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

D.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 more detailed view of the connections between entities, particularly in dynamic temporal domains, promises to aid analyses of the data. This paper 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 [10] exploits connections among individuals, as well as intrinsic attributes, to find patterns and make predictions. One notable property in relational, or network-structured, data sets is autocorrelation, or homophily: the tendency of connected entities to have similar attribute values [17]. However, raw data does not always expose the links that account for these correlations. To create a better view, raw data must be refined, whether by preprocessing it to identify real-world entities and their relations [8], or further by inferring latent structure, for instance at the level of groups or communities [16], [11], [7]. In this work, we identify finer-grained, strong associations among individuals in a large, dynamic 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 are affiliated with organizations, 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 describes the likelihood that two (or more) 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; or, discovering family structure in animal groups, from tagged individuals frequently sighted together (see Related Work). If we remove the temporal aspect of the problem and simply require a bipartite graph of affiliations, then we could generalize the model to find 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 with large organizations, where the original data does not describe associations among individuals at the desired level of detail. For instance, in our employment domain from the securities industry, people often work at branches of thousands. 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 just "reps." With over 600,000 reps under its jurisdiction, the NASD must focus its investigatory resources on those most likely to commit fraud or other violations of securities regulations. In conversations over the course of related projects [6][18], NASD representatives suggested that fraud may be committed by colluding groups of reps that move 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, such tribes will certainly also include 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: mostly low-risk or mostly high-risk.

Our source data is a table of employment histories: for each rep, a series of records containing 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 one to ~35,000 employees. The branch identities themselves have been inferred, through an earlier process of link consolidation from office addresses [6], 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, and almost a quarter refer to currently held jobs (as of May 2006). Even though many of the records are historical—referring to branches and reps no longer under NASD's jurisdiction—we use the whole collection.

Two constraints of the real-world data shape our approach. First, many employment histories include simultaneous, overlapping jobs or leave gaps between employments (at least, between employments in this industry). 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 treat job changes as the basic units in the task; instead, we direct our attention to the times and places that people have been co-workers.

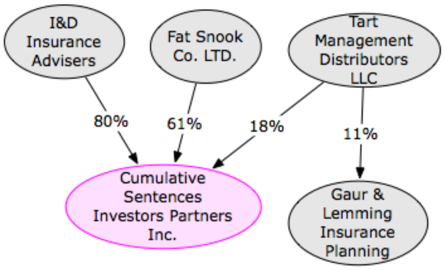
I&D Insurance Advisers
Fat Snook Co. LTD.
Tart Management Distributors LLC
80%
61%
18%
11%
Cumulative Sentences Investors Partners Inc.
Gaur & Lemming Insurance Planning

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

Second, 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 bought. Reps in this flow 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 3.3; roughly speaking, they indicate the percentage of employees at one branch that later work at the attached (destination) branch. Many of these transition percentages are high, which confirms that job movement in the industry is not random. Among branches of fewer than ten employees, about 73% have a destination branch 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 popular 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 exactly what we hope to factor out in order to find genuinely tight associations among individuals.

3. APPROACH

3.1 Basic Tribe-Finding Process

Formally, we are given a bipartite graph $G = (R \cup O, E)$ of reps $R = \{r_1, r_2, ..., r_n\}$ and organizations $O = \{o_1, o_2, ..., o_m\}$. Each edge $e \in E$ is annotated with a time interval: $e = (r_i, o_j, tstart_{ij}, tend_{ij})$. Our tribe-discovery process begins with finding 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 simply by iterating through the branches and recording every pair of reps $f_{ij} = (r_i, r_j)$ whose employment stints at a branch intersect.

For each pair, we then summarize their co-worker relationships, keeping track of the jobs where they coincide. We record 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:

q_{ij} = { the sequence of jobs $\{o_x, o_y, ...\}$ shared by r_i and r_j $\cup$ additional information described above}.

