D-MFCLMin: A New Algorithm for Extracting Frequent Conceptual Links from Social Networks

Hamid Tabatabaee
Young Researchers and Elite Club,
Mashhad Branch, Islamic Azad University, Mashhad, Iran

***Abstract*—Massive amounts of data in social networks have made researchers look for ways to display a summary of the information provided and extract knowledge from them. One of the new approaches to describe knowledge of the social network is through a concise structure called conceptual view. In order to build this view, it is first needed to extract conceptual links from the intended network. However, extracting these links for large scale networks is very time consuming. In this paper, a new algorithm for extracting frequent conceptual link from social networks is provided where by introducing the concept of dependency, it is tried to accelerate the process of extracting conceptual links. Although the proposed algorithm will be able to accelerate this process if there are dependencies between data, but the tests carried out on Pokec social network, which lacks dependency between its data, revealed that absence of dependency, increases execution time of extracting conceptual links only up to 15 percent.**

Keywords—Social network analysis; frequent conceptual link; data mining; graph mining

I. Introduction

Social network is a social structure that is composed of some agents (generally individuals or organizations) that are connected by one or more kind of dependencies, such as ideas and financial transactions, friends, relatives, web links, spread of diseases (epidemiology). Social networks exist in different categories some of which could be found in [1]. The results of various studies indicate that the capacity of social networks can be used in many individual and social levels in order to identify problems and determine solutions, establishing social relationships, organizational governance, policy making and advising people on track to achieve the objectives.

Social network analysis is a powerful tool for analyzing the nature and pattern of communication among members of a particular group. Social network analysis helps imagine and analyze complex set of relationships between relevant factors as the maps (graphs or photographs) of connected symbols, and patterns within these categories, and it also helps calculate and review the exact size, shape and density of the network as a whole and calculate the position of each element within it. For example, in the science of epidemiology, social network analysis is used to help understand how patterns of human contact helps or prevents the spread of diseases such as HIV in a population.

From a variety of social networks, online social network has received attention among researchers. A key aspect of many online social networks is being data-rich, and therefore providing unprecedented challenges and opportunities in terms of knowledge discovery and data mining. One of the most important fields of study of traditional data mining is exploring the frequent pattern. In the field of complex data structures such as networks, the issue of exploring frequent items is discussed in form of finding a subset of nodes (sub-graphs) that occur frequently arises in a network known as graph mining. Although primitive methods in this field have been using measures deriving from graph theory [2], new approaches known as social networks mining or simply link mining try to examine features of node in addition to the network structure to extract a new set of patterns [3]-[5].

Authors in [6] described a new approach named conceptual link to describe social networks. Conceptual link provides the knowledge about groups of nodes that densely connected to each other in a social network, and through a reduced structure, which is called as conceptual view, leads to a semantic view of social network. However, the problem of extracting conceptual link is like extracting of frequent itemsets [7] with NP-hard complexity [8]. In this paper, D-MFCLMin algorithm is presented which using the concept of dependence, and by pruning the search space, tries to reduce the time required to extract frequent conceptual links. The paper will be structured as follows. In Section 2, the concept of conceptual links is presented, and then in Section 3, proposed algorithms for the extraction of frequent conceptual links are introduced. Our proposed algorithm is presented in Section 4. Finally in Section 5, test results are presented.

II. Problem Description and Definitions

In the field of search for frequent conceptual links (FCL), a model is defined as “a set of links between the two groups of nodes, where the nodes in each group share common characteristics”. When these patterns are found on the network with enough repetition, they are seen as frequent patterns and called FCL [6]. More formally, assume that G = (V, E) is a network where V is the set of nodes and E is the set of edges with $E \subseteq V \times V$. V is defined as the relation $R(A_1, \dots, A_N)$ where each A_i is a attribute. Thus, every node $v \in V$ is defined by the tuple $(a_1, \dots, a_N)$ where $\forall k \in [1..N], v[A_k] = a_k$, is the attribute value A_k in v. An item is a logical expression as A = x where A is an attribute and x is a value. Empty items are shown as $\emptyset$. An itemset is a combination of items for example A1 = x and A2 = y and A3 = z. An itemset, m, which is a combination of k non-empty item is called a k-itemset and noted m^k ($|m^k| = k$).

Suppose that *m* and *sm* are two itemset. If sm ⊆ m, we say that *sm* is a *sub-itemset* and *m* is a *super-itemset* of *sm*. For example, *sm = xy* is a sub-itemset from *m = xyz* [6]. Set of all *t-itemset* made of V are shown with I^{t}. Moreover, UI^{t} is defined as follows (set of all itemset of maximum size t):

$$UI^{\mathrm{t}} = \cup_{k=1}^{t} I^{\mathrm{k}} \tag{1}$$

Suppose that *G* is a directed graph. Thus, for any itemset *m* on UI^{N}, $\mathrm{V_m}$ is shown as a series of nodes in *V* that is match the pattern *m* and defined as follows [6]:

- Set of links on the left *m* (LE_m): the set of links from *E* that starts from the nodes that satisfy *m*.

