The KOJAK Group Finder: Connecting the Dots via Integrated Knowledge-Based and Statistical Reasoning

Jafar Adibi, Hans Chalupsky, Eric Melz and Andre Valente

USC Information Sciences Institute
4676 Admiralty Way, Marina del Rey, CA 90292
Email: {adibi, hans, melz, valente}@isi.edu

Abstract

Link discovery is a new challenge in data mining whose primary concerns are to identify strong links and discover hidden relationships among entities and organizations based on low-level, incomplete and noisy evidence data. To address this challenge, we are developing a hybrid link discovery system called KOJAK that combines state-of-the-art knowledge representation and reasoning (KR&R) technology with statistical clustering and analysis techniques from the area of data mining. In this paper we report on the architecture and technology of its first fully completed module called the KOJAK Group Finder. The Group Finder is capable of finding hidden groups and group members in large evidence databases. Our group finding approach addresses a variety of important LD challenges, such as being able to exploit heterogeneous and structurally rich evidence, handling the connectivity curse, noise and corruption as well as the capability to scale up to very large, realistic data sets. The first version of the KOJAK Group Finder has been successfully tested and evaluated on a variety of synthetic datasets.

Introduction

The development of information technology that could aid law enforcement and intelligence organizations in their efforts to detect and prevent illegal and fraudulent activities as well as threats to national security has become an important topic for research and development. Since the amount of relevant information, tips, data and reports increases daily at a rapid pace, analyzing such data manually to its full potential has become impossible. Hence, new automated techniques are needed to take full advantage of all available information.

One of the central steps in supporting such analysis is link discovery (LD), which is a relatively new form of data mining. Link discovery can be viewed as the process of identifying complex, multi-relational patterns that indicate potentially illegal or threat activities in large amounts of data. More broadly, it also includes looking for not directly explainable connections that may indicate previously unknown but significant relationships such as new groups or capabilities (Senator, 2002) .

Link discovery presents a variety of difficult challenges. First, data ranges from highly unstructured sources such as reports, news stories, etc. to highly structured sources such as traditional relational databases. Unstructured sources need to be preprocessed first either manually or via natural language extraction methods before they can be used by LD methods. Second, data is complex, multi-relational and contains many mostly irrelevant connections (connectivity curse). Third, data is noisy, incomplete, corrupted and full of unaligned aliases. Finally, relevant data sources are heterogeneous, distributed and can be very high volume.

To address these challenges we are developing a hybrid link discovery system called KOJAK that combines state-of-the-art knowledge representation and reasoning (KR&R) technology with statistical techniques from the area of data mining. Using KR&R technology allows us to represent extracted evidence at very high fidelity, build and utilize high quality and reusable ontologies and domain theories, have a natural means to represent abstraction and meta-knowledge such as the interestingness of certain relations, and leverage sophisticated reasoning algorithms to uncover implicit semantic connections. Using data or knowledge mining technology allows us to uncover hidden relationships not explicitly represented in the data or findable by logical inference, for example, entities that seem to be strongly related based on statistical properties of their communication patterns.

The full KOJAK system contains a variety of experimental LD components such as an abductive, logic-based Pattern Finder to identify complex patterns of interest in the evidence and a Connection Finder to identify interesting and unusual entities and connections (Lin & Chalupsky 2003). In this paper we only report on the architecture and technology of its first fully completed module called the KOJAK Group Finder (GF). The Group Finder is capable of finding hidden groups and group members in large evidence databases. Our group finding approach addresses a variety of important LD challenges, such as being able to exploit heterogeneous and structurally rich evidence, handling the connectivity curse, noise and corruption as well as the capability to scale up to very large, realistic data sets. The first version of the KOJAK Group Finder has been successfully tested and evaluated on a variety of synthetic datasets.

The Group Detection Problem

A major problem in the area of link discovery is the discovery of hidden organizational structure such as groups and their members. There are of course many organizations and groups visible and detectable in real world data, but we are usually only interested in detecting certain types of groups such as organized crime rings, terrorist groups, etc. Group detection can be further broken down into (1) discovering hidden members of *known groups* (or group extension) and (2) identifying completely *unknown groups*.

A known group (e.g., a terrorist group such as the RAF) is identified by a given *name* and a set of known members. The problem then is to discover potential additional hidden members of such a group given evidence of communication events, business transactions, familial relationships, etc. For unknown groups neither *name* nor known members are available. All we know are certain suspicious individuals ("bad guys") in the database and their connection to certain events of interest. The main task here is to identify additional suspicious individuals and cluster them appropriately to hypothesize new real-world groups, e.g., a new money laundering ring. While our techniques address both questions, we believe group extension to be the more common and important problem.

