

Learning Representative Nodes in Social Networks

Ke Sun¹, Donn Morrison², Eric Bruno³, and Stéphane Marchand-Maillet¹

¹ Viper Group, Computer Vision & Multimedia Laboratory, University of Geneva,
Switzerland

² Unit for Information Mining & Retrieval, Digital Enterprise Research Institute,
NUIG, Galway, Ireland

³ Knowledge Discovery & Data Mining, Firmenich S.A., Switzerland

Abstract. We study the problem of identifying representative users in social networks from an information spreading perspective. While traditional network measures such as node degree and PageRank have been shown to work well for selecting seed users, the resulting nodes often have high neighbour overlap and thus are not optimal in terms of maximising spreading coverage. In this paper we extend a recently proposed statistical learning approach called *skeleton learning* (SKE) to graph datasets. The idea is to associate each node with a random *representative* node through Bayesian inference. By doing so, a prior distribution defined over the graph nodes emerges where representatives with high probabilities lie in key positions and are mutually exclusive, reducing neighbour overlap. Evaluation with information diffusion experiments on real scientific collaboration networks shows that seeds selected using SKE are more effective spreaders compared with those selected with traditional ranking algorithms and a state-of-the-art degree discount heuristic.

1 Introduction

In a social network, a small subset of *representative* nodes can help establish a hierarchical messaging scheme: the correspondence with each individual node is through a nearby *representative*. Despite that the word “representative” can be interpreted in different ways in social analysis, here, the purpose of such a hierarchy is to broadcast information efficiently with constrained resources. Locally, these representatives should lie in hub positions so as to minimize the routing cost to their nearby nodes. Globally, there should be as few representatives governing different regions so as to save resources.

From a machine learning perspective, a closely related problem is *spectral clustering* [1–3], where the network is partitioned into a fixed number of densely-connected sub-networks with sparser connections between them. This technique has been applied to social networks, e.g., for community detection [4, 5] and spam nodes identification [6]. It is powerful in depicting complex clusters with simple implementations. It is computationally expensive for large datasets, especially when a proper number of clusters has to be searched over [4].

In the data mining community, the graph-based ranking algorithms [7–9] have a profound impact on the present World Wide Web and citation analysis systems. They rank graph nodes based on the general idea that the value of one node is positively related to the value of its neighbours. These approaches are further investigated by machine learning researchers using spectral graph theory [10, 11] and random walks [12]. In our task of selecting representatives, the highly ranked nodes by these algorithms usually have high neighbour overlap because of the mutual reinforcement between connected high degree nodes.

Motivated by seeking effective marketing strategies, efforts have been made to select a set of influential individuals [13–15] and to maximize their influence through information diffusion [16, 17]. Although the optimization problem is generally NP hard [13], reasonable assumptions lead to polynomial-time solvable models [14] and efficient implementations with approximation bounds [15]. Targeting at similar objectives, methods from different perspectives are developed with improved speed and performance [18, 19].

This work provides a novel approach to measure the representativeness of graph nodes based on *skeleton learning* (SKE) [20]. It assigns each node a probability of being a representative and minimizes the communication cost from a random node to its corresponding representative. This method is different from other approaches in two aspects. First, the learned distribution has low entropy with the representative nodes having large probability and the non-representative nodes having probability close to zero. Second, the representative nodes are *mutually exclusive*: if a node already has a nearby representative, it will penalize the representativeness of other nearby candidates. Such exclusiveness is not implemented as heuristics in a greedy manner [18], but fits in a minimizing message length framework and allows global coordination in arranging the representatives.

The rest of this paper is outlined as follows. Section 2 introduces the skeleton learning. Section 3 presents the recent development of this approach on social network analysis. Section 4 and Section 5 show the experimental results on toy datasets and real social networks, respectively. Finally, Section 6 concludes.

2 Skeleton Learning

This section briefly reviews the recently proposed skeleton learning [20]. Given a set of samples $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n \subset \Re^D$, this unsupervised method learns a probability α_i for each \mathbf{x}_i ($\sum_{i=1}^n \alpha_i = 1$), so that the probability mass highlights the samples on the “skeleton” of the structures and diminishes on outliers.

