Toward Time-Evolving Feature Selection on Dynamic Networks

Jundong Li*, Xia Hu†, Ling Jian‡ and Huan Liu*
*Computer Science and Engineering, Arizona State University, USA
†Computer Science and Engineering, Texas A&M University, USA
‡College of Science, China University of Petroleum, China
Email: *jundongl@asu.edu, †hu@cse.tamu.edu, ‡jianlingupc@gmail.com, *huan.liu@asu.edu

***Abstract*—Recent years have witnessed the prevalence of networked data in various domains. Among them, a large number of networks are not only topologically structured but also have a rich set of features on nodes. These node features are usually of high dimensionality with noisy, irrelevant and redundant information, which may impede the performance of other learning tasks. Feature selection is useful to alleviate these critical issues. Nonetheless, a vast majority of existing feature selection algorithms are predominantly designed in a static setting. In reality, real-world networks are naturally dynamic, characterized by both topology and content changes. It is desirable to capture these changes to find relevant features tightly hinged with network structure continuously, which is of fundamental importance for many applications such as disaster relief and viral marketing. In this paper, we study a novel problem of time-evolving feature selection for dynamic networks in an unsupervised scenario. Specifically, we propose a TeFS framework by leveraging the temporal evolution property of dynamic networks to update the feature selection results incrementally. Experimental results show the superiority of TeFS over the state-of-the-art batch-mode unsupervised feature selection algorithms.**

I. Introduction

The proliferation of various information systems such as social media services, transportation systems and mobile cellular platforms enables the accessibility of sheer amounts of networked data. In such networks, nodes stand for entities while edges represent interactions between entities. To date, some dedicated learning algorithms have been put forward to derive actionable patterns or knowledge from these networked data. More often than not, in addition to the connectivity information, nodes in the networks are affiliated with a rich set of features. For instance, in academic networks, we are given a list of scholars as well as their specialties and research interests; in bioinformatics, we are provided with profiles of genes to uncover their behaviors. Recent studies have shown that network structures and node features are strongly correlated, and the exploration of the dependency could advance various applications [13], [17], [21]. The root cause of the correlations can be explained by some social science theories including social influence and homophily [6], [12].

Nonetheless, the illusion that all features are dovetailed with network topological structure is not always true. As in the case of academic collaboration networks, features like gender are rather more independent of network structure than discerning features like research interests. On top of that, deviating or noisy features that are not inconsistent with network topology may jeopardize the discovery of explainable patterns. These observations necessitate the usage of feature selection algorithms [7] to find a set of relevant features tightly hinged with network structure. Recently, a few advanced feature selection algorithms are outlined to gain insights from networked data [3], [9], [14], [19]. However, these methods predominantly focus on a static setting, while fail to characterize the evolution facts of networks and features. As per the dynamic network theory [20] in psychology and social sciences, network and features often co-evolve over time, and a small disturbance of network structure may result in a ripple effect on the drifts of feature patterns, and vice versa. With these unique characteristics, existing approaches probing dynamic networks are either correcting and adjusting the staleness of network mining algorithms or understanding the underlying evolution mechanisms [1], [23].

Despite the fundamental importance of analyzing dynamic attributed networks, the development of sophisticated learning models to find relevant features in a time manner is still in its infancy. Given the dynamic nature of networked data, it is necessary and of vital importance to perform feature selection to adapt to both network and content changes. In particular, we focus on time-evolving feature selection in an unsupervised scenario as label information is time and labor intensive to obtain. In this regard, the main challenges of performing time-evolving feature selection on dynamic networks lie in the following two aspects: (1) Even though network and features are often presented in different modalities, they are not mutually independent and could influence each other, feature selection algorithms should be able to seize the interconnections; (2) Both network topological structure and node features may evolve over time, feature selection needs to accommodate to the changes to correct the selected feature set timely. To tackle the above challenges, we propose a novel unsupervised time-evolving feature selection framework TeFS for dynamic networks. The main contributions are as follows: first, we formally define the problem of time-evolving feature selection for dynamic networks; second, we propose an unsupervised time-evolving feature selection framework TeFS for dynamic networks by leveraging the smooth evolving properties; third, we evaluate the efficacy and efficiency of the proposed TeFS framework on real-world datasets.

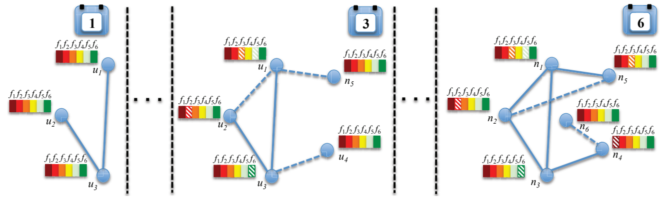


Fig. 1: Illustration of unsupervised time-evolving feature selection on dynamic networks. At each time step, new users may be added which may introduce new links, feature values of users may also change accordingly (shadowed color bar in the figure). The task is to effectively and efficiently maintain a subset of relevant features $\mathcal{F}^t$ at each time step t.

II. Problem Statement

We first summarize some notations used in this paper. We use bold uppercase characters for matrices (e.g., $\mathbf{A}$), normal lowercase characters for scalars (e.g., a), calligraphic fonts for sets (e.g. $\mathcal{F}$). $\mathbf{A}^t$ denotes the matrix at time step t. In addition, i-th row, j-th column, (i,j)-th entry and transpose of matrix $\mathbf{A}$ are denoted as $\mathbf{A}(i,:)$, $\mathbf{A}(:,j)$, $\mathbf{A}(i,j)$ and $\mathbf{A}'$, respectively. Also, trace of $\mathbf{A}$ is represented as $Tr(\mathbf{A})$ if it is square. For any matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$, its Frobenius norm is $||\mathbf{A}||_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{d}\mathbf{A}(i,j)^2}$, its $\ell_{2,1}$-norm is $||\mathbf{A}||_{2,1} = \sum_{i=1}^{n}\sqrt{\sum_{j=1}^{d}\mathbf{A}(i,j)^2}$.

In this work, we use the adjacency matrix $\mathbf{A}^t \in \mathbb{R}^{n\times n}$ to represent the network structure at time step t. We also assume that the network structure is encoded in an undirected graph, i.e., $\mathbf{A}^t = \mathbf{A}^{t'}$. Each linked instance is associated with a set of d features $\mathcal{F} = \{f_1, f_2, ..., f_d\}$ at each time step, $\mathbf{X}^t \in \mathbb{R}^{n\times d}$ denotes the feature information of all n instances at time step t. The feature matrix $\mathbf{X}^t$ and the adjacency matrix $\mathbf{A}^t$ both evolve over time. Following the settings of [2], [16] and for ease of presentation, we do not discuss the addition of new instances or new features, but our method can be naturally applied to deal with these situations. The general process of unsupervised time-evolving feature selection is illustrated in Fig. 1. As can be observed, it aims to find a subset of relevant features adaptively at each time step when both sources of information evolve. The formal problem statement is summarized as follows:

Problem 1. *Time-evolving method that captures the dynamics of feature information and network structure for a subset of relevant features at each time step $t, t+1, ..., t+i$.*

Given: *Feature information represented by a set of matrices $\{\mathbf{X}^t, \mathbf{X}^{t+1}, ..., \mathbf{X}^{t+i}\}$ and network structure encoded in a set of adjacency matrices $\{\mathbf{A}^t, \mathbf{A}^{t+1}, ..., \mathbf{A}^{t+i}\}$.*

Select: *Relevant feature subsets $\{\mathcal{F}^t, \mathcal{F}^{t+1}, ..., \mathcal{F}^{t+i}\}$ at each time step.*

III. The Proposed TeFS Framework

One widely adopted framework to analyze evolutionary network is to leverage the *temporal smoothness* property [1] which assumes that the structure of the network does not change significantly within a short period of time. In particular, given two consecutive time steps t_1 and t_2, it attempts to incrementally adjust the data mining results at time step t_2 by taking advantage of network structure perturbation and the results from the previous time step t_1. As there exhibits a strong dependency between network topology and node features, the evolution of networks influences and is dependent on the corresponding node feature changes. Therefore, to maintain the freshness of the end feature selection results, we build our model on the basis of NetFS [9], which is the state-of-the-art feature selection algorithm for networked data in a static setting. In particular, we present a dynamic model TeFS to employ the temporal smoothness property in updating the feature selection results incrementally.

A. Preliminary - NetFS

NetFS [9] is one of the state-of-the-art unsupervised feature selection algorithms for networked data. Before that, a vast majority of methods [8], [15] separate network structure characterization and feature modeling. For example in [15], latent factors are first extracted from the network structure; then these latent factors act as labels to help assess feature relevance. Therefore, the results of feature selection heavily depend on the quality of extracted latent factors, and noisy links may hinder the selection of discriminative features. To address this issue, NetFS embeds latent representation learning into feature selection:

$$\min_{\mathbf{U}^t\geq 0,\mathbf{W}^t} ||\mathbf{X}^t\mathbf{W}^t - \mathbf{U}^t||_F^2 + \alpha||\mathbf{W}^t||_{2,1} + \frac{\beta}{2}||\mathbf{A}^t - \mathbf{U}^t\mathbf{U}^{t'}||_F^2, \quad (1)$$

where $\mathbf{U}^t \in \mathbb{R}^{n\times k}$ is the network latent representation, β is used to balance the contribution of feature selection and latent representation learning, $\alpha||\mathbf{W}^t||_{2,1}$ is a sparse regularization term that controls the joint feature sparsity across all k latent factors. More details of NetFS can be referred to [9].

B. Dynamic Model - TeFS

We now discuss how to perform feature selection in a time manner when network structure and the corresponding feature values evolve over time on the basis of NetFS. Regarding the *temporal smoothness* assumption, at a new time step $t+1$, both network structure and feature information evolve smoothly. Therefore, $\mathbf{X}^{t+1}$ and $\mathbf{A}^{t+1}$ can be represented as:

$$\mathbf{X}^{t+1} = \mathbf{X}^t + \Delta\mathbf{X}, \text{ and } \mathbf{A}^{t+1} = \mathbf{A}^t + \Delta\mathbf{A}, \quad (2)$$

where $\Delta\mathbf{X}$ and $\Delta\mathbf{A}$ indicate the small changes of feature information and network structure between time steps t and $t+1$, respectively. Then feature selection problem at the new time step $t+1$ can be formulated by solving the following optimization problem:

$$\begin{aligned}\min_{\mathbf{U}^{t+1}\geq 0,\mathbf{W}^{t+1}} &||(\mathbf{X}^t + \Delta\mathbf{X})\mathbf{W}^{t+1} - \mathbf{U}^{t+1}||_F^2 + \alpha||\mathbf{W}^{t+1}||_{2,1} \\ &+ \frac{\beta}{2}||(\mathbf{A}^t + \Delta\mathbf{A}) - \mathbf{U}^{t+1}\mathbf{U}^{t+1'}||_F^2.\end{aligned} \quad (3)$$

Let $\mathbf{W}^{t+1}$ and $\mathbf{U}^{t+1}$ be represented as:

$$\mathbf{W}^{t+1} = \mathbf{W}^t + \Delta\mathbf{W}, \text{ and } \mathbf{U}^{t+1} = \mathbf{U}^t + \Delta\mathbf{U}, \quad (4)$$

then Eq. (3) can be reformulated as:

$$
\begin{aligned}
&\min_{\mathbf{U}^t+\Delta\mathbf{U}\geq 0,\mathbf{W}^t+\Delta\mathbf{W}} \mathcal{J}(\mathbf{W}^t+\Delta\mathbf{W},\mathbf{U}^t+\Delta\mathbf{U})\\
=&||(\mathbf{X}^t+\Delta\mathbf{X})(\mathbf{W}^t+\Delta\mathbf{W})-(\mathbf{U}^t+\Delta\mathbf{U})||_F^2\\
&+\frac{\beta}{2}||(\mathbf{A}^t+\Delta\mathbf{A})-(\mathbf{U}^t+\Delta\mathbf{U})(\mathbf{U}^t+\Delta\mathbf{U})'||_F^2\\
&+\alpha||\mathbf{W}^t+\Delta\mathbf{W}||_{2,1}\\
=&||\mathbf{X}^t\mathbf{W}^t+\Delta\mathbf{X}(\mathbf{W}^t+\Delta\mathbf{W})+\mathbf{X}^t\Delta\mathbf{W}-\mathbf{U}^t-\Delta\mathbf{U}||_F^2\\
&+\frac{\beta}{2}||\mathbf{A}^t+\Delta\mathbf{A}-\mathbf{U}^t(\mathbf{U}^{t'}+\Delta\mathbf{U}')-\Delta\mathbf{U}(\mathbf{U}^{t'}+\Delta\mathbf{U}')||_F^2\\
&+\alpha||\mathbf{W}^t+\Delta\mathbf{W}||_{2,1}.
\end{aligned}
\tag{5}
$$

According to the triangle inequality of matrix norms, we have the following from Eq. (5):

$$
\begin{aligned}
&||\mathbf{X}^t\mathbf{W}^t+\Delta\mathbf{X}(\mathbf{W}^t+\Delta\mathbf{W})+\mathbf{X}^t\Delta\mathbf{W}-\mathbf{U}^t-\Delta\mathbf{U}||_F^2\\
\leq&||\Delta\mathbf{X}\mathbf{W}^t+\mathbf{X}^t\Delta\mathbf{W}+\Delta\mathbf{X}\Delta\mathbf{W}-\Delta\mathbf{U}||_F^2\\
&+||\mathbf{X}^t\mathbf{W}^t-\mathbf{U}^t||_F^2,
\end{aligned}
\tag{6}
$$

$$
\begin{aligned}
&||\mathbf{A}^t+\Delta\mathbf{A}-\mathbf{U}^t(\mathbf{U}^{t'}+\Delta\mathbf{U}')-\Delta\mathbf{U}(\mathbf{U}^{t'}+\Delta\mathbf{U}')||_F^2\\
\leq&||\Delta\mathbf{A}-\Delta\mathbf{U}\mathbf{U}^{t'}-\mathbf{U}^t\Delta\mathbf{U}'-\Delta\mathbf{U}\Delta\mathbf{U}'||_F^2+||\mathbf{A}^t-\mathbf{U}^t\mathbf{U}^{t'}||_F^2
\end{aligned}
\tag{7}
$$

and

$$
||\mathbf{W}^t+\Delta\mathbf{W}||_{2,1}\leq||\mathbf{W}^t||_{2,1}+||\Delta\mathbf{W}||_{2,1}.
\tag{8}
$$

Integrating Eq. (6), Eq. (7) and Eq. (8), we find that the solutions of $\mathbf{W}^t$ and $\mathbf{U}^t$ in the following part

$$
\min_{\mathbf{W}^t,\mathbf{U}^t\geq 0}||\mathbf{X}^t\mathbf{W}^t-\mathbf{U}^t||_F^2+\frac{\beta}{2}||\mathbf{A}^t-\mathbf{U}^t\mathbf{U}^{t'}||_F^2+\alpha||\mathbf{W}^t||_{2,1}
\tag{9}
$$

can be obtained from time step t. Hence, we can only optimize the following part to approximate the solution of Eq. (5):

$$
\begin{aligned}
&\min_{\mathbf{U}^t+\Delta\mathbf{U}\geq 0,\Delta\mathbf{W}}||\Delta\mathbf{X}\mathbf{W}^t+\mathbf{X}^t\Delta\mathbf{W}+\Delta\mathbf{X}\Delta\mathbf{W}-\Delta\mathbf{U}||_F^2\\
&+\frac{\beta}{2}||\Delta\mathbf{A}-\Delta\mathbf{U}\mathbf{U}^{t'}-\mathbf{U}^t\Delta\mathbf{U}'-\Delta\mathbf{U}\Delta\mathbf{U}'||_F^2+\alpha||\Delta\mathbf{W}||_{2,1}
\end{aligned}
\tag{10}
$$

