Endorsement Deduction and Ranking in Social Networks

Hebert Pérez-Rosés
and Francesc Sebé

Department of Mathematics
University of Lleida,
Lleida, Spain
Email: [hebert.perez,fsebe]@matematica.udl.cat

Josep Ma. Ribó †

Department of Computer Science
University of Lleida,
Lleida, Spain

***Abstract*—Some social networks, such as LinkedIn and ResearchGate, allow user endorsements for specific skills. From the number and quality of the endorsements received, an authority score can be assigned to each profile, with respect to a specific skill. In this paper, we propose an authority score computation method that takes into account the relations existing among different skills. Our method is based on enriching the information contained in the digraph of endorsements corresponding to a specific skill, and then applying a ranking method admitting weighted digraphs, such as PageRank. We describe the method, and test it on a synthetic network of 1493 nodes, fitted with endorsements.**

***Keywords*–*Social Networks; Expertise retrieval;* LinkedIn*;* ResearchGate*;* PageRank**

I. Introduction

LinkedIn and ResearchGate are two prominent examples of professional social networks implementing the *endorsement* feature. A user can declare certain skills, and get endorsed for these skills by other users. From the endorsements shown in an applicant's profile, a potential employer can assess the applicant's skills with a higher level of confidence than say, by just looking at his/her CV.

The two endorsement systems are very similar: For each particular skill, the endorsements make up the arcs of a directed graph [4], whose vertices are the members' profiles. In principle, these endorsement digraphs could be used to compute an authority ranking of the members with respect to each particular skill. This authority ranking may provide a better assessment of a person's profile, and it could also be the core element of an eventual tool for finding people who are proficient in a certain skill, very much like a web search engine [6]. Expertise retrieval is the area of Computer Science that deals with those issues [1][5].

Now, people usually have more than one skill, with some of those skills being related. For example, the skill 'Java' is a particular case of the skill 'Programming', which in turn is strongly related with the skill 'Algorithms'. It may well happen that a person is not endorsed for the skill 'Programming', but he/she is endorsed for the skills 'Java' and 'Algorithms'. From those endorsements it can be deduced with a fair degree of confidence that the person also possesses the skill 'Programming'. In other words, a person's ranking with respect to the skills 'Java' and 'Algorithms' affects his/her ranking with respect to the skill 'Programming'.

If the members of a social network were consistent while endorsing their peers, this 'endorsement with deduction' would not add anything to simple (i.e., ordinary) endorsement. In this ideal world, if Alice endorses Bob for the skill 'Java', she would be careful to endorse him for the skill 'Programming' as well. In practice, however,

1) People are not consistent, for consistency would require a great effort. In an analysis of a small LinkedIn community we have detected several inconsistencies. For example, several users have been endorsed for 'C++' but not for 'Programming'.
2) People are not systematic. That is, people do not usually go over all their contacts systematically to endorse, for each contact and alleged skill, all those contacts which, according to their opinion, deserve such endorsement.
3) Skills lack standardization. In most of these social networks, a set of standard, allowed skills has not been defined. As a result, many related skills (in many cases, almost synonyms) may come up in different profiles of the social network.

Endorsement with deduction may help address those problems, and thus provide a better assessment of a person's skills. More precisely, we propose an algorithm that enriches the digraph of endorsements associated to a particular skill with new weighted arcs, taking into account the correlations between that 'target' skill and the other ones.

A. Contributions of this paper

This paper focuses on professional social networks allowing user endorsements for particular skills, such as LinkedIn and ResearchGate. Our main contributions can be summarized as follows:

1) We introduce *endorsement deduction*: an algorithm to enrich/enhance the information contained in the digraph of endorsements corresponding to a specific skill ('target' skill or 'main' skill) in a social network. This algorithm adds new weighted arcs (corresponding to other skills) to the digraph of endorsements, according to the correlation of the other skills with the 'main' skill. We assume the

existence of an ‘ontology’ that specifies the relationships among different skills.

2) After this pre-processing we can apply a ranking algorithm to the enriched endorsement digraph, so as to compute an authority score for each network member with respect to the main skill. In particular, we have used the (weighted) PAGERANK algorithm for that purpose, but in principle, any ranking method could be used, provided that it admits weighted digraphs. This authority score could be useful for a conceivable tool for searching people having a certain skill. Thus, the results of a query might be displayed in decreasing order of authority.
3) We propose a methodology to validate our algorithm, which does not rely as heavily on the human factor as previous validation methods, or on the availability of private information of the members' profiles. Following this methodology, we test our solution on a synthetic network of 1493 nodes and 2489 edges, similar to LINKEDIN, and fitted with endorsements [13].

To the best of our knowledge, this is the first proposal that ranks users of a social network according to their proficiency in some skill, based on endorsements. Moreover, we are not aware of any other work that suggests to enhance the endorsement digraph corresponding to some particular skill, with information obtained from related skills.

The rest of the paper is organized as follows: Section II provides the essential concepts, terminology and notation that will be used throughout the rest of the paper. It also describes the PAGERANK algorithm, including the variant for weighted digraphs. After that, our proposal is explained in Section III together with a simple example. In Section IV we compare the results obtained by ranking with deduction with those obtained by simple ranking, according to three criteria proposed by ourselves.

II. PRELIMINARIES

A. *Terminology and notation*

A *directed graph*, or *digraph* $D = (V, A)$ is a finite nonempty set V of objects called *vertices* and a set A of ordered pairs of vertices called *arcs*. The *order* of D is the cardinality of its set of vertices V. If (u, v) is an arc, it is said that v is *adjacent from* u. The set of vertices that are adjacent from a given vertex u is denoted by $N^+(u)$ and its cardinality is the *out-degree* of u, $d^+(u)$.

Given a digraph $D = (V, A)$ of order n, the adjacency matrix of D is an $n \times n$ matrix $\mathbf{M} = (m_{ij})_{n\times n}$ with $m_{ij} = 1$ if $(v_i, v_j) \in A$, and $m_{ij} = 0$ otherwise. The sum of all elements in the i-th row of M will be denoted Σm_{i*}, and it corresponds to $d^+(v_i)$.

A *weighted digraph* is a digraph with (numeric) labels or *weights* attached to its arcs. Given $(u, v) \in A$, $\omega(u, v)$ denotes the weight attached to that arc. In this paper, we only consider directed graphs with non-negative weights. The reader is referred to Chartrand and Lesniak [4] for additional concepts on digraphs.

B. PAGERANK *vector of a digraph*

PAGERANK [2][12] is a link analysis algorithm that assigns a numerical weighting to the vertices of a directed graph. The weighting assigned to each vertex can be interpreted as a relevance score of that vertex inside the digraph.

The idea behind PAGERANK is that the relevance of a vertex increases when it is linked from relevant vertices. Given a directed graph $D = (V, A)$ of order n, assuming each vertex has at least one outlink, we define the $n \times n$ matrix $\mathbf{P} = (p_{ij})_{n\times n}$ as,

$$p_{ij} = \begin{cases} \frac{1}{d^+(v_i)} & \text{if } (v_i, v_j) \in A, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Those vertices without oulinks are considered as if they had an outlink pointing to each vertex in D (including a loop link pointing to themselves). That is, if $d^+(v_i) = 0$ then $p_{ij} = 1/n$ for each j. Note that $\mathbf{P}$ is a stochastic matrix whose coefficient p_{ij} can be viewed as the probability that a surfer located at vertex v_i jumps to vertex v_j, under the assumption that the next movement is taken uniformly at random among the arcs emanating from v_i. When the surfer falls into a vertex v_i such that $d^+(v_i) = 0$, then he/she is able to restart the navigation from any vertex of D uniformly chosen at random. So as to permit this random restart behaviour when the surfer is at any vertex (with a small probability $1 - \alpha$), a new matrix $\mathbf{P}_\alpha$ is created as,

$$\mathbf{P}_\alpha = \alpha\mathbf{P} + (1 - \alpha)\frac{1}{n}\mathbf{J}^{(n)}, \tag{2}$$

where $\mathbf{J}^{(n)}$ denotes the order-n all-ones square matrix.

By construction, $\mathbf{P}_\alpha$ is a positive matrix [11], hence, $\mathbf{P}_\alpha$ has a unique positive eigenvalue (whose value is 1) on the spectral circle. The PAGERANK *vector* is defined to be the (positive) left-hand eigenvector $\mathcal{P} = (p_1, \ldots, p_n)$ with $\sum_i p_i = 1$ (the left-hand Perron vector of $\mathbf{P}_\alpha$) associated to this eigenvalue. The probability α, known as the *damping factor*, is usually chosen to be $\alpha = 0.85$.

