Crime Category Prediction of Cities Using an Ensemble of Various Classifiers

Hardy Hambsch   Marius Nolden   Jessica Ahring
Christian Peters

January 11, 2018

1 Introduction

In a time of limited assets regarding fighting crimes, this research aims to provide additional guidance to those distributing the resources by training machine learning classifiers to predict the criminal category of different cities.

After selecting the features of a city that have the most striking influence on its criminal category (i.e. high crime rate, medium crime rate or low crime rate), the classifiers are trained to learn the complex relations between these characteristics and the corresponding category. The different classifiers used for this task are namely k-Nearest Neighbors, Naïve Bayes, a Decision Tree and a Neural Network.

Each classifier is first trained on its own. The obtained models are then combined into an ensemble to merge the individual strengths of each classifier. The decision of the ensemble is found by conducting a majority voting. The metrics accuracy, precision, recall and f-measure are used to evaluate the results.

In the course of this research the open source data mining software WEKA is utilized to apply the algorithms to the dataset.

2 Dataset

Our dataset is based on real crime data which can be obtained from the Machine Learning Repository[1]. It contains 1994 instances. The whole set consists of a union of three subsets, namely the 1990 US Census, law enforcement data from the 1990 US LEMAS survey and crime data from the 1995 FBI UCR.

From the 128 attributes the dataset provides, only the most significant features are chosen. The exact procedure that lead to this subset of 13 attributes is described in a more detailed manner in section 4.2 on page 3. All chosen features (except, of course, for the class label) are continuous and normalized, no attribute has any missing values. A list of the attributes that will be used in the classification process is presented here:

- percentage of population that is african american
- percentage of population that is caucasian
- median household income
- percentage of households with investment / rent income in 1989

[1] http://archive.ics.uci.edu/ml/datasets/Communities+and+Crime

- percentage of households with public assistance income in 1989
- percentage of people under the poverty level
- percentage of people 25 and over with less than a 9th grade education
- percentage of people 16 and over, in the labor force, and unemployed
- percentage of people who are divorced
- percentage of kids in family housing with two parents
- percentage of kids born to never married
- percentage of people who do not speak English well
- percentage of people in owner occupied households
- crime category (the class to be predicted, either "High", "Medium" or "Low")

It should be noted that the class attribute "crime category" is an *artificial*[2] attribute, which is based on the attribute "total number of violent crimes per 100K population". This attribute, which is continuous in nature and therefore does not lend itself for classification, is essentially grouped into three classes ("High", "Medium" or "Low") based on its value. The details of this approach are outlined in section 4.1 on the next page.

3 Literature Review

A similar approach to ours was already developed by the authors of [Kha13] who employed WEKA for predicting the crime category using the same data as well. In order to do this, they selected a different set of features which is omitted here for the sake of saving space. Their method of selection as well as the assignment of the class labels "High", "Medium" and "Low" to each instance in the dataset is – countrary to our approach – solely based on subjective choice.

In order to perform the classification task, they chose two classifiers, namely a Naïve Bayesian and a Decision Tree. They did however use the same evaluation metrics as used in this paper, which allows for comparison of the different approaches.

In contrast to those ways of predicting crime there is a different approach illustrated in [Jon03]. In order to predict crime in the future they analyzed a dataset which contains the following attributes:

- Time
- Day
- Month
- Weather
- Location(geographical coordinates)

The records in the dataset are geographically mapped onto the certain regions of the United States of America. Using their own implementation of a cluster analysis they detect hot-spots of crime according to the given features. Afterwards an artificial neural network tries to forecast crime activity in the future.

4 Data Preparation

Before the classification can be done, a few preparation steps on the raw dataset are required, namely the assignment of class labels to each instance as well as feature selection to reduce the set of attributes to an acceptable size, i.e. to reduce the dimensionality of the feature space.

[2] this attribute is added by us

4.1 Assignment of Class Labels

To obtain the class labels used for classification, the continuous attribute "total number of violent crimes per 100K population" is divided into separate groups. Each group is assigned an ordinal class label that can be predicted during classification. If for example the number of groups is equal to three, one could assign the labels "High", "Medium" and "Low" to the corresponding groups.

The simple k-Means clustering algorithm is used to achieve this by comparing different amounts of groups, i.e. testing different values of k. The optimal value of k ($k = 3$) was determined using the elbow-method, as shown in figure 1.

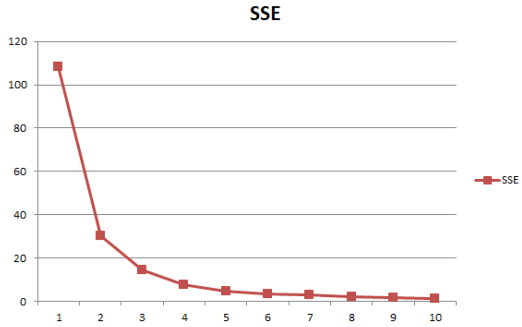

Figure 1: Sum of the squared errors for different choices of k

The three classes originating from this choice are namely "High", "Medium" and "Low", the percentage boundaries obtained during the clustering process result in the following intervals:

Low: $[0\%; 22\%]$
Medium: $(22\%; 56\%]$
High: $(56\%; 100\%]$

It should be noted, that the percentage boundaries are not to be confused with the actual crime rate because this feature was normalized before.

4.2 Feature Selection

To reduce the total of 128 attributes in the dataset to an acceptable amount, we employed a two-step approach that is presented in the following sections.

Selection by Judgement of Significance First we discussed each attribute separately and agreed whether we would include it based on its significance. In the course of this procedure, we obtained a ranking of the attributes we regarded as most significant.

