Classification of Segmentation data

Ramya Rao and Shwethambari Surendran

Indiana University , April 27th 2017

Abstract

In this paper, we have used machine learning algorithms to create a model that would learn and classify a data set containing segmented image data of seven outdoor images denoting the following seven classes - brickface, sky, foliage, cement, window, path, grass.The aim is to observe and visualize patterns in the data so that we may use best suited algorithms to create a learning model that would classify an independent test data set with high accuracy.

Introduction

The objective of this paper is to effectively classify segmented image data . This paper uses image segmentation data from the UCI Machine learning repository. We aim to identify attributes relevant to the image that are indicative of a particular class of outside imagery.

The images consist of attributes relating to seven different classes - brickface(Class 1),cement(Class 2),foliage (Class 3), grass (Class 4) ,path (Class 5) ,sky (Class 6) and window (Class 7) . To start classifying the images we employ machine learning algorithms that learn from a few classified category of images which are part of the training set.We first ensure that the data is represented correctly and try to filter features that may be redundant or highly correlated with the rest of the features.

The importance of the features are evaluated by plotting them graphically into a series of plots.We then evaluate the nature of the data and use the appropriate classifiers like Naive Bayes, SVM, KNN and decision trees for classification.We look at classifiers that will maximize the accuracy of the classification and also use ensemble methods like bagging and boosting to increase the efficiency of classification.

Data Set description

The data set that we have used in this paper is obtained from the UCI Machine Learning Repository titled Image Segmentation Data [5] and contains instances drawn randomly from a database of 7 outdoor images. These images are handsegmented and each one is a 3x3 region.

The 7 outdoor images fall in the classes of brickface(Class 1),cement(Class 2),foliage (Class 3), grass (Class 4) ,path (Class 5) ,sky (Class 6) and window (Class 7). There are 19 attributes for each record and are continuous in nature.The data is divided into 210 training records and 2100 test records . The predictors for each record are based on the results of numerous kinds of analyses on colour performed on the data like line extraction algorithms applied on the region to measure contrast and colour attributes in the RGB space .Features like hue and saturation and vedge-mean identify the presence of horizontal contrast which enhance the clarity of classification.There are no missing values in the dataset and there are 30 instances of each class in the training set. The test dataset contains 300 instances per class.

Feature Visualization And Principal Component Analysis

The dimension each record is 20 with the first column denoting the class and they are the result of different analyses performed in the color space of the images to relate them with the surface of the identifying outdoor class. The 19 numerical attributes are all continuous in nature.We plot the features to enable us to identify and filter feature components that may be redundant or too correlated in nature.We could just select a subset of features that are relevant and visualizing them through plots would help significantly.

The features of the data ,when inspected ,show that some of them are highly interrelated or redundant. For example , the columns for short-line-density-5 and short line-density 2,vedge-mean and vedge-sd and hedge-mean and hedge-sd are basically copies of each other.

There are different plots that can be used to visualize the given data set like scatterplot and boxplot as given below.

Scatter Plot

We construct a scatterplot of the features using the plot() function in R. On plotting the features against each other, we notice that the columns numbered 11 to 18 have strong correlation with each other as they are all calculations in the RGB space.

Box Plot

The visualization of the boxplot of the features will show how much the range of values for the predictor variables vary. We see that there are instances where the values of certain predictor variables remain constant like the columns for region-pixel-count owing to the same number of pixels being present in every 3x3 region.

Principal Component Analysis

Principal Component Analysis helps identify the predictors that have the most variance and represent the essential crux of the data.PCA tries to identify the feature points that represent