C. Alternating Optimization Algorithm of TeFS

As can be observed, the objective function in Eq. (10) is not a convex function w.r.t. $\Delta\mathbf{W}$ and $\Delta\mathbf{U}$ simultaneously. Thus we adopt an alternating optimization algorithm to tackle it.

1) Given $\Delta\mathbf{U}$, Update $\Delta\mathbf{W}$: First, when $\Delta\mathbf{U}$ is fixed, we remove the terms that are irrelevant to $\Delta\mathbf{U}$ in Eq. (10). Then, $\Delta\mathbf{W}$ can be obtained by minimizing the following:

$$
\min_{\Delta\mathbf{W}}||\Delta\mathbf{X}\mathbf{W}^t+\mathbf{X}^t\Delta\mathbf{W}+\Delta\mathbf{X}\Delta\mathbf{W}-\Delta\mathbf{U}||_F^2+\alpha||\Delta\mathbf{W}||_{2,1}.
\tag{11}
$$

Then we have the following theorem to update $\Delta\mathbf{W}$:

Theorem 1. *When $\Delta\mathbf{U}$ is fixed, $\Delta\mathbf{W}$ can be updated by:*

$$
\Delta\mathbf{W}=(\mathbf{X}^{t+1'}\mathbf{X}^{t+1}+\alpha\Delta\mathbf{D})^{-1}\mathbf{X}^{t+1'}(\Delta\mathbf{U}-\Delta\mathbf{X}\mathbf{W}),
\tag{12}
$$

where $\Delta\mathbf{D}\in\mathbb{R}^{d\times d}$ is a diagonal matrix with the i-th diagonal element as $\Delta\mathbf{D}(i,i)=\frac{1}{2||\Delta\mathbf{W}(i,:)||_2}$.

Proof: By computing the derivative of the objective function in Eq. (11) w.r.t. $\Delta\mathbf{W}$ and set it to be zero, we obtain:

$$
(\mathbf{X}^{t'}+\Delta\mathbf{X}^{t'})(\Delta\mathbf{X}\mathbf{W}^t+\mathbf{X}^t\Delta\mathbf{W}+\Delta\mathbf{X}\Delta\mathbf{W}-\Delta\mathbf{U})+\alpha\Delta\mathbf{D}\Delta\mathbf{W}=0.
\tag{13}
$$

Since $\mathbf{X}^{t+1'}\mathbf{X}^{t+1}$ is positive semidefinite and $\Delta\mathbf{D}$ is a positive diagonal matrix, $\mathbf{X}^{t+1'}\mathbf{X}^{t+1}+\alpha\Delta\mathbf{D}$ is also positive semidefinite. Hence, $\Delta\mathbf{W}$ has a closed-form solution:

$$
\Delta\mathbf{W}=(\mathbf{X}^{t+1'}\mathbf{X}^{t+1}+\alpha\Delta\mathbf{D})^{-1}\mathbf{X}^{t+1'}(\Delta\mathbf{U}-\Delta\mathbf{X}\mathbf{W}),
\tag{14}
$$

which completes the proof. ∎

2) Given $\Delta\mathbf{W}$, Update $\Delta\mathbf{U}$: Then we substitute the closed form solution of $\Delta\mathbf{W}$ into Eq. (10) and obtain:

$$
\begin{aligned}
\min_{\Delta\mathbf{U}+\mathbf{U}^t\geq 0}&Tr(\Delta\mathbf{U}'\Delta\mathbf{U})+Tr(\mathbf{W}^{t'}\Delta\mathbf{X}'\Delta\mathbf{X}\mathbf{W}^t)\\
&-2Tr(\mathbf{W}^{t'}\Delta\mathbf{X}'\Delta\mathbf{U})-Tr(\Delta\mathbf{W}'\mathbf{H}\Delta\mathbf{W})\\
&+\frac{\beta}{2}||\Delta\mathbf{A}-\Delta\mathbf{U}\mathbf{U}^{t'}-\mathbf{U}^t\Delta\mathbf{U}'-\Delta\mathbf{U}\Delta\mathbf{U}'||_F^2\\
&=Tr((\Delta\mathbf{U}-\Delta\mathbf{X}\mathbf{W}^t)'(\Delta\mathbf{U}-\Delta\mathbf{X}\mathbf{W}^t))\\
&-Tr((\Delta\mathbf{U}-\Delta\mathbf{X}\mathbf{W}^t)'\mathbf{X}^{t+1}\mathbf{H}^{-1}\mathbf{X}^{t+1'}(\Delta\mathbf{U}-\Delta\mathbf{X}\mathbf{W}^t))\\
&+\frac{\beta}{2}||\Delta\mathbf{A}-\Delta\mathbf{U}\mathbf{U}^{t'}-\mathbf{U}^t\Delta\mathbf{U}'-\Delta\mathbf{U}\Delta\mathbf{U}'||_F^2,
\end{aligned}
\tag{15}
$$

where $\mathbf{H}=\mathbf{X}^{t+1'}\mathbf{X}^{t+1}+\alpha\Delta\mathbf{D}$. The above problem is a box constrained optimization problem, and we propose to employ projected gradient descent method [11] to solve it.