Another important problem characteristic that influenced our solution approach concerns the data. Evidence available to law enforcement organizations is split into *primary* and *secondary* sources. Primary evidence is lower volume, high reliability, usually "owned" by the organization and can be searched and processed in arbitrary ways. Secondary evidence is usually not owned by the organization (e.g., might come from news articles or the Web), is higher volume, might only be searchable in restricted ways and might be associated with a cost (e.g., access might require a warrant). Our group detection approach needs to take these different characteristics into account to keep cost at a minimum and properly handle access restrictions to secondary data sources.

The KOJAK Group Finder

The KOJAK Group Finder is a hybrid logic-based/statistical LD component designed to solve group detection problems. It can answer the following questions:

- How likely is P a member of group G?
- How likely are P and Q members of the same group?
- How strongly connected are P and Q?

Figure 1 shows the general architecture. The system takes primary and secondary evidence (stored in relational databases) as input and produces group hypotheses (i.e., lists of group members) as output. The system works in four phases. First, a logic-based group seed generator analyzes the primary evidence and outputs a set of seed groups using deductive and abductive reasoning over a set of domain patterns and constraints. Second, an information-theoretic mutual information model finds likely new candidates for each group, producing an extended group. Third, the mutual information model is used to rank these likely members by how strongly connected they are to the seed members. Fourth, the ranked extended group is pruned using a threshold to produce the final output.

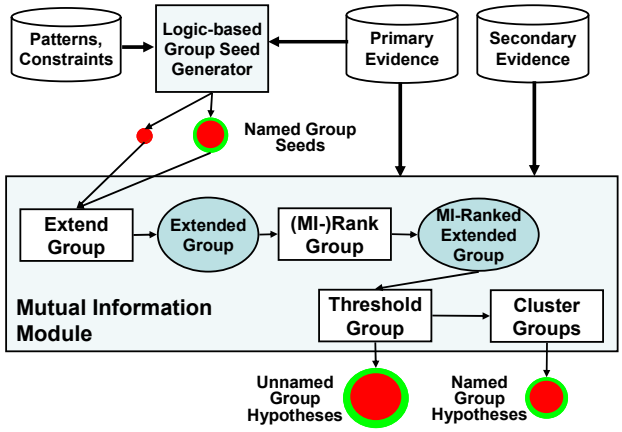


Figure 1: KOJAK Group Finder Architecture.

The processing for known and unknown groups is somewhat different at the beginning and end of the process. First, the seed generation for unknown groups is different, since there is less information available. Second, the generation of unknown groups involves an extra step because the extended groups need to be clustered to eliminate duplicates before the thresholding step.

The logic-based seed generation module is based upon the PowerLoom™ knowledge representation & reasoning system (PowerLoom, 2003). The mutual information module was implemented in the Matlab programming language. The modules are integrated by combining the C++ translation of PowerLoom and the C translation of the Matlab modules into a single program. Evidence databases are stored in MySQL and accessed from both Matlab and PowerLoom via ODBC. The primary and secondary evidence databases uses a very general evidence schema developed as part of DARPA's Evidence Extraction and Link Discovery (EELD) program (Senator, 2002) which should make it easy to transition to different domains.

The Need for a Hybrid Approach

Link discovery is a very challenging problem. It requires the successful exploitation of complex evidence that comes in many different types, is fragmented, incomplete, uncertain and very large-scale. LD requires reasoning with abstractions, e.g., that **brother-of** and **husband-of** are both subtypes of a **family-relation**, temporal and spatial reasoning, e.g., that cities are subregions of counties which are subregions of states, etc., common-sense type inferences, e.g., that if two people bought tickets for the same event, they probably were at one point in close spatial proximity in the same city, and constrained search, e.g., one might want to look more closely at people who joined a company around the same time a suspect joined. The knowledge

and ontologies needed for these types of inferences are very naturally modeled in a symbolic, logic-based approach as done in the logic-based seed generator of the KOJAK Group Finder. However, LD also needs detection and reasoning with statistical phenomena such as communication patterns, behavior similarity, etc., which requires cumulative analysis of evidence that cannot be done in logic but is most effectively done in specialized models such as our mutual information component. Such models, on the other hand, are not well-suited for the representation of complex domains and usually assume some data normalization and simplification. Given these characteristics of the problem, using a hybrid approach that combines the strengths of multiple paradigms is a natural choice. How these two approaches work together for the KOJAK Group Finder is described below.

