Seminar Report
Classification, Clustering and Application in Intrusion Detection System

Kaushal Mittal 04329024
M.Tech I Year
Under the Guidance of Prof. Sunita Sarawagi
KReSIT, Indian Institute of Technology Bombay

November 15, 2004

Abstract

Classification and clustering techniques in data mining are useful for a wide variety of real time applications dealing with large amount of data. Some of the applications of data mining are text classification, selective marketing, medical diagnosis, intrusion detection systems. Intrusion detection system are software system for identifying the deviations from the normal behavior and usage of the system. They detect attacks using the data mining techniques - classification and clustering algorithms. In this report, I discuss approaches based on classification techniques like naive bayesian classifiers, neural networks and WINNOW based algorithm. Approaches based on clustering techniques like hierarchical and density based clustering have been discussed to emphasize the use of clustering techniques in intrusion detection.

1 Introduction

Classification techniques analyze and categorize the data into known classes. Each data sample is labeled with a known class label. Clustering is a process of grouping objects resulting into set of clusters such that similar objects are members of the same cluster and dissimilar objects belongs to different clusters. In classification, the classes and number of classes is predefined. Training examples are used to create a model, where each training sample is assigned a predefined label. This is not the case with clustering. Classification techniques are examples of supervised learning and clustering techniques are examples of unsupervised learning.

Intrusion detection systems are softwares used for identifying the intentional or unintentional use of the system resources by unauthorized users. They can be categorized into misuse detection systems and anomaly detection systems. Misuse detection systems model attacks as a specific pattern and are more useful in detecting known attack patterns. Anomaly detection systems are adaptive systems that distinguish the behavior of the normal users from the other users. The misuse detection systems can detect specific types of attacks but are not generalized. They cannot detect new attacks until trained for them. On the other hand, anomaly detection systems are adaptive in nature, they can deal with new attacks, but they cannot identify the specific type of attacks. If the intrusion occurs during learning, then the anomaly detection system may learn the intruders behavior and hence may fail. Being more generalized and having a wider scope as compared to misuse detection systems, most of the current research focus on anomaly detection systems.

Data mining approaches can be applied for both anomaly and misuse detection. The data sample are a set of system properties, representing the behavior of the system/user. Classification techniques are used to learn a model using the training set of data samples. The model is used to classify the data samples as anomalous behavior instance or the normal behavior

instance. Clustering techniques can be used to form clusters of data samples corresponding to the normal use of the system. Any data sample with characteristics different from the formed clusters is considered to be an instance of anomalous behavior. Clustering based techniques can detect new attacks as compared to the classification based techniques.

A number of classification and clustering algorithms can be used for anomaly detection. [**?**] proposes the use of bayesian classifiers to learn a model that distinguishes the behavior of intruder from the normal users behavior. [**?**] proposes hierarchical clustering based algorithm for anomaly detection on network. [**?**] proposes the WINNOW based algorithm for anomaly detection. [**?**] proposes the use of neural networks and [**?**] proposes the use of density based clustering for anomaly detection.

Rest of the report is organized as follows: Section 2 discusses the bayesian classifiers and neural network based classification. Section 3 discusses the hierarchical and density based clustering. Section 4 discusses the anomaly detection approach based on WINNOW based algorithm and the use of the classification and clustering algorithms discussed in section 2 and section 3, for anomaly detection. Section 5 gives the conclusion.

2 Classification Techniques

In Classification, training examples are used to learn a model that can classify the data samples into known classes. The Classification process involves following steps:

1. Create training data set.
2. Identify class attribute and classes.
3. Identify useful attributes for classification (relevance analysis).
4. Learn a model using training examples in training set.
5. Use the model to classify the unknown data samples.

A variety of classification techniques viz. decision tree induction, bayesian classification, bayesian belief networks, neural networks etc. are used in data mining based applications. In this section, I discuss the naive bayesian classfiers and neural networks.

2.1 Naive Bayesian Classifiers

Naive bayesian classifiers use the bayes theorem to classify the new instances of data. Each instance is a set of attribute values described by a vector, $X = (x_1, x_2 \ldots, x_n)$. Considering m classes, the sample X is assigned to the class C_i if and only if

$$P(X|C_i)P(C_i) > P(X|C_j)P(C_j)$$

for all j in $(1, m)$ such that $j \neq i$.
The sample belongs to the class with maximum posterior probability for the sample. For categorical data $P(X_k|C_i)$ is calculated as the ratio of frequency of value x_k for attribute A_k and the total number of samples in the training set. For continuous valued attributes a guassian distribution can be assumed without any loss of generality.

