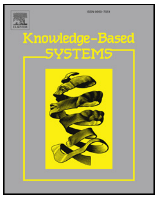
Knowledge-Based Systems 159 (2018) 233–243

Contents lists available at ScienceDirect

Knowledge-Based Systems

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



Collective behavior learning by differentiating personal preference from peer influence

Zan Zhang[a,b], Lin Liu[b], Hao Wang[*,a], Jiuyong Li[b], Daning Hu[c], Jiaqi Yan[d], Rene Algesheimer[e], Markus Meierer[e]

[a] *School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, Anhui, China*
[b] *School of Information Technology and Mathematical Sciences, University of South Australia, Adelaide, Australia*
[c] *Department of Informatics, University of Zurich, Zurich, Switzerland*
[d] *School of Information Management, Nanjing University, Nanjing, Jiangsu, China*
[e] *Department of Business Administration, University of Zurich, Zurich, Switzerland*

ARTICLE INFO

Keywords:
Classification with networked data
Causal analysis
Collective behavior
Propensity score

ABSTRACT

Networked data, generated by social media, presents opportunities and challenges to the study of collective behaviors in a social networking environment. In this paper, we focus on multi-label classification on networked data, for which behaviors are represented as labels and an individual can have multiple labels. Existing relational learning methods exploit the connectivity of individuals and they have shown better performance than traditional multi-label classification methods. However, an individual's behavior may be influenced by other factors, particularly personal preference. Hence, we propose a novel approach that integrates causal analysis into multi-label classification to learn collective behaviors. We employ propensity score matching and causal effect estimation to distinguish the contributions of peer influence and personal preference to collective behaviors and incorporate the findings into the design of the classifier. We further study behavior heterogeneity across subgroups in social networks, as people with different demographic features may behave differently due to different impacts of peer influence and personal preference. We estimate conditional average causal effects to analyze the impacts of peer influence and personal preference in different subgroups in social networks. Experiments on real-world datasets demonstrate that our proposed methods improve classification performance over existing methods.

1. Introduction

The advancement in social networks has produced massive amount of networked data. Increasing attention has been paid to the learning of human collective behaviors from networked data. For example, given some individuals' behaviors (e.g. adoption of certain products), how to infer the others' behaviors in the same social network. This can be considered as a classification problem where individuals' behaviors are the labels and the task is to learn a classifier from the labeled individuals, which then can be used to predict the behaviors of the other individuals.

A key challenge to networked data classification is that instances in the data are not independently identically distributed (i.i.d.) [1]. Individuals in a social network interconnect through different types of links. Conventional approaches, which usually assume that the individuals or instances are i.i.d., often have unsatisfactory performances with the data [2]. Relational learning (RL) has been proposed to address this problem by utilizing the connectivity between individuals [3,4]. Many studies have shown that the RL methods have better performance than traditional classifiers [5–7].

However, some existing RL methods only consider *peer influence*, without taking into account other factors. Peer influence is defined as how one's behaviors change with the change of his/her friends' behaviors [8]. In a networked dataset, an individual's friends are those directly connected to the individual in the network. However, peer influence may only provide partial information for correct labeling, since other factors, particularly an individual's *personal preference* can play an important role in their behaviors. In this paper, we use the term personal preference to represent the tendency of a person to have certain behavior (i.e. class label) as a result of his/her characteristics or personality. For instance, some people buy iPhones because they are Apple fans, instead of just being influenced by their friends. We consider

[*] Corresponding author.
E-mail address: jsjxwangh@hfut.edu.cn (H. Wang).

https://doi.org/10.1016/j.knosys.2018.06.027
Received 18 June 2017; Received in revised form 11 June 2018; Accepted 14 June 2018
Available online 31 July 2018
0950-7051/ © 2018 Elsevier B.V. All rights reserved.

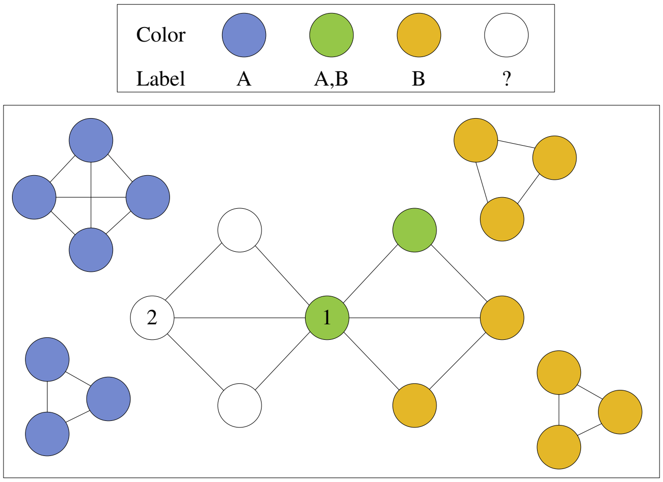


Fig. 1. A simple example of classification on networked data.

individuals with similar personal preference due to similar personality or characteristics tend to behave similarly, and the more similar two individuals are in their personal preference, the more likely they have the same behavior.

Although some existing RL methods consider both peer influence and the effects of personal preference, they do not distinguish the impacts of these two factors and consider that the contributions of these two factors are equal. Therefore, it is necessary to develop new method to consider both factors and distinguish their contributions. We use the example in Fig. 1 to show the effects of peer influence and personal preference on classification. There are two labels, A and B, and a node/individual's characteristics are indicated by node color. In this example, Node 1 has both two labels A and B. Node 2 connects to Node 1 and two other unlabeled nodes. Assume that Node 2 has one label only (label A or B), and our task is to label (classify) Node 2. Connectivity-based methods would classify Node 2 to have the same labels as Node 1 (both A and B) based on the connectivity, because in Node2's neighbors, only Node1's labels are known. However, an individual's behavior is not only a result of peer influence, but also due to personal preference. We can use personal preference to provide extra information for classification. Assume that Node2's characteristics are more similar to those of the nodes with label A than the nodes with label B. We can infer that Node 2 should have higher probability to be assigned label A, because Node2's personal preference is more similar to the nodes with label A.

From this example, we see that it is important to distinguish personal preference from peer influence and use both for classification. However, it is challenging to model and quantify the impacts of the two factors in networked data classification. For instance, in the context of adoption of iPhones, peer influence is associated with the presence of iPhone adopters in one's friends (called *adopter friends* hereafter). Personal preference is associated with having similar personal preference with other people. However, as the impacts of peer influence and personal preference are intertwined, it is difficult to estimate how much one's behavior is due to the influence of adopter friends and how much is a result of personal preference only.

Furthermore, the impacts of peer influence and personal preference vary across different subpopulations in social networks. There has been some work studying the behavioral heterogeneity [9–12]. For instance, political scientists and campaign professionals have conducted randomized experiments to investigate whether phone calls or in-person conversations are more effective at increasing candidate support. They considered research questions related to heterogeneity of subpopulations, e.g. "Do phone calls increase candidate support more from the female subpopulation than from the male subpopulation?" and "How does the effectiveness of phone calls change across subpopulations at different ages?" [13].

However, no study has been done on such heterogeneity for collective behavior learning from networked data. In learning collective behaviors in social networks, we are interested in similar questions regarding the heterogeneity in different subgroups. For example, for a female, is her friends' adoption of iPhones more likely to increase the chance for her to adopt an iPhone than for a male? Different people with different demographic features may behave differently due to different impacts of peer influence and personal preference. Therefore considering the heterogeneity of causal effects of peer influence and personal preference in different subgroups can help with accurate identification of their contributions to collective behaviors.

In this paper, we present **MCPP**, the **M**ulti-label **C**lassification algorithm which distinguishes **P**eer influence and **P**ersonal preference. We innovatively apply propensity score matching to identify and quantify the causal effect of peer influence on a node's labeling and thus to obtain the weights of peer influence and personal preference regarding their respective contributions to the labeling of a node. The weights are then used in the design of a multi-label relational classifier. We further propose (**MCPPS**), the **M**ulti-label **C**lassification algorithm which distinguishes **P**eer influence and **P**ersonal preference in **S**ubgroups to learn collective behavior while taking heterogeneity of subgroups into consideration. We use real social network datasets in our experiments. The results demonstrate that our proposed approaches can improve the performance of networked data classification.

The principal contributions of this paper as be summarized as follows:

- We propose a causal analysis approach to distinguishing the contributions of peer influence and personal preference to the collective behaviors in a social network environment, and we provide a method to examine the heterogeneity of peer influence and personal preference by estimating the conditional average causal effect in different subgroups.
- We design two multi-label classification algorithms based on the findings of the causal analyses. That is, we use the estimated causal effects of peer influence and personal preference to weight their respective contributions to the class membership probabilities (whereas existing methods either only consider a single factor or use equal weights for the two factors). We also show the effectiveness of the algorithms by making a comparative study with the state-of-the-art methods for networked data classification.

2. Problem definition

Let $\mathcal{G} = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{C}, \boldsymbol{F})$ represent a social network, where $\boldsymbol{V}$ is the set of nodes denoting individuals and $\boldsymbol{E}$ the set of undirected edges denoting the relationships between the nodes; $\boldsymbol{C}$ is the set of labels each for a behavior in $\mathcal{G}$; and $\boldsymbol{F}$ is the set of features describing an individual. For a node $v \in \boldsymbol{V}$, $\boldsymbol{N} \subset \boldsymbol{V}$ denotes the set of neighbor nodes directly linked to v.

