A Novel Group Detection Method for Finding Related Chinese Herbs

LIDONG WANG¹, YIN ZHANG² AND XIAODONG XU³

¹Hangzhou Normal University
Zhejiang, 310018 P.R. China
²College of Computer Science and Technology
Zhejiang University
Hangzhou, Zhejiang, 310027 P.R. China
³Zhejiang Chinese Medical University
Zhejiang, 310053 P.R. China

Email: violet_wld@163.com; yinzhi@zju.edu.cn; xdong_xu@126.com

In past decades, TCM (Traditional Chinese Medicine) has been widely researched through various methods in computer science, but none digs into huge amount of ancient TCM prescriptions and endless digital TCM information to display the compatible and incompatible relationship among herbs. To meet the challenge and to mine the groups of compatible herbs for further drug exploitation, we explore the property of herbal networks and introduce a novel community detection algorithm concerning both herbal attributes and graph structural factors. First, we calculate the attribute similarity for each paired herbs to construct the herbal graph. Then, a novel community detection algorithm named RWLT (Random Walk & Label Transmission) is proposed to detect herbal groups with near-linear time. The performance of RWLT has been rigorously validated through comparisons with representative methods against randomly created networks, real-world networks and herbal networks. According to the TCM expert, our method is capable of finding groups of Chinese herbs with intensive correlation, and is also able to separate the herbs with mutual incompatibility to be excluded into different communities.

Keywords: herbal group detection, community detection, random walk, label transmission, RWLT algorithm

1. INTRODUCTION

Traditional Chinese Medicine (TCM) is a treasure of Chinese people, and it has been recognized as a popular complementary and alternative medicine in Western countries. The major concern in TCM is how to consolidate and integrate the data to enable efficient discovery of novel knowledge from the dispersed data. There are about 12,800 currently known herbs and a large number of TCM digital books on the web. A comprehensive study of several herbs might require a researcher to search and learn many books. Merely locating all relevant books by using a simple search utility would be inefficient and time consuming. The current condition of publications or organizations in the TCM field makes extensive collaboration and complete information acquisition difficult.

Received January 8, 2014; revised May 2 & July 8, 2014; accepted September 22, 2014. Communicated by Chih-Ping Wei.
1 Corresponding author: Lidong Wang.
* This work was supported by Zhejiang Provincial Natural Science Foundation of China under Grant No. Q14-F020032, Chinese Knowledge Center for Engineering Science and Technology (CKCEST), Specialized Research Fund for the Doctoral Program of Higher Education (SRFDP) (20130101110136), and China Academic Digital Associative Library (CADAL).
Therefore, useful information in TCM needs to be effectively organized and mined from digital books for users.

In recent works, several data mining techniques have been applied in TCM field, such as syndrome differentiation [1], herbal combinational rule mining [2] and symptom name normalization [3]. Most of these studies mainly adapt to exploring the relationship among four elements in TCM, which are the herb, the prescription, the symptom and the syndrome. As of now, few studies have been focused on the group detection for Chinese herbs. He et al. [4] proposed an herbal clustering algorithm based on K-means. The efficacy of each herb is converted to an n-dimensional vector $X$ with only numeric values. Let $X = \{X_1, X_2, \ldots, X_n\}$ be a set of attribute vectors of $n$ herbs. Then, the K-means algorithm partitions $X$ into $k$ clusters. Unfortunately, the research only clusters the herbs with similar efficacy, which is not enough to mine the latent relationship between herbs.

In TCM, some herbs have to be combined for the purpose of disease treatment, which is already known as the prescriptions. A prescription usually contains at least four to twenty herbs. The potency of a single herb is usually limited, but when two herbs are used together, they would interact with each other and display their superiority over a single herb in the treatment of diseases, we call that these herbs have combinational rule. In China, lots of herbs have intensive combinational rules that have been learned from ancient times to the modern period. For new drug exploitation, researchers will try to combine two or more herbs for clinical trials. Thus, it would be meaningful if the computer technology can be utilized to mine a group of herbs that have similar attributes and latent association. As mentioned above, traditional clustering algorithms, e.g. K Means, have been used to solve the herb clustering problem. However, these algorithms can only cluster herbs with similar attributes. We may estimate pairwise combination when four herbs A, B, C and D are clustered in one group by K-means, but this is unreasonable in TCM theory. For graphical presentation of group data, when A is correlated with B, and B is correlated with C, it can be estimated that A may be correlated with C (a common phenomenon in complex network), and they may have the potential of combination, which is referred to as “latent correlation”. Thus, how to mine the “latent correlation” is our main task.

Obviously, finding communities within a graph is an efficient way to identify groups of related vertices. To effectively find a group of related Chinese herbs, we introduce a method named RWLT for group detection on our created herbal graph. The RWLT for finding groups of related herbs comprises two steps: (1) The creation of herbal graph; (2) Group detection on the herbal graph. In the following Section 2 and Section 3, the details of these two steps are described. We invited a professor from Zhejiang Chinese Medicine University as the expert to provide source data for our experiments and to verify the experimental results.

2. CREATION OF HERBAL GRAPH

2.1 Degree of Correlation

The correlation between two herbs depends on two main aspects: (1) attribute similarity and (2) combinational rule. The attributes of each herb are always described with certain terms. For instance, “Nature” refers to the temperature characteristics of the herb,
such as hot (热), cold (寒), slightly cold (微寒), etc. “Flavor” refers to the taste property of the herb, such as sour (酸), bitter (苦), slightly bitter (微苦), etc. It has to be noted that the slightly difference between two attributes, such as “bitter (苦)” and “slightly bitter (微苦)” is difficult to quantify in the way of “0” or “1”. Hence, we compute the attribute similarity between two herbs through string matching algorithms.

The first issue of attribute similarity measuring is to extract structured information from digital books or TCM websites that can be used directly for knowledge discovery. Structured information involves prescriptions and three attributes of each herb, namely efficacy, nature & flavor and channel tropism. We use natural language-processing techniques to extract structured information (see Tables 1 and 2) from authoritative TCM digital books and famous TCM websites.

