Finding Tribes: Identifying Close-Knit Individuals from Employment Patterns

Lisa Friedland
lfriedl@cs.umass.edu

David Jensen
jensen@cs.umass.edu

Department of Computer Science, University of Massachusetts Amherst
140 Governors Drive, Amherst, MA 01003-9264

ABSTRACT

We present a family of algorithms to uncover *tribes*—groups of individuals who share unusual sequences of affiliations. While much work inferring community structure describes large-scale trends, we instead search for small groups of tightly linked individuals who behave anomalously with respect to those trends. We apply the algorithms to a large temporal and relational data set consisting of millions of employment records from the National Association of Securities Dealers. The resulting tribes contain individuals at higher risk for fraud, are homogenous with respect to risk scores, and are geographically mobile, all at significant levels compared to random or to other sets of individuals who share affiliations.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications – *Data mining*; I.5.1 [**Pattern Recognition**]: Models – *Statistical*; J.4 [**Social and Behavioral Sciences**].

General Terms

Algorithms, Performance, Design.

Keywords

Social networks, dynamic networks, anomaly detection.

1. INTRODUCTION

In relational and social network data sets, social structure among individuals offers vital explanatory power for prediction tasks. Achieving a clearer view of the connections between entities, particularly in dynamic temporal domains, promises to aid analyses of the data. This research seeks to infer close relationships among certain co-workers, given a database of affiliation histories. Specifically, we search for groups of individuals, which we call *tribes*, that have anomalously similar job sequences within a large industry. We want to identify employees who were co-workers at multiple jobs, and to distinguish those who worked together intentionally from those who simply shared frequently occurring employment patterns in the industry.

Relational knowledge discovery [11] exploits connections among individuals, as well as intrinsic attributes, to identify patterns and make predictions. In relational, or network-structured, data sets, linked entities often display underlying dependencies such as autocorrelation, or homophily: the tendency of connected entities to have similar attribute values [18]. When the links are specified in source data, they can be used to infer large-scale structure, for instance at the level of groups or communities [17], [12], [8]. However, in other cases, the links themselves must first be inferred, whether by preprocessing to extract real-world entities [9], or by modeling their interactions. In this work, we identify fine-grained, strong associations among individuals in a large data set by finding small groups that are anomalously similar.

This novel task was inspired by a case study, but it can be applied to a number of domains. The important properties in the scenario are that "individuals," or one type of entity, are affiliated with "organizations," another type of entity, and that the affiliations change over time. We form a model of "typical" sequences of affiliations, which allows us to score any given sequence of affiliations based on its likelihood. Then, for each pair of individuals, we find the sequence they have in common (if any) and score it. The score correlates with the likelihood that two individuals shared the given affiliations by chance alone, under the null hypothesis of independent movement.

Other tasks with this structure include finding students that select classes together, given a table of students and their enrollments; inferring sets of cars traveling in caravan on a highway, given sightings at different locations and times [3]; and discovering family structure in animal groups, from tagged individuals frequently sighted together [4] (see Related Work). If we remove the temporal aspect of the problem and simply require a bipartite graph of affiliations, then a generalized version of the model could identify people with unusually similar tastes in movies, highly related documents sharing words that rarely co-occur, or friends within an album or yearbook containing photos of large groups.

Our model is particularly suited to situations involving large organizations, where the original data does not describe detailed associations among individuals. For instance, in our employment domain from the securities industry, thousands of people often share a loose relation of working at the same branch. In such cases, we can benefit from learning a model of typical affiliation patterns. Then, against this background, small groups doing unusual things stand out in contrast.

2. MOTIVATION

The National Association of Securities Dealers (NASD) regulates securities firms in the United States, with responsibility for preventing and discovering misconduct among its registered representatives, also called "reps." With over 600,000 reps under

© ACM, 2007. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, http://doi.acm.org/10.1145/1281192.1281226.

its jurisdiction, NASD must focus its investigatory resources on those reps most likely to commit fraud or other violations of securities regulations. In conversations over the course of related projects [7], [19], NASD representatives suggested that fraud is sometimes committed by colluding groups of reps that pass together through multiple places of employment. If we could identify "tribes" of reps moving together from job to job, we could test them for elevated rates of one or more indicators of fraud risk. Of course, finding tribes will certainly also identify harmless sets of friends that worked together in the industry, perhaps recruiting one another to new jobs. Our hope is that we will discover groups in which the reps tend to be homogenous with respect to fraud risk: mostly low-risk or mostly high-risk. Such tribes could then serve as starting points for detecting new fraud rings.

Our source data is a table of employment histories: for each rep, a series of records contains the branch identifier, start date, and end date for every employment the rep has held in the securities industry. The data set is large, containing (after some preliminary cleaning) 4.8 million records describing employments of 2.5 million reps at 560,000 branch offices. The branches range in size from 1 to 35,000 employees. The branch identities themselves have been inferred, through an earlier process of link consolidation from office addresses [7], from the 22,000 firms that have ever registered with NASD. The employment histories span the twentieth century through today, though most records are from the past fifteen years.

Two features of the real-world data shape our approach. First, many employment histories include simultaneous, overlapping jobs or leave gaps between employments. This muddies the concept of a transition between jobs: a rep does not necessarily leave one job when starting another, nor vice versa. Overlapping jobs are too common to consider discarding from the data: 20% of employees hold more than one job at some point, and 10% even begin multiple jobs (up to 16) on the same day. With transition dates ill defined, we cannot formulate this task as a search for employees changing jobs within a common interval of time. Therefore, we direct our attention to the times and places that people have been co-workers, as opposed to the boundaries between them.

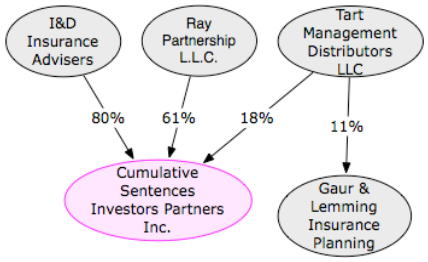

Figure 1. Example (hypothetical) of branch-branch transition patterns. The left-most edge indicates that 80% of the reps who ever worked at I&D Insurance were later employed by Cumulative Sentences. Only edges with high percentages are shown.

The second key feature is that mass movements of employees between jobs are common. In addition to continual flows between firms (e.g., common career paths within a given city), the businesses change: branches are closed or opened; firms merge or are acquired. Reps in these flows could end up being colleagues at multiple organizations without even knowing each other. We can visualize such trends as transition diagrams, as in Figure 1, to create a map of the whole industry. The meaning of the numbers along the edges will be discussed and refined in Section 4.2; roughly speaking, they indicate the percentage of employees at one branch that later work at the linked (destination) branch.

