Discovering and Explaining Abnormal Nodes in Semantic Graphs

Shou-de Lin and Hans Chalupsky

Abstract—An important problem in the area of homeland security is to identify abnormal or suspicious entities in large data sets. Although there are methods from data mining and social network analysis focusing on finding patterns or central nodes from networks or numerical data sets, there has been little work aimed at discovering abnormal instances in large complex semantic graphs, whose nodes are richly connected with many different types of links. In this paper, we describe a novel unsupervised framework to identify such instances. Besides discovering abnormal instances, we believe that to complete the process, a system has to also provide users with understandable explanations for its findings. Therefore, in the second part of the paper, we describe an explanation mechanism to automatically generate human-understandable explanations for the discovered results. To evaluate our discovery and explanation systems, we perform experiments on several different semantic graphs. The results show that our discovery system outperforms state-of-the-art unsupervised network algorithms used to analyze the 9/11 terrorist network and other graph-based outlier detection algorithms by a significant margin. Additionally, the human study we conducted demonstrates that our explanation system, which provides natural language explanations for the system's findings, allowed human subjects to perform complex data analysis in a much more efficient and accurate manner.

Index Terms—Anomaly detection, data mining, knowledge and data engineering tools and techniques, semantic graphs.

1 Introduction

Being able to automatically identify abnormal individuals such as potential terrorists or criminals in large amounts of relational data is an important knowledge discovery task for homeland security and law enforcement agencies. Intelligence analysts and crime investigators are often overwhelmed by the large data volumes they are faced with and need automated systems to help them search through this data for potential suspects and other relevant information. Although identifying such individuals is an important step, a useful system should also be able to generate human-understandable explanations revealing why a particular individual was chosen. For example, an explanation such as *"X is suspicious, because it is the only Mafiya group in the data set that has a member who ordered a contract murder"*[1] might help an analyst assess the relevance and validity of a particular result and guide follow-up investigations into the data to further confirm or deny the suspiciousness of an individual.

The kind of data we are focusing on throughout this paper are *semantic graphs*, which have become an important tool for analysts in the intelligence and law enforcement community [1], [2], [3]. A semantic graph is a graph where nodes represent objects of different types (for example, persons, papers, organizations, etc.) and links represent binary relationships between those objects (for example, friend, citation, etc.). We focus on semantic graphs with multiple different types of relations, which we call multi-relational networks (MRNs). Having multiple relationship types in the data is important, since those carry different kinds of semantic information, which will allow us to automatically compare and contrast entities connected by them. MRNs are a powerful yet simple representation mechanism to describe complex relationships and connections between individuals. For example, a bibliography network such as the one shown in Fig. 1 is an MRN that represents authors, papers, journals, organizations, etc., as nodes and their various relationships such as authorship, citation, affiliation, etc., as links. The Web is an MRN if we distinguish, for example, incoming, outgoing, and e-mail links. Data stored in relational databases or the ground facts in a knowledge base can usually also be easily described via MRNs (some transformations might be necessary, for example, to map n-ary relations onto binary links—see Section 2.1.1).

Why do we focus on finding *abnormal* instances in MRNs? Our assumption is that in the general population, criminal behavior is the exception, and it is reasonable to infer that people who look typical (that is, they have many similar peers) are not likely to be malicious. This is justified if we assume the data to be investigated to contain primarily innocent persons, whose behavior has a higher chance to be similar to somebody else's behavior in the same data set. Furthermore, it is reasonable to assume that malicious behaviors are often carried out by unusual methods to avoid detection. For example, criminals might try to disguise themselves by playing some pretend business role but are likely to get things wrong, since they are not real market actors. Such behavior is more likely to create unusual evidence, and therefore, a node with an

[1] This is output generated automatically by our system when applied to a synthetic data set in the domain of Russian organized crime.

- *S.-d. Lin is with the Computer Science and Information Engineering Department, National Taiwan University, #1 Roosevelt Road, Sec. 4, Taipei, Taiwan. E-mail: sdlin@csie.ntu.edu.tw.*
- *H. Chalupsky is with the USC Information Sciences Institute, 4676 Admiralty Way, Marina del Rey, CA 90292. E-mail: hans@isi.edu.*

Manuscript received 4 May 2006; revised 30 Apr. 2007; accepted 31 Aug. 2007; published online 20 Sept. 2007.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDESI-0229-0506.
Digital Object Identifier no. 10.1109/TKDE.2007.190691.

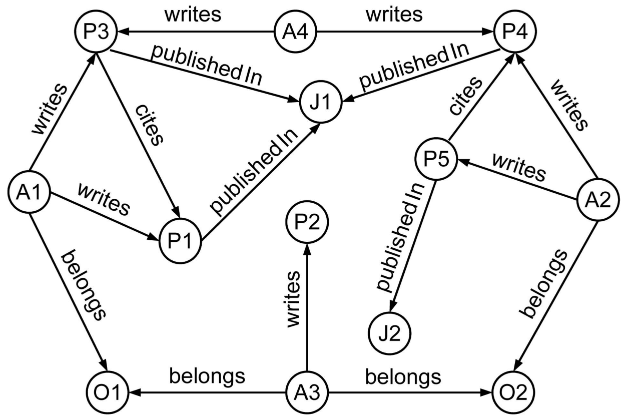

Fig. 1. A MRN for a bibliography domain. The capital letter of each node represents its type: A(author), P(paper), O(organization), and J(journal).

abnormal evidence profile has a higher chance to be suspicious compared with one that has many similar peers. Nevertheless, abnormality does not necessarily imply suspiciousness and could be due to other factors such as missing evidence, data entry errors, innocent unusual behavior, etc.

If one accepts the claim that abnormality might be a good indicator for criminal or malicious entities, the next important question to ask is what kind of system is best suited for identifying abnormal instances in semantic graphs: a rule-based system [4], a supervised learning system [3], or an unsupervised system? Rule-based systems generally perform some type of pattern match based on manually created rules to identify abnormal instances. Supervised learning systems take a set of manually labeled example individuals as inputs and then try to learn patterns or other descriptions to classify new individuals. The advantage of these approaches is that they can achieve relatively high precision due to the manual crafting of rules or selection of training examples. A severe disadvantage, however, is that they are domain dependent, expensive to create, and, most seriously, very sensitive to human bias. In other words, the only abnormal instances such a system can find are those that match the rules of a rule-based system or those similar to the training examples of a supervised system. However, malicious individuals constantly adapt their behavior to avoid capture; therefore, for purposes such as homeland security, we feel that it is important to have an unsupervised system that can identify truly novel results and avoid the biases described above. Another important advantage of an unsupervised system is that it can be easily adapted to a new domain, without having to produce a completely new set of rules or training examples, which could be very expensive and time consuming.

Traditional unsupervised network algorithms such as PageRank, random walks, and social network analysis (SNA) (for example, centrality and position analysis) have been applied for homeland security and crime analysis with a certain level of success [5], [6]; however, they all suffer a serious drawback that the *semantics* of links are not considered. That is, they are best suited for "single-relational networks" where no link-type information is available. For the MRNs focused on in this paper, we describe a novel unsupervised method that does exploit the additional information coming from the different types of links, and our experiments show that our method outperforms the above algorithms by a significant margin.

A major concern with automated information awareness systems is the generation of false positives where innocent individuals are mistakenly identified as suspicious. As pointed out in [7] and [8], false positives are a very serious issue for any homeland security system, since even a system with almost perfect 99 percent accuracy can result in the mislabeling of millions of innocent individuals when applied to a large population. Although an unsupervised discovery system such as ours has the potential of being able to identify additional abnormal individuals whose characteristics were previously unknown (which improves recall), it runs an even higher risk of increasing the number of false positives, since there is no preexisting knowledge to learn from.

Achieving 99 percent precision without completely sacrificing recall is already an extremely difficult goal for any automated data analysis system, yet it would still not solve the false-positive problem. For this reason, we propose an indirect solution to address the issue of verification, which we call *explanation-based discovery*. An explanation-based discovery system like ours can not only identify suspicious persons but also generate human-understandable explanations (for example, in natural language) to show the analyst why they were chosen by the system. This allows the analyst to better judge the validity of the results and also provides a direction for further investigation. For example, an explanation such as "this person is suspicious, because he has never left his house for the last five years" might suggest that he is not likely to be a person to pay attention to in terms of hijacking a plane. On the other hand, depending on the context, such an explanation might prompt additional investigation to find out why he has not left the home (maybe he was bedridden or trying to hide).

Although we focus on false positives here, false negatives are of course a very important problem as well, and any automated system or human analyst will always have to trade off the cost of a false negative (for example, missing a planned attack) with the cost of too many false positives such as incriminating innocent individuals or overlooking something due to being overwhelmed with too many useless hits.

The main contributions of this paper are the following: First, we provide an unsupervised framework to model the *semantics* of nodes in large complex semantic graphs into what we call *semantic profiles*. Second, we propose to detect potentially suspicious nodes as those with abnormal semantic profiles, and our experiments show that our method significantly outperforms other unsupervised network algorithms that use heuristics such as node centrality, importance, or cluster-based outliers. Third, we describe a novel explanation mechanism that facilitates verification of the discovered results by generating human-understandable natural language explanations describing the unique aspects of these nodes. The human study we conducted shows that the human subjects using these explanations were able to solve a complex data analysis task much faster and with much higher accuracy and confidence than without them.

