Efficient Partial Order Preserving Unsupervised Feature Selection on Networks

Xiaokai Wei* Sihong Xie* Philip S. Yu *†

Abstract

In the past decade, research on network data has attracted much attention and many interesting phenomena have been discovered. Such data are often characterized by high dimensionality but how to select meaningful and more succinct features for network data received relatively less attention. In this paper, we investigate unsupervised feature selection problem on networks. To effectively incorporate linkage information, we propose a Partial Order Preserving (POP) principle for evaluating features. We show the advantage of this novel formulation in several respects: effectiveness, efficiency and its connection to optimizing AUC. We propose three instantiations derived from the POP principle and evaluate them using three real-world datasets. Experimental results show that our approach has significantly better performance than state-of-the-art methods under several different metrics.

1 Introduction

In many machine learning tasks, one is often confronted with the problem of high dimensionality. Hence, feature selection [1] [2] has become an important technique since it can help alleviate the curse of dimensionality and speed up the learning process. Depending on the availability of class labels, feature selection algorithms can be classified into supervised methods and unsupervised methods. Our work focuses on unsupervised scenario as class labels are usually expensive to obtain. A variety of approaches has been developed for unsupervised feature selection by following different principles. In recent work, similarity-preserving approaches [1] [3] and regression based approaches using pseudo labels [4] [5] have gained much popularity among others.

Network data has become increasingly popular in the past decade, because of the proliferation of various social and information networks. Social media websites such as Facebook, Twitter have millions of users all across the world. Different forms of information networks, e.g, co-author network, citation network and protein interaction network, also attract considerable attention to analyze [6] [7].

However, traditional feature selection approaches assume that instances are independent and identically distributed (i.i.d). In relational data or information networks, the instances are implicitly or explicitly related, with certain correlation and dependency. For example, in research collaboration networks, the researchers who collaborate with each other tend to share more similar research topics than researchers with no collaboration. But traditional approaches are not able to exploit such rich information contained in the links. LUFS [8] is the first attempt to incorporate network information for unsupervised feature selection, but it uses the structural information at community level via social dimensions [9] and fails to exploit finer-grained link information. Also, LUFS requires several parameters, which are hard to tune in unsupervised setting.

Moreover, the ever increasing size of network data poses additional challenges to feature selection. For instance, Facebook and Linkedin have more than 1.28 billion[1] and 300 million[2] users as of 2014, respectively. However, state-of-the-art unsupervised feature selection methods [4] [5] [8] are prohibitively slow, as their time complexity is usually cubic of the number of features or instances. This makes these algorithms unpractical for large-scale and high-dimensional data.

In this paper, we present a new perspective to address these challenges regarding both effectiveness and efficiency. We propose a Partial Order Preserving (POP) framework, which allows for parameter-free mathematical formulation and efficient optimization. Rather than simply preserving the similarity or local manifold structure, POP aims to preserve the partial order of similarity. Network data have abundant partial order information: a node is usually more similar to its neighbors than to the other nodes. By exploiting such difference for feature selection, structural information distinguishing neighbors from non-neighbors is incorporated. As a consequence, more discriminative features can be selected. The main contribution of our work can be summarized in the following:

- We propose a new principle for feature selection on

*Department of Computer Science, University of Illinois at Chicago, {xwei2, sxie6, psyu}@uic.edu

†Institute for Data Science, Tsinghua University, Beijing, China

[1] http://en.wikipedia.org/wiki/Facebook

[2] http://en.wikipedia.org/wiki/Linkedin

networks: Partial Order Preserving (POP) principle, which selects features that best preserve partial orders. As state-of-the-art approaches are mostly pseudo-label based methods using $L_{2,1}$ norm [4] [5] [8], POP brings a new perspective to the problem of unsupervised feature selection.

- As the linkage relationship in the network is neither complete nor noise free, we present three instantiations of the POP principle, which are robust to noisy/incomplete link information and are parameter free in the objective functions.
- We develop a highly efficient and unified optimization algorithm for these three instantiations. This makes our methods applicable to large-scale datasets.
- We evaluate the proposed algorithms on three real world datasets, and show the advantage of our approach over the baseline methods using different metrics.

2 Related Work

In this section, we briefly review related work on feature selection (mainly on unsupervised feature selection).

2.1 Unsupervised Feature Selection for Traditional Data In the unsupervised setting, there are various principles to guide the feature selection process. One popular guiding principle is to preserve the local manifold structure or similarity [1] [10] [3]. Recently, pseudo label-based framework [4] [5] gained much popularity. Unsupervised Discriminative Feature Selection (UDFS) [4] introduces pseudo labels to better capture discriminative information and sparsity-inducing $L_{2,1}$ norm is used to select the feature in an iterative manner. Non-negative Discriminative Feature Selection (NDFS) [5] performs non-negative spectral analysis and feature selection simultaneously. But both UDFS and NDFS have computation complexity of $O(D^3T+n^2)$ (D is the number of features, T is the number of iterations, n is number of instances) as eigen-decomposition on $D \times D$ matrix is performed in each iteration. This severely refrains them from being applied to high dimensional data such as text or microarray data. Moreover, they have 3 $\sim$ 4 parameters to be specified in the objective function. In supervised learning, appropriate parameters can be found through grid search but in unsupervised setting, there is no straightforward way to tune the parameters.