The input samples are first encoded into a probability matrix $P_{n \times n} = (p_{i|j})$ as in Stochastic Neighbour Embedding [21], so that

$$p_{i|j} = \frac{\exp(-h_j \|\mathbf{x}_i - \mathbf{x}_j\|^2)}{\sum_{i:i \neq j} \exp(-h_j \|\mathbf{x}_i - \mathbf{x}_j\|^2)} \quad (1)$$

denotes the probability of node i receiving a message originated from node j with respect to space adjacency. In Eq.(1), $\|\cdot\|$ is 2-norm, and $h_j > 0$ is a kernel

width parameter, which can be fixed so that the entropy of $p_{\cdot|j}$ equals to a pre-specified constant [21]. The latent distribution $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)$ corresponds to a discrete random variable, or the index $j \in \{1, 2, \dots, n\}$ of a random *skeleton point* $\mathbf{x}_j \in \mathcal{X}$. By assumption, this random point sends out a message with respect to $P_{n \times n}$. Any $\mathbf{x}_i \in \mathcal{X}$, upon receiving such a message, can infer the location of the skeleton point using Bayes' rule as

$$q_{j|i} = \frac{\alpha_j \cdot p_{i|j}}{\sum_{j:j \neq i} \alpha_j \cdot p_{i|j}}. \quad (2)$$

The objective is to optimally route from a random location in \mathcal{X} to its skeleton point, which is implemented by minimizing

$$E(\boldsymbol{\alpha}) = - \sum_{i=1}^n \sum_{j:j \neq i} q_{j|i} \log p_{j|i} \quad (3)$$

with respect to $\boldsymbol{\alpha}$. Through such minimization, a compact set of skeleton positions with large α_i can be learned. As compared to clustering methods, the skeleton model is a prior distribution defined on the observations, and the effective number of skeleton points shrinks continuously during learning. Therefore no model selection is necessary to determine an appropriate number of clusters, and the learning process can be terminated at anytime to produce reasonable results. However, the effect of the kernel width parameter h_j must be carefully investigated depending on application. In image denoising [20], it shows better performance in preserving the manifold structure as compared to a state-of-the-art denoising approach [22]. The gradient-based algorithm has a complexity of $O(n^2)$ at each step, which limits its scalability.

3 Skeleton Learning on Graphs

The skeleton learning method introduced in Section 2 is performed on a set of coordinates for denoising and outlier detection. This section extends the idea to graph datasets and discusses related problems. Assume the input data is a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, \dots, n\}$ is the set of vertices and $\mathcal{E} = \{(i, j)\}$ is the set of edges. Throughout this paper, an undirected graph is treated as its directed version by replacing each edge $i \leftrightarrow j$ with two opposite arcs (i, j) and (j, i) . We aim to discover a random *representative* characterized by a discrete distribution $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)$ defined on \mathcal{V} , so that any random node in \mathcal{V} can communicate with it in the most efficient and economical way.

We first construct a channel between two random nodes so that the communication cost can be measured. The input graph \mathcal{G} can be equivalently represented by its *normalized adjacency matrix* $A = (a_{ij})$ with

$$a_{ij} = \delta_{ij}/d_i, \quad \delta_{ij} = \begin{cases} 1 & \text{if } (i, j) \in \mathcal{E}; \\ 0 & \text{otherwise,} \end{cases} \quad (4)$$

where d_i is the outdegree of the node i . If a node i has at least one outgoing link, the i 'th row of A defines a discrete distribution representing how likely i influences the other nodes according to the graph structure. To deal with nodes with no outgoing links or incoming links, we allow each node i to teleport to another random node with a small probability ν . In social networks, such teleportation models i 's influence through external ways not restricted by the network [7, 23]. Consider i sending a message to one unique receiver other than i at time 0. The probability that node j receives this message in one time step can be defined as $p_{j|i} = (1 - \nu)a_{ij} + \nu/(n - 1)$ ($j \neq i$). In matrix form it is equivalently

$$P = (1 - \nu)A + \frac{\nu}{n - 1}(\mathbf{e}\mathbf{e}^T - I), \quad (5)$$

where $\mathbf{e} = (1, \dots, 1)^T$ and I is the identity matrix. If we allow this message to pass around in \mathcal{G} for τ times ($\tau = 1, 2, \dots$) after time 0, the probability for each node j holding the message is given by the i 'th row of the matrix P^τ . It represents i 's indirect influence over \mathcal{G} through information spreading. In the extreme case when $\tau \rightarrow \infty$, all rows of P^τ will tend to be the same, or the equilibrium distribution corresponding to the PageRank (PR) measure [7]. To distinguish the “outgoing ability” of different nodes, τ should be a small value (e.g., 1 or 2) so that each node can only reach a local region around itself. Without loss of generality, we focus on the case $\tau = 1$ unless otherwise specified.