The relevance score assigned by PAGERANK to vertex v_i is p_i. This value represents the long-run fraction of time the surfer would spend at vertex v_i.

C. PAGERANK *vector of a weighted digraph*

When the input digraph is weighted, the PAGERANK algorithm is easily adapted so that the probability that the random surfer follows a certain link is proportional to its (positive) weight [15]. This is achieved by slightly modifying the definition, previously given in (1), of matrix $\mathbf{P}$ so that,

$$p_{ij} = \begin{cases} \frac{\omega(v_i, v_j)}{\sum_{v \in N^+(v_i)} \omega(v_i, v)} & \text{if } (v_i, v_j) \in A, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Nodes with no outlinks are treated in the same way as before.

III. ENDORSEMENT DEDUCTION AND RANKING

Let us consider a professional network in which users can indicate a set of topics they are skilled in, and be endorsed for those skills by other users. For each skill, we get an endorsement digraph. Our objective is to compute an authority ranking for a particular skill, which is not only based on the endorsement digraph of that particular skill, but also takes into account the endorsement digraphs of other related skills. From now on, the skill for which we want to compute the ranking will be called the *main skill*.

Let $S = \{s_0, s_1, \ldots s_\ell\}$ be the set of all possible skills, with s_0 being the main skill. Let $D_k = (V, A_k)$ denote the endorsement digraph corresponding to skill s_k, and let $\mathbf{M}_k$ be its adjacency matrix.

We now define the *skill deduction matrix* $\mathbf{\Pi} = (\pi_{kt})$ as follows: Given a pair of skills s_k and s_t, π_{kt} represents the probability that a person skilled in s_k also possesses the skill s_t. In other words, from s_k we can infer s_t with a degree of confidence π_{kt}. By definition, $\pi_{kk} = 1$ for all k. In this way, if some user endorses another user for skill s_k but no endorsement is provided for skill s_t, we can deduce that an endorsement (for s_t) should really be there with probability π_{kt}. In general, $\mathbf{\Pi}$ will be non-symmetric and sparse, thus it is better represented as a directed graph with weighted arcs.

Our proposal takes as input the skill deduction matrix $\mathbf{\Pi}$, together with those endorsement digraphs D_k, with $0 < k \leq \ell$, such that $\pi_{k0} > 0$. Without loss of generality, we will assume that the set of skills related to s_0 is $S_0 = \{s_k \mid k \neq 0,\ \pi_{k0} > 0\} = \{s_1, \ldots, s_\ell\}$.

The proposed endorsement deduction method constructs a weighted endorsement digraph $D_0^{we} = (V, A_0^{we})$ on skill s_0, with weights ranging from 0 to 1, considering the endorsements deduced from related skills $\{s_1, \ldots, s_\ell\}$.

1) First of all, if user v_i directly endorsed v_j for skill s_0, that is $(v_i, v_j) \in A_0$, then D_0^{we} has arc $(v_i, v_j) \in A_0^{we}$ with $\omega(v_i, v_j) = 1$ (that endorsement receives a maximum confidence level).
2) If $(v_i, v_j) \notin A_0$ but $(v_i, v_j) \in A_k$, for just one k, $1 \leq k \leq \ell$, then arc (v_i, v_j) is added to D_0^{we} with weight $\omega(v_i, v_j) = \pi_{k0}$, that is, the arc is assigned a weight that corresponds to the probability that v_i also considers v_j proficient in skill s_0, given an existing endorsement for skill s_k.
3) Finally, if $(v_i, v_j) \notin A_0$ but $(v_i, v_j) \in A_{k_1}, \ldots, A_{k_\ell}$, then the arc (v_i, v_j) is assigned a weight corresponding to the probability that v_i would endorse v_j for s_0 given his/her endorsements for $s_{k_1}, \ldots, s_{k_l}$. That is, let "$(s_{k_i} \to s_0)$" denote the event "endorse for skill s_0 given an endorsement for s_{k_i} (its probability is $p(s_{k_i} \to s_0) = \pi_{k_i,0}$) then (v_i, v_j) is assigned a weight that corresponds to the probability of the union event "$\cup_{k_i \in \{k_1, \ldots, k_\ell\}} (s_{k_i} \to s_0)$", assuming those events are independent.