2.2 Feature Selection for Network Data Traditional feature selection techniques assume data instances are independent and identically distributed (i.i.d), which is not the case in network data. In recent years, efforts have been made towards feature selection on relational data. [11] addresses supervised feature selection on network data via adding network-based regularization term to enforce similarity between neighbors. [12] explores supervised feature selection on social media data and integrates different types of relations into the feature selection framework. [13] studies co-selection of features and instances in social media since both features and instances can be noisy and irrelevant. [14] investigates unsupervised multi-view feature selection on social media but it does not utilize link information. Linked Unsupervised Feature Selection (LUFS) [8] is the only unsupervised feature selection method that utilizes link information. LUFS exploits network information through incorporating social dimension based regularization [9] into the UDFS framework [4]. So it shares the same downside of UDFS such as too many parameters and high computational cost. Also, in LUFS, network information is utilized at community/cluster level and finer-grained information in the links is ignored. In this paper, we propose a parameter-free framework for unsupervised feature selection on network data, which is more effective with lower computation burden.

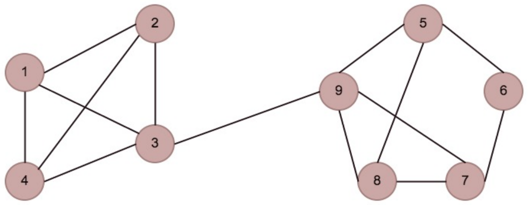


Figure 1: An example network with 9 nodes

3 Problem Formulation

3.1 Partial Order on Network In this section, we present several concepts as preliminaries of our Partial Order Preserving (POP) principle for feature selection. Our partial order is defined on an *information network*.

Definition 1. **Information Network** *An information network $G = (V, E, X)$ consists of V, the group of vertices, $E \subseteq V \times V$, the set of edges, and feature matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ ($i = 1 \ldots n$, $n = |V|$), where $\mathbf{x}_i \in \{0,1\}^D$ is the attribute vector of node v_i.*

In an information network, for each node v, the remaining nodes can be divided to two categories based on whether they are linked to v: *linked set* and *unlinked set*.

Definition 2. **Linked Set** *For a node $v \in V$, its linked set is defined as the set $\mathcal{L}(v)$ of all the nodes which* *are linked to* v*, i.e.,* $u \in \mathcal{L}(v) \Leftrightarrow (u, v) \in E$*.*

DEFINITION 3. **Unlinked Set** *For a node* $v \in V$*, its unlinked set is set of nodes* $\mathcal{U}(v)$ *which are not in the linked set of* v*, i.e.,* $\mathcal{U}(v) = V/\mathcal{L}(v)$

Traditional i.i.d assumption does not hold for data instances in networks because of the widely observed *homophily effect*. In recent years, many machine learning algorithms on networks try to exploit this fact: *friends are similar*. One popular technique is network based regularization [11] [15], which enforces neighbor nodes (i.e., nodes in linked set) to be similar.

But exploiting information solely from linked sets is not sufficient for feature selection. Though good features are likely to be shared by neighbors, not all features shared by neighbors are of high quality. For example, in citation network, neighbors (i.e., cited and citing paper) are usually of similar topic because of the homophily effect. As a result, they usually share some topical words (e.g. *SVM*, *LDA*). But indiscriminative words such as *propose* and *compare* are also shared by many neighbors. So we take one step further to exploit both the linked sets and unlinked sets: *friends are usually* **more** *similar than non-friends*. Good features should make neighbors look similar and non-neighbors not so similar. We formulate this idea as link-based partial order as follows.

DEFINITION 4. **Link-based Partial Order** *We formulate such property as partial order* $j >_i k$*, where node* v_j *and node* v_k *are in the linked set and unlinked set of node* v_i*, respectively. Node* v_i *is referred to as the pivot of this partial order. Such partial order is denoted as a triplet* (i, j, k) *or* $j >_i k$*.*

$$sim(v_i, v_j) > sim(v_i, v_k), v_j \in \mathcal{L}(v_i), v_k \in \mathcal{U}(v_i) \tag{3.1}$$

Let us take the network with 9 nodes in Figure 1 for example. The linked set $\mathcal{L}(v_3)$ of node v_3 is $\{v_1, v_2, v_4, v_9\}$, while its unlinked set $\mathcal{U}(v_3)$ is $\{v_5, v_6, v_7, v_8\}$. Generally speaking, $\{v_1, v_2, v_4, v_9\}$ should resemble v_3 more than $\{v_5, v_6, v_7, v_8\}$ to v_3. There are $4 \times 4 = 16$ partial order triplets (e.g., $(3, 1, 6)$, $(3, 1, 7)$, $(3, 2, 5)$) w.r.t *pivot* v_3.

This link-based partial order aims to capture the difference between linked set and unlinked set, i.e., what distinguishes linked set from unlinked set. The major difficulty of unsupervised feature selection comes from the lack of label, as the labels can provide clear guidance: features providing good separability of different classes are high-quality ones. In unsupervised scenario, we will show partial order can serve a similar purpose as class label. Features of good quality should be able to distinguish the linked set from the unlinked set, which is the intuition underlying our approach.