Assume a latent prior distribution $\boldsymbol{\alpha}$ of each node being the information source. The sender i , upon any node j receiving a message, can be identified with Bayesian inference in Eq.(2) (with i and j interchanged). The total communication cost for every node $j \in V$ to reply to its information source is given by Eq.(3). By minimizing such a cost, $\boldsymbol{\alpha}$ can be learned so that this communication loop is established in the optimal way.

More intuitively, consider without loss of generality the graph \mathcal{G} as a social network of n persons. A directed link $(i, j) \in \mathcal{E}$ means that i could easily influence j because of personal relationship, etc. One real-life example could be j “follows” i on some microblogging website. In this context, the meaning of being a representative can be understood from Eq.(3). To minimize $E(\boldsymbol{\alpha})$, on average $-\log p_{j|i}$ should be small, which means the representative j can perceive news from its surrounding nodes easily. As another condition, $q_{\cdot|i}$ should have low entropy, which means each person i selects the candidate which influences i the most, and *deselects* other nearby candidates being its representative.

Implementation

The skeleton learning is implemented by gradient descent to minimize $E(\boldsymbol{\alpha})$. The gradient of $E(\boldsymbol{\alpha})$ has the form [20]

$$\frac{\partial E}{\partial \alpha_j} = -\frac{1}{\alpha_j} \sum_{i:i \neq j} q_{j|i,\alpha} \left(\log p_{j|i} - \sum_{l:l \neq i} q_{l|i,\alpha} \log p_{l|i} \right). \quad (6)$$

Algorithm 1. Skeleton Learning on Graph Datasets

Input: A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with n nodes; a small teleport probability ν
Output: A discrete distribution $\alpha = (\alpha_1, \dots, \alpha_n)$ to measure the
representativity of each node

```

1 begin
2    $\alpha \leftarrow (1/n, \dots, 1/n); \gamma \leftarrow \gamma_0; p_0 = \nu/(n-1)$            //  $\gamma$ : learning rate
3   repeat
4      $\nabla = (\nabla_1, \dots, \nabla_n) \leftarrow \mathbf{0}; \nabla_c \leftarrow 0$ 
5     foreach node  $i$  in  $\mathcal{S}$  (see comments in the end) do
6        $p_i \leftarrow (1-\nu) \sum_{j:j \rightarrow i} \alpha_j / d_j + \nu(1-\alpha_i)/(n-1)$ 
           //  $d_j$  is the out-degree of node  $j$ 
7        $E_i \leftarrow -(1-\nu) \sum_{j:j \rightarrow i} \alpha_j \log p_{j|i} / d_j - \nu \sum_{j:j \neq i} \alpha_j \log p_{j|i} / (n-1)$ 
8        $E_i \leftarrow E_i / p_i$ 
9       foreach  $j$  in  $\text{Pre}(i)$  do      //  $\text{Pre}(i)$  is the set of predecessors
10      |  $\nabla_j \leftarrow \nabla_j - (1-\nu)(\log p_{j|i} + E_i) / (d_j p_i)$ 
11    end
12    foreach  $j$  in  $\text{Suc}(i)$  do      //  $\text{Suc}(i)$  is the set of successors
13      |  $\nabla_j \leftarrow \nabla_j - \nu(\log p_{j|i} - \log p_0) / ((n-1)p_i)$ 
14    end
15     $\nabla_i \leftarrow \nabla_i + \nu(\log p_0 + E_i) / ((n-1)p_i)$ 
16     $\nabla_c \leftarrow \nabla_c - \nu(\log p_0 + E_i) / ((n-1)p_i)$ 
17  end
18   $\nabla \leftarrow \nabla + \nabla_c$ 
19   $\alpha \leftarrow \alpha \circ \exp(-\gamma \nabla \circ \alpha)$            // " $\circ$ " is the element-wise product
20  normalize  $\alpha$  so that  $\sum_i \alpha_i = 1$ 
21  until convergence or the number of iterations reaches a pre-specified value
22 end
/* If  $\mathcal{S} = \mathcal{V}$ ,  $\alpha$  is updated on every full scan of the whole dataset;
if  $\mathcal{S}$  is a small random subset of  $\mathcal{V}$ ,  $\alpha$  is updated with stochastic
gradient descent (more efficient and scalable) */

```

Along $-\partial E / \partial \alpha_j$, the candidate weight α_j is adjusted at each step. Intuitively Eq.(6) says, for each node i within j 's reachable range, j serves as a potential information source of i (the value $q_{j|i,\alpha}$ is significant enough based on Eq.(2)), and such i provides feedback to α_j based on how efficiently it can reach back to j . If the length $-\log p_{j|i}$ is shorter than the average length $-\sum_{l:l \neq i} q_{l|i,\alpha} \log p_{l|i}$, then $\partial E / \partial \alpha_j < 0$ and α_j increases, which means that i “votes” for j to become its representative. On the other hand, if the route $i \rightarrow j$ is too costly, i casts a negative vote for j . This type of gradient was discussed in a statistical machine learning framework [24] and further explored here in a non-parametric setting.

