Generating Useful Network-based Features for Analyzing Social Networks

Jun Karamon, Yutaka Matsuo and Mitsuru Ishizuka

University of Tokyo

Published in Proc. of AAAI 2008

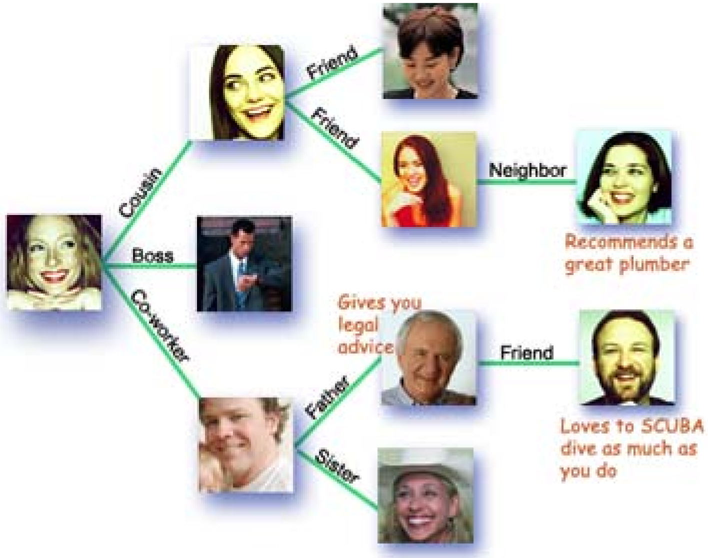
Presented by: Congyi Liu

OUTLINE

- Introduction
- Related Works
- Methodology
- Experiment Result
- Discussion and Conclusion

Social Network

- **Interaction** among users creates a social network among users. Many efforts are underway to analyze user intersections by analyzing social networks among users.
- **Link-based classification:** classifying samples using the relations and links that are present among them.
- **Link prediction:** predicting whether there would be a link between a pair of nodes (in the future) given the (previously) observed links.



Motivation

- **Motivation: Greater potential exists for new features using a network structure.**

- **Problems:**
  - **Numerous methods exist to aggregate features for link-based classification and link prediction;**
  - **The network structure among users influences each user differently;**
  - **It is difficult to determine useful feature aggregation in advance.**

Contribution

Propose an algorithm to identify important network-based features systematically from a given social network to analyze user behavior efficiently.

- **Define general operators that are applicable to the social network;**
- **The combinations of the operators provide different features;**
- **Using the datasets, @cosme and Hatena Bookmark, the performance of link-based classification and link prediction increase compared to existing approaches.**

Features used in Social Network Analysis

- **Density:** the number of edges in a (sub-)graph, expressed as a proportion of the maximum possible number of edges.
- **Centrality measures:** measure the structural importance of a node, e.g. the power of individual actors.
- **Characteristic path length:** the average distance between any two nodes in the network (or a component of it).
- **Clustering coefficient:** the ratio of edges between the nodes within a node's neighborhood to the number of edges that can possibly exist between them.
- **Structural equivalence**, **structural holes** ***…***

Other Features used in Related Works

Features used in link-based classification

Number of friends in a community
Number of adjacent pairs in S
Number of pairs in S connected via a path in E_C
Average distance between friends connected via a path in E_C
Number of community members reachable from S using edges in E_C
Average distance from S to reachable community members using edges in E_C

- S denotes the set of friends of an individual.
- E_c denotes the set of edges in the community C.

Features used in link prediction

name	feature
graphic distance	d_{xy}
common neighbors	$\|\Gamma(x) \cap \Gamma(y)\|)$
Jaccard's coefficient	$\frac{\|\Gamma(x)\cap\Gamma(y)\|}{\|\Gamma(x)\cup\Gamma(y)\|}$
Adamic / Adar	$\sum_{z\in\Gamma(x)\cap\Gamma(y)} \frac{1}{log\|\Gamma(z)\|}$
preferential attachment	$\|\Gamma(x)\| \cdot \|\Gamma(y)\|)$

- d_{xy} is the distance between node x and y.
- $\Gamma(x)$ is the set of nodes adjacent to node x.