Table 1: Symbol definitions

Symbol	Definition
$\mathbf{x}_i \in \{0,1\}^D$	Feature vector of node v_i
$\mathcal{L}(v_i)$	Linked set of node v_i
$\mathcal{U}(v_i)$	Unlinked set of node v_i
s_{ij}	Similarity between node v_i and v_j after feature selection
s_{ijk}	Difference between s_{ij} and s_{ik}
$j >_i k$	Partial order triplet in which $v_j \in \mathcal{L}(v_i), v_k \in \mathcal{U}(v_i)$
(i, j, k)	Same as above
Ω	Set of all partial order triplets (i, j, k)
$l(j >_i k)$	The extent to which $j >_i k$ is preserved
$L(>)$	The extent to which all partial orders are preserved
$\mathbf{w} \in \{0,1\}^D$	Feature selection indicator vector

3.2 Partial Order Preserving Feature Selection (POPFS) Suppose the feature vector of node v_i is $\mathbf{x}_i \in \{0,1\}^D$ and our goal is to select d ($d < D$) features. Without loss of generality, we assume binary features since categorical or numerical features can be transformed to binary features (e.g., by binning). In order to do feature selection, we introduce an indicator vector $\mathbf{w} = (w_1, w_2, \ldots, w_D)^T$, $w_i \in \{0,1\}$ ($\forall i = 1, \ldots, D$). Then we construct a diagonal matrix $diag(\mathbf{w})$ from $\mathbf{w}$. Therefore, the data instance $\mathbf{x}_i$ after feature selection is $diag(\mathbf{w})\mathbf{x}_i$. A set of important symbols used in this paper are summarized in Table 1.

Based on the link-based partial order defined above, it is desirable that partial order is preserved after feature selection. This can be formulated as follows.

(3.2)
$$\mathrm{sim}(diag(\mathbf{w})\mathbf{x}_i, diag(\mathbf{w})\mathbf{x}_j) > \mathrm{sim}(diag(\mathbf{w})\mathbf{x}_i, diag(\mathbf{w})\mathbf{x}_k)$$

In principle, $sim(\cdot, \cdot)$ could be any similarity metric defined on the feature vector, such as Cosine Similarity. To make the optimization simple, we use inner product as the similarity measure. We denote $sim(diag(\mathbf{w})\mathbf{x}_i, diag(\mathbf{w})\mathbf{x}_j)$ as s_{ij}. Rather than the absolute values of s_{ij} and s_{ik}, we are more interested in their relative difference s_{ijk}.

$$\begin{aligned} s_{ijk} &= s_{ij} - s_{ik} \\ &= \mathbf{x}_i^T diag(\mathbf{w})\mathbf{x}_j - \mathbf{x}_i^T diag(\mathbf{w})\mathbf{x}_k \end{aligned} \tag{3.3}$$

We further define an objective function $l(j >_i k \mid \mathbf{w})$ over the partial order triplet (i, j, k) to quantify to what extent the partial order $j >_i k$ is preserved.

$$l(j >_i k \mid \mathbf{w}) = f(s_{ijk} \mid \mathbf{w}) \tag{3.4}$$

A monotonically non-decreasing link function f is used to connect $l(j >_i k)$ with s_{ijk}. When s_{ijk} is large, it means (i, j, k) is well preserved; when s_{ijk} is small (e.g., a negative value), it means (i, j, k) is poorly preserved. Different types of link function can be adopted, for example, identity function or sigmoid function.

However, similar nodes may not be always connected in networks. For example, in co-author network, *Jiawei Han* and *Christos Faloutsos* have not collaborated though they work on similar research topics. So we cannot expect every $(j >_i k)$ derived from the network to be preserved. But in an aggregate sense, a set of good features should make the partial order triplets derived from network structure minimally violated (i.e., maximally preserved). Let us denote the set of all the partial order triplets as Ω.

$$\Omega = \{(i,j,k) | i \in V, j \in \mathcal{L}_i, k \in \mathcal{U}_i\} \tag{3.5}$$

We are interested in preserving the aggregated partial order $L(>)$. This leads to maximizing $l(\cdot)$ over all triplets with constraint $\sum_{i=1}^{D} w_i = d$ where d is the number of selected features.

$$\begin{aligned} \max_{\mathbf{w}} L(>) &= \sum_{(i,j,k)\in\Omega} l(j >_i k \mid \mathbf{w}) \\ &= \sum_{i\in V} \sum_{j\in\mathcal{L}_i} \sum_{k\in\mathcal{U}_i} f(s_{ijk} \mid \mathbf{w}) \\ \text{s.t. } & w_i \in \{0,1\}, \ \sum_{i=1}^{D} w_i = d \end{aligned} \tag{3.6}$$

4 Instantiations of the POP Framework

In previous section, we introduce the unified framework for Partial Order Preserving Feature Selection (POPFS). In this section, we present three instantiations of the POP principle: Simple POP, Probabilistic POP and Max-Margin POP, which have different interpretations.

4.1 Simple POP (SPOP) For simplest case of link function, we can use identity function as f. It is easy to show that the optimization problem in Eq. (3.6) is equivalent to calculating the following score for each feature.

$$score(a) = \sum_{(i,j,k)\in\Omega} I(i,j,a) - \sum_{(i,j,k)\in\Omega} I(i,k,a) \tag{4.7}$$

where $I(i,j,a)$ is an indicator function, which equals 1 if both nodes i and j have feature a and equals 0 otherwise. The first part of the score is the number of neighbor pairs sharing this feature a, which we refer to as the *linked score* of feature a; the second part of the score is the number of non-neighbor pairs sharing feature a, referred to as *unlinked score.* The final score of each feature is the difference between linked score and unlinked score. After we calculate the score using Eq. (4.7), we can simply select the top d features with the highest scores. By using identity link function, it does not consider interaction among features and therefore each feature can be evaluated independently.

This decomposition reveals several useful properties about SPOP and provides better understanding of this principle. If a feature's final score is above zero, it means its linked score is larger than its unlinked score. This indicates that, statistically, this feature appear more often in linked nodes than in non-linked nodes. Consider for example a citation network with papers from several topics (e.g., Machine Learning, Database, System). A generic feature (e.g., stop word) will have both high linked score and unlinked score because of its indiscriminative presence in nodes. The final score will be low as a result. The domain-specific features (e.g., *SVM*, *classification*) tend to have high linked scores and relatively low unlinked scores. Hence, the domain-specific terms will be retained and generic terms will be discarded by the feature selection process. As a result, unsupervised learning tasks, such as clustering, will benefit from this.

Although real-world networks can provide rich link information for constructing partial orders, they are often noisy by nature. If a noisy link connects two dissimilar nodes by accident, it will have minimal impact on the score calculated by SPOP. For example, given node v_i, consider two nodes $v_j \in \mathcal{L}_i$ and $v_k \in \mathcal{U}_i$. Suppose both v_j and v_k are not similar to v_i but v_j appears in $\mathcal{L}_i$ as noise. For an indiscriminative feature a, v_j and v_k would have similar probability to have it. So, by expectation this will not increase $score(a)$ since $E[I(i,j,a) - I(i,k,a)] \approx 0$. If we only utilize *linked set* through preserving Graph Laplacian without using *unlinked set*, feature selection would be possibly misled by noisy links. This illustrates another strength of preserving partial order against preserving the absolute value of similarity.

4.2 Probabilistic POP(PPOP) Though SPOP is simple and intuitive, it evaluates features individually and hence fails to take into consideration the correlation between features. In this and the following section, we develop two instantiations which evaluate features jointly.

From a generative point of view, we assume all the partial orders are generated from the indicator vector $\mathbf{w} \in \{0,1\}^D$. More specifically, we model the probability of preserving partial order $j >_i k$ as

$$P(j >_i k \mid \mathbf{w}) = \sigma(s_{ijk}) \tag{4.8}$$

where $\sigma(x) = 1/(1+e^{-x})$ is the sigmoid function. The larger s_{ijk} is, the more likely partial order $j >_i k$ is preserved. By assuming the partial orders to be independent, the probability $P(> |w)$ of all the partial orders being respected given $\mathbf{w}$ is,

$$P(>|\mathbf{w}) = \prod_{(i,j,k)\in\Omega} P(j >_i k|\mathbf{w}) \tag{4.9}$$
$$= \prod_{(i,j,k)\in\Omega} \sigma(s_{ijk})$$

The goal is to find the feature indicator vector $\mathbf{w}$ which maximizes $P(> |\mathbf{w})$ (i.e., to preserve the aggregated partial orders with maximum probability). Learning this model can be performed by maximizing the log-likelihood,

$$\max_{\mathbf{w}} \quad \log P(> |\mathbf{w}) = \sum_{(i,j,k)\in\Omega} \log P(j >_i k|\mathbf{w})$$
$$= \sum_{(i,j,k)\in\Omega} \log \sigma(s_{ijk}) \tag{4.10}$$
$$\text{s.t. } w_i \in \{0,1\}, \ \sum_{i=1}^{D} w_i = d$$

It provides a probabilistic interpretation for the partial order preserving principle. The connection between Eq. (5.17) and Eq. (3.6) is easy to see: $\log\sigma(\cdot)$ is used as the link function.

4.3 Max Margin POP (MMPOP) Structured learning methods, such as Structural SVM [16], have gained substantial popularity in the past decade and are powerful for combinatorial optimization. Preserving partial order is to well separate the linked and unlinked sets for each given pivot, which fits well into structural learning framework as follows.

$$\min_{\mathbf{w}} \quad \frac{1}{2}\|\mathbf{w}\|^2 \tag{4.11}$$
$$\text{s.t.} \quad s_{ijk} \geq 1, \forall(i,j,k) \in \Omega$$

However, in real world networks, the linked set and unlinked set are not always linearly separable using $\mathbf{w}$, as in the *Jiawei Han/Christos Faloutsos* example. So, to address this issue, we add an slack variable μ_{ijk} to impose soft margin.

$$\min_{\mathbf{w}} \quad \sum_{(i,j,k)\in\Omega} \mu_{ijk}$$
$$\text{s.t.} \quad s_{ijk} \geq 1 - \mu_{ijk}, \forall(i,j,k) \in \Omega \tag{4.12}$$
$$w_i \in \{0,1\}, \ \sum_{i=1}^{D} w_i = d$$

To make clear its connection to the Eq. (3.6) in the general framework, we rewrite it as follows.

$$\max_{\mathbf{w}} \quad \sum_{(i,j,k)\in\Omega} -\max(0, 1 - s_{ijk})$$
$$\text{s.t.} \quad w_i \in \{0,1\}, \ \sum_{i=1}^{D} w_i = d \tag{4.13}$$

So, Eq. (5.18) is equivalent to using negative hinge loss as link function in Eq. (3.6).

