In the Workshop on Mining Graphs and Complex Structures at the 2007 IEEE International Conference on Data Mining.

Discovering Structural Anomalies in Graph-Based Data

William Eberle
Tennessee Technological University
weberle@tntech.edu

Lawrence Holder
Washington State University
holder@wsu.edu

Abstract

The ability to mine data represented as a graph has become important in several domains for detecting various structural patterns. One important area of data mining is anomaly detection, particularly for fraud, but less work has been done in terms of detecting anomalies in graph-based data. While there has been some work that has used statistical metrics and conditional entropy measurements, the results have been limited to certain types of anomalies and specific domains. In this paper we present graph-based approaches to uncovering anomalies in domains where the anomalies consist of unexpected entity/relationship deviations that resemble non-anomalous behavior. Using synthetic and real-world data, we evaluate the effectiveness of these algorithms at discovering anomalies in a graph-based representation of data.

1. Introduction

Setting up fraudulent web-sites, "phishing" for credit cards and stealing calling cards are just some of the examples of scams that have affected everyone from the investor to corporations. In every case, the fraudster has attempted to swindle their victim and hide their dealings within a morass of data that has become proverbially known as the "needle in the haystack". Yet, even when the data is not relatively large, the ability to discover the nefarious actions is ultimately difficult due to the *mimicry* of the perpetrator.

Much of the information related to fraud resides in the relationships among the various entities involved in an incident. Recently there has been an impetus towards analyzing multi-relational data using graph theoretic methods. Yet, while there has been much written about graph-based data mining for intrusion detection [11], little research has been accomplished in the area of *graph-based anomaly detection*.

Lin and Chalupsky [6] took the approach of applying what they called rarity measurements to the discovery of unusual links within a graph. The AutoPart system presented a non-parametric approach to finding outliers in graph-based data [1]. Part of this approach was to look for outliers by analyzing how edges that were removed from the overall structure affected the minimum descriptive length (MDL) of the graph [9]. The idea of entropy was used by Shetty and Adibi [10] in their analysis of the famous Enron e-mail data set. Using bipartite graphs, Sun et al. [12] presented a model for scoring the normality of nodes as they relate to other nodes. Rattigan and Jensen went after anomalous links using a statistical approach [8].

Using information theoretic, probabilistic and maximum partial substructure approaches, we have developed three novel algorithms for analyzing graph substructures for the purpose of uncovering all three types of graph-based anomalies: modifications, insertions and deletions. In this paper, we define what we consider to be an anomaly as it relates to graphs. Then, we present the algorithms along with some simple examples, followed by our results using synthetic and real-world related data sets.

2. Graph-Based Anomalies

The idea behind the approach presented in this paper is to find anomalies in graph-based data where the anomalous substructure in a graph is part of (or attached to or missing from) a non-anomalous substructure, or the *normative* substructure. This definition of an anomaly is unique in the arena of graph-based anomaly detection. The concept of finding a pattern that is "similar" to frequent, or good, patterns, is different from most approaches that are looking for unusual or "bad" patterns. While other non-graph-based data mining approaches may aide in this respect, there does not appear to be any existing approaches that directly deal with this scenario.

***Definition**: A graph substructure S' is anomalous if it is not isomorphic to the graph's normative substructure S, but is isomorphic to S within X%.*

X signifies the percentage of vertices and edges that would need to be changed in order for *S'* to be isomorphic to *S*. The thrust of this definition lies in its relationship to fraud detection. If a person or entity is attempting to commit fraud, they will do all they can to convey their actions as close to legitimate actions as possible. The U.N. Office on Drugs and Crime states the first law of money laundering as “The more successful money-laundering apparatus is in imitating the patterns and behavior of legitimate transactions, the less the likelihood of it being exposed.” [4].

2.1 Anomaly Types

For a graph-based anomaly, there are several situations that might occur:

1. *A vertex exists that is unexpected.*
2. *An edge exists that is unexpected.*
3. *The vertex label is different than expected.*
4. *The edge label is different than expected.*
5. *An expected vertex is absent.*
6. *An expected edge between vertices is absent.*

There are three general *categories of anomalies*: insertions(1,2), modifications(3,4) and deletions(5, 6).

2.2 Assumptions

In order to address our definition of an anomaly, we make the following assumptions about the data.

Assumption 1: *The majority of a graph consists of a normative pattern, and no more than X% of the normative pattern is altered in the case of an anomaly.*

Since our definition implies that an anomaly constitutes a minor change to the prevalent substructure, we chose a small percentage (e.g., 10%) to represent the most a substructure would be changed in a fraudulent action.

Assumption 2: *Anomalies consist of one or more modifications, insertions or deletions.*

As was described earlier, there are only three types of changes that can be made to a graph.

Assumption 3: *The normative pattern is connected.*

In all cases, the data consists of a series of nodes and links that share common nodes and links.

3. Graph-Based Anomaly Detection

Most anomaly detection methods use a supervised approach, which requires a baseline of information from which training can be performed. In general, if one has an idea what is normal behavior, deviations from that behavior could constitute an anomaly. However, the issue with those approaches is that one has to have data to train the system, and the data has to already be labeled (i.e., fraudulent versus legitimate).

Our work has resulted in the development of three algorithms, which we have implemented using a tool called GBAD (Graph-Based Anomaly Detection). GBAD is an *unsupervised* approach, based upon the SUBDUE graph-based knowledge discovery system [2]. Using a greedy beam search and Minimum Description Length (MDL) heuristic, each of the anomaly detection algorithms uses SUBDUE to provide the normative pattern in an input graph. In our implementation, the MDL approach is used to determine the best substructure as the one that minimizes the following:

$$M(S,G) = DL(G \mid S) + DL(S)$$

where G is the entire graph, S is the substructure, $DL(G|S)$ is the description length of G after compressing it using S, and $DL(S)$ is the description length of the substructure.

Using GBAD as the tool for our implementation, we have developed three separate algorithms: GBAD-MDL, GBAD-P and GBAD-MPS. Each of these approaches is intended to discover all possible graph-based anomaly types as respectively set forth earlier.

3.1 Information Theoretic (GBAD-MDL)

The GBAD-MDL algorithm uses a Minimum Description Length (MDL) heuristic to discover the best substructure in a graph, and then subsequently examines all of the instances for similar patterns.

3.1.1 Algorithm

The detailed GBAD-MDL algorithm is as follows:

Alg. 1: proc **GBAD-MDL** (graph *G*, threshold *T*)

```
1. Find normative substructure $S$ minimizing
   $DL(S)+DL(G|S)$, where the instances $I_k$ of $S$ in $G$
   have matchcost($I_k$,$S$) < ($T$ * size($S$))
2. For each instance $I_k$ such that matchcost($I_k$,$S$) > 0
   a. freq($I_k$) = num instances of $S$ that exactly match $I_k$
   b. anomalyScore($I_k$) = freq($I_k$) * matchcost($I_k$,$S$)
3. Return all instances $I_k$ having minimal anomalyScore
```

With the inexact matching, the result will be those instances that are the “closest” (without matching exactly) in structure to the best structure (i.e., compresses the graph the most), where there is a tradeoff in the cost of transforming the instance to match the structure (matchcost), as well as the frequency with which the instance occurs, where the lower the value, the more anomalous the structure.

3.1.2 Example

The following is a simple example of results obtained using our implementation of the GBAD-MDL algorithm described above. In Noble and Cook’s work on graph-based anomaly detection [7], they presented the example shown in Figure 1.

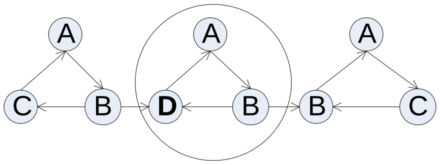

Figure 1. Example, with anomaly circled.

Running the GBAD-MDL algorithm on this example results in the (circled) anomalous substructure. With Noble and Cook's approach, the **D** vertex is shown to be the anomaly. While correct, the importance of this new approach is that a larger picture is provided regarding its associated substructure. In other words, not only are we providing the anomaly, but we are also presenting the context of that anomaly within the graph (the individual anomaly is in **bold**.)

3.2 Probabilistic (GBAD-P)

The GBAD-P algorithm uses the MDL evaluation technique to discover the best substructure in a graph, but instead of examining all instances for similarity, this approach examines all extensions to the normative substructure with the lowest probability. The difference between the algorithms is that GBAD-MDL is looking at instances of substructures with the same characteristics (e.g., size), whereas GBAD-P is examining the probability of extensions to the normative pattern to determine if there is an instance that includes edges and vertices that are probabilistically less than other possible extensions.

3.2.1 Algorithm

The detailed GBAD-P algorithm is as follows:

Alg. 2: proc **GBAD-P** (graph *G*, prob *P*, iterations *N*)

```
1. Find normative substructure $S$ minimizing
   $DL(S)+DL(G|S)$; where $I_j$ are instances of $S$ in $G$.
2. Compress $G$ by $S$, where all instances $I_j$ of $S$ in $G$ are
   each compressed to a new vertex $V$.
3. Iterate over each new vertex $V$, extending each
   vertex $V$ by all possible single edges $E$.
4. For instances $I_n$, where each instance of $I_n$ consists of
   $V$ and a unique extension, a substructure $S'$ consists
   of all matching instances $I_k$ from instances $I_n$.
5. For each instance $I_k$ , anomalyScore($I_k$) = number of
   instances of S' / $|I_n|$
6. Return instances $I_k$ with minimal anomalyScore $< P$.
7. Set $S$ to substructure definition of the $I_k$ with
   minimal anomaly score, and let $I_j$ be the instances of
   $S$ in $G$.
8. If current iteration < $N$, start next iteration at step 2.
```

anomalyScore(I_k) is the *probability* that an instance should exist given the existence of all of the extended instances. Given that $|I_n|$ is the total number of possible extended instances, freq(I_k) can never be greater, and thus anomalyScore(I_k) will never be greater than 1.0.

3.2.2 Example

Take the example shown in Figure 2.

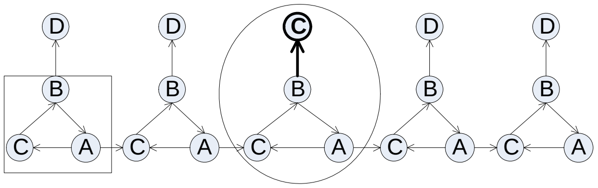

Figure 2. Example with instance of normative pattern boxed and anomaly circled.

After one iteration, the boxed instance in Figure 2 is one of the instances of the best substructure. Then, on the second iteration, extensions are evaluated, and the circled instance is the resulting anomalous substructure. Again, the edge and vertex (shown in **bold**) is labeled as the actual anomaly, but the entire anomalous substructure is output for possible analysis.

3.3 Max Partial Substructure (GBAD-MPS)

The GBAD-MPS algorithm again uses the MDL approach to discover the best substructure in a graph, then it examines all of the instances of parent (or ancestral) substructures that are missing various edges and vertices. The value associated with the parent instances represents the cost of transformation (i.e., how much change would have to take place for the instance to match the best substructure). Thus, the instance with the lowest cost transformation (if more than one instance have the same value, the frequency of the instance's structure will be used to break the tie if possible) is considered the anomaly, as it is closest (maximum) to the best substructure without being included on the best substructure's instance list.

3.3.1 Algorithm

The detailed GBAD-MPS algorithm is as follows:

Alg. 3: proc **GBAD-MPS** (graph *G*, cost *C*)

```
1. Find normative substructure $S$ minimizing
   $DL(S)+DL(G|S)$.
2. For each $S_n$, in the set of previously-generated
   substructures, where $S_n \subseteq S$ , let $I_n$ be the set of
   instances of $S_n$.
3. For each instance $I_k$ in the set of instances $I_n$, where
   matchcost($I_k$,$S$) > 0
   a. anomalyScore($I_k$) = $| I_n |$ * matchcost($I_k$,$S$).
4. Return instances $I_k$ having min anomalyScore < $C$.
```

Allowing the user to specify a cost of transformation *C,* we control the amount of “anomalousness” we are willing to accept. By our definition of an anomaly, we expect low transformation costs (to match best substructure). It should be noted that whenever we indicate a relationship between substructures as $x \subseteq y$, we are referring to the fact that x is a sub-graph of y, rather than x is a subset of y.

3.3.2 Example

Consider the slightly more complex graph shown in Figure 3.