$$\mathrm{LE_m} = \{e\ \in E; e = (a, b), a\ \ V_m\}$$

- The set of links on the right (RE_{m}) *m*: the set of links from E that enter the nodes that satisfy *m*.

$$\mathrm{RE_m} = \{e\ \in E; e = (a, b), b \in\ V_m\}$$

Definition 1. Conceptual links [6]: Suppose that *m1* and *m2* are two itemset and V_{m1} and V_{m2} are the set of nodes in *V* that satisfy *m1* and *m2* respectively. $\mathrm{E_{(m_1,m_2)}}$ is the set of links connecting the nodes in V_{m1} to the nodes in V_{m2},

$$E_{(m_1,m_2)} = LE_{m1} \cap \mathrm{RE_{m2}} = \{\mathrm{e\ \in\ E\ ;\ e\ =\ (a,b), a\ \in\ V_{m1}\ and\ b\ \in\ V_{m2}}\} \tag{2}$$

Definition 2. [6]: We call support $\mathrm{E_{(m_1,m_2)}}$ as ratio of links in *E* that belongs to $\mathrm{E_{(m_1,m_2)}}$.

$$\mathrm{supp}\left(\mathrm{E_{(m_1,m_2)}}\right) = \frac{\left|\mathrm{E_{(m_1,m_2)}}\right|}{|\mathrm{E}|} \tag{3}$$

Definition 3. [6]: It is said that there is FCL and we write *(m1, m2)* if support $\mathrm{E_{(m_1,m_2)}}$ is greater than a minimum support threshold β, i.e. $\mathrm{supp(E_{(m_1,m_2)})}\ >\ \beta$.

Definition 4. [6]: Suppose UI^{t} is the set of all itemset of maximum size *t* in *V*. The FL^{t} is defined as FCL extracted from these itemsets.

$$\mathrm{FL^t}\ = \cup_{\mathrm{m_1} \in UI^{\mathrm{t}}, \mathrm{m_2} \in UI^{\mathrm{t}}} \{\mathrm{E_{(m_1,m_2)}}; \frac{\left|\mathrm{E_{(m_1,m_2)}}\right|}{|\mathrm{E}|} > \beta\} \tag{4}$$

Property 1. [6]: According to definition 3, if link *(m₁, m₂)* is frequent, the set of $\mathrm{LE_{m1}}$ and $\mathrm{RE_{m2}}$ meet the following condition:

$$|\boldsymbol{LE_{m1}}| > \beta \times |\boldsymbol{E}| \textbf{ and } |\boldsymbol{RE_{m2}}| > \beta \times |\boldsymbol{E}| \tag{5}$$

Definition 5. The conceptual sub link [6]: suppose that two itemset sm_1 and sm_2 are sub-itemsets of m_1 and m_2 in UI respectively. Conceptual link (sm_1, sm_2) is called the sub-link of (m_1, m_2) , similarly (m_1, m_2) is called super-link (sm_1, sm_2) and written as $(sm_1, sm_2)\ \subseteq\ (m_1, m_2)$.

Property 2. Downward-closure [6]: If a conceptual link *l* is frequent all its sub-links are frequent too. Thus, if a link is not frequent, none of its super-links is frequent.

Definition 6. Maximum FCL [6]: Assume that β has a given support threshold value, we say that the maximum frequent conceptual link (MFCL), any FCL is so that no super-link of $\acute{\mathrm{l}}$ from l that is frequent exists. More formally:

$$\nexists\, \acute{\mathrm{l}} \in \mathrm{FL^N}\ \text{ so that }\ \mathrm{l}\ \subset\ \acute{\mathrm{l}} \tag{6}$$

III. Related Work

Popular approaches of mining social networks have been proposed to extract different forms of knowledge from these networks. Similar to the traditional field of data mining, social network mining addresses wide range of tasks such as classification, clustering, search for frequent patterns or link prediction. These methods can be divided into two groups [8]:

- Approaches based on predictive modeling that includes techniques that analyze current and past facts to make predictive assumptions about future or unknown events.
- Approaches based on descriptive modeling that cover a set of techniques whose aim is to summarize data by identifying some related features to describe how things organize and actually work [8].

In this study, the focus is on descriptive approach of the social network. These approaches can be divided into following four categories [5].

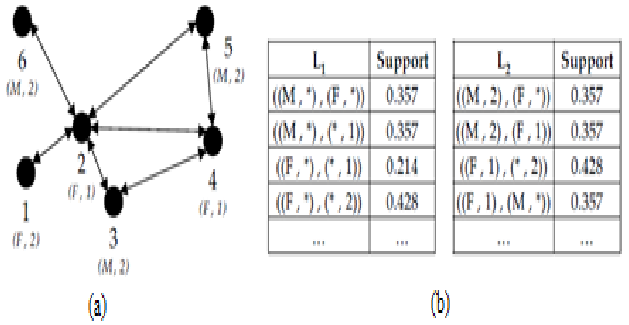
1) Link Based Clustering (also known as Community Detection) that searches a dense groups of nodes and its aim is to analyze network to several linked components (communities) in such a way that nodes in each component have high-density connections, while nodes in different components have the lowest density. Of the proposed methods in this category algorithm SLPA [9], TopGC [10], SVINET [11], MCD [12], CGGC [13], CONCLUDE [14], DSE [15] and SPICi [16] can be cited.