In general, each gradient descent step requires $O(|\mathcal{V}|^2)$ computation [20] because P is dense. However, the fact that most entries of the transition matrix P equal $\nu/(n-1)$ can lead to more efficient implementations. On graph datasets, the gradient in Eq.(6) is further written as

$$\begin{aligned}
\frac{\partial E}{\partial \alpha_j} &= -\frac{1-\nu}{d_j} \sum_{i:j \rightarrow i} \Delta_{ij} - \frac{\nu}{n-1} \sum_{i:i \rightarrow j} (\Delta_{ij} - \Delta_{i0}) - \frac{\nu}{n-1} \sum_{i:i \neq j} \Delta_{i0}, \\
\Delta_{ij} &= \frac{1}{p_i} \left(\log p_{j|i} + E_i \right), \quad \Delta_{i0} = \frac{1}{p_i} \left(\log \frac{\nu}{n-1} + E_i \right), \\
p_i &= (1-\nu) \sum_{j:j \rightarrow i} \frac{\alpha_j}{d_j} + \nu \frac{1-\alpha_i}{n-1}, \\
E_i &= - \sum_{j:j \neq i} q_{j|i} \log p_{j|i} = -\frac{1-\nu}{p_i} \sum_{j:j \rightarrow i} \frac{\alpha_j}{d_j} \log p_{j|i} \\
&\quad - \frac{\nu}{p_i(n-1)} \sum_{j:i \rightarrow j} \alpha_j \log p_{j|i} - \frac{\nu}{p_i(n-1)} \log \frac{\nu}{n-1} \left(1 - \alpha_i - \sum_{j:i \rightarrow j} \alpha_j \right).
\end{aligned}$$

In Algorithm 1, the simple gradient descent has a computational complexity of $O(|\mathcal{E}|)$ in each iteration. The stochastic gradient descent (SGD) [25] version reduces this computation time to $O(\max(d_i) \cdot |\mathcal{S}|)$. Besides the learning rate, the algorithm has only one parameter ν . By default we set $\nu = 0.2$ in the following experiments. On real large social networks, the node degrees follow an exponential distribution, which may lead to trivial solutions if ν is too large. For example, one node with significant number of links could become the sole representative over the whole network and communicate with the unconnected nodes through teleport. In this case we have to lower the value of ν to penalize the teleport communication and to discover more representatives.

4 Toy Problems

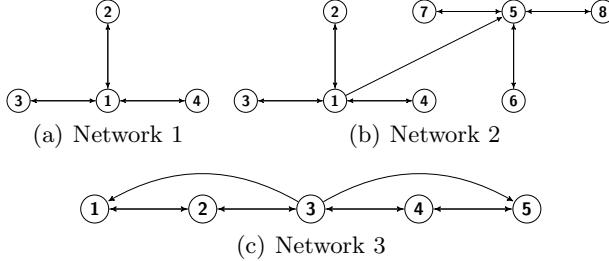
Figure 1 presents several toy social networks. Table 1 shows the α_i value and the PageRank value of each node. In Figure 1(a), only one person (node 1) is acquainted to all the others. SKE has successfully identified it as the sole representative. Figure 1(b) shows two groups of people, each with a central hub (node 1 and node 5), and a link from node 1 to node 5. The SKE values are very concentrated on these two centers, with node 5 having slightly larger weight due to the fact that no edge exists from node 5 to node 1. This type of penalty becomes clearer in the network shown in Figure 1(c). In this example, each node has exactly the same indegree. There is one node 3 which links to all the other nodes, while half of the linked nodes do not respond. Its α_i is close to zero, meaning that it has been identified as a spam node. In general, the PageRank values are less concentrated and do not reveal such information.