2 Modeling Abnormal Nodes in Semantic Graphs

The general idea is that a node is abnormal and suspicious if it carries *abnormal* or *unique* semantics in the network. To realize this concept in an automated system, we generate a *semantic profile* for each node by summarizing the graph structure surrounding it based on the different types of links and nodes connected to the node within a certain distance. We then identify abnormal nodes as those that possess abnormal semantic profiles.

Our observation is that the "semantics" of a semantic graph is primarily carried by the *structure of labeled links and nodes* in the network. Therefore, our system needs to utilize this structure, as well as the label information, to model the semantics of nodes in a way that is easy to compare and contrast. To realize this idea, in our framework, a node is modeled by a summarization of its surrounding labeled network structure based on sequences of labels (that is, paths), together with some statistical dependency measures, between these paths and the node.

We decompose the whole process into two stages. The first stage is structure modeling or feature selection. In this stage, the system *automatically* selects a set of features to represent the surrounding network structure of nodes. The second stage is a dependency computation or feature value generation stage. For this stage, we design a set of different models to compute the dependency between the features and the nodes in the MRN. A *semantic profile* of a node is then constructed as a standard feature vector based on the automatically selected features and the computed dependency values.

2.1 Feature Selection Stage

To explain the feature selection stage, we start with a motivating example derived from the bibliography network shown in Fig. 1. Let us assume that our goal is to find out which author among $A1$, $A2$, and $A3$ is the most abnormal (that is, plays different roles compared with others). After examining these nodes based on their connections to others, we can conclude several things about each of them:

- $A1$ published two journal papers ($P1$ and $P3$), and one of them cites the other. $A1$ belongs to organization $O1$, has a colleague $A3$, and coauthored one paper with $A4$.
- $A2$ published two journal papers ($P4$ and $P5$), and one of them cites the other. $A2$ belongs to organization $O2$, has a colleague $A3$, and coauthored one paper with $A4$.
- $A3$ published one paper $P2$ (no citation). $A3$ belongs to two organizations $O1$ and $O2$, and has two colleagues $A1$ and $A2$.

Based on the above description, it is not very hard to recognize that $A3$ has the most abnormal semantics, because its behavior, as captured by the network, is very different from the other two. The example shows that it is possible for humans to successfully identify abnormal nodes in such a network if we can somehow summarize their roles or the surrounding structure. However, since our ultimate goal is to design an automatic mechanism to perform this comparison, we need a systematic method to model the semantics or roles of the nodes to make them comparable and contrastable.

To illustrate this idea, we start by systematically listing the two-step paths starting from a node with their corresponding natural language interpretation. For example, below is the interpretation for $A1$:

- Path 1: $A1$—*writes*—$P1$—*publishedIn*—$J1$ ($A1$ writes $P1$, which is published in $J1$)
- Path 2: $A1$—*writes*—$P3$—*publishedIn*—$J1$ ($A1$ writes $P3$, which is published in $J1$)
- Path 3: $A1$—*writes*—$P1$—$cites^{-1}$—$P3$ ($A1$ writes $P1$, which is cited by $P3$; note that $relation^{-1}$ stands for the inverse of $relation$)
- Path 4: $A1$—*writes*—$P3$—*cites*—$P1$ ($A1$ writes $P3$, which cites $P1$)
- Path 5: $A1$—*writes*—$P3$—$writes^{-1}$—$A4$ ($A1$ writes $P3$, which is written by $A4$)
- Path 6: $A1$—*belongs*—$O1$—$belongs^{-1}$—$A3$ ($A1$ belongs to $O1$, which is the organization of $A3$)

By combining and condensing the information provided by those paths, one can come up with descriptions of nodes similar to the ones we have provided previously. For example, in the case of $A1$, Paths 1 and 2 tell us that $A1$ wrote two journal papers. The next two paths tell us that one paper cites the other. Path 5 reveals that $A1$ coauthored with $A4$, whereas the last path shows that $A1$ belongs to organization $O1$ and has a colleague.

This observation motivates a key idea of our approach for using these paths to capture the semantics of nodes. Another justification is that each path in a network can be translated into a standard logical notation by representing nodes as constants and links via binary predicates. Those predicates contain meanings and can be translated into the natural language, as we did for the above paths. For example, in Fig. 1, the path $A1$—*writes*—$P3$—*cites*—$P1$ can be represented as the conjunction $writes(A1, P3) \wedge cites(P3, P1)$. This logical expression partly characterizes the meaning of the nodes $A1$, $P1$, and $P3$. It only partially characterizes their meaning, since there are many other paths (or logical expressions) that also involve these nodes. In our view, it is the combination of all paths a node participates in that determines its semantics. This differs from standard denotational semantics in which a node's interpretation is the object it denotes and is more closely related to the formal semantics for semantic networks where the meaning of a node is determined by the whole network [9].

Given these observations, one naive approach to capture a semantic profile of a node is by treating all paths in the network as *binary features*, assigning true to the paths the given node participates in and false to the ones it does not. By doing this, we have essentially transformed a semantic graph into a propositional representation, where each node is a point in a high-dimensional space with attributes identifying its semantics based on its role in the network.

Although this describes the basic idea for representing the semantic profiles of nodes, there are still some problems that we must address. The first is that treating each path as a different feature generates an overfitting problem. Since each path is unique, the only nodes sharing a particular path feature would be those participating in the path, which would make these profiles useless for comparing nodes inside the path with the ones outside of it. A second problem relates to time and space complexity: A large semantic graph can easily contain millions of paths, and computation in such a high-dimensional space would be difficult and costly.

These issues motivate the search for a more condensed feature set that still can adequately capture the role semantics of instances. We do this by defining equivalence classes between different paths that we call *path types* and then use these path types instead of individual paths as features. Whether two individual paths are considered to be of the same type will depend on one of several similarity measures (which we call *metaconstraints*) we can choose. For example, we can view a set of paths as equivalent (or similar or of the same type) if they use the same order of relations. This view would consider the following two paths as equivalent:

$$cites(P2, P1) \wedge publishedIn(P1, J1),$$
$$cites(P2, P3) \wedge publishedIn(P3, J1).$$

Given these generalization strategies, the next question becomes how we can generate a meaningful representative set of path types? One way is to apply a *variable relaxation* approach. Taking the path $cites(P2, P1) \wedge publishedIn(P1, J1)$ as an example, we find there are five ground elements in this path: $cites$, $P1$, $P2$, $publishedIn$, and $J1$. If we relax one of its elements, say, $J1$, to a variable $?X$, then we get a new relaxed path type $cites(P2, P1) \wedge publishedIn(P1, ?X)$, which now represents a more general concept: "paper $P2$ cites paper $P1$ that is published in some journal." Relaxing further, we could also generalize a link such as $publishedIn$, which would give us $cites(P2, P1) \wedge ?Y(P1, ?X)$ or "paper $P2$ cites paper $P1$ that has something to do with something." In fact, we can generalize any combination of nodes or links in a path to arrive at a more general path type. These path types still convey meaning but do so at a more abstract level. This makes them more useful as features to compare and contrast different instances or nodes. In Section 2.3, we will discuss a small set of metaconstraints that allows our system to generate path-type features fully automatically.

2.1.1 Formal Definitions and Notation

Let us define these concepts more precisely:

Definition 1. *An MRN $M(V, E, L)$ is a directed labeled graph, where V is a finite set of nodes, L is a finite set of labels, and $E \subseteq V \times L \times V$ is a finite set of edges. Given a triple representing an edge, the functions* source, label, *and* target *map it onto its start vertex, label, and end vertex, respectively. The function $types(V) \rightarrow \{\{t_1, \dots, t_k\}, t_i \in L, k \geq 1\}$ maps each vertex onto its set of type labels.*

Note that edges are restricted to be binary, but any n-ary relation can be represented by introducing an additional element reifying the relationship and n binary edges to represent each argument. In fact, the data sets described in Section 4 represent various n-ary relationships such as a murder event that has a perpetrator, a victim, and a location as a set of binary edges associated with an event object.

Definition 2. *Let $M(V, E, L)$ be an MRN. The* inverse edge set *E^{-1} is the set of all edges (v_1, l^{-1}, v_2) such that $(v_2, l, v_1) \in E$.*

When analyzing an MRN, we will usually consider both its forward and inverse edge sets. Note that this is not the same as treating it as an undirected graph, since forward and inverse edges will participate in different path types.

Definition 3. *Let $M(V, E, L)$ be an MRN. A path p in M is a sequence of edges $(e_1, e_2, \dots, e_n)$, $n \geq 1$, such that each $e_i \in E$ and $target(e_i) = source(e_{i+1})$.*

Definition 4. *Let $M(V, E, L)$ be an MRN and P be a set of paths in M. A set of* path types *$PT(P)$ is a disjoint partition $\{pt_1, pt_2, \dots, pt_k\}$, $k \geq 1$, of P such that each pt_i is a set of paths $\{p_{i1}, \dots, p_{im}\}$, $p_{ij} \in P$, that are considered to be equivalent (that is, each pt_i is an equivalence class).*

Definition 5. *Let $M(V, E, L)$ be an MRN and P be a set of paths in M. A metaconstraint mc is any function $mc(P) \rightarrow P'$ such that $P' \subseteq P$.*

Definition 6. *Let $M(V, E, L)$ be an MRN, P be a set of paths in M, and mc be a metaconstraint. Then, the set of* path types *$PT_{mc}(P)$ defined by mc is the disjoint partition $\{pt_1, pt_2, \dots, pt_k\}$, $k \geq 1$, of P such that for each $pt_i = \{p_{i1}, \dots, p_{im}\}$, $p_{ij} \in P$, $mc(p_{i1}) = \dots = mc(p_{im})$, and there is no path $p \in P - pt_i$ such that $mc(p) = mc(p_{i1})$. That is, a metaconstraint can be viewed as mapping each path onto its representative in a path-type equivalence class.*

2.2 Feature Value Computation

A major advantage of using path types is that we do not limit ourselves to only binary features (that is, whether a node does or does not participate in a path). Instead, we can apply statistical methods to determine the dependence between a path type and a node and use it as the corresponding feature value.