### Table 1. An extracted prescription and its corresponding herbs.

<table>
<thead>
<tr>
<th>Prescription</th>
<th>Guizhi Tang (桂枝汤)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Herbs</td>
<td>Cassia Twig (桂枝)</td>
</tr>
</tbody>
</table>

### Table 2. An extracted herb and its attributes.

<table>
<thead>
<tr>
<th>Herb</th>
<th>Herba Ephedrae (麻黄)</th>
</tr>
</thead>
<tbody>
<tr>
<td>efficiency</td>
<td>Inducing perspiration (发汗)</td>
</tr>
<tr>
<td>nature &amp; flavor</td>
<td>Spicy (辛)</td>
</tr>
<tr>
<td>channel tropism</td>
<td>Lungs (肺)</td>
</tr>
</tbody>
</table>

Let $S_e, S_n, S_f$ denote “efficacy”, “nature & flavor” and “channel tropism” respectively, the “efficacy” similarity between two herbs $h$ and $h'$ can be measured by Jaro-distance metric [3]:

$$sim(S_e, S'_e) = 1/3(m/|S_e| + m/|S'_e| + (m-t)/m)$$

where $m$ is the number of matching characters between string $S_e$ and string $S'_e$, $t$ is a transposition for matching characters in different order. According to the expert’s instruction, the importance of these three elements should have the following rules: nature & flavor > efficacy > channel tropism. Thus, the attribute similarity between two herbs is defined as follows:

$$sim(h, h') = \alpha sim(S_e, S'_e) + \beta sim(S_n, S'_n) + \gamma sim(S_f, S'_f)$$

where $\alpha = 0.5, \beta = 0.3, \gamma = 0.2$. It has to be noted that the weight for each component is defined based on repeated experiments and expert’s instruction. We randomly select 90 herbs and divided them into 3 groups, then calculate the similarity of each pair with different combination of weights, which are listed in Table 3. The results of attribute similarity are verified by 10 students major in TCM, and select the combination to make sure that the average result is closest to the core theory of TCM.
Besides the attribute similarity, another equally important issue for calculating the degree of correlation is to find out the combinational rules of paired herbs. When \( h \) and \( h' \) are frequently used in combination with each other in a prescription, they are more likely to have combinational rule, which is denoted as \( C(h, h) = 1 \), else \( C(h, h') = 0 \). The combinational rules of herbs can be extracted from prescription dataset by the FP-growth algorithm [5]. Most of the combinational rules mined by the FP-growth method have been proved to agree with the reality of TCM. Based on this, the degree of correlation between two herbs can be defined as follows:

\[
\text{Corr}(h, h') = \frac{1}{2}(0.5 \times \text{sim}(S_e, S'_e) + 0.3 \times \text{sim}(S_n, S'_n) + 0.2 \times \text{sim}(S_c, S'_c)) + \frac{1}{2} C(h, h') \quad (3)
\]

Finally, we define that if \( \text{Corr}(h, h') \) is larger than 0.6, two herbs are considered to be related. The threshold 0.6 is selected based on repeated experiments to make sure that most of the returned connected paired herbs have to agree with the real knowledge of TCM. We select 50 common herbs and obtain correlation information of these herbs according to the instructions of experts, which has 112 pair of correlated herbs. Then, an analysis of the experimental results listed in Table 4 is carried out for threshold selection. As shown in Table 4, the optimal threshold should be 0.6 since it achieves 121 pairs of correlated herbs, and the returned number of correlated pairs is closest to the prior knowledge.

### Table 4. Experimental results of threshold selection.

<table>
<thead>
<tr>
<th>threshold</th>
<th>returned number of correlated paired herbs</th>
<th>precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>205</td>
<td>111/205</td>
</tr>
<tr>
<td>0.2</td>
<td>198</td>
<td>110/198</td>
</tr>
<tr>
<td>0.3</td>
<td>162</td>
<td>108/162</td>
</tr>
<tr>
<td>0.4</td>
<td>161</td>
<td>105/161</td>
</tr>
<tr>
<td>0.5</td>
<td>132</td>
<td>105/132</td>
</tr>
<tr>
<td>0.6</td>
<td>121</td>
<td>103/121</td>
</tr>
<tr>
<td>0.7</td>
<td>86</td>
<td>85/86</td>
</tr>
<tr>
<td>0.8</td>
<td>53</td>
<td>53/53</td>
</tr>
<tr>
<td>0.9</td>
<td>12</td>
<td>12/12</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0/0</td>
</tr>
</tbody>
</table>

### 2.2 Creation of Herbal Graph

To globally present the relationship among all paired herbs, we create a herbal graph based on the degree of correlation mentioned above, using a graph \( G(N, E, w) \), where \( N \) is the number of vertices of the graph, \( E \) is the number of edges, and \( w \) is the weight of the edge. Each vertex in the graph represents a herb, and an edge exists be-
between two vertices if two herbs are related. The value of $\text{Corr}(h, h')$ represents the weight of the edge connecting two herbs. A part of the graph is illustrated in Fig. 1. When the degree of correlation between two vertices is larger, the edge is drawn with heavier line. For example, the edge between the node “Libanotus” (乳香) and the node “Myrrh” (没药) is drawn with heavier line than many other lines, since these two herbs have high attribute similarity and combinational rule.

Fig. 1. A part of the created herbal graph.

3. GROUP DETECTION ON HERBAL GRAPH

3.1 Related Works

The main goal of group detection for herbs is not only to cluster herbs with similar attributes, but also to find a group of herbs that can be combined for better curative effect. Group detection is an effective method that measures vertex closeness based on structural similarity (e.g., the number of common neighbors between two vertices) [6]. The group detection algorithm on graph is also called community detection in much literature [7-14]. As the result of community detection, there should be many edges within each group and relatively few between the groups.

In the literature, algorithms developed to detect groups in graph can generally be divided into three main categories: modularity-based methods [7-9], spectral algorithms [10], methods based on statistical inference [11, 12], and other alternative methods [13, 14]. Modularity-based methods have been largely used in recent studies. Although many researchers focus on solving the resolution limit problem in modularity-based methods, Granell [15] has demonstrated that no one can completely diminish the effect of this limit. Spectral algorithm has the limitation that it requires prior knowledge of the number of groups, which is impossible to obtain in our work.

Recently, lots of random walk based methods are proposed in networks of different structures [13, 16]. Andrea [17] has carried out a comparative analysis of the performance of above algorithms, and concluded that the random walk based methods perform rather better compared with others. Random walk is a useful way to find communities. Dongen [18] described a random walk based algorithm named Markov Cluster Algo-
algorithm (MCL), which simulates a peculiar process of flow diffusion in a graph. As of now, the MCL is one of the most used community detection algorithms. However, the algorithm should scale as \(O(n^3)\) \((n\) is the node number in the graph). Pons [19] used random walk to define a distance measure between vertices. The distance is calculated from the probabilities that the random walker moves from a vertex to another in a fixed number of steps. Vertices are then grouped into communities through an agglomerative hierarchical clustering technique. The algorithm runs to completion in a time \(O(n^2d)\) on a sparse graph, where \(d\) is the depth of the dendrogram. Hu [20] designed a graph clustering technique based on signaling process with random walk scheme between vertices. The complexity of the algorithm is \(O((k+1)n^2)\), where \(k\) is the average degree of the graph.

From the above researches, we can see that most of recent methods based on random walk have high time complexity. Although there are several near-linear algorithms, such as LPA [21], FEC [13], et al., they are mostly applicable to special networks with good/obvious community structure. Considering the property of low clustering coefficient (see Section 4.2), the algorithm applicable to herbal graphs should have a good trade-off between effectiveness and efficiency, which remains challenging. Thus, we propose a novel algorithm based on random walk named RWLT that takes a near-linear time to run and can achieve ideal performance for herbal networks. The main idea of RWLT is as followings: suppose that a node has a label \(c\), then it can transmit its label to other nodes supposed to be in one group without stochastic process. In other words, the label shall be transmitted through a certain route. The algorithm should consider two questions: (1) How to establish the transmission route? (2) How to find the cutoff node when the label transmits through the certain route? To overcome these two problems, RWLT contains two corresponding phases which are Random Walk (RW) phase and Label Transmission (LT) phase.

### 3.2 Random Walk (RW) Phase

#### 3.2.1 Algorithm overview

An imaginary random walker walks freely from one node to another, following the links of a given graph. The walker’s route can be viewed as a stochastic process defined based on the links’ attributes. In particular, when the walker arrives at a node, it will select one of its neighbors at random and then go there.

Let \(X = \{X_l, l \geq 0\}\) denote a random walk series. Let \(P\{X_1 = N_i, 1 \leq N_i \leq n\}\) be the probability that the walker will arrive node \(N_i\) after going exactly \(l\) steps. \(X\) is a discrete Markov chain if we have:

\[
P\{X_l = N_i \mid X_0 = N_0, X_1 = N_1, \ldots, X_{l-1} = N_{l-1}\} = P\{X_l = N_i \mid X_{l-1} = N_{l-1}\}.
\]

(4)

Suppose node \(j\) is the neighbor of node \(i\). Let \(T\) be the transfer matrix, where \(T_{i \to j}\) be the probability of the walker walking from node \(i\) to its neighbor node \(j\). In a weighted herbal network, this probability can be computed as following:

\[
T_{i \to j} = \frac{W_{ij}}{\sum_j W_{ij}} = \frac{\text{Corr}(h_i, h_j)}{\sum_j \text{Corr}(h_i, h_j)}
\]

(5)

An imaginary random walker walks freely from one node to another, following the links of a given graph. The walker’s route can be viewed as a stochastic process defined based on the links’ attributes. In particular, when the walker arrives at a node, it will select one of its neighbors at random and then go there.
where $W_{ij}$ represents the weight of link $(i, j)$, $\sum_{j} W_{ij}$ is the weighted degree of node $i$. According to the homogeneous Markov chain, we have:

$$P(X_i = j \mid X_{i-1} = i) = T_{i\rightarrow j},$$

(6)

Then, let $T^l_i(i)$ be the probability that the walker starting from node $i$ can eventually arrive at a specific destination node $t$ with exactly $l$ steps. The main idea of random walk is that the probability a walker starts from any node and stays in the same community after a number of transitions, is greater than that the walker goes out to a different community. Based on this observation, we can find groups by examining localized and aggregated transition probabilities. The aggregated transition probability $T^l_i(i)$ can be estimated iteratively by

$$T^l_i(i) = \sum_{j \neq i} T_{i\rightarrow j} \cdot T^{l-1}_i(j).$$

(7)

The initialization for $T^l_0(i)$ should be: $T^0_i(i) = 0 \cdot I_{reg} + 1 \cdot I_{set}$. The rule for group detection is that walkers that start from nodes within the community of the destination node should reach the destination node easier within $l$ steps since more paths can be chosen. Mathematically speaking,

$$\sum_{i=1}^{n} T^l_{i}(i) > \sum_{i=1}^{n} T^l_{i}(k), \quad \text{for } i \in C, k \notin C,$$

(8)

where $C$ denotes the community containing destination node $t$. Based on this idea, the procedure for RW phase can be designed as follows:

**Step 1:** Calculate $T^l_i(i)$ for each node $i$;

**Step 2:** Rank all the nodes according to their associated value of $\sum_{l=1}^{l} T^l_i(i)$.

The algorithm for calculating transfer matrix $T$ is given as follows:

**Algorithm 1:** Computing transfer matrix $T$

Input: $A$, the adjacency matrix of a network; $t$, destination node; $l$, number of steps;

Output: $T$, transfer matrix;

1 for $i=1:n$
2 $T^n_0(i) = 0 \cdot I_{reg} + 1 \cdot I_{set}$;
3 end;
4 for $l = 1:l$
5 for $i = 1:n$
6 $T^l_i(i) = \sum_{j \neq i} T_{i\rightarrow j} \cdot T^{l-1}_i(j)$;
7 end;
8 end;
9 return $T$;
As shown above, $l$ is an important parameter required in this phase. The value of $l$ should not be set too small since walking only for a few steps will exclude the walkers that are a bit far from the destination node. Furthermore, when the value of $l$ is set too large, the nodes located at a bit far place from destination node will obtain a similar transition probability, which makes the ranking process (Step 2) be meaningless. Here, the value of $l$ that we are considering is set around the average distance between nodes within a network.

3.2.2 Destination node selection

As discussed above, the aggregated transition probability should be calculated through the introducing of destination node. Can the destination node be selected randomly?

![Fig. 2. A simple network.](image)

As shown in Fig. 2, the simple network contains two communities in different colors. When we set node 5 as destination node, node 6 has higher probability of eventually arriving at the destination within $l$ ($l = 3$) steps than node 1. Mathematically speaking, $\sum T_i(6) > \sum T_i(1)$. This means that when all the nodes are ranked according to $T_i(t)$, node 6 would have higher probability to be clustered in one group that node 5 belongs to. However, when node 3 is set as destination node, node 1 would have higher probability of eventually arriving node 3. Mathematically speaking, $\sum T_i(6) < \sum T_i(1)$, which conforms to the real community structure. This indicates that different destination node would lead to different community detection result. Selecting the nodes between communities as destination nodes may make walkers that start from nodes outside the community have a much higher probability of eventually arriving at the destination, which disturbs the identification of community structure. Therefore, it would be better to select destination node by avoiding the nodes located between communities. Radicchi [22] considered that, in many cases, edges connecting nodes in different communities are included in few or no triangles (or rectangles). On the other hand, many triangles (or rectangles) exist within clusters. Based on this, our paper proposes an automatic destination node selecting method which is listed as follows:

**Step 1:** Compute the degree of all nodes $d_i$ in network $G$.

**Step 2:** Let $r = \arg \max d_i$, $k = \arg \max d_i$. If $d_r = 2$, then set the node $r$ as the destination node; If $d_k = 1$, then set node $k$ as the destination node; else go to Step 3.

**Step 3:** For each pair of connecting nodes $\{m, n\}$, compute the number of triangles or rectangles $N_{m,n}$ containing both $m$ and $n$.

**Step 4:** Let $\{u, v\} = \arg \max_{\{m,n\}} N_{m,n}$, then randomly select the destination node $t$ from $\{u, v\}$.

In Step 2, it can easily be proved that the network $G$ is presented as lines or circles when $d_r = 2$. Thus, it is reasonable to set node $r$ as the destination node. When the degree
of node $k$ is 1, node $k$ will certainly not connect the nodes in other communities. In Step 4, the destination node is selected based on the principle proposed by Radicchi [22].