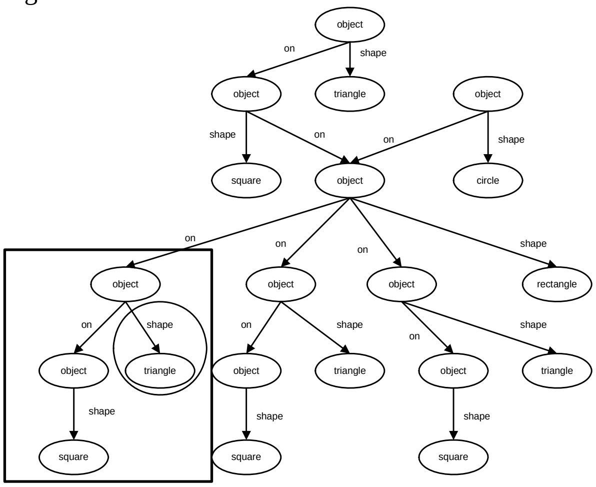

Figure 3. GBAD-MPS example.

Suppose we take one of the instances of the normative pattern (shown in the box), and remove an edge and its associated vertex (shown in the circle). Running GBAD-MPS on the modified graph results in the discovery of an anomalous substructure similar to the normative pattern, but missing the triangle object and its shape link.

4. Synthetic Experiments

For our synthetic experiments, we created graphs using a tool called *subgen* [14] that generates graphs based upon user-specified parameters, including:

- total number of vertices and edges
- list of possible vertex and edge labels and their probabilities
- substructure pattern
- amount of connectivity

Using these parameters, *subgen* computes the number of instances that need to be generated by calculating the size of a graph and dividing by the size of a substructure pattern (i.e., what we want to be the normative pattern). After the graph is built from these instances, randomly-labeled vertices are added in order to achieve the desired graph size, and randomly-labeled edges are added in order to achieve the specified connectivity level. Finally, any additional edges are added in order to achieve the desired graph size.

In order to be consistent across all experiments, we chose a star-cluster pattern as our normative pattern (i.e., a node with connections to several other nodes, and each of those nodes with several connections to other nodes). The choice of this pattern was somewhat arbitrary, but it also resembles many types of real-world data, such as networks, calling trees, and financial transactions. Each synthetic graph consisted of substructures containing a normative pattern (V number of vertices and E number of edges), connected to each other by one or more random connections, and each test consisted of AV number of anomalous vertices and AE number of anomalous edges.

Figure 4 shows the effectiveness of the GBAD-MDL approach. For graphs of varying sizes, from 100 vertices/edges to 10,000 vertices/edges, with a normative pattern consisting of 10 vertices/10 edges, the results were identical across the spectrum. In this figure, the X axis represents the thresholds, the Y axis is the percentage of anomalies discovered, and the Z axis indicates the sizes of the anomalies.

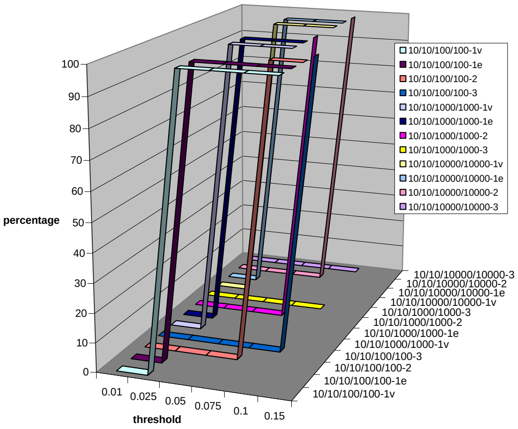

Figure 4. Percentage of GBAD-MDL runs where all anomalies discovered.

As expected, when the threshold is increased to accommodate the size of the anomaly with respect to the normative pattern, the anomalies are discovered 100% of the time. The drawback is that as the threshold is increased, so is the running time of the algorithm, and false positives, like noise, will increase (i.e., the size of the reported anomaly is equal to or smaller than that of the true anomaly).

Without changing any parameters, experiments using GBAD-P and GBAD-MPS resulted in less than a 100% discovery rate across all tests. However, when we increased SUBDUE’s beam width parameter so that GBAD could be provided a larger set of substructure instances to evaluate, the result was a 100% discovery rate. The reason that the number of substructures to evaluate has to be increased is that as the size of the anomaly grows (i.e., the number of vertices and edges inserted or deleted increases), the further away the cost of transformation for the anomalous instance is from the normative pattern. In addition, unlike with the GBAD-MDL tests, there were no false positives reported from any of the GBAD-P or GBAD-MPS synthetic tests. Using varying sizes of normative patterns and anomalies, each approach has shown to be useful at discovering a specific type of anomaly. While the algorithms do not appear to be useful outside of their intended targets, no graphs of any size or any anomaly went undetected by all three approaches.

