Stochastic Link and Group Detection

Jeremy Kubica[*] **Andrew Moore** **Jeff Schneider** **Yiming Yang**
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
{jkubica,awm,schneide,yiming}@cs.cmu.edu

Abstract

Link detection and analysis has long been important in the social sciences and in the government intelligence community. A significant effort is focused on the structural and functional analysis of "known" networks. Similarly, the detection of individual links is important but is usually done with techniques that result in "known" links. More recently the internet and other sources have led to a flood of circumstantial data that provide probabilistic evidence of links. Co-occurrence in news articles and simultaneous travel to the same location are two examples.

We propose a probabilistic model of link generation based on membership in groups. The model considers both observed link evidence and demographic information about the entities. The parameters of the model are learned via a maximum likelihood search. In this paper we describe the model and then show several heuristics that make the search tractable. We test our model and optimization methods on synthetic data sets with a known ground truth and a database of news articles.

Introduction

Link detection and analysis has long been important in the social sciences (Wasserman and Faust 1994) and in the government intelligence community. Recently the internet and other sources have led to both a flood of circumstantial data and and increased interest in new link detection methods.

Consider a database that logs the international flights taken by many travelers. Most of the people on a particular flight are unrelated to each other. However, with a large amount of data we may observe that certain pairs of travelers fly together more often than would be expected by chance. Considering n-tuples (n>2) of travelers can yield even stronger evidence. Extending the idea further, we can hypothesize the existence of groups (or cells) where not all members of the group interact directly but the accumulated evidence of all their observed interactions clearly identifies the group's existence and membership.

Our system takes two types of input data: 1) a database of entities and their demographic information and 2) a database of link data. By searching for maximum likelihood parameters in our model, the system outputs a set of group memberships, which can then be used to answer queries such as:

1. List all the members of group G1.
2. List all the groups for which E1 and E2 are both members.
3. List a set of suspected aliases (entities that are in the same group(s), but never appear in the same link).

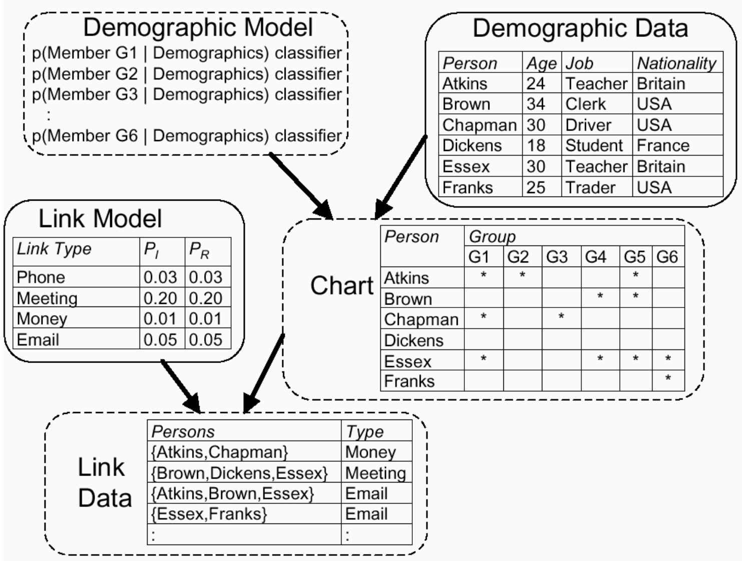
After discussing some related work, we describe our model and the methods we use to learn its parameters. We test our system on synthetic data since it allows us to compare with a known ground truth. We also show the output of running it on a large news article database to demonstrate its scalability. Finally we discuss extensions to the model and the additional queries we expect to be able to answer with them.

Related Work

Many of the social sciences use network analysis in their attempts to understand organizational structure and function (Wasserman and Faust 1994). A significant effort is focused on the structural and functional analysis of "known" networks. Similarly, the detection of individual links is important but is usually done with techniques that result in "known" links. Link analysis continues to expand into new domains including criminal intelligence (Sparrow 1991), large databases (Goldberg and Senator 1995), and the internet (Kautz *et al.* 1997). Our main distinction from these approaches is the probabilistic treatment of the data and the queries and the handling of n-ary links.

Recently the fields of computer science and statistics have also seen a significant interest in link and group detection. In (Cohn and Hofmann 2001) Cohn and Hoffman present a model for document and hypertext connectivity, where the document terms play a role similar to our demographic information. Their model explicitly assumes that documents are generated by a mixture of sources. In contrast, we assume that links are generated either randomly or by a mixture of a single group and noise. In (Taskar *et al.* 2001) Taskar et. al. propose a clustering approach based on the relational aspects of the data. Despite the additional power that may be yielded by incorporating the relational structure of the data, it is not immediately clear how to best adapt this

[*] Supported by the Fannie and John Hertz Foundation


Demographic Model
p(Member G1 | Demographics) classifier
p(Member G2 | Demographics) classifier
p(Member G3 | Demographics) classifier
:
p(Member G6 | Demographics) classifier
Demographic Data
Link Model
Chart
Link Data

Person	*Age*	*Job*	*Nationality*
Atkins	24	Teacher	Britain
Brown	34	Clerk	USA
Chapman	30	Driver	USA
Dickens	18	Student	France
Essex	30	Teacher	Britain
Franks	25	Trader	USA

Link Type	P_I	P_R
Phone	0.03	0.03
Meeting	0.20	0.20
Money	0.01	0.01
Email	0.05	0.05

Person	*Group*					
	G1	G2	G3	G4	G5	G6
Atkins	*	*			*	
Brown				*	*	
Chapman	*		*			
Dickens						
Essex	*			*	*	*
Franks						*

Persons	*Type*
{Atkins,Chapman}	Money
{Brown,Dickens,Essex}	Meeting
{Atkins,Brown,Essex}	Email
{Essex,Franks}	Email
:	:

Figure 1: Probabilistic model of group membership and link generation. Dashed borders indicated hidden information and solid borders indicate observed data

problem into a well-defined and powerful relational structure. In (Gibson *et al.* 1998) Gibson et. al. discuss the use of hyperlink information to find “web communities”. Their technique makes use of the inherent directionality of hyperlinks to define hubs and authorities. In contrast, we examine links without this directional property and thus do not make use of the concept of hubs and authorities. For example, the concept of a authority does not have a well defined meaning when talking about a list of people seen at lunch together.

While our approach is similar to those above, the main difference is in how we structure the generative model. We assume links a generated as noisy subsets of group members who have definite group membership. To this end, our model is designed to easily and directly capture the group membership nature of the data, including the fact that a person can be a member of many groups.

One approach we have elected to avoid is a mixture model in which each person is assumed to belong to a single hidden group that must be inferred from data. This model (which could be implemented as a simple mixture model) would allow evidence to associate a person probabilistically with any group (caused for example by a political party affiliation, for example). But as the evidence for that group membership increased it would by definition push down the probabilities for competing groups (e.g. a coffee-shop they may frequent). In the limit of infinite data everyone would each be a member of only one group. In contrast, the model presented here will allow simultaneous high confidence membership in many groups.

A Probabilistic Model of Group Membership and Link Generation

The goal of the algorithm is to find groupings of people given demographics and link data. Here, as in many clustering algorithms, the number of groups is given by the user and the groupings are then “discovered” so as to optimize some criteria. To this end, our proposed model is designed to capture a fairly diverse and noisy link generation process. Given this generative model, the algorithm attempts to find the groupings that maximize the probability of having seen the input. Figure 1 shows the model, which takes the form of a Bayesian network.

Our problem is simply stated: Given **evidence** in the form of *observed demographic data* and *observed link data*, find the most likely values of the three remaining data structures in the three remaining network nodes: the demographic model, the link model, and most importantly, the chart.

We will now proceed to describe each component of the network.

- **Demographic Data (DD).** DD contains all the people under consideration and their demographic information. The word “demographic” should not be interpreted too narrowly as it can include any information available about that person. In fact, neither the algorithm nor our software implementation assumes a pre-specification of its fields.

  DD is an observed node.

- **Demographic Model (DM).** DM is a model predicting group membership for each of N_G possible groups. Note that instead of one classifier predicting a N_G-valued output, we have N_G classifiers, the i'th of which predicts the True/False-valued output of whether the person is in Group i. This allows people to be in multiple groups. Our current model is very simple: we assume that group memberships are conditionally independent of each other given demographics. Further, our classifiers are also simple: they are merely naive Bayes classifiers. In later work both of the preceding assumptions may be relaxed if the statistics warrant.

  DM is a hidden node.

- **Chart (CH).** The Chart represents which people are in which groups. The generative model assumes that memberships of people in groups within the chart has been determined by, for each group g and each person p:

  1. Look up p's demographics in DD.
  2. Use DM to predict $P(p \in g \mid p\text{'s demographics})$.
  3. Randomly choose whether $p \in g$ according to this probability.

  We can thus easily define $log\, P(CH|DM, DD)$ as:

  $$log\, P(CH|DM, DD) = \sum_p \sum_G \begin{cases} log\, P(p \in G|\text{DM }, p\text{'s dems}) & p \in G \\ log\,(1 - P(p \in G|\text{DM }, p\text{'s dems})) & p \notin G \end{cases}$$

  CH is a hidden node.

- **Link Model (LM).** LM consists of various parameters (introduced when we define Link Data) used to determine the probabilities of various observed links. LM is a hidden node.

- **Link Data (LD).** The second piece of input, the link database (LD), is a set of records specifying n-tuples of entities that are linked by some event. Again, the word “link” should not be interpreted too narrowly as the event may be as circumstantial as their having simultaneously traveled to the same country.

  Further, our definition of a link is as an inherently noisy piece of information. We explicitly provide for two different types of noise. Specifically, links are generated assuming that it is possible to have both completely random links (Innocent Link Assumption) and group generated links that contain random non-group members (Innocent Link Member Assumption). We interchangeably use the terms “innocent” and “guilty” to mean “coincidental” and “indicative of a true linkage.” These two types of innocence are captured by the probabilities P_I and P_R, which represent the probability of an innocent link and the probability of a innocent (random) link member respectively. Another important aspect of the model is the incorporation of different link types. Each link type is assumed to have a different set of properties, the values P_I and P_R which are recorded in the Link Model (LM). For example, the link that captures people being in the same city can be expected to have significantly higher P_I then a link that captures a monetary transaction.

  $P(LD|LM, CH)$ is defined generatively by declaring that a link L between k people from a link type with parameters P_I and P_R is generated as:

  – With probability P_I: place k random, unique people into L.

  – Else with probability $(1 - P_I)$:

  ∗ Randomly choose a group G from the set of groups that have at least k people.

  ∗ While $|L| < k$

  · With probability P_R add a random person who is not a member of G, $p \notin G$ and $p \notin L$, to L.

  · Else with probability $(1-P_R)$ add a random member of G, $p \in G$ and $p \notin L$, to L.

  We make the simplifying assumption that the priors for each group generating a link are equal. This allows us to calculate $log\, P(LD|LM, CH)$ as:

$$\begin{aligned} &log\, P(LD|LM, CH) \\ &= \textstyle\sum_{LinkType} \sum_{L \in LinkType} log\, P(L|LM, CH) \end{aligned} \quad (1)$$

  where $P(L|LM, CH)$ is the probability of an individual link given the link model and the chart and, under the above generative assumptions, is:

$$P(L|LM, CH) = \left(\frac{P_I}{\binom{N_P}{|L|}} + \frac{1 - P_I}{N_G} \sum_G P(L|G, LM) \right)$$

  $P(L|G, LM)$ is the probability that link L was generated given that group G generated it and is defined as below in (2). Note in the equation below, K_1 is the number of people in the link that are members of group G and K_2 is the number of people in the link that are not members of G.

$$P(L|G, LM) = \left(\frac{(P_R)^{K_2} (1 - P_R)^{K_1} \binom{K_1+K_2}{K_1}}{\binom{|G|}{K_1} \binom{N-|G|}{K_2}} \right) \quad (2)$$

  One important possible approximation is shown in (3). This allows the concept of a group “owning” a link and can lead to significant computational speedups as discussed below. It makes the assumption that for each link L, there is usually overwhelming evidence of which group generated the link, or else overwhelming evidence that the link was generated randomly. Thus the approximation assumes that just one of the probabilities in 2 dominates all the others.

$$\begin{aligned} &log\, P(L|LM, CH) \\ &\approx logMAX \left(\tfrac{P_I}{\binom{N_P}{|L|}}, \tfrac{1-P_I}{N_G} MAX_G(P(L|G, LM)) \right) \end{aligned} \quad (3)$$

Fitting the hidden nodes

We wish to find

$$\begin{aligned} \underset{LM, CH, DM}{\text{argmax}} \quad & P(LM, CH, DM|DD, LD) = \\ \underset{LM, CH, DM}{\text{argmax}} \quad & \tfrac{P(LM, CH, DM, DD, LD)}{P(DD, LD)} = \\ \underset{LM, CH, DM}{\text{argmax}} \quad & P(LM, CH, DM, DD, LD) = \\ \underset{LM, CH, DM}{\text{argmax}} \quad & \begin{pmatrix} P(LD|LM, CH)P(LM)\times \\ P(CH|DM, DD)P(DM)P(DD) \end{pmatrix} \end{aligned}$$

where the second step is justified by noticing that P(DD,LD) is constant within the argmax. By noting that P(DD) is also constant, and by assuming a uniform prior over LM and DM, we see we need to find

$$\underset{LM, CH, DM}{\text{argmax}} \; P(LD|LM, CH)P(CH|DM, DD) =$$

$$\underset{CH}{\text{argmax}} \; (\underset{LM}{\max} \; P(LD|LM, CH))(\underset{DM}{\max} \; P(CH|DM, DD))$$

Thus the outer loop of our search is over charts, and for each new chart we quickly compute the value of LM that maximizes the likelihood of the LD the value of DM that maximizes the likelihood of CH.

At each iteration of optimization, one or more changes are made to the chart. The MLE DM is then calculated directly from the new chart and the DB—this is very fast for a naive Bayes LM. The LM can also be optimized using a simple EM update, although for the experiments described below the LM was held fixed. Finally, $log\, P(LM, CH, DM|DD, LD)$ is calculated. The difference between the two log-likelihoods represents the improvement, or lack there of, in making that change.

In future work we will evaluate the extent to which our current direct optimization approach could be accelerated by an EM algorithm.

Stochastic hill-climbing was used for optimization. In addition, the model described above was designed to be *optimization friendly*. For example, by allowing links generated by group g to probabilistically have some non-g participants, the optimization is able to fill in groups one person at a time and see incremental improvements in likelihood.

The optimization method used was a noisy hill climbing method. Specifically, at each iteration a bit (p, g) in the chart was flipped and the new score $log\, P(LM, CH, DM|DD, LD)$ was calculated. Moves resulting in improvement were always accepted and all other moves were accepted with some probability P_{worse}. It was found that looking at a number of neighbors before "committing" one, greatly improves performance. Looking at more neighbors per iteration allows the search to more thoroughly explore the immediate neighborhood. At the limit where the search examines all neighbors, the optimization becomes gradient descent.

It is also important to note that when using such an optimization method, the max approximation shown in (3) can lead to significant computational speedups. These speedups result from the fact that for a given (p, g, t) flip $\Delta log\, P(LD|LM, CH)$ can be approximated by looking at $log\, P(L|G, LM)$ for only a few groups instead of all of the groups. Specifically, for each link $\Delta log\, P(L|LM, CH)$ can be approximated as:

$$\Delta log\, P(L|LM, CH) \approx P(L|G_{MAX}, LM)_{OLD}$$
$$-MAX\left(P(L|G_{MAX}, LM), P(L|g, LM), \frac{P_I}{\binom{N_P}{|L|}}\right)$$

where G_{MAX} was the most probable group to have generated link L before the flip and $P(L|G_{MAX}, LM)_{OLD}$ was its score before the flip. In other words, the change in the probability of a link can be approximated by the three cases: the former G_{MAX} still "owns" the link, group g now "owns" the link, or it is more probable that the link is now "innocent". Thus, approximating $\Delta log\, P(LD|LM, CH)$ for multiple neighbors can be done in time independent of the number of groups, N_G. This independence can lead to an $O(N_G)$ speedup for examining neighbors. Additional speedups can also be gained by caching G_{MAX} and $P(L|G_{MAX}, LM)_{OLD}$ for each link.

Empirical results

Tubeworld

Initial tests were performed on a simulated world called Tubeworld. Tubeworld, shown in Figure 2, consists of a finite two dimensional world where people occupy randomly generated points. Each group is defined by a parallelogram and consists of all people whose location falls within this parallelogram. Groups can overlap and are assumed to contain at least two members. The small rectangles in Figure 2 represent the people and are color-coded according to the groups to which a person belongs. The real-valued (x,y)-coordinates of the people are hidden from the algorithm. Demographic information consists of breaking the X and Y world coordinates into D discrete blocks, giving a total of D^2 possible demographic labelings. Finally, the N_L links are generated based on the groups and according the the generating assumption given above.

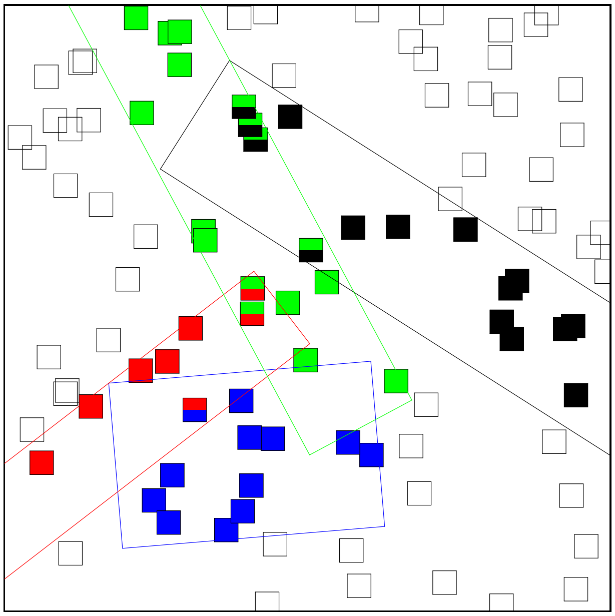


Figure 2: Example tubeworld with 100 people and 4 groups.

Tubeworld Results

The tests consisted of 50 randomly generated tubeworlds, each containing 100 people, 4 groups, and 1000 links. Runs using the Innocent Link Assumption (ILA) and/or Innocent Link Member Assumption (ILMA) used the values $P_I = 0.1$ and $P_R = 0.1$. The charts produced are the result of set number of optimization iterations.

In order to compare the performance of the algorithm using various combinations of the ILA and the ILMA, a 4 by 4 test was used. For each tubeworld the linkbase was generated using each possible combination of ILA and ILMA, namely: ILA + ILMA, ILA, ILMA, none. In addition, for each such linkbase the chart was learned using each possible combination of ILA and ILMA. For each generating/learned pair, error in the learned chart was calculated as the difference in the log-likelihood of the learned chart and the log-likelihood of the generating chart. The average errors are shown in Table 1. The columns represent the different learning methods and the rows represent the different generation methods. Note that larger, less negative, numbers indicate better performance.

In addition, we examined a second performance metric, a paired membership error test. This test looked at all possible pairs of people and all groups. The error between two charts for each pair of people, p_1 and p_2, is defined as the absolute value of the difference between the number groups in each chart that contain both p_1 and p_2. The total error between two charts is the sum of errors for each pair of people. Thus, if two charts are identical this error would be zero, because each pair of people would belong to the same number of groups in both charts. This test has the advantages of

True World:	Model assumes ILA/ILMA	Model assumes ILA	Model assumes ILMA	Model assumes neither
ILA/ILMA	-89.0	-543.7	-159.2	-3134.0
ILA	-84.5	-338.9	-186.1	-3194.6
ILMA	-36.7	-602.9	-101.7	-3271.6
-	-169.6	-336.2	-306.1	-1505.4

Table 1: Log-likelihood error rates for charts learned. The rows and columns represent the probability assumptions for generation and learning respectively.

being simple, computationally cheap, and providing a good measure of the difference between two charts. The average error for each of the 16 probability assumptions are given in Table 2 below.

True World:	Model assumes ILA/ILMA	Model assumes ILA	Model assumes ILMA	Model assumes neither
ILA/ILMA	316.5	2102.3	930.1	7700.0
ILA	132.5	877.2	626.7	4940.1
ILMA	91.4	1576.5	165.6	5836.1
-	34.1	160.3	47.8	813.0

Table 2: Paired person error rates for charts learned. The rows and columns represent the probability assumptions for generation and learning respectively.

