San Francisco Crime Classification

Sirui Tan, Haoyue Bai, Guihao Liang
Electrical Engineering
Columbia University
st2957@columbia.edu hb2479@columbia.edu gl2520@columbia.edu

Abstract— **The project gains insights into the intrinsic relationships between temporal as well as geographical information and the category of crimes through using various analytics approaches; main-stream open source data analytics tools are applied in practice in order to discover the relative pros and cons of each tool.**

The value of this projects lies in three aspects: Firstly it demonstrates the solution to yet another facet of big data: the complexity of data, rather than the sheer amount; Secondly, it showed the prematurity of Spark as a general-proposed machine learning library by comparative machine learning experiments; Thirdly, it managed to extract empirical information from visualization as well as insights from machine learning, both of which would be helpful for predicting the crime occurrence of San Francisco in the future.

Crime Classification; visualization; decession tree; gradient boosting decession tree; Spark; sci-kti learn

I. Introduction

San Francisco Crime Classification, which is a problem from Kaggle competition, aims at proposing a feasible classification on the 39 different types of crimes and make prediction on the category of a certain crime reported in the reporting system.

The city by the bay has a high crime rates, besides the distance is only several miles between this area to Oakland which is the top two most dangerous city in the United States. To analysis the crime incident dataset and acquire useful information about crime distribution and the change of different crime rate is very meaningful in predicting and preventing the occurrence of certain crime in San Francisco in the future.

The aim of our team is to visualize the dataset, to use the information from it to extract features and to train the e classifier so that we could predict the categories of crime that occurred during these 12 years.

This is an intractable problem since not only the dataset is highly heterogeneous and sparsely distributed, but also it requires strong machine learning background and well-trained mathematic analysis ability, which provides a good opportunity to improve ourselves.

II. System Overview

Overview of Dataset

The dataset contains incidents derived from SFPD Crime Incident Reporting system [1]. The data ranges from 1/1/2003 to 5/13/2015, with more than 800 thousand records and 39 categories of crime.

The data field of the dataset comprise of dates, category, day of week, PD district, address, longitude and latitude. Dates denotes the timestamp of the crime incident. PD district is the name of the Police Department District which administer the street the crime happened. The approximate street address of the crime incident is shown in Address field. X and Y denotes the longitude and latitude of the position the crime happened.

III. Algorithm

Overview of Methodology

Visualization is applied to filter those meaningful data at the start of the project. Then various strategies, such as counting-based log odds ratio and principal components analysis (PCA), are applied to pre-process the data. Afterwards, the pre-processed dataset is fed into classification model based on the python Scikit-learn and spark machine learning lib.

Visualization of Dataset

Visualization of dataset contains three main processes: erase the data with wrong latitude and longitude (very few of them); utilize Seaborn to plot topographic map and histogram of distribution of all category of crime.

In the first place, although there is only very few of wrong data, data with wrong latitude and longitude information is discarded and therefore dataset is concentrated into the core area of San Francisco.

The Seaborn, which is a Python visualization library based on matplotlib, provides a high-level interface for drawing attractive statistical graphics [2], and is used to draw the picture to visualize the distribution of 39 different types of crimes.

In order to have a panorama of all categories of crime in the dataset, four kinds of graphs for each category are visualized.

First is topographic maps for the whole 12 years so that we could roughly know the spatial distribution for certain crime. Since there are 39 categories of crime, 39 topographic maps would be received by visualization. For the sake of having a better analysis of the change of spatial distribution, topographic maps for each year are also plotted. There are 468 (12 x 39) graphs. Since the number of crime may fluctuate in different months and hours, mouth distribution and hour distribution of each category are plotted in a form of histogram.

At this point of view, it is concluded intuitively that the data is highly sparse in time and has coloration with location or exact time stamp. So both location and time will be dominant factors to aid our classification to decide the category of the crime.

Feature Extraction Strategy

The data relative to the date is divided into smaller parts, for instance, date time in format of dd/mm/yyyy is separated into dd, mm and yyyy, in order to increase the degree or dimension of the data pool, since high dimensional matrix can be generated to feed training model. In this way, a higher machine learning accuracy can be obtained.

