A Report on

Anomaly Detection using Gaussian Multivariate System

Dual Degree in

COMPUTER SCIENCE AND ENGINEERING

By

Rajnish Kumar Ranjan

15JE001619

SESSION: 2017-2018

UNDER THE GUIDANCE OF

Dr. Rajendra Pamula

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY

(INDIAN SCHOOL OF MINES)

DHANBAD-826004

Acknowledgements

I would like to express the deepest appreciation to my project mentor Mr. Praphula Kumar Jain, who has the attributes. He continually and convincingly conveyed a spirit of adventure in regard to the project, and an excitement in regard to the teaching.

I am also thankful to Dr. Rajendra Pamula for his invaluable support and encouragement as well as all faculty member of the department for providing time to time guidance during this period.

Finally, I must say that no height is ever ever achieved without some sacrifices made at some end and it is here where I owe our special debt to our parents and our friends for showing their generous love and care throughout the entire period of time.

Rajnish Kr. Ranjan
Computer Science & Engineering.
IIT(ISM) Dhanbad

Abstract

Anomalies are patterns in data that do not conform to a well-defined notion of normal behavior. One-class Support Vector Machines calculate a hyperplane in the feature space to distinguish anomalies, but the false positive rate is always high and parameter selection is a key issue. So, we propose a novel one-class framework for detecting anomalies, which takes the advantages of both boundary movement strategy and the effectiveness of evaluation algorithm on parameters optimization.

First, we search the parameters by using a particle swarm optimization algorithm. Each particle suggests a group of parameters, the area under receiver operating characteristic curve is chosen as the fitness of the object function. Second, we improve the original decision function with a boundary movement. After the threshold has been adjusted, the final detection function will bring about a high detection rate with a lower false positive rate.

Table of Contents

1. Introduction

Anomaly detection refers to the problem of finding patterns in data that do not conform to expected behaviour. These nonconforming patterns are often referred to as anomalies, outliers, discordant observations, exceptions, aberrations, surprises, peculiarities, or contaminants in different application domains. With the development of information technology, vast quantities of data are captured and stored. The capacity, dimensions, and complexity of database have grown rapidly, but usually the real data set could not be used directly for data mining due to ignorance, human errors, rounding errors, transcription factor, instrument malfunction, and biases. Anomaly detection is used to find the objects that do not comply with the general behaviour of the data and then lead to potentially useful information. Anomalies can be translated to significant or critical actionable information. Anomaly detection can be applied to many fields, such as credit card fraud detection, security systems, medical diagnosis, network intrusion detection, and information recovery.

Simply, anomaly detection is the task of defining a boundary around normal data points so that they can be distinguishable from outliers. But several different factors make this notion of defining normality very challenging. E.g. normal behaviour usually evolve in certain domains and the notion that is considered normal in the present could change in future. Moreover, defining the normal region which separates outliers from normal data points is not straightforward in itself.

Here, we will implement anomaly detection algorithm (in Python) to detect outliers in computer servers.To keep things simple we will use two features 1) throughput in mb/s and 2) latency in ms of response for each server. The Gaussian model will be used to learn an underlying pattern of the dataset with the hope that our features follow the gaussian distribution. After that, we will find data points with very low probabilities of being normal and hence can be considered outliers. For training set, we will first learn the gaussian distribution of each feature for which mean and variance of features are required. Numpy provides the method to calculate both mean and variance (covariance matrix) efficiently. Similarly, Scipy library provide method to estimate gaussian distribution.

2. Motivation

Today anomaly detection methods are of major interest to the world and are used in very different and various domains like computer intrusion detection, credit card and telephone fraud detection, spam detection, and so on. Here, we have introduced a new unsupervised method for anomaly detection, based on a combination of a Self-Organizing Map and Particle Swarm Optimization that fuse information from various sources. It is a simple, time and space consuming method that can be used in different domains. In this paper, we wished to implement it for crisis management, so we chosed forest fires and their detection. In comparison with some other methods, like the Self-Organizing Map, Dempster–Shafer and Bayesian Estimation, we obtained good results. Like other anomaly detection methods, when abnormal cases are rare, our suggested method has better results than when they are not.

