San Francisco Crime Classification

Charles Hale, Feng Liu, {cphalepk, liufeng}@stanford.edu

Problem

Many crimes happen in San Francisco every day. It may be helpful for both residents and policy makers to understand which category of crimes are more likely to happen in a certain location at certain time.

In our project, we built up two models:

1. Naive Bayesian Model;
2. Mixture of Gaussians Model;

Our goal is to predict the probability that a crime belongs to certain category based on its time and location. In particular, we are seeking to minimize the multivariate log-loss:

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij})$$

Data

The data sets are downloaded from Kaggle .

The data are records of crimes that happened in San Fransisco ranging from 1/1/2003 to 5/13/2015, containing the incidents derived from SFPD Crime Incident Reporting system. The training set and test set rotate every week, meaning week 1,3,5,7... belong to test set, week 2,4,6,8 belong to training set.

For each data point, we have: category (only in training set), date, description (only in training set), day of week, PD district, resolution (only in training set), address, longitude, and latitude.

Features

1. In the Naive Bayes method, temporal coordinate, day of week, hour of a day, longitude, and latitude are used as features. Specially, the temporal coordinate is represented with the serial number of time intervals of 14 days.
2. For the Mixture of Gaussians model, the day of the week and month are mapped as a 1-hot signal. The regular time hour as well as the cosine of the hour are used.

Generalized Additive Model

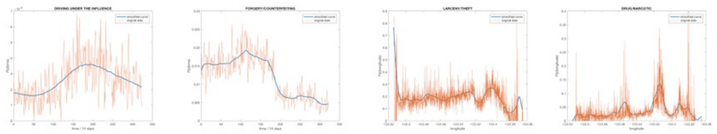
- **Basic formula:**

$$\begin{aligned} & P(z|x_1, x_2, ..., x_p) \\ &= \frac{P(z)P(x_1, x_2, ..., x_p|z)}{P(x_1, x_2, ..., x_p)} \\ &= \frac{P(z)\prod_i P(x_i|z)}{\prod_i P(x_i)} \\ &= \frac{\prod_i P(z|x_i)}{P(z)^{p-1}} \end{aligned}$$

here all the features are assumed to be independent and conditionally independent on the crime category z. By operating logarithm on both side, we obtain a GAM. Thus we can regress on probabilities conditioned on different features.

We used local weighted linear regression to estimate the conditional probabilities. Examples:

P(category|time): *P*(category|longitude):



* Parameter τ for local weighted regression is tuned using cross-validation.

- **Results (multi-class logarithmic loss):**

  training loss: 2.5345;
  test loss (Kaggle): 2.53674;
  leaderboard: 852 out of 2335.
  *The trivial prediction ranks 1563/2335; we moved forward 771 places!

- **Discussion:**

  The examples above show that, the tendency of probability for each category varies differently upon features (time, location, etc.) among different categories. And this is the cornerstone of our model.

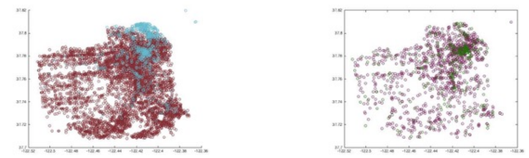
  The assumptions proposed in the model impose strong restrictions on the data. However, the two assumptions of independence and conditional independence can be verified, in some extent, by calculating the correlation between features, which are primarily in the order of 0.01 or less, except for that of longitude and latitude around 0.1.

Mixture of Gaussians Model

Under this model, we take the assumption that crime is a set of trends, where a trend is determined by a cluster of reports that are similar in location and time. We find underlying trends using the Mixture of Gaussians EM Algorithm.

$$\begin{aligned} P(i|x_1, x_2, \ldots, x_n) &= \sum_{j=1}^{k^{(i)}} P(trend^{(ij)}|x_1, x_2, \ldots, x_n) \\ &= \sum_{j=1}^{k^{(i)}} \frac{P(x_1, x_2, \ldots, x_n|trend^{(ij)})P(trend^{(ij)})}{P(x_1, x_2, ..., x_n)} \\ &\propto \sum_{j=1}^{k^{(i)}} P(x_1, x_2, \ldots, x_n|trend^{(ij)})P(trend^{(ij)}) \\ &\propto \sum_{j=1}^{k^{(i)}} P(x_1, x_2, \ldots, x_n|trend^{(ij)})N(trend^{(ij)}) \end{aligned}$$

Assault clusters: *Fraud clusters:*



- **Results (multi-class logarithmic loss):**

  training loss: 2.6787
  test loss: 2.6845

- **Discussion:**

  Moving to the EM-algorithm resulted in a modest performance increase over k-means.
  This method, as implemented, is not numerically stable. Heavy regularization had to be added to the EM Algorithm to avoid ill-conditioned covariance matrices.

- **Future:**

  Using Factor Analysis in small and rare classes may help improve numerical stability.
  Using smarter algorithms for choosing the amount of clusters could also result in a performance boost.

Prostitution clusters: