JMLR: Workshop and Conference Proceedings 1:1–5, 2016 Submitted 11/5/2016; Published 11/5/2016

Introduction to Machine Learning - EN 605.449.81 - Lab 5

Max Henry MHENRY22@JHU.EDU

Abstract

In this lab, naive Bayes, logistic regression and two neural networks, the Perceptron and Adaline, are compared using five classification datasets. These groups of learners come from the linear model family of algorithms. Linear models are limited to learning concepts from data that is linearly separable. My results show that naive Bayes is the best perform, logistic regression is the worst, and the neural networks perform on par with each other.

Keywords: naive Bayes, logistic regression, Perceptron, Adaline

1. Problem Statement

The purpose of the assignment was to implement and compare the performance of linear models on five classification data sets. The linear models used include naive Bayes, logistic regression, and the neural networks Perceptron and Adaline. I hypothesize that the Perceptron and Adaline networks will performance in lock-step with each other and naive Bayes will be the best performer.

2. Algorithms

2.1. Naive Bayes

Naive Bayes predicts a class label from $argmax_c P(c|f_1, ..., f_d)$ where c is the certain class label and $f_1, ..., f_d$ are the attributes of the unseen instance. It leverages Bayes Rule by trying to predict a class label given the unseen instance. It makes the assumption that all features f_i are conditionally independent of each other given the class label. This means the calculation becomes $argmax_c P(c)\Pi_{i=1}^{d} P(f_i|d)$.

2.2. Logistic Regression

Logistic regression is a generalized linear model that measures the probability of a data instance belonging to a certain class by estimating the relationship between data attributes and the class label with the logistic function. During logistic regression, one learns a weight for each data attribute and predicts the probability of the data instance belonging to that class by calculating the dot product of the weights and the data instance and using it to estimate $P(class_label|instance)$ according to

$$P(class_label|instance) = e^w/(1 + e^w)$$

where w is the aforementioned dot product. This calculation is used to estimate the probability of a data instance's memebership to a class and need run for each class besides one,

which can be estimated by calculating one minus the sum of the other probabilities. To learn the weights for each attribute the following update rule is used:

$$w_{ji+1} \leftarrow w_{ji} + \eta \sum_k x_i^k * \delta(c^k = c_j) - P(c^k = c_j | x^k, \mathbb{W}))$$

where j is the class label, i is the attribute, k is a particular training example and delta is the Kronecker delta.

2.3. Perceptron

A Perceptron neural network is a single-layer neural network which learns weights for each attribute of the input instance. The dot product of these weights and the data instance is inputted into an activation function to determine the classification of the instance. In these experiments, a sigmoid function is used as the activation function. To learn the weights of the network, one predicts the classification for a particular data instance. If that prediction matches the instance's class label, the weights are not updated. In a 0/1 classification problem, if the prediction does not match and predicts 0, then the instance is subtracted from the current weights to create new weights. If the prediction does not match and predicts 1, then the instance is added to the current weights to create new weights. Learning ends when all the training instances are correctly classified or the weights are not updated by a given threshold. The Perceptron weights are guaranteed to converge for problems that are linearly-separable.

2.4. Adaline

An Adaline neural network is a single-layer neural network which learns weights for each attribute of the input instance.The dot product of these weights and the data instance is inputted into an activation function to determine the classification of the instance. In these experiments, a sigmoid function is used as the activation function. The difference between an Adaline and Perceptron neural network occurs when learning the attribute weights. Whereas the Perceptron adjusts weights after the activation function has been applied, an Adaline network updates the weights with the raw output of the dot product of the weights and the instance. The weights are updated according to

$$w_{t+1} \leftarrow w_t + \eta(o - y)x$$

where o is the correct class label, y is the raw output of the dot product and x is the data instance.

3. Experimental Approach

3.1. Data Cleaning

Due to the unprocessed nature of the data, each continuous-valued dataset was discretized using 10 bins of equal size between the attributes min and max value. In order to be fair to both algorithms, discretized attributes were translated with a one-hot encoder, mostly to 10-bit strings. In the case of multi-class (N classes), the original dataset was split into

N datasets, one for each class, and translated into a 0/1 class setup for all models except logistic regression, which was implemented to handle multi-class problems. In addition, any missing values were replaced with a random number draw from the distribution of instances with the same class label.

