Rapport du stage M1

Une approache de classification consensuelle
pour la détection de fraudes

You Jiang

Supervisor: Zahia Guessoum

Université Pierre et Marie Curie
Sept 2017

Contents

Introduction

0.1 About the internship

In the summer of 2017, I've effected an internship at the end of Master 1. The topic proposed by the supervisor Zahia Guessoum is called "Une approache de classification consensuelle pour la détection de fraudes". The objectif of this intern is to make a survey on the fraud detection area, and try to implement a detection system. At the beginning, I've read some scientific surveys on this topic to have a general idea of this field. The frauds detection contains many sub-field such as credit card fraud detection, telecommunication fraud detection, or insurance fraud detection etc. I've write a report to resume the most common type of frauds, and the techniques applied in these fields. In this intern, I concentrate myself in the subject called "Online auction fraud detection". After the first glance on this field by resuming the general presentation of the fraud detection, I started implementing a system inspired by some paper mentioned in the first report.

0.2 Introduction to the fraud detection

With the development of technologies and the large use of online payment or transaction of money, the security become a subject important today. Recently, the fraud becomes a problem witch causes a billion of dollars loss every year, So that the detection of the fraud becomes really important. Many scientist make their effort to study the major type of fraud and look for solutions. the frauds can be divided into these domains, credit card fraud, telecommunication frauds, assurance frauds and computer intrusion fraud. In this report we try to resume the most commune frauds, and some techniques of the detection.

0.3 The online auction fraud and detection techniques

The history of online auction fraud detection is a combating between the system builders and fraudsters. The non-delivery fraud or the mis-representation-of-items fraud have been diminished for long time since the peer-reputation-review system has been employed. However the new system has still some weak points that the fraudsters could take advantage of them, such as inflated-reputation fraud or shilling bids fraud. One solution is called anomaly data detection, witch compare the user's historical bid's average attributes to those of the data set, to identify the unfamiliar user behaviours. In this report, we present two techniques: clustering and neuron network. The clustering is a unsupervised method and it divide the user's data into multiple small groups. With the assumption that most normal user has similar behaviours, this technique could identify easily the suspicious user. Secondly, the neuron network could learning the form of the clustered data, so that it could recognise the unfamiliar data patterns. At last, The system could run the detection program each time it finished an auction, witch make the system detecte in teal time the frauds.

This report present how this system is implemented. Considering the complexity of this project, we present this work in 3 chapters: Generation of synthetic data, Classification of data, and the Real time detection. In the first place, we are introducing the generated synthetic data. we present here the attribute

of users, the seller and bidder agents the auction and bids. Second, there exists an amount of techniques of detecting the frauds, and we choose to present two machine learning approach among them. At last we have a discussion of the limits of these implementation.

Chapter 1

Generated synthetic data

In the real commercial world, the data is treated as the secrete of the company, especially for the grand international firm. That is the most common difficulty for the most researchers to find a well-defined benchmark. In the paper of Tsang S.et al. 2012[1] , the author introduced a syntactic method to generate the similar data. This work starts from here.

1.1 The System

The system is create mostly by Java with a complement library: JADE. Actually, the generation of data is based on the multi-agent system. The basic model is called Seller-Buyer model. With this model, we create a similar one, called Seller-Bidder model. The Seller propose auctions to the Bidder, and the Bidder generate some synthetic data for this auction. To implement this agent-based system, we use the JADE platform.

1.1.1 The syntactic data

There are two groups parameters in this system. The first part describe the function of the sellers and the auction. An auction is defined by a set of attributes: . We use the Gaussian distribution to generate auctions. On the second part, the bidder generate some bids by the predefined distribution. Here we introduce some parameters of this system. The original idea come from the paper of Ford et al. 2012 . [2] To define the data, actually, we take some characteristic label to describe the users, such as averageBidTime, averageBidAmount, bidsPerAuction , reputation.

1.2 Implementation

Firstly, we create 2 agents to communicate with each other. when they finished generating the auctions, the bidder agent Starts to calculate the average values At last , it creates a file "training_data.txt". Then it starts to cluster the data into small groups, and recreates a new file called "clustered_data.txt". At last, it trains a tensor flow model with the clustered and labelled data.

Chapter 2

Classification of data

The classification of data is studied in a long range of time. Recently, with the development of machine learning, some scientist tried these approach to deal with the anomaly detection. Actually, we present here two representative methods : The clustering method and the Neuron network method.

2.1 Approach: Clustering

Clustering is an unsupervised learning algorithm. It divides the the data into many smaller clusters depends on their similarity, so that each cluster contains the feature vectors witch represent the similar users. The similarity is defined as an angle between two vectors. Before the learning starts, we normalise the data by the maximum of their columns, so that the algorithm of clustering could have done better a job. The pseudo code :

The Similarity of two clusters Ca and Cb is defined as a product of vector of features x and y, where x (respectively y) is the average feature values of all elements a (respectively b) in the cluster Ca (respectively Cb).