Next, we show how to construct the weighted adjacency matrix of D_0^{we} by iteratively adding deduced information from related skills. Computations are shown in (4). After the k-th iteration, matrix $\mathbf{Q}_k$ corresponds to the weighted digraph of skill s_0 after having added deduced information from skills $s_1, \ldots, s_k$. The matrix computed after the last iteration $\mathbf{Q}_\ell$ corresponds to the weighted adjacency matrix of digraph D_0^{we}. Computations can be carried out as follows,

$$\mathbf{Q}_0 = \mathbf{M}_0 \tag{4a}$$

$$\mathbf{Q}_k = \mathbf{Q}_{k-1} + \pi_{k0}((\mathbf{J}^{(n)} - \mathbf{Q}_{k-1}) \circ \mathbf{M}_k), \quad \text{for } k = 1, \ldots, \ell, \tag{4b}$$

where the symbol '$\circ$' represents the Hadamard or elementwise product of matrices.

Note that (4b) acts on the entries of $\mathbf{Q}_{k-1}$ that are smaller than 1, and the entries equal to 1 are left untouched. If some entry $\mathbf{Q}_{k-1}(i, j)$ is zero, and the corresponding entry $\mathbf{M}_k(i, j)$ is non-zero, then $\mathbf{Q}_{k-1}(i, j)$ takes the value of $\mathbf{M}_k(i, j)$, modified by the weight π_{k0}. This corresponds to the second case above.

If $\mathbf{Q}_{k-1}(i, j)$ and $\mathbf{M}_k(i, j)$ are both non-zero, then we are in the third case above. To see how it works, let us suppose that some entry $\mathbf{M}_0(i, j)$ is zero, but the corresponding entries $\mathbf{M}_1(i, j), \mathbf{M}_2(i, j), \mathbf{M}_3(i, j), \ldots$, are all equal to 1. In other words, person i does not endorse person j for the main skill (skill 0), but does endorse person j for skills $1, 2, 3, \ldots$. In order to simplify the notation, we will drop the subscripts i, j, and we will refer to q_k as the (i, j)-entry of $\mathbf{Q}_k$. Applying (4), we get:

$$\begin{aligned} q_0 &= m_0 = 0 \\ q_1 &= q_0 + \pi_{1,0}(1 - q_0) = \pi_{1,0} \\ q_2 &= q_1 + \pi_{2,0}(1 - q_1) = \pi_{1,0} + \pi_{2,0}(1 - \pi_{1,0}) \\ &= \pi_{1,0} + \pi_{2,0} - \pi_{1,0}\pi_{2,0} \\ &\vdots \end{aligned}$$

which corresponds to the probabilities of the events $(s_1 \to s_0)$, $(s_1 \to s_0) \cup (s_2 \to s_0)$, and so on.

Once we have the matrix $\mathbf{Q}_\ell = (q_{ij})_{n \times n}$, we can apply any ranking method that admits weighted digraphs, such as the weighted PAGERANK algorithm [15]. For that purpose, we have to construct the normalized weighted link matrix $\mathbf{P}$, as in (3):

$$p_{ij} = \begin{cases} \frac{q_{ij}}{\Sigma q_{i*}} & \text{if } \Sigma q_{i*} > 0, \\ \\ \frac{1}{n} & \text{if } \Sigma q_{i*} = 0. \end{cases} \tag{5}$$

Then we compute $\mathbf{P}_\alpha$ from $\mathbf{P}$, as in (2), and we finally apply the weighted PAGERANK algorithm on $\mathbf{P}_\alpha$.

A. An example

As a simple illustration, let us consider a set of three skills: 'Programming', 'C++' and 'Java'. The probabilities relating them, depicted in Figure 1, have been chosen arbitrarily, but in practice, they could have been obtained as a result of some statistical analysis.