Many of these transition percentages are high, which establishes that job movement in the industry is not random. For instance, among branches of fewer than ten employees, about 73% have some destination where at least 90% of the employees later end up. Among larger branches, 30% of the branches have some destination where at least 50% of their employees go. These figures increase slightly if we ask which transitions are common within a given year—to spotlight abrupt shifts like mergers—as opposed to throughout the life of a branch office. This structured transition pattern is what we hope to factor out in order to find genuinely tight associations among individuals.

3. TASK DESCRIPTION AND APPROACH

3.1 Formulation

In the most general setting, we define this task to be the identification of anomalously related entities. As input, we require a bipartite graph $G = (R \cup A, E)$ of entities $R = \{r_1, r_2, \ldots, r_n\}$ and attributes (in this case, organizations) $A = \{a_1, a_2, \ldots, a_m\}$. The entities should connect to at least several attributes, on average, so at not to be too simply characterized. Each attribute should attach to a large number of entities, enough so that the behavior of this set of entities can be modeled. The current formulation requires that an entity's attributes be sequentially ordered (e.g., chronologically), while a more general extension would consider an unordered set.

The groups of entities we wish to return are those sharing unusual combinations of attributes. Our strategy for this task revolves around developing a good definition of "unusual." For an entity group to be considered anomalous, the shared attributes themselves need not be unusual, but the particular configuration of them should be; entity sets that are alike in typical ways are not part of our target. Our scores are similar in spirit to tf-idf weights in that they emphasize unusual shared attributes [20]; however, our method for estimating joint likelihoods is unique. In this paper, we approximate a joint distribution over sets of attributes by modeling the co-occurrence rate (or transition rate) of each pair of attributes. Then, the likelihood of an attribute set can be computed as a function of the pairwise co-occurrence rates.

3.2 Basic Tribe-Finding Process

We are given a bipartite graph $G = (R \cup A, E)$ of reps and organizations. In the NASD data, each edge $e \in E$ is annotated with a time interval: $e = (r_i, a_j, tstart_{ij}, tend_{ij})$. The general tribe-discovery process, assuming we are given such time intervals, is summarized in Figure 2. For our application, it begins with listing all pairs $f_{ij} = (r_i, r_j)$ of individuals that have ever worked together. This can be a large list (2.6 billion pairs, in our case), generated by iterating through the branches and recording every pair of reps $f_{ij} = (r_i, r_j)$ whose employment stints at a branch intersect.

Find-Tribes($G = (R \cup A, E)$, *timeAnnotations*)

```
1  Determine-Candidate-Pairs(G = (R ∪ A, E),
     timeAnnotations):
   1   F = null, pairAnnots = null
   2   FOR each org of A
   3       Get all reps associated with org
   4       FOR each pair (r_i, r_j) of reps
   5           IF timeInterval(r_i, org) and
                 timeInterval(r_j, org) overlap THEN
   6               F = F ∪ (r_i, r_j)
   7               pairAnnots[r_i, r_j] appends org
                     and times
   8   return (graph H = (R, F), pairAnnots)
2  Score-Candidate-Pairs(F, modelParameters)
3  Recover-Tribes(H, d):
   1   FOR each f_ij in F
   2       IF score(f_ij) < d THEN delete f_ij
   3   return (Tribes = connectedComponents(H))
```

Figure 2. Tribes algorithm.

For each pair, we then summarize their co-worker relationships, keeping track of the jobs where they coincide. We note additional information, such as the date the reps first coincide at each job, and the total time spent at overlapping jobs. The algorithm stores the pairs in a new graph $H = (R, F)$, where $F = \{f_{ij}\}$, and each edge is annotated with:

$pairAnnots_{ij}$ = { the sequence of jobs $\{a_x, a_y, \ldots\}$ shared by r_i and r_j $\cup$ additional information described above}.

For purposes of efficiency, we retained only the rep pairs that had at least three jobs in common. This left us a graph $H' = (R, F')$, with $|R| = 2.5$ million, and $|F'|$ = approximately 3 million pairs of individuals that are co-workers multiple times: the candidates for tribes.

The algorithm proceeds by identifying all significant pairs. We compute a score $c_{ij}(pairAnnots_{ij})$ for each edge in F', measuring how significant or unusual its sequence of shared jobs is. Section 4.1, which follows, discusses the choice of function to use for c_{ij}.

Once the significance scores are computed, we pick a threshold d for the scores and remove all edges f_{ij} for which $c_{ij} < d$. Then, we compute the connected components of H', which are designated the tribes. The output of the algorithm is a list of tribes: sets of reps within components of size two or higher in H'.

Computationally, Step 1 of Figure 2 is the most expensive. If the maximum degree of a branch is k, then we must consider $O(|A|k^2)$ potential pairs and store information about each pair. Once we have created the graph H and pared it to a smaller H', the remaining steps are in $O(|F'|)$. Estimating the model parameters will generally require $O(|E|)$, one pass through the source data.

4. SCORING/RANKING FUNCTIONS

The choice of scoring methods constitutes the heart of the task. (We also use the term "ranking method," since we only use the scores to rank the pairs.) We propose and compare several.

4.1 Simple Measures

Given a sequence of jobs, we must decide whether it is unusual for a pair of co-workers to have worked together at all of these jobs. Two straightforward methods for ranking the pairs are:

- JOBS = the number of jobs in the shared sequence
- YEARS = the number of years of overlap

Computing JOBS is a straightforward count of the job sequence. For years, we choose to add up the length of each overlap period, so that if a pair of reps works simultaneously at two branches for ten years, this counts as twenty years of overlap.

These simple methods treat all branches equivalently. As described earlier, however, reps in the securities industry do not behave as if they are picking jobs out of a hat. Instead, they tend to follow patterns caused by industry events and influenced by geographical and other factors. Accounting for these patterns motivates the probabilistic models that follow.

4.2 Probabilistic Model

In developing a simplified model for the job history data, there is a tradeoff between flexibility and performance. We want the model to flexibly mimic the characteristics of each branch without exactly reproducing the original data. In addition, the procedure must be tractable on a large data set. The process of computing all pairs of co-workers is time- and space-intensive, so it would be infeasible, for example, to generate random replicates of the network and re-compute shared job sequences. Attempting to strike the right balance, we model rep movement across branches as a modification of a Markov chain over organizations, ignoring timing and duration.

If each rep held one job at a time, and changed it at each time step, we could model movement using an ordinary Markov chain, as follows: Each rep picks a start branch randomly. Then at each step, the rep's next branch is decided probabilistically based only on the current branch. We ignore actual time spent at each job; at each step in the Markov process, a rep either moves to a new branch, or leaves the workplace. We also assume that transition probabilities are static over time and that each rep chooses jobs independently. Using this model, we could estimate the probability of a rep having a job sequence such as in Figure 3a as

$$x = P(\text{Branch A} \rightarrow \text{Branch B} \rightarrow \text{Branch C} \rightarrow \text{Branch D}) = p_A \cdot t_{AB} \cdot t_{BC} \cdot t_{CD}.$$

The quantities to estimate are

$p_i = P$(start at Branch i)

= (# reps ever at Branch i) / (# reps in database)

$t_{ij} = P$(transition from Branch i to Branch j | [given that] currently at Branch i)

= (# reps who leave Branch i and next go to Branch j) / (# reps ever at Branch i).

Under the null hypothesis of independent movement, if P(Rep 1 holds this sequence of jobs) = x, then P(Reps 1 and 2 each hold this sequence of jobs) = x^2. Since ranking by x is equivalent to ranking by x^2, it is enough to calculate x. Similarly, it is not necessary to compute the denominator of p_i.

If job sequences in the database were as simple as Figure 3a, this model would be sufficient. However, Figure 3b is more typical of the data. The reps in this example start at the same branches, split apart for a few years, come back together, and then both begin two jobs (Branches C and D) at the same time. To allow for these more complex situations, we adjust the model such that it is no longer a Markov chain, while keeping the probability calculations almost the same.

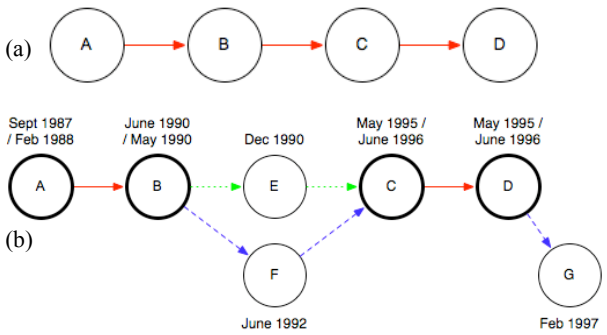

Figure 3. Job sequences to score. Nodes represent branches. In (b), we display differing trajectories for two reps, but only the shared jobs (in bold) are used for scoring. Dates describe start dates for each rep at each job.

The first modification is to allow reps to have different paths between shared jobs, such as from Branch B to Branch C in Figure 3b. To do this, we replace the quantity t_{ij} with a new quantity v_{ij}:

v_{ij} = P(move to Branch j *at any point* after Branch i | currently at Branch i)
= (# reps who go to Branch j *at any point* after working at Branch i) / (# reps ever at Branch i).

Now, each $v_{ij} \geq t_{ij}$, and the transition probabilities leaving a branch no longer sum to 1 ($\sum_i t_{ij} = 1$, but $\sum_i v_{ij} \geq 1$). We cannot generate sequences as part of a Markov process using the v_{ij} probabilities, but we can still score an existing sequence of jobs using these estimates of how likely each transition is to occur. For Figure 3b, we now calculate P(Branch B ⟶ Branch C) without regard for the intermediate branches. This modification is much cleaner than an alternate approach that might attempt to compute direct transition probabilities along all possible paths.

The other modification is to allow for simultaneous jobs. We treat the shared job sequences as if they are in a definite order, but the underlying situations can be complicated. For example, in Figure 3b, the reps work at Branches C and D simultaneously, not one after the other. To extend the model to handle these situations, we replace the quantity v_{ij} with a new quantity w_{ij}:

w_{ij} = P(move to Branch j at any point *simultaneous to or after* Branch i | currently at Branch i)
= (# reps who start at Branch j at any point *simultaneous to or after* starting at Branch i) / (# reps ever at Branch i).

The same caveats apply as for v_{ij}: the transition probabilities become less precise and correct with respect to direct transitions, but they can now be used in these more general situations.

4.3 Probabilistic Family

The probabilistic scoring model described above, which we refer to as PROB, treats jobs in a sequence as being ordered by time, but it does not take into account when the transitions occur. A transition is considered equally probable whenever it takes place. We create two variations on the model by changing the treatment of time.

First, we account for varying transition probabilities. We hypothesize that the scoring will be more accurate if we can represent single-event mass movements, as well as changes in industry patterns over time. For instance, consider the case where 30% of reps at Branch A eventually move to Branch B, but in 1997 Branch A was purchased by Branch B, so 99% of the reps who were at Branch A in 1997 also worked at Branch B in 1997. To account for such variations, rather than scoring a transition based on the probability of a rep moving from Branch A to Branch B, we describe a more specific event. Now, the rep is moving from Branch A at time X, to Branch B at time Y (specifically, the rep is first seen at Branch A at time X, and then first seen at Branch B at time Y which is equal to or later than time X). Time is divided into bins, with bins representing one year or more. Each branch has its own bin divisions, depending on the number of employees at the branch in different years. We allocate the bins so that there are at least 10 people who worked at each branch in each bin period, provided the branch has had that many employees during its history.

The parameters needed for this new model, called PROB-TIMEBINS, require changing p_i and (again) w_{ij}. We now compute

p_{iX} = # reps ever at Branch i during time X / # reps in db

y_{iXjY} = # reps ever at Branch i during time X and at Branch j during time Y, where Y ≥ X / # reps ever at Branch i during time X.

We take the opposite extreme for the second variation. The PROB model is not very informed about time: because the w_{ij} values describe the probability of being at Branch j anytime after or simultaneous to being at Branch i, only the relative order of i and j matter. To find out how important that directionality of time is, we create a simpler model, PROB-NOTIME, which ignores even the order of job moves. For this model, we use the original p_i (again, no need to compute the denominator), and a transition quantity z_{ij}, representing the raw number of reps who are at both branches i and j during their careers. There is an ambiguity in this formulation, in that now we should be able to score a set of shared branches regardless of the order in which they are presented; however,

transition probability from Branch i to Branch j
= (z_{ij} / # reps ever at Branch i)
≠ (z_{ij} / # reps ever at Branch j)
= transition probability from Branch j to Branch i.

As PROB-NOTIME turns out to work almost as well as PROB (see Section 5) and allows this framework to be applied to situations without a time ordering, we hope to explore the issue of ordering the branches in future work. For now, we use the same (temporal) ordering of branches as used in the other methods.

The JOBS ranking falls out as a trivial probabilistic model. If all branch transitions are considered to have the same probability, and all branches have the same probability of serving as a starting branch, then the ranking is equivalent to counting the number of shared jobs.

5. EVALUATION AND RESULTS

Ideal tribes consist of reps that know each other and have coordinated their movements among jobs. Since we cannot verify the personal relationships among thousands of securities reps across the country, we evaluate our tribes using indirect measures. First, we examine structural characteristics of the tribes produced with the various scoring methods. Then, we analyze the tribes' patterns of risk scores and geographic movement.

5.1 Tribe Structure

Using the basic process described in Section 3.2, we compiled a list (the edges *F'*) of the 3 million pairs of reps in the database that shared at least three different jobs. We ranked these pairs using the five scoring functions described in Section 4: JOBS, YEARS, PROB, PROB-TIMEBINS, and PROB-NOTIME. All but JOBS give quasi-continuous values as scores. For these, we can choose a threshold *d* to keep any desired number of pairs. When we compute the connected components of these pairs, we get a set of tribes of assorted sizes and a corresponding set of reps in these tribes. For JOBS, the scores are discrete: all pairs have at least 3 jobs, and the maximum number of shared jobs is 25. To compare the different scoring functions, for each continuous method we determine a cutoff *d* such that the resulting number of reps in the tribes matches (+/- 1) the number of reps in tribes formed with JOBS.

Figures 4 and 5 display structural characteristics of some tribe sets matched in this manner. We omit these characteristics for the variations on PROB (PROB-TIMEBINS and PROB-NOTIME), as they are substantially similar to those for PROB. Figure 4 indicates that PROB creates more tribes, and smaller tribes, than JOBS or YEARS. Figure 5 further shows that the majority of pairs created by the PROB ranking go into tribes of size two—pairs of associated reps. In contrast, JOBS and even more so YEARS, in order to get an equally large set of reps, provide many more pairs—edges in the graph *F'*—but the additional edges go to fill in the enormous components[1], instead of creating new, small groups.

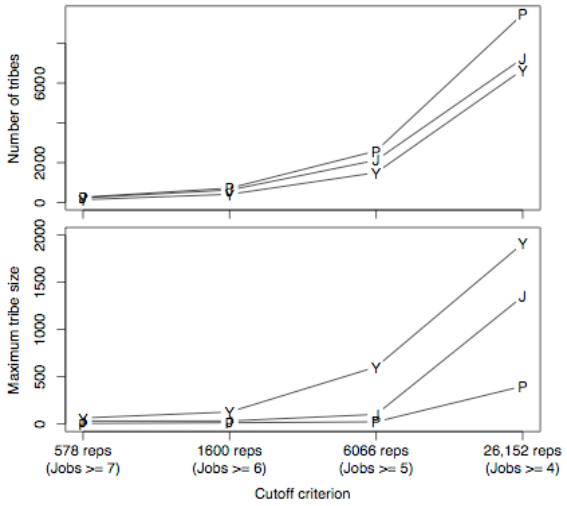

Figure 4. Number of tribes and maximum tribe size for equal-size sets of reps produced using JOBS (J), PROB (P), or YEARS (Y) to rank pairs.

[1] Components with hundreds or even with dozens of nodes are unlikely to be tribes of the kind we are looking for. In practice, we would probably disregard tribes with more than ten members. Dropping the larger tribes does not seem to change the evaluation measures, so we leave them in for the remaining analysis.

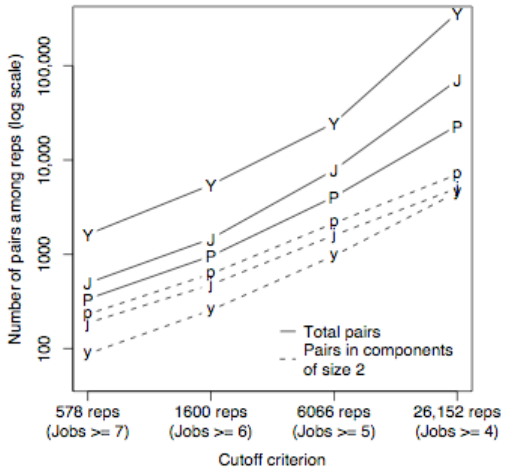

Figure 5. Number of pairs, and number of pairs in two-person tribes, for equal-size sets of reps produced using JOBS (J, j), PROB (P, p), or YEARS (Y, y).

We can see this effect from another perspective by considering the rarity of high-ranked job sequences. For JOBS and PROB, the scores are based solely on the job sequence; therefore, if a number of reps all share an identical job sequence, then the scores of their edges are equal. If that (shared) score passes the threshold, then the whole set of reps will be included in the tribes. For this reason, a ranking that scores common job sequences as significant will have large connected components among its tribes.

Table 1 shows the average, for each pair included in tribes, of the number of times its shared job sequence occurs among the 3 million pairs. The low averages for the PROB ranking confirm that this model succeeds in scoring rare sequences as significant. JOBS also brings in fairly rare sequences. For YEARS, when one pair passes the threshold *d*, others with the same job sequence do not necessarily cross it, since the score depends on how long the co-workers are together. However, we see that the reps working together for the longest times tend to have common sequences of jobs. For comparison, among all 3 million pairs, the job lists repeat an average of 40.72 times.

Table 1. Average number of times a job sequence occurs among all pairs of reps.

Ranking	# reps in tribes			
	578	**1600**	**6066**	**26,152**
PROB	1.06	1.07	1.21	1.51
JOBS	1.16	1.35	2.05	4.31
YEARS	315.73	194.05	87.07	224.78

Figure 6, below, gives a sense of how diverse the resulting tribes are. It shows, for several cutoffs, the percentage overlap between the set of reps produced by PROB and the equal-size set produced by each other ranking. We see that the PROB variations, particularly PROB-NOTIME, give results fairly close to PROB. The rep sets created by JOBS are related but substantially different, while those of YEARS have almost no overlap.

