Predicting Safety Risks from Street View Photos (April 2017)

Qi Chen, Kimberly Beswick, Salman Alsalman, Abdullah Al-Mahmoud, Sanchit Arora
Computer Science Department
San Diego State University

Abstract— Predicting safety risks from street view photos is a systematic approach for both identifying and analyzing patterns and trends in crime. In recent years, crime data from different heterogeneous sources have given immense opportunities to the research community to effectively study crime pattern and prediction tasks in actual real data. In this paper, we will discuss research that takes into account a variety of crime related variables to predict regions which have high probability for crime occurrence in San Diego, San Francisco, Seattle, Chicago and New York. We will then predict the average crime rate for a given street view photo.

Index Terms— Crime distribution, clustering, data mining, algorithms, Street view etc.

I. Introduction

For many years, research in the area of crime analysis has been used towards the mitigation of crime and public safety. With the advent of data mining techniques, public data availability, there has been an exponential increase in the creation of data analytics and visualization tools for predicting crime and handling the public-safety systems. Using powerful learning models geared for image analysis, one can create predictive models that may be used to identify areas with higher crime risks in order to increase public safety, guide housing development, aid in insurance assessment, and help law enforcement.

Task description

The present research work proposes the use of an amalgamation of data mining techniques that are linked with a common aim of predicting and visualizing the crime rate. For this purpose, the following research objectives were formulated.

1. To develop a data cleaning algorithm that cleans the crime dataset, by removing unwanted data, or denoising it.
2. To explore and enhance clustering algorithms to identify crime patterns from data.
3. To explore and enhance classification algorithms to predict future crime behavior of a location given with Street View photos.
4. Collect data from different cities and study the learning model classification effectiveness for different cities.

Challenges

As a continuation of the research done for San Diego crime data in Fall of 2016, this project aims to expand the breadth of research by increasing the number of cities from one to five. The cities chosen for the study are: Chicago, San Francisco, Seattle and New York. Large cities with high population density like New York and chicago pose new challenges to the study, due to the high amount of data. We mitigated these issues by filtering the data points. If the dataset was too large, we used data mining methods to screen representation points. Our study has four major steps:

- Classify points of interest by different crime levels.
- Obtain the street view photos through Google Streetview API.
- An enhanced clustering algorithm to detect hot spot zones in the dataset.

- An enhanced classification model is used to predict the crime rate for a given street view photo.

II. Experiment

Dataset Description*:* The dataset was collected from these official website.

San Diego (SD):

http://www.sandag.org/index.asp?classid=14&subclassid=21&projectid=446&fuseaction=projects.detail

New York (NY):

https://data.cityofnewyork.us/browse?tags=crime

San Fransisco (SF):

https://data.sfgov.org/Public-Safety/Police-Department-Incidents-Previous-Year-2016-/ritf-b9ki

Chicago (CHI):

https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2/data

Seattle (SEA):

https://data.seattle.gov/Public-Safety/Seattle-Police-Department-Police-Report-Incident/7ais-f98f/data

1. Data Labeling

Preprocessing: DeNoise

The coordinate points beyond San Diego city are considered as noise. In order to remove them we set Northeast and Southwest point of the San Diego City to be the maximum and minimum boundary for the screen data.

northeast= [33.3,-116.9]; % max latitude and longitude
southwest= [32.5,-117.4]; % min latitude and longitude

Same action for other cities, this step can remove noise data and redundant data (happen several crime events in one place).

northeast= max(GIS); % max latitude and longitude
southwest= min(GIS); % min latitude and longitude

2. Crime Density Evaluation

K-Nearest Neighbor

There are 3 steps to get crime density for each point:

1. Select a data to be center point. Set radius=R mile (we use 1 mile in this study).
2. Select an R*4 window, calculate distance from each point in this window to center point.
3. Count crime quantity in R mile radius, use this number to present crime density for this point.

After study, we find R value will influence crime density distribution. Big R will make distribution smoother, but distortion. After study, 1 mile is good tradeoff. We then do normalization and get crime level:

Normalization(0~1): crimerisk=(density-min)/range

*crime level: crimelevel=ceil(crimerisk*level)*

Using this method, we get crime level with any address in San Diego area.