2.2.1 Performing Random Experiments on a Multirelational Network

Measures such as conditional probability and mutual information (MI) are commonly used to compute statistical dependence. These measures rely on the existence of nondeterministic dependency between random variables. However, a normal MRN, unlike a Bayesian network or a general belief network, is a deterministic graph structure instead of a probabilistic one. It represents the relationships between nodes, and normally, there are no probabilities associated with these nodes and links. As a result, questions such as "what is the MI between a node x and a path type pt" are ill defined, because not only is there no uncertainty associated with x and pt, but they are also not random variables.

To apply statistical dependency measures to a deterministic MRN in order to compute the correlation between nodes and path types, we introduce a set of *random experiments* carried out on the MRN. Based on the results of these experiments, we can create certain random variables and use them to measure the dependency between a node and a path type. To elaborate this idea, we introduce two random experiments to select a path in an MRN:

- **Random Experiment 1 (RE1).** Randomly pick a path from the MRN. In this case, the chance of each path to be selected is $1/|Path|$, where $|Path|$ is the total number of paths in the MRN.

- **Random Experiment 2 (RE2).** First randomly pick a node in the MRN and then randomly pick a path that starts from the selected node.

Any of these random experiments produces a single path as the output. However, the probability of each path to be selected varies, depending on the selection policy. Based on the single-path output of an experiment, we can then introduce two random variables S and PT:

- S: The starting node of this selected path.
- PT: The path type of this selected path.

Note that both S and PT are discrete random variables, where the number of possible realizations of S equals the total number of nodes in the MRN, and that of PT equals the total number of path types. Given these random variables, we can now compute the dependencies between nodes and path types in a variety of ways, described below.

2.2.2 Measuring Node/Path Dependence via Contribution

We start by an observation that each path type contains multiple realizations in the data set. Take the path type $writes(?X, ?Y)$ as an example: an instance $A1$ might occur in many paths of this type (say,

$$writes(A1, P1), \ldots, writes(A1, P99),$$

representing that $A1$ wrote 99 papers), whereas another instance $A2$ might occur only in a few (say, one time). Assuming that in the whole data set, only $A1$ and $A2$ write papers, we can say that $A1$ contributes 99 percent, $A2$ contributes 1 percent, and the rest contributes 0 percent to this "writing a paper" path type. That is, if a path is selected randomly, then the conditional probability is 99 percent for the event "given the path is of type writes, then the starting node is $A1$" or $p_{RE1}(S = A1|PT = writes)$. In this case, we can say that the dependency measure between the node $A1$ and path type $writes$ is 0.99.

Definition 7. *Let s be a node and pt be a path type. Then, the $contribution_k$ relative to the Random Experiment k of node s to path type pt is the conditional probability between the two random variables S and PT:*

$$contribution_k(s, pt) = p_k(S = s|PT = pt).$$

The contribution value therefore encodes the dependency information between a node and a path type. The concept is intuitive and understandable, since it basically encodes the relative frequency with which a particular path type is associated with a node. This is a very useful and important characteristic for generating human-understandable explanations, which will be described later. Besides the contribution measure, we also designed additional measures to compute node-path dependency based on MI and pointwise MI (PMI). These measures and experiments evaluating them are described in [10]. Finally, given these path-type features and their contributions with respect to nodes as feature values, we can construct the semantic profiles of nodes.

2.3 Metaconstraints

As described above, path types can be generated from paths by a method called variable relaxation. Given that every element in a path can be relaxed to a variable, how can the system systematically select a set of path types? In this section, we describe several metaconstraints that control how the system determines path types automatically from the given data. Moreover, these high-level constraints also provide users a means to introduce their biases or domain knowledge if necessary. There are essentially two types of constraints: 1) path equivalence constraints that determine when two paths are considered to be equivalent or of the same path type and 2) filter constraints that restrict the set of paths considered in feature selection, for example, path-length constraints and link-type constraints would fall into this category.

Metaconstraint 1 (relation-only constraint). This is a path equivalence constraint that tells the system to treat paths with the same sequence of relations (links) as of the same path type. In other words, it considers only link types and ignores any information provided by the nodes in a path. This is the default constraint used by our system to create semantic profiles, since in general, we assume that the typed links play a more important role and can convey more information than the names or types of nodes in a path.

Metaconstraint 2 (maximum path length). This is a filter constraint used by the system to limit the path length while selecting path types as features. Constraining the path length is important for a variety of reasons: for one, the farther away a node or link is from the source node, the less impact it has on the source node's semantics. Moreover, the longer a path is, the harder it is for humans to make sense of it. Path length is also an important performance determiner, given that considering longer and longer paths can lead to an explosion in path instances and path-type features. By default, our system uses a maximum path length of five, which has worked well on the various data sets we have analyzed so far.

Based on Metaconstraints 1 and 2, the system can fully automatically extract a set of path types from an MRN to represent the semantics of the nodes. Note that if the maximum path length chosen is k, all path types of length 1 to k that occur in the data will be selected as features.

Metaconstraint 3 (node- and link-type constraints). These are filter constraints that allow users to express their preference in terms of the types of nodes and links that should be considered (instead of considering all link types, which is done by default). For example, one could specify that at least one of the nodes in a path needs to be of type *person* or that one of the links needs to be a *murder* link.

3 Abnormal Node Discovery with UNICORN

We can now describe how our node discovery program UNICORN identifies abnormal nodes in an MRN or semantic graph.[2] The information flow is shown in the upper part of Fig. 2. First, we compute the set of path types

[2] Besides referring to a mythical animal, UNICORN also stands for "**UN**supervised **I**nteresting-instance dis**CO**very in multi**R**elational **N**etworks."

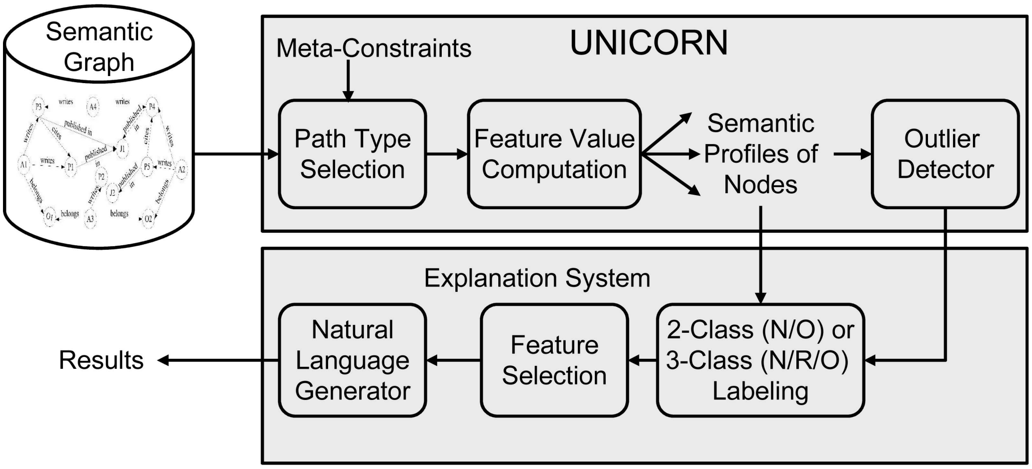
Semantic Graph
UNICORN
Meta-Constraints
Path Type Selection
Feature Value Computation
Semantic Profiles of Nodes
Outlier Detector
Explanation System
Natural Language Generator
Feature Selection
2-Class (N/O) or 3-Class (N/R/O) Labeling
Results

Fig. 2. Information flow in UNICORN.

to use as features in the semantic profiles of nodes. It is important to note that users can choose either to incorporate their domain knowledge through metaconstraints or use the default relation-only constraint in this stage. In the default configuration, *no user input* is required to determine the set of path-type features from the given MRN. Once the set of path types is determined, UNICORN computes the dependency (that is, either contribution or PMI values based on one of the random experiments) as the feature values to generate a semantic profile for each node. Finally, UNICORN applies an outlier ranking algorithm to find nodes with abnormal semantic profiles. The following pseudocode describes this more formally:

```
function UNICORN (M, mc, k)
// $M$ is an MRN $< V, E, L >$
// mc is a meta-constraint for selecting path types
//     (e.g., the default "relation-only")
// $k$ is the maximum path length for path types
var profile array$[|V|, |PT|]$ of float
$PT :=$ extractPathTypes$(M, mc, k)$
for $n \in V$
    for $pt \in PT$
        profile$[n, pt] :=$ getContribution$(M, n, pt)$
return getOutliers(profile)

function extractPathTypes(M, mc, k)
$PT := \{l_1, l_2, \ldots, l_{|L|}\}$
for pathLength := 1 to $k - 1$
    for $pt \in PT$ where length(pt) = pathLength
        for $l \in L$
            $pt' :=$ concatenate$(pt, l)$ // add $l$ to end of $pt$
            if pathExists$(M, pt')$ and satisfies$(pt', mc)$
                $PT := PT \cup pt'$
return $PT$

function getContribution$(M, s, pt)$
starts := number of paths of type pt in M that s starts
all := number of paths of type pt in M
return $starts/all$
```

An important aspect of UNICORN is that it is designed as a *framework* with a variety of options at each stage that users can choose from. In the feature selection stage, metaconstraints provide users a certain amount of flexibility to introduce their preferences, without affecting the overall advantage of being domain independent. In the feature value generation stage, users can choose from two different random experiments (RE1 as default), as well as from several dependency models such as contribution (the default), MI, and PMI measures, each of which has a slightly different view and intuition for node/path dependency. In the final stage, users can apply different types of outlier ranking algorithms to find different types of abnormal instances. By default, UNICORN uses Ramaswamy's distance-based outlier algorithm, which finds outliers based on the largest kth neighbor distance [11].

