Discovering hidden me edges in a Social Internetworking Scenario*

F. Buccafurri, G. Lax, A. Nocera, D. Ursino

Francesco Buccafurri

bucca@unirc.it

Università Mediterranea di Reggio Calabria

SEBD 2012

Venezia, 24-27 giugno 2012

(*) An Extended version of this paper will appear in the ECML-PKDD 2012 Proc.
Discovering Links among Social Networks
by F.Buccafurri, G. Lax, A. Nocera, D. Ursino.

Motivations

- Online Social Networks
- Social Network Analysis and Social Network Mining
- Graph-based Organization
- Crucial role of relationships
- Social Internetworking Scenario

Motivations

- Bridges
  - http://www.facebook.com/buccafurri
  - http://www.flickr.com/people/bucca/
- Key role of `me` edges
- `me` edges can be declared by users
- Missing `me` edges

Goal

Detecting `me` edges in a Social Internetworking Scenario

- Can common-neighbor approaches for link prediction be applied?
- Our solution is based on a notion of node similarity
- Two contributions:
  - String similarity between the associated accounts
  - Recursive notion of common-neighbor similarity

Social Internetworking Scenario

A t-Social-Internetworking Graph $G = \langle N, E_f \cup E_m \rangle$

- N is partitioned into t subsets $S_1, \ldots, S_t$
- E_f is the set of friend edges
- Each $(a, b) \in E_m$ is such that $a \in S_i$ and $b \in S_j$ with $i \neq j$
- E_m is the set of me edges
- $(a, b) \in E_m$ means that a is a bridge

Similarity Notion

Intuitive definition:

- A missing `me` edge between *a* and *b* is detected if both:
  - there exists a string similarity between the accounts of *a* and *b* (several string similarity functions exist: Jaro-Winkler, Levenshtein, QGrams, Monge-Elkan, Soundex, etc.)
  - and the top-k similar pairs composed by friends of *a* and *b* are similar enough

Similarity Notion

As for the top-k similar pairs we have to define how to select such top-k pairs

Definition 2. Given a positive integer k_0, a pair of nodes $a, b \in N$ such that $S(a) \neq S(b)$, a string-similarity metric Q, and a non-negative integer n we inductively define $Top_Q^n(a, b, k_0)$ as follows:

1. $Top_Q^0(a, b, k_0)$ is any subset of $C = \{(x^a, y^b) \mid x^a \in \Gamma(a), y^b \in \Gamma(b)\}$ containing the top-k_0 elements of C w.r.t. the metric Q.
2. For any $0 < i \leq n$, $Top_Q^i(a, b, k_0)$ is any subset of $C = \{(x^z, y^w) \mid (z, w) \in Top_Q^{i-1}(a, b, k_0), x \in \Gamma(z), y \in \Gamma(w)\}$ containing the top-k_i elements of C w.r.t. the metric Q, where $k_i = \lceil \frac{k_0}{(1+i)^{1+i}} \rceil$.

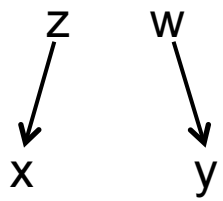
□

Similarity Notion

Introduction of a reduction coefficient to prevent the distortion effect of famous people

Definition 3. Let $x, y, z, w \in N$ be nodes of G such that $x \in \Gamma(z)$, $y \in \Gamma(w)$, and $S(z) \neq S(w)$. We define: $\gamma(x^z, y^w) = min(\delta(x^z), \delta(y^w))$ where $\delta(a^b) = \frac{max(|\Gamma(a)|,|\Gamma(b)|)}{max(|\Gamma(a)|,|\Gamma(b)|)+||\Gamma(a)|-|\Gamma(b)||}$ for any pair of nodes $a, b \in N$. □

Consider the case:



The reduction coefficient is close to 1 when the degree of z and x (w and y) are similar

Similarity Notion

We are now ready to introduce the string similarity operator.