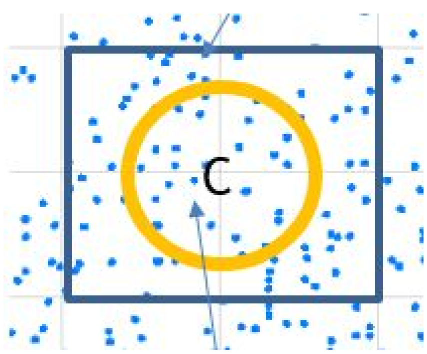


Fig 1: Depicts the Data selection window with one mile radius.

K-Means (Select represent points)

Due to the large amount of data in each city, especially in populous cities such as New York and Chicago, the size of the datasets needed to be constrained. If we download pictures for each point, it would create an unwieldy amount of data, and would require excessive computation capacity in the data classification step. For purposes of simplification, we have run K-means to select represent points for each city. This was done to save computational time and resources.

In this study, we set K=5000. It means we separate a city into 5000 communities and select one point for each community. We think they have the same feature. This filters the dataset into a more manageable size.

Example result:

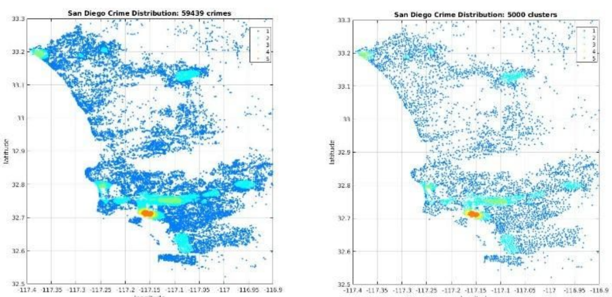


Fig 2: Crime distribution in SD

We extracted 5000 points from 59439 crimes in SD, they keep the same distribution.
It's voronoi picture:

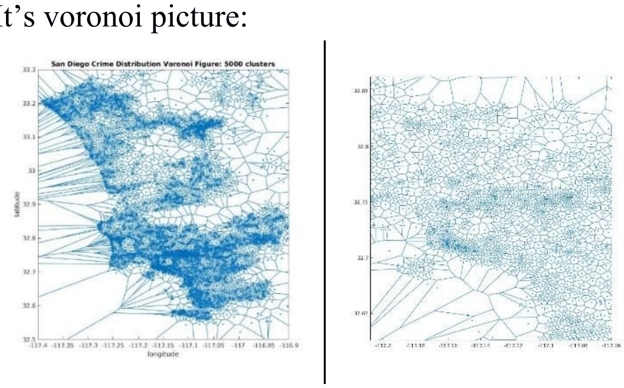


Fig 3: Voronoi picture in SD

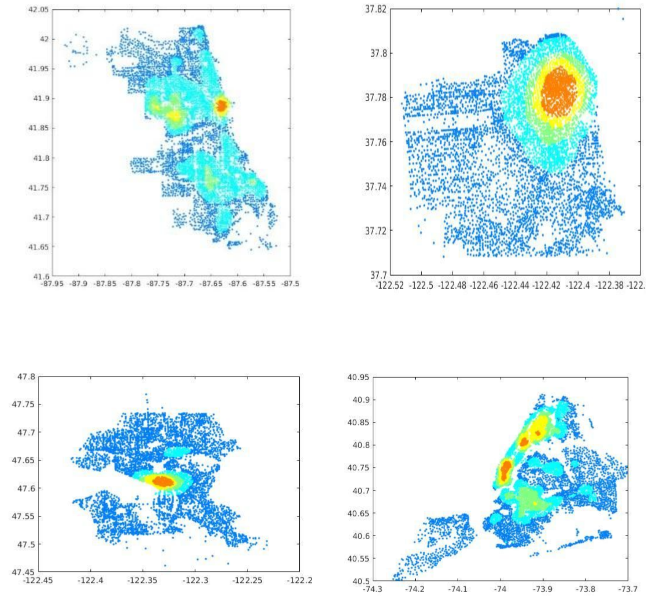


Fig 4: CHI/SF/SEA/NY Crime Distribution

Level	SD	CHI	SF	SEA	NY
5	76	69	487	183	159
4	30	129	328	240	436
3	178	503	455	174	766
2	730	2258	912	407	1288
1	3985	2040	2817	3995	2350

Table 1: Crime level statistic (5000 points, 5 means the highest risk)

According to these results, the crime risk order (high to low) is NY SF CHI SEA SD. SD is the safest cities in these 5 cities.

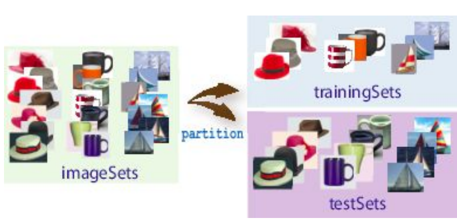
3. Data Classification

Image category classifier using Bag of features is one of the model to classify a sample image into different category using the bag of features for each image. The model constructs a “vocabulary” of SURF features representative of each category by extracting the SURF features from all the images of each category. It reduces the number of features by quantization of feature space using K means clustering and constructs the bag of vocabulary. A Linear Multiclass SVM classifier is used for classification.

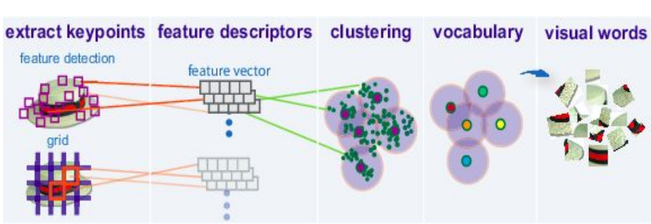
We extract 8 images for 8 orientations (N,S,E,W,NE,SE,NW,SW) for each hot spot of each crime level and separate them as two training set and testing set for testing purpose. We then test the model for different size of testing and training set and for different values of C (Box Constraint) We thus compare the accuracy of the model for different crime levels in order to find a model with better performance and better accuracy and precision rate.

We then train the model with different dataset and try to predict the crime rate for a random location within San Diego given, the street view photo of that location.

Step 1: Set up image category



Step 2: Create bag of features



Step 3: Train an image classifier with bag of visual words

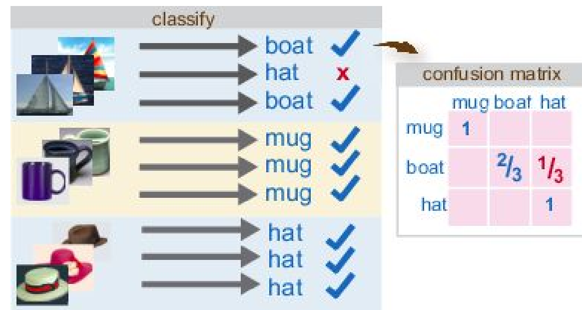


Fig 5 : Steps to construct and train a classifier model to predict the category of a random street view image. [1]

III. MAJOR RESULTS & ANALYSIS

There are several key parameter in this study.

- Percentage: proportion of training data
- C: Box Constraint, bigger C has more tolerance over noises.
- Orientation quantity: pick up how many picture from 1 point. 1~72.
- Method that select training data: random or sequential.

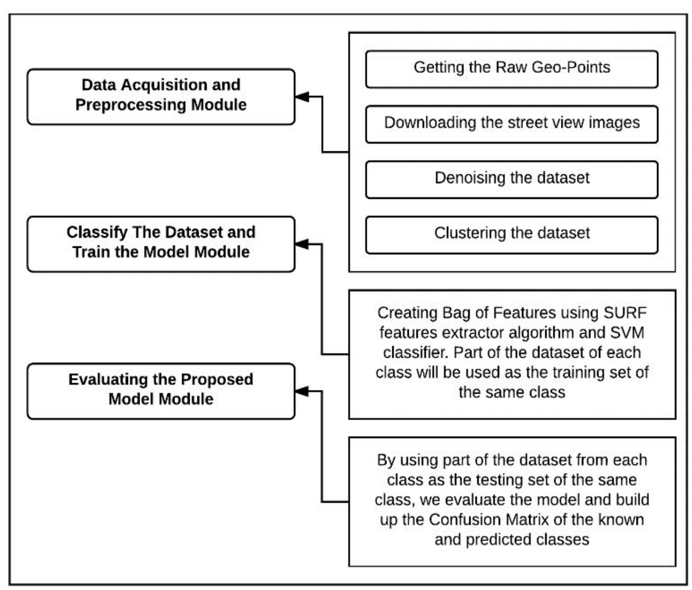


Fig 6: Data Analysis Workflow

The following section are experiences and summary by each team member.