With these, the detailed time-evolving feature selection framework TeFS that adaptively perform feature selection at each time step is illustrated in Algorithm 1. We first obtain $\mathbf{W}^t$ and $\mathbf{U}^t$ at the first time step t by NetFS. Then at following time steps, given the variation of feature information $\Delta\mathbf{X}$ and the variation of adjacency matrix $\Delta\mathbf{A}$ between consecutive time steps, we fix $\Delta\mathbf{W}$ to update $\Delta\mathbf{U}$ and fix $\Delta\mathbf{U}$ to update $\Delta\mathbf{W}$ iteratively until the objective function value in Eq. (10) converges. At each time step, we sort the features in a descending order according to the new transformation matrix $\mathbf{W}^t+\Delta\mathbf{W}$ and return top-m ranked features. Typically, the larger the value $\|(\mathbf{W}^t+\Delta\mathbf{W})(i,:)\|_2$ is, the more important the i-th feature is.

Algorithm 1 TeFS framework

Input: feature information $\{\mathbf{X}^t,\mathbf{X}^{t+1},...,\mathbf{X}^{t+i}\}$ and adjacency matrix $\{\mathbf{A}^t,\mathbf{A}^{t+1},...,\mathbf{A}^{t+i}\}$ between the time period $\{t,t+1,...,t+i\}$, parameters α, β.
Output: m most relevant features at each time step.

```
Obtain W^t and U^t at time step t by NetFS;
Sort features according to W^t(i,:) in a descending order and
    select the top-m ranked ones;
for j = 1, 2, ..., i do
    Obtain the perturbations ΔX and ΔA;
    Set k = 0 and initialize ΔD_k;
    while objective function in Eq. (10) not converge do
        Obtain ΔU_{k+1} by projected gradient descent;
        Obtain ΔW_{k+1} by Eq. (14);
        k = k + 1;
    end while
    W^{t+j} = W^{t+j-1} + ΔW;
    U^{t+j} = U^{t+j-1} + ΔU;
    Sort features according to W^{t+j}(i,:) in a descending order
        and select the top-m ranked ones
end for
return m most relevant features at each time step.
```

	BlogCatalog	Flickr	Epinions
# of Users	5,196	7,575	5,665
# of Features	8,189	12,047	10,382
# of Classes	6	9	24
# of Time Steps	20	20	17

TABLE I: Detailed information of the datasets.

D. TeFS Time Complexity Analysis

In this subsection, we analyze the time complexity of the proposed TeFS framework. In each iteration, we first fix $\Delta\mathbf{U}$ to update $\Delta\mathbf{W}$ by Eq. (14). Instead of computing the inverse of matrix $\mathbf{X}^{t+1'}\mathbf{X}^{t+1} + \alpha\Delta\mathbf{D}$ directly which needs $\mathcal{O}(d^3)$ operations, we obtain $(\mathbf{X}^{t+1'}\mathbf{X}^{t+1}+\alpha\Delta\mathbf{D})^{-1}\mathbf{X}^{t+1'}$ by solving a linear equation system which only requires $\mathcal{O}(nd^2)$. The term of $\Delta\mathbf{X}\mathbf{W}$ can be pre-computed before iteration to improve the computational efficiency. Assume that the total number of nonzero elements in $\Delta\mathbf{U} - \Delta\mathbf{X}\mathbf{W}$ and $\mathbf{U}^t$ are $\tilde{a}$ and a, respectively. Then the number of operations to obtain $\Delta\mathbf{W}$ is $\mathcal{O}(nd^2 + \tilde{a}d)$. Otherwise, the cost is $\mathcal{O}(nd^2 + ad)$ if we rerun NetFS. Next, we fix $\Delta\mathbf{W}$ to update $\Delta\mathbf{U}$, the major computational cost involves in the computation of the gradient for the projected gradient method. Suppose the total number of nonzero elements in $\Delta\mathbf{A}$ and $\mathbf{A}^t$ are $\tilde{b}$ and b, respectively. The computation cost of gradient of $\mathcal{J}(\Delta\mathbf{W}, \Delta\mathbf{U})$ w.r.t. $\Delta\mathbf{U}$ is $\mathcal{O}(nc^2 + \tilde{b}c + n^2d + n\tilde{a})$. On the other hand, if we do not choose the incremental method, the cost of obtaining gradient is $\mathcal{O}(nc^2+bc+n^2d+na)$. In practice, since $\tilde{a} < a$ and $\tilde{b} \ll b$, TeFS is more efficient than its static counterpart.

