Learning Predictive Models for Link Formation

Joshua O’Madadhain\textsuperscript{1,2}, Padhraic Smyth\textsuperscript{2}, Lada Adamic\textsuperscript{1}

\textsuperscript{1}HP Labs, Palo Alto; \textsuperscript{2}Department of Computer Science, UC Irvine
is acquainted, but hasn’t worked with

person Bernardo doesn’t know

some spurious results
no clear community structure
CiteSeer (co-authorship)
some missing data
56K authors, 223K relationships
HP Labs email evolves over time
500 people, 20K relationships
Related Work

• investigating network growth models’ relation to link formation (Newman, 2001)
• using links to predict classes/attributes of entities (Getoor, Taskar, Koller, Provost, Jensen)
• predicting link types based on known entity classes (Taskar, Koller 2003)
• predicting links based on distances in projected Euclidean space (Hoff, Raftery, Handcock 2003)
• ranking potential links using a single graph-based feature (Liben-Nowell, Kleinberg 2004)
• predicting links between blogs based on common references (Adar, Adamic, Zhang, Lukose 2004)
Our Contribution

a general model for relationship strength
- learned from data
- easy to interpret
- allows combination of multiple features and measurements
- scalable to large networks
- can be augmented by other models
Relationship Strength Model

- basic model: edge presence, type, weight
  - how do you combine multiple edge weights or types?
  - direct links are generally only part of the story
  - “no link” may not mean “no relationship”

- more general model can incorporate indirect measures
  - distance measures
  - structural similarity
  - attribute similarity
  - etc.
Relationship Strength Model

- relationship strength \(\Leftrightarrow p(A \text{ is linked to } B \mid f_A, f_B, f_{A,B})\)
  
  \((f_X: \text{features of } X, f_{X,Y}: \text{features of relationship } (X,Y))\)
Potential Applications

- ranking associates by relationship strength
- network exploratory analysis (most surprising pairs)
- identifying missing/misplaced links (CORA/CiteSeer)
  - with data reliability model
- prediction of future relationships (HP email)
  - with communication/collaboration temporal pattern model
- recommending relationships (PeopleFinder²)
  - with relationship utility model
Building The Model

- measure correlation of feature values with link presence
- remove link (if any) before measuring features; avoids bias
- combine values with a conditional independence model (assumes features are independent given class):

\[
p(f_1, f_2, \ldots f_n|\text{link}) = \prod_i p(f_i|\text{link})
\]

\[\implies p(\text{link}|f_1, f_2, \ldots, f_n) \propto p(\text{link}) \prod_i p(f_i|\text{link})\]
Entity Attributes

- number of neighbors
- number of publications/contacts/emails/etc.
- individual characteristics
  - topic model
  - activities
- affiliations
  - corporation, university
  - co-publication in journal or conference
- demographic data
  - geographical location
Relational Features

• distance measure
  • shortest path, mean first passage time, relative importance
• structural similarity
  • Jaccard coefficient: \[ \frac{|N(A) \cap N(B)|}{|N(A) \cup N(B)|} \]
  • Adamic/Adar coefficient: \[ \sum_{C \in N(A) \cap N(B)} \frac{1}{\log |N(C)|} \]
  • neighborhood cosine: \[ \frac{w_{N(A)} \cdot w_{N(B)}}{\|w_{N(A)}\| \|w_{N(B)}\|} \]
• known relationships in other networks
• similarity of characteristics
  • distribution of interests (topics): KL-divergence
• shared affiliation
Experiments

- validation: measure model’s ability to correctly predict link presence/absence based on feature values
- Naive Bayes classifier
  - uses conditional independence model to define $p(\text{link} | \text{features})$ and $p(\neg\text{link} | \text{features})$
  - $p(\text{link} | \text{features}) > p(\neg\text{link} | \text{features}) \Rightarrow \text{“present”}$
  - can capture nonmonotonic feature dependencies
Experiments (2)

