Classifying Leaves with Machine Learning Techniques

Computer Science Department
College of Wooster
Wooster, OH

Computer Science Department
College of Wooster
Wooster, OH

Abstract

Machine learning techniques are used to classify leaves based on their shapes. The performance of naive Bayes classification, decision trees, and k-Nearest Neighbor classification are compared. The technique with the best average performance is decision tree classification with 65.36% accuracy over testing data. We also evaluate the change in performance of these techniques after applying Principal Component Analysis to reduce the number of attributes in the data set.

Keywords: classifying leaves, decision trees, Bayesian learning, kNN, principal component analysis.

Introduction

In this paper, we will be working with a data set that contains 100 different leaves with 16 examples for each (Mallah, Cope, and Orwell 2013). It has 64 attributes, based on the shapes of the leaves. Different approaches used to obtain the attributes from images of leaves will be discussed. The performances of decision trees, neural networks, naive Bayes classifier, and k-nearest neighbor algorithms, will be compared. Principal component analysis (PCA) will be applied to the data set to reduce the number of attributes. The tradeoff between computation time and accuracy will be evaluated for varying numbers of attributes (after PCA).

Obtaining Data From Leaf Images

Several steps are used to obtain shape signatures from images of leaves (Beghin et al. 2010). The first step is to use Otsu's thresholding method to get greyscale images, and the second step is to get two contour signatures from these images. The first contour signature measures the distance between the contour points and the center of the leaf, and the second signature measures the angles between the contour points and the center point (Beghin et al. 2010). A Gaussian filter is used to correctly differentiate lobed leaves from serrated leaves (Beghin et al. 2010). The results from both shape and texture analysis methods in (Beghin et al. 2010) are fairly poor, but when the two data sets are combined, better results are achieved.

Our data set uses images of leaves and extracts the attributes with the following methods (Mallah, Cope, and Orwell 2013). To extract texture data from leaves, their vein fabric can be analyzed. In (Cope et al. 2010), three different texture analysis methods are explored: Gabor filters, co-occurrence matrices, and Fourier descriptors. In testing, their own method is introduced that combined these techniques to achieve better results.

This dataset contains 100 species of leaves with 16 samples for each, giving us 1,600 instances (Mallah, Cope, and Orwell 2013). Each sample has 64 attributes obtained with these methods (Mallah, Cope, and Orwell 2013). Figure 1 illustrates how two leaves from the same species can vary greatly.

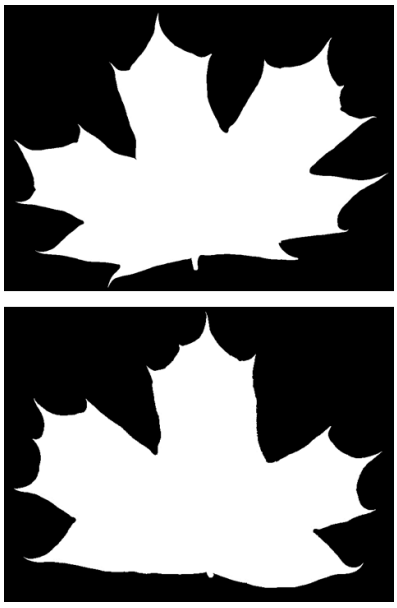

Figure 1: Two leaves both from the species Acer Platanoids illustrate the variation of different leaves within the same species (Mallah, Cope, and Orwell 2013).

Data Preparation

Two different random shuffling approaches are applied to the data before using machine learning techniques. The first approach randomly shuffles and splits the data into training data with 1,200 instances (75%) and testing data with the remaining 400 (25%) instances. The second approach uses the same training and testing data sizes, but randomly selects 12 instances from each species. This equal representation shuffling approach prevents species from being over or under represented. To reduce the effect of outliers on test results, we averaged results over 50 trials.

Naive Bayes

One machine learning technique tested is the naive Bayes classifier. This technique uses Bayesian learning with the assumption that all attributes are independent, to predict the most likely classification for an instance (Mitchell 1997). The naive Bayes classifier correctly classifies 62.69% of the entire shape data set (without separating the data set into training and testing). The similar method conducted in (Beghin et al. 2010), achieves 69.2% accuracy, although only 18 species of trees are examined and the testing and training percentages are not mentioned. We observe from the confusion matrix that some species of leaves are perfectly classified, while others are completely misclassified. This is a common issue for leaf classification because leaves from the same species can vary greatly. Figure 2, illustrates the performance of the naive Bayes classifier for all 100 species. When using the naive Bayes classifier for new instances, it performs poorly.