The lower part of Fig. 2 describes the subsystem that produces the explanations for UNICORN's results, which will be elaborated in more detail in Section 5.

4 Evaluation of UNICORN

In its default configuration, UNICORN returns a list of nodes ranked in descending order by their kth neighbor distance, that is, nodes with the largest distance to their kth neighbor are listed at the top. What we want to evaluate here is some notion of *quality* of the rankings UNICORN generates. Unfortunately, there is no gold standard available to us describing what it means for an instance to be truly abnormal, nor do we have labeled test data describing such instances that makes evaluation a somewhat challenging problem (cf., [12]).

UNICORN finds anomalous instances based on their semantic profiles and the chosen outlier computation. The evaluation question we are trying to answer below is whether for some relevant data sets of interest, the abnormality scores UNICORN computes correspond with some real-world notion of suspiciousness, interestingness, or, most generally, usefulness.

We address this question by applying UNICORN to a set of third-party-generated synthetic data sets in the domain of Russian organized crime. The primary reason for using synthetic data is that it comes with an answer key and ground truth describing entities of interest such as perpetrators that need to be found, which allows us to measure the

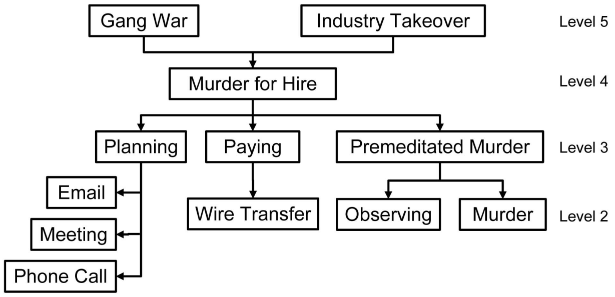

Fig. 3. Event-type hierarchy of the simulated Russian organized crime data.

quality of a result, which is generally not possible with natural data such as the movie data set described in Section 6. Other reasons are that real data from the area of law enforcement or intelligence analysis is generally difficult to come by due to access restrictions and privacy concerns. After applying UNICORN to these data sets, we compare its performance with a selection of other state-of-the-art unsupervised network algorithms that have been used for problems in law enforcement and counterterrorism.

The data sets are part of a large suite of simulated data sets developed during the US Defense Advanced Research Projects Agency (DARPA)'s Evidence Extraction and Link Discovery Program for the purpose of evaluating link discovery algorithms such as pattern matchers, group detectors, etc., (see [13] for additional context). The data was generated by a simulator of a Russian organized crime (or Mafiya with a "y") domain that simulates the whole process of ordering, planning, and executing high-level criminal activities such as murders for hire or gang wars with a large number of possible variations and records an incomplete and noisy picture of these activities in the generated evidence files (for example, financial transactions, phone calls or email, somebody being observed at a location, somebody being killed by someone unknown, etc.). The hierarchy of event types is shown in Fig. 3. The highest level events, gang wars and industry takeovers, both involve lower level events such as contract murders, which in turn involve some planning, financing, execution, etc. Gang wars occur between two rivaling Mafiyas, and industry takeovers are attempts by one Mafiya to take over an industry controlled by another. The general task for an algorithm is to detect these high-level activities (not reported in the data) from the low-level incomplete noisy evidence. For evaluation, the details of these activities and their participants are described in an answer key.

Despite the obvious simplification over real organized crime activity, the generated data is quite large and complex. We tested on six different data sets (D1 to D6) whose characteristics are described in Tables 1 and 2. Each data set contains roughly 6,000-9,000 nodes, whereas the number of links ranges from 8,000 to 16,000. There are 16 different node types representing objects and events and 31 different link types representing the relationships between those nodes. Each data set contains 42 Mafiya groups and 21 different industries. The data sets differ in size (large or medium), observability (low, average, or high), and noise level (0, 1, or 2). The size parameter does not reflect the total number of nodes and links in the data but instead how many contract murder events there are. Large-sized data sets contain 20 contract murder events and both a gang war and industry takeover, whereas medium-sized ones only have 10 contract murder events and only an industry takeover. Observability stands for the amount of evidence that is actually revealed. In other words, the lower the observability is, the less evidence of events is reported in the data. The noise parameter represents the degree of noise in level-2 events, with approximately 2,000 noise events for level-2, 1,000 noise events for level-1, and no noise event for level-0 data sets. Given these settings, D1 is the most difficult data set (large size, low observability, and high noise), whereas D6 is the least difficult.

TABLE 1
Properties of the Simulated Russian Organized Crime Data

Data	Size	Observability	Noise Level	Total Nodes	Total Links
D1	large	low	2	9429	16257
D2	large	average	1	8418	12897
D3	large	high	0	6346	8349
D4	medium	low	2	9142	15425
D5	medium	average	1	8033	12687
D6	medium	high	0	6172	7848

We used this data to perform the following experiment: for each data set, we feed the evidence data graph to UNICORN and ask it to rank the 42 Mafiyas based on their abnormality (using their semantic profiles and Ramaswamy's distance-based outlier algorithm). We then check how well the top-ranked Mafiyas correspond to the Mafiyas of interest reported in the answer key (three for large-sized and one for medium-sized data). We compare UNICORN's performance with that of a set of unsupervised network algorithms that fall into two classes: 1) The first class is composed of centrality-based ranking algorithms such as PageRank [14], Hypertext Induced Topic Selection (HITS) [15], and Betweenness Centrality [16], which all compute some form of importance or authority score based on the connectivity of a node in a graph. PageRank and Betweenness were chosen, specifically, since they have been applied previously to analyze terrorist networks [5], [6]. 2) The second class of algorithms is outlier based, which is similar to UNICORN, such as an outlier detector based on Markov graph clustering [17] and an unlabeled length-only variant

TABLE 2
Node and Link Types

Node Types	Link Types	
BankAccount	accountHolder	orgMiddleman
Business	callerNumber	payee
Email	ceo	payer
Industry	dateOfEvent	perpetrator
Mafiya	deliberateActors	phoneNumber
Meeting	deviceUsed	receiverNumber
Murder	employees	recipient
MurderForHire	eventOccursAt	relatives
Observing	geoSubregions	sender
Paying	hasMember	socialParticipants
Person	mediator	subevents
PhoneCall	hitContractor	transferMoneyFrom
PhoneNumber	hitman	transferMoneyTo
Planning	objectsObserved	victim
PremeditatedMurder	operatesInRegion	vor
WireTransfer	orgHitman	

TABLE 3
Performance Results for Various Algorithms on the Simulated Russian Organized Crime Data Sets

Dataset		**UNICORN**	HITS	Unlabeled UNICORN	Reverse MCO	Reverse PageRank	Betweenness	Dataset Average	Page Rank	MCO	Average All
D1	GM1	***2***	5	5	7	4	32		39	36	
	GM2	***3***	15	30	5	***2***	41		41	38	
	ITM	***1***	***1***	***1***	13	21	***1***		22	30	
	Perfect	3	1	1	0	1	1	38.9%	0	0	29.2%
	max Err	0.0%	30.8%	69.2%	25.6%	46.2%	97.4%	44.9%	97.4%	89.7%	57.1%
	avg Err	0.0%	13.7%	26.5%	18.8%	17.9%	59.0%	22.6%	84.6%	86.3%	38.4%
D2	GM1	***2***	***2***	***1***	12	12	***2***		31	31	
	GM2	***3***	***3***	4	***2***	6	10		37	41	
	ITM	***1***	***1***	***2***	18	25	28		18	25	
	Perfect	3	3	2	1	0	1	55.6%	0	0	41.7%
	max Err	0.0%	0.0%	2.6%	38.5%	56.4%	64.1%	26.9%	87.2%	97.4%	43.3%
	avg Err	0.0%	0.0%	0.9%	22.2%	34.2%	29.1%	14.4%	70.9%	80.3%	29.7%
D3	GM1	***1***	***1***	***1***	7	5	***2***		38	36	
	GM2	***2***	***2***	***2***	5	***3***	9		40	38	
	ITM	***3***	6	4	***2***	***2***	12		41	41	
	Perfect	3	2	2	1	2	1	61.1%	0	0	45.8%
	max Err	0.0%	7.7%	2.6%	10.3%	5.1%	23.1%	8.1%	97.4%	97.4%	30.4%
	avg Err	0.0%	2.6%	0.9%	6.8%	1.7%	14.5%	4.4%	99.1%	95.7%	27.7%
D4	ITM	2	2	2	***1***	9	10		34	42	
	Perfect	0	0	0	1	0	0	16.7%	0	0	12.5%
	avg Err	2.4%	2.4%	2.4%	0.0%	19.5%	22.0%	8.1%	80.5%	100.0%	28.7%
D5	ITM	2	5	3	***1***	***1***	19		42	42	
	Perfect	0	0	0	1	1	0	33.3%	0	0	25.0%
	avg Err	2.4%	9.8%	4.9%	0.0%	0.0%	43.9%	10.2%	100.0%	100.0%	32.6%
D6	ITM	***1***	***1***	***1***	3	***1***	***1***		42	40	
	Perfect	1	1	1	0	1	1	83.3%	0	0	62.5%
	avg Err	0.0%	0.0%	0.0%	4.9%	0.0%	0.0%	0.8%	100.0%	95.1%	25.0%
Algorithm Average	Perfect	**83.3%**	58.3%	50.0%	33.3%	41.7%	33.3%		0.0%	0.0%	
	max Err	**0.8%**	8.4%	13.6%	13.2%	21.2%	41.7%		93.8%	96.6%	
	avg Err	**0.8%**	4.7%	5.9%	8.8%	12.2%	28.1%		89.2%	92.9%	