- training data: 250 linked, 250 non-linked pairs
- test data: 1000 pairs
- data point selection
  - training and test data randomly selected from same network
  - sparse graphs: neighborhood of an entity
    - neighborhood: proximal and/or similar entities
    - these are entity pairs which are most likely to be linked, so discrimination is more difficult and more interesting
  - dense graphs: any vertex pair

- networks represented/manipulated using JUNG (Java Universal Network/Graph) API (http://jung.sourceforge.net)
## Features Used

<table>
<thead>
<tr>
<th>Feature</th>
<th>CiteSeer, CORA</th>
<th>email</th>
<th>co-occurrence</th>
</tr>
</thead>
<tbody>
<tr>
<td>source degree</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>target degree</td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>topic distribution KL- divergence</td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Jaccard coefficient</td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Adamic/Adar coefficient</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>weighted cosine neighbor similarity</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>network distance</td>
<td>(x)</td>
<td></td>
<td>(x)</td>
</tr>
<tr>
<td>hierarchy distance</td>
<td></td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>
Evaluation: Feature Utility

- vertex degree generally not very informative
  - may be more useful as a basis for a prior
- topic model similarity: moderately informative
  - can be helpful when entity is isolated (no links)
- network distance: informative
  - but relatively expensive to calculate, esp. in denser graphs
- organizational hierarchy distance: informative
  - but generally not available
- structural similarity: informative
## Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORA (any vertex)</td>
<td>99.5 (99.9)</td>
<td>11.5</td>
<td>100 (100)</td>
</tr>
<tr>
<td>CORA (neighborhood)</td>
<td>81 (80)</td>
<td>46</td>
<td>77 (77)</td>
</tr>
<tr>
<td>CORA (neighborhood, with distance)</td>
<td>91 (80)</td>
<td>73</td>
<td>71 (71)</td>
</tr>
<tr>
<td>CiteSeer (neighborhood)</td>
<td>82 (72)</td>
<td>65</td>
<td>80 (80)</td>
</tr>
<tr>
<td>CiteSeer (neighborhood, with distance)</td>
<td>89 (72)</td>
<td>77</td>
<td>70 (70)</td>
</tr>
<tr>
<td>HP email (any vertex)</td>
<td>85 (88)</td>
<td>46</td>
<td>88 (88)</td>
</tr>
<tr>
<td>HP email (neighborhood)</td>
<td>80 (99)</td>
<td>79</td>
<td>76 (100)</td>
</tr>
<tr>
<td>HP co-occurrence (neighborhood)</td>
<td>84 (99.99)</td>
<td>67</td>
<td>80 (100)</td>
</tr>
<tr>
<td>HP co-occurrence (neighborhood, with distance)</td>
<td>89 (99.99)</td>
<td>72</td>
<td>80 (100)</td>
</tr>
</tbody>
</table>
CORA

CiteSeer

HP email

HP co-occurrence
PeopleFinder$^2$: an application of link inference

(Eytan Adar, Lada Adamic, Joshua O’Madadhain)

- **goal:** enable individuals within HP to
  - search for experts by topic
  - find knowledge communities
  - use social networks to connect to experts

- **approach:** construct social networks from name co-occurrence in large collection of HP intranet documents

- **application of link inference:**
  - plain co-occurrence data yields many spurious connections
  - train link inference algorithm on network
  - combine co-occurrence and inference (probability) scores with a fuzzy AND to yield relationship strength (and ranking)
Bernardo

Spurious results are filtered out
Community structure, including cross-collaborations, is more clear
Ongoing and Future Work

• individual-focused predictions
  • given network and incoming contacts for an individual, reconstruct outgoing contacts (and inverse: given outgoing, predict incoming)
• predicting network evolution over time
  • who will you work with in the future?
  • when will those relationships start and end?
• richer models
  • learn best feature combinations, feature weights
  • incorporate entity attention/capacity
  • additional features: spatial distances, affiliations, etc.
  • regression (predict amount of collaboration/contact)
  • joint probability of several links