The Link Prediction Problem for Social Network

Giulio Rossetti

KDDLab, ISTI-CNR, Pisa, Italy
giulio.rossetti@isti.cnr.it

WMR
9-05-2012, Pisa

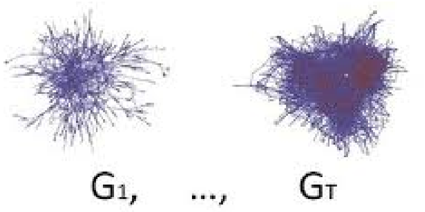
Index

Motivation & Problem Definition

- **Motivation:** Understanding how networks evolve

- **Problem definition**

  Given a snapshot of a network at time t, we seek to accurately predict the edges that will be added to the network during the interval $(t, t^{'})$



Applicative scenarios

- To suggest interactions or collaborations that haven't yet been utilized within an organization
- To monitor terrorist networks - to deduce possible interaction between terrorists (without direct evidence)
- Friendship prediction (Used in Facebook and Linked In)

Real Life Example

Esample: Co-authorship network for scientists

- Scientists who are “close” in the network will have common colleagues & circles $\rightarrow$ likely to collaborate
- Scientists who have never collaborated might in future $\rightarrow$ hard to predict

Goal: make that intuitive notion precise & understand which measures of “proximity” lead to accurate predictions

Methods for Link Prediction

- Take the input graph during a training period $[G_0 = (V, E)]$
- Pick a pair of nodes (u, v)
- Assign a connection weight $score(u, v)$
- Make a list in descending order of *score*
- Verify the prediction on the future graph $[G_1 = (V, E_{new})]$

score is a measure of *proximity*

Any ideas for measures?

Evaluate the results

Given a predictor p is there a way to decide if it is a "good" one?

We need to verify if p outperform the random predictor.

- **Random Predictor**: each edge have the same probability to appear in the future
- **Performance**: $performance(p) = \frac{TP}{TP+FP}$

$$ratio = \frac{performance(p)}{performance(p_{random})} = \frac{performance(p)}{\frac{|E_{new}|}{\frac{|V|*(|V|-1)}{2} - |E_{old}|}}$$

if $ratio > 1$ the predictor p is meaningful.

Comparing performances of different predictors

Which predictors give the beter performance over the same graph?

	p'	n'
p	TP	FN
n	FP	TN

Confusion Matrix

Usually we analyze either the performances ratio, ROC courves and Precision Recall courves.

ROC and PR courves

ROC and PR spaces are isomorphic.

Precision Vs. Recall :

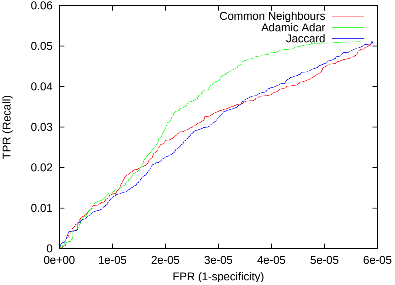
- Precision: $PPV = Performance = \frac{TP}{TP+FP}$
- Recall: $TPR = \frac{TP}{TP+FN}$

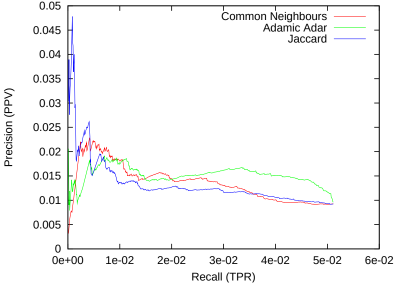
ROC (Receiver operating characteristic):

- 1-Specificity: $FPR = \frac{FP}{FP+TN}$
- Recall: $TPR = \frac{TP}{TP+FN}$

Another measure often used is AUC (area under curves).