3.2. Experiments

Each algorithm was run against each of the five datasets using 10-fold cross-validation. The success rate, calculated as the number of correctly labeled instances out of the total number of instances in the test set, was averaged over every fold. This number was used to compare the algorithms.

4. Results

Table 1: Classification Accuracy

Dataset	Naive Bayes	Perceptron	Adaline
Breast Cancer	96.52%	65.65%	74.35%
Glass (building_windows_float_processed)	70.48%	66.67%	49.05%
Glass (building_windows_non_float_processed)	69.05%	64.76%	46..67%
Glass (vehicle_windows_float_processed)	82.38%	91.90%	36.19%
Glass (vehicle_windows_non_float_processed)	100%	100%	53.81%
Glass (containers)	95.24%	93.81%	48.57%
Glass (tableware)	97.62%	95.71%	48.10%
Glass (headlamps)	97.14%	86.19%	48.10%
Iris (Setosa)	100%	66.67%	64%
Iris (Versicolour)	94.67%	66.67%	66.67%
Iris (Virginica)	90.67%	66.67%	82%
Soybean (D1)	100%	77.50%	80%
Soybean (D2)	100%	77.50%	77.50%
Soybean (D3)	100%	77.50%	80%
Soybean (D4)	100%	40%	70%
Voting	91.63%	61.63%	61.62%

Table 2: Logistic Regression Accuracy

Dataset	Logistic Regression
Breast Cancer	37.68%
Glass	12.86%
Iris	33.33%
Soybean	22.50%
Voting	39.07%

5. Conclusions

Naive Bayes outperforms every other model on every dataset besides one. In most cases, it is dominant. In Ng and Jordan, it is proposed that naive Bayes converges quicker than logistic regression, which would give naive Bayes an advantage for smaller data sets. This is important for these datasets because the largest dataset only had 800 examples in it. This could explain the performance gap between naive Bayes and logistic regression. In addition, multinomial logistic regression was used whereas the naive Bayes model was given each dataset as a 0/1 classification problem. For a clearer comparison, new experiments should be run with binary logistic regression instead.

In the case of the Perceptron and Adaline, their performances were similar for most datasets excluding the Glass dataset. In most cases, Adaline slightly outperformed the Perceptron. Given the similar performances on nearly all the datasets, Adaline's performance on the Glass data set should be seen as an outlier. In general, the Perceptron and Adaline performed on par with each other.

6. Summary

In this lab, naive Bayes, logistic regression, and two neural networks had their performances on five classification datasets. Naive Bayes was the top performer, followed by the Adaline network, the Perceptron and finally logistic regression.

7. References

[1] UCI machine learning repository: Breast cancer Wisconsin (diagnostic) data set. (1995, November 1). Retrieved September 11, 2016, from https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wi

[2] UCI machine learning repository: Iris Data set. (1988, July 1). Retrieved September 11, 2016, from https://archive.ics.uci.edu/ml/datasets/Iris

[3] Index of /ml/machine-learning-databases/soybean. Retrieved September 11, 2016, from https://archive.ics.uci.edu/ml/machine-learning-databases/soybean/

[4] UCI machine learning repository: Congressional voting records data set. (1987, April 27). Retrieved September 11, 2016, from https://archive.ics.uci.edu/ml/datasets/Congressional+Voting+Records

[5] Rish, Irina. "An empirical study of the naive Bayes classifier." IJCAI 2001 workshop on empirical methods in artificial intelligence. Vol. 3. No. 22. IBM New York, 2001.

[6] Peng, Chao-Ying Joanne, Kuk Lida Lee, and Gary M. Ingersoll. "An introduction to logistic regression analysis and reporting." The journal of educational research 96.1 (2002): 3-14.

[7] Rosenblatt, Frank. "The perceptron: a probabilistic model for information storage and organization in the brain." Psychological review 65.6 (1958): 386.

[8] Widrow, Bernard. Adaptive" adaline" Neuron Using Chemical" memistors.". 1960.

[9] UCI machine learning repository: Glass identification data set. (1987, September 1). Retrieved September 11, 2016, from https://archive.ics.uci.edu/ml/datasets/Glass+Identification

[10] Jordan, A. "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes." Advances in neural information processing systems 14 (2002): 841.