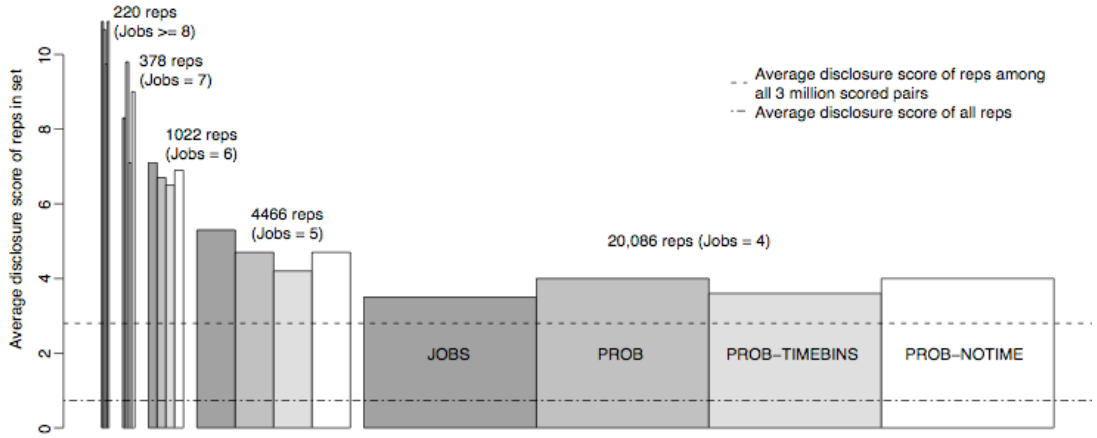

Figure 7. Disclosure scores of the top-ranked reps. Bar widths reflect the number of reps in each set.

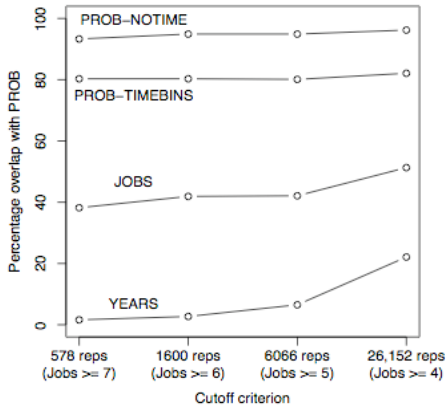

Figure 6. Percent overlap of rep set with that from PROB.

5.2 Disclosure Scores

As part of their oversight, NASD and other regulatory organizations require disclosures to be filed on reps for a variety of actions they commit and events that take place. These disclosures span categories such as customer complaints, bankruptcies, criminal charges, and regulatory actions; some are mundane and merely reflect administrative reporting requirements, while others represent serious breaches of trust. We can use these disclosures as assessments of past behavior or as predictors of future fraud risk. We compute a "disclosure score" for each rep as a weighted sum of their disclosures, where serious categories are weighted more highly (the weights were developed in consultation with NASD); in this system, the vast majority of reps are assigned a score of zero.

When we examine the disclosure scores of reps in tribes, we find that they score well above average, and that the scores of reps at the top of the rankings are higher than those lower down. Figure 7 displays the average disclosure scores of reps under different ranking systems. The reps are ordered and placed into bins based on which cutoff causes the rep to be included in the set of tribes. The bin widths correspond to the number of reps in the bin. Within each bin, the four bars correspond to sets of reps produced by JOBS, PROB, PROB-TIMEBINS, and PROB-NOTIME.

Results for YEARS are not displayed, as its scores are low: all fall below the higher dashed line. In fact, for the highest-ranked reps, the values are below the lower dashed line, and unlike with the other ranking systems, they rise as we move down the list of reps, reaching 2.4 for the largest set of reps. This may imply that the reps who have worked together for many years are least of all likely to commit fraud.

One alternative explanation for the high disclosure scores seen among these top reps is that the reps who have held such sequences of jobs together may simply have longer careers than average, and so have accumulated more disclosures over the years. We tested this explanation by dividing all reps into groups based on the number of jobs they have held and the number of years they have spent in the industry. Given a top-ranked set of reps from the tribes, we replaced the disclosure score of each rep with the average score from the rep's matched group, and recalculated the average for the set. If the matched disclosure scores were elevated, then our top-ranked reps would simply have long histories. In fact, the matched scores all give averages close to 2.8, the height of the dashed line, which means that length of career does not explain the high scores.

5.3 Disclosure Score Correlation within Tribes

If the tribes are of good quality *and* the conjecture is correct that reps at high risk of disclosures often move in tribes, then we would expect each tribe's disclosure scores to be homogenous. That is, disclosure scores of individuals within a tribe would be correlated: some tribes would have multiple members with high scores, while other tribes would have low scores. Judging tribes by the properties of their members' disclosure scores is not ideal, since the outcome depends on that second conjecture. In addition, since the frequency of disclosures is very low, under this lens only high-risk tribes look conclusively like high-quality tribes; low-risk tribes are hard to distinguish from random sets of reps. Finally, note the potential problem of incomplete information: reps that appear low-risk compared to their tribe-mates might just have evaded detection. It is precisely these individuals that the NASD may be interested in investigating in the future.

We performed several experiments to test whether the tribes are homogenous with respect to disclosure scores. First, we examined individual pairs of reps, using a chi-square test to assess whether reps with positive disclosure scores pair with others with positive scores more often than expected at random. If we used all the

pairs that formed tribes, then reps in large components would be represented more than once; to avoid this, we only performed this test on the tribes of size 2. Since the rankings are all significant at the $p \leq 10^{-7}$ level, we compared them using the phi-square statistic, which is chi-square normalized to have a maximum value of 1. By this measure, all five rankings are more or less equally significant, as shown at the top of Figure 8.

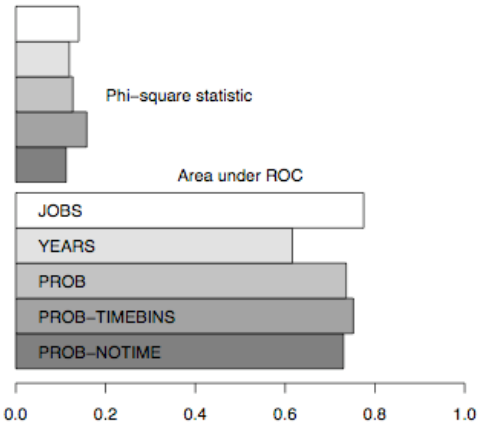

Figure 8. Comparison of tribe homogeneity using cutoff criterion of 1600 reps.

Next, we set up a prediction task with the tribes: we tried to predict the disclosure score of each rep. For each target rep, we took the other reps in the same tribe, averaged their disclosure scores, and used this average as the predicted value. We can compute an AUC (area under the ROC curve) for these predictions if the classification task is binary. The AUC values shown are for the task "is the rep's score higher than the average for this set?" By this measure, JOBS comes out a little more correlated than PROB-TIMEBINS, followed by the other PROB rankings, and YEARS trails.

5.4 Geographic Movement

The final indirect measure we use is the postal codes of the branches. If groups of reps move geographically, particularly large distances, this suggests they are staying together intentionally. Reps participating in the natural patterns of branch changes are less likely to move to far-off places together. We use the five-digit zip codes associated with most branches as a way to estimate geographic movement. The first digit designates a broad region of the United States, and the first three correspond to a particular large city or local region. Counting the number of unique one-digit or three-digit zip code prefixes associated with a rep pair's list of shared branches gives an idea of the geographic mobility of the pair. As with disclosure scores, since we expect many high-quality tribes will not have geographic movement, this measure can only be used to evaluate tribes in the aggregate.

Figure 9 displays information about geographic movement. For each pair in the set, we calculated how many unique one-digit and three-digit zip codes are covered by the shared jobs, as well as how many shared jobs there are with zip code information (96% of the branches have zip codes available). The values shown are the averages over the distinct shared job lists among the pairs.

The PROB rankings show the greatest mobility when we look at the number of zip codes covered. This is more surprising when we consider that the pairs in JOBS have a greater number of shared jobs, yet move less geographically. Pairs in the YEARS ranking move least of all, even less than the average among the 3 million scored pairs, which means that long-term co-workers tend to settle down. These long-term YEARS tribes—judging from their low disclosure scores, low overlap with the others, and low movement—do not seem to be the type of tribes we are looking for.

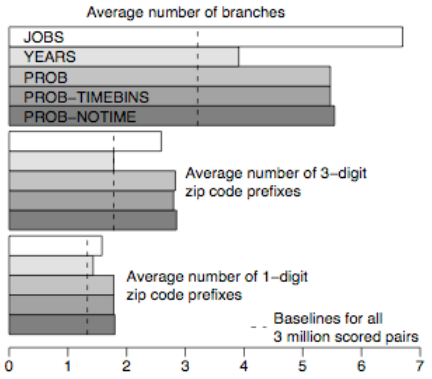

Figure 9. Comparison of geographic mobility using cutoff criterion of 1600 reps. For the distinct job sequences among each ranking's pairs, bars show average numbers of branches and zip code prefixes.

5.5 Discussion

To sum up what we have seen, the rankings JOBS, PROB, PROB-TIMEBINS, and PROB-NOTIME create tribes whose reps have higher disclosure scores, on average, than random (Section 5.2). Reps with high (or non-zero) disclosure scores are associated in tribes with other such reps under these rankings. At the cutoffs giving 1600 reps, PROB-TIMEBINS has a higher phi-square than the others, whereas JOBS gives the highest AUC; these results vary at other cutoffs, with phi-square remaining highest for either PROB-TIMEBINS or JOBS, and the highest AUC traded among JOBS and all the PROB–based models (Section 5.3). The PROB models create tribes that cross more zip codes among their shared jobs, even though the reps in JOBS have a higher number of shared jobs (Section 5.4). The PROB models produce more individual pairs in tribes, while JOBS and YEARS produce larger connected components as tribes (Section 5.1).

The fact that the JOBS and PROB models perform comparably at various cutoffs, yet pick different sets of reps, suggests that there is room for improvement by combining the best of both systems. Of the tribes ranked highly by JOBS but not PROB, some, on inspection, appear to be just the types we hoped to avoid: pairs of reps taking a large number of very common transitions together. Others look like good tribes, and it appears PROB may miss them because of poor probability estimates at small branches. When both reps at a two-person branch move to the same new job, it is impossible to tell whether they moved together because their firm was bought, or because they wanted to stay together. The PROB model assumes the former, calculating the move as 100% likely to occur by chance, but this may not be the best policy. More generally, the PROB model seems to favor large firms, either because the probability estimates are more stable there, or perhaps because it is possible to create smaller transition probabilities from larger firms. We have not yet succeeded in correcting for

this property, and the conclusion might be that the model is simply better suited for situations with large branches.

Qualitatively, many of the tribes look convincing when the reps' job histories are displayed together. It is a compelling feature that transition dates often coincide closely, even though the model did not use them.

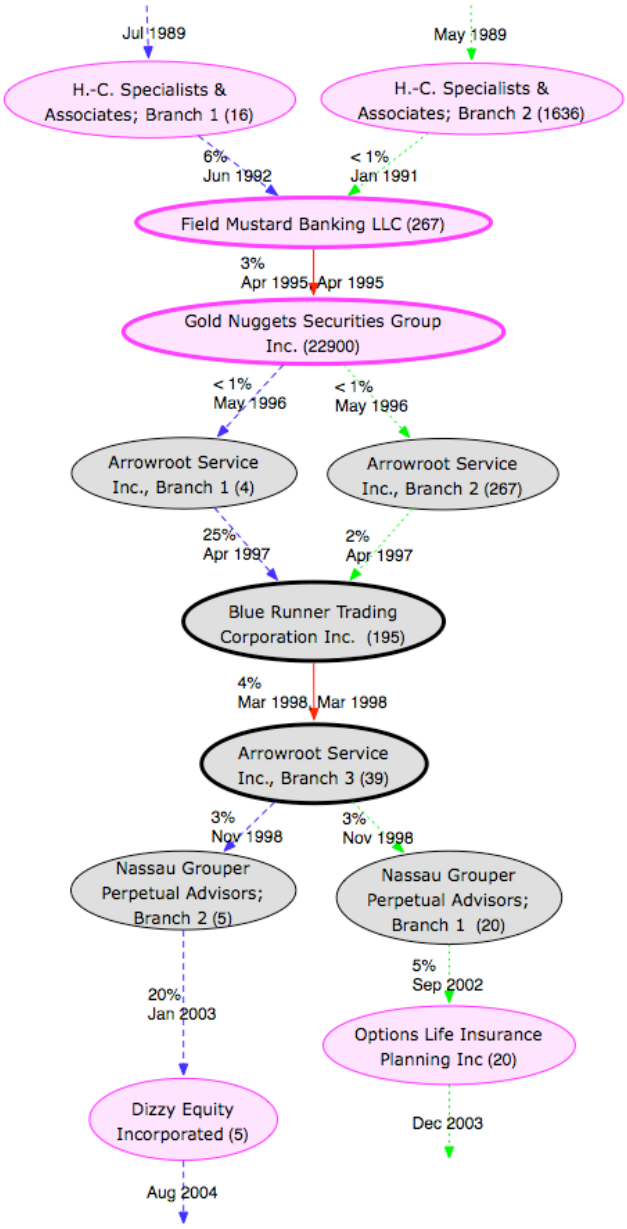

Figure 10. Example tribe ranked highly by PROB but not by JOBS. Nodes indicate branches and their sizes. Arrows leading into a node show the starting dates of employment and the transition probabilities. Solid lines are moves executed by both reps in the pair; dashed and dotted lines are moves by one member only. Firm names are fictitious.