IV. Experiments

We perform an empirical study to assess the performance of the proposed TeFS framework. In particular, we attempt to answer the following two questions: (1) How is the quality of selected features by TeFS? (2) How is the efficiency of the proposed TeFS compared with its batch-mode version that reruns each time step?

A. Datasets

We collect three real-world networks BlogCatalog, Flickr, and Epinions for the evaluation purpose. The BlogCatalog and Flickr datasets are already used in [18]. To simulate their evolving process, we randomly disturb 0.1% edges and feature values on 20 time steps. Epinions[1] is a product review site in which users can share their reviews about products. Users themselves can also build trust networks to seek advice from others. Features come from the bag-of-words model of user reviews, while the major categories of reviews by users are taken as the ground truth of class labels. In Epinions, since the time information when users build trust relationships is available, we crawled and collected the site into 17 different time steps to form a dynamic dataset. Detailed statistics of these three datasets are shown in Table I.

[1]http://epinions.com/

metric	ACC (%)				
# of features	200	400	600	800	1000
LapScore	26.83	28.41	27.34	24.35	31.87
SPEC	18.34	18.01	18.65	19.32	21.01
NDFS	24.12	30.82	32.56	31.78	34.35
LUFS	21.30	21.89	31.65	32.01	32.36
NetFS	49.99	43.04	43.25	42.89	42.09
TeFS	50.13	42.70	42.46	42.96	43.35
metric	NMI				
# of features	200	400	600	800	1000
LapScore	0.0821	0.0754	0.0586	0.0454	0.0793
SPEC	0.0021	0.0019	0.0054	0.0068	0.0081
NDFS	0.1019	0.1650	0.1867	0.1368	0.1705
LUFS	0.0421	0.0405	0.1419	0.1498	0.1611
NetFS	0.3321	0.2345	0.2404	0.2318	0.2383
TeFS	0.3472	0.2403	0.2289	0.2348	0.2525

TABLE II: Clustering results comparison on BlogCatalog.

B. Baseline Methods and Experimental Settings

It should be mentioned that to the best of our knowledge, the proposed TeFS framework is the first attempt to perform time-evolving feature selection on dynamic networks. We compare it with several state-of-the-art unsupervised feature selection algorithms that work in a static setting: (1) LapScore [5] - it evaluates feature importance via its ability to preserve locality property; (2) SPEC [22] - it ranks features by spectral analysis; (3) NDFS [10] - it selects features by a joint nonnegative spectral analysis and a $\ell_{2,1}$-norm regularization; (4) LUFS [15] - it performs feature selection for networked data by integrating the idea of social dimension and linear discriminative analysis; (5) NetFS [9] - it is a variant of TeFS that reruns each time step without incremental update.

First, to evaluate the quality of selected features, we follow a standard evaluation practice by the clustering performance w.r.t. selected features. Two clustering evaluation metrics, i.e., *normalized mutual information* (NMI) and *clustering accuracy* (ACC) are used. In particular, each feature selection algorithm is first applied to select features; then we apply a standard clustering method - K-means to perform clustering based on the selected features. Since K-means may converge to local minima, we repeat the process 20 times and report the average clustering performance. Second, to assess the efficiency of the proposed time-evolving feature selection framework, we record its cumulative running time over all time steps and compare it with its static counterpart NetFS.

In unsupervised scenarios, it is not easy to determine the optimal parameter settings of baseline methods. Therefore, we set their parameters according to the suggestions from the original paper. We find that the in TeFS, clustering performance is not sensitive to parameters α and β, and it is safe to tune them in a wide range. Empirically, we set α as 10 and β as 0.1.

C. Effectiveness of TeFS

We first investigate the effectiveness of the proposed time-evolving feature selection framework TeFS. We vary the number of selected features among $\{200, 400, 600, 800, 1000\}$ for BlogCatalog, Flickr and Epinions. We perform TeFS on all

metric	ACC (%)				
# of features	200	400	600	800	1000
LapScore	12.26	12.71	12.38	13.07	13.26
SPEC	12.14	12.42	13.18	13.53	14.61
NDFS	15.41	17.25	26.27	28.56	35.54
LUFS	11.89	19.24	20.08	22.56	23.24
NetFS	23.04	31.52	33.60	36.21	35.52
TeFS	23.14	31.40	34.28	35.49	35.28
metric	NMI				
# of features	200	400	600	800	1000
LapScore	0.0067	0.0072	0.0064	0.0155	0.0181
SPEC	0.0018	0.0054	0.0074	0.0155	0.0141
NDFS	0.0349	0.0582	0.0900	0.1072	0.1744
LUFS	0.0152	0.0798	0.0956	0.1321	0.1288
NetFS	0.1161	0.1655	0.2028	0.2056	0.2167
TeFS	0.1192	0.1589	0.2125	0.2089	0.2134

TABLE III: Clustering results comparison on Flickr.