3.2.3 Label transmission (LT) phase

Based on the ranked node list obtained from RW phase, the community structure of the destination node is revealed. The community that contains the specified destination can be distilled by properly transmitting the label through the ranked node list. During the label transmission phase, a node is initialized with a label which then transmits step by step via the sorted list of nodes based on their corresponding values in $T'_{\ell}(i)$. The key issue is to find the cutoff node where the transmission process stops. The process of label transmission has to ensure that the transmitted nodes can make up a strong community structure. Inspired by the definition of “strong community” [22], the main idea of label transmission is defined as the follows. Suppose that node $i$ carries a label denoting the community to which it belongs to, the label of $i$ has to be influenced by the labels that the maximum number of its neighbors have. If $C_1, C_2, \ldots, C_p$ are the labels that are currently active in the network, $N^{C_i}$ is the number of neighbors node $i$ has with label $C_i$, $N^{\overline{C_i}}$ is the number of neighbors of nodes $i$ that does not have label $C_i$, and $d_i^{in}, d_i^{out}$ are the degrees of node $i$ within and outside of its community $U$. Then we describe the LT phase in the following. Suppose a ranked node list $M = \{m_1, m_2, \ldots, m_n\}$. Firstly, initialize the label of node $m_1$ as the label $C_k$. Then, the label $C_k$ is transmitted, starting from the top node $m_1$, through $M$ in order. The transmission algorithm stops if the nodes inside the community (labeled with $C_k$) and outside the community meet the following conditions simultaneously:

\begin{align}
    \text{Condition 1:} & \quad N_i^{C_k} \leq N_i^{\overline{C_k}} \quad i \in \{\text{the nodes directly connect } U\} \& i \notin U \\
    \text{Condition 2:} & \quad d_i^{in} > d_i^{out} \quad \forall i \in U
\end{align}

At the end of the process nodes with the same label are grouped together as communities.

The Condition 1 holds for the nodes outside the community and located between two communities, and requires them to have at least as many neighbors outside the community $U$ as it has with the neighbors inside. The stopping criterion in Condition 2 requires each node in community $U$ to have strictly more neighbors within its community than outside. Thus, the stopping criterion takes into account both inside and outside topological structure of the detected group.

3.2.4 Overview of RWLT algorithm

Table 5 provides an overview of our proposed algorithm RWLT for identifying hidden groups in a graph. Note that the algorithm is recursive, and the nodes that have not been transmitted have to return to the RW phase for another group extraction.

**Proposition 1:** The time complexity of the RWLT algorithm for group mining from a graph is bounded by $O(K(l(n + m)))$, where $K$ is the number of groups detected, $l$ is the number of steps that the walker travels, $n$ and $m$ correspond to the numbers of nodes and links of the network, respectively.
Table 5. The RWLT algorithm for group detection.

<table>
<thead>
<tr>
<th>Algorithm 2: RWLT(A)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> A, the initial adjacency matrix of a network;</td>
</tr>
<tr>
<td><strong>Output:</strong> ﾘ, the detected group set;</td>
</tr>
<tr>
<td>1 ( M = \text{RW}(A) ); // ( M ) represents the sorted node list.</td>
</tr>
<tr>
<td>2 ([R, \text{pos}] = \text{LT}(M)); // \text{pos} represents the index of cutoff point in ( M ).</td>
</tr>
<tr>
<td>3 ( \mathcal{R} = \mathcal{R} \cup R );</td>
</tr>
</tbody>
</table>
| 4 \text{if \( \text{pos} = \text{length}(M) \) then return;}
| 5 \text{calculate adjacent matrix } A' \text{ of network } G(N', E'), \text{where } N' = M[\text{pos} + 1: \text{length}(M)]; \text{//} N' \text{ represents the nodes that have not been transmitted.}
| 6 \text{RWLT}(A'); |

**Proof:** (1) During RW phase, computing \( T^k_i(i) \) is the most expensive step. As shown in Algorithm 1, it takes \( O(n) \) time for initialization of \( T^0_i(i) \). In step 6, it would take \( O(d_i) \) time for calculating \( T^k_i(i) \), where \( d_i \) denotes the degree of node \( i \). Thus, the time complexity of calculating \( T^k_i(i) \) for all nodes (from steps 4 to 8) is

\[
1 \cdot \sum_{i=1}^{n} O(d_i) = O(lm). \quad (10)
\]

Besides, we use \( O(n) \) time to sort nodes by counting sort algorithm, and also use about \( O(m + n) \) time to conduct destination node selection algorithm. Thus, the overall time complexity of RW phase is \( O(lm + n + 3n) \).

(2) The LT phase takes a near-linear time for the algorithm to run to its completion. Initializing every node with unique labels requires \( O(n) \) time. At each node \( x \), it requires a worst-case time of \( O(N_x) \) for each Condition, where \( N_x \) denotes the neighbors of node \( i \). This process does not need to traverse through all nodes, and hence an overall worst-case time is \( O(m) \). Thus, the overall time required by the LT phase is \( O(m + n) \).

(3) Let \( T = T(1) + T(2) + T(5) \), where \( T(1) \), \( T(2) \) and \( T(5) \) are the time required by Step 1, 2, 5 depicted in Table 5. Because \( T(5) < O(n) \), we have \( T(1) + T(2) + T(5) = O(l(n + m)) \). It is shown in Table 5 that RWLT algorithm is recursive. Suppose that \( K' \) is the total number of times recursively calling RWLT, and \( K \) is the number of detected groups. It can easily be shown that \( K' = K - 1 \). Thus, we have

\[
K'(O(l(m + n))) < O(Kl(m + n)). \quad (11)
\]

Therefore, the total time complexity of RWLT is bounded by \( O(Kl(n + m)) \).

4. MATERIALS

The method described above was implemented in MatlabR2012a and was applied to two herbal networks. Besides, we also tested it with the LFR benchmark and real-world networks.
4.1 The LFR Benchmark and Real-world Networks

The LFR benchmark is a case of manually and randomly created network, in which groups are of different sizes and nodes which have different degrees. We define the network as: $G(\text{Num_nodes}, \text{average}_k, \text{max_degree}, c_{\text{min}}, c_{\text{max}}, e_1, e_2, \mu)$, which creates various network in different structures. Num_nodes denotes the total number of nodes, average_k denotes average degree, max_degree means maximum degree, $e_1$ and $e_2$ are the exponent of the degree distribution and community size distribution respectively. $c_{\text{min}}$ and $c_{\text{max}}$ are the minimum and maximum sizes of communities respectively. Mixing parameter $\mu$ expresses the ratio between the external degree of node with respect to its community and the total degree of the node. We notice that communities are well defined when $\mu$ gets small. For a thorough analysis, we consider various versions of the benchmark with different $\mu$. The parameters for the model are set as follows: $G(1000, 20, 50, c_{\text{min}}, 5 \times c_{\text{min}}, -2, -1, \mu)$, where $c_{\text{min}}$ is set to 10 or 20, $\mu$ varies from 0 to 0.5.

We also applied our algorithm to the following real-world networks: (1) dolphin network [23], an undirected social network of frequent associations between 62 dolphins in a community living off Doubtful Sound; (2) US college football network [24], which consists of 115 college teams represented as nodes and has edges between teams that played each other during the regular season in the year 2000; (3) Facebook social-network, obtained from 10 ego-networks, consisting of 193 circles and 4093 users [25]. Facebook data is fully labeled, in the sense that each circle is considered to be a cohesive community.