In naive bayesian approach the attributes are assumed to be conditionally independent. Inspite of this assumption, naive bayesian classifiers give satisfactory results because focus is on identifying the classes for the instances, not the exact probabilities. Application like spam mail classification, text classification can use naive bayesian classifiers. Theoretically, bayesian classifiers are least prone to errors. The limitation is the requirement of the prior probabilities. The amount of probability information required is exponential in terms of number of attribute, number of classes and the maximum cardinality of attributes. With increase in number of classes or attributes, the space and computational complexity of bayesian classifiers increases exponentially.

2.2 Neural Networks

An artificial neural network consists of connected set of processing units. The connections have weights that determines how one unit will affect other. Subset of such units act as input nodes, output nodes and remaining nodes constitute the hidden layer. By assigning activation to each of the input node, and allowing them to

propagate through the hidden layer nodes to the output nodes, neural network performs a functional mapping from input values to output values. The mapping is stored in terms of weights over connection.

Backpropagation network are simple feed forward neural networks. Input is submitted to the network and the activation at each level are cascaded forward, ending up with activations at output nodes. During training backpropagation algorithm [**?**] is used to tune the values of weights over connection. The error at the output layer is calculated and backpropagated. This feedback is used at intermediate level to readjust the weights. Performance of training phase depends on the learning rate used for adjusting the weights. Too small value of learning rate makes learning very slow. Conversely, too large value may result in oscillation of weights between wrong values and network may take long time to learn. The training stops when weights tend to converge or the network is able to classify the samples correctly. After training the backpropagation network can be used as a model for classification of new instances.

They are adaptive in nature, tolerant to noisy data and can classify instances for which they are not trained. However, training may take a long time and is an irreversible process. Also the knowledge representation in neural networks is not directly interpretable by humans.

3 Clustering Techniques

Clustering involves unsupervised learning - number of classes and the classes are not known in prior. In this section I discuss the hierarchical and density based approaches for clustering.

3.1 Hierarchical Clustering Algorithms

These algorithms group the data into a tree of clusters forming a hierarchical structure. The clusters are merged or split based on some distance measurement that accounts for the similarity or difference between the samples respectively. The distance can be euclidean distance, mean distance, maximum distance, average, centroid etc. The number of clusters acts as a parameter for restricting the level of clustering. Clustering stops when the required number of clusters have been formed or depth of the clustering tree has reached to a specified value. Hierarchical clustering algorithms can be categorized into:

- Agglomerative algorithms - based on bottom up approach.
- Divisive algorithms - based on top down approach.

3.1.1 Agglomerative Algorithms

These algorithms initially assign each sample to be a separate cluster. The clusters with the least distance are merged to get larger clusters till the termination condition is satisfied or a single cluster is left.

1. **BIRCH - Balanced Iterative Reducing and Clustering using Hierarchies.**
   In BIRCH, summary of statistics of the cluster, called as cluster feature, is calculated for each sub-cluster consisting on n samples. A height balanced CF(clustering feature) tree is dynamically constructed with samples as the leaf nodes CF of the children as non leaf nodes. A sample is kept in the closest leaf node. When the size of leaf node becomes larger than the threshold, the node splits and the CF is recalculated for the individual nodes and updated in the tree. The complexity of the algorithm is $O(n)$ where n is the number of objects to be clustered. It is known to generate the best clusters with the available resources but do not work well if the clusters are not spherical in shape (because it use the notion of radius to consider the boundary of the cluster).

2. **CURE - Clustering using Representatives.**
   This algorithm works well for non-spherical shaped clusters also. In CURE cluster is represented by a set of representative points generated by randomly selecting the scattered points in the cluster and shrinking them by fractions to reach the cluster center. The two clusters with the closest pair

of representative points are merged. More than one representative points allows non-spherical shaped clusters. However, the aggregate interconnectivity is ignored and hence the categorical attributes cannot be handled.

3.2 Density based Clustering Algorithms

The set of samples forming a dense region are treated as clusters. Since the clusters are based on density, not on distance, they need not necessarily be spherical.

3.2.1 DBSCAN - Density based Spatial Clustering of Application with Noise.

This algorithm identify regions with sufficiently high density as clusters. All the samples within the radius ϵ form ϵ-neighbourhood of the sample. ϵ-neighbourhood for each sample points is called a core group i.e. initial cluster. All the objects that are density reachable, connected or directly density reachable are merged to form larger clusters. This continues till no more merging of clusters is possible. If spatial indexes are used, the complexity of DBSCAN algorithm is $O(nlogn)$, otherwise it is $O(n^2)$. The algorithm is useful for applications with spatial databases and noise.