4.4 Connection to AUC Optimization To further justify using the POP principle for feature selection, we show how it is related to optimizing AUC. AUC (Area Under ROC Curve) is a widely used metric for evaluating binary prediction problem such as recommender system and link prediction. Optimizing the objective based on POP optimizes the AUC for link prediction.

$$AUC(v_i) = \frac{1}{|\mathcal{L}_i||\mathcal{U}_i|} \sum_{j\in\mathcal{L}_i} \sum_{k\in\mathcal{U}_i} I(s_{ijk} > 0) \tag{4.14}$$

where indicator function $I(\cdot)$ returns 1 if $s_{ijk} > 0$ and 0 otherwise .

$$AUC = \frac{1}{|V|} \cdot \sum_{i\in V} AUC(v_i) \tag{4.15}$$
$$= \frac{1}{Z} \sum_{(i,j,k)\in\Omega} I(s_{ijk} > 0)$$

where $Z = |\mathcal{L}_i||\mathcal{U}_i||V|$ is a normalizing constant. Comparing the objective function in Eq. (3.6) with Eq. (4.15), it is obvious to observe the connection with AUC optimization. AUC uses a non-continuous indicator function $I(\cdot)$ as the loss function, while PPOP and MMPOP use continuous loss function (logistic loss and hinge loss, respectively) to approximate the non-continuous counterpart.

Features selected by methods following different principles tend to have different properties. From the analogy between POP and AUC, we know that features selected by POP based methods are optimal in terms of preserving the network structure. This implies that POP-based feature selection methods can be particularly useful for link prediction task.

5 Optimization

For Simple POP (SPOP), one only needs to calculate linked score and unlinked score and rank features by their final scores. Optimization for MMPOP and PPOP is a mixed $0-1$ integer programming problem, which is NP-hard in general. To make optimization tractable, we relax the "0/1" constraint in the integer programming problem by replacing $w_i \in \{0,1\}$ with $w_i \in \mathcal{R}$. Such real-valued weights can be intuitively interpreted as features' *Importance Score*. Then we can rank the features by their importance scores in $\mathbf{w}$ and output the top d features. A challenge for all POP instantiations is that, there are a large number of potential partial order combinations ($O(n|E|)$). It would be very inefficient to iterate through all these $O(n|E|)$ partial order triplets. So we propose to use a bootstrap sampling based technique, *Stochastic (Sub)Gradient Descent*, to solve the optimization problem. In addition to efficiency, sampling based technique is also more robust to noise and outliers.

The objective functions of all three instantiations are convex since they use convex link function f. Since the link functions in SPOP and PPOP are both differentiable, the optimization problem can be efficiently solved by Stochastic Gradient Descent (SGD) method. But MMPOP uses hinge loss which is not differentiable. To solve the optimization problem of MMPOP, we can calculate subgradient and employ Stochastic Subgradient Descent. Hence, all three instantiations can be solved using a unified framework, which is presented in Algorithm 1. In each iteration, we sample a triplet (i,j,k), calculate the (sub)gradient and update $\mathbf{w}$.

Simple POP has the simplest form of gradient.

$$\frac{\partial l(j >_i k)}{\partial \mathbf{w}} = \frac{\partial}{\partial \mathbf{w}} s_{ijk} \tag{5.16}$$

For probabilistic POP (PPOP), the gradient for one sample is calculated as follows:

$$\frac{\partial l(j >_i k)}{\partial \mathbf{w}} = \frac{\partial}{\partial \mathbf{w}} f(s_{ijk}) \quad = \frac{e^{-s_{ijk}}}{1+e^{-s_{ijk}}} \cdot \frac{\partial}{\partial \mathbf{w}} s_{ijk} \tag{5.17}$$

For Max Margin POPFS (MMPOP), we calculate the subgradient and only update the weight vector when $1 - s_{ijk} > 0$:

$$\frac{\partial l(j >_i k)}{\partial \mathbf{w}} = \begin{cases} \frac{\partial}{\partial \mathbf{w}} s_{ijk} & \text{if } s_{ijk} < 1 \\ 0 & \text{otherwise} \end{cases} \tag{5.18}$$

For these three approaches,

$$\frac{\partial}{\partial w_p} s_{ijk} = \begin{cases} 1 & \text{if } x_{ip} = 1 \ \& \ x_{jp} = 1 \ \& \ x_{kp} = 0 \\ -1 & \text{if } x_{ip} = 1 \ \& \ x_{jp} = 0 \ \& \ x_{kp} = 1 \\ 0 & \text{otherwise} \end{cases} \tag{5.19}$$

where x_{ip} is the p-th feature in x_i. From the gradient formula of three approaches, one can observe that the gradient on the p-th feature in SPOP is not influenced by other features. In PPOP and MMPOP, the gradient is impacted by s_{ijk}: when s_{ijk} is large, the gradient is a small value ($e^{-s_{ijk}}/(1+e^{-s_{ijk}})$) in PPOP or 0 in MMPOP. Such updating scheme addresses the redundancy issue in feature selection.