Definition 4. Given a pair of nodes $a, b \in N$ such that $S(a) \neq S(b)$, a string-similarity metric Q, two integers $n \geq 0$ and $k_0 > 0$, we inductively define the *similarity operator* $T_Q^n(a, b, k_0)$ as follows:

1. $T_Q^0(a, b, k_0) = Q(a, b)$.
2. $T_Q^i(a, b, k_0) = (1 - \beta_i) \cdot T_Q^{i-1}(a, b, k_0) + \beta_i \cdot \frac{\sum_{(x^z, y^w) \in Top_{\widetilde{Q}}^i(a,b,k_0)} \widetilde{Q}(x^z, y^w)}{|Top_{\widetilde{Q}}^i(a,b,k_0)|}$.

where $\beta_i = \frac{1}{(i+1)^{i+1}}$ and $\widetilde{Q}(x^z, y^w) = \gamma(x^z, y^w) \cdot Q(x, y)$, for any $x, y, z, w \in N$ nodes of G such that $x \in \Gamma(z)$, $y \in \Gamma(w)$, and $S(z) \neq S(w)$. □

β_i makes quickly less important the common-neighbors contribution as far as the iteration proceeds

Similarity Notion

- The similarity operator can not be effectively if a termination policy is not defined
- To do this, we introduce the following:

Definition 5. Given a pair of nodes $a, b \in N$ such that $S(a) \neq S(b)$, a string-similarity metric Q, an integer number $k_0 > 0$, and a real number $\epsilon > 0$, we define the ϵ-*similarity* $S^{\epsilon}_{Q}(a, b, k_0)$ between a and b w.r.t. Q as $T^{h}_{Q}(a, b, k_0)$, where $h > 0$ is the least number (if any) such that $|T^{h}_{Q}(a, b, k_0) - T^{h-1}_{Q}(a, b, k_0)| < \epsilon$. □

Basic properties

Existence:

Theorem 1. *Given a pair of nodes $a, b \in N$ such that $S(a) \neq S(b)$, a string-similarity metric Q, an integer number $k_0 > 0$, and a real number $\epsilon > 0$, then the ϵ-similarity $S^{\epsilon}_{Q}(a, b, k_0)$ between a and b w.r.t. Q exists.*

Feasibility:

Theorem 2. *Given a pair of nodes $a, b \in N$ such that $S(a) \neq S(b)$, a string-similarity metric Q, and an integer number $k_0 > 0$, then the number of visited nodes in G for the computation of $T^{h}_{Q}(a, b, k_0)$ is $O(d^2 \cdot k_0)$, where d is the maximum node degree.*

me Edge Detection

- It is based on the similarity notion
- We consider only promising pairs
- We found that some of the nodes belonging to the neighbors of the two nodes linked by a me edge are, in their turn, linked by a me edge (with a higher probability)
- Promising pairs are nodes belonging to the neighbors of the two nodes linked by a me edge

me Edge Detection

- Our algorithm starts from $(a,b) \in E_m$
- For each pair (a',b') not in E_m such that $a' \in \Gamma(a)$ and $b' \in \Gamma(b)$
  - if string similarity$(a',b') \leq th_c$, then discard (a',b')
  - else compute $s = S^{\epsilon}_{Q}(a', b', k_0)$
  - if $s > th_d$, then (a',b') is detected as missing me edge
- We remark that the metric here used includes the reduction factor Υ of Definition 3.