One of the advantages of these algorithms is that they do not just return the pattern of the anomaly – they also return the actual anomalous instances within the data. In a real-world scenario, that can be invaluable to an analyst who may need to act upon a fraud situation before the losses are too great. The disadvantage of these algorithms is that they are focused on specific anomalies: modifications, insertions or deletions. Thus, in a real-world scenario, it would require that all three algorithms be used in conjunction, as the type of anomaly would most likely be unknown.

5. Real-World Experiments

5.1 Cargo Shipments

One area that has garnered much attention recently is the analysis and search of imports into the United States. A large number of imports into the U.S. arrive via ships at ports of entry along the coast-lines. Thousands of suspicious cargo, whether it be illegal or dangerous, are examined by port authorities every day. Due to the volume, strategic decisions must be made as to which cargo should be inspected, and which cargo will pass customs without incident. A daunting task that requires advanced analytical capabilities to maximize effectiveness and minimize false searches.

Using shipping data obtained from the Customs Border and Protection (http://www.cbp.gov/), we are able to create a graph-based representation of the cargo information where row/column entries are represented as vertices, and labels convey their relationships as edges. Figure 5 shows a portion of the actual graph that we will use in our experiments.

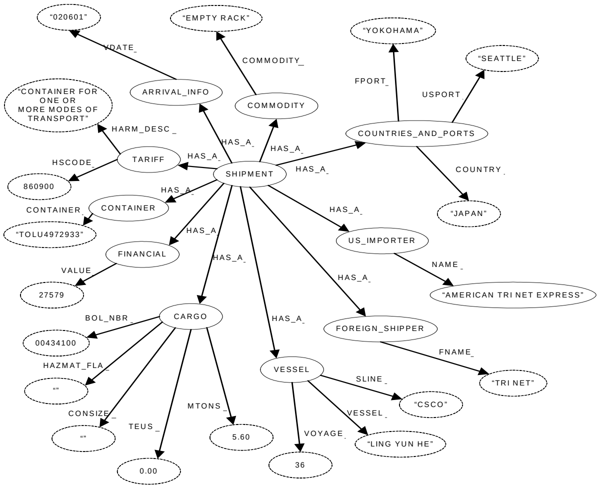

Figure 5. Example graph of cargo information.

While we were not given any labeled data from the CBP, we can draw some results from simulations of publicized incidents. Take for instance the example from a press release issued by the U.S. Customs Service. The situation is that almost a ton of marijuana is seized at a port in Florida [13]. In this drug smuggling scenario, the perpetrators attempt to smuggle contraband into the U.S. without disclosing some financial information about the shipment. In addition, an extra port is traversed by the vessel during the voyage. For the most part, the shipment looks like it contains a cargo of toys and bicycles from Jamaica.

When we run all three algorithms on this graph, GBAD-MDL is unable to find any anomalies, which makes sense considering none of the anomalies are modifications. When the graph contains the anomalous insertion of the extra traversed port, the GBAD-P algorithm is able to successfully discover the anomaly. Similarly, when the shipment instance in the graph is missing some financial information, GBAD-MPS reports the instance as anomalous.

5.2 Intrusion Detection

One of the more applied areas of research when it comes to anomaly detection can be found in the multiple approaches to intrusion detection. The reasons for this are its relevance to the real world problem of networks and systems being attacked, and the ability of researchers to gather actual data for testing their models. The most used data set for this area of research is the 1999 KDD Cup dataset [5].

The KDD Cup data consists of connection records, where a connection is a sequence of TCP packets. Each connection record is labeled as either "normal", or one of 37 different attack types. Each record consists of 31 different features (or fields), with features being either continuous (real values) or discrete. In the 1999 competition, the data was split into two parts: one for training and the other for testing. Groups were then allowed to train their solutions using the training data, and were then judged based upon their performance on the test data. Since the GBAD approach uses unsupervised learning, we will run the algorithms on the test data so that we can judge our performance versus other approaches.

Not surprisingly, each of the algorithms has a different level of effectiveness when it comes to discovering anomalies in intrusion detection data. Using GBAD-MDL, our ability to discover the attacks is relatively successful. Across all data sets, 100% of the attacks are discovered. However, all but the apache2 and worm attacks produce some false positives. 42.2% of the test runs do not produce any false positives, while runs containing snmpgetattack, snmpguess, teardrop and udpstorm attacks contribute the most false positives. False positives are even higher for the GBAD-P algorithm, and the discovery rate of actual attacks decreases to 55.8%. GBAD-MPS shows a similarly bad false positive rate at 67.2%, and a worse discovery rate at 47.8%.

It is not surprising that GBAD-MDL is the most effective of the algorithms, as the data consists of TCP packets that are structurally similar in size across all records. Thus, the inclusion of additional structure, or the removal of structure, is not as relevant for this type of data, and any structural changes, if they exist, would consist of value modifications.

6. Conclusions and Future Work

The three algorithms presented in this paper are able to discover an anomaly when it consists of a small change to the normative pattern. Using the minimum description length principle and probabilistic approaches, we have been able to successfully discover anomalies in graphs and patterns of varying sizes with minimal to no false positives. Results from both synthetic and real-world data demonstrate the effectiveness of the approaches. We are pursuing experiments on other domains that can be represented as graphs, including telecom and social networks. While our results are effective in detecting anomalies in a security area such as cargo shipments, other possible applications of these approaches include post-9/11 terrorist networks and the Enron e-mail datasets (e.g., detecting anomalies in e-mail patterns).

7. References

[1] Chakrabarti, D. *AutoPart: Parameter-Free Graph Partitioning and Outlier Detection*. Knowledge Discovery in Databases: PKDD 2004, 112-124, 2004.

[2] Cook, D. and Holder, L. *Graph-based data mining*. IEEE Intelligent Systems 15(2), 32-41, 1998.

[3] Customs and Border Protection Today, "Illegal textile entries: a way to save a few bucks?", March 2003. (http://www.cbp.gov/xp/CustomsToday/2003/March/illegal.xml)

[4] Hampton, M. and Levi, M. *Fast spinning into oblivion? Recent developments in money-laundering policies and offshore finance centres*. Third World Quartely, Volume 20, Number 3, June 1999, pp. 645-656, 1999.

[5] KDD Cup 1999. Knowledge Discovery and Data Mining Tools Competition. 1999. *(http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html*)

[6] Lin S. and Chalupsky, H. *Unsupervised Link Discovery in Multi-relational Data via Rarity Analysis*. Proceedings of the Third IEEE ICDM International Conference on Data Mining, 171-178, 2003.

[7] Noble, C. and Cook, D. *Graph-Based Anomaly Detection*. Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 631-636, 2003.

[8] Rattigan, M. and Jensen, D. *The case for anomalous link discovery*. ACM SIGKDD Explor. Newsl., 7(2):41-47, 2005.

[9] Rissanen, J. *Stochastic Complexity in Statistical Inquiry*. World Scientific Publishing Company, 1989.

[10] Shetty, J. and Adibi, J. *Discovering Important Nodes through Graph Entropy: The Case of Enron Email Database*. KDD, Proceedings of the 3rd international workshop on Link discovery, 74-81, 2005.

[11] Staniford-Chen, S., Cheung, S., Crawford, R., Dilger, M., Frank, J., Hoagland, J. Levitt, K., Wee, C., Yip, R. and Zerkle, D. *GrIDS – A Graph Based Intrusion Detection System for Large Networks*. Proceedings of the 19th National Information Systems Security Conference, 1996.

[12] Sun, J, Qu, H., Chakrabarti, D. and Faloutsos, C. *Relevance search and anomaly detection in bipartite graphs*. SIGKDD Explorations 7(2), 48-55, 2005.

[13] U.S. Customs Service: *1,754 Pounds of Marijuana Seized in Cargo Container at Port Everglades*. November 6, 2000. (http://www.cbp.gov/hot-new/pressrel/2000/1106-01.htm).

[14] D. Cook and L. Holder. Substructure Discovery Using Minimum Description Length and Background Knowledge. Artificial Intelligence Research, 1:231-255, 1994.