The behaviors studied here refer to the collective behaviors shared by a group of individuals in a social network, e.g. buying a product. For $\boldsymbol{C}=\{c_1, c_2, \ldots, c_m\}$, the behaviors of an individual $v \in \boldsymbol{V}$ can be described by a binary vector, $\boldsymbol{l}= (l^{c_1}, l^{c_2}, \ldots, l^{c_m})$, where $l^{c_k} = 1$ if $c_k \in \boldsymbol{C}$ is a label of v; otherwise $l^{c_k} = 0$. For instance, if $\boldsymbol{C}= \{c_1, c_2, c_3\}$, representing the three behaviors considered in a social network, e.g. buying an iPhone, a Samsung or Sony phone, then $\boldsymbol{l}= (0, 1, 0)$ indicates that v bought a Samsung phone.

Our goal is to predict individuals' behaviors based on the observed behaviors of other individuals in the same social network. The major problem addressed in this paper can be defined as follows.

Problem Definition. Given $\mathcal{G} = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{C}, \boldsymbol{F})$, and assume that $\forall v' \in \boldsymbol{V'}$ where $\boldsymbol{V'} \subset \boldsymbol{V}$, its behavior vector $\boldsymbol{l'}$ is known. The goal of this paper is to predict the behavior vector $\boldsymbol{l}$ for each $v \in (\boldsymbol{V} \setminus \boldsymbol{V'})$.

We consider that peer influence and personal preference are the two major factors impacting individuals' behaviors, and in our design of the multi-label classifier, we take both factors into account.

To represent the probability of a label c belonging to a node v due to **P**eer **I**nfluence, we use $P_{pi}(l^c)$. To represent the probability of c belonging to v due to **P**ersonal **P**reference, we use $P_{pp}(l^c)$. Let α^c and β^c stand for the weights for the contributions of peer influence and personal preference on label c, respectively, and we assume $\alpha^c + \beta^c = 1$. Our method estimates $P(l^c)$, the overall class membership probability of label c belonging to v as:

$$P(l^c) = \alpha^c P_{pi}(l^c) + \beta^c P_{pp}(l^c) \tag{1}$$

This seemingly simple formulation is a novel contribution in networked data classification. Existing methods either do not consider personal preference ($\beta^c = 0$), or use equal weights for the two factors ($\alpha^c = \beta^c$), and assume that the values of α^c and β^c are unchanged for all labels.

Connectivity based methods assume that $P(l^c)$ is only affected by the nodes connected to v ($\beta^c = 0$), i.e. they only use peer influence. The wvRN algorithm [4] is one of the most successful connectivity based methods. It predicts the labels of a node based on the labels of the node's immediate neighbors. Specifically, wvRN estimates $P(l^c)$ as the mean of the class membership probabilities of v's neighbors. The model can be described as:

$$P(l^c) = P_{pi}(l^c) \tag{2}$$

Some methods consider both peer influence and personal preference, but assuming equal contributions by them ($\alpha^c = \beta^c$). For example, the SCRN algorithm [14] extends wvRN by introducing a class propagation probability calculated based on the similarity between the features of node v's and the reference features for label c. So the propagation probability indicates the influence of personal preference. Then SCRN estimates $P(l^c)$ based on the labels of its immediate neighbors and the class propagation probability as:

$$P(l^c) = P_{pi}(l^c) P_{pp}(l^c) \tag{3}$$

Moreover, for different labels, the contributions of peer influence and personal preference are different. For example, considering buying an iPhone vs. choosing a restaurant, in the former case, the influence of personal preference is often larger. Therefore, estimating the distinct contributions of peer influence and personal preference for different labels is very important. Our formulation captures the diverse roles of the two factors for different labels by α^c and β^c, whose values depend on the value of label c.

For ease of reference, Table 1 lists the main symbols used in this paper and their meanings.

Table 1
Summary of mathematical symbols.

$\mathcal{G}$	a social network, $\mathcal{G} = (\boldsymbol{V}, E, C, F)$
$\boldsymbol{V}$	node set of $\mathcal{G}$, $\boldsymbol{V} = \{v_1, v_2, ..., v_n\}$
$\boldsymbol{E}$	edge set of $\mathcal{G}$, $\boldsymbol{E} \subseteq \boldsymbol{V} \times \boldsymbol{V}$
$\boldsymbol{C}$	label set of $\mathcal{G}$, $\boldsymbol{C} = \{c_1, c_2, ..., c_m\}$
$\boldsymbol{F}$	feature set of $\mathcal{G}$, $\boldsymbol{F} = \{F_1, F_2, ..., F_q\}$
$\boldsymbol{f}$	the feature vector of $v \in \boldsymbol{V}$, $\boldsymbol{f} = (f_1, f_2, ..., f_q)$, where f_i is a value of $F_i \in \boldsymbol{F}$ $(1 \le i \le q)$
$\boldsymbol{l}$	the binary label vector of a node $v \in \boldsymbol{V}$, $\boldsymbol{l} = (l^{c_1}, l^{c_2}, ..., l^{c_m})$, $l^{c_i} = 1$ $(1 \le i \le m)$ if $c_i \in \boldsymbol{C}$ is a label of v and $l^{c_i} = 0$ otherwise
$\boldsymbol{V'}$	the subset of nodes whose labels are known
α^c	the weight for the contribution of peer influence on label c
β^c	the weight for the contributions of personal preference on label c
$P_{pi}(l^c)$	the probability of label c belonging to a node $v \in \boldsymbol{V}$ due to Peer Influence
$P_{pp}(l^c)$	the probability of the label c belonging to a node $v \in \boldsymbol{V}$ due to Personal Preference
$P(l^c)$	the overall class membership probability of label c belonging to a node $v \in \boldsymbol{V}$
T^c	the binary variable representing the treatment status of a node $v \in \boldsymbol{V}$, $T^c = 1$ if v is treated (i.e. has one or more neighbors with label c) and $T^c = 0$ otherwise
Y^1	the potential outcome of v receiving the treatment $(T^c = 1)$
Y^0	the potential outcome of v without receiving the treatment $(T^c = 0)$
ACE^c	the average causal effect of T^c on $v \in \boldsymbol{V}$. By definition, $ACE^c = E(Y^1) - E(Y^0)$
Y^{obs}	the observed outcome of T^c on $v \in \boldsymbol{V}$
$\boldsymbol{X}$	the set of covariate variables, and $\boldsymbol{X} \subset \boldsymbol{F}$
$\boldsymbol{U}$	the set of features specifying a subgroup, and $\boldsymbol{U} \subset \boldsymbol{F}$
PS^c	the propensity score of an individual $v \in \boldsymbol{V}$ (for whom $\boldsymbol{X} = \boldsymbol{x}$) receiving treatment T^c
$CACE^c$	conditional average treatment effect of T^c on $v \in \boldsymbol{V}$ in a subgroup $\boldsymbol{U} = \boldsymbol{u}$
$d(v)$	the degree of a node $v \in \boldsymbol{V}$
$\boldsymbol{N}$	the set of all neighboring nodes of $v \in \boldsymbol{V}$
$w(v, v')$	the weight $v' \in \boldsymbol{N}$, of a neighboring node of v, when classifying v
$\boldsymbol{S}$	the set of nodes with similar feature values as v
γ	ratio parameter controlling the number of similar nodes
$P(l^c \mid \boldsymbol{S})$	the probability of a node in $\boldsymbol{S}$ that is labeled with c
$\bar{\boldsymbol{f}}_{S^c}$	the mean feature value of the nodes in $\boldsymbol{S}$ that are labeled with c
$cos(\boldsymbol{f}, \boldsymbol{f'})$	the cosine similarity between nodes v and v' whose feature vectors are $\boldsymbol{f}$ and $\boldsymbol{f'}$ respectively

As discussed previously, in different subgroups, the contributions of peer influence and personal preference can be different. Therefore, in this paper, we also study the heterogeneity and consider it in classifier design.

Human behaviors are fundamentally a causal process. Hence, we take a causal approach to estimate the impacts of personal preference and peer influence. We will describe the proposed causal approach in the next section.

3. Methods

As shown in Eq. (1), our method estimates $P(l^c)$, the overall class membership probability of v having label c, based on the contributions of peer influence and personal preference, weighted by α^c and β^c respectively. In this section, we firstly present how to estimate α^c and the β^c (Section 3.1) using the causality based approach. Then we describe how to calculate the probabilities due to the two factors, $P_{pi}(l^c)$ and $P_{pp}(l^c)$, respectively (Section 3.2).

3.1. *Estimating the contributions of peer influence and personal preference*

3.1.1. *Quantifying the two factors without considering heterogeneity*

We follow the potential outcome model [8,15] to estimate the causal effect of peer influence. Comparing to correlation analysis, causal approaches assess the effect of a cause variable on the outcome while eliminating the influence of other factors, enabling us to have a "purer" estimation of peer influence.