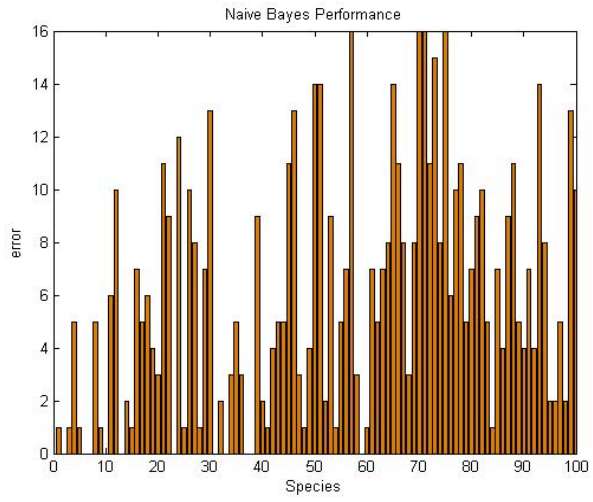

Figure 2: Bar graph of Naive Bayes performance.

Decision Trees

Another technique we test is classification with decision trees. Decision trees correctly classify 65.36% of the instances with the equal representation shuffling approach and 64.77% with the first random shuffling approach. These trees created from the equal representation shuffling approach correctly classify an average of 71.49% over the entire data set. The performance of one of these trees is shown below in Figure 3.

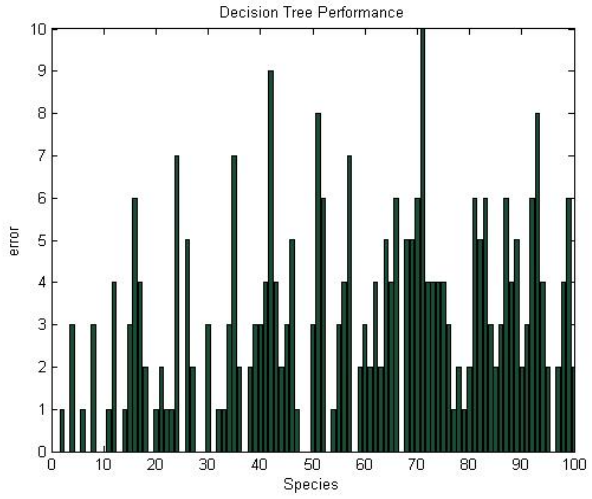

Figure 3: Bar graph of classification tree performance.

K-Nearest Neighbor

We apply the k-Nearest Neighbor algorithm, which plots the instances in $\mathcal{R}^n$, where n is the number of attributes, and classifies new instances based on their location (Mitchell 1997). For our work, Euclidean distance is used to find the k nearest points used to classify new instances. With $k = 1$, we obtain the highest accuracies with 62.27% for random shuffling and 53.49% for shuffling with equal representation. Figure 4 shows an example performance of this technique.

Principal Component Analysis

Principal component analysis can be used to reduce the number of attributes in a data set if they are correlated with

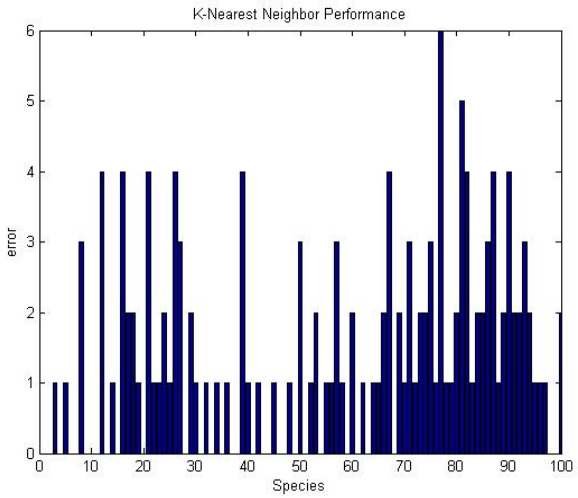

Figure 4: Bar graph of K-Nearest Neighbor performance.

each other (Jolliffe 2002). We use PCA to transform the 64 attributes into principal components that are ordered in decreasing variance. These components are generated to be orthogonal to each other so they are uncorrelated (Jolliffe 2002). The first five principal components account for 97.6% of the variance. We create a new data set with these five components.

Performance After PCA

The performances of the previously tested techniques are reevaluated with the new data set obtained after PCA. As listed in Table 1, decision trees have the lowest decrease in performance. The increased performance for naive Bayes classification is likely caused by the generation of the principal components to be orthogonal to each other, because naive Bayes uses the assumption that all attributes are uncorrelated.