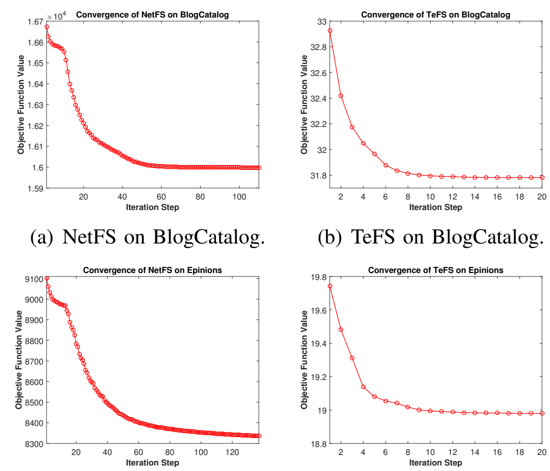
metric	ACC (%)				
# of features	200	400	600	800	1000
LapScore	15.29	13.45	13.08	12.81	11.47
SPEC	14.20	12.87	12.76	11.08	10.26
NDFS	12.81	11.82	12.41	12.40	12.63
LUFS	13.21	11.28	11.42	11.59	12.56
NetFS	14.04	16.66	18.27	20.48	20.46
TeFS	14.03	16.87	18.30	20.29	20.70
metric	NMI				
# of features	200	400	600	800	1000
LapScore	0.0142	0.0205	0.0228	0.0272	0.0208
SPEC	0.0180	0.0207	0.0238	0.0253	0.0258
NDFS	0.0229	0.0250	0.0292	0.0323	0.0383
LUFS	0.0232	0.0252	0.0240	0.0271	0.0333
NetFS	0.0387	0.0580	0.0691	0.0693	0.0892
TeFS	0.0456	0.0576	0.0655	0.0682	0.0899

TABLE IV: Clustering results comparison on Epinions.

these three dynamic networks, record the clustering results at each time step and report the average NMI and ACC. Since LapScore, SPEC, NDFS, LUFS and NetFS are not designed for dynamic networks, we run these baseline algorithms each time step to select relevant features and also report the average clustering results. The comparison results are shown in Table II, Table III and Table IV. The higher the ACC and NMI values are, the better the quality of selected feature is. As can be observed, most of the time, TeFS and its static version NetFS both outperform all other baseline methods LapScore, SPEC, NDFS and LUFS. To further validate the conclusions, we also perform one-tailed t-test between TeFS, NetFS, and other baseline methods and the test results show that TeFS and NetFS are significantly better (with a 0.05 significance level). The reason is that traditional feature selection methods (LapScore, SPEC and NDFS) fail to take advantage of linkage structure on networks. In addition, the superiority of TeFS and NetFS over LUFS can be attributed to the unified framework which joint models information from different modalities. On the contrary, LUFS separates the network modeling and feature selection, the performance of feature selection heavily depends on the obtained social dimensions. We can also find that TeFS achieves comparable clustering performance as NetFS. It indicates that even though we adopt an approximation method to reduce computational cost, it does not bring any negative effects by jeopardizing the clustering performance.

D. Efficiency

In TeFS, we try to minimize the upper bound of the objective function each time variations occur. In this way, we can only focus on optimizing the variation of model parameters $\Delta\mathbf{W}$ and $\Delta\mathbf{U}$. In this subsection, we first investigate the convergence rate of TeFS and its batch-mode version NetFS. Fig. 2 shows the convergence comparison results on BlogCatalog and Epinions datasets on a specific time step. It can be clearly seen from the figure that the objective function value of TeFS decreases and reaches a stable state much more quickly than its batch-mode version NetFS. TeFS converges within 20 iterations while NetFS needs more than 100 iterations to converge. This observation suggests that



(a) NetFS on BlogCatalog. (b) TeFS on BlogCatalog.

(c) NetFS on Epinions. (d) TeFS on Epininons.

Fig. 2: Convergence rate comparison.

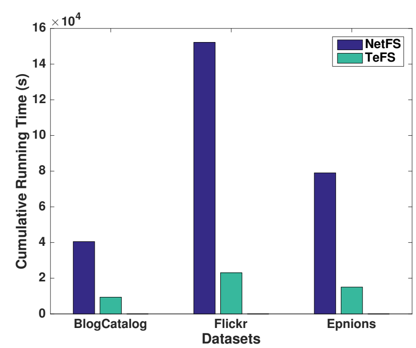


Fig. 3: Cumulative running time comparison.

TeFS has a faster convergence rate. Also, we compare their cumulative running time on all these three dynamic datasets. The cumulative running time comparison results are shown in Fig. 3, the results show that the proposed TeFS is significantly more efficient than NetFS, the speed up is $4.3\times$, $6.6\times$ and $5.3\times$ on BlogCatalog, Flickr, and Epinions, respectively.