For each algorithm, the ranking of the target Mafiyas GM1, GM2, and/or ITM are shown (ideal rankings in bold italics), as well as the number of perfect rankings and the maximum and average positional ranking errors (which are the same for D4-D6).

of UNICORN that does not consider link labels but only lengths of paths. All of the comparison algorithms work on unlabeled graphs only, so any edge label information is removed before the algorithm is applied (to the best of our knowledge, there are no other algorithms of this kind available that do use edge label information). Note that the data sets were not expressly designed to test anomaly detectors, so it is not clear a priori whether anomalous Mafiyas will or should match the ones reported in the answer key.

Results are shown in Table 3. For each algorithm, we show how it ranks the two gang war Mafiyas GM1 and GM2 and/or the industry takeover Mafiya ITM described in the answer key of each data set. A perfect algorithm would rank the target Mafiyas at positions 1–3 for data sets D1–D3 and at position 1 for data sets D4–D6. The "Perfect" score for each algorithm lists how many Mafiyas were ranked perfectly (that is, there is no "innocent" Mafiya ranked higher than it) for each data set. We also average and normalize these scores to show what overall percentage of Mafiyas was ranked in perfect position. To evaluate the quality of nonperfect rankings, we compute a position error that measures on the average how many innocent candidates are ranked higher than the target ones. The rationale for this is that an investigator or analyst following the generated ranking would have to investigate all these false positives before the particular target Mafiya was reached and, presumably, identified as a hit. We compute a normalized maximum error "max Err" based on the worst of the rankings. For example, for Betweenness on data set D3 the worst ranking is 12 (for the industry takeover Mafiya); therefore, the error would be computed as $(12 - 3)/39 * 100 \text{ percent} = 23.1 \text{ percent}$. The average position error "avg Err" averages these errors over all target Mafiyas (for medium-sized data sets D4–D6, the maximum and average errors are the same).

Algorithms are listed in order of performance. UNICORN using its default relation-only constraint to automatically generate path-type features and a maximum path length of five scores best overall with 83.3 percent of perfect rankings and an average position error of less than 1 percent. Its closest competitor HITS has an average error that is six times larger. The difference is even more significant on the most difficult data set D1. The unlabeled version of UNICORN that was run in an identical setup but computing path-type contributions based on path length only and ignoring all edge label information comes in third. This shows that for this data, a significant portion of the relevant information does come from basic path frequencies and connectivity, but adding label information can very significantly improve the accuracy of the generated rankings.

MCO stands for "Markov Clustering-based Outlier Detection" and uses the MCL graph clustering package [17] to find outliers in the evidence graph. Clustering finds outliers not directly but as a side effect, and we had to use MCL in the following way to generate a ranked list of outliers: By sweeping MCL's inflation parameter from low to high, we generated clusterings with finer and finer granularity containing more and more single-element clusters (outliers). We stop once all 42 Mafiyas are reported in an outlier cluster and use the inflation parameter value at which a Mafiya became an outlier as its ranking (the strongest outliers being those generated with the lowest inflation value and most coarse-grained clustering). Given the strong performance of outlier detectors such as UNICORN, it was somewhat surprising that MCO performed very badly and that its rankings were in fact negatively correlated with the desired target Mafiyas. Upon further analysis, it turned out that MCO's top-ranked Mafiyas had generally a higher node degree (an average 9.9 for its top five) than its bottom-ranked results (an average 6.6 for its bottom five), whereas the average node degree for Mafiyas was 7.8 with an average maximum of 12 and a minimum of 4.3. Moreover, the average node degree of the target Mafiyas to be found was also 7.8 ($\sigma = 1.9$). Therefore, on average, there was nothing special about their node degree, whereas MCO preferred those with higher degrees. One possible explanation for this behavior is that when we ask MCO to generate finer and finer clusters, nodes with a high degree present a problem, since they are connected to many other nodes that would generally increase cluster size; therefore, marking them as single-element outliers can increase the global quality score of the resulting clusters.

For this reason, we added a "Reverse MCO" pseudoalgorithm for comparison that simply reverses the list generated by MCO, which in turn performs quite well. PageRank also generated rankings that were negatively correlated, and we added a "Reverse PageRank," which was a much better predictor of the desired target Mafiyas. Both of these results indicate that for this data, high connectivity or a hub score can be a bad predictor for the target nodes sought.

In summary, some centrality and authority scorers such as HITS and Betweenness, as well as some outlier detectors such as UNICORN and Unlabeled UNICORN, picked up the desired target Mafiyas with good to reasonable accuracy. However, UNICORN, which is anomaly based and the only algorithm that took edge labels into account, consistently outperformed or matched all other algorithms on all data sets with an average error that is six times smaller than its closest competitor and 83.3 percent of all Mafiyas being ranked in perfect position. Performance differences were most significant on the hardest data set D1. Note that the average maximum error for each data set is a good match for our predicted data set difficulty based on the parameters reported in Table 1, with D1 being the most difficult and D6 being the easiest data set.

We believe the major reason that UNICORN outperformed the other algorithms by a significant margin is twofold: First, it has the capability to utilize the information provided by different types of links, whereas the other algorithms do not take different semantics of links into account. Second, using abnormality as a heuristic for finding suspicious instances seems to be a better option than using the centrality or importance of nodes.

We also performed experiments to rank the 21 industries in each data set with very similar results, which are reported in detail in [10]. Additionally, we evaluated the performance on the *local node* discovery problem where we try to identify nodes that are abnormally connected to a *given source* such as one of the gang war Mafiyas or the industry in an industry takeover. Results show (see [10]) that this task is significantly easier and that for most of the data sets except the hardest ones (that is, data sets D1, D4, and D5 with low observability and high noise), all algorithms can successfully identify the crime participants.

One reason why all algorithms perform significantly better here is that by providing a high-quality source node, we provide a seed suspect, which has a much higher chance to have strong connections to other suspects than an average node. Therefore, finding important or strong connections to a seed suspect is likely to turn up other suspects ("guilt by association"), which is a phenomenon also exploited by some group detection algorithms [18]. The result reveals an interesting phenomenon where suspects have both important and abnormal connection with other suspects, since they can be detected with both types of mechanisms. However, again, except for the two hardest data sets D1 and D4 with low observability and high noise, our algorithm outperforms the others by a significant margin.

We also evaluated how UNICORN performs with different choices of maximum path length (see [10]). The results show that path length correlates positively with performance on the crime data sets. As reported in Table 3, for $k = 5$, the results are close to perfect. The results suggest that it is useful to consider information provided by nodes and links that are multiple steps away from the source node in our framework. Note that the improvement decreases gradually with increasing path length (33 percent from $k = 1$ to $k = 2$, but only 11 percent from $k = 4$ to $k = 5$). The deterioration of the improvement shows that the quality of information does not improve linearly with the increasing number of features, which implies a reasonable hypothesis that the farther away a node or link is from the source, the less impact it has on it.

5 Generating Explanations for Abnormal Nodes

The idea of our explanation-based discovery system is to summarize why abnormal nodes are different from the rest in some human-understandable form such as natural language. The goal is to provide users information to assist them in judging whether the reported instances are really suspicious or not. For example, in the experiment described above, our system reported the Mafiya groups it found to be most abnormal. Since this was a synthetic data set with an answer key, we could look at that to see whether the results it came up with made sense. In a real-world situation, no answer key is available, and conceivably, more information is required to convince an analyst or investigator of the validity and relevance of a result.

5.1 Explaining the Uniqueness of Nodes

This section describes a novel mechanism to generate explanations for the abnormal nodes discovered by our system. Recall that we find such nodes by looking for those that possess abnormal semantic profiles derived from the semantic graph. Technically, this is done by first generating a set of features (path types), together with their feature values (contributions) for each node, and then applying a distance-based outlier algorithm to extract abnormal nodes. The process of generating explanations can be viewed as a kind of summarization process that identifies a small amount of key features that cause a node to be unique and describes those and their values in a human-understandable form.