4.2 Herbal Networks

We constructed two herbal networks to validate the effectiveness of our algorithm. The first dataset was extracted from digital books. According to the expert, we selected two authoritative digital books, 《Zhong Hua Yao Dian》 (中华药典) and 《Fang Ji Da Ci Dian》 (方剂大辞典), as the source of herbal dataset and prescription dataset. Herbal dataset extracted from 《Zhong Hua Yao Dian》 involves 474 common herbs and their relative attributes. Prescription dataset extracted from 《Fang Ji Da Ci Dian》 involves about 100,000 prescriptions. The output of combinational rule mining algorithm (FP-growth) involves about 6500 paired herbs. The result of attribute similarity calculation from herbal dataset, taking into consideration the combinational rules of these herbs, contains 642 paired herbs. Thus, the final created herbal graph contains 642 edges and 332 vertices (herbs). All of these herbs are classified manually into 21 main categories and 49 sub-categories according to the expert. We denote this herbal network as HN1.

The second herbal dataset was extracted by web crawler from TCM websites. The final herbal network constructed from this dataset contains 3390 edges and 1138 vertices. These 1138 herbs are classified into 84 categories according to expert’s instruction. We denote this network as HN2. The topological graphs of the two networks are shown in Fig. 3.

The resulting two networks have the similar property to scale-free network. The property of scale-free means the resulting graph has a power law distribution in its Degree, that is, the frequency of Degree $k$ is given by $p_k = k^{-r}$, where $r$ is the power law exponent. The property of scale free is shown in Fig. 4. The number of vertices (y-axis) is
plotted against the degree of vertex. Moreover, the average clustering coefficient of HN1 and HN2 are 0.305 and 0.241 respectively, which denotes the herbal networks do not represented as good community structure.

Fig. 3. Two herbal networks.
5. RESULTS

We wanted to have a representative subset of algorithms, which exploit some of the most interesting ideas and techniques that have been developed over the recent years. Apparently we could not perform analysis for all existing techniques since their number is huge. Thus, in our experiment, we considered several state-of-the-art algorithms for comparison, such as GN [24], CNM [8], CFinder [26], FEC, MCL, LPA [21] and Infomap [27]. These methods can be divided into three main categories: modularity based algorithm (GN and CNM), random walk based algorithm (FEC, MCL, RWLT and Infomap) and other alternative methods (CFinder and LPA). As reported by [17], Infomap is one of the most competitive methods. The CFinder is a local algorithm that looks for communities that may overlap, i.e. share nodes. The LPA is conducted based on label propagation, in which the label is propagated with iterative ways until the label of all nodes reaches the stable state. It has to be noticed that although our LT phase is proposed based on label transmission, the idea in the LT phase is intrinsically different from LPA. In our method the label propagates from destination node to the cutoff node along a certain route, while in LPA, each node is initialized with a unique label that has to be iteratively updated according to the criterion that most of its neighbors currently have, and there is no certain route for LPA.

To fully evaluate our proposed method, two criteria are considered here, which are the Accuracy and the NMI (Normalized Mutual Information) [28].

5.1 Results on Randomly Created Networks

In Fig. 5 it shows the results of our experiments. Each point of every curve corresponds to an average over 10 realizations of the network. The variable on x-axis is the mixing parameter  \(\mu\), the value on y-axis denotes NMI.

As shown in Fig. 5, the difference in the performance of the algorithm is remarkable. Most methods perform well, although all of them start to fail when  \(\mu\) is close to 0.5. The CFinder fails to detect the communities even when  \(\mu\approx 0\), which was already known from the literature. The GN algorithm performs about as well as the MCL, but inferior to CNM. CNM performs rather well, but does not achieve ideal performance when  \(\mu\) is small. Among four random walk based algorithms (MCL, RWLT, Infomap and FEC), there is big difference between the MCL and the others. For the networks with various structures, the RWLT and Infomap perform relatively better than others on average. The MCL does not have a remarkable performance than other kind of methods, as it starts to fail when  \(\mu > 0.3\).
5.2 Results on Real-world Networks

We averaged the value of NMI and cluster number over 10 realizations in each test, and the experimental result are shown in Table 6. As shown, modularity based methods (GN and CNM) do not have high NMI scores because a high modularity might not necessarily result in a true partitioning. The Cfinder has a rather poor performance on both randomly created networks and real-world networks. However, the random walk based methods do not all perform well. The FEC are outperformed by the GN and LPA in the result of football network. Moreover, we notice that the FEC does not outperform as much as it does on randomly created network. This is due to the fact that the community structure in real-world networks is not well defined. This phenomenon clarifies that the results of FEC algorithm is sensitive to the structure of networks. Meanwhile, the performance of MCL differs considerably from three networks. The RWLT and Infomap perform better than other algorithms and demonstrate the best overall ranks in terms of NMI. Although Infomap performs well for both real-world networks and random created networks, it has to optimize the objective function during random walk phase, which would result in the decrease of efficiency.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Dolphin network NMI(%)</th>
<th>Cluster number</th>
<th>US college football network NMI(%)</th>
<th>Cluster number</th>
<th>Facebook social network NMI(%)</th>
<th>Cluster number</th>
<th>Average-rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>GN</td>
<td>44.17(6)</td>
<td>4</td>
<td>87.92(6)</td>
<td>10</td>
<td>53.12(4)</td>
<td>16</td>
<td>5.33</td>
</tr>
<tr>
<td>CNM</td>
<td>44.43(5)</td>
<td>5</td>
<td>88.93(5)</td>
<td>7</td>
<td>53.04(5)</td>
<td>18</td>
<td>5</td>
</tr>
<tr>
<td>LPA</td>
<td>52.29(4)</td>
<td>6.5</td>
<td>89.20(4)</td>
<td>11.2</td>
<td>60.72(1)</td>
<td>18.68</td>
<td>3</td>
</tr>
<tr>
<td>Cfinder</td>
<td>43.21(7)</td>
<td>5</td>
<td>75.31(8)</td>
<td>10</td>
<td>48.23(7)</td>
<td>14</td>
<td>7.33</td>
</tr>
<tr>
<td>MCL</td>
<td>42.41(8)</td>
<td>13</td>
<td>92.35(2)</td>
<td>16</td>
<td>41.03(8)</td>
<td>15</td>
<td>6</td>
</tr>
<tr>
<td>FEC</td>
<td>52.93(3)</td>
<td>4</td>
<td>80.27(7)</td>
<td>9</td>
<td>52.30(6)</td>
<td>15</td>
<td>5.33</td>
</tr>
<tr>
<td>Infomap</td>
<td>56.60(2)</td>
<td>3</td>
<td>92.38(1)</td>
<td>12</td>
<td>58.42(3)</td>
<td>12</td>
<td>2</td>
</tr>
<tr>
<td>RWLT</td>
<td>88.71(1)</td>
<td>2</td>
<td>91.69(3)</td>
<td>12</td>
<td>59.51(2)</td>
<td>11</td>
<td>2</td>
</tr>
</tbody>
</table>
Fig. 7. Community detection result of RWLT on dolphin network.