2.1.1 How to use the code

To execute the programme, here is the commands:
python cluster.py <input file name> <output file name>

the program read a file as inputs, and then write the data into . at the end of file. At the end of each line, the clustering algorithm add a number to label the data. Since the data varies all the time, it is difficult to divide the data set into exactly two groups. So in this program, I simplified the idea: The majority users are normal and just a little of them are weird. Which means that the majority data (nearly 95%) is labelled as '0', and the rest of them is labelled as '1'. The accuracy of the learning is about 99%.

2.2 Approach: Neuron network

The real time classifier is based on the neuron network. The net has three layers, the input layer(size equals to the feature vector), the hidden layer(size not mentioned), and the output layer(2 neurons represent the normal and suspicious.) Active function: sigmoid function.

in this report, I implement a neural network detector with the tensorflow library. we used a training dataset labelled by the cluster. the learning model is here:

$$x * w + b = y$$

where x is the user's attribute(inputs), y is the labelled data(out put, suspicious or not). w and b are the wight and bias.

The neuron network could adjust the w and b so that this model recognise well if the user is a fraudster.

Algorithm 1 Cluster generation.

Input: A set of data points and a predefined minimal similarity
Output: a set of clusters that meet the minimal similarity requirement

```
1.  GenerateClusters (DataSet dPoints, ClusterSet
    clusters, double minSimilarity)
2.  if size (clusters) == 0 // initially, there are zero
    clusters
3.    for each element e in dPoints
4.      create a new cluster c for e and add c into
        clusters
5.    return GenerateClusters (dPoints, clusters,
      minSimilarity)
6.  else if  size (clusters) == 1 // there is only one
    cluster
7.    return clusters
8.  else // there are at least two clusters in set clusters
9.  initialize maxSimilarity to 0
10.   initialize mergeClusters to false
11.   for each pair of clusters c1 and c2 in clusters
12.     calculate the similarity between c1 and c2
13.   if similarity > maxSimilarity && similarity ≥
      minSimilarity
14.     maxSimilarity = similarity
15.     set mergeClusters to true
16.   if mergeClusters == true
17.     merge c1 and c2 into a new cluster c3
18.     replace c1 and c2 by c3 in clusters
19.   return GenerateClusters (dPoints, clusters,
      minSimilarity)
20.  else // no more clusters can be merged
21.     return clusters
```

Figure 2.1: the clustering algorithm mentioned in the article: A real time self-adaptive classifier for identifying suspicious bidders in online auction(2012, Ford et al)

$$\mathrm{SIM}(C_a, C_b) = \vec{x} \cdot \vec{y} = \sum_{i=1}^{n} x_i * y_i, \tag{5}$$

$$\vec{x} = \frac{1}{N_a} \sum_{i=1}^{Na} \vec{a}_i \quad \vec{y} = \frac{1}{N_b} \sum_{i=1}^{N_b} \vec{b}_i, \tag{6}$$

Figure 2.2: the similarity of the feature vector x and y of two users

2.2.1 How to use the code

To train the model: **python training_tensor.py <input labelled data file>**
after the training it create a temporary directory 'tmp', which contains the parameters of the graph. Thanks to the tensor flow library, it is easy to rebuilt the tensor flow graph from the directory. The algorithm print the accuracy of this model by testing the data in the tests set.

To test if the input data is suspicious: **python tensor_watcher.py <input data list...>**

The watcher could out put a value 0 to indicate that the input is normal, and 1 to indicate that the user is suspicious.

Chapter 3

Detection of frauds in real time

Here we present an example of the frauds: Shilling bids fraud. Sometimes a seller try to bid his object with some fake account, so that the final price is much higher than normal.

3.1 Detection with Neuron network

3.1.1 How to use the code

To train the model, **In Java, execute PrincipalAuctionTrain.java**
To start the real time detection, **In Java, execute PrincipalAuction.java**
After the training it create a temporary directory 'tmp', which contains the parameters of the graph. Thanks to the tensor flow library, it is easy to rebuilt the tensor flow graph from the directory.

Conclusion and future work

These implementation have not been perfect and there are many limits. Firstly, the attribute of generated data maybe not enough to present the other frauds. Secondly, the detection method is not static, so it rebuilt the tensor flow graph each time when it execute. As a consequence, the execution time may be too long.

Bibliography

[1] Sidney TsangGillian DobbieYun Sing Koh. Generating Realistic Online Auction Data. *AI 2012: Advances in Artificial Intelligence. AI 2012*, 7691:120–131, 2012.

[2] Haiping Xu Benjamin J. Ford and Iren Valova. A Real-Time Self-Adaptive Classifier for Identifying Suspicious Bidders in Online Auction. *Oxford University Press on behalf of The British Computer Society.*, 2012.