To explain an abnormal point, the explanation system first needs to select a subset of features that contribute the most to its uniqueness. That is, it has to determine a subset of features whose feature values in combination are sufficient to distinguish the abnormal node from other points. In addition, it has to analyze how such an outlier performs in terms of these features to separate it from the rest, and then, it has to produce a human-understandable explanation based on that.

To select a small set of dominant features, we treat an explanation as a process of *classification*, where describing the differences between instances is accomplished by finding a method to classify different types of points into different classes and then characterizing these classes to a human. This idea is implemented by assigning a special class label to the outlier (labeled O in Fig. 2) and then applying a human-understandable classifier such as a decision tree to separate the outlier class from the rest (the normal class labeled N in Fig. 2). The process of the decision tree classifier can then be rendered into a human-understandable explanation. All explanations shown in this paper use this two-class N/O explanation scheme; however, in [10], we also developed a three-class scheme that uses an additional reference class R.

We designed two different strategies for explanation. The first uses a normal continuous decision tree that employs the information gain heuristic for feature selection. It produces a set of decision rules describing the uniqueness of a node, which can be rendered into explanations such as the following example from the organized crime data set:

uid667 is the only 1 Mafiya that has

- *larger than 20.8 percent contribution for [hasMember, ceo] (eliminates 40 nodes) and*
- *smaller than 11.1 percent contribution for [hasMember, sender] (eliminates 1 node).*

This example tells us that *uid667* is the only Mafiya that has a more than 20.8 percent chance to be the starting node S of a path of type "S has some member who is the CEO of some company" and a smaller than 11.1 percent chance to start a path of type "S has some member who is the sender of some communication event."

The second strategy is to give higher priority to the decision boundary at zero, which we call *zero/nonzero separation*. For the same example node, we now get this explanation:

uid667 is the only 1 Mafiya that has

- *nonzero [hasMember, hitContractor] (eliminates 38 nodes),*
- *zero [operatesInRegion] (eliminates 2 nodes), and*
- *zero [hasMember, victim] (eliminates 1 node).*

The above decision rules tell us that *uid667* is the only Mafiya that has some member who ordered a murder (that is, contracted a hit), does not operate in some region, and does not have a member who is a victim in some event.

The final step of explanation generation is to translate the path-based network representation into natural language. For example, we want to translate the following path:

$$John \xrightarrow{emails} ?P \xrightarrow{fatherOf} ?Q \xrightarrow{travelsTo} USA$$

into "John sends an email to somebody whose child travels to USA." Due to the space limitation, we do not present the technical details of the natural language generation system, which have been described in [19].

TABLE 4
Evaluation Results for the Explanation System

	ITM	GM1	GM2	Avg. Conf.	Avg. Time
Task 1	10%	10%	10%	0.3	60.0 min
Task 2	90%	90%	90%	2.2	22.5 min

For each task, the percentage of subjects that correctly identified the target Mafiyas ITM, GM1, and GM2 is shown, as well as their average confidence and time to solve the task.

Here is another example of a zero/nonzero explanation for an abnormal movie person generated by our system from the KDD Movie data set described in Section 6:

Salvador Dali is the only one actor in the data set (which contains 10,917 candidates) that

- *is the visual director of some movie,*
- *is the writer of some movie, and*
- *never is the sibling of some movie person.*

5.2 Evaluating the Explanation System

In this section, we describe an evaluation of the usefulness of our explanation system based on the synthetic organized crime data set. The goal is twofold: first, we want to know whether the explanations generated by our system can assist human subjects to make more accurate and confident decisions in terms of identifying the crime participants, and second, we want to know whether they can reduce the time needed to make these identifications. To answer these questions, we designed three tasks: In Task 1, we provided subjects the original D1 data set as a file (see Table 1) with English translations for each relation. We then asked them to select three Mafiyas from 10 given candidates that are most likely to have participated in the gang wars (GM1 and GM2) and industry takeover (ITM). In Task 2, we provided the zero/nonzero explanations for those 10 candidates and asked the subjects to perform the same task based only on the provided explanations. We recorded the time (the maximum time allowed was 60 minutes) and confidence (from 0 to 4, 0 means no confidence at all) for each task. To avoid interference and bias among different tasks, we changed the names of the candidates for each task and also told the users that they are from different data sets. We tested on 10 human subjects. The results are described in Table 4.

For Task 1, for each Mafiya group sought, only one human subject did successfully identify it within the time limit. Six of the 10 subjects gave up after spending less than 60 minutes on it for the reason that they believed the data to be too complicated for them to analyze. This is understandable given the thousands of nodes and links that subjects have to keep track of as the system does. Our human subjects' feedbacks indicate that reasoning with higher degree paths is very hard for humans, in particular, compared with the machine, which only took less than 1 minute to generate the semantic profiles for the Mafiyas and produce the explanations. Note that the times recorded here are the average time for the subjects who reported at

TABLE 5
Two Types of Explanations for Hitchcock in the Movie Data

(A) Zero/non-zero separation is on:

Hitchcock is the only 1 actor in the dataset (which contains 10917 candidates) that

- *is the mentor of some movie person, and*
- *is affected by some movie person*

(B) Zero/non-zero separation is off:

Hitchcock is the only 1 actor in the dataset (which contains 10917 candidates) that has

- *larger than 4.1% chance to be the starting node S of the paths with the path type representing "S directed some movie"*

least one correct candidate. The confidence level for Task 1 is close to 0 (0.3). This demonstrates that the original network with baseline explanation (that is, simply translating every single relation into English) is very difficult for humans to analyze within a limited amount of time. The results for Task 2 show that armed with zero/nonzero explanations, subjects do much better and have a very high chance (90 percent) to identify the abnormal candidates within much less time spent (22.5 minutes) and much higher confidence (2.2).

6 Application to Two Real-World Data Sets

Below, we demonstrate how UNICORN can find abnormal nodes in real-world natural data sets. Our goal is to show how the complete system works, as well as that it is domain independent and can be applied to find abnormal or interesting instances in arbitrary semantic graphs. For this purpose, we applied UNICORN to Gio Wiederhold's KDD Movie data set,[3] and to the HEP-Th High Energy Physics Theory (HEP-Th) bibliography data set from the 2003 KDD Cup.[4]

In the first network generated from the movie data, there are about 24,000 nodes representing movies (9,097), directors (3,233), actors (10,917), and some other movie-related persons (500) (the numbers in parentheses show the number of different nodes for each entity type). We extracted about 100,000 relations between these nodes. There are 44 different relation types in this data set, which can be divided into three groups: relations between people (for example, spouse and mentor), relations between movies (for example, remake), and relations between a person and a movie (for example, director and actor). To make the explanations more understandable, we chose to use at most three features to explain nodes and limit the maximum path length to four.

Table 5 illustrates UNICORN's two different explanation types to explain the uniqueness of Hitchcock when viewed as an actor. According to our system, he is one of the most abnormal actors in the movie data set, which is not surprising, given the data set's bias for Hitchcock movies and his much more prominent role as a director. In (A), zero/nonzero separation is used, and the system extracts two features (that is, nonzero contributions of two path types) to separate him from the rest of the world. In (B), the standard information gain heuristic is used, which unveils important information for the uniqueness of Hitchcock by showing that he directed more movies than any other actor in the data set.

TABLE 6
Two Types of Explanations for an Abnormal Researcher in the High-Energy Physics Data

(A) Zero/non-zero separation is on:

C.N. Pope is one of the only 1145 authors in the dataset (which contains 7152 candidates) that

- *published two or more papers with the same keyword, and*
- *never has a colleague that once belonged to the two institutes he has ever belonged to, and*
- *has a coauthor from the same organization*

(B) Zero/non-zero separation is off:

C.N. Pope is the only 1 author in the dataset (which contains 7152 candidates) that has

- *larger than 1.01% chance to be the starting node S of the paths with the path type representing "S wrote a paper that cites his/her own paper"*

For the second network, we extracted six different types of nodes and six types of links (plus their six inverses) from the HEP-Th bibliography data. Nodes represent paper IDs (29,014), author names (12,755), journal names (267), organization names (963), keywords (40), and publication times encoded as year/season pairs (60). Most of these node types and their associated link types are illustrated in Fig. 1. There were 43,099 different nodes and 477,423 links overall.

Table 6 shows two explanations for an abnormal author in the HEP-Th data. As can be seen from (A), using zero/nonzero separation with three features is not sufficient to distinguish C.N. Pope from the rest of the authors, since there are still 1,145 other authors with the same characteristics. However, in (B), we can see that by using the standard information-gain heuristic, his abnormality can be explained with a single feature, which is that he has the highest chance of citing his own paper. One reason for this different explanation characteristic is that the HEP-Th network has more nodes and links but much fewer link types than the movie data. Since our path-type features are constructed by permuting and composing sequences of relations, the HEP-Th data set is a denser data set with more points and fewer features compared to the movie data set. In a dense data set, zero/nonzero separation might generally not be as useful, since we need more precise information to explain the uniqueness of a node.