While the results of the 16x16 pairwise significance tests are omitted for space considerations, we do examine the significance in the success of the learners using the ILMA/ILA versus other learners. Specificaly, we examine the statistical significance of the difference in mean paired person error rates between column 1 and the other columns for a given row. With two exceptions, the differences in the mean paired person errors between the models learned using the ILMA/ILA and all other assumptions were statistically significant, using $\alpha = 0.05$. The two exceptions were that when the True World was "ILMA" or "Neither" then there is no significant difference between the scores for "Learner ILMA" and Learner "ILMA/ILA".

From Table 1 and Table 2 we can see the relative performance of the different learners on linkbases generated by different methods. On average, learners using both the ILMA and ILA performed better than the other learners. This supports the belief that incorporating both forms of innocence allows the model to represent a wider range of underlying generation models while not harming performance on simpler models.

The results reveal that assuming only guilty links but innocent link members leads to superior performance over assuming innocent links and no guilty link members. One reason for this might be the fact that under the ILMA if link sizes are small, random links may still be attributed to a group to which one of the link members belongs. Thus, although the link is innocent, the ILMA can account for it as a guilty link with a few innocent members. This may be important in cases where a search is adding members to a group. Another important trend is the poor performance of the learners that do not assume either type of innocence. Since they do not make either assumption, the link data is assumed to be noiseless. This assumption is inherently flawed when working with real world data.

It is also interesting to note the extent to which the *optimization friendly* assumption of the previous section was important. On average the learners using both the ILMA and ILA outperformed the other learners regardless to how the linkbases were generated. One example of this benefit is as follows. A pair of people might be in a group together, but initially be placed in the wrong group. Assuming they have a large number of links in common, removing one of them may result in a drastic worsening in score. Using the ILMA and the ILA might reduce this problem, because some of these links can temporarily be accounted for by innocent interactions until both people are moved.

Finally, it is interesting to note that performance of all learners on charts generated using ILMA and ILA is worse than on charts generated with other assumptions. This suggests an increased difficulty of these problems arising from the larger amount of noise in the linkbase.

News article experiments

In order to test scalability and performance on real data, we ran our algorithms on a database of news articles. The data is part of a benchmark evaluation collection named Reuters Corpus Volume I and contains over 800,000 news stories from Reuters between August 1996 and August 1997. The corpus has been used by the 2001 Text Retrieval Conference (TREC-10) for the evaluation of document filtering methods. For our tests we selected a subset of approximately 35,000 articles.

Automated extraction of named entities, such as *Person*, *Organization*, *Location*, *Date* and *Time*, has been successfully applied to many information extraction problems since its initial success in the Message Understanding Conferences (MUC) (Borthwick *et al.*). A Hidden Markov Model (HMM) approach by BBN is one of the most successful methods, which obtained a performance of 96% on the F_1 measure (the harmonic average of recall and precision) on English documents in a MUC evaluation(Bikel *et al.* 1997). We applied a pre-trained BBN HMM model to automatically extract person's proper names from the articles.

We then treated each article as a link between all of the people mentioned in the article. We preprocessed the results by excluding all articles that referred to less than two people. Following that, we also eliminated any people that were not referred to by any articles with at least two people in them. The final result was a database of 9000 entities and a set of 9913 (2-ary or higher) links relating them to each other.

Sample results. After some manual experimentation we found 30 to be a good number of groups for this data set. It turned out that most of the 9000 entities were not mentioned frequently enough to merit their inclusion in groups. After about two hours of optimization on a 1 Gigahertz Pentium, some examples of the groups found are:

```
  G2 (john major,dick spring,ernesto
zedillo,zedillo,richard alston,sabah, abdus samad
azad,stella mapenzauswa,finmin,mccurry,viktor
klima,ron woodward, alexander smith,iss price,glenn
somerville,yevgeny primakov,washington, joan
gralla,bernie fraser,stahl,danka,sally,palladium,van
der biest,fausto)
```

```
  G22 (clinton,blair,tom brown,ernesto
zedillo,leon,neumann,h.d.  deve gowda, rob davies,karmen
korbun,fran,consob,saharan blend,englander,garcia,
bruce dennis,jonathan lynn,laurence lau,h.  carl mc-
call,fraser, anne vleminckx,delphis,collin co,elaine
hardcastle,alain van der biest, david martin)
```

These groups illustrate some successes and remaining issues. The name Washington appears in group G2, but in fact was misidentified as a person's name by the named entity extraction software. Similarly, ISS price is a technical label used in quoting various financial instruments. A larger problem is the frequent occurence of single name entities. This happens when the writer uses only a first or last name to refer to a person and effectively results in numerous aliases appearing in the database. In some cases the connection is found. For example, Ernesto Zedillo appears with Zedillo in G2. However, the match is not made in G22 and Bill Clinton is also not included in G22 with Clinton.