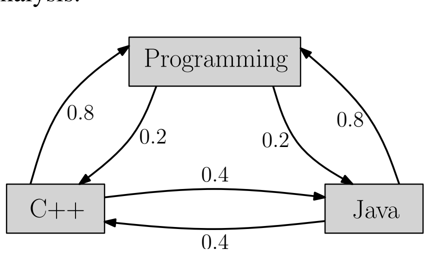
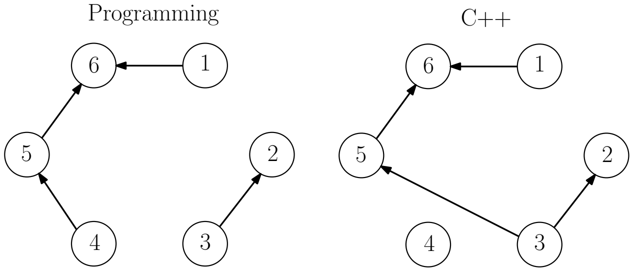


Figure 1. Directed graph representing a skill deduction matrix $\mathbf{\Pi}$.

Let us further assume that we have a community of six individuals, labeled from '1' to '6'. Figure 2 shows the

ISBN: 978-1-61208-444-2


Programming

C++

Figure 2. Endorsements for 'Programming'(left) and 'C++'(right).

endorsement digraphs among the community members for the skills 'Programming' and 'C++'.

Let us suppose that the skill 'Programming' is our main skill (skill 0). Thus, $\mathbf{Q}_0 = \mathbf{M}_0$ is the adjacency matrix of the digraph shown in Figure 2 (left). If we compute the PAGERANK for the skill 'Programming', without considering its relationships with other skills, we get the following scores ($\mathcal{P}(v)$ denotes the PAGERANK score assigned to vertex v): $\mathcal{P}(1) = \mathcal{P}(3) = \mathcal{P}(4) = 0.0988$, $\mathcal{P}(2) = \mathcal{P}(5) = 0.1828$, and $\mathcal{P}(6) = 0.3380$.

In other words, on the basis of the endorsements for 'Programming' alone, the individuals '2' and '5' are tied up, and hence equally ranked.

Now we will include the endorsements for 'C++' in this analysis (skill 1). We apply (4) to compute $\mathbf{Q}_1$, as follows:

$$\mathbf{Q}_1 = \mathbf{Q}_0 + \pi_{1,0}((\mathbf{J}^{(6)} - \mathbf{Q}_0) \circ \mathbf{M}_1),$$

where $\pi_{1,0} = 0.8$, and $\mathbf{M}_1$ is the adjacency matrix of the digraph shown in Figure 2 (right). This yields the endorsement digraph depicted in Figure 3.

The PAGERANK scores assigned to nodes in that digraph are: $\mathcal{P}(1) = \mathcal{P}(3) = \mathcal{P}(4) = 0.0958$, $\mathcal{P}(2) = 0.1410$, $\mathcal{P}(5) = 0.2133$, and $\mathcal{P}(6) = 0.3585$. The individuals '2' and '5' are now untied, and we have better grounds to trust Programmer '5' over Programer '2'.

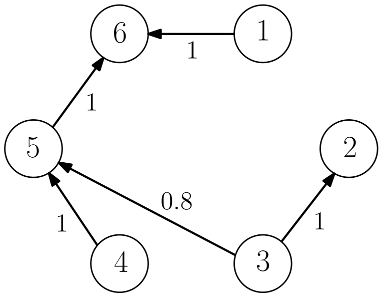
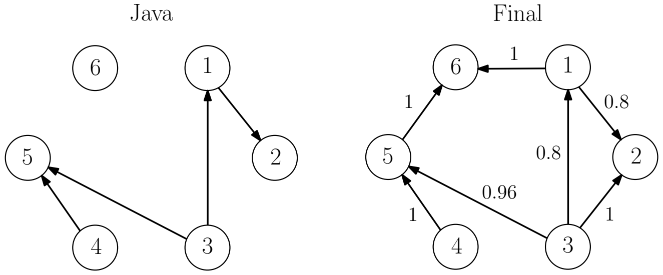


Figure 3. Endorsements for 'Programming', with information deduced from 'C++'.

Let us now suppose that the endorsement digraph for 'Java' is the one given in Figure 4 (left). We can include the endorsements for 'Java'in the same manner:

$$\mathbf{Q}_2 = \mathbf{Q}_1 + \pi_{2,0}((\mathbf{J}^{(6)} - \mathbf{Q}_1) \circ \mathbf{M}_2),$$

where again $\pi_{2,0} = 0.8$. The result is given in Figure 4 (right).


Java

Final

Figure 4. Endorsements for 'Java' (left), and endorsements for 'Programming', with information deduced from 'C++' and 'Java' (right).