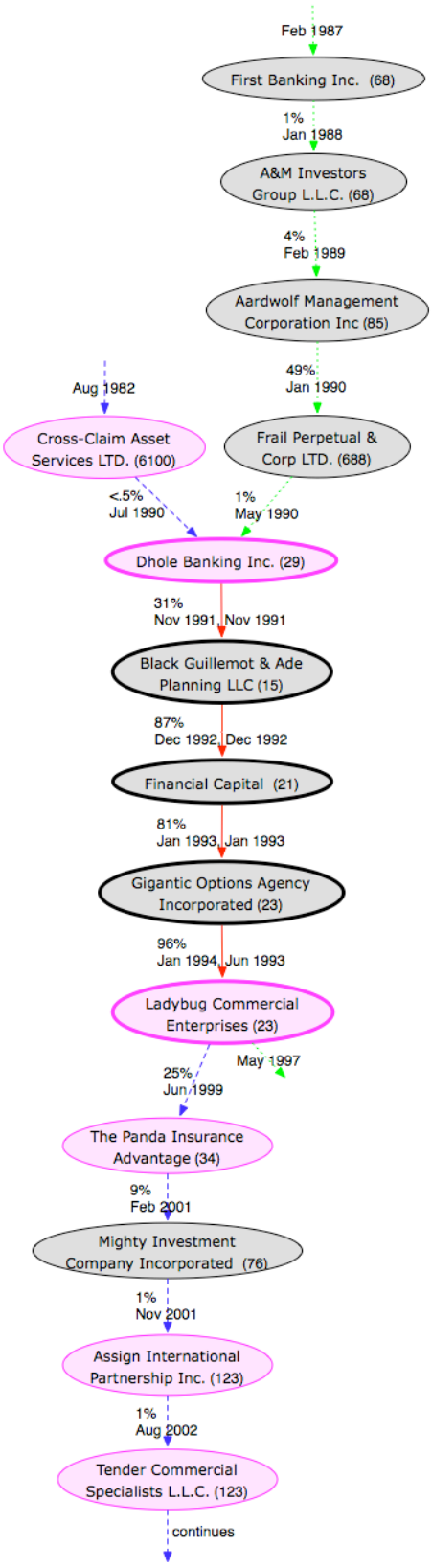

Figure 11. Example tribe ranked highly by JOBS but not by PROB. Firm names are fictitious.

As examples, Figures 10 and 11 display the career histories of two potential tribes. Each of these tribes consists of a single pair of reps. The pair in Figure 10 was scored by PROB as highly significant, while that in Figure 11, even though it has a long history together and was ranked highly by JOBS, appears to be following typical patterns; it was scored as not significant by PROB. As it turns out, the reps from the significant pair have disclosure scores of 18 and 24, primarily since in April 1996 they were both fired (disclosures show an Internal Review and a Termination for each). One of the reps from the non-significant pair has no disclosures, while the other was fired in 1997 for "diversion of profitable trades to personal," for which they received a score of 12.

6. RELATED WORK

Our task of identifying small, anomalously similar groups is novel within relational knowledge discovery but has analogs in other fields. Within the analysis of complex relational and social networks, it is common to cluster the graph or otherwise infer hidden group structure [17], [12], but usually the aim is to find large-scale communities, such as among webpages [8], employees in a single organization [22], or bottlenose dolphins [15]. In addition, these algorithms are typically designed for static or time-collapsed networks, whereas the temporal aspect is important for us.

In time series analysis, there is research within the database community on efficiently finding identical or similar sequences [1], and on constructing flexible definitions of similarity [5]. Econometrics has a related concept called cointegration: two time series X and Y (e.g., of stock prices) may be cointegrated if X_t is useful for predicting Y_{t+1} [10]. However, in these fields, time series are traditionally numerical. Furthermore, in our task we wish to find sequences that are not just similar, but also anomalous.

Anomaly detection, often applied to the security task of intrusion detection, does highlight unusual time-sequence patterns against a background of normal activity, often learning a background model from the data [21]. A recent paper by Eskin [6] offers a clear formulation that treats the data as a mixture model of normal with anomalous sequences, a technique that could be useful for scoring pairs in our scenario, although we would still need to specify the form of the normal model as we do here. For anomaly detection in relational data, Lin and Chalupsky [14] offer a measure of path rarity that can be used to find the closest match to a given individual, although it does not compare one set of individuals to another.

In modeling dynamic networks, a few papers offer related ideas. Magdon-Isamil et al. [14], searching for hidden groups, propose a Markov chain model of how individuals' group affiliations change over time, a model general enough to allow multiple simultaneous memberships along with individual preferences. This framework could potentially make our probabilistic model cleaner, although it would need to be heavily constrained to reduce the number of parameters required. Lahiri and Berger-Wolf [13] introduce an algorithm for dynamic graphs that predicts future interactions (edges) at each time step based on patterns of interactions at previous time steps. With an appropriate mapping of our branch transitions into their interactions, this approach might provide a different way of modeling the background transition patterns we try to capture.

Semi-Markov models are a standard type of stochastic process for modeling transitions with timing information [23]. They contain parameters not only for transition probabilities between states, but also for durations of stay in each state. While our models neglected durations (in order to focus on simultaneity of affiliations), semi-Markov models would provide a natural, richer way to model the transition processes with respect to individual reps.

The caravan identification task mentioned in the introduction has a realistic motivation from the military: Burns et al. [3] describe a system that uses airborne video surveillance data to detect convoys moving on the ground. The image data they use, however, is not nearly clear enough to make statistical modeling feasible.

Most intriguingly, animal biologists have long faced something like the tribe-finding task: given observations of animals in groups, taken at different time points, they ask which pairs of animals are highly associated. (These "association patterns" are used as the links for animal social networks [15], [13].) The most common association measure, the Half-Weight Index [4], is a simple function of the number of times the animals are seen together vs. apart, but Bejder et al. propose a more sophisticated network randomization test [2]. We are investigating this literature as part of ongoing work, and note a few aspects here. First, the associations are impossible to verify directly, but there is work validating the methods through simulation. Second, the models ignore time, which seems reasonable in that domain given that each distinct group is only observed once.

7. CONCLUSIONS AND EXTENSIONS

One of the strengths of this work is that, beginning with no explicit knowledge of this industry, we can discover, model, and factor out typical job transitions, even though in real life these are caused by a combination of geography, career tracks, and other factors. Moving forward, we may extend our model by incorporating external or domain-specific information. For example, we could consider relationships between reps who work in the same city but not at the same branch, and we could better handle some odd cases of reps with many simultaneous jobs given a better understanding of the industry and the data sources.