Despite these difficulties, several interesting patterns appear in the groups. G22 contains the leaders of the US, the UK, and Mexico. It also illustrates an unintended result. Tom Brown and Jonathan Lynn are writers. Many of the groups ended up consisting of a combination of various writers and the subjects they most often wrote about. Writers ended up being linked by writing about common subjects and subjects ended up being linked even when not appearing in the same article, by being written about by the same person.

Detecting aliases. As already mentioned, the intentional or unintentional use of aliases presents a serious problem for link detection algorithms. As a first step toward identifying aliases we consider a specific type: single user, non-co-occurring aliases. These are aliases used by only one individual and have the property that they never appear together in the same link. This type of alias is a poor model for what happens in news articles, but may be a very good model for the use of fake passports for example. Provided the fake passport is not shared with others, you do not expect to see the real and the fake passport to be used for entry to the same country in a short period of time or for them both to be used to board the same commercial flight.

We propose a simple algorithm to detect these types of aliases. We search for pairs of individuals that are placed in the same group, but never appear together in any link. We rank these hypothesized aliases according to the size of the group in which they appear. In general, membership in smaller groups is a stronger indication of a link between two entities.

Ideally, we could test our algorithm by identifying some of the aliases already existing in the news article data and checking if our algorithm can find them. Since the only method we know of doing this is the manual identification of the aliases we chose an alternative test. We automatically generated aliases for a random set of 450 of the 9000 entities in the database. In each case, we went through the link data referring to each of those 450. For each link, with probability 50%, we substituted the name of the entity with its alias. In this case, we were rarely able to detect the aliases. The problem is that most of the 9000 entities appear in very few links. Taking the small number of links and cutting them in half (by relabeling half of them to be the newly created alias) made it difficult to even get these entities into appropriate groups, much less identify which were aliases for each other. Unfortunately, this is exactly the real problem encountered when doing link detection in the face of aliases.

To simplify the task, we selected only the entities with at least 10 links in the database (there are 273) and made aliases for them. The following table shows the results of the alias detection algorithm:

```
True aliases:           273
Hypothesized aliases: 57849
Group rank  False Negatives  False Pos
     0            272              76
     1            272             190
     2            271             628
     3            269            2579
     4            268            5318
     5            263           12536
     6            263           21251
     7            260           30741
     8            258           44141
     9            256           57832
```

The i'th row specifies the number of false positives and false negatives found when considering the aliases found in only the groups which are smaller than the i'th group (ranked according to size). Using all of the groups we see that out of 273 true aliases, only 17 were found and 57832 false discoveries were made. The high number of false positives is not necessarily bad. Some of the original names in the news articles really are aliases as we observed earlier. Also, the whole purpose of the group model is to identify members of the same group even though there may be no direct link evidence between them. Many of the false positives are just the result of the algorithm doing its job. Finally we observe that a random selection of 57832 pairs would not expect to find any of the 273 aliases by chance. Usually an alias detection algorithm would be used to generate hypothetical aliases that would be checked by other means.

Research Area Groupings from Web Pages

A third source of test data came from the Carnegie Mellon University Robotics' Institute webpages. Specifically, we looked at the groupings of people within the Robotics' Institute based purely on their publicly declared research interests. Each person with an official webpage was treated as an entity. Each link was defined by a single research interest, such as machine learning, and included all people who listed that interest on their official page. Note that in this case no demographic information was used, but one could consider using such information as whether a person is a faculty member or a student.

The algorithm was run with 8 groups. We expected to find groupings that roughly matched project groups and lab groups. The results returned were largely consistent with people's declared interests, but were often noisy combinations of several related lab groups and people with similar interests. This is most likely due to the fact that the Robotics' Institute contains significantly more than 8 lab groups and a significant number of members of the Robotics' Institute did not declare any research interests.