Peer influence is defined as how one's behaviors change with the change of his/her friends' behaviors. Therefore, we consider having adopter friends (peer influence) as the treatment to an individual, and the individual's adoption behavior as the outcome. For a given label c and an individual v, let T^c be the treatment variable as defined below:

$$T^c = \begin{cases} 1, & v \text{ has one or more neighbors with label } c \\ 0, & \text{otherwise} \end{cases}$$

Each individual v has two potential outcomes: Y^1, the potential outcome of v receiving the treatment ($T^c = 1$), and Y^0, the potential outcome of v not receiving the treatment ($T^c = 0$). Then the causal effect of T^c on v's behavior regarding label c is:

$$CE(T^c) = Y^1 - Y^0 \tag{4}$$

When we aggregate the causal effect on all individuals, we have the average causal effect (ACE) of T^c as:

$$ACE^c = E(Y^1) - E(Y^0) \tag{5}$$

However, in reality, each individual only has one outcome, i.e. we cannot observe the adoption behaviors of the same individual with and without adopter neighbors at the same time. We can only observe either Y^1 or Y^0 for each individual. Therefore to estimate causal effects, matching methods [8,16] are often used to obtain a treated group (individuals with $T^c = 1$) and a control group (individuals with $T^c = 0$) such that the distributions of the covariates (denoted as $\boldsymbol{X}$) in the two groups are similar. In this way, individuals in the two groups have similar characteristics (described by the covariates) except their status of getting treatment T^c. Hence the effects of the covariates on the outcomes can be eliminated, and we can estimate ACE^c as the difference in the average (observed) outcomes of the two groups:

$$ACE^c = E(Y^{obs}|T^c = 1) - E(Y^{obs}|T^c = 0) \tag{6}$$

where Y^{obs} is the observed outcome of a node.

In a social network, normally the dimensionality of the feature set is high, thus using the above illustrated exact matching on all features will result in small treatment and control groups as many unmatched samples are dropped. Instead of matching on all features, the propensity score summarizes the features of an individual into one single value (a scalar), and matching based on propensity score [17] has been shown to be effective for high-dimensional datasets. Therefore, in this paper, we use propensity score matching.

Definition 1 (*PROPENSITY SCORE*). Let T^c be a binary treatment and $\boldsymbol{X}$ the set of covariates. The propensity score of an individual v (for whom $\boldsymbol{X} = \boldsymbol{x}$) is defined as the conditional probability of the individual receiving the treatment given $\boldsymbol{X} = \boldsymbol{x}$:

$$PS^c = P(T^c = 1|\mathbf{X} = \mathbf{x}) \tag{7}$$

Propensity scores are commonly estimated using logistic regression [17].

When applying propensity score matching, for each label c, for each treated node v, i.e. a node has 1 or more neighbors with label c, we choose an untreated individual v' such that among all the untreated nodes, the propensity score of v' is the closest to v's propensity score. Then we add v to the treated group and v' to the control group. Finally we use the dataset only containing the matched samples in causal effect estimation.

We are interested in the relative contributions of peer influence and personal preference instead of their absolute causal effects. Therefore, we estimate the weight of peer influence, α^c using the ratio of ACE^c (contribution to the outcome/behavior purely due to peer influence) to the observed average outcome in the treated group:

$$\alpha^c = \frac{ACE^c}{E(Y^c|T^c = 1)} = \frac{E(Y^{obs}|T^c = 1) - E(Y^{obs}|T^c = 0)}{E(Y^c|T^c = 1)} \tag{8}$$

After the weight of peer influence is estimated, we consider that the remaining contribution to one's behavior is from personal preference. Therefore, we estimate the weight of personal preference as the remaining portion of the overall contribution:

$$\beta^c = 1 - \alpha^c = 1 - \frac{ACE^c}{E(Y^{obs}|T^c = 1)} = \frac{E(Y^{obs}|T^c = 0)}{E(Y^{obs}|T^c = 1)} \tag{9}$$

3.1.2. Quantifying the two factors when considering heterogeneity

In this section, we study how peer influence (and personal preference) would vary in different subgroups of population, by estimating the Conditional Average Treatment Effect of peer influence, T^c, in each of the subgroups described by the set of features $\boldsymbol{U} \subset \boldsymbol{F}$ [18]:

$$CACE^c_{\boldsymbol{u}} = E(Y^{obs}|T^c = 1, \boldsymbol{U} = \boldsymbol{u}) - E(Y^{obs}|T^c = 0, \boldsymbol{U} = \boldsymbol{u}) \tag{10}$$

For the above estimation, we still utilize propensity score matching to obtain the treated group ($T^c = 1$) and the control group ($T^c = 0$), but in a subgroup instead of the whole population.

Similar to the case in the whole population, in each subgroup, we are also interested in the relative contributions of peer influence and personal preference. So we estimate the weight of peer influence, $\alpha^c_{\boldsymbol{u}}$ using the ratio of $CACE^c_{\boldsymbol{u}}$ to the observed average outcome in the treated group in the subpopulation as follows:

$$\begin{aligned}\alpha^c_{\boldsymbol{x}} &= \frac{CACE^c_{\boldsymbol{x}}}{E(Y^{obs}|T^c = 1, \boldsymbol{U} = \boldsymbol{u})}\\ &= \frac{E(Y^{obs}|T^c = 1, \boldsymbol{U} = \boldsymbol{u}) - E(Y^{obs}|T^c = 0, \boldsymbol{U} = \boldsymbol{u})}{E(Y^{obs}|T^c = 1, \boldsymbol{U} = \boldsymbol{u})}\end{aligned} \tag{11}$$

We can then estimate the weight of personal preference, $\beta^c_{\boldsymbol{u}}$ as the remaining portion of the overall contribution in the subgroup:

$$\beta^c_{\boldsymbol{u}} = 1 - \alpha^c_{\boldsymbol{u}} = 1 - \frac{CACE^c_{\boldsymbol{u}}}{E(Y^{obs}|T^c = 1, \boldsymbol{U} = \boldsymbol{u})} = \frac{E(Y^{obs}|T^c = 0, \boldsymbol{U} = \boldsymbol{u})}{E(Y^{obs}|T^c = 1, \boldsymbol{U} = \boldsymbol{u})} \tag{12}$$

3.2. Estimating class membership probabilities

In this section, we present our approach for estimating $P_{pi}(l^c)$ and $P_{pp}(l^c)$. We firstly present the estimation without considering heterogeneity in different subgroups in Section 3.2.1, then we describe the method when the heterogeneity is considered in Section 3.2.2.

3.2.1. Class membership probabilities without considering heterogeneity

3.2.1.1. Estimating $P_{pi}(l^c)$. Peer influence affects the likelihood for one to have the same behavior as the peers. It is reasonable to assume that the degree of peer influence on node v to have label c is related to the number of v's peers with label c, and the importance of the peers. The more neighbors of v having a certain label, the larger the peer influence v may receive from the neighbors, and thus the higher the likelihood for v to have the same label. Furthermore, the more important a neighbor of v is, the larger the influence the neighbor would have on v's decision, increasing the likelihood for v to have the same label as the neighbor.

In a social network, generally a node with higher degree has more labels than a node with lower degree. Traditional relational learning methods consider that the high degree nodes have large influence in a social network, so when calculating the weight of a node, larger weights are assigned to nodes of higher degree. However, some researchers [19] argued that comparing with a small number of high degree nodes, a large number of low or medium degree nodes should be considered more reliable in classification as discussed below.

Since a node with a large number of labels may be more likely to have irrelevant labels (label noise), we consider that nodes of high degree contain high label noise when we use them in classifying low degree nodes. Conversely, low or medium degree nodes are reliable when they are used in classifying high degree nodes, since they introduce smaller label noise. Therefore, we assign larger weights to low or medium degree nodes. When node v' is used in the classification of node v, we define the weight of v', $w(v, v')$ as:

$$w(v, v') = \frac{1}{d(v')} \tag{13}$$

where $d(v')$ is the degree of v' and the range of $w(v, v')$ is (0,1].

Therefore, given v's neighboring nodes, $\boldsymbol{N}$, we estimate $P_{pi}(l^c)$ as:

$$P_{pi}(l^c) = P(l^c|\boldsymbol{N}) = \frac{1}{z}\sum_{v'\in\boldsymbol{N}} w(v, v') \times P(l'^c) \tag{14}$$

where $z = \sum_{v'\in\boldsymbol{N}} w(v, v')$, is the normalization factor and $P(l'^c)$ is the probability of the label c belonging to v'. We use collective inference [4,14] to estimate the probability for unlabeled nodes. When we calculate the probabilities, the labels of a nodes neighbors are updated dynamically. That is, once we have obtained the estimation of $P(l^c)$, then the probability will be used in the estimation of another unlabeled node's probabilities if v is in the other unlabeled node's neighborhood.

3.2.1.2. Estimating $P_{pp}(l^c)$. It is reasonable to assume that two individuals are more likely to have similar labels, if they have similar personalities (no matter whether they are linked or not). Hence, in order to estimate $P_{pp}(l^c)$, we can find a set of nodes with similar features as v, denoted as $\boldsymbol{S}$, using cosine similarity:

$$\boldsymbol{S} = \{v' | cos(\boldsymbol{f}, \boldsymbol{f}') > \gamma\} \tag{15}$$

where $\boldsymbol{f}$ and $\boldsymbol{f}'$ are the feature vectors of v and v' respectively, and γ is a ratio parameter controlling the number of similar individuals.