San Francisco (Kim)

One of the most pressing decisions for the dataset analysis was image resolution. Due to the high computational cost of running the HOG classification algorithm, this effectively placed a bottleneck on time and resources. For maximum flexibility, images of size 640 x 640 pixels were downloaded from Google Streetview so that they could be downscaled later if need be.

Challenges: The first test was attempted on folders with over 2,000 images each. This data size, with current technology, became unwieldy. Therefore, the number of images per category was scaled to 400, then 300 images per category. The images per category was eventually chosen to be 161 per folder due to the high computational cost. At a resolution of 640 by 640 pixels, approximately 40KB each and approximately 6.44MB per category.

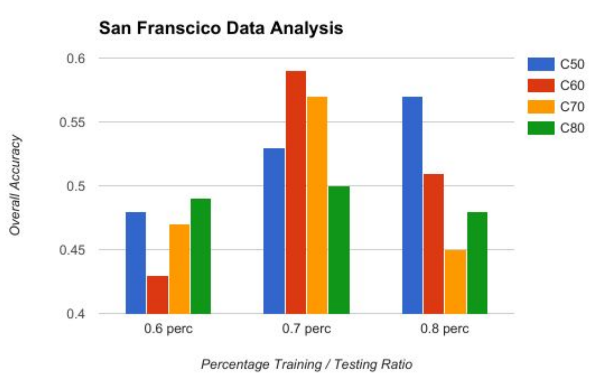


Fig 7

Images was selected based on their Streetview orientation. Only forward-facing pictures were used in the classifier to have a more homogeneous training data. These images were sorted by selecting only the third and fourth image out of the eight orientations. Because the Google car turns occasionally while collecting images, these images were not always a perfectly straight-on view of the road. Before running the HOG classifier, the folders were visually scanned to eliminate extraneous or irrelevant images.

The training to testing ratio was attempted at three different values: 60% training to 40% testing, 70% training to 30% testing, and 80% training to 20% testing. In addition, four different C values for the box constraint parameter were passed to the templateSVM function. As shown in the histogram, training to testing data at a 70% to 30% split is over the most effect for all C values. The highest overall accuracy was at an 80% training data, and there is a clear trend toward more training data leading to a higher accuracy. This suggests that with higher computational power and larger training sets, the HOG classifier could attaining a higher accuracy.

The current best results are: 59% accuracy for 640x640 px images, 161 images per category, C value of 60, and 70% training to 30% testing datasets.

New York (Salman)

New York original dataset consists of 478,805 data points. Which means we need to process 3,830,440 images of size 640x640. Considering the high computational cost and the limited resources we have, this number has reduced by clustering the crimes on the map. After normalizing the dataset using the K-Means, the total number of points of the dataset was 5,000 data points. These points were distributed as the following for each crime level: 2350 points for crime level 1, 1288 points for crime level 2, 766 points for crime level 3, 436 points for crime level 4 and 159 for crime level 5.

Challenges: There were many challenges during collecting the datasets and processing them. First, Google Street View is limited by 25K requests every day. NY dataset needed 40K requests; however, splitting the points into two patches is not efficient. We needed to make sure that we download the whole images of each crime level during the same run. Therefore, instead of spending two days to complete the dataset, we needed four days. Secondly, the downloaded images were not all correct and useful for processing. Some of the images were taken indoor and some were blocked with no data on them. These images were removed manually to make sure the final dataset are clean and useful. This consumed enormous amount of time. Finally, processing and running the HOG classification algorithms for the dataset on a regular computers failed. So, to make it to run, the number of images on each class were reduced to 100 images.