For purposes of efficiency, we retain only the rep pairs that have at least three jobs in common. This leaves 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 $c_{ij}(q_{ij})$ for each edge in F', measuring how significant or unusual its sequence of shared jobs is. The rest of Section 3 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 retain only 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'.

3.2 Scoring/Ranking Functions

The choice of scoring methods constitutes the heart of the task. (Strictly speaking, we only use the scores to create a ranking of the pairs, so we also use the term "ranking method.") We propose and compare several. 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 simple 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 geographical and other factors. Accounting for these patterns motivates the probabilistic models that follow.

3.3 Probabilistic Model

In developing a simplified model for the job history data, there is a tradeoff in how specific to make it. 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. (Say, all reps start their careers at the same time; it does not matter for the eventual model). 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 transition probabilities are static over time. If this were our model, then the quantities we would need to estimate are:

p_i = P(start at branch i), and
t_{ij} = P(transition from branch i to branch j | [given that] currently at branch i).

Then, we could estimate the probability of a rep having any given job sequence as:

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

The probabilities are straightforwardly estimated using:
p_i = # reps ever at branch i / # reps in database
t_{ij} = # reps who leave branch i and next go to branch j / # reps ever at branch i.

Using the ordinary Markov chain and the null hypothesis of independent movement, we would score the sequence of Figure 2 as shown.

1. P(rep 1 holds this sequence of jobs) = x
2. P(reps 1 and 2 each hold this sequence of jobs) = x^2.
3. P(some two reps in database hold this sequence of jobs) follows a binomial distribution, with n = # reps in the database, and $p = x^2$.

Steps 2 and 3 are monotonic transformations of 1, so if the scoring function only needs to return a ranking, it is enough to calculate x. Further, it is not necessary to compute the denominator of p_i. For the example in Figure 2a), the score would be $p_A \cdot t_{AB} \cdot t_{BC} \cdot t_{CD}$ = $(4234)(.005)(.01)(.005)$. (The "<5%" of the diagram would be an exact figure normally. The self-loop of one rep at Branch A is ignored.) For situations where reps start or end at separate jobs, we only score the sequence they share.

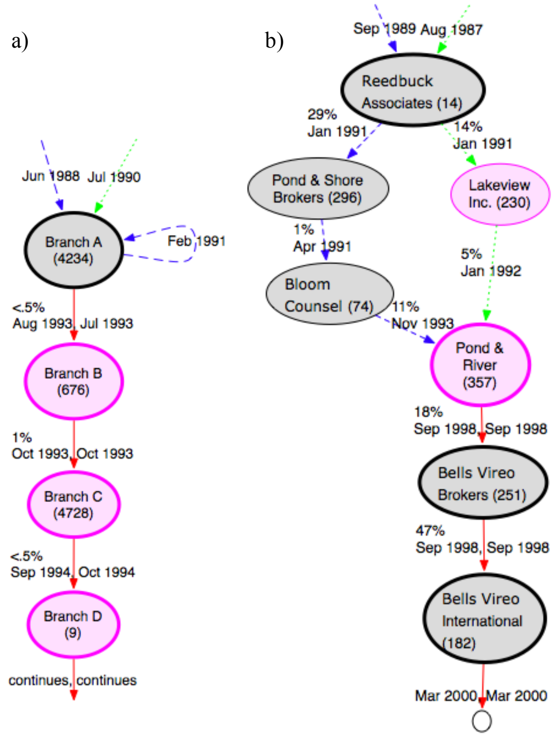

Figure 2. Job sequences to score. Nodes indicate branches and their sizes. Arrows leading into a node show the dates the new job is started, and the transition probabilities (see text). Solid lines are moves executed by both reps in the pair; dashed lines are moves by one member only, and dotted are by the other. The firm names are fabricated to suggest correspondences visible in the real data. These diagrams are modified from pairs scored as significant; hence, the reps' start dates match closely, although the timing information was not used in scoring.