Counting-based model with log odds ratio is used to process the address information. Given the fact that there are thousands of addresses, it requires a huge capacity to learn the amount of parameters.

Count-based tools provided by Azure [4] is used to map text-based data, like address, into values, i.e., the time one entry appears. Thus, compact set of features for the dataset under the summary of data features generated by the previous counting process is generated. At this stage, fewer features that are correlated mutually are left, making up for faster training and learning. Then tools provided by Azure is applied to compute log odds ratio from the counting based data, and then further used to transform the count-based data into log odds ratio format. The log odds ratio is a method to show how strongly the property of A is associated with the property or the absence of B. That is, we want to correlate these addresses. If the crime category Fraud in address A is highly associated with B, it can be predicted that crime happens in B is probably fraud rather than other types of crime.

Dimensional reduce is applied on the pre-processed data. And Principal component analysis (PCA) is implemented to linear transform dataset. After preprocessing is well performed and finished, the data is reconstructed into to lower dimension and the degree of our data dimension is shrunk by discarding those irrelevant data and leaving principal components highly correlated with categories.

Classification

Two methods are employed concurrently: one is to implement the machine learning library of the Spark, the other is to employ python ski-learn to do data analysis.

Spark

For Spark part, decision tree strategy is applied to classify pre-processed data. Linear SVM, which is simple to implement on with Spark, could not support classifications for number of types beyond 2 in Spark. We utilize the decision trees in Spark. Decision tree is widely used since the method is easy to interpret, handle categorical features, extend to the multiclass classification setting, do not require feature scaling, and are able to capture non-linearity and feature interactions [5]. More specifically, the random forest trees and Gradient-Boosted trees algorithm will be applied to our data model.

Data files are loaded into spark in order of years, that is, training model on each year instead of stewing all the data together. And here, it is assumed that the categories of crime are irrelevant to which year's data chosen to be classified, despite the fact that some new types of crime like digital fraudulence will take place more frequently in recent years than in years like 2003, 2004. Furthermore, we set every row of our file in format as this way: first item of as the label and all the data follows as float type, in which way, we can convert the file into Spark RDD and handle data input more conveniently.

As the random forest trees is implemented on the pre-processed dataset, the node impurity, which is a measure of the homogeneity of labels as to choose between candidate splits, is set as Gini impurity, which is often used for classification. Then randomSplit method can be invoked as to randomly split the dataset into 80% as training input and 20% as test input (for result validation and evaluation use).

The RandomForest and GradientBoostedTrees class in the pyspark.mllib.tree module are adopted to do classification. Here the train data generated above is fed into the classifier, and number of classes to discriminate is set as 6. As for the depth of the tree, the max depth is set to be 30, which is the deepest depth limited by the Spark. The max number of binary bits representing the float type data is set as 32. Thereafter, model.predict is used to evaluate the test instances and compute the test error.

Scikit-learn

scikit-learn is one of the most prevalent python-based machine learning libraries. It provides simple and efficient tools for data mining and data analysis.

Here we used sklearn for cross validation, grid search and comparison of different classification methods, including

Support Vector Machine (SVM), Random Forest and Gradient Boosting Tree.

The application of sklearn startes from applying pandas and numpy to extract relevant features, including those from parsing timestamp, latitude, longitude and address features extracted using log odd ration calculation of count factorization.

After features are drawn, PCA functionalities of sklearn are used to reduce feature dimensionality. Each classification model has multiple parameters, the choice of which greatly affect the quality of classifier. So grid search for optimal parameter combinations was applied before the final model is decided.

A 5-fold cross validation scheme is applied here for each classification model, from which the model was trained and classification results calculated

IV. Experiment Results

Visualization Results

As is stated in IV.Algorithm, there are four kinds of graphs for each category are visualized: topographic maps for whole 12 year; topographic maps for every year; month distribution; hour distribution.