Fig. 7 shows the partition result obtained for the dolphin network and Fig. 6 shows the solution obtained for the US college football network. As shown in Fig. 7 and Table 6, the algorithm partitions the network into two clusters, which is consistent with the original group number. Other algorithms all partition the network into more than 2 clusters. In Fig. 6, we can see that the algorithm can effectively identify all the conferences with the exception of Sunbelt and IA Independents. The sunbelt conference breaks into two, which is regarded as reasonable in [21]. The 5 independent teams in IA Independents are distributed over three clusters. This partition is due to the fact that these teams do not belong to any conference and are hence assigned to a conference where they have played the maximum number of their games.

Referring to large-scale real world network, Fig. 8 shows the partition result of Facebook social network and an example ego-network from Facebook. Compared with other methods, the number of groups detected by RWLT is closest to the real number of ego-networks. To discover social circles, the ego-network from the user “348” is further partitioned in Fig. 8. We can see that the number of detected circles is less than 14, which is the number of ground-truth circles. This is because lots of users belong to multiple circles simultaneously in Facebook network. We evaluated the effectiveness of circle detection for each ego-network, the average value of NMI is shown in Table 6. Although RWLT performs slightly inferior to LPA, the returned cluster number of LPA is far different from the real number.
5.3 Results on Herbal Networks

5.3.1 Comparative analysis

Table 7 shows the average value of accuracy, NMI and cluster number over 10 realizations. We can see that modularity based methods, such as the GN and the CNM algorithms, have the similar results in respect of Accuracy and NMI, and these algorithms are most applicable to unweighted and undirected networks. The Cfinder has a rather poor performance on both randomly created networks and herbal networks. However, not all random walk based methods perform well. The FEC and MCL methods are outperformed by the GN and LPA. FEC performs poor because FEC considers sign attribute as the clustering attribute, while the herbal graphs only have weighted information. The LPA does not outperform as much as it does on real-world networks. The disadvantage of LPA is that it transmits the label without a step of random walk, only assigns the label that most of its neighbors currently have, without considering the nodes outside the community, which makes the results be sensitive to the graphs have no good community structure. Infomap performs well, but the running time described in Section 5.4 depicts that it is not an ideal method for herbal networks when compared with RWLT. In general, the above comparisons on several algorithms show that the RWLT would be more suita-
<table>
<thead>
<tr>
<th>Methods</th>
<th>Accuracy HN1</th>
<th>Accuracy HN2</th>
<th>NMI HN1</th>
<th>NMI HN2</th>
<th>Cluster Number HN1</th>
<th>Cluster Number HN2</th>
<th>Average-rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>GN</td>
<td>0.8764(2)</td>
<td>0.682(2)</td>
<td>0.629(4)</td>
<td>0.492(3)</td>
<td>39</td>
<td>95</td>
<td>2.75</td>
</tr>
<tr>
<td>CNM</td>
<td>0.8763(3)</td>
<td>0.679(3)</td>
<td>0.628(5)</td>
<td>0.494(2)</td>
<td>35</td>
<td>91</td>
<td>3.25</td>
</tr>
<tr>
<td>LPA</td>
<td>0.862(5)</td>
<td>0.632(5)</td>
<td>0.622(6)</td>
<td>0.429(6)</td>
<td>34.5</td>
<td>78</td>
<td>5.5</td>
</tr>
<tr>
<td>Cfinder</td>
<td>0.665(8)</td>
<td>0.421(8)</td>
<td>0.532(8)</td>
<td>0.392(8)</td>
<td>22</td>
<td>73</td>
<td>8</td>
</tr>
<tr>
<td>FEC</td>
<td>0.789(6)</td>
<td>0.618(7)</td>
<td>0.594(7)</td>
<td>0.417(7)</td>
<td>35</td>
<td>88</td>
<td>6.75</td>
</tr>
<tr>
<td>MCL</td>
<td>0.762(7)</td>
<td>0.620(6)</td>
<td>0.638(3)</td>
<td>0.468(5)</td>
<td>34</td>
<td>74</td>
<td>5.25</td>
</tr>
<tr>
<td>Infomap</td>
<td>0.882(1)</td>
<td>0.675(4)</td>
<td>0.710(1)</td>
<td>0.489(4)</td>
<td>35</td>
<td>90</td>
<td>2.5</td>
</tr>
<tr>
<td>RWLT</td>
<td>0.874(4)</td>
<td>0.718(1)</td>
<td>0.680(2)</td>
<td>0.626(1)</td>
<td>35</td>
<td>87</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 7. Experimental results of different algorithms over herbal networks.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Accuracy HN1</th>
<th>Accuracy HN2</th>
<th>NMI HN1</th>
<th>NMI HN2</th>
<th>Cluster Number HN1</th>
<th>Cluster Number HN2</th>
<th>Average-rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>GN</td>
<td>0.8764(2)</td>
<td>0.682(2)</td>
<td>0.629(4)</td>
<td>0.492(3)</td>
<td>39</td>
<td>95</td>
<td>2.75</td>
</tr>
<tr>
<td>CNM</td>
<td>0.8763(3)</td>
<td>0.679(3)</td>
<td>0.628(5)</td>
<td>0.494(2)</td>
<td>35</td>
<td>91</td>
<td>3.25</td>
</tr>
<tr>
<td>LPA</td>
<td>0.862(5)</td>
<td>0.632(5)</td>
<td>0.622(6)</td>
<td>0.429(6)</td>
<td>34.5</td>
<td>78</td>
<td>5.5</td>
</tr>
<tr>
<td>Cfinder</td>
<td>0.665(8)</td>
<td>0.421(8)</td>
<td>0.532(8)</td>
<td>0.392(8)</td>
<td>22</td>
<td>73</td>
<td>8</td>
</tr>
<tr>
<td>FEC</td>
<td>0.789(6)</td>
<td>0.618(7)</td>
<td>0.594(7)</td>
<td>0.417(7)</td>
<td>35</td>
<td>88</td>
<td>6.75</td>
</tr>
<tr>
<td>MCL</td>
<td>0.762(7)</td>
<td>0.620(6)</td>
<td>0.638(3)</td>
<td>0.468(5)</td>
<td>34</td>
<td>74</td>
<td>5.25</td>
</tr>
<tr>
<td>Infomap</td>
<td>0.882(1)</td>
<td>0.675(4)</td>
<td>0.710(1)</td>
<td>0.489(4)</td>
<td>35</td>
<td>90</td>
<td>2.5</td>
</tr>
<tr>
<td>RWLT</td>
<td>0.874(4)</td>
<td>0.718(1)</td>
<td>0.680(2)</td>
<td>0.626(1)</td>
<td>35</td>
<td>87</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 7. Experimental results of different algorithms over herbal networks.

Table 7. Experimental results of different algorithms over herbal networks.

Table 7. Experimental results of different algorithms over herbal networks.

ble for herbal group discovery, and requires neither optimization of a predefined objective function nor prior information about the communities.