Then the estimation of $P_{pp}(l^c)$ depends on two factors: (1) number of nodes in $\boldsymbol{S}$ that have label c. The larger the number, the more likely v is labeled with c; (2) the degree of similarity between v and nodes in $\boldsymbol{S}$ labeled with c.

We firstly estimate $P(l^c|\boldsymbol{S})$, the probability of a node in $\boldsymbol{S}$ having label c as:

$$P(l^c|\boldsymbol{S}) = \frac{1}{|\boldsymbol{S}|} \sum_{v' \in \boldsymbol{S}} P(l'^c|\boldsymbol{S}) \tag{16}$$

We use the cosine similarity between $\boldsymbol{f}$ and $\overline{\boldsymbol{f}}_{\boldsymbol{S}^c}$, the mean feature of the nodes in $\boldsymbol{S}$ that are labeled with c, to estimate the similarity between v and the nodes in $\boldsymbol{S}$ that are labeled with c. Then we have:

$$P_{pp}(l^c) = P(l^c|\boldsymbol{S}) \times \cos(\boldsymbol{f}, \overline{\boldsymbol{f}}_{\boldsymbol{S}^c}) \tag{17}$$

where $\overline{\boldsymbol{f}}_{\boldsymbol{S}^c} = \frac{1}{|\boldsymbol{S}|} \sum_{v' \in \boldsymbol{S}} P(l'^c = 1) \times \boldsymbol{f}$.

3.2.1.3. Estimating P(l^c). We can estimate the overall class membership probability of node v belonging to class c, $P(l^c)$, based on α^c, β^c, $P_{pi}(l^c)$ and $P_{pp}(l^c)$ as follows:

$$\begin{aligned} P(l^c) &= \alpha^c P_{pi}(l^c) + \beta^c P_{pp}(l^c) \\ &= \alpha^c \times \frac{1}{z} \sum_{v' \in \boldsymbol{N}} w(v, v') \times P(l'^c) + \beta^c \times P(l^c|\boldsymbol{S}) \times \cos(\boldsymbol{f}, \overline{\boldsymbol{f}}_{\boldsymbol{S}^c}) \end{aligned} \tag{18}$$

3.2.2. Calculating class membership probabilities when considering heterogeneity

In order to estimate the overall class membership probability of nodes in different subgroups, firstly we stratify the dataset into subgroups based on the stratifying features $\boldsymbol{U}$. Then for each subgroup, we use Eqs. (11) and (12) to estimate the weights of peer influence and personal preferences in the subgroup. Furthermore, $P_{pi}(l^c)$ and $P_{pp}(l^c)$, class membership probabilities due to peer influence and personal preference are estimated by following Eqs. (14) and (17), respectively, however, regarding the subgroup only. That is, in Eq. (14), the neighbors of node v and in Eq. (17) the set of nodes similar to v, are all within the subgroup.

Finally, for a node v in subgroup $\boldsymbol{U} = \boldsymbol{u}$, we estimate its overall class membership probability of have label c, $P(l^c)$ as follows:

$$\begin{aligned} P(l^c) &= \alpha_{\boldsymbol{u}}^c P_{pi}(l^c) + \beta_{\boldsymbol{u}}^c P_{pp}(l^c) \\ &= \alpha_{\boldsymbol{u}}^c \times \frac{1}{z} \sum_{v' \in \boldsymbol{N_x}} w(v, v') \times P(l'^c) + \beta_{\boldsymbol{u}}^c \times P(l^c|\boldsymbol{S_u}) \times \cos(\boldsymbol{f}, \overline{\boldsymbol{f}}_{\boldsymbol{S_u}^c}) \end{aligned} \tag{19}$$

3.3. Algorithms

Based on the discussions in previous sections, in this section, we propose the two algorithms for collective behavior learning, firstly the **M**ulti-label **C**lassification algorithm by distinguishing **P**eer influence from **P**ersonal preference (MCPP), then the **M**ulti-label **C**lassification algorithm by distinguishing **P**eer influence from **P**ersonal preference in **S**ubgroups (MCPPS).

3.3.1. MCPP

As shown in Algorithm 1, the input of MCPP includes the networked data $\mathcal{G}$ (the set of nodes $\boldsymbol{V}$, the set of edges $\boldsymbol{E}$, the set of labels $\boldsymbol{C}$ and the set of features $\boldsymbol{F}$ for all nodes), the label vector $\boldsymbol{l}'$ for each labeled node v' in $\boldsymbol{V}' \subset \boldsymbol{V}$, and the maximum number of iterations *Max_Iter*. The output is the overall class membership probabilities for all unlabeled nodes.

MCPP iteratively updates the overall class membership probabilities of all $v \in (\boldsymbol{V} \setminus \boldsymbol{V}')$ using the model defined in Eq. (18). A collective inference procedure is utilized to propagate label information through the network. The procedure terminates when one of the stopping criteria is met: no change in the labels assigned to nodes in $\boldsymbol{V} \setminus \boldsymbol{V}'$ or the specified *Max_Iter* has been reached.

Module 1 of Algorithm 1 (Lines 1–11) is to estimate α^c and β^c using the labeled nodes, $\boldsymbol{V}'$. For each label c, we define and initialize two sets $\boldsymbol{V}'_{T^c=1}$ and $\boldsymbol{V}'_{T^c=0}$ for the (matched) treated and control groups (Lines 1 and 2). Then we divide $\boldsymbol{V}'$ into two disjoint parts $\boldsymbol{V}'_1$ and $\boldsymbol{V}'_0$ based on the value of the treatment variable by putting nodes having 1 or more neighbors with label c into$\boldsymbol{V}'_1$, and nodes having no neighbors with label c into$\boldsymbol{V}'_0$ (Line 3). The propensity score of each labeled node is then computed (Lines 4 and 5). For each $v_i \in \boldsymbol{V}'_1$, we choose $v_j \in \boldsymbol{V}'_0$ whose propensity score is the closest to the propensity score of v_i, then we put the matched pair, v_i into $\boldsymbol{V}'_{T^c=1}$ and v_j into $\boldsymbol{V}'_{T^c=0}$ (Lines 6–9). At the end of the Module, using the matched treated and control groups, we estimate α^c and β^c according to Eqs. (8) and (9) (Line 10).

Module 2 (Lines 12–14) is to identify the similar nodes for each unlabeled node based on cosine similarity, as specified in Eq. (14).

Module 3 (Lines 15–25) is for estimating class membership probability of each unlabeled node. Firstly, the class membership probabilities due to peer influence ($P_{pi}(l^c)$) and personal preference ($P_{\text{pp}}(l^c)$) are estimated according to Eqs. (14) and (17), respectively. Then using these two probabilities and the weights, α^c and β^c obtained in Module 1, finally we estimate the overall class probability $P(l^c)$ according to Eq. (18).

3.3.2. MCPPS

Like MCPP, MCPPS (see Algorithm 2) iteratively updates the class probabilities of nodes in $\boldsymbol{V} \setminus \boldsymbol{V}'$ until all label classifications between iterations are stable or the user specified *Max_Iter* is reached.

In Module 1 of Algorithm 2, we first stratify the dataset into disjoint subgroups based on $\boldsymbol{U}$ (Line 1). For each subgroup $\boldsymbol{U} = \boldsymbol{u}$, we estimate $\alpha_{\boldsymbol{u}}^c$ and $\beta_{\boldsymbol{u}}^c$ using the same steps for estimating α^c and β^c in MCPP (Algorithm 1), within the subgroup. In Module 2, in each group, for every node in the group, we identify its similar nodes (Lines 8–12). Then in Module 3, we estimate the overall class membership probability for each unlabeled node by following the same steps of Module 3 in MCPP (Lines 13–22).

4. Experiments

4.1. Datasets

We use three real-world multi-label relational datasets, MOD, BlogCatalog and DBLP to evaluate the classification performance of our proposed algorithms. A summary of the three datasets can be found in Table 2.

The MOD (Movies-on-Demand) dataset is collected from a telephone company which offers telephone and Movies-on-Demand service. The dataset gives us the great opportunity to study the heterogeneity of personality and peer influence since it contains user information (anonymized). MOD consists of 10,284 users and each user is represented by a node. Two users are linked together if they have at least two phone calls in 2013. We use the 15 most popular movies in 2013 as labels. A user has a label if he/she has purchased a movie, and each user can have multiple labels. Our task is to classify users. For each user, we construct the feature vector with 15 attributes, such as gender, age, and contact language.

The BlogCatalog dataset [1] is collected by Tang and Liu [20]. This

[1] http://leitang.net/social_dimension.html

```
Input: $\mathcal{G} = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{C}, \boldsymbol{F})$, $\boldsymbol{l'}$ for each of the set of labeled nodes $v' \in \boldsymbol{V'} \subset \boldsymbol{V}$, and $Max\_Iter$.
Output: For each $v \in (\boldsymbol{V} \setminus \boldsymbol{V'})$, $P(l^c)$ regarding each $c \in \boldsymbol{C}$
Steps:
    Module 1. Estimating $\alpha^c$ and $\beta^c$
 1: for each label $c \in \boldsymbol{C}$ do
 2:    $\boldsymbol{V'}_{T^c=1} = \emptyset$; $\boldsymbol{V'}_{T^c=0} = \emptyset$
 3:    divide $\boldsymbol{V'}$ into two disjoint parts: $\boldsymbol{V'_1}$ and $\boldsymbol{V'_0}$
 4:    compute propensity score $PS_i^c$ for each $v_i \in \boldsymbol{V'_1}$
 5:    compute propensity score $PS_j^c$ for each $v_j \in \boldsymbol{V'_0}$
 6:    for each $v_i \in \boldsymbol{V'_1}$ do
 7:       choose $v_j \in \boldsymbol{V'_0}$ $(j = argmin_j(|PS_i^c - PS_j^c|))$
 8:       $\boldsymbol{V'}_{T^c=1} \leftarrow \boldsymbol{V'}_{T^c=1} \cup \{v_i\}$;   $\boldsymbol{V'}_{T^c=0} \leftarrow \boldsymbol{V'}_{T^c=0} \cup \{v_j\}$
 9:    end for
10:    compute $\alpha^c$ and $\beta^c$ according to Eq.8 and Eq. 9
11: end for
    Module 2. Identifying $\boldsymbol{S}$
12: for each $v \in (\boldsymbol{V} \setminus \boldsymbol{V'})$ do
13:    identify $\boldsymbol{S}$ according to Eq.15
14: end for
    Module 3. Estimating $P(l^c)$
15: #iteration=0
16: repeat
17:    for each $v \in (\boldsymbol{V} \setminus \boldsymbol{V'})$ do
18:       for each label $c \in \boldsymbol{C}$ do
19:          compute $P_{\mathrm{PI}}(l^c)$ according to Eq. 14
20:          compute $P_{\mathrm{PP}}(l^c)$ according to Eq. 17
21:          compute $P(l^c)$ according to Eq. 18
22:       end for
23:    end for
24:    #iteration++
25: until #iteration>$Max\_Iter$ or classifications converge to stable values
```

Algorithm 1. Multi-label classification by distinguishing peer influence from personal preference (MCPP).

dataset contains a network consisting of 10,312 bloggers who have submitted their blogs to BlogCatalog. In this network, a blogger is represented by a node, and two bloggers are linked together if they are friends in BlogCatalog. A bloggers interests could be gauged by the categories he/she publishes blogs in. The BlogCatalog dataset uses 39 categories as labels. An blogger has a label if he/she has published a blog in the corresponding category and each blogger can have multiple labels. Our task is to classify bloggers. The BlogCatalog dataset does not

```
Input: $\mathcal{G} = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{C}, \boldsymbol{F})$, $\boldsymbol{l'}$ for each of the set of labeled nodes $v' \in \boldsymbol{V'} \subset \boldsymbol{V}$, and $Max\_Iter$.
Output: For each $v \in (\boldsymbol{V} \setminus \boldsymbol{V'})$, $P(l^c)$ regarding each $c \in \boldsymbol{C}$
Steps:
    Module 1. Estimating $\alpha_x^c$ and $\beta_x^c$
 1: stratify dataset into disjoint subgroups based on $U$
 2: for each subgroup $\boldsymbol{U}=\boldsymbol{u}$ do
 3:    for each label $c \in \boldsymbol{C}$ do
 4:       repeat steps 2∼9 of MCPP (Algorithm 1) with nodes in the subgroup
 5:       compute $\alpha_u^c$ and $\beta_u^c$ according to Eq. 11 and Eq. 12, respectively
 6:    end for
 7: end for
    Module 2. Identifying $S_u$
 8: for each subgroup where $\boldsymbol{U}=\boldsymbol{u}$ do
 9:    for each $v$ in the subgroup do
10:       identify $S_u$ according to Eq.16 (with the subgroup)
11:    end for
12: end for
    Module 3. Estimating $P(l^c)$
13: #iteration=0
14: repeat
15:    for each $v \in (\boldsymbol{V} \setminus \boldsymbol{V'})$ do
16:       for each label $c \in \boldsymbol{C}$ do
17:          repeat steps 19∼20 of MCPP (Algorithm 1)
18:          compute $P(l^c)$ according to Eq. 19 (using the weights $\alpha_u^c$ and $\beta_u^c$ in the nodes
    subgroup)
19:       end for
20:    end for
21:    #iteration++
22: until #iteration>$Max\_Iter$ or classifications converge to stable values
```

Algorithm 2. Multi-label classification by distinguishing peer influence and personal preference in subgroups (MCPPS).

Table 2
Dataset summary.

Data	MOD	BlogCatalog	DBLP
Number of nodes	10,284	10,312	8865
Number of links	8,866	333,983	12,989
Maximum degree	39	3,992	86
Average degree	1.7	65	3
Number of labels	15	39	15
Average number of labels per node	2.2	1.4	2.3

contain nodes' features, so we use the method based on social dimensions to extract features from the network structure [20].

The DBLP dataset [2] is extracted from the DBLP database by Wang and Sukthankar [14]. This dataset contains a network consisting of 8865 authors who have co-authorship. In this network, an author is represented by a node, and two authors are linked together if they have co-authored at least two papers. The DBLP dataset uses 15 representative conferences as labels. An author has a label if he/she has published a paper in a conference and each author can have multiple labels. Our task is to classify authors. The DBLP dataset also does not contain nodes' features, so we also use social dimension based method to extract features, as we did with the BlogCatalog dataset above.

4.2. Experiment setup

In the experiments, we compare our work with the following state-of-the-art methods for classifying collective behaviors:

1. wvRN [4], a representative relational learning classifier. wvRN makes predictions based solely on the labels of a nodes linked neighbors (i.e. peer influence), so it is a connectivity based method. A nodes predicted membership of a label is constructed as the weighted mean of its neighbors' memberships of the label.
2. SCRN [14], an approach which extends wvRN by social features computed using the social dimension framework [1,20]. It firstly uses social features to calculate a class-propagation probability for a node. Then it makes prediction of a node's labels based on the labels of its neighbors, the weights between a node and its neighbors, and its class-propagation probability.
3. SNBC [19], a recently proposed approach for classifying nodes by random walk. For classifying a node, it takes a random walk from the node and makes a decision based on how nodes in the respective k^{th}-level neighborhood are labeled.

We have to construct features for nodes, since the DBLP and BlogCatalog datasets do not contain nodes' features. We use the ModMax algorithm [20] to construct features for the DBLP and the BlogCatalog. The dimensionality of features is set to 50. Then for each dataset, the R package, MatchIt [21] is adopted to do propensity score based matching, and we choose the Nearest Neighbor method in the package.

The ratio parameter γ is set to 0.6 for DBLP, 0.4 for BlogCatalog and 0.9 for MOD. We will discuss how to choose the value of γ in Section 4.4. *Max_Iter* of our two algorithms is set to 30, since all of the experiments showed that after 20 iterations, the label classifications between iterations converged.

Both wvRN and SCRN calculate the similarity of linked nodes, $w(v, v')$, when estimating the label of node v. We use the *Degree similarity measure* in [14] to calculate the similarity of linked nodes for SCRN, since the experiment results in [14] have shown that this measure achieves the highest accuracy. It calculates $w(v, v')$ by the normalized fraction of connections between v and v' among all of v's connections [14]. For a fair comparison, we use the degree similarity measure to calculate the similarity of linked nodes for wvRN too.

In SCRN, the class-propagation probability is calculated by the similarity between nodes' social features and class reference features. We use *Inner Product and Generalized Histogram Intersection Kernel* (GHI) [38] to calculate the similarity, since Wang and Sukthankar have shown that GHI outperforms the other measures [14].

We use the same Relaxation Labeling approach [39,40] in the collective inference framework for both SCRN and wvRN as Wang and Sukthankar did in [14]. In the relaxing procedure, the default parameter k is set to 1 and the parameter α is set to 0.99. The maximum number of iterations of SCRN and wvRN is set to 30, since all of the experiments showed that after 15 iterations, the label classifications between iterations converged.

In SNBC, the default regularization parameter λ is set to 2^{-6}. The sample size for stochastic gradient descent is set to $min(1000, |\boldsymbol{V}'|)$ as Nandanwar and Murty did in [19], where $|\boldsymbol{V}'|$ is the number of labeled nodes. The maximum number of iterations of SNBC is set to 1000.

For evaluation purpose, we assume that the number of labels for the unlabeled nodes is already known and assign the labels according to the top-ranking set of class-membership probabilities. Such a scheme has been adopted for multi-label evaluation in social network datasets [1]. We randomly sample a portion of nodes as labeled, and classify the remaining unlabeled nodes as done in [1,20]. We report the average performance of 10 runs in terms of two commonly used multi-label criteria: Micro-F1 score and Macro-F1 score. These criteria measure the performance from different aspects. Micro-F1 score is largely determined by the common labels, since this measure weights instances evenly. Macro-F1 score is sensitive to the performance for rare labels, since all labels are weighted evenly in the calculation of Macro-F1. Details of Micro-F1 score and Macro-F1 score can be found in [14].

In the experiments, we compare the performance of our MCPP algorithm to the 3 methods described above on the three real-world datasets. Because only the nodes' features of the MOD dataset have actual meaning, it would make real sense to stratify this dataset into subgroups for evaluating the MCPPS algorithm. We stratify the MOD dataset by gender, into the male and female subgroups. We do not stratify the MOD dataset by multiple covariates simultaneously, since it will divide the dataset to many small subgroups and thus lose statistical power. Then we compare MCPPS with gender subgroups to MCPP on the MOD dataset.