The proposed method is further tested on two “Primary School Cumulative Networks” [26]¹, where the nodes represent students or teachers and the edges represent their face-to-face interactions. We only consider *strong interactions*, which are defined as all such edges (A, B) if A and B has interacted for at least

¹ <http://www.sociopatterns.org/datasets/primary-school-cumulative-networks>

Table 1. SKE and PageRank results on the toy networks in Figure 1

	Node	1	2	3	4	5	6	7	8
Network 1	PageRank	0.46	0.18	0.18	0.18				
	SKE	0.99	0.00	0.00	0.00				
Network 2	PageRank	0.17	0.06	0.06	0.06	0.32	0.11	0.11	0.11
	SKE	0.41	0.01	0.01	0.01	0.44	0.04	0.04	0.04
Network 3	PageRank	0.17	0.22	0.22	0.22	0.17			
	SKE	0.20	0.30	0.00	0.30	0.20			

**Fig. 1.** Toy social networks

2 minutes and on at least 2 occasions. As a result, there are 236 nodes and 1954 edges in network 1, and there are 238 nodes and 2176 edges in network 2.

Figure 2 shows the visualization of the network on day 1, where the SKE value α_i is intuitively presented by the size of the corresponding circle and the node degree is presented by the color density. We see that the representatives (large circles) do not necessarily have high degrees (dense color), and vice versa. This is further confirmed by looking at the accurate measurements given in Table 2. Among the top ranked nodes, some have small degrees, such as node “1843” in day 1, node “1521” and “1880” in day 2. We further look at the average degree of their neighbours (D_n). All these three nodes have a relative small value of D_n , which means some of their neighbours are poorly-connected. The connections with these non-so-popular nodes are highly valued in the SKE measurement. On the other hand, among the bottom ranked nodes, some have large degrees, such as node “1628” and node “1428” in day 1, node “1766” and “1778” in day 2. Generally they have a relative large D_n value. Although they are well-connected, their relationships are mostly established with popular nodes, and thus have little value. We also see that the SKE measure has a “sharper” distribution as compared to PageRank, where the tail nodes have very small values. The effective number of skeleton points (given by $\exp\{-\sum_i \alpha_i \log \alpha_i\}$, or the number of uniformly distributed points with the same entropy as α) is 95.8 and 91.3 in network 1 and 2, respectively.

5 Information Diffusion on Collaboration Networks

We test the proposed approach as a seeding method for information diffusion in social networks [15], so that a small subset of *seeds* (corresponding to the representatives as discussed above) could influence as many nodes as possible.

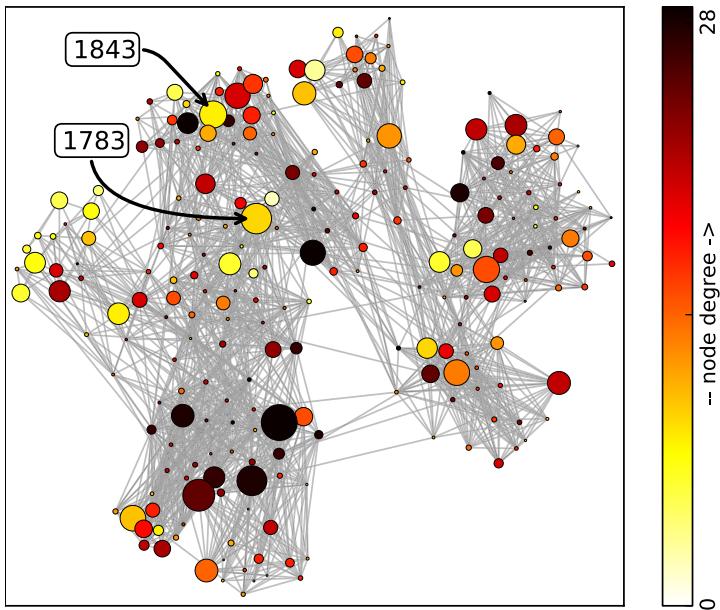


Fig. 2. SKE results on Primary School Cumulative Networks (day 1). The node size represents the SKE value. The node color represents its degree (the darker the higher).

Table 2. Different measures of nodes in the cumulative network dataset. For each day, the columns show (1) node ID (2) class ID if the node is a child, or “Teachers” (3) degree D (4) average degree D_n of its neighbours (5) PageRank (PR) in percentage (6) α_i (SKE) in percentage. The nodes are ordered by α_i .