If we apply PAGERANK to this final digraph we get: $\mathcal{P}(1) = 0.1178$, $\mathcal{P}(2) = 0.1681$, $\mathcal{P}(3) = \mathcal{P}(4) = 0.0945$, $\mathcal{P}(5) = 0.2027$, and $\mathcal{P}(6) = 0.3224$.

With the aid of the new endorsements, Programmer '1' differentiates itself from Programmers '3' and '4'.

IV. SIMPLE RANKING VS. RANKING WITH DEDUCTION

A. Evaluation criteria

Several criteria and measures have been developed for evaluating information retrieval and ranking systems, such as *precision*, *recall*, *F-measure*, *average precision*, *P@n*, etc. (see [3], Sec. 1.2). All these measures rely on a set of assumptions, which include, among others, the existence of:

1) a benchmark collection E of personal profiles (potential experts),
2) a benchmark collection S of skills,
3) a (total binary) judgement function $r : E \times S \to \{0, 1\}$, stating whether a person $e \in E$ is an expert with respect to a skill $s \in S$.

Unfortunately, none of these assumptions applies in our case. To the best of our knowledge, there does not exist any reliable open-access ground-truth dataset of experts and skills, connected by endorsement relations. To begin with, the endorsement feature is relatively new, and still confined to a few social networks, so that not enough data has accumulated so far. On the other hand, LINKEDIN does not disclose sensitive information of its members (including their contacts or their endorsements), due to privacy concerns.

The third assumption is also problematic: Even if we had a dataset with endorsements, we would still need a 'higher authority', or an 'oracle', to judge about the expertise of a person. Moreover, since our goal is to rank experts, a binary oracle would not suffice.

Traditionally, ranking methods have been validated by carrying out surveys among a group of users [6], which in our opinion, is very subjective and error-prone. We propose a more objective validation methodology, which is based on the following criteria:

1) Our ranking with deduction is close to the ranking provided by PAGERANK. This criterion is based on the assumption that GOOGLE's PAGERANK is widely accepted as a good method, as it has been validated by millions of users for more than fifteen years now. If we use endorsement deduction in connection with PAGERANK, results should not differ too much from PAGERANK.

2) Ranking with deduction results in less ties than PAGERANK. Ties are an expression of ambiguity, hence a smaller number of ties is desirable. In the example of Section III, we have seen that ranking with deduction resolves a tie produced by PAGERANK. However, this has to be confirmed by meaningful experiments.
3) Ranking with deduction is more robust than PAGERANK to *collusion spamming*. Collusion spamming is a form of *link spamming*, i.e., an attack to the reputation system, whereby a group of users collude to create artificial links among themselves, and thus manipulate the results of the ranking algorithm, with the purpose of getting higher reputation scores than they deserve [7][8].

B. Experimental setup and results

Our experimental benchmark consists of a randomly generated social network that replicates some of the features of LINKEDIN at a small scale [13]. LINKEDIN consists of an undirected *base network* (L), or *network of contacts*, and for each skill, the corresponding endorsements form a directed subgraph of (L). In [10], Leskovec formulates a model that describes the evolution of several social networks quite accurately, including LINKEDIN, although this model is limited to the network of contacts (L), and does not account for the endorsements, since that feature was introduced in LINKEDIN later. We have implemented Leskovec's model and used it to generate an undirected network of contacts with 1493 nodes and 2489 edges.

Additionally, we have considered five skills: 1. Programming, 2. C++, 3. Java, 4. Mathematical Modelling, 5. Statistics. We have chosen these skills for two main reasons:

1) These five skills abound in a small LINKEDIN community consisting of 278 members, taken from our LINKEDIN contacts, which we have used as a sample to collect some statistics.
2) These five skills can be clearly separated into two groups or clusters, namely programming-related skills, and mathematical skills, with a large intra-cluster correlation, and a smaller inter-cluster correlation. This is a small-scale representation of the real network, where skills can be grouped into clusters of related skills, which may give rise to different patterns of interaction among skills.