Algorithm 1 Stochastic (sub)gradient descent algorithm for POP

```
$\mathbf{w} \leftarrow [0, 0, \ldots, 0]$
for ($t$ in $1..T$) do
    step size $\eta_t \leftarrow \frac{1}{\lambda t}$
    update $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta_t * \Delta_t$, using corresponding formula (5.16), (5.17) or (5.18) for $\Delta_t$
end for
Sort features w.r.t. $w[i]$ and output the top $d$ features
```

The optimization error can be bounded as shown in the following theorem.

Theorem 5.1. *Assume that the data is bounded such that $max_i \ x_i^T diag(w) x_i < R$ and $R \geq 1$. In algorithm 1 at iteration T, with $\lambda \leq \frac{1}{4}$, and batch-size $B = 1$, $\bar{\mathbf{w}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{w}_T$ be the average $\mathbf{w}$ by iteration T. Then, with probability of at least $1 - \delta$,*

$$f(\bar{\mathbf{w}}) - minf(\mathbf{w}^*) \leq \frac{21R^2 ln(T/\delta)}{\lambda T}. \tag{5.20}$$

Proof Sketch: Algorithm 1 is an instance of PEGASOS without a projection step on one-class data. Corollary 2 in [17] proves the same bound for traditional SVM input (without a projection step).

In each iteration, it takes $O(m)$ time to update $\mathbf{w}$, where m is the average number of non-zero features in each data point. This effectively exploits the fact that, in many datasets, m is often small though D can be large. If we sample T triplets of (i,j,k), the overall time complexity is $O(mT)$. Since our goal is feature selection, only the rank of weights w_i is needed. It means $\mathbf{w}$ does not need to be too precise (i.e., δ does not need to be very small). By employing SGD algorithm in Algorithm 1, SPOP, PPOP and MMPOP can be efficiently solved for large-scale networks. In addition, SGD can be updated in an online fashion. This is very useful since new nodes continuously join real-world networks.

6 Experiment

In this section, we conduct systematic experiments on three publicly available datasets. We compare our POP methods with four baselines on both efficiency and effectiveness. To illustrate how POP methods differ from existing mechanisms, we evaluate the selected features on both clustering task and link prediction task.

Table 2: Statistics of three datasets

Statistics	Citeseer	Cora	Wiki
# of instances	3312	2708	3363
# of links	4598	5429	33219
# of features	3703	1433	4973
avg. # of non-zero features per instance	31.75	18.17	630.57
# of classes	6	7	19

Experimental results show that POP can select well-rounded features which achieve top performance in both tasks.

6.1 Datasets We use three publicly available network datasets: Citeseer dataset, Cora Dataset and Wikipedia dataset [3] [18]. The statistics of three datasets are summarized in Table 2.

6.2 Baselines We compared our approach to the following baseline methods.

- All Features.
- Link Only: Spectral clustering using network links.
- Laplacian Score (LS): Laplacian score [1] selects the features which can best preserve the local manifold structure.
- UDFS: Unsupervised Discriminative Feature Selection [4] is a state-of-the-art pseudo-label based approach for i.i.d data. Unlike Laplacian score, UDFS selects features jointly rather than individually.
- LUFS: Linked Unsupervised Feature Selection is a state-of-the-art unsupervised feature selection method [8] designed for linked social media data, which combines the idea of social dimension [9] with UDFS.

6.3 Efficiency In this section, we investigate the efficiency of POP Feature Selection (POPFS) and the baseline approaches. Baseline methods UDFS and LUFS rely on an iterative method to converge to a local optima. In each iteration, it heavily involves matrix computation and therefore is very inefficient even for a medium-sized ($1000 \sim 10000$) feature set. POPFS has a convex formulation and can be optimized by Stochastic Gradient Decent (SGD). In practice, sampling a small portion of partial order triplets is usually enough. In our experiment, we find sampling $|E| \sim 2|E|$ triplets ($|E|$ is the number of edges) is sufficient for good performance.

Table 3: Running time (seconds) of different feature selection algorithms

Dataset	LS	UDFS	LUFS	SPOP	PPOP	MMPOP
Citeseer	10	1234	1420	1	2	2
Cora	5	161	113	1	1	1
Wiki	23	2536	2788	19	22	19

Table 3 reports the running time of different feature selection algorithms. POPFS requires much less running time than baseline methods (especially UDFS and LUFS). For example, on Citeseer dataset, UDFS takes nearly 20 minutes to converge, while POPFS only needs 1 or 2 seconds. The running time of LS is relatively close to POPFS but it only evaluates features individually. Real world social networks (e.g. Facebook and Linkedin) or information networks (e.g., DBLP and biological network) have ever increasing sizes in terms of both number of instances and number of features. Our SGD-based approach can significantly reduce computation time without trading off too much effectiveness.

6.4 Results on Clustering In this section, we evaluate the quality of selected features by their clustering performance. Following the typical setting [4] [8] of evaluation for unsupervised feature selection, we use Accuracy and Normalized Mutual Information (NMI) to evaluate the result of clustering. Accuracy is measured as follows.

$$Accuracy = \frac{1}{n}\sum_{i=1}^{n}\mathcal{I}(c_i = map(p_i)) \tag{6.21}$$

where p_i is the clustering result of data point i and c_i is its ground truth label. $map(\cdot)$ is a permutation mapping function that maps p_i to a class label using Kuhn-Munkres Algorithm.