Logic-Based Seed Generation

The first phase of the KOJAK group detection process is the generation of seed groups. Each seed group is intended to be a good hypothesis for one of the actual groups in the evidence data, even though the number of seed members known or inferable for it might be significantly less than its actual members. The reasons for using this logic-based, seeded approach are threefold. First, the information in primary and secondary evidence is incomplete and fragmented. By "connecting the dots" via logical inference we can extract information that is not explicitly stated and our statistical methods would not be able to infer. Second, because the MI model needs to analyze access-restricted secondary data, it needs good initial focus such as seed groups of "bad guys" in order to query the data selectively. The seeded approach therefore dramatically reduces data access cost as well as MI-processing time. Third, logical reasoning can apply constraints to the information available as well as rule out or merge certain group hypotheses.

To generate seed groups we use the PowerLoom KR&R system to scrub every piece of available membership information from primary evidence (which is smaller volume, less noisy and can be searched arbitrarily). Given the size of primary evidence data we are working with (O(10,000) individuals and O(100,000) assertions) we can simply load it directly from the EDB into PowerLoom using its database interface and a set of import axioms.

The process of finding seeds is different for known and unknown groups. For known groups, we start with a query to retrieve existing groups and their explicitly declared members. We then employ a number of logic rules to infer additional group members by connecting data that is available but disconnected. For example, in the synthetic datasets available to us members of threat groups participate in exploitation cases (meant to model threat events such as a terrorist attack). To find additional members of a group we can look for exploitations performed by a group that have additional participants not explicitly known to be members of the group. The PowerLoom definition below for the relation **memberAgentsByParticipation** formalizes this type of reasoning (**memberAgents** relates a group and its members; **deliberateActors** relates groups or people to an event):

```
(DEFRELATION memberAgentsByParticipation ((?g Group) (?p Person))
 :<= (AND (Group ?g)
          (Person ?p)
          (FAIL (memberAgents ?g ?p))
          (EXISTS (?c) (AND (ExploitationCase ?c)
                            (deliberateActors ?c ?g)
                            (deliberateActors ?c ?p)))))
```

For unknown groups, we use rules to look for patterns on events to find seeds. The basic idea is to find teams participating in threat events that no (known) group is known to be responsible for. Since people who participate in a threat event are part of a threat group, teams of people who are found to jointly participate in a threat event that cannot be attributed to a known group can be used as seeds for unknown groups. Note, however, that such teams may be subsets of one of the known groups or that two or more of the teams may be part of the same unknown group. For that reason, it is vital to use merging techniques later to combine teams (or their extended groups) if appropriate.

The logic module can also check constraints to help in the merging of hypotheses. For example, a strong hint that two groups may be the same is that their members participated in the same exploitation events. The rule below finds groups who participated in a given exploitation event indicating a potential duplicate group hypothesis if more than one group is found:

```
(DEFRELATION groupHasMemberWhoParticipatedInEvent
                ((?g Group) (?e VulnerabilityExploitationCase))
  :<= (AND (Group ?g)  (VulnerabilityExploitationCase ?e)
           (EXISTS ?p (AND (Person ?p)
                           (OR (memberAgents ?g ?p)
                               (memberAgentsByParticipation ?g ?p))
                           (deliberateActors ?e ?p)))))
```

The use of **memberAgentsByParticipation** shows that these rules not only encode complex queries but also interconnect to build a real domain model. There are about 50 complex rules of this type that are specific to group discovery. Even though the synthetic dataset used in our experiments was designed to be relatively poor in link types and attributes, the data is still quite complex. It contains 72 entity types (22 of which are actually instantiated) and 107 relations and attributes (25 of which are actually instantiated in the data). These entity and relation types are further organized by an ontology (developed by Cycorp) whose upward closure from the entity and relation types in the data contains a hierarchy of about 620 concepts (or classes) and 160 relations. Adding this to the O(100,000) assertions representing the evidence we have a fairly large and complex knowledge base to work with.

While the examples given above are specific to the synthetic group discovery domain, the approach is general and applicable to other areas. Evidence data will always be fragmented. Such fragmentation is usually easy to handle by a human analyst, but it can be a big obstacle for an automated system. Using a logic-based model of the

domain is a very powerful approach to overcome this problem and connect evidence fragments in useful ways.

Finding Strong Connections Via a Mutual Information Model

After exploiting the various explicit and implicit evidence fragments given in the EDB to generate a seed group, we try to identify additional members by looking for people that are strongly connected with one or more of the seed members. To find two strongly connected entities, we need to aggregate the many other known links between them and statistically contrast those with connections to other entities or the general population. This cannot be done via a logic-based approach and instead is achieved via an information-theoretic mutual information model.