If job sequences in the database were as simple as Figure 2a), this model would be sufficient. However, Figure 2b) is more typical of the data. The reps in this example start at the same branch, split apart for a few years, come back together, and then both begin two jobs at related companies at the same time. To allow for these more complex situations, we adjust the model in such a way that it is no longer a Markov chain, but the probability calculations are almost the same.

The major modification is to allow reps to have different paths between shared jobs, as is shown near the top of Figure 2b). To do this, we change the quantity t_{ij}, which describes the probability that a rep moves to branch j immediately after branch i, to a new quantity v_{ij}, describing the probability that a rep moves to branch j *at any point* after working at branch i. Now, each v_{ij} ş t_{ij}, and the transition probabilities leaving a branch no longer sum to 1 ($\sum_i t_{ij} = 1$, but $\underset{i}{\geq} v_{ij} \geq 1$). 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 2b), we then calculate

P(Reedbuck $\longrightarrow$ Pond & River) (a percentage not displayed in the figure), 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 drawback is that even in the case of direct moves we compute with v_{ij}, though t_{ij} would be more appropriate. It may be the case that $v_{ij} >> t_{ij}$ only for branches j that are rarely reached directly from i, and $v_{ij} \approx t_{ij}$ for branches j that are reached directly; if so, then the substitution is not a problem.

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 instance, rep 1 can start at branch A, then add branch B, while rep B starts at branch B and later adds branch A. Then the reps overlap at B before they overlap at A, although rep 1 never left branch B for branch A. Or, as in Figure 2b), the reps may be at both Bells Vireo firms simultaneously, not one after the other. To extend the model to handle these situations, we replace the quantity v_{ij}, the probability that a rep moves to branch *j at any point after* working at branch i, with a new quantity w_{ij}, the probability that a rep works at branch *j at any point simultaneous to or after* working at branch i. The same caveats apply as for v_{ij}: the transition probabilities become less precise and correct, but can now be used in these more general situations. The transition probabilities shown in Figure 2, and later in Figures 4 and 5, are actually w_{ij} values, so the example calculation for Figure 2a) is computed as we discussed earlier, but the meanings of the probabilities are different.

3.4 Family of Models

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 the years. For instance, consider the case where 30% of reps at branch A eventually move to branch B, but 99% of the reps at branch A in 1997 were seen at branch B later in 1997 after it purchased branch A. So, 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 at 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 ş 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, as 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 (final) 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 ordered they are presented in; however:

transition probability from i to j = (z_{ij} / p'_i) ş (z_{ij} / p'_j) = transition probability from j to i.

As PROB-NOTIME turns out to work almost as well as PROB (see Section 4) 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 branches have the same probability for being started at, then the ranking is equivalent to counting the number of shared jobs.

4. EVALUATION AND RESULTS

Ideal tribes consist of reps that certainly know each other and have coordinated their movements among jobs. Since we cannot directly 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.

4.1 Tribes Produced

Using the basic process described in Section 3.1, we compiled a list (the edges F') of the 3.07 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 Sections 3.2-3.4: 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; then, when we compute the connected components of the 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 three 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. Tables 1-3 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.

Table 1. Tribe network structure for JOBS ranking

JOBS criteria	# reps	# pairs	# tribes	max tribe size	# reps in tribes size 2
jobs ş 7	578	495	232	31	374
jobs ş 6	1600	1461	623	32	952
jobs ş 5	6066	7855	2124	101	3188
jobs ş 4	26,152	70,209	7244	1350	10,044

Table 2. Tribe network structure for PROB ranking

# reps	# pairs	# tribes	max tribe size	# reps in tribes size 2
578	336	266	6	464
1600	958	718	13	1240
6066	4072	2591	23	4284
26,152	23,193	9468	400	14,064

Table 3. Tribe network structure for YEARS ranking

# reps	# pairs	# tribes	max tribe size	# reps in tribes size 2
578	1624	140	64	176
1600	5446	408	127	512
6066	24,672	1498	604	1934
26,152	362,966	6669	1910	9092

Naturally, 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 perhaps ten members. Dropping the larger tribes does not seem to change the evaluation measures, so we leave them in for the remaining analysis. What the tribe structures in these tables show is that the PROB ranking is more inclined to produce tribes of size two—pairs of associated reps. 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, instead of creating new small groups.