Cumulative Network Day 1						Cumulative Network Day 2					
Node	Class	D	D_n	PR	SKE	Node	Class	D	D_n	PR	SKE
1890	2B	35	19.3	0.80	4.3	1745	Teachers	28	21.5	0.57	4.4
1650	Teachers	25	16.0	0.65	3.4	1521	Teachers	8	6.9	0.47	3.3
1783	1A	19	18.6	0.46	3.1	1668	Teachers	21	19.2	0.47	3.0
1743	2B	28	18.9	0.67	3.0	1443	5B	23	17.1	0.58	2.9
1843	3A	15	16.1	0.41	2.4	1880	4B	8	8.3	0.43	2.7
:						:					
1628	2A	23	21.3	0.57	0.0	1766	1A	11	18.7	0.31	0.0
1649	2A	7	16.3	0.26	0.0	1799	1A	9	19.6	0.26	0.0
1483	5A	4	16.0	0.18	0.0	1772	1A	8	15.9	0.26	0.0
1858	2B	7	21.0	0.23	0.0	1519	4A	2	21.0	0.13	0.0
1511	5A	9	20.3	0.27	0.0	1819	4B	5	12.6	0.27	0.0
1428	5B	12	21.1	0.33	0.0	1778	1A	11	22.9	0.29	0.0
1803	4B	6	18.2	0.22	0.0	1753	Teachers	3	23.0	0.14	0.0
1710	2B	5	24.0	0.19	0.0	1710	2B	8	29.6	0.21	0.0
1854	2B	5	23.4	0.18	0.0	1807	4B	2	11.5	0.16	0.0
1898	2B	5	23.8	0.19	0.0	1760	1A	3	23.7	0.14	0.0

We use the collaboration networks in the Stanford Large Network Dataset Collection² [27]. **ca-GrQc** is a co-authorship network of physics publications, compiled from the General Relativity section of Arxiv. It has 5,242 nodes representing authors and 14,496 edges representing co-authorships. Similarly, **ca-HepTh**, **ca-HepPh** and **ca-AstroPh** are collaboration networks of different domains on Arxiv. Their sizes denoted by #nodes/#edges are 9,877/25,998, 12,008/118,521, 18,772/198,110, respectively. All datasets are undirected and unweighted, which means the accurate number of times that two authors have collaborated is discarded. For each dataset, five different seeding methods are applied, which select seeds based on descending degree, descending PageRank value, descending SKE value (α_i) computed by simple gradient descent and stochastic gradient descent, and the degree discount heuristic [18], respectively. The damping factor in PageRank is set to be 0.85 by convention. The teleport probability ν in SKE is set to be 0.2.

To evaluate the seeding quality, we apply two different diffusion models, namely Independent Cascade Model (ICM) [17] and Linear Threshold Model (LTM) [16], once the selected seeds are marked as being *activated*. Both of these models expand the activated set of nodes in discrete steps with respect to the network structure. In ICM, an activated node v has a single chance to activate a neighbour w with the probability $p_{v,w}$. By discarding the number of times that two authors have cooperated, diffusion with ICM becomes harder because v has one link instead of several parallel links connecting with j . In LTM, a node v will be activated once the proportion of activated neighbours reaches a node-specific threshold θ_v . At convergence, the size of the activated set quantifies the influence of the initial seeds. To average the effect of random factors, the experiment for each seeding set is repeated for 100 times.

Figure 3(a-d) shows the size of the activated set varying with the number of seeds on **ca-GrQc** and **ca-HepTh** using ICM. Figure 3(g) condenses the results on **ca-AstroPh** and **ca-HepPh** using 100 seeds with ICM. Generally SKE and DegreeDiscount outperform PageRank, which in turn outperforms Degree. Basically SKE tries to place the minimum number of seeds while guaranteeing the influence coverage over different regions. The good performance of DegreeDiscount is due to a similar mechanism to penalize clustered seeds [18]. While the two approaches are comparable on **ca-GrQc**, seeding with SKE is obviously better on **ca-HepTh** (ICM probability=0.2), **ca-AstroPh** and **ca-HepPh** (ICM probability=0.1). Moreover, SKE can adapt to network changes through online learning, which is not straightforward for DegreeDiscount. Note, the SGD version of SKE has comparable performance with the simple implementation and is about ten times faster. Figure 3(h) shows the results with different values of ν in the range [0.01, 0.3]. We see that the influence coverage has a small variation and thus is not sensitive to this configuration.