Many tests were attempts; however, 12 standard tests were done among each city on this project. The tests split based on two main parameters: the ratio, R, of the training dataset over the testing dataset and the value of the box constraint of the support vector machine, C. R was attempted at three different percentages: 60% to 40%, 70% to 30%, and 80% to 20%. In addition, four different C values were passed to the SVM function: 50, 60, 70 and 80.

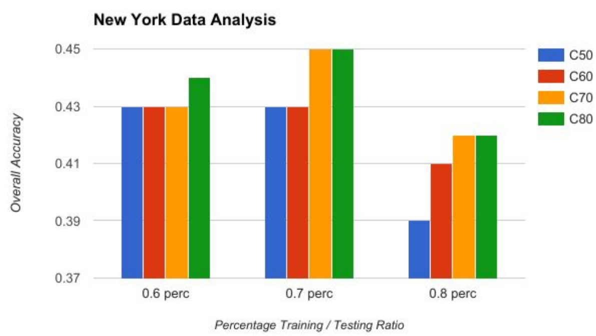


Fig 8

As a result, the larger the C, the better the average accuracy. Also, using the dataset with R equal to 70% gave the best result with average accuracy of 45%.

Seattle (Abdullah)

Seattle dataset has 5,000 data points with 3973 points for crime level 1, 414 for crime level 2, 202 for crime level 3, 238 for crime level 4 and 172 for crime level 5.

Challenges: The first challenge was with downloading streetview images. Google Street View API limits number of streetview images to be downloaded every day to 25.000. Since 8 images are needed for each point, the total number of images is 40.000. So, a decision was made to download each crime level images in a separate cycle. The second challenge is that number of points for crime level 1 is 3973 points, which means number of images would be 31784. So, images for crime level one has been separated into two cycles.

Since it is time consuming to perform the tests on all images, number of sample images chosen to perform the tests is 100. The training to testing ratio was attempted at three different values: 60% training to 40% testing, 70% training to 30% testing, and 80% training to 20% testing. In addition, four different C values for the box constraint parameter were passed to the templateSVM function. As shown in the histogram, training to testing data at a 80% to 20% split is over the most effect for C values 50 and 80. At a 70% to 20% split, the most effect for C value is 60. Finally, at 60% to 40% split, the most effect for C value is 70. Overall, the best results were at 80% to 20% split with C value equals to 80.

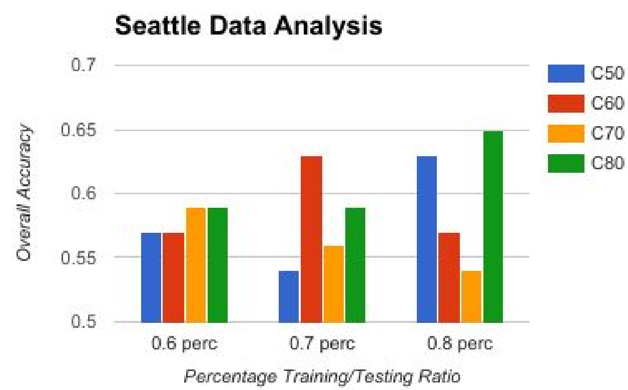


Fig 9

Chicago (Sanchit)

Challenge: the first challenge faced was that it was very time consuming. Since 8 images are needed for each cluster, the total number of images is 40.000. Each image is at a resolution of 640 by 640 pixels, approximately 50KB each and approximately 40MB per category which lead to huge data for the HOG Classify. So, the decision was made to reduce the image resolution to 200 by 200 pixels which made the program run faster as compared. Another challenge was that some of the downloaded images was not connected to any of the crime location. The decision was taken to delete the irrelevant images manually and run the program.