4 Intrusion Detection Systems

This section briefly discusses the data mining approaches proposed for intrusion detection system. The characteristics of a good intrusion detection system are:

1. High detection rate.
2. Less false alarms.
3. Less CPU cycles.
4. Quick detection of intrusion.

The user profile, system behavior comprising of the statistics related to the network, CPU, memory, processes, softwares and applications used by the users constitute the test data for the intrusion detection system. A large number of system tools and utilities plugged in with operating systems can be used to collect this data. For windows operating system application like perfmon and netstat are used whereas for Linux systems top, tcpdump, strace, etc are used.

Rest of the section discusses the approaches for anomaly detection based on the classification and clustering techniques described in section 2 and section 3.

4.1 Naive Bayesian Approach

[**?**] proposes the use of naive bayesian approach for anomaly detection in systems with windows operating system. [**?**] measure around 1500 features every second. Features are specific system properties like average CPU utilization, average of last 10 values of data transfer rate, memory utilisation, number of processes etc. A data sample comprises of values for each of these 1500 features. The problem is defined as to classify each data sample into anomalous or normal category.

[**?**] assume that the features are conditionally independent given the category. The training set is used to calculate the prior probabilities. The prior probabilities are used to calculate the probability of obtaining the current measurement, given the possible categories. The current samples is assigned a label for which the probability calculated is maximum. [**?**] conducted experiments over test data and found the detection rate of this approach to be 57.8%. The low detection rate is on account of the assumption that the features are conditionally independent.

4.2 Neural Network

[**?**] proposes the use of neural networks for anomaly detection. The approach consists of maintaining a database of a sequence of system calls made by each program to the operating system, used as the signature for the normal behavior. If online sequence of system calls for a program differ from the sequence in the database anomalous behavior is registered. If significant percentage of sequences do not match then alarm for intrusion is raised.

Backpropagation network is trained with training set of sequences of system calls. A leaky bucket algorithm is used to capture the temporal locality of the anomalous sequences. When closely related anomalous sequences are faced,

counter gets a large value and when a normal sequence is obtained the counter gradually drops down to zero. This leads to intrusion detection only when a lot of similar anomalous sequences are obtained, thereby representing the behavior of intruder.

4.3 Hierarchical Clustering

[**?**] proposes the use of graph clustering for intrusion detection over networks. The approach consists of using agglomerative clustering to form clusters of nodes communicating extensively with each other. The nodes or systems on the network can be represented by a graph. In the graph, nodes represent the systems and edges with weight represent the amount of data exchange between the system corresponding the nodes linked by the edge. This graph is decomposed into the number of clusters such that nodes within each cluster exchange data with other nodes in the cluster extensively.

Feature vector consisting of values for features like degree of nodes, average outgoing traffic etc. is calculated. A neural network is used to learn the mapping from these feature values to the normal behavior or anomalous behavior. If the intruder use the system, the traffic over the network changes, resulting in the change of the feature values, leading to the detection by the neural network as anomalous behavior.

4.4 Density based Local Outliers

[**?**] proposes a density based clustering approach for anomaly detection. Data sample corresponding to the anomalous behavior is considered as an outlier. This approach assigns a LOF (Local outlier factor) to each data sample. Greater the LOF, greater is the probability of sample being an outlier. The k-distance is computed for the k^{th} nearest neighbor for each sample. DBSCAN algorithm is used to find k-neighbourhood for the data sample and form clusters of samples corresponding to the normal behavior. Large value of LOF for a data sample indicates it is distant from the clusters of samples corresponding to normal behaviour. Hence the sample is an outlier.

This approach does not requires tuning and is adaptive in nature. Most of the techniques discussed above learn the behavior of a specific user and detect deviation from that behaviour as anomalous. For systems with multiple valid users, the requirement is to consider the behaviour of each of the N valid users as normal and the remaining users as anomalous. For such requirements this approach is useful. It creates density based clusters corresponding to the behavior of the N users. Any sample not resembling the behaviour of any of the N users will lie outside the cluster and will be considered as outlier.

4.5 WINNOW based Algorithm

[**?**] proposes a WINNOW based algorithm for anomaly detection. Most of the approaches discussed above did not achieved high detection rates. Experimental results proves that this approach can achieve detection rate of about 95% with less than one false alarm per day. The data collected is same as described in subsection **??**. Perfmon is used to measure around 200 different properties and 1500 different features corresponding to these properties. Each data sample is a vector containing the value of these 1500 features.

The algorithm consists of three phases.