As Figure 3(e-f) displays, the performance of SKE falls behind PageRank on the LTM experiments. Such results are expected. LTM, as well as the weighted independent cascade model [15], requires a certain proportion of v 's neighbours

² <http://snap.stanford.edu/data/index.html>

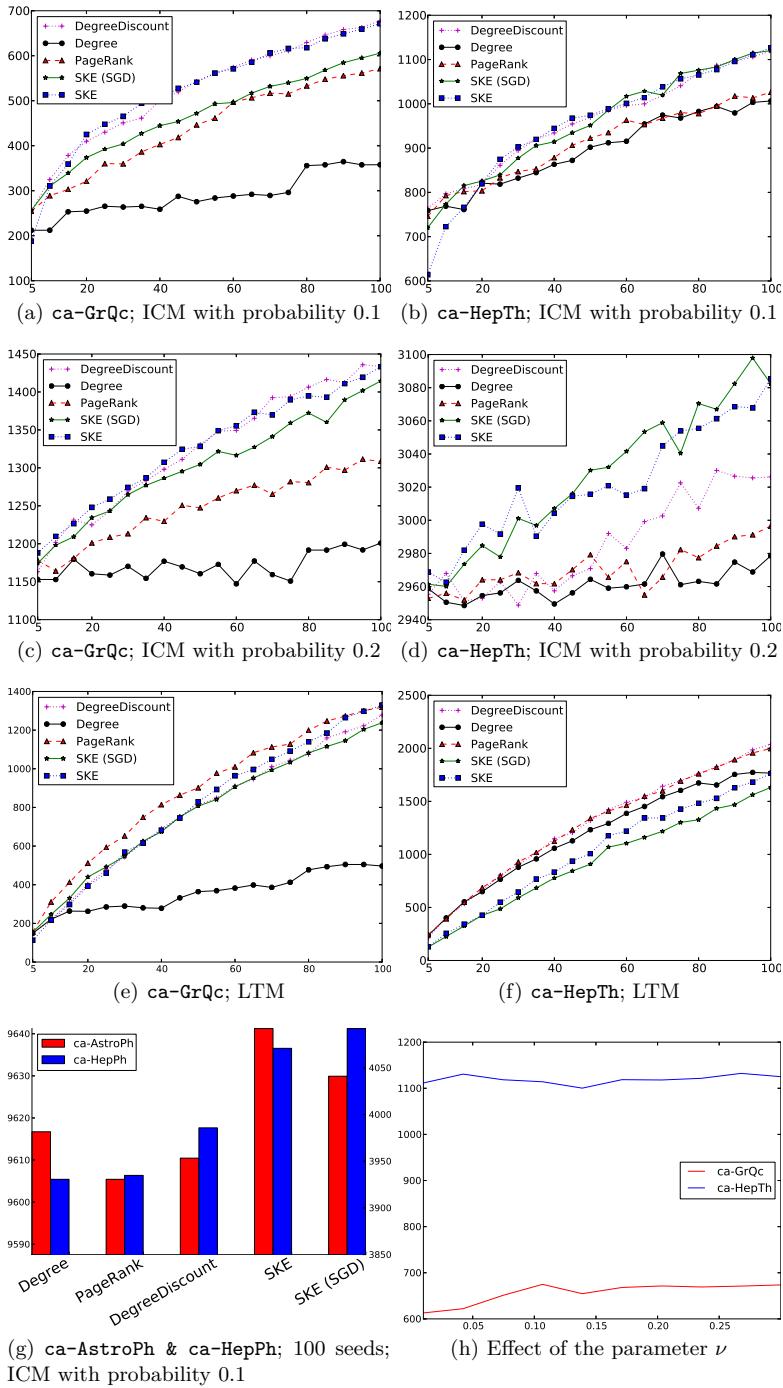


Fig. 3. The number of influenced nodes on collaboration networks

to be activated in order for v to be activated. However, SKE places seeds so that they have less overlap and thus is not recommended in similar diffusion models, where more exposure to the spread increases the likelihood of activation.

6 Conclusion

We have extended a recently proposed skeleton learning approach [20] to social network analysis. From an information diffusion perspective, the method aims to identify representative individuals that have greater potential influence over the network. In a minimizing communication cost framework, the gradient-based optimization naturally allows nodes to cast negative votes to each other in order to derive a set of mutually exclusive candidates. Consequently, the resulting representatives lie in different regions which helps avoid overlap of neighbour sets. The computational complexity in each optimization step is improved from $O(|\mathcal{V}|^2)$ to $O(|\mathcal{E}|)$ (\mathcal{V} : node set; \mathcal{E} : edge set) and is further boosted with stochastic gradient descent. As presented in our experiments, this approach is able to discover important individuals who have fewer connections and are thus not considered by traditional methods such as PageRank. On real collaboration networks with the independent cascade model [17], the proposed method outperforms the traditional ranking algorithms and the degree discount heuristic [18]. As for future work, we are interested in varying this technique for linear threshold model [16] and exploring other application scenarios such as community finding [5, 28].