Normalized Mutual Information (NMI) is calculated as follows. Let C be the set of clusters from the ground truth and C' is obtained from a clustering algorithm.

$$NMI(C, C') = \frac{MI(C, C')}{max(H(C), H(C'))} \tag{6.22}$$

where $H(C)$ and $H(C')$ are the entropy of C and C' and $MI(C, C')$ is the mutual information. Higher value of NMI indicates better quality of clustering.

We use the default parameter setting suggested in the original papers for the baseline methods. For the number of pseudo classes in UDFS and LUFS, we use the ground-truth number of classes. As in previous work

[3] For detailed information about the datasets, one can refer to `http://linqs.cs.umd.edu/projects//projects/lbc/index.html`

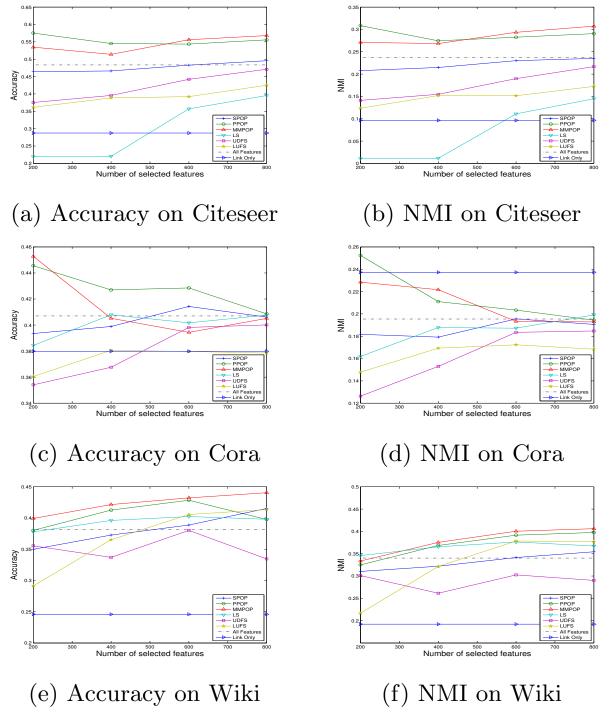
[4] [8], we use K-means[4] for evaluation. Since Kmeans is affected by the initial seeds, we repeat the experiment for 20 times and report the average performance. We vary the number of features from 200 to 800, with an increment of 200. The KMeans clustering performance for three datasets is shown in Figure 2.

Among three POP instantiations, MMPOP and PPOP have better clustering performance than SPOP. This demonstrates the importance of evaluating features in a joint manner. SPOP does not take into consideration correlation between features and the redundancy in selected features makes the clustering result suboptimal. With only 200 features, MMPOP and PPOP can obtain much better accuracy and NMI than using all the features. For instance, compared with using all features, MMPOP with 200 features improve the accuracy of KMeans by 10.6% on Citeseer dataset. Besides the improved accuracy and NMI, using selected features rather than all features would also result in speed-up of clustering time.

When comparing POP with the baseline methods, we observe that POP based methods (especially PPOP and MMPOP) consistently perform better than baseline methods in terms of both accuracy and NMI. This indicates that POP is an effective criterion for selecting high-quality features. Also, POP tends to obtain good performance with a small number of features (i.e., 200 to 400) while baseline methods normally need more features (i.e., 600 to 800).

Another thing worth noting is the poor performance of clustering with only link structure. Since links in networks are often sparse and noisy, structural information alone is not sufficient to obtain good clusters. But using link structure as guidance to select features achieves much better performance, which illustrates the strength of the POP feature selection. Baseline LUFS exploits link information via extracting social dimensions [9] from links. But social dimensions extracted from noisy and sparse links can be unreliable and this may further mislead the feature selection process.

6.5 Partial Order Preserving Property Our approach (POP) has an objective of preserving partial order as described in previous sections. In this section, we illustrate this partial order preserving effect through kNN (we use $k = 1$) link prediction. For each node v, we retrieve the top 1 node u of highest similarity to v. We test if this retrieved node u is an actual neighbor of node v on the network. The precision@1 is shown in Figure 3.



(a) Accuracy on Citeseer (b) NMI on Citeseer

(c) Accuracy on Cora (d) NMI on Cora

(e) Accuracy on Wiki (f) NMI on Wiki

Figure 2: KMeans Results on Three Datasets

Since this 1NN retrieval uses content only, the prediction performances of all methods are very limited. It also indicates that many similar nodes are not connected in these three datasets. Under such circumstances, POP approaches still outperform other feature selection baselines. This means POP is robust to incomplete link structure.