2) Hybrid clustering that simultaneously considers attributes and the structure of the nodes to identify clusters. The aim of this new type of approaches is partitioning of the network by balancing structural similarities and attributes so that nodes with common attributes are grouped in one partition and the nodes inside partition are densely linked. These approaches provide a more conceptual partition of the network that is not necessarily proportional to context. Of clustering methods SA-Cluster [4] and CESNA [17] can be cited.

3) Frequent Sub-graph mining. The most widely used definition of a pattern is as a connected sub graph [18]. Therefore, techniques that focus on the search for frequent patterns in social networks aim to identify sub graphs that occur frequently in a database or a very large network of networks, based on a minimum threshold value. Among the prominent methods in this category, Apriori-based algorithms [19] and pattern growth [20] can be cited.

4) FCL combines network structure information and node attributes for providing knowledge about groups of nodes, which have more connections in a social network. Extracting MFCL creates a complexity similar to frequent item set, since it is proven that this complexity is NP-hard. Extracting all MFCLs from a social network may be a challenging problem and computationally intensive. According to the definitions of the concept of conceptual links, we deal with the methods provided for extracting these links.

If search space is very large, discovering all the frequent links in a network is very costly. In a simple approach, it is necessary to produce all set of possible items and then examine the frequency of each pair of them. To reduce this time, at the beginning, FLMIN algorithm [21] was proposed. This algorithm used a bottom-up approach by applying property 2 to gradually reduce the search space to include a superset of items that will potentially exist in FCLs. In Fig. 1, a sample of conceptual links extracted by FLMIN algorithm is shown.



L_1	Support
((M , *) , (F , *))	0.357
((M , *) , (* , 1))	0.357
((F , *) , (* , 1))	0.214
((F , *) , (* , 2))	0.428
...	...

L_2	Support
((M , 2) , (F , *))	0.357
((M , 2) , (F , 1))	0.357
((F , 1) , (* , 2))	0.428
((F , 1) , (M , *))	0.357
...	...

Fig. 1. A sample of conceptual links extracted by FLMIN algorithm [21]

In [22] MAX-FLMin algorithm was presented. In this algorithm, the aim is finding MFCLs. Compared with previous algorithm, this algorithm only uses itemsets that satisfy property 1 to create links, and then they are checked for being frequent. In addition, in the process of examining the created link in order to add it to frequent links, this algorithm checks lack of existence of maximum frequent link compared to the current link.

In H-MFCLMin algorithm [5] in order to accelerate the extraction of MFCLs, some of the itemsets are filtered. The filtered itemsets includes itemsets that the number of their matched nodes in the network are less than the threshold α. α is an input parameter for the algorithm. This filtering is done with the argument that there is little likelihood that an itemset with low frequency can attract a high proportion of links in the network and therefore by filtering these kinds of itemsets, despite the reduction in the search space, certain information will not be lost from the final conceptual network.

IV. The Proposed Algorithm

In this paper, D-MFCLMin algorithm is proposed to extract conceptual links. By pruning the search space by applying the concept of dependency, this algorithm accelerates the extraction of conceptual links. In the following, first we introduce the concept of dependency, and then we will go on to present pseudo code of the proposed algorithm.

A. Definitions

Definition 7. Dependency: Suppose m^t and n^t are two itemset. We say m^t is dependent on n^t and show it as $n^t \rightarrow m^t$, if $\forall v \in V_{m^t}$, we have $v \in V_{n^t}$. We show all dependencies of an itemset such as m^t in the form of $D(m^t)$.

$$D(m^t) = \{n^t | \, n^t \rightarrow m^t\} \tag{7}$$

Definition 8. Set of selected itemset: Assume that FL^t is the set of extracted FCL from itemsets with maximum *t* items. LI_{sel}^t (RI_{sel}^t) is a set of itemsets used to create these links.

$$LI_{sel}^t = \{m;\ E_{(m,n)} \in\ FL^t\}$$

$$RI_{sel}^t = \{m;\ E_{(n,m)} \in\ FL^t\} \tag{8}$$

Property 3: If itemset n^t is not in any of the extracted FCLs in FL^t $\left(n^t\ \notin LI_{sel}^t(RI_{sel}^t)\right)$, then none of the itemsets that depend on it ($\{m^t | n^t \in D(m^t)\}$) will be at FL^t.

Proof: Assume that m^t is the itemset that depend on the itemset n^t, and suppose that n^t is not involved in any FCL ($n^t \notin\ LI_{sel}^t(RI_{sel}^t)$), so according to definition of FCL, for all itemset such as n_j:

$$|LE_{n^t} \cap RE_{n_j}| < \beta \times |E|$$

$$(\text{or } |RE_{n^t} \cap LE_{m_j}| < \beta \times |E)$$

Moreover, according to the definition 6, we know that

$$V_{m^t} \subseteq V_{n^t}$$

So we have:

$$|LE_{m^t}| \leq \left|LE_{n^t}\right| \text{ and } |RE_{m^t}| \leq \left|RE_{n^t}\right|,$$

As a result:

$$|LE_{m^t} \cap RE_{n_j}| < \beta \times |E|$$

$$(\text{or } |RE_{m^t} \cap LE_{n_j}| < \beta \times |E|)$$

And therefore the property is proven.