Table 1: Comparison of the Machine Learning techniques' performances before and after applying PCA.

TECHNIQUE	ORIGINAL DATA (%)	PCA DATA (%)
DECISION TREES	65.36	63.84
KNN (EQUAL SELECTION)	53.49	46.49
KNN (RANDOM SHUFFLE)	62.27	52.81
NAIVE BAYES	62.69	64.75

To see how the number of principal components affects accuracy, we test the results for decision trees with random shuffling. We vary the number of principal components from 1 to 8 and the results are shown in Figure 5. This shows that after 5 principal components are included, the increase in performance for each additional component drops significantly. We also measure the total time elapsed after 50 trials. As Figure 6 shows, the increase in run-time is approximately linear after selecting two principal components.

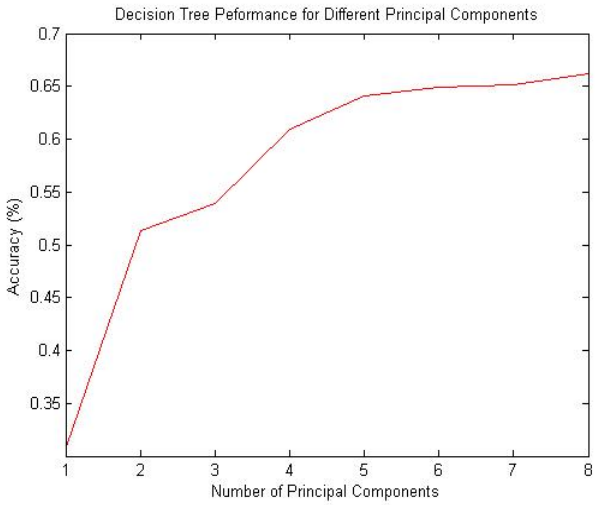

Figure 5: The performances of decision trees for varying numbers of principal components.

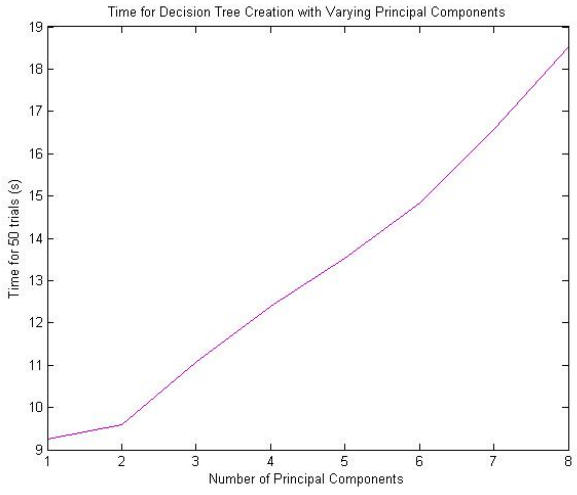

Figure 6: The time elapsed for 50 decision trees trials with varying numbers of principal components.

Conclusions and Future Work

After testing the naive Bayes classifier, decision trees, and k-Nearest Neighbor, we found that decision trees have the best performance. The performance of decision trees is 65.36%, which shows the need to combine more attributes (from different leaf signatures) to obtain better classification. Principal component analysis gives promising results for attribute reduction. The five principal components with 97.6% of the variance allow for a small decrease in performance, while significantly improving the efficiency of data analysis.

Future possibilities for this research include the combining of additional leaf data sets after PCA, and determining what other leaf attributes could be useful for classification. We would also like to explore the causes behind the kNN performance differences. A useful extension of this work would be a mobile application for active species classification for images of leaves.

References

Beghin, T.; Cope, J.; Remagnino, P.; and Barman, S. 2010. Shape and Texture Based Plant Leaf Classification. *Advanced Concepts for Intelligent Vision Systems* 345–353.

Cope, J.; Remagnino, P.; Barman, S.; and Wilkin, P. 2010. Plant Texture Classification Using Gabor Co-Occurrences. *Advances in Visual Computing* 669–677.

Jolliffe, I. T. 2002. *Principal Component Analysis, Second Edition*. Springer.

Mallah, C.; Cope, J.; and Orwell, J. 2013. Plant Leaf Classification Using Probabilistic Integration of Shape, Texture and Margin Features. Pattern Recognition and Applications, in press.

Mitchell, T. M. 1997. *Machine Learning*. MIT Press and The McGraw-Hill Companies, Inc.