Experiments

- Goal: determining the performance of our proposal w.r.t. the state of the art of common-neighbors techniques
- Implementation:
  - Social networks: Twitter, LiveJournal, YouTube, Flickr
  - XFN and FOAF standards to extract user's friendship
  - 2 Quad-Core E5440 processor and 16 GB of RAM with the CentOS 6.0 Server

Experiments

1. Performance of the common-neighbors techniques:

- We start from a set M of 100 node pairs linked by a me edge
- We run each technique for each pair in M trying to detect each me edge
- We denote by M' the set of detected me edges
- We measure the sensitivity of the techniques as $\frac{|M'|}{|M|}$

Coinciding with recall= $\frac{|TP|}{|TP|+|FN|}$

Experiments

Index Name	*Definition*	*Sensitivity*
Salton Index (SAI)	$s_{ab}^{SAI} = \frac{\lvert\Gamma(a) \bigcap \Gamma(b)\rvert}{\sqrt{\lvert\Gamma(a)\rvert \times \lvert\Gamma(b)\rvert}}$	0.01
Jaccard Index (JAI)	$s_{ab}^{JAI} = \frac{\lvert\Gamma(a) \bigcap \Gamma(b)\rvert}{\lvert\Gamma(a) \bigcap \Gamma(b)\rvert}$	0.01
Sorensen Index (SOI)	$s_{ab}^{SOI} = \frac{2\lvert\Gamma(a) \bigcap \Gamma(b)\rvert}{\lvert\Gamma(a)\rvert + \lvert\Gamma(b)\rvert}$	0.01
Hub Promoted Index (HPI)	$s_{ab}^{HPI} = \frac{\lvert\Gamma(a) \bigcap \Gamma(b)\rvert}{min(\lvert\Gamma(a)\rvert, \lvert\Gamma(b)\rvert)}$	0.00
Hub Depressed Index (HDI)	$s_{ab}^{HDI} = \frac{\lvert\Gamma(a) \bigcap \Gamma(b)\rvert}{max(\lvert\Gamma(a)\rvert, \lvert\Gamma(b)\rvert)}$	0.01
Leicht-Holme-Newman Index (LHNI)	$s_{ab}^{LHNI} = \frac{\lvert\Gamma(a) \bigcap \Gamma(b)\rvert}{\lvert\Gamma(a)\rvert \times \lvert\Gamma(b)\rvert}$	0.01
Resource Allocation Index (RA)	$s_{ab}^{RA} = \sum_{z \in \Gamma(a) \bigcap \Gamma(b)} \frac{1}{\lvert\Gamma(z)\rvert}$	0.01
Local Path Index (LPI)	$s_{ab}^{LPI} = A^2 + \epsilon A^3$ (A is G's adjacency matrix)	0.03

No effective result is obtained when common-neighbors techniques are adopted

Experiments

2. Sample-driven method validation:

- We start from the set M
- We find a set ¬M of node pairs not connected by a me edge
- We run our technique on each node in M U ¬M to detect me edges
- We measure precision= $\frac{|TP|}{|TP|+|FP|}$ and recall= $\frac{|TP|}{|TP|+|FN|}$ of our technique w.r.t. several string-similarity function

Experiments

Function	*Precision*	*Recall*
Jaro-Winkler	0.558	0.920
QGrams	0.908	0.690
Levenshtein	0.877	0.710
Smith-Waterman	0.840	0.790
Smith-Waterman-Gotoh	0.779	0.810
Monge-Elkan	0.779	0.810
Needleman-Wunch	0.500	1.000
Jaro	0.555	0.910
Soundex	0.500	0.990

QGrams (resp., Needleman-Wunch) proved to be the one capable of assuring the best precision (resp., recall)

Experiments

3. Expert-based method validation:

- We benefit from the support of a human expert
- We start from a set of 160 `me` edges and run our technique (with QGrams) to find new potential `me` and `not me` edges
- A human expert checks the correctness of the classification (true, false and unknown)
- We measure an accuracy equal to 0.85

Conclusion

- We studied the problem of discovering missing me edges in a Social Internetworking Scenario
- Common-neighbors techniques are not effective in this context
- We defined a suitable notion of “inter-social-network” similarity, which is recursive
- We have defined an algorithm to detect whether there is a missing `me` edge between two given nodes
- The experimental analysis has shown the good correctness and completeness of our proposal