Comparison with Correlation Ranking In the second step, we compared our choices with the attributes that have the highest correlation with the classes. We found that the majority of the attributes that we selected based on common sense was already present in the ranking. However, a few attributes like "population" were discarded because of a correlation near zero.

5 Experimental Results

The different approaches used for the classification purpose and the corresponding results are described in the following sections.

5.1 Approaches and Settings

k-Nearest Neighbors

The first classifier we chose is the k-Nearest Neighbors classifier. The parameter k was obtained by performing a parameter search on a leave-one-out cross validation. In terms of accuracy, $k = 30$ was the best choice. The distance measure is euclidean distance, the best results were obtained without distance weighting.

Naïve Bayes

Another approach is the Naïve Bayes classifier. Despite some attributes obviously not being independent, it was still employed because it proved to be useful in other areas as well.

Decision Tree

We generated a pruned J48-Decision Tree. The best results were achieved by using a confidence factor of $c = 0.045$ for pruning the tree. This value was found by utilizing WEKAs capabilities of linear parameter search.

Neural Network

A multilayer perceptron is used as a Neural Network. The structure consists of 13 input neurons (one for each feature), a hidden layer of 17 neurons precedes the output layer containing three neurons (one for each class). A learning rate of $\alpha = 0.1$ is used, a momentum of $m = 0.1$ is applied in each training step of the backpropagation algorithm. The number of hidden neurons was found by conducting a linear search once again, testing all possible values between zero and 30. The same procedure was applied in order to determine the optimal learning rate as well as the momentum term.

Ensemble

In hope for enhanced results by combining the strenghts of the different classifiers, all models were combined into an ensemble. The vote of the whole ensemble was found by conducting a majority voting.

5.2 Results

Each classifier as well as the ensemble was evaluated using 10-fold cross validation. As a result, the comparison of the different evaluation metrics can be found in figure 2.

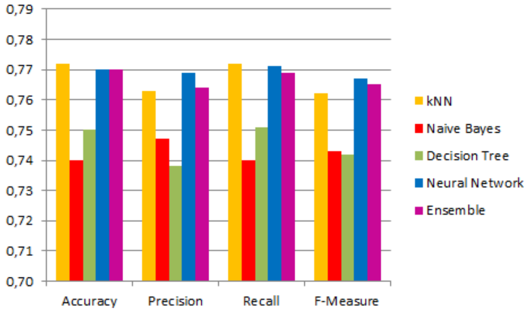

Figure 2: Evaluation metrics of the different classifiers

The detailed results of the ensemble learner are illustrated in the corresponding confusion matrix in table 1.

Table 1: Confusion matrix of the ensemble

classified →	**Low**	**Medium**	**High**	Total
Low	1125	126	8	1259
Medium	160	305	57	522
High	15	95	103	213
Total	1300	526	168	1994

The recall of the classifiers with respect to each class is given in figure 3.

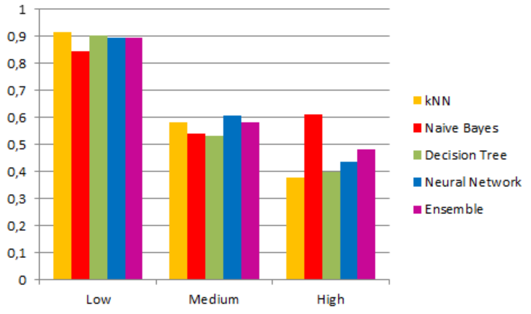

Figure 3: Recall of the different classifiers

5.3 Discussion and Outlook

As shown in figures 2 and 3, each classifier does indeed have its own strengths and weaknesses. While k-Nearest Neighbors excels in terms of accuracy and overall recall, the Neural Network performs best in terms of precision and f-measure. A fairly interesting observation can be made when looking at the different recall values with respect to the class "High": Here, the Naïve Bayesian classifier achieves by far the most impressive results when compared to the other models. This turns out to be a contrast to the overall scores of this classifier, which are merely average.

Contrary to our initial believe that the combination of all classifiers into an ensemble would enhance the results, we have to admit, that this was not quite the case. Overall, the ensemble always ranks among the best, but it never reached top spot in any category. In fact it turned out to be worse than the Neural Network in every category except for the recall with respect to the class "High".

This seemingly paradox incidence can be explained when thinking about the effects of combining different classifiers: When merging the different models, not only the strengths are combined but also the weaknesses. This results in the observed pattern: By combining the classifiers, the strengths and weaknesses average out, thus the obtained results are also average.

It should be noted however, that there are methods to bias the results of the ensemble more towards the stronger side of the spectrum. By giving each of the voters a different weight, one could emphasize the strengths of one classifier and mitigate the weaknesses of another.

Nevertheless, developing such a method to intelligently balance the weights of an ensemble would most likely require further study, which could form the basis of another research project.

References

[Jon03] J. Andrew Ware Jonathan J. Corcoran Ian D. Wilson. “Predicting the geo-temporal variations of crime and disorder”. English. In: *International Journal of Forecasting* 19(4) (Oct. 2003), pp. 623–634. URL: http://www.sciencedirect.com/science/article/pii/S0169207003000955.

[Kha13] Nasim Khanahmadliravi. “An Experimental Study of Classification Algorithms for Crime Prediction”. English. In: *Indian Journal of Science and Technology* 6(3) (Mar. 2013), pp. 4219–4225. ISSN: 0974-6846. URL: http://52.172.159.94/index.php/indjst/article/download/31230/27028.