### 5.3.2 Detected herbal groups

In Figs. 9 and 10, we show the detected results by RWLT on HN1 and HN2 respectively. In Fig. 9, we present in two ways to lay out the results of the HN1. To show clearly the detected groups, we lay out each group in a box and sort the boxes by group size in Fig. 9 (b). With respect to the scale of HN2, we lay out the results in typical case without labels in Fig. 10. From Fig. 9 (a) and 10 we can see that there are many edges within each group and relatively few among the groups. To further present the usefulness of our results, four groups are selected from Fig. 9 (b) to demonstrate the features, as illustrated in Fig. 11.

![Fig. 9. The result of group detection by the RWLT algorithm on HN1. Two ways are presented to lay out the results; (a) Laying out the entire graph in typical mode; (b) Laying out each graph in its own box and sort the boxes by group size.](image-url)
It is shown in Fig. 11 (a) that all the herbs in the detected group G8 have the same efficacy of “inducing diuresis to alleviate edema”, which can demonstrate the effectiveness of our method and the power of the automated method to bring together information from different sources. Furthermore, the result allows TCM practitioners to suggest latent connections between herbs of one group. For instance, although there is no direct correlation between “Poria Cocos” (茯苓) and “Rhizoma Alismatis” (泽泻), we can suggest that “Poria Cocos” and “Rhizoma Alismatis” can be used together. Because these two herbs have similar attributes, and the relationship between “Poria Cocos” and “Grifola” (猪苓), “Grifola” and “Rhizoma Alismatis” shows that these three herbs have intensive combinational rule. Actually, the latent relationship between “Poria Cocos” and
“Rhizoma Alismatis” has been demonstrated effective in the prescription “Fu Ling Ze Xie Tang” (茯苓泽泻汤). For another instance, we can find a link between “Lagenaria Siceraria” (葫芦) and “Stigmata Maydis” (玉米须) in G8, and there is no record show that these two herbs can be combined for treatment. However, according to Professor Xu, these two herbs can actually be combined to strengthen the efficacy of “inducing diuresis to alleviate edema”, which has been proved by clinical trial. This example demonstrates that clustering the herbs with similar attribute may indicate the combination of them to increase their medical effectiveness.

Similar results can also be obtained from G2 and other groups. In group G2, although there is no edge between “Notopterygium Root” (羌活) and “Stephania Tetrandra” (防己), they can be combined in “Qiang Huo Fang Ji Tang” (羌活防己汤), which has been recorded in an ancient book named 《Yi Xue Zheng Zhuan》. Other paired herbs such as “Radix Sileris” (防风) and “Radix Angelicae Pubescentis” (独活) can also be found to have combinational rule in “Fang Feng Du Huo Tang” (防风独活汤). In addition, the combination of “Chinese Ephedra” (麻黄) and “Radix Angelicae Pubescentis” (独活) has been recorded in an ancient book named 《Sheng Ji Zong Lu》 as a remedy for skin disease. In group G3, “Ginseng” (人参) can be combined with “donkey-hide gelatin” (阿胶) for the treatment of coughs, and also has intensive correlation with “Rhizoma Polygonati” (黄精), for the therapy of “nourishing Yin and moistening lung”. Such combinations have been proved by clinical trial and put into production. It is clear that the TCM knowledge from lots of ancient books may be neglected or omitted by modern researchers, and it would be time consuming for a researcher to ascertain this kind of latent connection manually from all digital books. Our algorithm in this paper can definitely help researchers find out hidden information that is extremely useful for further study in TCM.

Besides the implied connection mined from our algorithm, we find another useful result that all the herbs having mutual incompatibility are placed in different communities. In ancient Chinese literature about herbal medicine, it is recorded that some herbs are incompatible with others and may never be used in combination, otherwise toxic reactions, harmful side-effects or a diminished therapeutic effect may result. The most important are the “eighteen incompatible herbs” and the “nineteen antagonistic herbs”. For example, “Radix Glycyrrhizae” (甘草) is incompatible with “Radix Euphorbiae Pekinensis” (京大戟), “Radix Euphorbiae Kansui” (甘遂) and “Flos Daphnes Genkw” (芫花). We can see from Fig. 11 (c) that “Radix Euphorbiae Pekinensis”, “Radix Euphorbiae Kansui” and “Flos Daphnes Genkw” are clustered in the group G9 except the “Radix Glycyrrhizae”, which is shown in another group depicted in Fig. 9 (b). This result shows that our method is also able to separate the herbs with mutual incompatibility to be excluded into different communities. Table 8 shows the number of discovered compatible pairs and incompatible pairs on two herbal networks. It has to be noted that we only count the number of pairs that have been proved by clinical test.

<table>
<thead>
<tr>
<th></th>
<th>HN1</th>
<th>HN2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of compatible pairs</td>
<td>≈ 500</td>
<td>≈ 2100</td>
</tr>
<tr>
<td>Number of incompatible pairs</td>
<td>12</td>
<td>21</td>
</tr>
</tbody>
</table>
There exist some compatible examples that our method cannot discover, such as “Radix Angelicae Pubescentis” (独活) and “Futokadsura Stem” (海风藤), “Pinellia Ternata” (半夏) and “Monkshood” (附子). This is mainly due to two reasons. Firstly, some paired herbs have completely different attributes, hence they do not have direct connection in the step of correlation degree calculation. Secondly, the structural factor may lead to misclassification, like the structure of nodes 5 and 6 presented in Fig. 2. Besides, there exist some misclassified cases, such as the detected group (top left corner in Fig. 9 (b)) marked with yellow on HN1, which should be divided into three groups.

5.3.3 Comparative analysis on groups

We select these four groups (in Fig. 11) since the partitions are close to prior knowledge, and they are not the results that can be obtained by all methods. Table 9 shows the algorithms detecting the corresponding groups. As shown, G2 can only be detected by our method, and other groups can only be obtained by several methods.

<table>
<thead>
<tr>
<th>Group</th>
<th>Methods detecting corresponding group</th>
</tr>
</thead>
<tbody>
<tr>
<td>G8</td>
<td>LPA, Infomap, RWLT</td>
</tr>
<tr>
<td>G9</td>
<td>Infomap, RWLT</td>
</tr>
<tr>
<td>G2</td>
<td>RWLT</td>
</tr>
<tr>
<td>G3</td>
<td>RWLT, CNM, GN</td>
</tr>
</tbody>
</table>