Intuition

- Recognizing that traditional studies in social science have demonstrated the usefulness of several indices, we can assume that feature generation toward the indices is also useful.
- **Feature Generation:**

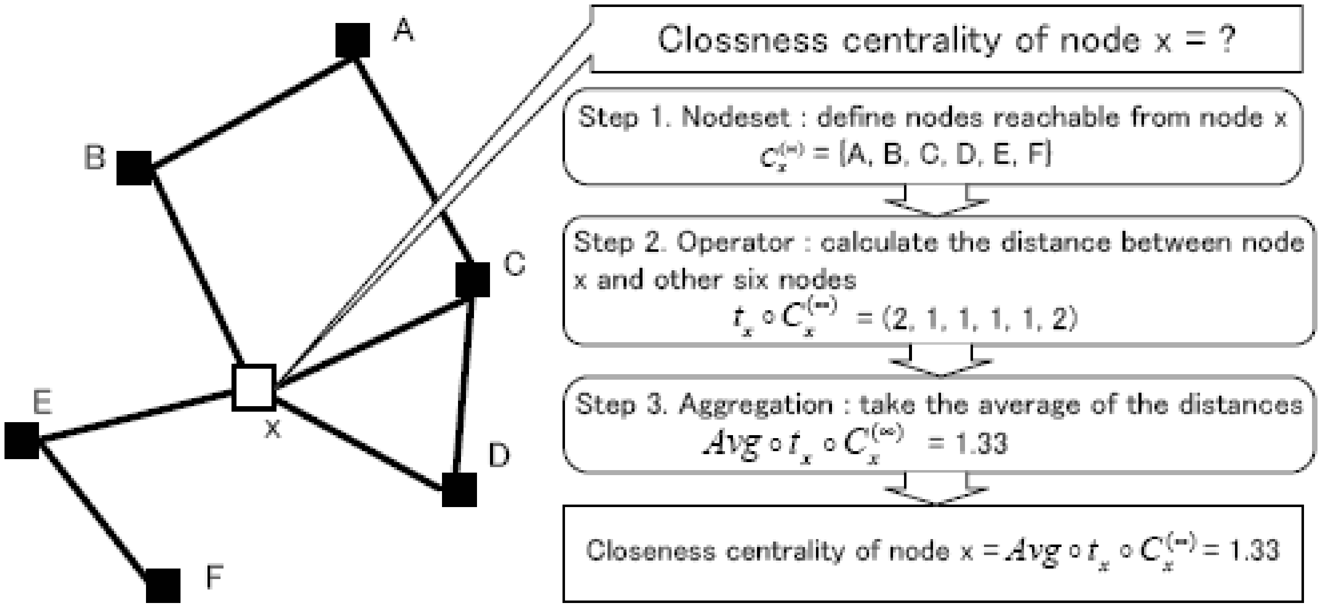


Figure 1: Flow of feature generation

Feature Generation

- **Step 1: Defining a Node Set**
  - **Based on a network structure**
    - i.e. $C_x^{(k)}$ is a set of nodes within distance k from x.
  - **Based on the category of a node**
    - i.e. $N_{A=a}$ Define the node set for which the categorical value A is a
- **Step 2: Operation on a Node Set**
  - **Define operators with respect to two nodes; then expand it to a node set**
    - $s^{(k)}(x, y)$ returns 1 if nodes x and y are within distance k, and 0 otherwise.
    - $u_x(y, z)$ returns 1 if the shortest path between y and z includes node x.
    - $u_x \circ N$ returns a set of values for each pair of $y,z \in N$.
    - $Operator \circ N = \{Operator(x,y) \mid x \in N, y \in N, x \neq y\}$
- **Step 3: Aggregation of Values**
  - **Based on a list of values, several standard operations can be added to the list.**
    - i.e. summation (*Sum*), average (*Avg*), maximum (*Max*), and minimum (*Min*)
- **Step 4: Optionally, we can take the average, difference, or product of two values obtained in Step 3.**

For Link Prediction: Relational Features

- **Generate network-based features which represent a score (i.e. connection weight) on two nodes x and y.**
  - **i.e. Calculate preferential attachment ($|\Gamma(x)| \cdot |\Gamma(y)|$) by respectively counting the links of nodes x and y, thereby obtaining a value as the product of two values.**