We have computed the co-occurrences of the five skills in our small community, resulting in the co-occurrence matrix $\mathbf{\Pi}_1$ of (6). The entry $\mathbf{\Pi}_1(i,j)$ is the ratio between the number of nodes that have been endorsed for both skills, i and j, and the number of nodes that have been endorsed for skill i alone.

$$\mathbf{\Pi}_1 = \begin{pmatrix} 1 & 0.42 & 0.42 & 0.5 & 0.33 \\ 0.62 & 1 & 0.62 & 0.25 & 0.12 \\ 0.62 & 0.62 & 1 & 0.12 & 0.12 \\ 0.75 & 0.25 & 0.12 & 1 & 0.5 \\ 0.5 & 0.12 & 0.12 & 0.5 & 1 \end{pmatrix} \tag{6}$$

Now, for each skill we have constructed a random endorsement digraph (a random sub-digraph of the base network), in such a way that the above co-occurrences are respected. We have also taken care to respect the relative endorsement frequency for each individual skill. The problem of constructing random endorsement digraphs according to a given co-occurrence matrix is not trivial, and may bear some interest in itself [13]. The base network and the endorsement digraphs can be found at [14].

Next, we have computed two rankings for each skill, one using the simple PAGERANK algorithm, and another one using PAGERANK with deduction. For PAGERANK with deduction we have used the skill deduction matrix $\mathbf{\Pi}_2$ given in (7). This matrix has been constructed by surveying a group of seven experts in the different areas involved. For real cases involving a large set of skills, $\mathbf{\Pi}_2$ must be generated automatically via data mining techniques.

$$\mathbf{\Pi}_2 = \begin{pmatrix} 1 & 0.7 & 0.7 & 0.4 & 0.3 \\ 1 & 1 & 0.6 & 0.4 & 0.3 \\ 1 & 0.7 & 1 & 0.4 & 0.3 \\ 0.3 & 0.2 & 0.2 & 1 & 0.8 \\ 0.3 & 0.2 & 0.2 & 1 & 1 \end{pmatrix} \tag{7}$$

For each skill, we have computed the correlation between both rankings, and the number of ties in each case, according to the first two criteria described above. Additionally, in order to test the robustness of the method to collusion spamming, we have added to each endorsement digraph, a small community of new members (the *cheaters*), who try to subvert the system by promoting one of them (their *leader*) as an expert in the corresponding skill. We have chosen the most effective configuration for such a spamming community, as described in [7], and depicted in Figure 5. Thereupon, we have compared the position of the leader of cheaters in simple PAGERANK with its position in PAGERANK with deduction.

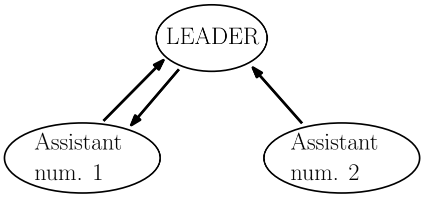


Figure 5. Link spam alliance: Three people collude to promote one of them.

Table I summarizes the results of the aforementioned experiments. We can see that there is a very high correlation between PAGERANK with deduction and PAGERANK without deduction for all skills, according to the values of Kendall's τ and Spearman's ρ correlation coefficients. With respect to the second criterion, the experiments also yield unquestionable results: For all skills, the number of ties is significantly reduced. As for the third criterion, in all cases there is a detectable drop in the position of the leader of cheaters, which may lead us to conclude that PAGERANK with deduction is more robust to collusion spam than simple PAGERANK. The last column of the table contains the difference between the position of the leader with deduction and without deduction, expressed as a percentage.

However, this may not lead us to the conclusion that PAGERANK with deduction is an effective mechanism against collusion spam. Actually, the spam alliance that we have introduced in our experiments is rather weak. If we strengthen the spam alliance, then PAGERANK with deduction may also be eventually deceived.

Several effective mechanisms have been proposed to fight collusion spam, an example being the so-called *asymmetric*

TABLE I. Results of the experiment with low-density endorsement digraphs

Skill	Number of endor-sements (arcs)	Correlation		Number of ties			Position of leader		
		ρ	τ	without deduction	with deduction	% reduction	without deduction	with deduction	% fall
Programming	220	0.89	0.76	1460	1316	10%	1	48	3%
C++	140	0.85	0.63	1478	1304	12%	4	48	3%
Java	137	0.85	0.63	1486	1292	13%	1	48	3%
Math Modeling	134	0.85	0.63	1483	1318	11%	1	45	3%
Statistics	128	0.85	0.63	1486	1304	12%	1	45	3%
AVG						11.6%			3%