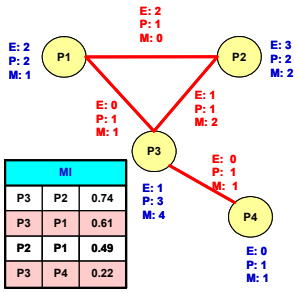
The mutual information model can identify entities strongly connected to a given entity or a set of entities and provide a ranked list based on connection strength. To do this it exploits data such as individuals sharing the same property (e.g., having the same address) or being involved in the same action (e.g., sending email to each other). Since such information is usually recorded by an observer we refer to it as evidence. Time is often also an important element of evidence and is also recorded in the EDB. Without loss of generality we only focus on individuals' actions in this paper, but not on their properties.

We transform the problem space into a graph in which each node represents an entity (such as a person) and each link between two entities represents the set of actions (e.g., emails, phone calls etc.) they are involved in. For each node we represent the set of its actions with a random variable, which can take values form the set of all possible actions. Figure 2 illustrates this concept. There are four people and three possible actions: sending *Email*, making a *Phone Call* and participating in a *Meeting*. When a person is not involved in any of the above-mentioned actions we indicate that with the empty action φ. For example, we can represent P_1's actions with the random variable X_1 which takes values from the set $\{E, P, M, \varphi\}$ at any given time.

Most individuals in the LD evidence space are connected to each other either directly or indirectly. For example, two people may eat at the same restaurant, drink coffee at the same cafe and take the same train to work every day without any strongly meaningful connection. On the other hand, three individuals may be strongly connected if they engage in atypical phone call patterns.

To address this problem we measure the *mutual information* (MI) between the random variables representing individuals' activities. MI is a measure of the dependence between two variables. If the two variables are independent, the MI between them is zero. If the two are strongly dependent, e.g., one is a function of another; the MI between them is large. We therefore believe that two individuals' mutual information is a good indicator whether they are in fact strongly connected to each other or not compared to the rest of the population.

There are other interpretations of MI, for example, as the stored information in one variable about another



MI		
P3	P2	0.74
P3	P1	0.61
P2	P1	0.49
P3	P4	0.22

Time	Action	P1	P2	P3	P4
1	P1 Email P2	E	E	Φ	Φ
2	P3 Phone Call P4	Φ	Φ	P	P
3	P2 Meeting P3	Φ	M	M	Φ
4	P1 Email P2	E	E	Φ	Φ
5	P2 Email P3	Φ	E	E	Φ
6	P1 Phone Call P3	P	Φ	P	Φ
7	P1 Phone Call P2	P	P	Φ	Φ
8	P2 Meeting P3	Φ	M	M	Φ
9	P3 Meeting P4	Φ	Φ	M	M
10	P1 Meeting P3	M	Φ	M	Φ
11	P2 Phone Call P3	Φ	P	P	Φ

Figure 2: MI Example. P1, P2, P3 and P4 represent people. E, P and M stand for *Email*, *Phone Call* and *Meeting* respectively. The table on the right shows activities among individuals and the table on the left shows the MI among them.

variable or the degree of predictability of the second variable by knowing the first. Clearly, all these interpretations are related to the same notion of dependence and correlation. The *correlation function* is another frequently used quantity to measure dependence. The correlation function is usually measured as a function of distance or time delay between two quantities. It has been shown that MI measures the more general (non-linear) dependence while the correlation function measures linear dependence (Li, 1990). Therefore, MI is the more accurate choice to measure dependence. One of the important characteristics of MI is that it does not need actual variables values to be computed, instead it only depends on the distribution of the two variables. In classical information theory (Shannon, 1948) MI between two random variables X and Y is defined as:

$$MI(X;Y) = \sum_x P(x) \sum_y P(y \mid x) . \log \left(P(y \mid x) \Big/ \sum_x P(x) P(y \mid x) \right)$$

where $P(x)$ is the $Prob(X = x)$, $P(y)$ is the $Prob(Y = y)$ and $P(y \mid x)$ stands for $Prob(Y = y \mid X = x)$. In addition, $MI(X;Y) = H(Y) - H(Y|X) = H(X) - H(X|Y)$, where the conditional entropy $H(X|Y)$ measures the average uncertainty that remains about X when Y is known (see (Adibi et al. 2004) for more details about the MI model).

Group Expansion via Mutual Information

Given that we can use the mutual information calculation to find strongly connected individuals, we can exploit this capability to expand the seed groups provided in phase 1 by the logic-based KR&R module. This expansion is done in the following steps:

(1) For each seed member in a seed group we retrieve all activities it participates in from primary and secondary data and add any new individuals found to the group. This step therefore expands the seed group graph by one level. Note, that we obey query restrictions for secondary data and only ask one focused query per seed member.

(2) Now we view the expanded group as the universe and compute MI for each connected pair in the graph.