Definition 9. Parents of an itemset: For each itemset m^t, $(t > 1)$، two parents are shown as parent1 (m^t) and parent2 (m^t), (parent1 (m^t), parent2 $(m^t) \in I^{t-1}$) so that:

$$m^t = \text{parent1}\ (m^t)\ .\ \text{parent2}\ (m^t)$$

Definition 10. Dependency Level: For each itemset *m*, the dependence level is shown with $DL(m)$ and defined as follows:

$$DL(m) = \begin{cases} 0 & \text{if } D(m) = \emptyset \\ \max_{n \in D(m)} DL(n) + 1 & \text{else} \end{cases} \tag{9}$$

B. Pseudo Code of D-MFCLMin Algorithm

The pseudo code for proposed algorithm is given below. Similar to H-MFCLMin, input parameters are α and β that are threshold value related to itemset and link support respectively.

Similar to H-MFCLMin [5], in the first iteration (*t = 1*), *1-itemset* LI_{cand}^1 (RI_{cand}^1) are created according to the properties 1, 2 (lines 6 and 7). After creating these lists, the set of their itemsets are ordered in terms of the amount of their support in ascending order. Unlike H-MFCLMin, before the search for FCLs, in iteration *t*, the dependencies between itemsets in $LI_{cand}^t(RI_{cand}^t)$ are obtained.

For this purpose, set of *t-itemsets* of $LI_{cand}^t(RI_{cand}^t)$ are mutually joined and then, based on the amount of support of resulted itemsets, the existence of dependency between two joined itemset is checked. In the absence of dependency, resulted itemset is inserted to the list for the next iteration $\mathrm{LI}_{\mathrm{cand}}^{\mathrm{t+1}}(\mathrm{RI}_{\mathrm{cand}}^{\mathrm{t+1}})$ as one of the candidate itemsets (lines 25-11). This insertion is done in a way that the order of the list of items remains in ascending order in terms of the amount of support.

After determining the dependencies among the itemsets of iteration *t*, their dependence level is calculated and then $\mathrm{LI}_{\mathrm{cand}}^{\mathrm{t}}(\mathrm{RI}_{\mathrm{cand}}^{\mathrm{t}})$ is sorted by increasing order of the level of dependence (line 26). After sorting, the search for FCLs is done. Founded FCLs are added to FL^t list and then by removing sub FCLs links located in FLV_{max}, are added to FLV_{max} as MFCL (lines 44-27).

More exactly, this search is done so that for every itemset $\mathrm{m_i} \in \mathrm{LI}_{\mathrm{cand}}$ and $\mathrm{m_j} \in \mathrm{RI}_{\mathrm{cand}}$, with the condition that $|\mathrm{m_i}| = \mathrm{t}$ or $|\mathrm{m_j}| = \mathrm{t}$ is checked whether the link $(\mathrm{m_i}, \mathrm{m_j})$ is frequent or not. Before this check, set of the dependent itemset $\mathrm{m_i}$ and $\mathrm{m_j}$ are checked. If none of the sets of dependent itemsets are added in FL^t, checking the frequency of this pair is ignored (line 33). Recall that the itemsets in $\mathrm{LI}_{\mathrm{cand}}^{\mathrm{t}}$ and $\mathrm{RI}_{\mathrm{cand}}^{\mathrm{t}}$ are arranged in ascending order of dependency, so when check an itemset, all of its dependant itemsets, has already been investigated at this iteration. After this step, similar to H-MFCLMin algorithm, checking the frequency of the link is done (line 34). If the link is frequent, $(\mathrm{m_i}, \mathrm{m_j})$, $\mathrm{m_i}$ are added to $\mathrm{LI}_{\mathrm{sel}}^{\mathrm{t}}$ and $\mathrm{m_j}$ is added to $\mathrm{RI}_{\mathrm{sel}}^{\mathrm{t}}$.

After the review of itemsets in $\mathrm{LI}_{\mathrm{cand}}^{\mathrm{t}}$ and $\mathrm{RI}_{\mathrm{cand}}^{\mathrm{t}}$, itemsets of $\mathrm{LI}_{\mathrm{cand}}^{\mathrm{t+1}}$ and $\mathrm{RI}_{\mathrm{cand}}^{\mathrm{t+1}}$ are modified to extract FCLs at iteration *t*. At this point, any itemset $m^{\mathrm{t+1}}$ ($m^{\mathrm{t+1}} \in \mathrm{LI}_{\mathrm{cand}}^{\mathrm{t+1}}(\mathrm{RI}_{\mathrm{cand}}^{\mathrm{t+1}})$) whose both parent itemsets (Definition 8) are not in $\mathrm{LI}_{\mathrm{sel}}^{\mathrm{t}}$ ($\mathrm{RI}_{\mathrm{sel}}^{\mathrm{t}}$) are removed from the list (49-45).