We can see this effect from another perspective by considering the frequency 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 4 examines this frequency of high-ranked job sequences. It displays the average, for each pair included in tribes, of the number of times its job sequence occurs among the 3 million pairs. The low averages for the PROB ranking confirms 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, 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 actually have common sequences of jobs.

Table 4. For each job sequence among the top-ranked pairs, average number of times it occurs among all pairs of reps. Among all 3 million pairs, the job lists repeat an average of 40.72 times.

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

Error! Reference source not found. below gives a sense of how diverse the tribes produced by different scoring methods are. It shows, for several cutoffs, the percentage overlap between the set of reps selected by PROB and each other ranking. We see that the PROB variations give results fairly close to PROB, particularly PROB-NOTIME. The reps sets created by JOBS are related but substantially different, while those of YEARS have almost no overlap.

Table 5. Percent overlap of rep set with that from prob

Ranking	# reps in tribes			
	578	1600	6066	26,152
JOBS	38.2%	41.9%	42.1%	51.3%
YEARS	1.6%	2.7%	6.5%	22.1%
PROB-TIMEBINS	80.3%	80.3%	80.1%	82.1%
PROB-NOTIME	93.3%	94.9%	94.9%	96.2%

4.2 Disclosure Scores

As part their oversight, the 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 required to be reported, 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 the tribes are strongly enriched for reps with high scores. Figure 3 displays the average disclosure scores of reps in different ranking systems. The reps are assigned to bins A-G based on what cutoff causes the rep to be included in the set of tribes (see Table 6). For instance, bins A-D comprise the top 578 reps, and A-E comprise the top 1600; bin E contains the reps that appear at ranks 579 to 1600. The bin widths correspond to the number of reps in the bin, for bins A-E; bins F and G would be too wide to fit in the diagram, so their widths as displayed are not meaningful.

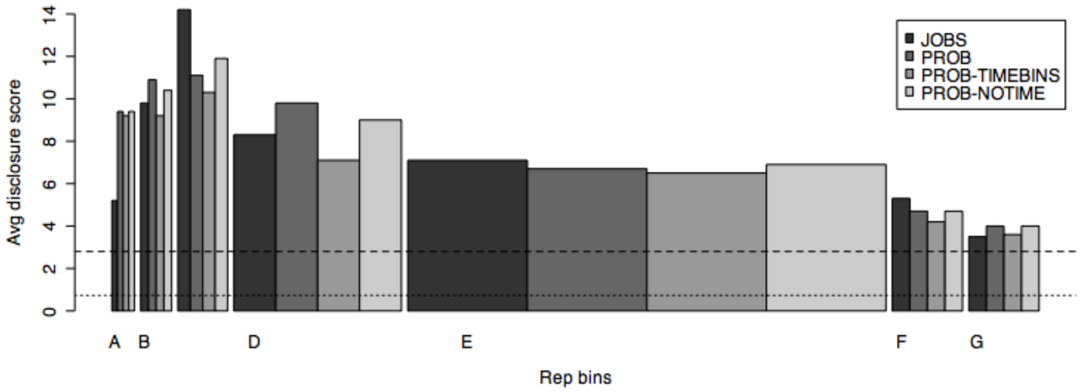

Figure 3. Disclosure scores of the top-ranked reps.

Within each bin, the four bars correspond to reps produced by JOBS, PROB, PROB-TIMEBINS, and PROB-NOTIME. The dashed horizontal line at score = 2.8 is the average disclosure score of all the (unique) reps among the 3 million pairs (*F′*). The dotted line at score = 0.7 is the average score for all reps in the database.

Table 6. Bins used in Figure 3.