(3) Next we look for individuals that either have high MI score with one of the seed members or with all seed

members when viewed as a single "super individual". Members whose score is below a certain (fairly lax) user-defined threshold are dropped from the list.
(4) In this step the MI engine repeats the whole procedure by expanding the expanded group from the previous step one more level and recalculates MI for the new graph. For known groups we stop here and pass the result to the final thresholding step.
(5) For unknown groups we usually have much smaller seed sets and therefore repeat the previous step one more time to achieve appropriately-sized group hypotheses.
The group expansion procedure is performed for each seed group generated by the KR&R module and generates an MI-ranked list of possible additional members for each seed group. This list is initially kept fairly inclusive and needs to undergo proper thresholding before it can be reported to a user or passed on to another LD component.

Threshold Selection and Thresholding

The result of the process described above is a list of extended groups where members are ranked by their mutual information scores. In order to produce and report a definite result on which members we believe are actually part of the group, we need to cut the ordered list at some threshold. The problem is how to set the threshold so that we get "good" (or even "optimal") recall and precision for a particular application scenario. We used an empirical method that selects a threshold for a dataset based on an empirical analysis of a number of groups in different types of datasets. This method is discussed further in the section describing the experimental results. The good news is that (1) our group detection process generates a very selective ranking (i.e., we reach high recall fairly early) and (2) in real-world situations a good ranking is often more important than picking the best possible cutoff, since human analysts might be willing to accept a certain number of false positives in order to maximize the number of true positives they are after.

Handling Noise Via a Noisy Channel model

So far we assumed that we are capable to observe all evidence accurately. However, such accuracy occurs rarely in real world databases. We therefore consider the following kinds of noise in the formulation of our model:
Observability (Negative Noise): This factor describes how much of the real data was observable. Not all relevant events that occur in the world will be observed or reported and might therefore not be known to LD components.
Corruption: This type of noise varies from typos to misspelled names all the way to intentional misinformation.

The negative noise phenomenon has been discussed extensively in the communication literature. We adopt the view of a classical noisy channel scenario where a sender transmits a piece of information to a receiver. The transmission goes through a channel with certain noise properties. In our domain we view the ground truth (GT)

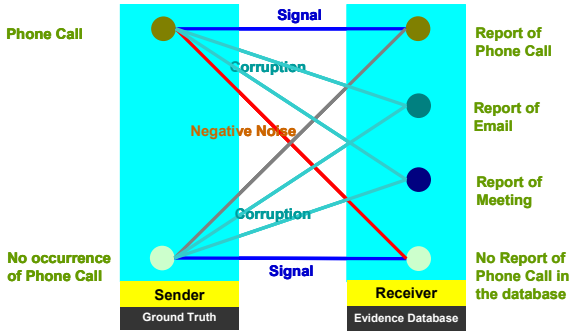


Figure 3: Noise model for a given "Phone Call"

as the "sender" and the evidence database (EDB) as the "receiver". While in a noiseless environment information is recorded in the EDB without error, in a noisy environment we have a noisy channel, which may alter every piece of evidence transmitted through it with some small probability *p(noise)*. For instance, negative noise occurs if there is a phone call in the ground truth but no record of it in the EDB. Corruption occurs, for example, if there is no phone call in the ground truth but a record indicating one in the EDB. The MI framework is a natural fit for such model. Figure 3 illustrates a noisy channel for a given phone call.

Complexity and Dataset Scale

Real-world evidence data sets can be very large and we have to make sure that our techniques scale appropriately. The largest synthetic datasets we have analyzed so far contained O(100,000) events and O(10,000) individuals. Running the KOJAK GF on such a dataset takes roughly 5 minutes for the logic-based seed generation and about 10-20 minutes to run the MI engine on a 2Ghz Pentium-IV desktop with 1Gb of RAM. Runtime for the MI engine varies depending on the overall connectivity of the data. While this is perfectly acceptable at the moment, we will eventually need to handle datasets that are at least two orders of magnitude larger, so let us look a bit closer at the architecture and algorithm complexity involved.
The complexity of the MI model is relatively low. The MI engine expands only a limited number of nodes in the problem space starting from the seed members of a group. How many individuals are considered depends on how deeply we grow the link graph to build an extended group. So far, one to two levels have been sufficient. Computing MI between two individuals is $O(N*M)$ where N is the average number of people connected to a given individual and M is the average number of links a person is involved in. Unless N and M grow significantly with larger datasets, the overall complexity is primarily dependent on the number of threat groups we are looking for.
To be able to handle such large datasets in the logic-based seed generation phase, we built a new database access layer into PowerLoom that allows us to easily and transparently map logic relations onto arbitrary database tables and views. By using these facilities we can keep smaller data portions such as the primary data in main

memory for fast access and processing, while keeping potentially very large secondary data sets in an RDBMS from where we page in relevant portions on demand. Particular attention was paid to be able to offload large join processing to the RDBMS wherever possible to avoid doing it inefficiently tuple-by-tuple in PowerLoom. This gives us an architecture where we use a traditional RDBMS for storage and access to very large datasets but enrich it with a deductive layer that allows us to formulate more complex queries where necessary. The complexity of the resulting system depends heavily on the nature of the queries and domain rules used which so far has proven to be manageable. For example, the current system uses an ontology with about 800 concept and relation definitions and about 50 complex, non-taxonomic rules that link evidence fragments without any performance problems.

Experimental Set-Up

We have applied the KOJAK Group Finder to a wide variety of synthetic data. Access to real world databases has been a main concern in AI, machine learning and data mining communities in the past. The LD community is not an exception in this matter. In particular, since the LD goal is to relate people, place and entities, it triggers privacy concerns. The balance between privacy concerns and the need to explore large volumes of data for LD is a difficult problem. These issues motivate employing synthetic data for performance evaluation of LD techniques.

Synthetic Data

For the purpose of evaluating and validating our techniques, we tested them on synthetic datasets developed by Information Extraction & Transport, Inc. within the EELD Program (Silk 2003, Schrag 2003). These synthetic datasets were created by running a simulation of an artificial world. The main focus in designing the world was to produce datasets with large amounts of relationships between agents as opposed to complex domains with a large number of entity properties.

From the point of view of group detection, the artificial world consists of *individuals* that belong to *groups*. Groups can be *threat groups* (that cause *threat events*) or *non-threat-groups.* Targets can be exploited (in threat and non-threat ways) using specific combinations of resources and capabilities; each such combination is called a *mode.* Individuals may have any number of capabilities or resources, belong to any number of groups, and participate in any number of exploitations at the same time. Individuals are *threat* individuals or *non-threat* individuals. Every threat individual belongs to at least one threat group. Non-threat individuals belong only to non-threat groups. Threat groups have only threat individuals as members. Threat individuals can belong to non-threat groups as well. A group will have at least one member qualified for any capability required by any of its modes. Non-threat groups carry out only non-threat modes.

The evidence available in the dataset for our analysis consists of two main types of information:

Number of entities	10,000
Number of Links	100,000
Number of Distinct Threat Pattern	20
Lowest Signal to clutter ratio	0.3(-5 db)
Lowest Signal to Noise Ratio	.008(-21 db)
Observability	50%-100%
Corruption of Evidence	0-25%

Table 1: Synthetic Data Characteristics

(1) *Individual and group information*. The existence of most individuals and some of the groups is available directly in the evidence. The groups available in the evidence are known or named groups discussed earlier.

(2) *Activities from individuals*. Individuals participate in activities related to resources, capabilities and events. Much like in the real world, information about those activities is not available directly, but rather indirectly as transactions (e.g., phone calls or email messages).

Synthetic Data Characteristics

One of the key advantages of using a simulated world is that we are able to test our system against a wide range of datasets. In other words, we are able to create datasets with (almost) arbitrary characteristics, and therefore better understand the potential and limitations of our techniques.

Some of the features used in defining the datasets are in Table 1. The values displayed are typical for the datasets we used in our evaluation; each dataset employs different values for each of these features. Of particular interest are observability (how much of the artificial world information is available as evidence), corruption (how much of the evidence is changed before being reported) and clutter (how much irrelevant information that is similar to the information being sought is added to the evidence).

Evaluation Metrics

The quality of groups we find can be measured with traditional precision and recall metrics defined as follows: Given a proposed group G with g members which matches an answer group A with a members, and given that of the g proposed members only c are correct, precision $P=c/g$ and recall $R=c/a$. Another metric that helps us analyze precision and recall in aggregate is the F-measure:

$$F = -\frac{(b^2+1)PR}{b^2P+R}$$

The F-measure both requires and allows us to specify the desired trade-off between precision and recall through the b variable. A value of b=1 indicates that precision and recall are equally important; b = 2 means that recall is twice as important as precision, etc. That is, using the F-measure allows users of our module to specify their own desired trade-offs in terms of b.