V. Related Work

In this section, we briefly review related work on feature selection for networked data. Traditional feature selection assumes that data instances are independent and identically distributed (i.i.d.). However, it is not the case on networks where data instances are inherently interconnected by links. With the prevalence of networked data in many domains, feature selection on networks has gained increasing attention. In [3], a supervised feature selection algorithm FSNet is proposed. FSNet captures the correlation between feature information and network structure via graph regularization. Tang and Liu [14] extended to consider various social relationships and investigated how these relations can be leveraged in finding relevant features. In [19], the authors proposed a partial order network relation to guide feature selection. However, they not explicitly use content information. LUFS [15] uses both content information and network structure to perform feature selection. In particular, they exploit social dimensions derived from link information to guide feature selection in the content space. A critical issue of LUFS is that it is sensitive to noisy links. Hence, Li et al. [9] proposed a robust unsupervised feature selection method to model user latent representation and feature selection in a joint framework. As real-world networked data is often not static but constantly evolving over time. Feature selection for dynamic and streaming data has received increasing attention recently. In [8], the authors argued that features could be presented in a streaming fashion in social media platforms. Therefore, they proposed a USFS framework to dynamically maintain a subset of relevant features for feature streams. Guo et al. [4] proposed a node classification algorithm for streaming network. They take the dynamic changes of network and drifts of node contents into account to find an optimal set of features for node classification. Our work differs from [4], [8] in that their methods either use node labels to guide feature selection or do not consider the evolution of network structures, while our work performs feature selection dynamically in an unsupervised scenario when both sources of information evolve over time. Therefore, it is more challenging.

VI. Conclusion and Future Work

In this paper, we propose a time-evolving unsupervised feature selection framework TeFS for dynamic networks. The proposed framework takes both the content drift and the network structure changes into account to find relevant features adaptively. To capture the interactions and correlations when both feature and network structure change, we propose to model feature and network information jointly. In particular, based on the temporal smoothness property of dynamic networks, a time-evolving method TeFS is proposed. Typically, instead of rerunning the static model from scratch each time step when variations happen, we propose an efficient way to update the feature selection results from previous time steps incrementally. Also, we propose a powerful optimization framework to solve the optimization problem of TeFS. Experimental results on real-world dynamic networks validate the effectiveness and efficiency of the proposed time-evolving feature selection framework TeFS.

Future research can be focused on two directions. First, in this work, we only consider homogeneous network structures. Real-world networks are often presented in a multi-dimensional and multi-mode way. We would investigate how to select relevant features on these networks. Second, in addition to irrelevant, redundant and noisy features, there are many noisy instances in the network, it is interesting to investigate how to perform feature selection and outlier detection simultaneously for dynamic networks.

Acknowledgements

This material is, in part, supported by National Science Foundation (NSF) under grant number IIS-1614576.

References

[1] Charu Aggarwal and Karthik Subbian. Evolutionary network analysis: A survey. *CSUR*, 47(1):10, 2014.

[2] Chen Chen and Hanghang Tong. Fast eigen-functions tracking on dynamic graphs. In *SDM*, 2015.

[3] Quanquan Gu and Jiawei Han. Towards feature selection in network. In *CIKM*, 2011.

[4] Ting Guo, Xingquan Zhu, Jian Pei, and Chengqi Zhang. Snoc: streaming network node classification. In *ICDM*, 2014.

[5] Xiaofei He, Deng Cai, and Partha Niyogi. Laplacian score for feature selection. In *NIPS*, 2005.

[6] Timothy La Fond and Jennifer Neville. Randomization tests for distinguishing social influence and homophily effects. In *WWW*, 2010.

[7] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *arXiv preprint arXiv:1601.07996*, 2016.

[8] Jundong Li, Xia Hu, Jiliang Tang, and Huan Liu. Unsupervised streaming feature selection in social media. In *CIKM*, 2015.

[9] Jundong Li, Xia Hu, Liang Wu, and Huan Liu. Robust unsupervised feature selection on networked data. In *SDM*, 2016.

[10] Zechao Li, Yi Yang, Jing Liu, Xiaofang Zhou, and Hanqing Lu. Unsupervised feature selection using nonnegative spectral analysis. In *AAAI*, 2012.

[11] Chih-Jen Lin. Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10):2756–2779, 2007.

[12] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, pages 415–444, 2001.

[13] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93, 2008.

[14] Jiliang Tang and Huan Liu. Feature selection with linked data in social media. In *SDM*, 2012.

[15] Jiliang Tang and Huan Liu. Unsupervised feature selection for linked social media data. In *KDD*, 2012.

[16] Hanghang Tong, Spiros Papadimitriou, Jimeng Sun, Philip S Yu, and Christos Faloutsos. Colibri: fast mining of large static and dynamic graphs. In *KDD*, 2008.

[17] Xin Wang, Roger Donaldson, Christopher Nell, Peter Gorniak, Martin Ester, and Jiajun Bu. Recommending groups to users using user-group engagement and time-dependent matrix factorization. In *AAAI*, 2016.

[18] Xufei Wang, Lei Tang, Huiji Gao, and Huan Liu. Discovering overlapping groups in social media. In *ICDM*, 2010.

[19] Xiaokai Wei, Sihong Xie, and S Yu Philip. Efficient partial order preserving unsupervised feature selection on networks. In *SDM*, 2015.

[20] James D Westaby. *Dynamic network theory: How social networks influence goal pursuit.* American Psychological Association, 2012.

[21] Jaewon Yang, Julian McAuley, and Jure Leskovec. Community detection in networks with node attributes. In *ICDM*, 2013.

[22] Zheng Zhao and Huan Liu. Spectral feature selection for supervised and unsupervised learning. In *ICML*, 2007.

[23] Dawei Zhou, Kangyang Wang, Nan Cao, and Jingrui He. Rare category detection on time-evolving graphs. In *ICDM*, 2015.