Topographic Maps for 12 Year

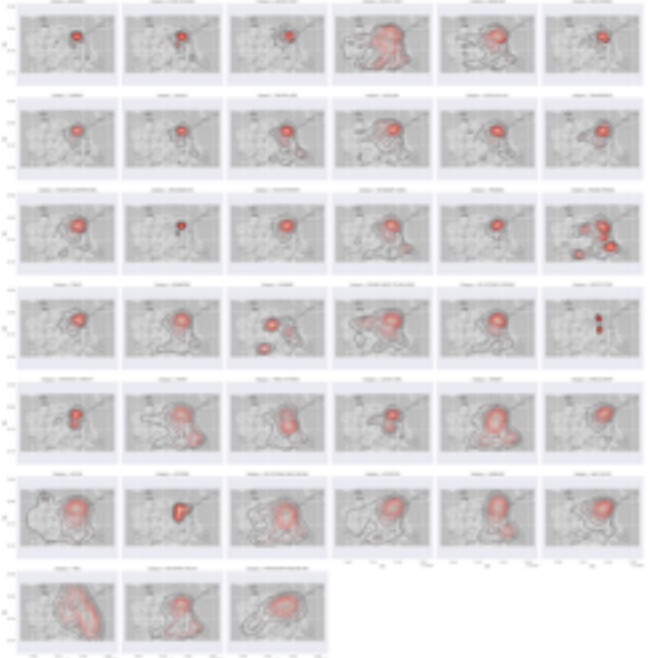

Figure 4.1 Topographic Maps for 39 Categories of Crime

Figure 4.1 illustrates 39 topographic maps for each kinds of crime for the whole 12 years. From these graphs, we could discover that some crime has very concentrated distribution in certain area, while others is widespread. Different category has different spatial distribution characteristics.