4.3. Classification results and discussions

Table 3 reports the Macro-F1 and Micro-F1 scores with the MOD dataset. It can be seen that MCPP consistently outperforms the other methods. MCPP captures a node's intrinsic likelihood of belonging to a label by quantifying the contributions of peer influence and personal preference, enabling a more accurate classification. SNBC has better performance than wvRN and SCRN. SCRN and wvRN have similar performance. Table 4 shows the results with the BlogCatalog dataset. MCPP again has the best performance. SCRN and wvRN have similar performance and they outperform SNBC. With the DBLP dataset (Table 5), MCPP also has the best performance. SCRN achieves better performance than wvRN whereas wvRN outperforms SNBC. These results have strongly demonstrated the effectiveness of MCPP.

Table 6 shows that MCPPS (gender) outperforms MCPP as measured with Micro-F1 score, whereas in terms of Macro-F1 scores, in most of the cases, MCPP performs better. Because Micro-F1 score is largely determined by the common labels and Macro-F1 score is sensitive to the performance for rare labels [14], the results in Table 6 confirms the existence of heterogeneity. These results also show that distinguishing peer influence from personal preference and considering their heterogeneity in collective behavior learning is a valuable and promising research direction. The approach has the potential to achieve better multi-label classification for networked data.

[2] http://ial.eecs.ucf.edu/Data/SCRN-Data.zip

Table 3
Classification results on MOD.

Proportion of labeled nodes		5%	7%	10%	20%	30%	40%
Micro-F1(%)	MCPP	**28.92**	**32.51**	**33.51**	**33.99**	**34.63**	**35.08**
	wvRN	18.43	19.55	20.58	23.13	24.06	27.02
	SCRN	18.58	19.80	20.67	23.23	24.21	27.05
	SNBC	22.79	23.11	27.66	30.13	30.59	31.28
Macro-F1(%)	MCPP	**18.92**	**19.62**	**20.46**	**21.69**	**22.83**	**23.08**
	wvRN	15.31	16.44	16.62	18.78	19.60	21.64
	SCRN	15.34	16.44	16.59	18.76	19.66	21.56
	SNBC	16.44	17.43	17.91	20.13	21.23	22.48

Table 4
Classification results on blogcatalog.

Proportion of labeled nodes		30%	40%	50%	60%	70%	80%
Micro-F1(%)	MCPP	**31.95**	**34.30**	**35.04**	**35.91**	**36.26**	**36.56**
	wvRN	26.22	29.55	31.03	31.98	32.74	33.69
	SCRN	26.08	29.51	31.02	31.96	32.63	33.63
	SNBC	23.96	24.32	24.97	25.32	25.73	26.27
Macro-F1(%)	MCPP	**17.99**	**20.34**	**21.65**	**22.81**	**23.44**	**23.64**
	wvRN	13.39	15.52	17.24	17.89	18.02	19.40
	SCRN	13.09	15.29	17.18	17.49	17.94	19.20
	SNBC	6.72	6.85	7.53	7.91	8.35	8.72

Table 5
Classification results on DBLP.

Proportion of labeled nodes		40%	50%	60%	70%	80%	90%
Micro-F1(%)	MCPP	**69.28**	**71.73**	**73.93**	**75.60**	**77.44**	**79.34**
	wvRN	68.51	70.29	72.52	73.27	75.33	76.85
	SCRN	68.88	70.67	72.88	74.12	75.81	77.64
	SNBC	64.84	68.29	70.92	72.64	73.24	75.71
Macro-F1(%)	MCPP	**62.86**	**65.56**	**68.23**	**70.24**	**72.71**	**74.68**
	wvRN	62.42	64.19	66.47	67.57	69.73	69.98
	SCRN	62.36	64.18	66.54	68.07	69.78	70.43
	SNBC	59.77	62.43	64.32	65.66	65.84	67.34

Table 6
Classification results of MCPPS compared with MCPP.

Proportion oflabeled nodes		5%	7%	10%	20%	30%	40%
Micro-F1(%)	MCPPS(gender)	**29.24**	**32.69**	**34.05**	**34.21**	**35.21**	**35.18**
	MCPP	28.92	32.51	33.51	33.99	34.63	35.08
Macro-F1(%)	MCPPS(gender)	18.01	19.30	**21.02**	21.34	21.58	21.60
	MCPP	**18.92**	**19.62**	20.46	**21.69**	**22.83**	**23.08**

4.4. Sensitivity of classification with respect to node similarity

In the experiments, we have fixed the ratio parameter γ to 0.9, 0.4, and 0.6 for the MOD, BlogCatalog and DBLP datasets respectively. In this section, we examine how the performance of MCPP is affected by the values of γ. The change of the performance versu the values of γ are plotted in Figs. 2–4 for the three datasets, respectively. To make the figures legible, we only plot the cases when 10%, 20% and 30% of nodes in the network are used as training data for MOD, and 40%, 50%, 60% of nodes in the network are used as training data for BlogCatalog and DBLP.

For the MOD dataset, as seen from Fig. 2, better results are obtained when γ increases. MCPP has the best performance when $\gamma = 0.9$. For the BlogCatalog dataset, as seen from Fig. 3, when $\gamma \leq 0.4$, better results are obtained when γ increases, but the Macro-F1 and Micro-F1 scores decrease when γ is greater than 0.4. For the DBLP dataset (Fig. 4), the performance of MCPP becomes stable as γ increases beyond 0.6. These observations have justified the MCPP's parameter setting in Section 4.2.

5. Related work

In the past decades, many multi-label classification methods have been proposed [22–24] and multi-label learning has been successfully applied to various applications such as text categorization [25,26], image classification [41–44], bioinformatics [27] and music categorization [45]. An intuitive approach to multi-label classification is to decompose the classification regarding multiple-labels into multiple independent binary classification problems. More sophisticated approaches exploit the correlations between different labels. Traditional multi-label classifiers assume that instances are i.i.d. However, as discussed in the Introduction, networked data does not satisfy the i.i.d. assumption, hence many methods were proposed to exploit the connectivity between instances, including:

- *Classification based on connectivity*. Macskassy and Provost [4] presented the weighted-vote relational neighbor classifier (wvRN) that performs relational classification via the weighted average of the estimated class-membership probabilities of a nodes neighbors. A collective inference procedure is utilized to propagate the label information through the network. Macskassy and Provost [4] have showed that the relational neighbor classifier performs well by comparing it to probabilistic relational models [46] and relational probability trees [7] on three data sets from published work. Goldberg et al. [28] used two edge types to denote the affinity or disagreement in class labels of linked objects and incorporated link type information into discriminate learning. Heatherly et al. [29] introduced a link type relational Bayes classifier to predict a nodes class labels according to their neighbors labels and their link types.
- *Classification by extracting social features*. Social networks usually contain many communities, and nodes in the same community structure have stronger relationships among themselves, compared to the rest of the networks. Therefore, a number of techniques for generating features from networks structures have been proposed [47,48]. Lu and Getoor proposed the network only link-based classifier (LBC) [6]. LBC extracts relational features of a node by aggregating the label information of its neighbors. Then a relational classifier can be constructed based on labeled data and nodes' features.

  Tang et al. [1,20] proposed the social dimension framework (SocDim) for node classification. They extracted nodes' social features based on network information. These social features describe diverse affiliations of nodes in the network, and can be used as the nodes' features for discriminative learning by a linear SVM classifier [49]. Deep-Walk [30] and LINE [31] are social representation learning approaches based on random walks. They capture neighborhood similarity and community membership in latent representations. These label independent representations are then used in multi-label classification.
- *Classification by combining connectivity and social features*. Wang and Sukthankar [14] extended wvRN by introducing a class-propagation probability to capture the likelihood of a node belonging to a class. They calculated the class-propagation probability based on the similarity of social features which were extracted using the scalable edge clustering method proposed in [1]. Wang and Sukthankar have showed SCRN has better performance than wvRN and SocDim on collaborative networks [14].
- *Classification based on random walk*. Some studies use random walk for classification of nodes. Zhou et al. [32] proposed a globally consistent learning approach on the lines of spectral clustering. The approach updates a node's label using information the node receives from its neighbors. Lin and Cohen proposed the Multi Rank Walk [50] approach, which is based on the principle of random graph walks similar to Page Rank [51]. The class of any unlabeled node is decided as the one which has the highest probability of containing terminal nodes of the random walk. Nandanwar and Murty [19]

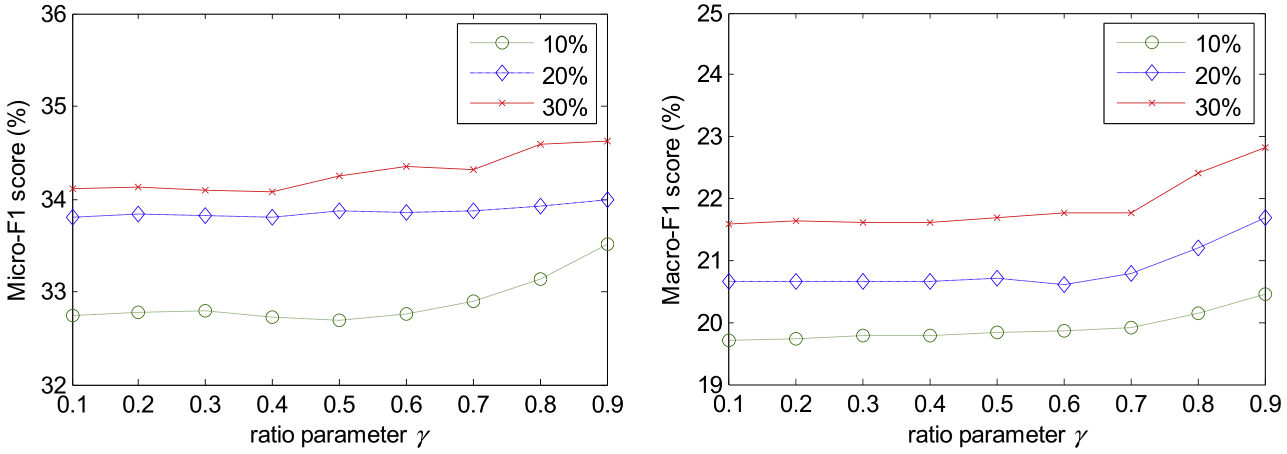


Fig. 2. Sensitivity study of MCPP on the MOD dataset.

proposed a SNBC algorithm. The approach is a structural neighborhood-based classifier learning using a random walk. For classifying a node, it takes a random walk from the node and makes a decision based on how nodes in the respective *k*th-level neighborhood are labeled. Nandanwar and Murty have showed SNBC has better performance than SCRN and SocDim on sparse citation networks.