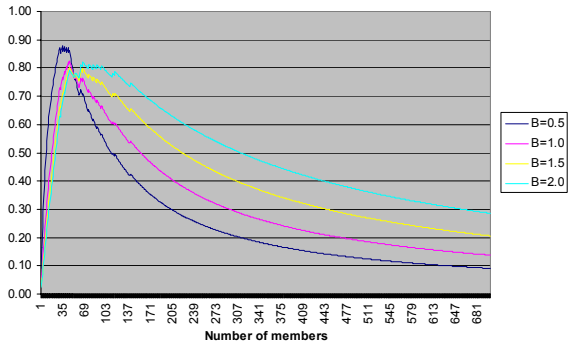


Figure 4: *F*-measure curves for different thresholds for a typical group.

Experimental Results

Overall empirical properties

Figure 4 shows a typical set of *F*-measure curves for different thresholds. An important property is that our *F*-measure curves have maximums (and thus optimums). Notice also that *F*-measure curves for higher values of *b* have wider "peaks", which means they are more "forgiving" in threshold selection (a given variation of threshold provokes a smaller variation in F-measure.)

Threshold Analysis

Focusing on the *F*-measure, we defined an empirical model that allowed us to predict good threshold values for a given type of dataset. Datasets vary in many dimensions, in particular on their levels of observability, corruption, and clutter. Our goal was to define a model parametric on these dataset dimensions.

One key initial step is to define the base for the model. Possible bases include the average size of the groups we are looking for (if sufficiently known), the size of extended group and the size of the seed group. Our empirical analysis indicated that the best alternative is to use the size of extended group as a basis for defining the threshold. We found that the ratio between the real size of the group we would be looking for and the size of the extended group we created as a hypothesis varies little and is usually around 11%-14%. Another advantage is that this measure is organic to the mutual information model, that is, no additional information is needed.

The empirical model consists of defining one specific threshold (as a percentage of the extended group size) for each type of dataset. We used thirteen types of datasets that employed combinations of different values for the parameters in Table 1. We then analyzed the *F*-measure curves to find optimums for each *b*-value (i.e., trade-off between precision and recall) and type of dataset. For example, for a *b* of 1, we predicted a threshold of 8% for a baseline dataset, 6% for a dataset with more clutter, 9% for a dataset with low observability and 3% for a dataset with both additional clutter and low observability. These thresholds are then used to predict the best threshold for a new dataset of a particular type.

		Logic Module		KOJAK Group Finder				
Data set	Number of Groups	Avg. Precision	Avg. Recall	Avg. Precision	Precision Variance	Avg. Recall	Recall Variance	Avg. F-Measure (b=1.5)
Plain	14	1	0.53	0.81	0.005	0.87	0.010	0.85
High clutter	11	1	0.53	0.59	0.010	0.86	0.014	0.74
Low observability	16	1	0.52	0.70	0.004	0.72	0.026	0.71
Both	19	1	0.50	0.88	0.005	0.66	0.011	0.75

Table 2: Scores for applying the KOJAK Group Finder to datasets of increasing complexity (known groups only).

Results

We have applied KOJAK to 26 datasets of varying complexity and characteristics. Table 2 shows some sample metrics for four datasets. Since there are many groups in each dataset we provide mean and variance values for precision and recall among all groups in a dataset. The average *F*-measure for known groups varies between 0.71 and 0.85. Note that the differences in the properties of the datasets cause the best *F*-measure to be obtained with different recall and precision values. This shows that "harder" datasets, where precision drops more steeply require lower thresholds that yield lower recalls and higher precision values. A more detailed analysis with ROC curves is presented in (Adibi et al. 2004).

Table 2 also compares the KOJAK results against a baseline of using only the logic module. The results show that the logic module is very accurate (precision = 1), meaning all members found are provably correct. However, since the evidence is incomplete the logic module achieves a maximum recall of about 50%.

We also evaluated our threshold prediction model. We found that the average *F*-measure for these datasets compares to the optimum *F*-measure obtained by using the best possible threshold for each group would result only in a difference of around 6%. In other words, the threshold model only "misses" 6% of whatever was available in the extended groups.

Related Work

Link discovery (LD) can be distinguished from other techniques that attempt to infer the structure of data, such as classification and outlier detection. Classification and clustering approaches such as that of Getoor et al. (2001) try to maximize individual similarity within classes and minimize individual similarity between classes. In contrast, LD focuses on detecting groups with strongly connected entities that are not necessarily similar. Outlier detection methods attempt to find abnormal individuals. LD, on the other hand, identifies important individuals based on networks of relationships. Additionally, outlier techniques require large amounts of data including normal and abnormal cases, and positive and negative noise. This is inappropriate for LD applications that need to detect threats with few or no available prior cases.

Mutual information has also been used in other domains such as finding functional genomic clusters in RNA expression data and measuring the agreement of object models for image processing (Butte, 2000).