Topographic Maps for Every Year

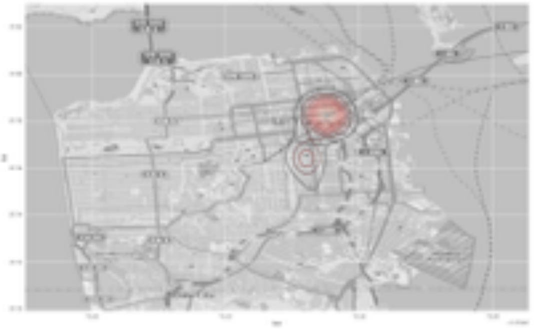

Figure 4.2(a) Topographic Maps for Drug/Narcotic in 2003

Figure 4.2(b) Topographic Maps for Drug/Narcotic in 2004

Figure 4.2(c) Topographic Maps for Drug/Narcotic in 2005

Figure 4.2(d) Topographic Maps for Drug/Narcotic in 2006

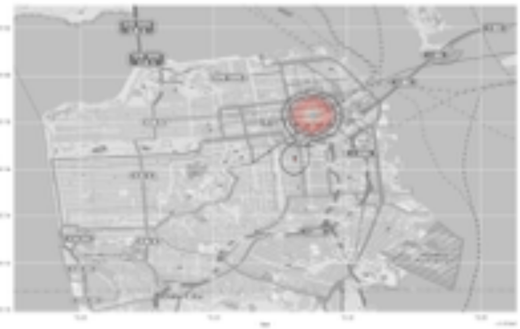

Figure 4.2(e) Topographic Maps for Drug/Narcotic in 2007

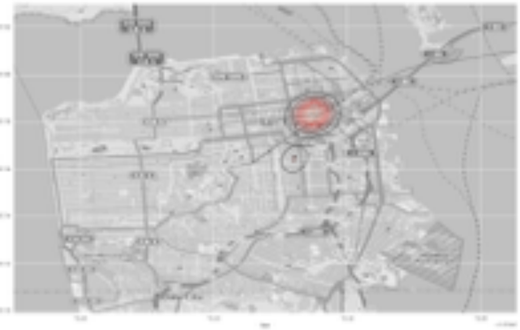

Figure 4.2(f) Topographic Maps for Drug/Narcotic in 2008

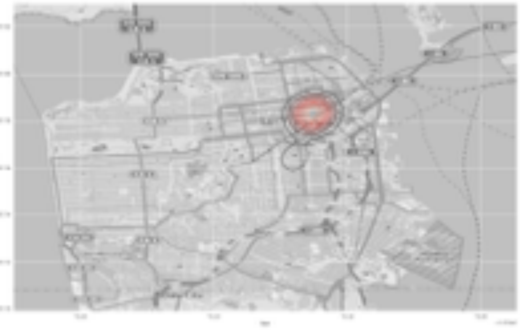

Figure 4.2(g) Topographic Maps for Drug/Narcotic in 2009

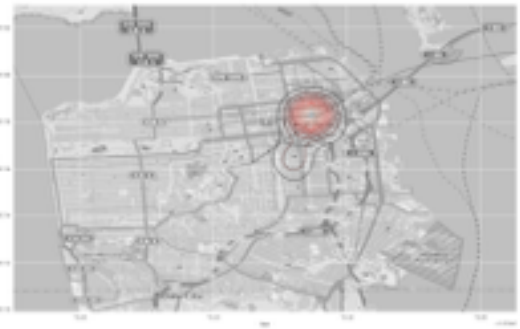

Figure 4.2(h) Topographic Maps for Drug/Narcotic in 2010

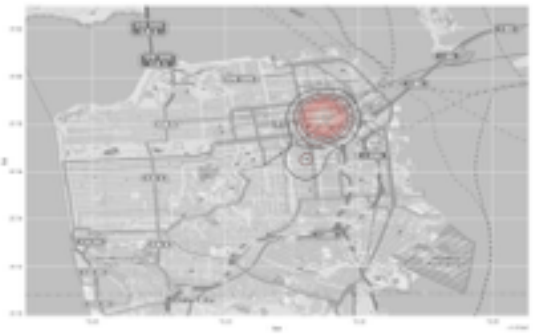

Figure 4.2(i) Topographic Maps for Drug/Narcotic in 2011

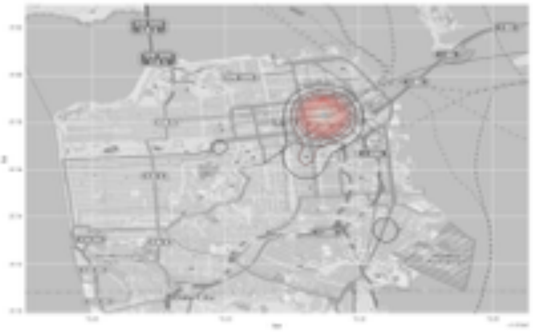

Figure 4.2(j) Topographic Maps for Drug/Narcotic in 2012

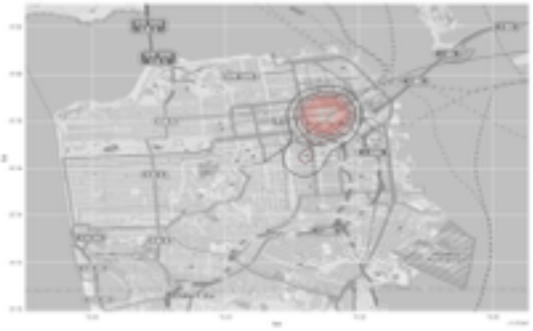

Figure 4.2(k) Topographic Maps for Drug/Narcotic in 2013

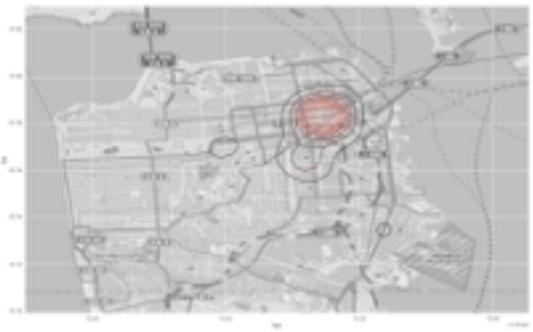

Figure 4.2(l) Topographic Maps for Drug/Narcotic in 2014

Figure 4.2(a)-(l) shows topographic distribution of Drug/Narcotic from 2003 to 2014. As time goes by, although there is some slight change in the contour line, the center location of the crime basically does not change, which could be viewed as a distinct characteristic of Drug/Narcotic crime. Similarly, we could view the change of spatial distribution of other crimes.