To fully demonstrate the superiority of our method over two herbal networks, we provide three groups detected by Infomap, GN and CNM for comparative analysis. Fig. 12 (a) illustrates one group detected by GN and CNM. Compared with Fig. 11 (a), it is obvious that the detected group does not contain “Semen Hoveniae” (枳椇子), which should be placed in the group since it has intensive correlation with “Lagenaria Siceraria” (葫芦), “Stigmata Maydis” (玉米须) and “Grifola” (猪苓). Fig. 12 (b) shows the group detected by Infomap, which contains far less herbs compared with the group detected by RWLT (see Fig. 11 (b)). Obviously, the herb “Chinese Ephedra” (麻黄) is not detected in the group, which also occurs in the result of other methods. The result is inconsistent with the reality in TCM. As recorded in a famous Formulation “Ma Huang Du Huo Tang” (麻黄独活汤), “Chinese Ephedra” (麻黄) can always be combined with “Radix Angelicae Pubescentis” (独活) and “Radix Sileris” (防风) for relieving fever with chills. Moreover, the combination of “Chinese Ephedra” (麻黄) and “Stephania Tetrandra” (防己) has also been proved in ancient Chinese medicine. Thus, with the exclusion of “Chinese Ephedra” (麻黄) in G2, these latent combinations would not be discovered, and the direct relationship between “Chinese Ephedra” (麻黄) and “Semen lepidii” (葶苈子) is also lost. Fig. 12 (c) shows one group detected by Infomap containing the “Chinese Ephedra” (麻黄). However, the partition is not consistent with prior knowledge, and has little significance since “Chinese Ephedra” has no valuable relationship with others except “Cassia Twig” (桂枝).
A NOVEL GROUP DETECTION METHOD FOR FINDING RELATED CHINESE HERBS

Fig. 12. Samples of groups detected by other methods: (a) G1 detected by GN and CNM; (b) G2 detected by Infomap; (c) G3 detected by Infomap.

Table 10. List of paired herbs only discovered by RWLT.

<table>
<thead>
<tr>
<th>Number</th>
<th>Paired herbs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>“Chinese Ephedra” (麻黄) and “semen lepidii” (葶苈子)</td>
</tr>
<tr>
<td>2</td>
<td>“Chinese Ephedra” (麻黄) and “Radix Angelicae Pubescentis” (独活)</td>
</tr>
<tr>
<td>3</td>
<td>“Chinese Ephedra” (麻黄) and “Radix Sileris” (防风)</td>
</tr>
<tr>
<td>4</td>
<td>“Chinese Ephedra” (麻黄) and “Stephania Tetrandra” (防己)</td>
</tr>
<tr>
<td>5</td>
<td>“Ligusticum sinense” (藁本) and “Piper cubeba” (荜澄茄)</td>
</tr>
</tbody>
</table>

Based on the above instances, we can list 5 paired herbs with direct relationship or latent relationship that can only be discovered by RWLT (see Table 10). Actually, there exist more valuable pairs that only appear in the result of RWLT, but we do not list all due to the great workload.

In general, group detection based on RWLT not only clusters some herbs with intensive correlation, even other methods cannot discover, but also implies connections among herbs may be overlooked or would require much time and efforts to be found manually. Furthermore, several herbs have mutual incompatibility can also been found in different communities. Although our results are not meant to perfectly model reality of TCM, it is valuable for the researcher to conduct further research, such as new drug discovery [29], on some unconnected herbs in the same group, and to perform deep study on TCM knowledge management.

5.4 Actual-time Performance of RWLT

In addition to the computational complexity analysis provided in Section 3.2.4, we have recorded the actual needed computational time of random walk based methods to analyze the networks. We ran the experiments on a computer with a CPU of 2.26 GHz and the memory size of 4Gbytes. The operating system was Windows 7, and the simulation was implemented and tested using Matlab R2012a. We repeated all random walk based methods 10 times for each network, and the averaged actual computational time taken is shown in Table 11. In Table 11, we note that the RWLT method uses less running time when applied to real-life network with different scale. The RWLT gets a similar running time compared with the FEC and Infomap, but when the scale of network gets larger, the superiority of the RWLT becomes more obvious. As indicated by Tables 6, 7 and 11, we can conclude that the RWLT gives the best trade-off between effectiveness and efficiency.
### Table 11. Average actual computational time for different networks.

<table>
<thead>
<tr>
<th>Networks</th>
<th>Average computational time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MCL</td>
</tr>
<tr>
<td>Karate</td>
<td>0.0128</td>
</tr>
<tr>
<td>Football</td>
<td>0.1561</td>
</tr>
<tr>
<td>Dolphin</td>
<td>0.0872</td>
</tr>
<tr>
<td>HN1</td>
<td>0.8260</td>
</tr>
<tr>
<td>HN2</td>
<td>8.9803</td>
</tr>
</tbody>
</table>

Furthermore, we also applied the RWLT algorithm to LFR benchmark with different sizes to examine how the actual computational time would change with respect to the network size. Based on Fig. 13, we note that: (1) when the RWLT algorithm is applied to the network with 5000 nodes, the required computational time is quite low, as bounded by 10 seconds; (2) the actual computational time is approximately linear with respect to the network size.

![Fig. 13. Actual computational time versus the size of the network.](image)

### 6. CONCLUSIONS

We have presented a knowledge discovery technique named RWLT for TCM digital books or websites that produces detailed useful results. The method produces a list of groups of related herbs that are designed to summarize available information and to indicate herbs that are likely to be used together for special therapeutic effect. The key idea behind our algorithm rests on random walk scheme. We have tested the RWLT algorithm by using different types of networks. The experimental results show that our proposed RWLT algorithm produces good performance in both speed and clustering capability.

It is important to note that our method is not meant to perfectly model TCM reality, but to function as a tool for TCM practitioners. In fact, because herbs within a group are linked by edges from degree of correlation, it is almost certain that they are related somehow. Thus, our method is an effective way to mine and summarize important information from various sources. Furthermore, some unrelated herbs in one group allow researchers to make an attempt to do further study, such as new drug exploitation and combinational rule analysis. We have to emphasize that although our method can suggest some paired
herbs with potential combinational rules, they have to be proved through repeated clinical test prior to using.

Additionally, with respect to the result of RWLT on the herbal networks, there may have several detected groups with large scale, which is difficult for us to analyze, especially for large-scale networks. For future work, we will focus on combining other effective group detection algorithms to further subdivide large groups. Hence an aggregate of all the different solutions by different algorithms can provide a final community structure containing the most useful information.

REFERENCES

5. J. Han, J. Pei, Y. Yin, and R. Mao, “Mining frequent patterns without candidate generation: a frequent-pattern tree approach,” Data Mining and Knowledge Discovery, Vol. 8, 2004, pp. 53-87.
13. B. Yang, W. K. Cheung, and J. Liu, “Community mining from signed social net-
**Lidong Wang** (王李冬) was born in December, 4, 1982. She received the MS degree in computer Science from Ningbo University. She received the Ph.D. degree from College of Computer Science and Technology, Zhejiang University. She is currently a teacher in Hangzhou Normal University. Her current research interests include image processing, machine learning and text mining.

**Yin Zhang** (张引) was born in Lanzhou. She received the Ph.D. degree in Computer Science from Zhejiang University. She is currently an Assistant Professor in College of Computer Science and Technology, Zhejiang University. Her current research interests include knowledge discovery, machine learning, digital library and information and knowledge management.

**Xiaodong Xu** (徐晓东) is currently a Professor in Zhejiang Chinese Medical University. He obtained his Master degree from Beijing University. He has lectured and published a lot. He has (co)authored over 40 publications on the drug exploitation in traditional Chinese medicine. His research interests include prescriptions, drug exploitation and combinational rule analysis.