Algorithm 1: D-MFCLMin Algorithm

```
Require: G = (V;E): Network, $\beta \in$[0..1]: Link support threshold and $\alpha \in$ [0..1]: Itemset filtering threshold
1. $\mathrm{FL_{Vmax}}$: Set of MFCLs $\leftarrow \emptyset$
2. LIcand: Stack of left-hand itemset candidates $\leftarrow \emptyset$
3. RIcand: Stack of right-hand item set candidates $\leftarrow \emptyset$
4. $\mathrm{FL^t}$: List of frequent conceptual links $\leftarrow \emptyset$
5. t: Iteration $\leftarrow$ 1
{Generation of the 1-itemsets}
6. $\mathrm{LI}_{\mathrm{cand}}^{1} \leftarrow$ Generate 1-itemsets m from V such as $|\mathrm{V_m}| > \alpha$ and $|\mathrm{LE_m}| > \beta \times |E|$
7. $\mathrm{RI}_{\mathrm{cand}}^{1} \leftarrow$ Generate 1-itemsets m from V such as $|\mathrm{V_m}| > \alpha$ and $|\mathrm{RE_m}| > \beta \times |E|$
8. Sort $\mathrm{LI}_{\mathrm{cand}}^{1}$ , $\mathrm{RI}_{\mathrm{cand}}^{1}$ itemsets by their Supports
9.   t $\leftarrow$ 1
10.  do
       {Determining Dependencies between $\mathrm{LI}_{\mathrm{cand}}^{\mathrm{t}}(\mathrm{RI}_{\mathrm{cand}}^{\mathrm{t}})$ itemsets}
11.    for all item set $\mathrm{m_i^t} \in \mathrm{LI}_{\mathrm{cand}}^{\mathrm{t}}(\mathrm{RI}_{\mathrm{cand}}^{\mathrm{t}})$ do
12.      for all item set $\mathrm{m_j^t} \in \mathrm{LI}_{\mathrm{cand}}^{\mathrm{t}}(\mathrm{RI}_{\mathrm{cand}}^{\mathrm{t}})$ do
13.        if ($\mathrm{m_i^t}$ and $\mathrm{m_j^t}$ share $\mathrm{t-1}$ item)
14.          $\mathrm{m_k^{t+1}} \leftarrow$ join $\mathrm{m_i^t}$ and $\mathrm{m_j^t}$
15.          if  (sup($\mathrm{m_k^{t+1}}$) = sup($\mathrm{m_i^t}$))
16.            add  $\mathrm{m_j^t}$ to D($\mathrm{m_i^t}$)
17.          else
18.            if ($|\mathrm{V}_{\mathrm{m_k^{t+1}}}| > \alpha$ and $\left|\mathrm{LE}_{\mathrm{m_k^{t+1}}}\right| > \beta \times |E|$ ($\left|\mathrm{RE}_{\mathrm{m_k^{t+1}}}\right| > \beta \times |E|$) )
19.              add $\mathrm{m_k^{t+1}}$ to $\mathrm{LI}_{\mathrm{cand}}^{\mathrm{t+1}}(\mathrm{RI}_{\mathrm{cand}}^{\mathrm{t+1}})$
20.              parent1 ($\mathrm{m_k^{t+1}}$ ) $\leftarrow \mathrm{m_i^t}$
21.              parent2 ($\mathrm{m_k^{t+1}}$ ) $\leftarrow \mathrm{m_j^t}$
22.            end if
23.          end if
24.      end for
25.    end for
26.    Sort $\mathrm{LI}_{\mathrm{cand}}^{\mathrm{t}}(\mathrm{RI}_{\mathrm{cand}}^{\mathrm{t}})$ itemsets by their calculated dependency level
       {Generation of frequent conceptual  links}
27.    $\mathrm{FL^t} \leftarrow \emptyset$
28.    $\mathrm{L_{sel}^t} \leftarrow \emptyset$
29.    $\mathrm{R_{sel}^t} \leftarrow \emptyset$
30.    for all item set $\mathrm{m_i} \in \mathrm{LI_{cand}}$ do
31.      for all item set $\mathrm{m_j} \in \mathrm{RI_{cand}}$ do
32.        if ($|\mathrm{m_i}| = \mathrm{t}$ or $|\mathrm{m_j}| = \mathrm{t}$)
33.          if ($\exists(\mathrm{m_k},\ \mathrm{m_j}) \in \mathrm{FL^t},\ \forall\, \mathrm{m_k} \in \mathrm{D(m_i)}$ and $\exists(\mathrm{m_i},\ \mathrm{m_k}) \in \mathrm{FL^t},\ \forall\, \mathrm{m_k} \in \mathrm{D(m_j)}$)
34.            if ($\exists$ l $\in \mathrm{FL^t}$ such as $(\mathrm{m_i},\ \mathrm{m_j}) \subset$ l and $|(\mathrm{m_i},\ \mathrm{m_j})| > \beta \times |\mathrm{E}|$)
35.              add $(\mathrm{m_i},\ \mathrm{m_j})$ to $\mathrm{FL^t}$
36.              remove all q $\in \mathrm{FL_{Vmax}}$ such as q $\subset (\mathrm{m_i},\ \mathrm{m_j})$
37.              add $(\mathrm{m_i},\ \mathrm{m_j})$ to $\mathrm{FL_{Vmax}}$
```

```
                add $m_i$ to $L^t_{sel}$
                add $m_j$ to $R^t_{sel}$
            end if
        end if
    end if
  end for
end for
for all item set $m_i \in LI^{t+1}_{cand}(RI^{t+1}_{cand})$ do
    if $(parent1(m_i) \notin L^t_{sel}(R^t_{sel})$ and $parent2(m_i) \notin L^t_{sel}(R^t_{sel}))$
        remove $m_i$ from $LI^{t+1}_{cand}(RI^{t+1}_{cand})$
    end if
end for
$t \leftarrow t + 1$
while $FL^t \neq \emptyset$ and allCombinations() = false
return FLVmax
```