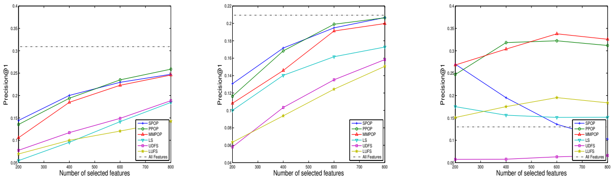
POP approaches outperform the baseline methods (LS, UDFS, LUFS) significantly. PPOP and MMPOP usually improve the performance of three other baselines by more than 50% on each dataset. This illustrates that POP's strength in respecting the network structure due to its connection to AUC optimization. The three instantiations of POP perform similarly on Citeseer and Cora datasets. But on Wiki dataset, the performance of SPOP degrades significantly. This is because SPOP ignores the correlation between features and only analyzes each feature individually. This might not result in serious problem when the number of non-zero features in each instance is low (e.g., Citeseer and Cora). However, it would lead to degenerated performance when the number of non-zero features per instance is large, which is the case in Wiki dataset.

LUFS has the ability to incorporate network structure through social dimension. But it utilizes the network information at a community level and fails to exploit the finer grained information of networks. To fur-

[4] We use the code at `http://www.cad.zju.edu.cn/home/dengcai/Data/Clustering.html`

Table 4: Average document frequency (df) of selected features (top 400)

Dataset	All features	LS	UDFS	LUFS	SPOP	PPOP	MMPOP
Citeseer	28.40	10.23	102.39	76.11	134.30	84.48	70.81
Cora	34.34	52.62	71.61	56.59	80.53	58.42	55.67
Wiki	426.42	598.71	946.91	678.41	1084.40	274.31	262.20



(a) Precision@1 on Citeseer (b) Precision@1 on Cora (c) Precision@1 on Wiki

Figure 3: 1NN Results on Three Datasets

ther understand the difference between different methods, we present the average *document frequency* (df) of features selected by each approach. As shown in Table 4, UDFS tends to select features with high df. This might be fine for clustering, but it loses too much microscopic information. In comparison, PPOP and MMPOP can make a more balanced selection without favoring features with high df in particular. In summary, the features selected by POP are not only better for macroscopic analysis such as clustering, but also good at microscopic analysis because POP respects the local partial order.

7 Conclusion

Network structures present valuable information as well as new challenges to feature selection. In this paper, we develop an efficient unsupervised feature selection algorithm for network data based on partial order preserving (POP) principle, a new perspective on using links to guide feature selection. Our method is conceptually simple and computationally efficient, whereas state-of-the-art approaches typically involve heavy matrix computation and are intractable for large real world networks. Also, state-of-the-art approaches usually have several parameters to tune. In contrast, our approach is parameter-free. Experiments indicate that our approach significantly outperforms state-of-the-art methods in terms of both efficiency and effectiveness.

Acknowledgements

This work is supported in part by NSF through grants CNS-1115234, and OISE-1129076, and the Pinnacle Lab at Singapore Management University.

References

[1] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection." in *NIPS*, 2005.

[2] F. Nie, H. Huang, X. Cai, and C. H. Q. Ding, "Efficient and robust feature selection via joint l2, 1-norms minimization." in *NIPS*, 2010, pp. 1813–1821.

[3] Z. Zhao, L. Wang, and H. Liu, "Efficient spectral feature selection with minimum redundancy." in *AAAI*, 2010.

[4] Y. Yang, H. T. Shen, Z. Ma, Z. Huang, and X. Zhou, "l2, 1-norm regularized discriminative feature selection for unsupervised learning." in *IJCAI*, 2011, pp. 1589–1594.

[5] Z. Li, Y. Yang, J. Liu, X. Zhou, and H. Lu, "Unsupervised feature selection using nonnegative spectral analysis," in *AAAI*, 2012.

[6] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E*, vol. 69, no. 2, p. 026113, Feb. 2004.

[7] L. Backstrom and J. Leskovec, "Supervised random walks: predicting and recommending links in social networks." in *WSDM*, 2011, pp. 635–644.

[8] J. Tang and H. Liu, "Unsupervised feature selection for linked social media data." in *KDD*, 2012, pp. 904–912.

[9] L. Tang and H. Liu, "Relational learning via latent social dimensions." in *KDD*, 2009.

[10] Z. Zhao and H. Liu, "Spectral feature selection for supervised and unsupervised learning." in *ICML*, vol. 227, 2007, pp. 1151–1157.

[11] Q. Gu and J. Han, "Towards feature selection in network," in *CIKM*, 2011, pp. 1175–1184.

[12] J. Tang and H. Liu, "Feature selection with linked data in social media." in *SDM*, 2012, pp. 118–128.

[13] ——, "Coselect: Feature selection with instance selection for social media data." in *SDM*. SIAM, 2013, pp. 695–703.

[14] J. Tang, X. Hu, H. Gao, and H. Liu, "Unsupervised feature selection for multi-view data in social media." in *SDM*. SIAM, 2013, pp. 270–278.

[15] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized non-negative matrix factorization for data representation," *PAMI*, vol. 33, no. 8, pp. 1548–1560, 2011.

[16] T. Joachims, T. Finley, and C.-N. Yu, "Cutting-plane training of structural SVMs," *Machine Learning*, vol. 77, no. 1, pp. 27–59, 2009.

[17] S. Shalev-Shwartz, Y. Singer, and N. Srebro, "Pegasos: Primal estimated sub-gradient solver for svm." in *ICML*, vol. 227, 2007, pp. 807–814.

[18] P. Sen, G. M. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, "Collective classification in network data," *AI Magazine*, vol. 29, no. 3, pp. 93–106, 2008.