Forest fires are a major environmental issue, creating economical and ecological damage while endangering human lives. Fast detection is a key element for controlling such phenomenon. To achieve this, one alternative is to use automatic tools based on local sensors, such as provided by meteorological stations. In effect, meteorological conditions (e.g. temperature, wind) are known to influence forest fires and several fire indexes, such as the forest Fire Weather Index (FWI), use such data. In this work, we explore a DataMining (DM) approach to predict the burned area of forest fires. Five different DM techniques, e.g. Support Vector Machines (SVM) and Random Forests, and four distinct feature selection setups (using spatial, temporal, FWI components and weather attributes), were tested on recent real-world data collected from the northeast region of Portugal.

3. Related Work

Anomaly detection systems work by trying to identify anomalies in an environment.

At the early stage, the research focus lies in using rule-based expert systems and statistical approaches. But when encountering larger datasets, the results

of rule-based expert systems and statistical approaches become worse. Thus, many data mining techniques have been introduced to solve the problem. Among these techniques, the Artificial Neural Network (ANN) is widely used and has been successful in solving many complex practical problems.

Some discussed a generic method for anomaly detection that could be used in different areas, but others are about an exclusive subject. Here, we try to introduce their significance.

Some believe unsupervised methods are the best choice for anomaly detection, since they do not need any previous knowledge and only try to find anomalous patterns and cases. While supervised methods can detect only pre-known abnormal cases, unsupervised methods can recognize new and unknown objects. Rouil et al. , Eskin et al. and Zakia and Akira tried to express some unsupervised methods for anomaly detection. Also, Guthrie et al. tried to develop an anomaly detection method for finding anomalous segments in a document. Their method is unsupervised; they assumed that there is no data with which to characterize "normal" language. This method is not a classification or clustering method. The method returns a list of all segments ranked, by how anomalous they are with respect to the whole document.

Meanwhile, results show that data fusion methods have good results in this area. Chen and Aickelin have constructed a Dempster–Shafer based anomaly detection system using the Java 2 platform. First, they used the Wisconsin Breast Cancer Dataset (WBCD) and then the Iris plant dataset, for their experiments. Thirdly, they experimented using an e-mail dataset, which had been created using a week's worth of e-mails (90 e-mails) from a user's sent box, with outgoing e-mails (42 e-mails) sent by a computer infected with the netsky-d worm. The aim of the experiment was to detect the 42 infected e-mails. They used D–S to combine features of the e-mails to detect the worm infected e-mails.

Some various intelligent approaches have also been used for anomaly detection, one of which is the artificial immune system. Greensmith et al. represented a new algorithm for anomaly detection, based on simulation of the human immune system. According to the authors claim, the algorithm performs well on the task of detecting a ping-based port scan and may also be applied to other detection or data correlation problems, such as the analysis of radio signal data from space, sensor networks, internet worm detection and other security and defense applications. Another experiment in this field has undertaken by Twycross and Aickelin.

Artificial neural networks are another intelligent method used for anomaly detection. Brause et al. used a compound method based on rule-based systems and an artificial neural network for credit card fraud detection. Other neural network-based credit card fraud detection has been undertaken by Hassibi, Dorronsoro et al. and Syeda et al. Wang et al. proposed a new approach called FC-ANN based on ANN and fuzzy clustering to solve the problem and help IDS achieve a higher detection rate.

An important part of anomaly detection methods is focused on computer intrusion detection.

The task of an intrusion detection system is to protect a computer system by detecting and diagnosing attempted breaches of the integrity of the system.

The scope of this review will encompass core methods of CI, including artificial neural networks, fuzzy systems, evolutionary computation, artificial immune systems, swarm intelligence and soft computing.

4. Data-Set

Data-set (Throughput vs Latency) for all three kind of servers is given in the excel sheets is given in tr_server_data.csv, gt_server_data.csv and cv_server_data.csv.

1. tr_server_data.csv = Data for technical reporting servers.
2. Gt_server_data.csv = data for game tracking servers.
3. Cv_server_data.csv = Data for document transfer servers.

5. Model Formulation

Here, we will implement anomaly detection algorithm (in Python) to detect outliers in computer servers. This algorithm is dissuced by Andrew Ng in his course of Machine Learning on Coursera. To keep things simple we will use two features 1) throughput in mb/s and 2) latency in ms of response for each server. The Gaussian model will be used to learn an underlying pattern of the dataset with the hope that our features follow the gaussian distribution. After that, we will find data points with very low probabilities of being normal and hence can be considered outliers. For training set, we will first learn the

gaussian distribution of each feature for which mean and variance of features are required. Numpy provides the method to calculate both mean and variance (covariance matrix) efficiently. Similarly, Scipy library provide method to estimate gaussian distribution.

By first importing requried libraries and defining functions for reading data, mean normalizing features and estimating gaussian distribution. We will apply multivariate Gaussian distribution.

In probability theory and statistics, the **multivariate normal distribution** or **multivariate Gaussian distribution** is a generalization of the one-dimensional (univariate) normal distribution to higher dimensions. One definition is that a random vector is said to be k-variate normally distributed if every linear combination of its k components has a univariate normal distribution. Its importance derives mainly from the multivariate central limit theorem. The multivariate normal distribution is often used to describe, at least approximately, any set of (possibly) correlated real-valued random variables each of which clusters around a mean value.

```
import matplotlib.pyplot as plt

import numpy as np

from numpy import genfromtxt

from scipy.stats import multivariate_normal
```

```
def read_dataset(filePath,delimiter=','):
    return genfromtxt(filePath, delimiter=delimiter)


def feature_normalize(dataset):
    mu = np.mean(dataset,axis=0)
```

```
    sigma = np.std(dataset,axis=0)
    return (dataset - mu)/sigma


def estimateGaussian(dataset):
    mu = np.mean(dataset, axis=0)
    sigma = np.cov(dataset.T)
    return mu, sigma


def multivariateGaussian(dataset,mu,sigma):
    p = multivariate_normal(mean=mu, cov=sigma)
    return p.pdf(dataset)
```

In probability theory and statistics, the **multivariate normal distribution** or **multivariate Gaussian distribution** is a generalization of the one-dimensional (univariate) normal distribution to higher dimensions. One definition is that a random vector is said to be k-variate normally distributed if every linear combination of its k components has a univariate normal distribution. Its importance derives mainly from the multivariate central limit theorem. The multivariate normal distribution is often used to describe, at least approximately, any set of (possibly) correlated real-valued random variables each of which clusters around a mean value.

```
def selectThresholdByCV(probs,gt):
    best_epsilon = 0
    best_f1 = 0
    f = 0
    stepsize = (max(probs) - min(probs)) / 1000;
    epsilons = np.arange(min(probs),max(probs),stepsize)
```

```
    for epsilon in np.nditer(epsilons):
        predictions = (probs < epsilon)
        f = f1_score(gt, predictions, average = "binary")
        if f > best_f1:
            best_f1 = f
            best_epsilon = epsilon
    return best_f1, best_epsilon

tr_data = read_dataset('tr_server_data.csv')
cv_data = read_dataset('cv_server_data.csv')
gt_data = read_dataset('gt_server_data.csv')

n_training_samples = tr_data.shape[0]
n_dim = tr_data.shape[1]

plt.figure()
plt.xlabel("Latency (ms)")
plt.ylabel("Throughput (mb/s)")
plt.plot(tr_data[:,0],tr_data[:,1],"bx")
plt.show()



mu, sigma = estimateGaussian(tr_data)
p = multivariateGaussian(tr_data,mu,sigma)
```

```
p_cv = multivariateGaussian(cv_data,mu,sigma)

fscore, ep = selectThresholdByCV(p_cv,gt_data)

outliers = np.asarray(np.where(p < ep))



plt.figure()

plt.xlabel("Latency (ms)")

plt.ylabel("Throughput (mb/s)")

plt.plot(tr_data[:,0],tr_data[:,1],"bx")

plt.plot(tr_data[outliers,0],tr_data[outliers,1],"ro")


plt.show()
```

Next, define a function to find the optimal value for threshold (epsilon) that can be used to differentiate between normal and anomalous data points. For learning the optimal value of epsilon we will try different values in a range of learned probabilities on a cross-validation set. The f-score will be calculated for predicted anomalies based on the ground truth data available. The epsilon value with highest f-score will be selected as threshold i.e. the probabilities that lie below the selected threshold will be considered anomalous.

We have all the required pieces, next let's call above defined functions to find anomalies in the dataset. Also, as we are dealing with only two features here, plotting helps us visualize the anomalous data points.