Bin	JOBS criteria determining bin size	# reps in bin
A	jobs ş 12	48
B	9 ş jobs ş 11	66
C	jobs = 8	106
D	jobs = 7	358
E	jobs = 6	1022
F	jobs = 5	4466
G	jobs = 4	20,086

The overall trend is very encouraging for these rankings. They all score well above average, and the average disclosure scores of reps produced at the top of the rankings are higher than those lower down. The smaller bins A-C are more variable, as they contain only 220 reps total. It is interesting to see, however, that the disclosure score of JOBS drops in the highest bin (A), and then compensates for it in bin C.

YEARS is not displayed, as its scores are low: all fall below the dashed line. In fact, in bins A-C the values are below the dotted line, and unlike with the other ranking systems, they rise as we move down the list of reps, reaching 2.4 in bin G. This might 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 here 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 test this explanation by dividing all reps into groups based on the number of jobs they have held and the number of years they have in the industry. Given a top-ranked set of reps from the tribes, we replace the disclosure score of each rep with the average score from the rep's matched group, and recalculate the average for the set. If the matched disclosure scores are elevated, then our top-ranked reps simply have long histories. In fact though, the matched scores all give averages close to 2.8, the height of the dashed line, which means that the length of their careers does not explain away the high scores.

4.3 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, some tribes would tend to 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 expected 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 here: 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 perform several experiments to test whether the tribes are homogenous with respect to disclosure scores. First, we examine 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 take all the pairs that form tribes, then reps in large components will be represented more than once; to avoid this, we only perform this test on the tribes of size 2. Since the rankings are all significant at the p ş 10^{-7} level, we can compare them using the phi-square statistic, which is chi-square normalized to have maximum value 1. By this measure, all five rankings are more or less equally significant, as shown in Table 7.

Table 7. Comparison of tribe homogeneity, using top 1600 reps

Ranking	# pairs	# tribes	Phi-sq	Avg disc	AUC
JOBS	1461	623	0.140	7.9	0.775
YEARS	5446	408	0.119	1.4	0.616
PROB	958	718	0.127	7.9	0.736
PROB-TIMEBINS	960	714	0.158	7.1	0.752
PROB-NOTIME	965	718	0.112	7.9	0.730

Next, we set up a prediction task with the tribes: we try to predict the disclosure score of each rep. For each target rep, we take the other reps in the same tribe, average their disclosure scores, and use 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.

4.4 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 is an indicator they are staying together intentionally. Reps participating in the natural patterns of branch changes are less likely to be moving to far-off places together. We have the five-digit zip codes associated with most branches (96%). 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 a rough 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.

Table 8 displays information about geographic movement. For each pair in the set, we calculate how many unique 1-digit and 3-digit zip codes are covered by the shared jobs, as well as how many shared jobs there are with zip code information (96% of branches have zip codes available). The numbers shown are the averages over the distinct 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 more shared jobs, yet move less geographically. Pairs in the YEARS ranking move least of all, even less than the average of the 3 million, 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.

Table 8. Comparison of geographic mobility, using top 1600 reps

Ranking	# unique job sequences	Avg # 1-digit zips	Avg # 3-digit zips	Avg # branches with zips avail
JOBS	1085	1.58	2.59	6.70
YEARS	738	1.43	1.78	3.91
PROB	896	1.78	2.83	5.47
PROB-TIMEBINS	899	1.78	2.80	5.47
PROB-NOTIME	893	1.80	2.85	5.54
all scored pairs	75,321	1.33	1.78	3.21

4.5 Discussion

To sum up what we have seen, all the rankings JOBS, PROB, PROB-TIMEBINS, and PROB-NOTIME create tribes whose reps have higher disclosure scores, on average, than random (Section 0). Reps with high (or non-zero) disclosure scores are associated in tribes with other such reps under all rankings. At the cutoffs giving 1600 reps, PROB-TIMEBINS has a higher phi-square than the others, whereas JOBS gives the highest AUC; these 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 4.3). 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 4.4). The PROB models produce more individual pairs in tribes, while JOBS and YEARS produce larger connected components as tribes (Section 4.1).