A more illuminating example reverses the above roles of people and research interests. In this case each research interest is treated as a separate entity. A link is then defined as all of the entities that appear together under a single person's research interests. Again, no demographic information was used. The algorithm was then run with 5 groups and found results that agree with intuition. For example, two of the groups were:

`G0 (actuators, control, field robotics, legged locomotion, manipulation, mechanisms, mechatronics, mobile robots, motion planning, multi-agent systems, space robotics)`

`G4 (animation, graphics, computer vision, visualization, geometric modeling, human-computer interaction, image compression, image processing, machine learning, object recognition, pattern recognition, sensor fusion, stereo vision, video systems, visual tracking)`

Note that both of these groups agree with intuition for a grouping of research areas within the field of robotics. It is also important to note that many of the items in the groups, while intuitively similar, did not appear together on a page. In other words, while some people's webpages contain many consistent interests such as "3-D perception, computer vision, mobile robots, and range data", a large number of people's interests contained diverse topics such as "computer vision, entertainment robotics, machine learning, mobile robots, and obstacle avoidance" or only a incomplete list of what be considered related interests, such as "machine learning" without "artificial intelligence".

Discussion

The algorithm described in this paper is only the core of the link detection and analysis system currently under development. We are working on the following extensions:

Posterior probability distribution of charts. Ultimately, the discovery of the maximum likelihood instantiation of model parameters (even if it could be found), may not be the most useful result. A more desirable alternative is a posterior distribution of parameter instantiations. We will use Markov Chain Monte Carlo (MCMC) methods to generate distributions from which we will be able to answer the following additional queries:

1. What is the probability that entity E1 is a member of group G1? This is answered by counting the frequency of samples for which E1 is in G1.
2. What is the probability that E1 and E2 are in the same group? Again, simple counting is used.

Dynamic group membership. In reality, we expect entities' memberships in groups to evolve rather than being static across all time covered by the link data. A straightforward extension to the chart allows it to represent each entity's membership in groups at each discrete time step. The model is extended such that membership at time t depends probabilistically on the demographic model, the demographic data, and the membership at time $t-1$. The result is significantly more parameters in the model, thus making the optimization more difficult. By solving the computational challenge we hope to obtain more reliable answers to the queries and their time-based analogs.

Conclusion

We have proposed a generative model for group membership and link generation. The strength of this model is its ability to process probabilistic link data and reason probabilistically about links and group memberships. This approach simultaneously provides the ability to incorporate information about n-ary ($n > 2$) links.

We have developed a search method to optimize the parameters of that model given observational data. Our experimental results show the ability of the optimization to identify groups and links, as well as generating alias hypotheses. The experimental results and the computation required to generate them show that the performance of the system is still constrained by the ability of the optimizer to find the best parameter settings and our future work will focus on scalability. Finally, we have described our plans for a complete system capable of answering a broad array of probabilistic queries about links, membership in groups, and aliases.

References

D.M. Bikel, S. Miller, R. Schwartz, and R. Weischedel. Nymble: a high-performance learning named-finder. In *In Fifth Conference on Applied Natural Language Processing*, 1997.

A. Borthwick, J. Sterling, E. Agichtein, and R.Grishman. Description of the mene named entity system as used in muc-7. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, Fairfx, Virginia.

David Cohn and Thomas Hofmann. The missing link - a probabilistic model of document content and hypertext connectivity. In *Neural Information Processing Systems 13*, 2001.

D. Gibson, J. Kleinberg, and P. Raghavan. Inferring web communities from link topology. In *Proc. 9th ACM Conference on Hypertext and Hypermedia*, 1998.

H. Goldberg and T. Senator. Restructuring databases for knowledge discovery by consolidation and link formation. In *First International Conference on Knowledge Discovery and Data Mining*, 1995.

H. Kautz, B. Selman, and M. Shah. The hidden web. *AI Magazine*, 1997.

M. Sparrow. The application of network analysis to crminal intelligence: an assessment of prospects. *Social Networks*, 13, 1991.

B. Taskar, E. Segal, and D. Koller. Probabilistic clustering in relational data. In *Seventeenth International Joint Conference on Artificial Intelligence*, pages 870–876, Seattle, Washington, August 2001.

S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.