In this work, we had access to a complete history of employments and disclosures so far. In practical use, tribe identification will be more of an ongoing process, a situation we need to consider; it will be more difficult to recognize tribes when they have shared only a few jobs.

The most interesting aspect of our formulation, compared to related work, is our accounting for simultaneous jobs and different paths between the same jobs. We needed to allow for multiple affiliations starting and ending at arbitrary times, yet our model does not describe the network's changes day by day; instead, we observed certain discrete events (job transitions, and co-workers intersecting at a job) as time moved forward.

It may be worthwhile to incorporate more timing information, such as job durations, into our model, or other properties like the lengths of reps' non-intersecting careers. In the direction of simplifying, we plan to explore the time-oblivious version of the model (PROB-NOTIME), to see how well it can be applied to other types of tasks. In addition, we may incorporate a clustering or other dimensionality-reduction technique for the branches, either as an initial step in order to produce fewer but more robust

transition probabilities, or afterwards to further analyze the resulting transition graph. More immediately, we are investigating adjustments that may improve the model's behavior with small branches. Finally, we hope to experiment with other domains and data sets.

8. ACKNOWLEDGMENTS

The National Association of Securities Dealers provided generous research support for this work, and worked with us to scope and further refine this analysis. In particular, Henry Goldberg and John Komoroske provided invaluable assistance. We also thank Bret Aarden and Michael Cuthbert for helpful comments in the course of this project, Cindy Loiselle for her careful editing, and Marc Maier for reminding us about zebras. This research is supported by the Central Intelligence Agency, the National Security Agency and National Science Foundation under NSF grant #IIS-0326249. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements either expressed or implied, of the funding sources or the U.S. Government.

9. REFERENCES

[1] Agrawal, R., Lin, K. I., Sawhney, H. S., and Shim, K. Fast similarity search in the presence of noise, scaling, and translation in times-series databases. In *Proc. 21st Int. Conf. on Very Large Data Bases (VLDB '95)*, 490-501.

[2] Bejder, L., Fletcher, D., and Bräger, S. A method for testing association patterns of social animals. *Animal Behaviour, 56,* 3 (Sept. 1998), 719-725.

[3] Burns, J., Connolly, C., Thomere, J., and Wolverton, M. Event recognition in airborne motion imagery. In *Capturing and Using Patterns for Evidence Detection—Papers from the AAAI Fall Symposium*. AAAI Press, 2006.

[4] Cairns, S. J. and Schwager, S. J. A comparison of association indices. *Animal Behaviour, 35,* 5 (Oct. 1987), 1454-1469.

[5] Das, G., Gunopulos, D., and Mannila, H. Finding similar time series. *Principles of Data Mining and Knowledge Discovery (PKDD '97)*, 88-100.

[6] Eskin, E. Anomaly detection over noisy data using learned probability distributions. In *Proc. 17th International Conf. on Machine Learning (ICML '00)*, 255-262.

[7] Fast, A., Friedland, L., Maier, M., Taylor, B., and Jensen, D. Data pre-processing for improved detection of securities fraud in relational domains. In *Proc. 13th ACM Int. Conf. on Knowledge Discovery and Data Mining (KDD '07)*.

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

[9] Goldberg, H. G. and Senator, T. E. Restructuring databases for knowledge discovery by consolidation and link formation. In *Proc. 1st ACM Int. Conf. on Knowledge Discovery and Data Mining (KDD '95)*, 136-141.

[10] Granger, C. W. J. Some properties of time series data and their use in econometric model specification. *J. Econometrics* 16 (1981), 121-130.

[11] Jensen, D. and Neville, J. Data mining in social networks. In *Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers* (National Academy of Sciences, November 7-9, 2002). National Academies Press, Washington, DC, 2003, 287-302.

[12] Kubica, J., Moore, A., Schneider, J., and Yang, Y. Stochastic link and group detection. In *Proc. 18th Nat. Conf. on Artificial Intelligence (AAAI '02)*, 798-804.

[13] Lahiri, M. and Berger-Wolf, T. Y. Structure prediction in temporal networks using frequent subgraphs. *IEEE Symposium on Computational Intelligence and Data Mining (CIDM '07)* (April, 2007, Honolulu, Hawaii).

[14] Lin, S. and Chalupsky, H. Unsupervised link discovery in multi-relational data via rarity analysis. *Third IEEE International Conference on Data Mining (ICDM '03),* 171.

[15] Lusseau, D. and Newman, M. E. J. Identifying the role that individual animals play in their social network. *Proc. R. Soc. London B (Suppl.)* 271 (2004), S477-S481.

[16] Magdon-Ismail , M., Goldberg, M., Wallace, W., and Siebecker, D. Locating hidden groups in communication networks using hidden Markov models. In *Proc. NSF/NIJ Symposium on Intelligence and Security Informatics* (June 2003), 126-137.

[17] Neville, J. and Jensen, D. Leveraging relational autocorrelation with latent group models. In *Proc. 5th IEEE Int. Conf. on Data Mining (ICDM '05),* 322-329.

[18] Neville, J., Şimşek, Ö., and Jensen, D. Autocorrelation and relational learning: Challenges and opportunities. In *Proc. Workshop on Statistical Relational Learning, 21st Int. Conf. on Machine Learning* (2004).

[19] Neville, J., Şimşek, Ö., Jensen, D., Komoroske, J., Palmer, K., and Goldberg, H. Using relational knowledge discovery to prevent securities fraud. In *Proc. 11th ACM Int. Conf. on Knowledge Discovery and Data Mining (KDD '05)*.

[20] Salton, G. and Buckley, C. Weighting approaches in automatic text retrieval. *Information Processing and Management*, *24*, 5 (1988), 513–523.

[21] Teng, H. S. and Chen, K. Adaptive real-time anomaly detection using inductively generated sequential patterns. *IEEE Symposium on Security and Privacy* (1990), 278.

[22] Tyler, J. R., Wilkinson, D. M., and Huberman, B. A. Email as spectroscopy: Automated discovery of community structure within organizations. *Communities and Technologies*. Kluwer, B. V., Deventer, Netherlands, 2003, 81-96.

[23] Weiss, E. N., Cohen, M. A., and Hershey, J. C. An iterative estimation and validation procedure for specification of semi-Markov models with application to hospital patient flow. *Operations Research*, 30, 6 (Nov. - Dec. 1982), 1082-1104.