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 2-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, although the model did not use them.

As examples, Figures 4 and 5 display the career histories of two potential tribes. Each of these tribes consists of a single pair of reps. The pair in Figure 4 was scored by PROB as highly significant, while that in Figure 5, 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" and received a score of 12 for this.

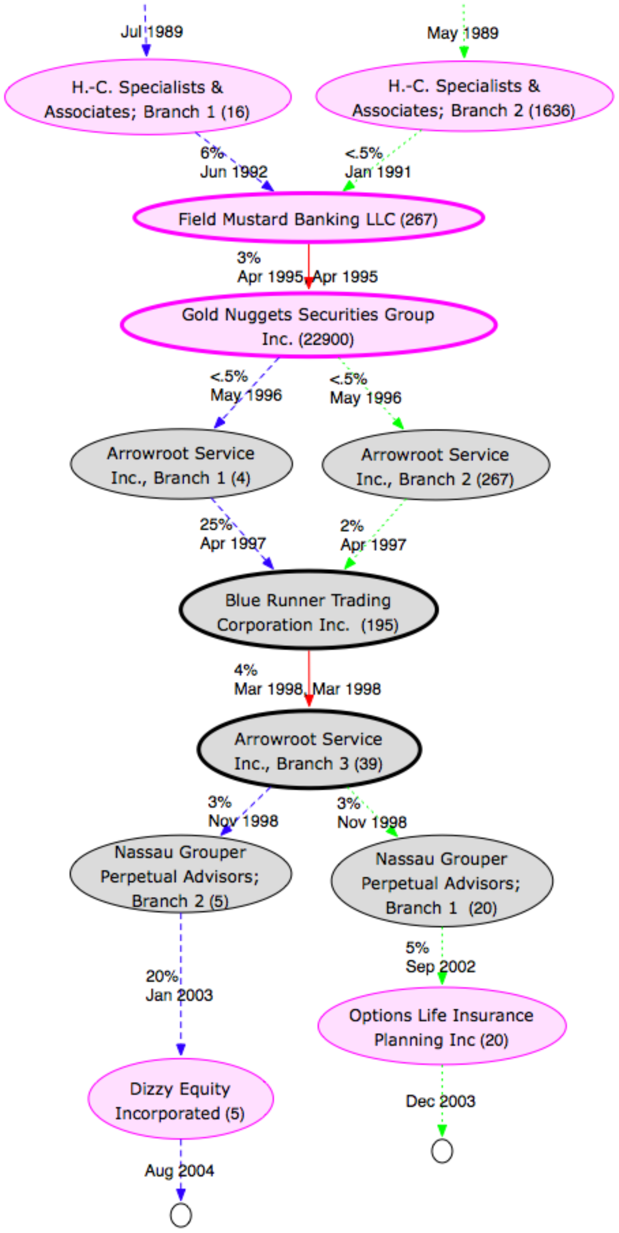
Jul 1989
May 1989
H.-C. Specialists & Associates; Branch 1 (16)
H.-C. Specialists & Associates; Branch 2 (1636)
6% Jun 1992
<.5% Jan 1991
Field Mustard Banking LLC (267)
3% Apr 1995
Apr 1995
Gold Nuggets Securities Group Inc. (22900)
<.5% May 1996
<.5% May 1996
Arrowroot Service Inc., Branch 1 (4)
Arrowroot Service Inc., Branch 2 (267)
25% Apr 1997
2% Apr 1997
Blue Runner Trading Corporation Inc. (195)
4% Mar 1998
Mar 1998
Arrowroot Service Inc., Branch 3 (39)
3% Nov 1998
3% Nov 1998
Nassau Grouper Perpetual Advisors; Branch 2 (5)
Nassau Grouper Perpetual Advisors; Branch 1 (20)
5% Sep 2002
20% Jan 2003
Options Life Insurance Planning Inc (20)
Dizzy Equity Incorporated (5)
Dec 2003
Aug 2004

Figure 4. Example tribe ranked highly by PROB but not by JOBS. Refer back to Figure 2 for meanings of labels. Firm names are fictitious.