Month Distribution Histogram for Every Year

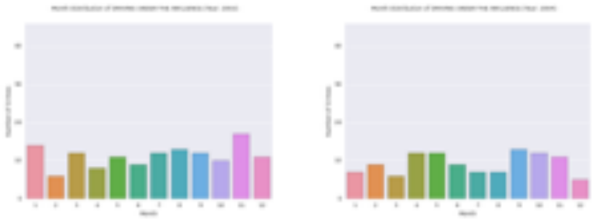

Figure 4.3(a) Month Distribution for Driving under the Influence in 2003 and 2004

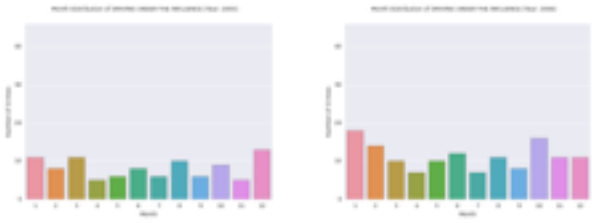

Figure 4.3(b) Month Distribution for Driving under the Influence in 2005 and 2006

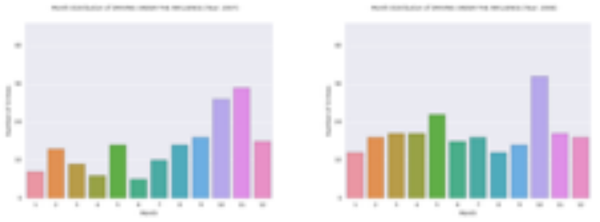

Figure 4.3(c) Month Distribution for Driving under the Influence in 2007 and 2008

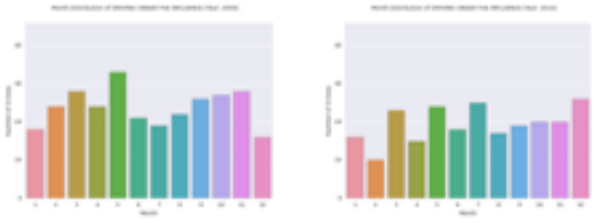

Figure 4.3(d) Month Distribution for Driving under the Influence in 2009 and 2010

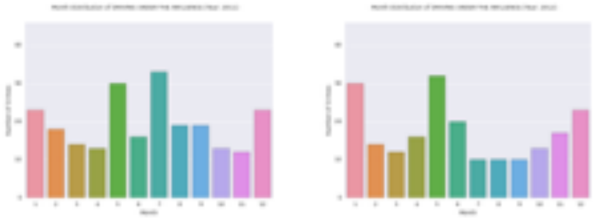

Figure 4.3(e) Month Distribution for Driving under the Influence in 2011 and 2012

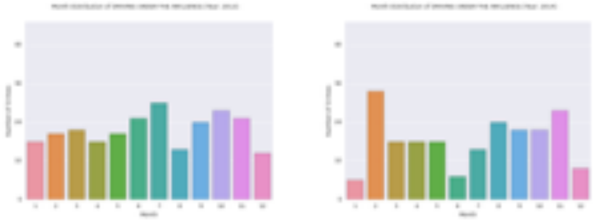

Figure 4.3(f) Month Distribution for Driving under the Influence in 2013 and 2014

Figure 4.3(a)-(f) illustrates month distribution for driving under the influence. From these graphs we know that month may not have a high correlation with this crime category. However, it also illustrates that the occurrence of this crime has increased from 2003 to 2009.