C. Analysis of the Proposed Algorithm

First, the cost of H-MFCLMin algorithm is discussed. Suppose that we want check the existence of conceptual link between the two itemset m^i_1 and m^j_2 ($i = t$ *or* $j = t$) at iteration $t(m^i_1 \in LI^t_{cand}, m^j_2 \in RI^t_{cand})$. To this end, the edges of the network whose source node belong to m^i_1 and their destination node belongs to m^j_2 will be counted, the cost of this study can be obtained as follows:

$$C(m^i_1, m^j_2) = 2.N.|E| \tag{10}$$

In the above equation, N is the number of features of each itemset. To search for a node belonging to an itemset, it is enough to compare attribute values of nodes with the itemset, which will have cost of N, and because this action should be done for source and destination of each of the edges, double of these costs will be imposed.

In D-MFCLMin algorithm, by taking into account the dependencies, the above costs will change as follows:

$$C(m^i_1, m^j_2) = C_d + (1 - p)(2.N.|E|) \tag{11}$$

In the above relation, C_d is the cost of calculating the dependencies of two itemset m^i_1 and m^j_2, and p is the probability that dependencies on these two itemsets would stop counting the edges of social network to check for conceptual link between them. Value of C_d depends on the number of dependencies of the itemsets being checked and the number of conceptual links found in the intended iteration. In the algorithm D-MFCLMin, for every pair of items being checked, their dependency of participation in the conceptual links that have been found so far in the current phase is evaluated, so this cost is as follows:

$$C_d = (|D(m^i_1)| + |D(m^j_2)|)|FL^t| \tag{12}$$

Therefore, in the following the number of two factors of the itemset dependencies and conceptual links are examined.

1) The number of dependencies of an itemset: There is no possibility to determine the exact number of dependencies of an itemset, so we will consider their maximum number. For simplicity, we assume that the number of itemsets in iteration *t*, in LI^t_{cand} are RI^t_{cand} equal. According to this assumption, in rest of paper we assume no difference between the two sets and therefore to be concise we will use the abbreviation I^t. As already mentioned, the set of itemsets in each iteration are ordered based on the support arranged in ascending. Based on the assumption of the existence of maximum possible dependencies in the set I^t, the first itemset will not be dependent on any itemset, the second itemset only may be dependent on the first itemset, the third itemset at most will be dependent on two previous itemset, and so on, so the maximum number of dependencies between all itemsets in the set I^t is equal to:

$$\frac{|I^t|(|I^t|- 1)}{2} \tag{13}$$

By considering the uniform distribution of this dependency between itemsets of this set, the maximum number of dependencies for each itemset is obtained as:

$$|D(m^t)| = \frac{|I^t|- 1}{2} \tag{14}$$

It should be noted that the maximum number of itemset in iteration can be obtained from the following recursive relation:

$$|I^M| = T(N, M) = \begin{cases} \sum_{i=1}^{N} K_i & M = 1 \\ \prod_{i=1}^{N} K_i & N = M \\ \sum_{i=M}^{N} K_i.T(i-1, M-1) & \text{else} \end{cases} \tag{15}$$

In the above relation K_i shows the number of possible values for i-th feature. For example, about the characteristics of gender, the number of possible values is equal to 2.

2) The number of conceptual links: The second factor affecting the cost of checking dependencies is the number of conceptual links found in a step ($|FL^t|$). Given the steady growth of the number of conceptual link, the maximum number of conceptual links assessed per pair itemset is equal to:

$$\frac{(2|UI^t||I^t|- |I^t|^2)^2}{2} \tag{16}$$

According to the above values, the number of conceptual links that are checked for every pair itemset on average is equal to:

$$\frac{2|UI^t||I^t|- |I^t|^2}{2} \tag{17}$$

According to relations (14) and (17), the overall amount of C_d is obtained as follows:

$$\mathrm{C_d} = \frac{2|\mathrm{UI^t}||\mathrm{I^t}|^2 - |\mathrm{I^t}|^3}{2} \tag{18}$$

Now, with regard to determining the amount of the dependencies cost, we will analyze the behavior of the proposed algorithm.