Our work can be distinguished from other group detection approaches such as Gibson, (1998) and Ng, (2001) by three major characteristics. First, GF is unique since it is based on a hybrid model of semantic KR&R and statistical inference. There are very few approaches that use semantic information. Second, in our approach each type of relation (link) is valuable and treated differently, in contrast to work in fields such as Web analysis and social networks. Third, with our technique, multiple paths between individuals or groups (direct or indirect) imply a strong connection which is different from techniques which focus on finding chains of entities.

The work closest to our own is that of Jeremy Kubica et al. (Kubica, 2002; Kubica, 2003) that uses a probabilistic model of link generation based on group membership. The parameters of the model are learned via a maximum likelihood search that finds a Gannt Chart that best explains the observed evolution of group membership. The approach has a strong probabilistic foundation that makes it robust in the face of very low signal-to- noise ratios.

Another recent approach to the LD problem is the use of probabilistic models (Cohn, 2001; Friedman, 1999; Getoor, 2001). Kubica et al. (2001) present a model of link generation where links are generated from a single underlying group and then have noise added. These models differ significantly from ours since we do not assume a generative model of group formation, but rather probabilistically determine each entity's membership.

Conclusion and Future Work

In this paper we introduced the KOJAK Group Finder (GF) as a hybrid model of logic-based and statistical reasoning. GF is capable of finding potential groups and group members in large evidence data sets. It uses a logic-based model to generate group seeds and a multi-relational mutual information model to compute link strength between individuals and group seeds. Noise and corruption are handled via a noisy channel model. Our GF framework is scalable and robust, and exhibits graceful degradation in the presence of increased data access cost and decreased relational information. The Group Finder is best-suited for problems where some initial information or group structure is available (e.g. finding hidden members of existing groups vs. detecting completely new groups) which is a common case in many real world applications. Group detection is useful for law enforcement, fraud detection, homeland security, business intelligence as well as analysis of social groups such as Web communities.

There are several lines of ongoing and future work, such as, determining group leaders by measuring their entropy, use of temporal information for more focused access to relevant information as well as employing sampling and data streaming techniques to deal with very large datasets.

Acknowledgments. This research was sponsored by the Defense Advance Research Projects Agency and Air Force Research Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-01-2-0583. 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 DARPA, AFRL or the U.S. Government.

References

Adibi, J., Valente, A., Chalupsky, H. & Melz, E. (2004). Group detection via a mutual information model. Submitted to *KDD 2004*.

Butte, A. & Kohane, I. (2000). Mutual information relevance networks: Functional genomic clustering using pairwise entropy measurements. *Pacific Symposium on Biocomputing*. Honolulu, Hawaii.

Cohn, D. & Hofmann, T. (2001). The missing link: a probabilistic model of document content and hypertext connectivity. *Advances in Neural Information Processing Systems* 13: 430–436.

Friedman, N., Getoor, L., Koller, D. & Pfeffer, A. (1999). Learning probabilistic relational models. *IJCAI 1999*, San Francisco, Morgan Kaufmann Publishers.

Getoor, L., Segal, E., Taskar, B., & Koller, D. (2001). Probabilistic models of text and link structure for hypertext classification. *IJCAI 2001 Workshop on Text Learning: Beyond Supervision*. Seattle, Washington.

Gibson, D., Kleinberg, J. & Raghavan, P. (1998). Inferring Web communities from link topology. In *Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia*. New York, ACM Press.

Kubica, J., Moore, A., Cohn, D. & Schneider, J. (2003). Finding underlying connections: a fast method for link analysis and collaboration queries. *International Conference on Machine Learning (ICML)*.

Kubica, J., Moore, A., Schneider, J. & Yang, Y. (2002). Stochastic link and group detection. *Eighteenth National Conference on Artificial Intelligence (AAAI)*.

Li, W. (1990). Mutual information functions versus correlation functions. *Journal of Statistical Physics* 60: 823-837.

Lin, S. & Chalupsky, H.. Using unsupervised link discovery methods to find interesting facts and connections in a bibliography dataset. *SIGKDD Explorations*, 5(2): 173-178, December 2003

Ng, A., Zheng, A. & Jordan, M. (2001). Link analysis, eigenvectors and stability. *IJCAI 2001*.

PowerLoom (2003). www.isi.edu/isd/LOOM/PowerLoom.

Schrag, R. et. al. (2003). EELD Y2 LD-PL Performance Evaluation, Information Extraction and Transport, Inc.

Senator, T. (2002). Evidence Extraction and Link Discovery, *DARPA Tech 2002*.

Shannon, C. (1948). A Mathematical Theory of Communication. *Bell System Tech. Journal* 27: 379-423.

Silk, B. & Bergert, B. (2003). EELD Evidence Database Description, Information Extraction and Transport, Inc.