Hour Distribution Histogram for Every Year

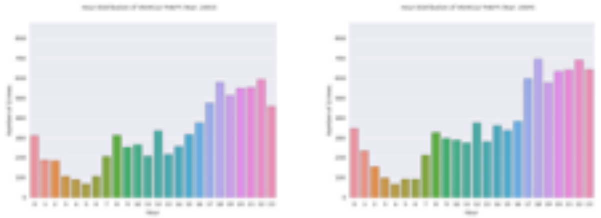

Figure 4.4(a) Hour Distribution for Vehicle Theft in 2003 and 2004

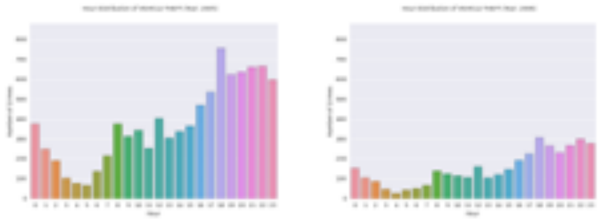

Figure 4.4(b) Hour Distribution for Vehicle Theft in 2005 and 2006

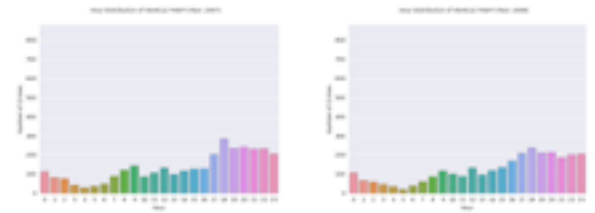

Figure 4.4(c) Hour Distribution for Vehicle Theft in 2007 and 2008

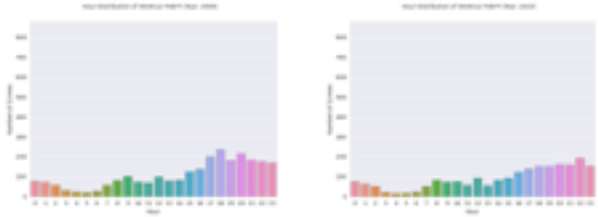

Figure 4.4(d) Hour Distribution for Vehicle Theft in 2009 and 2010

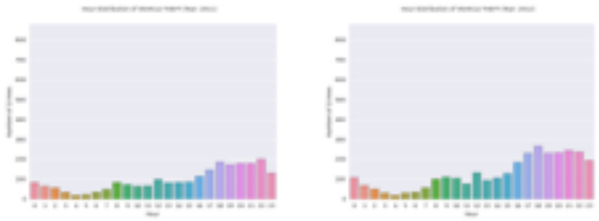

Figure 4.4(e) Hour Distribution for Vehicle Theft in 2011 and 2012

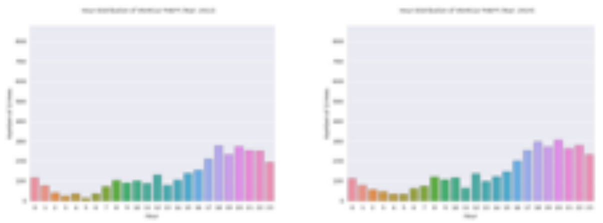

Figure 4.4(f) Hour Distribution for Vehicle Theft in 2013 and 2014

Similarly, Figure 4.4(a)-(f) illustrates hour distribution of Vehicle Theft from 2003 to 2014. The number of this crime has been decreasing since 2003, especially it dropped rapidly from 2005 to 2006. In addition, vehicle theft often happens in the afternoon and evening, but rarely happens at 4 or 5am.

Classification Results

Table 4.1 Classification Accuracy.

Table 4.1 demonstrates the result of the classification on the corresponding dataset of years from 2003 to 2014 using scikit-learn, following steps mentioned above.

Comparison of Classification Results between Spark and Scikit-learn

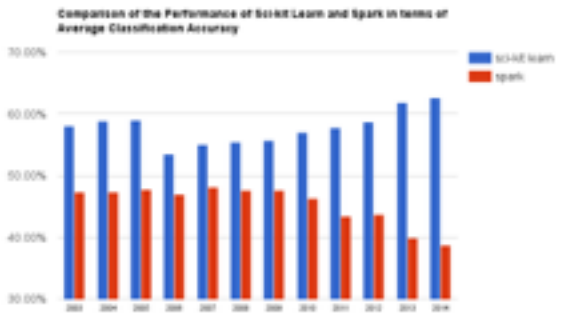

Figure 4.5 Comparison between learning accuracy scikit-learn and Spark.