TABLE II. Results of the experiment with higher-density endorsement digraphs

Skill	Number of endor-sements (arcs)	Correlation		Number of ties			Position of leader		
		ρ	τ	without deduction	with deduction	% reduction	without deduction	with deduction	% fall
Programming	427	0.76	0.63	1428	625	56%	1	175	12%
C++	1793	0.97	0.93	1005	575	43%	66	178	7%
Java	1856	0.97	0.93	1005	566	44%	63	180	8%
Math Modeling	1406	0.95	0.89	1130	652	42%	56	168	7%
Statistics	1447	0.96	0.90	1113	580	48%	58	169	7%
AVG						47%			8%

reputation systems. A complete survey of such systems is given in [9]. Presumably, these mechanisms will give better results when combined with deduction.

On the other hand, our endorsement digraphs are rather sparse. It is reasonable to predict that if we should consider more skills, and if the total number of endorsements should increase, then the effects of PageRank with deduction will be stronger.

In order to verify this prediction, we have carried out a second experiment on the same base network and the same set of skills, increasing the number of endorsements. Thus, we have generated a second set of endorsement digraphs, with a larger number of arcs. This time we cannot enforce the co-occurrences observed in our small LinkedIn community. Subsequently we have performed the same computations on this second set of endorsement digraphs, obtaining the results recorded in Table II. These results fully confirm our prediction: There is an increase in the correlation coefficients (except in one case), as well as a larger reduction in the number of ties, and a more significant fall in the position of the leader of cheaters.

Acknowledgements

The authors acknowledge partial support by the Spanish Government under projects TIN2010-18978 and MTM2013-46949-P, and by the Government of Catalonia under grant 2014SGR-1666. We also thank the anonymous referees for their timely and constructive comments, which have helped us improve the content and the format of the paper. Finally, the first two authors wish to dedicate this paper to the memory of our friend and co-author Josep Maria Ribó Balust.

References

[1] K. Balog, Y. Fang, M. de Rijke, P. Serdyukov, and L. Si, "Expertise Retrieval," Foundations and Trends in Information Retrieval, vol. 6, no. 1–2, 2012, pp. 127–256.

[2] P. Berkhin, "A Survey on PageRank Computing," Internet Mathematics, vol. 2, no. 1, 2005, pp. 73–120.

[3] S. Ceri, A. Bozzon, M. Brambilla, E. Della Valle, P. Fraternali, and S. Quarteroni, Web Information Retrieval. Springer, 2013.

[4] G. Chartrand and L. Lesniak, Graphs and Digraphs. CRC Press, Boca Raton, fourth ed., 2004.

[5] H. Deng, I. King, and M. R. Lyu, "Enhanced Models for Expertise Retrieval Using Community-Aware Strategies," IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics, vol. 42, 2012, pp. 93–106.

[6] K. Fujimura, H. Toda, T. Inoue, N. Hiroshima, R. Kataoka, and M. Sugizaki, "BLOGRANGER - A multi-faceted blog search engine," Procs. WWW 2006, 2006, pp. 22–26.

[7] Z. Gyöngyi and H. Garcia-Molina, "Link Spam Alliances," Procs. 31st VLDB, 2005.

[8] Z. Gyöngyi and H. Garcia-Molina, "Web Spam Taxonomy," Procs. AIRWeb, 2005.

[9] K. Hoffman, D. Zage, and C. Nita-Rotaru, "A Survey of Attack and Defense Techniques for Reputation Systems," ACM Computing Surveys, vol. 42, 2009, pp. 1–31.

[10] J. Leskovec, "Dynamics of Large Networks," PhD Thesis, School of Computer Science, Carnegie-Mellon University, 2008.

[11] C. D. Meyer, Matrix Analysis and Applied Linear Algebra. SIAM, 2001.

[12] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web," Technical Report, Stanford InfoLab, 1998.

[13] H. Pérez-Rosés and F. Sebé, "Synthetic Generation of Social Network Data with Endorsements," Journal of Simulation, 2014, DOI:10.1057/jos.2014.29.

[14] H. Pérez-Rosés and F. Sebé, Dataset of Endorsements, http://www.cig.udl.cat/sitemedia/files/MiniLinkedIn.zip, accessed in November, 2015.

[15] W. Xing and A. Ghorbani, "Weighted PageRank Algorithm," Procs. of the 2nd Annual IEEE Conference on Communication Networks and Services Research, 2004, pp. 305–314.