The worst situation in the proposed algorithm occurs when despite the large amount of dependencies, there is no pruning. The amount of pruning depends on the number of conceptual links found, as the number of conceptual link found is low, an increase in dependency, will be more likely in pruning the itemsets. On the other hand, the number of FCLs depends on the amount of β, as the value of this parameter is less, more FCLs will be found. Therefore, we expect that the proposed algorithm shows a weaker performance when β is a small amount.

V. Experiments and Results

In this section, the results of the assessment of the proposed method (D-MFCLMin) are provided. H-MFCLMin method is considered as the method used for comparison. First, in the next section, data set used is introduced, and then we will examine the results.

A. *Dataset*

In this study, dataset of a social network called Pokec was used [23]. *Pokec* is the most popular online social network in Slovakia. This dataset includes altered profiles of the users of this social network with links of friendship between them. It should be noted that in Pokec friendship relationship are directed. User's profile includes 59 fields that only eight fields are mandatory. In Table I, the features of these eight fields are shown.

B. *Results*

As mentioned earlier, in order to evaluate the performance of the proposed algorithm, its results were compared with the results of H-MFCLMin algorithm. It should be noted that the output of both methods is similar in the sense that, there are no differences in the extracted FCL in the two algorithms. In Fig. 2, the conceptual view derived from *Pokec* is shown by taking value 0.3 for β. An interesting feature shown in this figure is the two-way communications between itemsets. In fact, if there are conceptual links between the itemset A to B, there is a conceptual link between B to A itemset too. As already mentioned, the mentioned social network is directional, which means that friendship is one-sided. However, with the resulting outputs, it is revealed that the users of this social network have bilateral friendship relations.

TABLE I. Mandatory Fields Features in Social Network

Field title	**Type of field**	**Domain**	**Description**
User _id	Integer	The number of users-1	An integer that maps the user name of choice
Public	Boolean	True. False	Profile's Being Public
Completion percentage	Integer	[1-100]	The completion percent of user profile attributes
Gender	Boolean	True. False	
Region	Textual	[1-183]	User living area *
last_ login	date time	1999 to 2012	Last logon of the user
Registration	date time	1999 to 2012	registration time of the user in the system
Age	Integer	[1-100]	User age

* Frequently areas in Slovakia but some areas included in the Czech and Germany as well

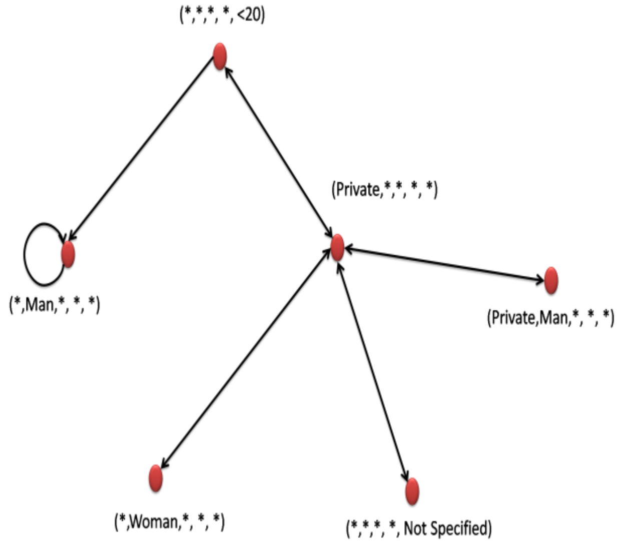


Fig. 2. Conceptual view derived from Pokec (β = 0.3).

Although the proposed algorithm (D-MFCLMin) and H-MFCLMin algorithm extract similar conceptual views from the social network, the time taken to do this, is slightly different in two algorithms. In Fig. 3, the run time of each of these two algorithms to extract MFCL from Pokec social network is shown at different values of parameter β. It should be noted, parameter α value is considered as equal to zero. Both algorithms have been run 10 times and the achieved average execution time is considered as their run time.

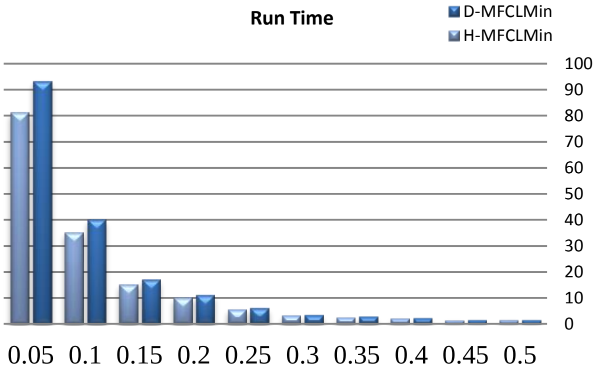


Fig. 3. The run time of the two algorithms, D-MFCLMin and H-MFCLMin in different values of β.