The reason behind the scene is that the spark is not dedicatedly designed to do machine learning but large scale data processing. For instance, the maximum tree depth is limited in Spark while ski-learn can provide deeper depth for training which will lead to a better classification result.

V. Conclusion

Empirical Information Benefits for Crime Reduction

From visualization of crime distribution, we could conclude many empirical information about the distribution as well as high incidence of certain crime categories, which could be beneficial for police to focus on specific blocks and time period.

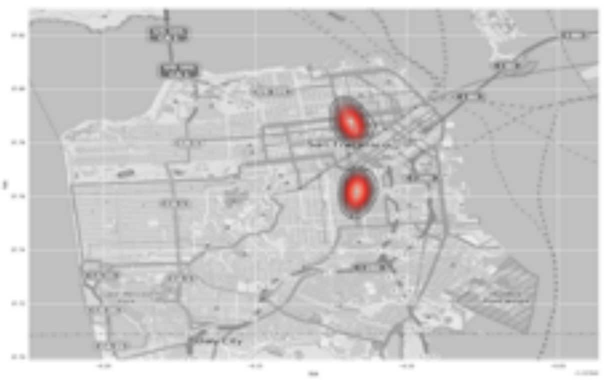

Fig 5.1 Topographic Map of Prostitute over 12 years

Year	2003	2004	2005	2006	2007	2008
Accuracy %	58.5	56.8	54.8	59.4	60.5	60.6
Year	2009	2010	2011	2012	2013	2014
Accuracy %	60	60.6	63	63.4	66.5	67.4

For instance, from the graph above, two red parts in the fig 5.1 illustrates the fact that these two blocks are with the highest possibility of prostitute.

Decision Tree Outperformances SVM

The decision tree can handle data that mutually correlated with impurity discrimination method, while support vector machine cannot differentiate unless data can be completely split out.

Insight from PCA

The PCA procedure can provide information about the principal components which facilitates the further classification with machine learning strategy. At start, with 39 types of crime to handle with, PCA is applied to find the principal components of the data, and the result is rendered below. Intuitively, address is surmised as the key feature from the rudimentary analysis of the dataset from the visualization. Eventually, the PCA gives credit to this hypothesis.

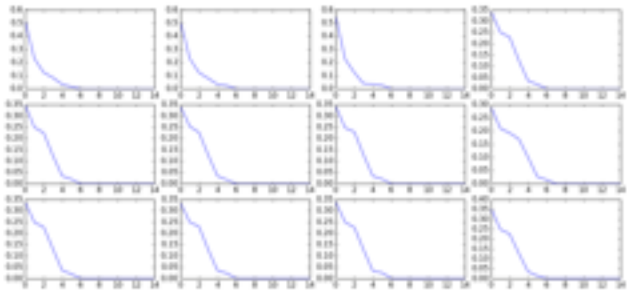

Figure 5.2 PCA procession results

Contributions of Each Team Member

In this project, Sirui Tan is responsible for data preprocessing, feature extraction and machine learning; Haoyue Bai is responsible for data visualization, machine learning and presentation preparation. Guihao Liang is responsible for video demonstration and report writing.

Acknowledgment

We would like to thank professor Lin for his great teaching and granting us the precious opportunity to learn and practice big data analytics skills. We also want to thank all TAs, especially Junkai Yan who provided us with a lot of valuable advices and encouraged us when we faced obstacles.

References

1. https://www.kaggle.com/c/sf-crime/data/
2. http://stanford.edu/~mwaskom/software/seaborn/
3. https://en.wikipedia.org/wiki/Odds_ratio
4. https://msdn.microsoft.com/en-us/library/azure/dn913056.aspx
5. http://spark.apache.org/docs/latest/mllib-decision-tree.html