ROC and PR courves



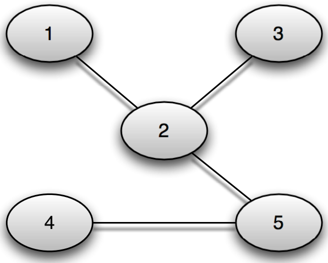


Classes of approaches

Link Prediction could be tackled in two different different ways:

- Unsupervised
- Supervised

Unsupervised Link Prediction



We want to define a set of standard proximity measures unrelated to the particular network

Unsupervised Link Prediction

Unsupervised measurements could rely on different structural property:

- **Neigborhood** measures
  - Common Neighbors, Adamic Adar, Jaccard, Preferential Attachment
- **Path-based** measures
  - Graph distance, Katz
- **Ranking**
  - Sim Rank, Hitting time, Page Rank

Neighborhood Measures

"How many friends we have to share in order to become friends?"

Common Neighbors: the more friends we share, the more likely that we will become friends

$$score(u, v) = |\Gamma(u) \cap \Gamma(v)|$$

Jaccard: the more similar our friends circles are, the more likely that we will become friends

$$score(u, v) = \frac{|\Gamma(u) \cap \Gamma(v)|}{|\Gamma(u) \cup \Gamma(v)|}$$

Neighborhood Measures

"How many friends we have to share in order to become friends?"

Adamic Adar: the more *selective* our mutual friends are, the more likely that we will become friends

$$score(u, v) = \sum_{z \in \Gamma(u) \cap \Gamma(v)} \frac{1}{\log(|\Gamma(z)|)}$$

Preferential Attachment: more friends we have, the more likely that we will become friends

$$score(u, v) = |\Gamma(u)| * |\Gamma(v)|$$

Path-based Measures

"How distant we are?"

Graph Distance: (negated) length of shortest path between u & v

Katz$_\beta$: weighted sum over all the paths between u & v

$$score(u,v) = \sum_{l=1}^{\infty} \beta^l \left| paths_{u,v}^{\langle l \rangle} \right|$$

where: $paths_{u,v}^{\langle l \rangle}$ ={paths of length exactly l from u to v}

SimRank

"Two nodes are similar to the extent that they are joined by similar neighbors"

$$similarity(u, v) = \gamma * \frac{\sum_{a \in \Gamma(u)} \sum_{n \in \Gamma(v)} similarity(a,b)}{|\Gamma(u)| * |\Gamma(v)|}$$

$$score(u, v) = similarity(u, v)$$

Results & Limits

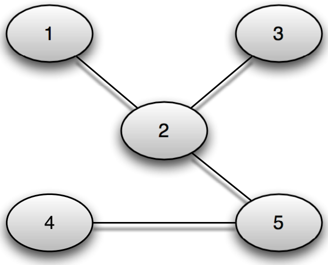
Results

- No single clear winner
- Many predictors outperform the random predictor ⇒ there is useful information in the network topology

Limits

- Different kinds of network are described by general closed formulae
- Adamic Adar & Katz (the best unsupervised predictors) have an overall performance between 10% and 16%.

Supervised Link Prediction



We want to extract knowledge from the network in order to make predictions

Supervised Link Prediction: Classification

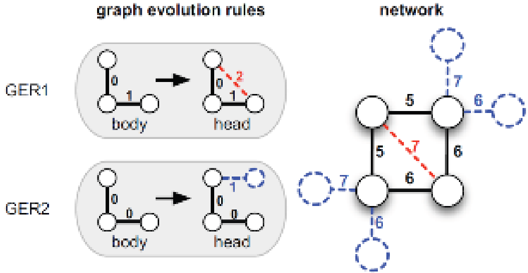
The process is now splitted in 2 parts:

1) Learning a model
2) Use the model for the prediction

The natural way: build a **Classificator** over a set of attributes.

Supervised Link Prediction: Evolutive Pattern

Evolution rules could be extracted from the network in order to predict recurrent pattern. Example: **GERM**



Results & Limits

Results

- Higer performances wrt the unsupervised approaches

Limits

- The two-phase predictive process is slower than the unsupervised ones.

Possible extensions

Several kinds of extensions of the seen models are possible:

- Temporal & evolutive analysis
- Link strength
- Multidimensionality
- Semantic enrichment (geographic information...)
- ...

Conclusions

Predict the evolution of a network is not an easy task because:

- Networks are containers of weak links
- False Positive issue
- Simple approaches are not so good
- Complex approaches have high costs