- **Define a node set that is relevant to both node x and node y.**
  - **i.e. Common neighbors ($|\Gamma(x) \cap \Gamma(y)|$) depend on the number of common nodes which are adjacent to nodes x and y.**

- **Several operators should be added/modified for link prediction aside from link-based classification to cover more features.**
  - **i.e. Operator u_x is modified as $u_{xy}(z,w)$, which returns 1 if the shortest path between z and w includes l_{xy} and 0 otherwise.**

Operator List

Table 3: Operator list

Step	Notation	Input	Output	Description	LC*	LP*
1	$C_x^{(k)}$	node x	a node set	nodes within distance k from x	√(1)	√(1)
	$C_y^{(k)}$	node y	a node set	nodes within distance k from x		√(1)
	$N_{A=a} \cap C_x^{(k)}$	node x	a node set	nodes within distance to x and the attribute A is a	√(3)	
	$C_x^{(k)} \cap C_y^{(k)}$	node x and y	a node set	nodes within distance k from x and within distance k from y		√(2)
	$C_x^{(k)} \cup C_y^{(k)}$	node x and y	a node set	nodes within distance k from x or within distance k from y		√(2)
2	$s^{(k)}$	a node set	a list of values	1 if connected within distance k, 0 otherwise	√(1)	√(1,2)
	t	a node set	a list of values	distance between a pair of nodes	√(1)	√(1,2)
	t_x	a node set	a list of values	distance between node x and other nodes	√(2)	√(1,2)
	γ	a node set	a list of values	number of links in each node		√(2)
	u_x	a node set	a list of values	1 if the shortest path includes x, 0 otherwise	√(2)	√(1,2)
	e_x	a node set	a list of values	structural equivalence between node x and other nodes		√(2)
3	*Avg*	a list of values	a value	average of values	√(1)	√(1,2)
	Sum	a list of values	a value	summation of values	√(1)	√(1,2)
	Min	a list of values	a value	minimum of values	√(1)	√(1,2)
	Max	a list of values	a value	maximum of values	√(1)	√(1,2)
4	*Diff*	two values	value	difference of two values		√(1,2)
	Avg	two values	value	average of two values		√(1,2)
	Product	two values	value	product of two values		√(1,2)
	Ratio	two values	value	ratio of two values	√(4)	√(1,2)
	Max	two values	value	maximum of two values		√(1,2)
	Min	two values	value	minimum of two values		√(1,2)

- *: LC stands for link-based classification; LP stands for link prediction. The number in the parentheses is the Method number.
- Aggregate operators in Step 4 are optional. This aggregates two feature values obtained in Step 3 into a single feature value.

Constraints

- **64 features for link-based classification.**
- **For link prediction, we can generate 126 features in Method 1 and 160 features in Method 2.**
- **Some resultant features sometimes correspond to well-known indices.**
  - i.e. Denote the network density as $Avg \circ s^{(1)} \circ N.$
- **Regarding link prediction, we can also generate several features that are often used in relevant studies in the literature.**
  - i.e. Common neighbors is realized by $Ratio\{Sum \circ t_{xy} \circ (C_x^{(1)} \cap C_y^{(1)}), Sum \circ t_{xy} \circ (C_x^{(1)} \cup C_y^{(1)})\}$

Datasets

- **@cosme dataset**
  - **Data selection for link-based classification**
    - ① Choose a community as a target; ② select users in the community as positive examples; ③ As negative examples, select those who are not in the community but who have friends who are in the target community.
  - **Data selection for link prediction**
    - ① The positive examples are picked up randomly among links created between time T and T' (T < T' < T''); ② The negative examples are those created between time T' and T''.

- **Hatena Bookmark dataset**
  - **First define similarity between users.**
  - **Create training and test data similarly to the @cosme dataset**

Results: Link-based Classification

Table 4: Recall, precision, and F-value as adding operators.

	(a) @cosme			(b) Hatena Bookmark		
	Recall	Precision	F-val.	Recall	Precision	F-val.
baseline	0.43	0.600	0.495	0.628	0.704	0.661
Method 1	0.387	0.593	0.465	0.499	0.726	0.581
Method 2	0.432	0.581	0.491	0.509	0.720	0.585
Method 3	0.499	0.574	0.532	0.673	0.707	0.681
Method 4	0.604	0.607	0.604	0.692	0.758	0.717

Results: Link-based Classification

Table 5: Top 10 effective features in the @cosme dataset for link-based classification.