6. Proposed Work

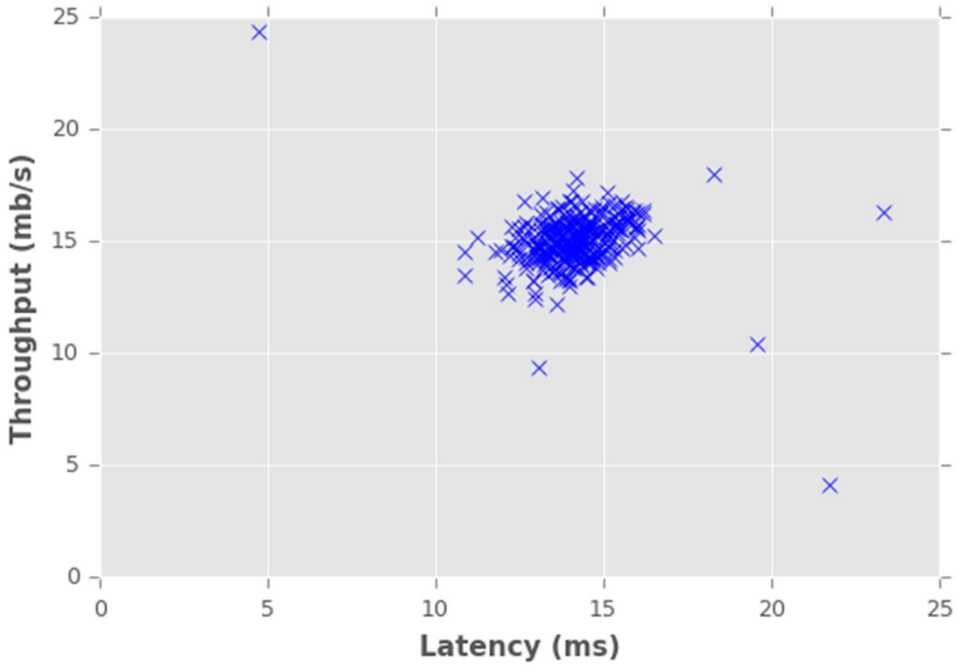
For each category of anomaly detection techniques, we have identified a unique assumption regarding the notion of normal and anomalous data. When applying a given technique to a particular domain, these assumptions can be used as

guidelines to assess the effectiveness of the technique in that domain. Ideally, a comprehensive survey on anomaly detection should allow a reader to not only understand the motivation behind using a particular anomaly detection technique, but also provide a comparative analysis of various techniques. But the current research has been done in an unstructured fashion, without relying on a unified notion of anomalies, which makes the job of providing a theoretical understanding of the anomaly detection problem very diffi- cult. A possible future work would be to unify the assumptions made by different techniques regarding the normal and anomalous behavior into a statistical or maTo Appear in ACM Computing Surveys, 09 2009. 54 · Chandola, Banerjee and Kumar chine learning framework.

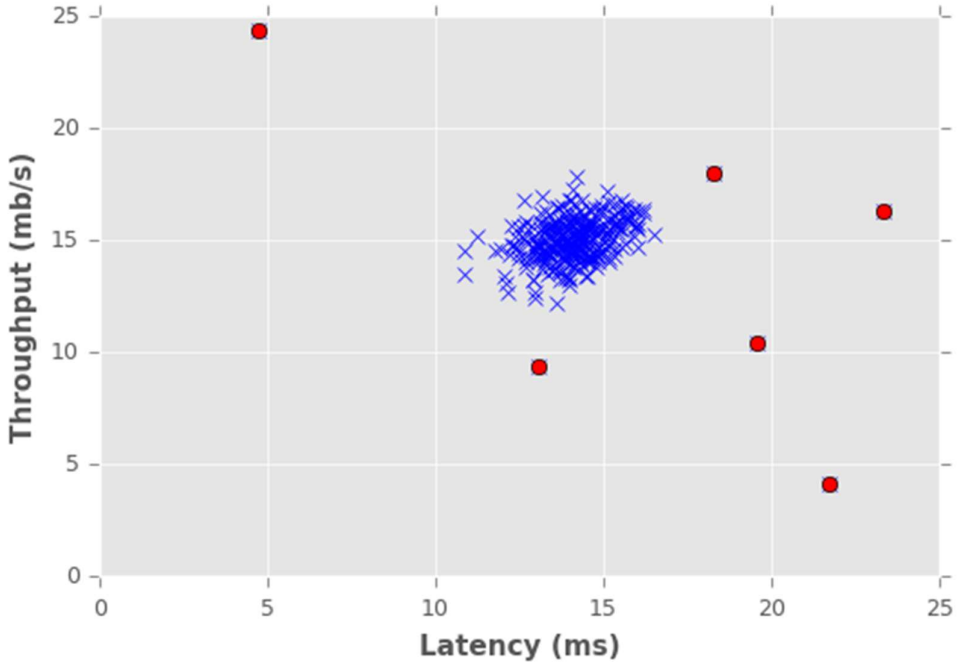
There are several promising directions for further research in anomaly detection. Contextual and collective anomaly detection techniques are beginning to find increasing applicability in several domains and there is much scope for development of new techniques in this area. The presence of data across different distributed locations has motivated the need for distributed anomaly detection techniques.