In statistical learning, people usually choose samples randomly from the original dataset to represent the key characteristics of the original dataset [34]. However, some researchers focus on how to actively select the labeled nodes from the network to be used for training a better classification model. Many sampling methods have been proposed, which are mainly of three types: methods based on randomly selecting nodes [35], randomly selecting edges, and the exploration techniques that simulate random walks or virus propagation to find a representative sample of the nodes [36]. The forest fire sampling model [37] is similar to a random walk based approach. Firstly, this method selects a random seed node and burning a fraction of its outgoing links to neighboring nodes. If a link gets burned, the node at the other endpoint gets a chance to burn its own links. The selection process is recursive until no new nodes are selected, or when the required node size has been reached.

The above mentioned methods focus on exploiting the relational dependency among the nodes for classification. However, Aral et al. [8] acknowledged that peer influence and homophily are the two major factors, influencing people's product adoption decisions in a social network. They adopted the sample matching method to distinguish peer influence from homophily effects among connected individuals in social networks. It was shown in [8] that matching with propensity score could estimate the contributions of peer influence and homophily more accurately, comparing to random matching. Aral et al. [33] built and analyzed data-driven simulation of the effectiveness of seeding (network targeting) strategies by distinguishing peer influence from homophily.

Individuals differ from one another and differ in their response to treatments, so the causal effects should vary with population composition [10,16]. Many studies on causal inference recognized the importance of population heterogeneity [9]. For example, in the research on the effect of schools on students' academic performance, Morgan [11] using propensity score matching found that the effect of Catholic schooling differs for students in different subpopulations. A similar example, people normally think that attending elite universities will get better payoff. However, Brand and Halaby [12] found the effect of attending elite universities differs for individuals in different subgroups. They used propensity score matching to find that an elite college education is beneficial for students who have socioeconomically disadvantaged backgrounds. These studies all find that the causal effects differ from subpopulations.

For collective behavior learning, no work has considered the heterogeneity, so in this paper, we study the contributions of peer influence and personal preference with considering heterogeneity.

6. Conclusion and future research

In this paper, we aim to develop multi-label classifiers to predict

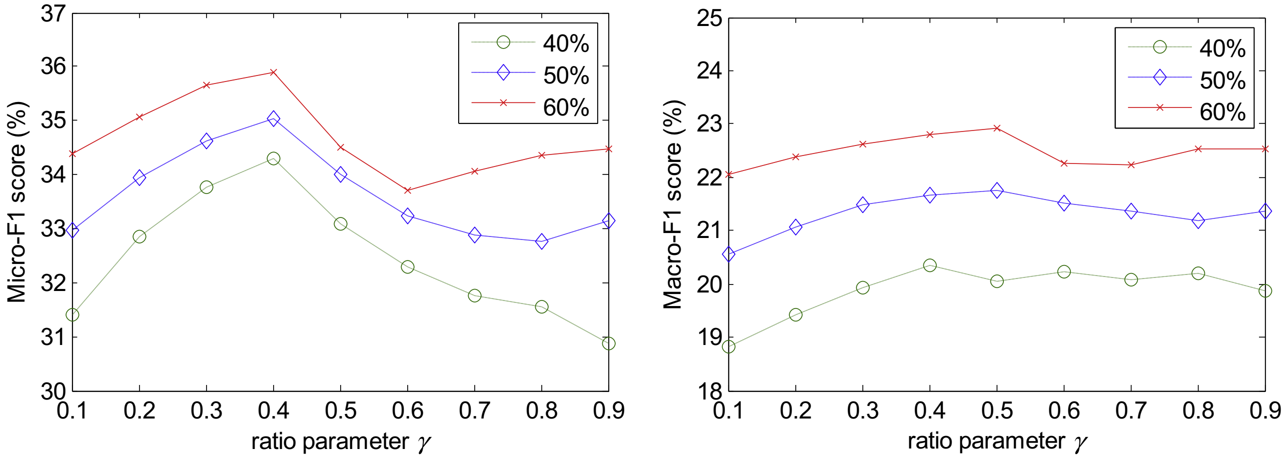


Fig. 3. Sensitivity study of MCPP on the Blogcatalog dataset.

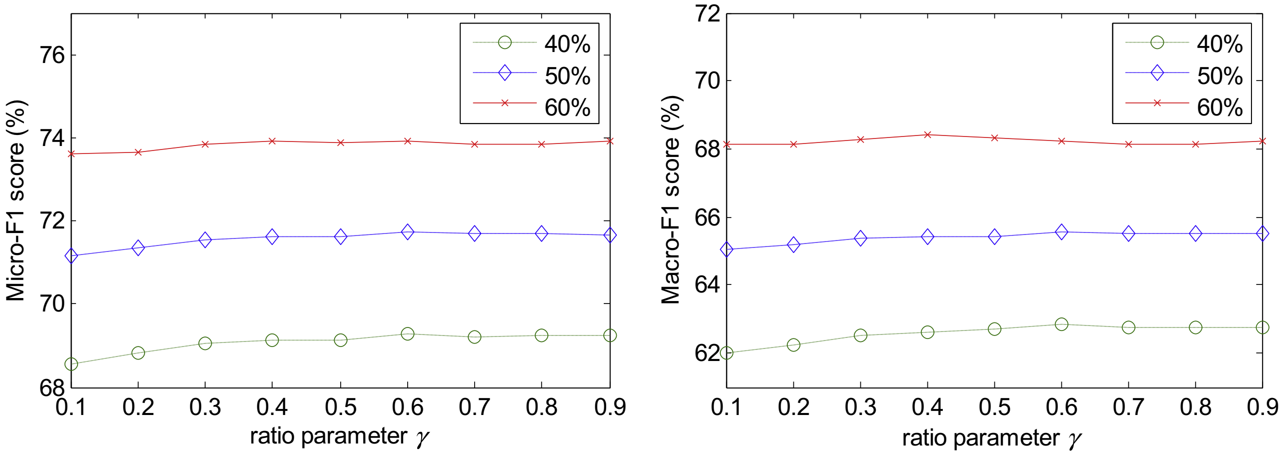


Fig. 4. Sensitivity study of MCPP on the DBLP dataset.

collective behaviors in social networks. We have proposed a causation-based multi-label relational classifier (MCPP) to predict collective behavior. MCPP uses causal analysis to distinguish the effect of peer influence from personal preference on one's behaviors based on networked data. Then the findings are used to design a multi-label relational classifier which estimates the behaviors of an individual. We have also made an improvement in classifier design. In order to reduce label noise, MCPP weights low degree nodes more than higher degree nodes. These help MCPP in achieving better classification performance. We have further studied the heterogeneity across subpopulations in a social network and proposed the MCPPS algorithm, which uses conditional average causal effect to estimate the impacts of peer influence and personal preference in different subpopulations. Empirical studies on real-world datasets demonstrate that our proposed approaches improve classification performance on networked data.

In this paper, our predictive algorithm of individual adoption behaviors only accounts for peer influence (α) and personal preference ($\beta = 1 - \alpha$). In reality other factors may affect adoption behaviors, and the model, we may have the third factor to capture the influence of other factors. In our future research, we intend to develop a more comprehensive model that considers the third factor influencing individual adoption behaviors in networks.

Another limitation of our study is that, the treated node is defined as he/she has one or more neighbors with label c. The peer influence with few neighbor having label c may be different from that with many neighbors having label c. We plan to explore how to use the propensity score matching procedure for continuous variables to model the peer influence more precisely in future. Also, the key to the success of the proposed method is the estimation of the causal effect of peer influence, and for accurate estimation of the causal effect needs the accurate calculation of propensity scores for matching. However, the estimation of causal effect could be biased when there are unmeasured confounders or wrong variables are included in propensity score estimation. There is need to improve the proposed methods by adopting new advanced causal effect estimation methods in future.