4.5.1 Training Phase

The data samples are collected corresponding to the normal user and an equal number of samples corresponding to the intruder (Any user other than the normal user). The values for most of the features are continuous. They are discretized into ten bins. The values of a feature are classified into ten bins by fitting the standard distribution functions like uniform, guassian, exponential and erlang. The distribution function for which the root mean square error is minimum represents the probability distribution for the feature. The probability distribution is obtained by normalizing the frequency in each bin with the total count of samples. [**?**] propose to use the WINNOW based algorithm to assign weights to each feature to model the normal behaviour.

The WINNOW based algorithm is as follows

1. Initialize the weights for each feature, w_f, as 1.

2. For each training sample

   (a) Initialize votes_for and votes_against to be 0.

   (b) For all features, if relative probability of the feature is less than the constant r, add the weight w_f to votes_for otherwise add it to votes_against.

   (c) If $votes\_for > votes\_against$ then the measurement is anomalous.

   (d) If sample corresponds to normal user and is considered anomalous, the weights of all features that voted for raising alarm are reduced to half of their current values. Conversely, if anomalous sample is treated as normal, the weights of all features that voted against raising the alarm are reduced to half of their current values.

4.5.2 Tuning Phase

Tuning data consists of data samples from the normal user and the intruders(other users). This phase involves calculation of three system parameters W - the window size, $Thresh_{mini}$ and $Thresh_{full}$. For different combinations of these parameters, following steps are executed:

1. The feature values of test sample collected each second, vote for mini alarms. If the ratio of votes_for and votes_against is greater than $Thresh_{mini}$, then a mini alarm is raised.

2. If number of mini alarms in last W seconds is greater than $Thresh_{full}$ then raise an alarm signaling intrusion. After each such alarm wait for W seconds to avoid the samples from overlapping.

The goal is to select the value of these parameters to maximize the intrusion detection rate and minimize the false alarms.

4.5.3 Operation Phase

In this phase the learned statistical model, along with the values of the parameters $W, Thresh_{mini}$ and $Thresh_{full}$ are used to detect anomalous behaviour. The system can retrain and retune to adjust to the changing behavior of the normal user. Even during this phase WINNOW algorithm has to be used to adjust to the changing behavior of the user, otherwise false alarm rate will increase and intrusion detection rate will gradually drop.

4.5.4 Experimental Evaluation and Analysis

[**?**] have conducted experiments to analyze the performance of the WINNOW based algorithm proposed. The analysis show that if tuning parameters are carefully selected, the intrusion detection rate reaches 95% with less than one false alarm per day. The tuning parameter W is to be selected carefully. Too small the value of W larger the number of false alarms, due to overlapping of the samples with the samples that have already raised an alarm. As W increases, the false alarm rate decreases, but intruder gets more time to use the system before being detected. The system can adapt itself to learn the changing behavior of the normal user. If the tuning parameters are wrongly selected, the system may learn the intruder's behavior also.

5 Conclusion

Intrusion detection systems are one of the key areas of application of data mining techniques. Naive bayesian classifiers though performs well for most of the applications inspite of the assumption of conditional independency, does not provide good results for intrusion detection systems. Clustering techniques can be used for intrusion detection, as they can detect unknown attacks also. They are useful for misuse detection as well as anomaly detection systems.

WINNOW based algorithm provides higher detection rates and lower false alarm rates as compared to the other approaches discussed. The system involves less CPU cost. The only costly phase is the tuning phase. Oftenly, intrusion or misuse of system can be best described by excessive usage of resources and events that do not occur frequently. For eg. too many print jobs, downloads etc. The system assumes the samples collected from all other users, except the normal user as samples of anomalous behavior. In practice, this may not be a good representa-

tive set for anomalous behavior. It is necessary to test the system with data consisting of real data samples corresponding to intrusive behavior.

References

[1] Jude Shavlik and Mark Shavlik, *Selection, Combination, and Evaluation of Effective Software Sensors for Detecting Abnormal computer Usage*, KDD 2004, Seattle, Washington, USA., 2004.

[2] A. Lazarevic, L. Ertoz, A. Ozgur, J. Srivastava and V. Kumar, *A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection*, Proc. SIAM Conf. Data Mining, 2003.

[3] A. Ghosh, A. Schwartzbard and M. Schatz, *Learning Program Behavior Profiles for Intrusion Detection*, USENIX Workshop on Intrusion Detection and Network Monitoring, April 1999.

[4] J. Tolle and O. Niggemann, *Supporting Intrusion Detection by Graph Clustering and Graph Drawing*, RAID 2000.

[5] Tom M. Mitchell, *Machine Learning*, McGraw-Hill, International Edition 1997.

[6] Jiawei Han and Micheline Kamber, *Data Mining Concepts and Techniques*, Morgan Kaufmann, 2001.