Homework 4: Naïve Bayes Classification and Logistic Regression

CS 545: Machine Learning, Winter 2017

Ben Wilson

Introduction

Naïve Bayes classification uses a probabilistic approach to compute the likelihood that an instance belongs to a certain class. For each class, prior probability and conditional probabilities are calculated from the training set, assuming that the features are iid. The classifier then takes the class with the highest likelihood to be the prediction. In this assignment we used the same dataset as from Homework 3, the Spambase dataset from https://archive.ics.uci.edu/ml/datasets/Spambase.

Experiment 1: Classification with Naïve Bayes

The Spambase dataset was split into two, roughly equal, sets with proportionate numbers of each class. The first set of data was treated as the training data and was used to compute the prior probabilities of each class along with the mean and standard deviation of each feature assuming a Gaussian distribution.

For each instance of the test set, the likelihood that the instance belongs a class is computed. To make the calculations easier, some simplifications are made: since the denominator is the same for each class, it can be ignored; instead of calculating the product of many small values, the log of the product is used to avoid values getting too close to zero; after taking the natural log of the Gaussian distribution, it can be seen that the results rely only on the log of the variance and the mean squared error over the variance, additional constants and scalars can be safely ignored.

After running all instances of the test data through the Naïve Bayes classifier, results typical of the following are obtained (see Figure 1 on the following page for confusion matrix):

- Accuracy = 82.17%
- Precision = 70.10%
- Recall = 95.12%

The Naïve Bayes classifier didn't perform as well as the SVM on the same dataset. In the SMV implemented in Homework 3, the accuracy was typically

		Pred	
		P	***N***
Act	***P***	858	44
	N	366	1032

Fig. 1: Confusion Matrix for the Naïve Bayes Classifier

around 92.5% and the precision was around 91.5%. The only noticeable metric that the Naïve Bayes outperformed on was recall. By looking at the confusion matrix in Figure 1, it can be seen the that the classifier is far more likely to generate false positives than false negatives.

The assumption that the dataset is independent seems, intuitively, to be false. Many of the features measured in this dataset tend to appear together; the number of dollar signs, overuse of the recipients name, and strings of capitalized letters, for instance. However, the Naïve Bayes classifier is accurate +80% percent in this case where a random guess would only be correct roughly 40% of the time.

One rational for why the Naïve Bayes classifier works as well as it does seems to come from the fact that the mean and standard deviation of strong indicating features (e.g. dollar signs and capital letters) carry a lot of weight when making predictions. If these values tend to dominate the calculations, it might explain why the classifier tended to predict positive classes in a majority of instances, even though positive cases represented a minority.

Experiment 2: Classification with Logistic Regression

For the last experiment, a Logistic classifier is applied to the same dataset as experiment 1. The LogisticRegression classifier was used from the sklearn Python module. After trying multiple settings, it was found that the default setting provided the best results. Namely, the liblinear solver, l1 penalty, and an inverse regularization of 1 was used.

The Logistic classifier performed much better than the Naïve Bayes, but roughly the same as the SVM. The Logistic classifier had higher accuracy and precision over the Naïve Bayes, but with 3% less recall. The Logistic classifier seemed to perform just as well as the SVM, if not slightly better in some cases.

After running all instances of the test data through the Logistic classifier, results typical of the following are obtained (see Figure 2 on the following page for confusion matrix):

- Accuracy = 91.57%
- Precision = 89.86%
- Recall = 88.47%

		Pred	
		P	***N***
Act	***P***	798	104
	N	90	1308

Fig. 2: Confusion Matrix for the Logistic Classifier