Acknowledgments

This work was supported by ARC Discovery grant DP170101306, the National 973 of China under grant 2013CB329604, the National Key Research and Development Program of China under grant 2016YFB1000901, the National Natural Science Foundation of China (NSFC) under grant 61503114, the SSSTC Research-innovation Community (RiC) on Big Data Research (RIC 01-032014), the ZUNIV Fonds zur Frderung des Akademischen Nachwuchses (FAN), and the China Scholarship Council.

References

[1] L. Tang, X. Wang, H. Liu, Scalable learning of collective behavior, IEEE Trans. Knowl. Data Eng. 24 (2012) 1080–1091.
[2] B. Taskar, P. Abbeel, D. Koller, Discriminative probabilistic models for relational data, Proceeding of the Eighteenth Conference of Uncertainty in Artificial Intelligence, (2002), pp. 485–492.
[3] L. Getoor, B. Taskar, Introduction to Statistical Relational Learning, The MIT Press, 2007.
[4] S. Macskassy, F. Provost, A simple relational classifier, Proceedings of the Second Workshop on Multi-Relational Data Mining at Ninth ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, (2003), pp. 64–76.
[5] J. Neville, D. Jensen, Iterative classification in relational data, Proceedings of the Workshop on Learning Statistical Models from Relational Data at the Seventeenth AAAI National Conference on Artificial Intelligence, (2000), pp. 42–49.
[6] Q. Lu, L. Getoor, Link-based classification, Proceedings of the Twentieth International Conference on Machine Learning, (2003), pp. 496–503.
[7] J. Neville, D. Jensen, L. Friedland, M. Hay, Learning relational probability trees, Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, (2003), pp. 625–630.
[8] S. Aral, L. Muchnik, A. Sundararajan, Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks, Proceedings of the National Academy of Sciences, 106 (2009), pp. 21544–21549.
[9] K. Imai, M. Ratkovic, Estimating treatment effect heterogeneity in randomized program evaluation, Ann. Appl. Stat. 7 (2013) 443–470.
[10] J.E. Brand, J. Thomas, Causal effect heterogeneity, The Series Handbooks of Sociology and Social Research, Springer, 2013.
[11] S. Morgan, Counterfactuals, causal effect heterogeneity, and the catholic school effect on learning, Sociol. Educ. 74 (2001) 341–374.
[12] J.E. Brand, C. Halaby, Regression and matching estimates of the effects of elite college attendance on educational and career achievement, Soc. Sci. Res. 35 (2006) 749–770.
[13] A.S. Gerber, D.P. Green, Field experiments and natural experiments, The Oxford Handbook of Political Methodology, Oxford University Press, 2008.
[14] X. Wang, G. Sukthankar, Multi-label relational neighbor classification using social context features, Proceedings of the Nineteenth ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, (2013), pp. 464–472.
[15] D. Rubin, Causal inference using potential outcomes, J. Am. Stat. Assoc. 100 (2005) 322–331.
[16] Y. Xie, J. Brand, B. Jann, Estimating heterogeneous treatment effects with observational data, Sociol. Methodol. 42 (2012) 314–347.
[17] P.R. Rosenbaum, D. Rubin, The central role of the propensity score in observational studies for causal effects, Biometrika 70 (1983) 41–55.
[18] G.W. Imbens, J.M. Wooldridge, Recent developments in the econometrics of program evaluation, J. Econ. Lit. 47 (2005) 5–86.
[19] S. Nandanwar, M.N. Murty, Structural neighborhood based classification of nodes in a network, Proceedings of the Twenty Second ACM SIGKDD international Conference on Knowledge Discovery in Data Mining, (2016), pp. 1085–1094.
[20] L. Tang, H. Liu, Relational learning via latent social dimensions, Proceedings of the Fifteenth ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, (2009), pp. 817–826.
[21] H. Daniel, K. Imai, G. King, E. Stuart, MatchIt: nonparametric preprocessing for parametric causal inference, J. Stat. Softw. 42 (2011) 1–28.
[22] S. Xu, X. Yang, H. Yu, Multi-label learning with label-specific feature reduction, Knowl. Based Syst. 104 (2016) 52–61.
[23] M.L. Zhang, Z.H. Zhou, A review on multi-label learning algorithms, IEEE Trans. Knowl. Data Eng. 26 (2014) 1819–1837.
[24] Q. Wu, M.K. Ng, Y. Ye, Multi-label collective classification via Markov chain based learning method, Knowl. Based Syst. 63 (2014) 1–14.
[25] A. McCallum, Multi-label text classification with a mixture model trained by EM,

Proceedings of the AAAI'99 Workshop on Text Learning, (1999).

[26] B. Al-Salemi, S. Noah, M. Aziz, RFBoost: an improved multi-label boosting algorithm and its application to text categorisation, Knowl. Based Syst. 103 (2016) 104–117.

[27] Z. Barutcuoglu, R.E. Schapire, O.G. Troyanskaya, Hierarchical multi-label prediction of gene function, Bioinformatics 22 (2006) 830–836.

[28] A. Goldberg, X. Zhu, S. Wright, Dissimilarity in graph-based semi-supervised classification, Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics, (2007), pp. 155–162.

[29] R. Heatherly, M. Kantarcioglu, X. Li, Social network classification incorporating link type, Proceedings of the IEEE International Conference on Intelligence and Security Informatics, (2009), pp. 19–24.

[30] B. Perozzi, R. Al-Rfou, S. Skiena, DeepWalk: online learning of social representations, Proceedings of the Twentieth ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, (2014), pp. 701–710.

[31] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, Q. Mei, Line: large-scale information network embedding, Proceedings of the Twenty Fourth International Conference of World Wide Web, (2015), pp. 1067–1077.

[32] D. Zhou, O. Bousquet, T.N. Lal, J. Weston, B. Scholkpf, Learning with local and global consistency, Adv. Neural Inf. Process. Syst. 16 (2004) 321–328.

[33] S. Aral, L. Muchnik, A. Sundararajan, Engineering social contagions: optimal network seeding in the presence of homophily, Netw. Sci. 1 (2013) 125–153.

[34] J. Friedman, T. Hastie, R. Tibshirani, The Elements of Statistical Learning, Springer Series in Statistics, Springer., 2001. 1

[35] M.P.H. Stumpf, C. Wiuf, R.M. May, Subnets of scale-free networks are not scale-free: Sampling properties of networks, Proceedings of the National Academy of Sciences of the United States of America, 102 (2005), pp. 4221–4224.

[36] J. Leskovec, C. Faloutsos, Sampling from large graphs, Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, (2006), pp. 631–636.

[37] J. Leskovec, J. Kleinberg, C. Faloutsos, Graphs over time: densification laws, shrinking diameters and possible explanations, Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, (2005), pp. 177–187.

[38] S. Boughorbely, J.P. Tarel, N. Boujemaa, Generalized histogram intersection kernel for image recognition, Proceedings of the IEEE International Conference on Image Processing, (2005), pp. 161–165.

[39] S. Chakrabarti, B. Dom, P. Indyk, Enhanced hypertext categorization using hyperlinks, Proceedings of the ACM SIGMOD International Conference on Management of Data, (1998), pp. 307–318.

[40] J.S. Yedidia, W.T. Freeman, Y. Weiss, Constructing free energy approximations and generalized belief propagation algorithms, IEEE Trans. Inf. Theory 51 (2005) 2282–2312.

[41] M.R. Boutell, J. Luo, X. Shen, C.M. Brown, Learning multi-label scene classification, Pattern Recognit. 37 (9) (2004) 1757–1771.

[42] Z. Chen, Z. Chi, H. Fu, D. Feng, Multi-instance multi-label image classification: a neural approach, Neurocomputing 99 (2013) 298–306.

[43] K. Zhao, H. Zhang, Z. Ma, Y. Song, J. Guo, Multi-label learning with prior knowledge for facial expression analysis, Neurocomputing 157 (2015) 280–289.

[44] M.L. Zhang, J.M. Pea, V. Robles, Feature selection for multi-label naive Bayes classification, Inf. Sci. 179 (19) (2009) 3218–3229.

[45] K. Trohidis, G. Tsoumakas, G. Kalliris, I. Vlahavas, Multilabel classification of music into emotions, Proceedings of the Ninth International Conference on Music Information Retrieval, (2008), pp. 325–330.

[46] D. Koller, A. Pfeffer, Probabilistic frame-based systems, Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence, (1998), pp. 580–587.

[47] B. Gallagher, T. Eliassi-Rad, Leveraging label-independent features for classification in sparsely labeled networks: an empirical study, Advances in Social Network Mining and Analysis, Springer, 2010, pp. 1–19.

[48] K. Henderson, B. Gallagher, L. Li, L. Akoglu, T. Eliassi-Rad, H. Tong, C. Faloutsos, It's who you know: graph mining using recursive structural features, Proceedings of the Seventeenth ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, (2011), pp. 663–671.

[49] L. Tang, S. Rajan, V.K. Narayanan, Large scale multi-label classification via metalabeler, Proceedings of the Eighteenth International Conference on World Wide Web, (2009), pp. 211–220.

[50] F. Lin, W.W. Cohen, Semi-supervised classification of network data using very few labels, Proceedings of the International Conference on Advances in Social Networks Analysis and Mining, (2010), pp. 192–199.

[51] L. Page, S. Brin, R. Motwani, T. Winograd, The pagerank citation ranking: Bringing order to the web, Technical Report 1999-66, Stanford InfoLab, 1999.