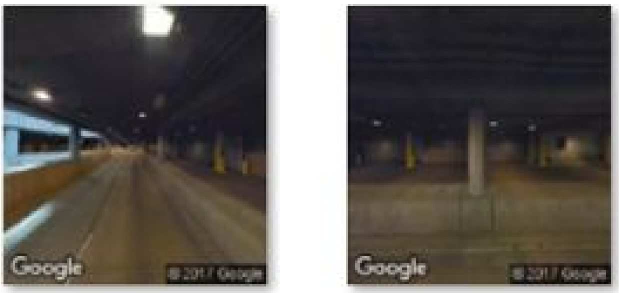


Fig 10: Relevant Images

Fig 11: Irrelevant Images

The 672 images were taken per each category. The training to testing ratio was attempted at three different values: 60% training to 40% testing, 70% training to 30% testing, and 80% training to 20% testing. In addition, four different C values for the box constraint parameter were passed to the templateSVM function. As shown in the histogram, training to testing data at an 80% to 20% split is over the most effect for C values 50 and 80. At a 70% to 20% split, the most effect for C value is 60. Finally, at 60% to 40% split, the most effect for C value is 70. Overall, the best results were at 70% to 20% split with C value equal to 50.

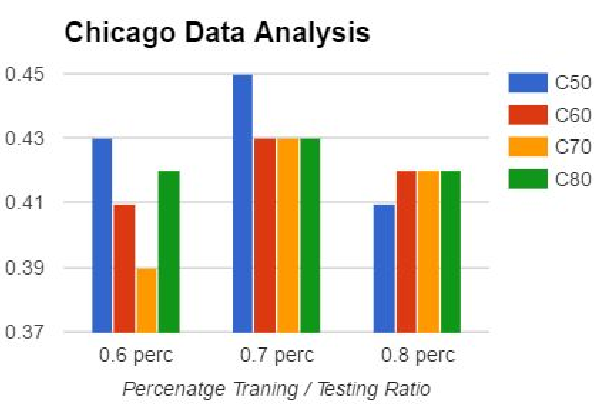


Fig 12

San Diego (Qi)

First, set 8 pictures in one point, select training data randomly, do study about percentage and C. it is result:

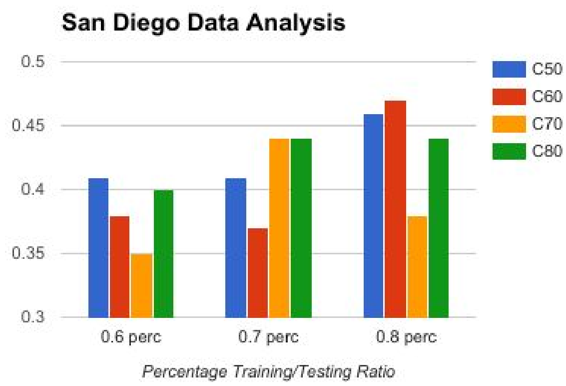


Fig 13

For the best result, the percentage is about 0.8, and C value is about 80.

Secondly, study orientation quantity in one point and training set selection method (randomize or sequential). Percentage is 0.6, C=50. It is result. We can find using different selection method has big influence to result.

Orientation	1	4	8	16	36	72
Acc(random)	0.3	0.33	0.42	0.52	0.67	0.74
Acc(sequent)		0.25	0.36	0.26	0.28	

Table 2: The results using different orientations

Finally, scene recognition and classification is different with object recognition and classification. The most of pre-trained model focus on object recognition and segmentation. We can't just use them simply. Scene learning should be a new challenge at computer vision area.

IV. FURTHER STUDY

Conclusion

This research work focus on predicting crime patterns through street view photos. Due to diversity of scene, the highest accuracy is about 0.59 at these cities. It means the current feature selection method and SVM cannot do a reliably good prediction by photos.

Professor Liu prefers to study and experiment the multi-instance learning approach. Salman found A Multiple Instance Learning Library named MILL. The library developed by Jun Yang, School of Computer Science, Carnegie Mellon University. Here is the link: http://www.cs.cmu.edu/~juny/MILL/

Furthermore, we want to study other model/method. Last but not least, we hope that our pipeline can help law enforcements and keep our community safe for everyone.

Acknowledgement

This paper describes research done at SDSU in the department of Computer Science. We are thankful to Dr. Xiaobai Liu for his guidance and cooperation.

References:

[1]https://www.mathworks.com/help/vision/ug/image-classification-with-bag-of-visual-words.html

[2]Kobayashi, Takumi. "BFO Meets HOG: Feature Extraction Based on Histograms of Oriented P.d.f. Gradients for Image Classification." 2013 IEEE Conference on Computer Vision and Pattern Recognition (2013): n. pag. Web. 1 May 2017.