Feature	Description
$Sum \circ t \circ (C_x^{(\infty)} \cap N_{C=c})$	Number of links among nodes reachable from x and attribute C is c.
$Sum \circ s^{(1)} \circ C_x^{(1)}$	Number of links among nodes adjacent to x.
$Avg \circ t \circ C_x^{(\infty)}$	Characteristic path length of nodes reachable from x.
$Avg \circ t \circ (C_x^{(\infty)} \cap N_{C=c})$	Characteristic path length of nodes reachable from x and attribute C is c.
$Sum \circ u_x \circ (C_x^{(\infty)} \cap N_{C=c})$	Betweenness centrality of nodes reachable from x and attribute C is c.
$Sum \circ t_x \circ C_x^{(1)}$	Number of nodes adjacent to x.
$Sum \circ s^{(1)} \circ (C_x^{(1)} \cap N_{C=c})$	Number of links among positive nodes adjacent to node x.
$Avg \circ u_x \circ C_x^{(1)}$	Betweenness centrality of nodes adjacent to x.
$Max \circ e_x \circ C_x^{(\infty)}$	Maximum of the structural equivalent of nodes reachable from x.
$Sum \circ e_x \circ (C_x^{(\infty)} \cap N_{C=c})$	Summation of the structural equivalent of nodes reachable from x and attribute C is c.

Results: Link Prediction

Table 6: Recall, precision, and F-value in the @cosme dataset as adding operators.

	Recall	Precision	F-value
graphic distance	0.1704	0.6687	0.2708
common neighbors	0.1704	0.6687	0.2708
Jaccard coefficient	0.1396	0.7031	0.2326
Adamic/Adar	0.1704	0.6686	0.2708
preferential attachment	0.5553	0.5779	0.5658
Method 1	0.5772	0.6333	0.5982
Method 2	**0.5687**	**0.6721**	**0.6130**

Results: Link Prediction

Table 7: Top 10 effective features in the @cosme dataset for link prediction (Method 1)

Feature	Description
$Max\{Avg \circ t \circ C_x^{(2)}, Avg \circ t \circ C_y^{(2)}\}$	Maximum of the average distance.
$Max\{Sum \circ s^{(1)} \circ C_x^{(1)}, Sum \circ s^{(1)} \circ C_y^{(1)}\}$	Maximum of the clustering coefficient.
$Min\{Sum \circ t_x \circ C_x^{(1)}, Sum \circ t_x \circ C_y^{(1)}\}$	Minimum of the number of adjacent nodes.
$Max\{Avg \circ s^{(1)} \circ C_x^{(2)}, Avg \circ s^{(1)} \circ C_x^{(2)}\}$	Minimum of the network density.
$Max\{Avg \circ u_x \circ C_x^{(2)}, Avg \circ u_x \circ C_y^{(2)}\}$	Maximum of the betweenness centrality.
$Min\{Avg \circ t \circ C_x^{(2)}, Avg \circ t \circ C_y^{(2)}\}$	Minimum of the average path length.
$Max\{Sum \circ u_x \circ C_x^{(2)}, Sum \circ u_x \circ C_y^{(2)}\}$	Maximum of the betweenness centrality.
$Max\{Sum \circ t_x \circ C_x^{(1)}, Sum \circ t_x \circ C_y^{(1)}\}$	Maximum of the number of adjacent nodes.
$Avg\{Sum \circ s^{(1)} \circ C_x^{(1)}, Sum \circ s^{(1)} \circ C_y^{(1)}\}$	Average of the clustering coefficient.
$Sum \circ u_x \circ C_x^{(2)} - Sum \circ u_x \circ C_y^{(2)}$	Difference of the betweenness centrality.

Discussion

- Consider a tradeoff: keeping operators simple and covering various indices.
- Other features cannot be composed in the current setting.
- Do not argue that the operators defined are optimal or better than any other set of operators.
- The number of features becomes huge when they increasingly add operators.

Conclusion

- Can generate features that are well studied in social network analysis, along with some useful new features, in a systematic fashion.
- Applied the proposed method to two datasets for link-based classification and link prediction tasks and thereby demonstrated that some features are useful for predicting user interactions.

Thank You!