Acknowledgements. This work is supported by Swiss National Science Foundation (SNSF) via the NCCR (IM)2, the European COST Action on Multilingual and Multifaceted Interactive Information Access (MUMIA) via the Swiss State Secretariat for Education and Research (SER), and Irish CLIQUE Strategic Research Cluster (SFI grant no. 08/SRC/I1407).

References

- Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 22(8), 888–905 (2000)
- Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: *NIPS* 14, pp. 849–856. MIT Press (2002)
- Luxburg, U.: A tutorial on spectral clustering. *Stat. Comput.* 17(4), 395–416 (2007)
- White, S., Smyth, P.: A spectral clustering approach to finding communities in graphs. In: *Proc. SDM 2005*, pp. 274–285 (2005)
- Newman, M.E.J.: Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E* 74(3), 036104 (2006)
- Xu, K.S., Kliger, M., Chen, Y., Woolf, P.J., Hero III, A.O.: Revealing social networks of spammers through spectral clustering. In: *Proc. ICC 2009*, pp. 735–740 (2009)
- Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. *Comput. Networks ISDN* 30(1-7), 107–117 (1998)

8. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. *J. ACM* 46(5), 604–632 (1999)
9. Lempel, R., Moran, S.: The stochastic approach for link-structure analysis (SALSA) and the TKC effect. *Comput. Netw.* 33(1-6), 387–401 (2000)
10. Chung, F.R.K.: Spectral Graph Theory. Amer. Math. Soc. (1997)
11. Zhou, D., Huang, J., Schölkopf, B.: Learning from labeled and unlabeled data on a directed graph. In: Proc. ICML 2005, pp. 1036–1043 (2005)
12. Agarwal, A., Chakrabarti, S.: Learning random walks to rank nodes in graphs. In: Proc. ICML 2007, pp. 9–16 (2007)
13. Domingos, P., Richardson, M.: Mining the network value of customers. In: Proc. SIGKDD 2001, pp. 57–66 (2001)
14. Richardson, M., Domingos, P.: Mining knowledge-sharing sites for viral marketing. In: Proc. SIGKDD 2002, pp. 61–70 (2002)
15. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: Proc. SIGKDD 2003, pp. 137–146 (2003)
16. Granovetter, M.: Threshold models of collective behavior. *Am. J. Sociol.* 83(6), 1420–1443 (1978)
17. Goldenberg, J., Libai, B., Muller, E.: Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Market. Lett.* 12(3), 211–223 (2001)
18. Chen, W., Wang, Y., Yang, S.: Efficient influence maximization in social networks. In: Proc. SIGKDD 2009, pp. 199–208 (2009)
19. Kitsak, M., Gallos, L., Havlin, S., Liljeros, F., Muchnik, L., Stanley, H., Makse, H.: Identification of influential spreaders in complex networks. *Nature Phys.* 6(11), 889–893 (2010)
20. Sun, K., Bruno, E., Marchand-Maillet, S.: Unsupervised skeleton learning for manifold denoising. In: Proc. ICPR 2012, pp. 2719–2722 (2012)
21. Hinton, G.E., Roweis, S.T.: Stochastic Neighbor Embedding. In: NIPS 15, pp. 833–840. MIT Press (2003)
22. Hein, M., Maier, M.: Manifold denoising. In: NIPS 19, pp. 561–568. MIT Press (2007)
23. Myers, S.A., Zhu, C., Leskovec, J.: Information diffusion and external influence in networks. In: Proc. SIGKDD 2012, pp. 33–41 (2012)
24. Xu, L.: Learning algorithms for RBF functions and subspace based functions. In: Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods and Techniques, pp. 60–94. IGI Global (2009)
25. Bottou, L.: Online algorithms and stochastic approximations. In: Online Learning and Neural Networks. Cambridge University Press (1998)
26. Stehlé, J., Voirin, N., Barrat, A., Cattuto, C., Isella, L., Pinton, J., Quaggiotto, M., Van den Broeck, W., Régis, C., Lina, B., Vanhems, P.: High-resolution measurements of face-to-face contact patterns in a primary school. *PLoS ONE* 6(8), e23176 (2011)
27. Backstrom, L., Leskovec, J.: Supervised random walks: predicting and recommending links in social networks. In: Proc. WSDM 2011, pp. 635–644 (2011)
28. Lancichinetti, A., Radicchi, F., Ramasco, J.J., Fortunato, S.: Finding statistically significant communities in networks. *PLoS ONE* 6(4), e18961 (2011)