquet, T.N. Lal, J. Weston, B. Schölkopf, Learning with local and global consistency, in: NIPS-04, 2004, pp. 321–328.

[42] N. Karmarkar, A new polynomial-time algorithm for linear programming, in: Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing, ACM Press, 1984, pp. 302–311, http://dx.doi.org/10.1145/800057.808695.

[43] S. Mehrotra, On the implementation of a primal-dual interior point method, SIAM J. Optim. 2 (4) (1992) 575–601, http://dx.doi.org/10.1137/0802028.

[44] J. Bioucas-Dias, M. Figueiredo, A new twist: Two-step iterative shrinkage/thresholding algorithms for image restoration, IEEE Trans. Image Process. 16 (12) (2007) 2992–3004, http://dx.doi.org/10.1109/tip.2007.909319.

[45] F.A. Potra, S.J. Wright, Interior-point methods, J. Comput. Appl. Math. 124 (1–2) (2000) 281–302, http://dx.doi.org/10.1016/s0377-0427(00)00433-7.

[46] L. Wei, E. Keogh, Semi-Supervised Time Series Classification, SIGKDD-06, ACM Press, 2006, pp. 748–753, http://dx.doi.org/10.1145/1150402.1150498.

[47] C. Granell, R.K. Darst, A. Arenas, S. Fortunato, S. Gómez, Benchmark model to assess community structure in evolving networks, Phys. Rev. E 92 (1) (2015) 012805, http://dx.doi.org/10.1103/physreve.92.012805.

[48] D. Dheeru, E. Karra Taniskidou, UCI Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences, 2017, URL http://archive.ics.uci.edu/ml.

[49] H.A. Dau, E. Keogh, K. Kamgar, C.-C.M. Yeh, Y. Zhu, S. Gharghabi, C.A. Ratanamahatana, Yanping, B. Hu, N. Begum, A. Bagnall, A. Mueen, G. Batista, The UCR time series classification archive, 2018, https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.