Tables 7 and 8 present some examples of abnormal movies, directors, and actors generated from the KDD Movie data set. In these examples, we use the explanation mechanism itself to determine who is abnormal. To do that, UNICORN first generates the semantic profile for each node and then applies the explanation mechanism to explain them. Based on the explanations, we assign each node into one of two groups, the *abnormal* group and the *not-so-abnormal* group. A node belongs to the abnormal group if UNICORN can explain its uniqueness by three or fewer features using zero/nonzero separation; otherwise, it will be assigned to the not-so-abnormal group. The example nodes shown have been randomly picked from the abnormal group.

Note that given a semantic profile, our explanation system can explain not only abnormal nodes but also not-so-abnormal nodes. For example

[3] Available at http://archive.ics.uci.edu/beta/datasets/Movie.

[4] Available at http://www.cs.cornell.edu/projects/kddcup.

TABLE 7
Examples of Abnormal Movies from the Movie Data

(A) Zero/non-zero separation is on:

"Snow White and the Seven Dwarfs" is the only 1 movie in the dataset (which contains 9097 candidates) that

- *has a composer who is also an actor, and*
- *is remade from some movie adapted from a book, and*
- *is remade from some movie that has a composer*

(B) Zero/non-zero separation is off:

"Phantom of The Paradise (1974)" is the only 1 movie in the dataset (which contains 9097 candidates) that has

- *larger than 5.3% chance to be the starting node S of the paths with the path type representing "S has been remade into a movie that has some visual director"*

TABLE 8
Examples of Abnormal Directors from the Movie Data

(A) Zero/non-zero separation is on:

Woody Allen is the only 1 director in the dataset (which contains 3233 candidates) that

- *acted in some movie that was remade into some other movie, and*
- *never acted in some movie that has a producer*

(B) Zero/non-zero separation is off:

Stephen King is the only 1 director in the dataset (which contains 3233 candidates) that has

- *larger than 36.5% chance to be the starting node S of the paths with the path type representing "S wrote some book adapted for a movie"*

L. Anderson is one of the only ***48*** *actors in the data set (which contains 10,917 candidates) that*

- *produced some movie,*
- *directed some movie, and*
- *never is the writer of some movie.*

In this case, the best our system can do based on three zero/nonzero features is to separate the nodes from the other $(10,917 - 48 = 10,869)$ nodes. In other words, there is still a group of 47 actors that are similar to Anderson. This example demonstrates that our explanation system can be used to generate explanations for every node regardless of whether it is unique or not. Moreover, the structure and size of explanations reveals the degree of abnormality of the nodes.

7 Related Work

7.1 Network Analysis for Homeland Security and Crime

In general, intelligent graph analysis methods have become popular to solve problems in homeland security [3], [20], [21] and crime mining [22], [23]. For semantic graphs, Adibi et al. [18] describe a method combining a knowledge-based system with MI analysis to identify threat groups given a set of seeds. Krebs [5] applied SNA to the 9/11 terrorists network and suggested that to identify covert individuals; it is preferable to utilize multiple types of relational information to uncover the hidden connections in evidence. This conclusion echoes our decision of performing discovery on top of a multirelational semantic graph. There are also link discovery and analysis algorithms proposed to predict missing links in graphs or relational data sets [24], [25]. Recently, several general frameworks have been proposed to model and analyze semantic graphs such as relational Bayesian networks, relational Markov networks, and relational dependency networks [26], [27], [28]. However, these frameworks aim at exploiting the graph structure to learn the dependency (that is, joint probability) or the posterior probability of events (nodes) or relations between them based on training examples. Our goal is very different, since we are focusing on identifying abnormal instances without introducing any training bias. Moreover, we also want to be able to generate human-understandable explanations for the system's findings.

7.2 Interesting Instance Discovery

The first part of our work (that is, identifying abnormal nodes) is closely related to our previous work on discovering various types of *interesting* instances in MRNs. In [29], we propose a rarity-based mechanism to identify interesting paths in a semantic graph. This paper proposed four different similarity measures for paths in the network and use rarity as a measurement to identify interesting ones. In our 2003 KDD Cup participation [30], we applied both rarity-based and abnormality-based methods to the HEP-Th bibliography data to find interesting entities based on their loop contributions. The first part of this paper can be seen as a generalized version of these algorithms. In [12], we addressed issues of verification for such unsupervised instance discovery systems, which eventually lead to the development of the explanation-based discovery system described in Section 3.

7.3 Social Network Analysis

Social networks consist of a finite set of actors (nodes) and the binary ties (links) defined between them. The goal of SNA is, in a nutshell, to provide better understanding of the structure of a given network [16]. Although most of the analyses are focused on finding social patterns and subgroups, there are a small number of SNA tasks resembling our instance discovery problem. *Centrality analysis* aims at identifying important instances in the network based on their connectivity with others: An actor is important if it possesses a high node degree (degree centrality) or is close to other nodes (closeness centrality). An actor is importantly connected to two source actors if it is involved in many connections between them (betweenness and information centrality). The major difference between centrality analysis and our approach is that we are trying to model suspiciousness by abnormality instead of centrality or prestige. *Social positions analysis* targets finding individuals who are similarly embedded in networks of relations. The similarity between actors is measured by whether they have similar ties with other actors. We generalize this concept by using paths and their statistical contributions. The generalized path features and their relative frequency of occurrence allow us to exploit more information from the network to capture the deeper meanings of instances.

Another main difference between the problems SNA handles and our problem is that most of the SNA approaches are designed to handle only one-mode or two-mode networks (that is, there are at most two types of actors) [16], whereas such a limitation does not exist in our analysis. Moreover, existing statistical and graph-theoretical methods for centrality, position, and role analysis do not distinguish between different link types and their different semantics, which limits their usage in more complicated networks, as demonstrated by our experiments. The relational networks we are dealing with are also not limited to social networks. They can be any relational graph, for example, a thesaurus such as WordNet. Finally, SNA does not address the problem of automatically generating human-understandable explanations for its results.

7.4 Relational Data Mining

Relational data mining (RDM) deals with mining of relational tables in a database. It is related to our problem in the sense that an MRN is a type of relational data and can be translated into relational tables. RDM searches a language of relational patterns to find patterns that are valid in a given relational database [31]. Morik [32] proposes a way to find interesting instances in this relational domain by first learning rules and then searching for the instances that satisfy one of the following three criteria: exceptions to an accepted given rule, not being covered by any rule, and negative examples that prevent the acceptance of a rule. Angiulli et al. propose a similar idea by using default logic to screen out the outliers [33]. Both methods require a certain amount of domain knowledge or training examples for supervised learning, which makes them different from ours, since we prefer to discover abnormal instances that are not expected by users nor biased by training data. There is, however, one type of unsupervised discovery problem called interesting subgroup discovery that tries to discover subsets that are unusual [34], [35], [36]. For example, interesting subsets are those whose distribution of instances based on a certain objective function is different from that of the whole data set. The major difference between our problem and subgroup discovery is that we are not looking for groups but individual instances; thus, the concept of abnormal statistical distribution is not directly applicable.

7.5 Outlier Explanation

There is a small amount of prior work on outlier explanation. For distribution-based outliers, Yamanishi and Takeuchi propose to combine statistical methods with supervised methods to generate outliers [37]. The statistical method is first applied to learn the data distribution and then to identify the outliers. Once the outliers are detected, the classification method can be applied to extract the filtering rules as explanation. Their approach is suitable for a situation where the distribution is known but not for a distance-based scenario in which the outliers could be very diverse. The idea of applying a classification method for explanation is similar to our explanation system. The key differences are that we can introduce an additional reference class in the explanation (see [10]) and that we treat each outlier separately. Furthermore, we translate the relevant features and their values into natural language. Yao et al. propose to apply a classification method to generate explanations for association rules. Their system utilizes external information that was not used in association-rule mining to generate the condition in which the rules hold [38], [39]. The external information, however, is not accessible in our problem.

8 Conclusions and Future Work

In this paper, we described a general unsupervised framework for identifying abnormal individuals in large complex semantic graphs. Our first contribution is the development of a novel framework called UNICORN, which can summarize the semantics of nodes into propositional semantic profiles. Given such profiles, we can compare and contrast nodes and exploit distance-based outlier algorithms to identify abnormal and suspicious nodes. Since the method is unsupervised and does not require training examples or user-defined features, it has the potential to discover truly novel instances without being biased by human analysts or training examples. Nevertheless, the process of selecting which relationships to use and creating a semantic graph can introduce some bias. Therefore, our general recommendation is to include as much information as possible and to let UNICORN's modeling and explanation mechanism determine which evidence (path) is important and which is not.

Motivated by issues of verification and the danger posed by false positives that might mistakenly incriminate innocent individuals, our system needs to not only identify suspicious nodes but also be able to explain to an analyst why they were chosen. To this end, we designed and implemented a novel explanation mechanism to produce natural language explanations for abnormal nodes, which is the second major contribution of this paper. The experiments show that our system outperforms several state-of-the-art network algorithms in terms of identifying abnormal nodes in a complex synthetic data set about organized crime. In an experiment with human subjects, we demonstrate that the explanation system can significantly improve the result quality, confidence, and efficiency of the subjects when analyzing a complex data set. Our application section demonstrates that the UNICORN framework is domain independent and can be applied not only to identify abnormal instances in crime and intelligence data sets but also to find such interesting instances in any MRN with a variety of potential applications in scientific discovery and data analysis.

Our algorithms have been successfully applied to networks with over 40,000 nodes and 475,000 links, which can be analyzed in under 2 minutes on a standard PC-based workstation. However, an important future direction is to further improve the scalability of the system. What is most expensive is the computation of feature values, since it requires the system to count a potentially large number of paths. We are currently investigating different sampling methods to approximate these values, which should enable us to further improve UNICORN's scalability.

Acknowledgments

This research was partly sponsored by the US Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory (AFRL), 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 US Government.

REFERENCES

[1] M. Sparrow, "The Application of Network Analysis to Criminal Intelligence: An Assessment of the Prospects," *Social Networks,* vol. 13, pp. 251-274, 1991.

[2] T. Senator and H. Goldberg, "Break Detection Systems," *Handbook of Knowledge Discovery and Data Mining,* W. Kloesgen and J. Zytkow, eds., pp. 863-873, Oxford Univ. Press, 2002.

[3] D. Jensen, M. Rattigan, and H. Blau, "Information Awareness: A Prospective Technical Assessment," *Proc. Ninth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '03),* pp. 378-387, 2003.

[4] J. Wang, F. Wang, and D. Zeng, "Rule+Exception Learning-Based Class Specification and Labeling in Intelligence and Security Analysis," *Proc. Workshop Intelligence and Security Informatics (WISI '06),* vol. 3917, pp. 181-182, 2006.

[5] V. Krebs, "Mapping Networks of Terrorist Cells," *Connections,* vol. 24, no. 3, pp. 43-52, 2001.

[6] J. Qin, J. Xu, D. Hu, M. Sageman, and H. Chen, "Analyzing Terrorist Networks: A Case Study of the Global Salafi Jihad Network," *Intelligence and Security Informatics,* P. Kantor et al., ed., pp. 287-304, Springer, 2005.

[7] "Total Information Overload," *Scientific Am.,* editorial, Mar. 2003.

[8] B. Simons and E. Spafford, *Letter from the Association for Computing Machinery's US Public Policy Committee to the Senate Armed Services Committee,* Jan. 2003.

[9] R. Hill, "Non-Well-Founded Set Theory and the Circular Semantics of Semantic Networks," *Intelligent Systems: Third Golden West International Conference: Edited and Selected Papers,* E. Yfantis, ed., pp. 375-386, Kluwer, 1995.

[10] S. Lin, "Modeling, Finding, and Explaining Abnormal Instances in Multi-Relational Networks," PhD dissertation, Dept. of Computer Science, Univ. of Southern California, 2006.

[11] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient Algorithms for Mining Outliers from Large Data Sets," *Proc. ACM SIGMOD '00,* pp. 427-438, 2000.

[12] S. Lin and H. Chalupsky, "Issues of Verification for Unsupervised Discovery Systems," *Proc. Workshop Link Analysis and Group Detection (LinkKDD '04),* 2004.

[13] R. Schrag, "A Performance Evaluation Laboratory for Automated Threat Detection Technologies," *Proc. Performance Measures for Intelligent Systems Workshop (PerMIS '06),* 2006.

[14] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web," *Proc. Seventh Int'l World Wide Web Conf. (WWW '98),* pp. 161-172, 1998.

[15] J. Kleinberg, "Authoritative Sources in a Hyperlinked Environment," *J. ACM,* vol. 46, no. 5, pp. 604-632, 1999.

[16] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications.* Cambridge Univ. Press, 1994.

[17] S. Dongen, "A Cluster Algorithm for Graphs," Technical Report INS-R0010, Nat'l Research Inst. for Math. and Computer Science, Amsterdam, The Netherlands, 2000.

[18] J. Adibi, H. Chalupsky, E. Melz, and A. Valente, "The KOJAK Group Finder: Connecting the Dots via Integrated Knowledge-Based and Statistical Reasoning," *Proc. 16th Innovative Applications of Artificial Intelligence Conf. (IAAI '04),* pp. 800-807, 2004.

[19] S. Lin, "Generating Natural Language Descriptions for Paths in the Semantic Network," final project report, Dept. of Linguistics, Univ. of Southern California, 2006.

[20] R. Popp, "Countering Terrorism through Information Technology," *Comm. ACM,* vol. 47, no. 3, pp. 36-43, 2004.

[21] H. Chen and F. Wang, "Artificial Intelligence for Homeland Security," *IEEE Intelligent Systems,* vol. 20, no. 5, pp. 12-16, 2005.

[22] J. Xu and H. Chen, "Criminal Network Analysis and Visualization," *Comm. ACM,* vol. 48, no. 6, pp. 101-107, 2005.

[23] H. Chen and J. Xu, "Intelligence and Security Informatics for National Security: A Knowledge Discovery Perspective," *Ann. Rev. of Information Science and Technology,* vol. 40, pp. 229-289, 2006.

[24] B. Taskar, M.-F. Wong, P. Abbeel, and D. Koller, "Link Prediction in Relational Data," *Advances Neural Information Processing Systems —Proc. 17th Ann. Conf. Neural Information Processing Systems (NIPS '03),* 2004.

[25] S. Adafre and M. Rijke, "Discovering Missing Links in Wikipedia," *Proc. Workshop Link Discovery: Issues, Approaches and Applications (LinkKDD '05),* 2005.

[26] R. Bunescu and R. Mooney, "Relational Markov Networks for Collective Information Extraction," *Proc. 21st Int'l Conf. Machine Learning (ICML '04),* 2004.

[27] J. Neville and D. Jensen, "Dependency Networks for Relational Data," *Proc. Fourth IEEE Int'l Conf. Data Mining (ICDM '04),* 2004.

[28] M. Jaeger, "Relational Bayesian Networks," *Proc. 13th Ann. Conf. Uncertainty in Artificial Intelligence (UAI '97),* 1997.

[29] S. Lin and H. Chalupsky, "Unsupervised Link Discovery in Multi-Relational Data via Rarity Analysis," *Proc. Third IEEE Int'l Conf. Data Mining (ICDM '03),* pp. 171-178, 2003.

[30] S. Lin and H. Chalupsky, "Using Unsupervised Link Discovery Methods to Find Interesting Facts and Connections in a Bibliography Dataset," *SIGKDD Explorations,* vol. 5, no. 2, pp. 173-178, Dec. 2003.

[31] S. Dzeroski, "Data Mining in a Nutshell," *Relational Data Mining,* S. Dzeroski and N. Lavrac, eds. pp. 1-27, Springer, 2001.

[32] K. Morik, "Detecting Interesting Instances," *Proc. ESF Exploratory Workshop Pattern Detection and Discovery,* 2002.

[33] F. Angiulli, R. Ben-Eliyahu-Zohary, and L. Palopoli, "Outlier Detection Using Default Logic," *Proc. 18th Int'l Joint Conf. Artificial Intelligence (IJCAI '03),* pp. 833-838, 2003.

[34] P. Flach and N. Lachiche, "Confirmation-Guided Discovery of First-Order Rules with Tertius," *Machine Learning,* vol. 42, nos. 1-2, pp. 61-95, 2001.

[35] S. Wrobel, "An Algorithm for Multi-Relational Discovery of Subgroups," *Proc. First European Symp. Principles of Data Mining and Knowledge Discovery (PKDD '97),* pp. 78-87, 1997.

[36] W. Klosgen, "Explora: A Multipattern and Multistrategy Discovery Assistant," *Advances in Knowledge Discovery and Data Mining,* U. Fayyad et al., ed., pp. 249-271, AAAI/MIT Press, 1996.

[37] K. Yamanishi and J. Takeuchi, "Discovering Outlier Filtering Rules from Unlabeled Data: Combining a Supervised Learner with an Unsupervised Learner," *Proc. Seventh ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '01),* pp. 389-394, 2001.

[38] Y. Yao, Y. Zhao, and R. Maguire, "Explanation-Oriented Association Mining Using a Combination of Unsupervised and Supervised Learning Algorithms," *Proc. Canadian Conf. Artificial Intelligence,* pp. 527-531, 2003.

[39] Y. Yao, Y. Zhao, and R. Maguire, "Explanation Oriented Association Mining Using Rough Set Theory," *Proc. Ninth Int'l Conf. Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing (RSFDGrC '03),* pp. 165-172, 2003.

Shou-de Lin received the PhD degree in computer science and the master's degree in computational linguistics from the University of Southern California. He is an assistant professor in the Computer Science and Information Engineering Department, National Taiwan University (NTU), where he also holds a joint appointment with the Graduate Institute of Networking and Multimedia. Before joining NTU, he was a postdoctoral researcher at Los Alamos National Laboratories. He received a best paper award in the IEEE 2003 Web Intelligence Conference and was awarded second place in the Open Task of the 2003 KDD Cup (together with Dr. Chalupsky). His research interests include machine discovery, knowledge discovery in social, and semantic networks, and natural language processing.

Hans Chalupsky received the master's degree in computer science from the Vienna University of Technology, Austria, and the PhD degree in computer science from the State University of New York, Buffalo. He is a project leader in the Information Sciences Institute, University of Southern California, where he leads the Loom Knowledge Representation, Reasoning, and Discovery Group. He has more than 20 years of experience in the design, development, and application of knowledge representation and reasoning systems such as PowerLoom, and he is the principal architect of the KOJAK Link Discovery System. His research interests include knowledge representation and reasoning systems, knowledge and link discovery, ontology translation and maintenance, semantic interoperability, and reasoning with incomplete information.