As can be seen, at high levels of β, both algorithms have almost the same performance but with a lower value of this parameter, the difference in the time of two algorithms becomes greater. This difference is the time that takes to proposed algorithm to determine the dependencies between itemsets. It is noteworthy that, unfortunately, the dependency between itemset of used dataset is zero, so in fact no pruning is done due to the dependency in this experiment. However, as in the figure above is shown, despite the lack of existence of dependency in the *Pokec*, in the worst case (small values of β) run time of the proposed algorithm is ultimately up to 15 percent more than H-MFCLMin. However, if there is dependency between itemsets, the possibility of pruning the search space and thus accelerating the extraction of FCLs will be possible, and thus the difference in performance of the two algorithms will be a greater increase.

VI. Conclusion

Widespread use of social networks has caused very high volume of information so knowledge extraction has become one of the areas of interest for researchers. FCLs are one of the approaches to extract knowledge from these networks that in addition to the data related to communications emphasizes the data related to the existence of these networks. In this paper, by introducing and using the concept of dependency, a new algorithm is presented to accelerate the extraction of FCLs. The existence of dependencies between data causes a pruning of portion of the search space and thus accelerates the process of extracting conceptual links. Due to the lack of dependency in the used dataset, this acceleration was not observed, but the test results showed that despite the lack of dependencies, the proposed algorithm compared with H-MFCLMin algorithm has almost the same performance.

References

[1] Aggarwal, C. C. (2011). An introduction to social network data analytics. In Social Network Data Analytics. Edited by C. Aggarwal. Springer, 1–15.

[2] West, D.B. (2000). Introduction to graph theory. 2nd end. Prentice Hall, Englewood Cliffs.

[3] Tian, Y., Hankins, R.A., & Patel, J.M. (2008). Efficient aggregation for graph summarization. In: Proceedings of the 2008 ACM SIGMOD international conference on management of data, ACM, 567–580.

[4] Zhou, Y., & Cheng, H., & Yu, J.X. (2009). Graph clustering based on structural/attribute similarities. VLDB Endow 2(1):718–729.

[5] Stattner, E., & Collard, M. (2013). Towards a hybrid algorithm for extracting maximal frequent conceptual links in social networks. In: IEEE international conference on research challenges in information science, 1–8.

[6] Stattner, E. & Collard, M. (2012). Social-based conceptual links: Conceptual analysis applied to social networks. International Conference on Advances in Social Networks Analysis and Mining.

[7] Yang, G. (2004). The complexity of mining maximal frequent item sets and maximal frequent patterns. In KDD 04: Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data mining, ACM Press, 344–353.

[8] Stattner, E. & Collard, M. (2015). Descriptive Modeling of Social Networks, Procedia Computer Science, Volume 52, 226-233.

[9] Xie, J., & Szymanski, B. K. (2012). Towards linear time overlapping community detection in social networks. In PAKDD (2), 25–36.

[10] Macropol, K., & Singh, A. K. (2010). Scalable discovery of best clusters on large graphs. PVLDB, 3(1):693–702.

[11] Gopalan, P. K. & Blei, D. M. (2013). Efficient discovery of overlapping communities in massive networks. Proceedings of the National Academy of Sciences, 110(36):14534–14539.

[12] Riedy, J., Bader, D. A., & Meyerhenke, H. (2012). Scalable multithreaded community detection in social networks. In Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2012 IEEE 26th

[13] Ovelgonne, M., & Geyer-Schulz, A. (2012). An ensemble learning strategy for graph clustering. In Graph Partitioning and Graph Clustering, 187–206. International, 1619–1628.

[14] De Meo, P., Ferrara, E., Fiumara, G., & Provetti, A. (2014). Mixing local and global information for community detection in large networks. J. Comput. Syst. Sci., 80(1):72–87.

[15] Chen, J., & Saad, Y. (2012). Dense subgraph extraction with application to community detection. Knowledge and Data Engineering, IEEE Transactions on, 24(7):1216–1230.

[16] Jiang, P., & Singh, M. (2010). Spici: a fast clustering algorithm for large biological networks. Bioinformatics, 26(8):1105–1111.

[17] Yang, J., McAuley, J., Leskovec, J. (2013). Community Detection in Networks with Node Attributes. in Data Mining (ICDM), 2013 IEEE 13th International Conference on, 1151-1156.

[18] Getoor, L., & Diehl, C.P. (2005). Link mining: a survey. SIGKDD Explore News l 7:3–12.

[19] Agrawal, R., & Srikant, R. (1994). Fast Algorithms for Mining Association Rules in Large Databases. VLDB Conference, 487–499.

[20] Han, J., Pei, J., Yin, Y., & Mao, R. (2003), Mining frequent patterns without candidate generation: A frequent-pattern tree approach. Data Mining and Knowledge Discovery.

[21] Stattner, E. & Collard, M. (2012). FLMin: An Approach for Mining Frequent Links in Social Networks. 4th International Conference, NDT 2012, Dubai, UAE, April 24-26, 2012, Proceedings, Part II, 449-463.

[22] Stattner, E. & Collard, M. (2012). MAX-FLMin: An Approach for Mining Maximal Frequent Links and Generating Semantical Structures from Social Networks. 23rd International Conference, DEXA 2012, Vienna, Austria, Proceedings, Part I, 468-483.

[23] Takac, L., Zabovsky, M. (2012). Data Analysis in Public Social Networks. International Scientific Conference & International Workshop Present Day Trends of Innovations, Poland.