5. RELATED WORK

Our task of identifying small, anomalously similar groups is novel within the world of 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 [16], [11], but usually the aim is to find large-scale communities, such as among webpages [7], employees in a single organization [20], or bottlenose dolphins [14]. 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 [4]. 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} [9]. However, in these fields, time series are traditionally numerical. Furthermore, in our task we need to find sequences that are not just similar, but anomalously similar.

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 [19]. A recent paper by Eskin [5] 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 [13] 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. [13], searching for hidden groups, propose a Markov chain model of how individuals' group affiliations change over time, one 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 [12] 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.

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 studied in above papers [14], [12].) The most common association measure, the Half-Weight Index [3], 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 given that each group is only observed once.

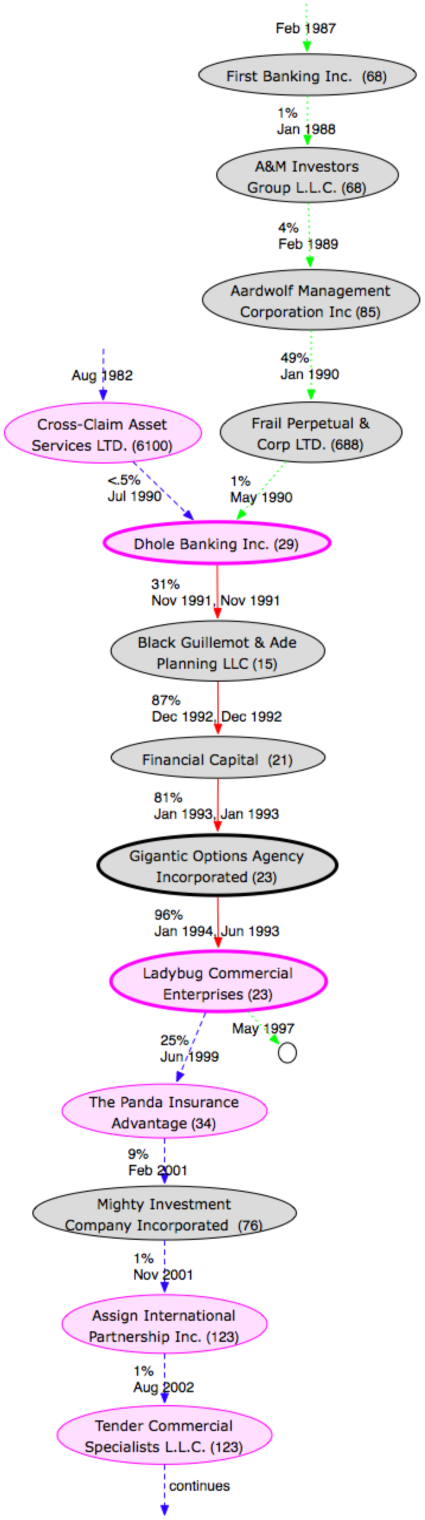

Figure 5. Example tribe ranked highly by JOBS but not by PROB. Firm names are fictitious.

6. 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 online 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. 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.

7. 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. This work was supported by ITR: the Central Intelligence Agency, the National Security Agency and National Science Foundation under NSF grant #IIS-0326249. 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.

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

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

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

[6] Fast, A., Friedland, L., Maier, M., Taylor, B., and Jensen, D. Data pre-processing for improved detection of securities fraud in relational domains. (Submitted to KDD '07.)

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

[8] 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.

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

[10] 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.

[11] 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.

[12] 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).

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] Neville, J., ş imşek, 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).

[18] Neville, J., ş imşek, 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).*

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

[20] 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.