Another upcoming area where anomaly detection is finding more and more applicability is in complex systems. An example of such system would be an aircraft system with multiple components. Anomaly detection in such systems involves modeling the interaction between various components.

7. Result



The plot shows throughput in mb/s in y-axis, while latency in ms of response for each server in y-axis.



Wherever throughput delivered is not as expected to the latency in the server, anomaly detected.

Each of those point is coloured by red.

8. Conclusion

Today anomaly detection methods are of major interest to the world and are used in very different and various domains like computer intrusion detection, credit card and telephone fraud detection, spam detection, and so on. Here, we have introduced a new unsupervised method for anomaly detection, based on a combination of a Self-Organizing Map and Particle Swarm Optimization that fuse information from various sources. It is a simple, time and space consuming method that can be used in different domains. In this paper, we wished to implement it for crisis management, so we chosed forest fires and their detection. In comparison with some other methods, like the Self-Organizing Map, Dempster–Shafer and Bayesian Estimation, we obtained good results. Like other anomaly detection methods, when abnormal cases are rare, our suggested method has better results than when they are not.

We have implemented this method in various domains and wish to investigate its results in some others

Reference

1. S. Anurag, W.O. Christian**Performance comparison of particle swarm optimization with traditional clustering algorithms used in self-organizing map**

   International Journal of Computational Intelligence, 5 (1) (2009), pp. 32-41

2. Wu Shelly Xiaonan, W. Banzhaf**The use of computational intelligence in intrusion detection systems: a review**

   Review Article Applied Soft Computing, 10 (1) (2010), pp. 1-35

3. Xiaojin, Zhu “Semi-supervised learning literature survey”, Computer Sciences TR 1530, University of Wisconsin–Madison, Last modified on July 19 (2008).

4. D. Swagatam, A. Ajith, K. Amit**Automatic kernel clustering with a multi-elitist particle swarm optimization algorithm**

   Pattern: Recognition Letters, 29 (5) (2008), pp. 688-699

5. Chen, Q. and Aickelin, U. Dempster–Shafer for anomaly detection, *Proceedings of the International Conference on Data Mining DMIN 2006*, Las Vegas, USA, pp. 232–238 (2006).

6. Gang Wang, Jinxing Hao, Jian Ma, Lihua Huang**A new approach to intrusion detection using artificial neural networks and fuzzy clustering**

   Original Research Article Expert Systems with Applications, 37 (9) (2010), pp. 6225-6232ArticlePDF (531KB)

7. Rouil, R., Chevrollier, N. and Golmie, N. "Unsupervised anomaly detection system using next-generation router architecture", *Military Communication Conference MILCOM*, USA (2005).

8. E. Eskin, A. Arnold, M. Prerau, L. Portnoy, S. Stolfo**A geometric framework for unsupervised anomaly detection: detecting intrusions in unlabeled data**

   Data Mining for Security Applications, Kluwer (2002)

9. Zakia, F. and Akira, M. "Unsupervised outlier detection in time series data", *Proceedings of the Second International Special Workshop on Databases for Next-Generation Researchers SWOD2006*, Atlanta, GA, pp. 51–56 (Apr. 2006).

10. Guthrie, D., Guthrie, L., Allison, B. and Wilks, Y. "Unsupervised anomaly detection", *IJCAI 2007*, pp. 1624–2162 (2007).

11. Chatzigiannakis, V., Androulidakis, G., Pelechrinis, K., Papavassiliou, S. and Maglaris, V. "Data fusion algorithms for network anomaly detection: classification and evaluation", *Proceedings of the Third International Conference on Networking and Services*, pp. 50–51 (2007).
12. Yu, D. and Frincke, D. "Alert confidence fusion in intrusion detection systems with extended Dempster–Shafer theory", ACM-SE 43: *Proceedings of the 43rd Annual Southeast Regional Conference*, 2, pp. 142–147 (2005).

13. Te-Shun, C., Sharon, F., Wei, Z., Jeffrey, F. and Asad, D. "Intrusion aware system-on-a-chip design with uncertainty classification", *The 2008 International Conference on Embedded Software and Systems-ICESS* (2008).

14. Siaterlis, C., Maglaris, B. and Roris, P. "A novel approach for a distributed denial of service detection engine", National Technical University of Athens, Athens, Greece (2003).