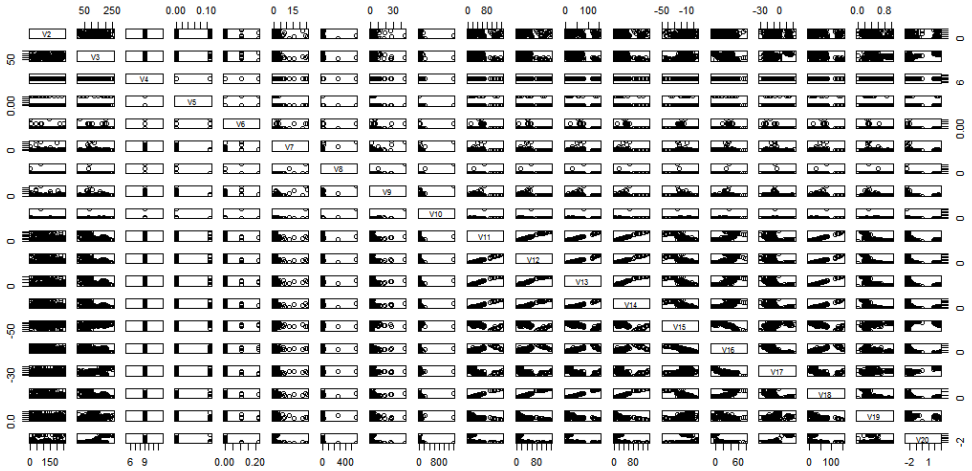

Figure 1: Scatterplot of all features in training data.

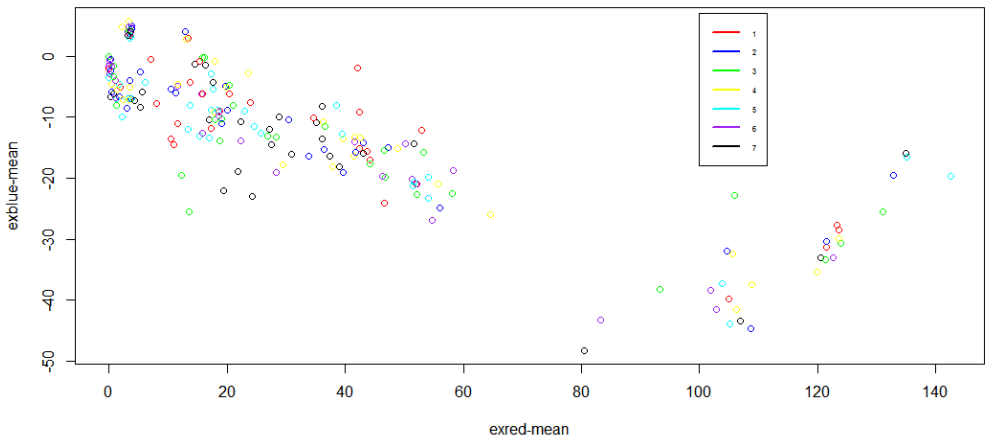

Figure 2: Scatterplot between features - Exred-mean and Exblue-mean.

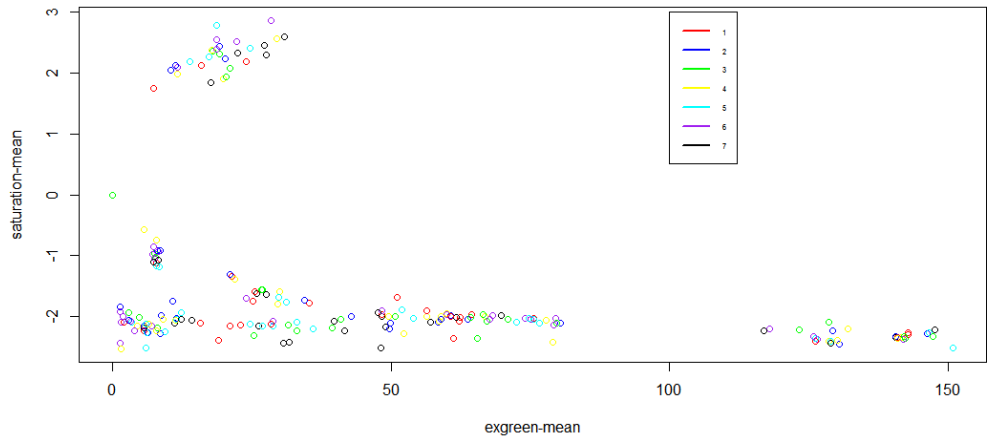

Figure 3: Scatterplot between feature - ExGreen-mean and Saturation-mean.

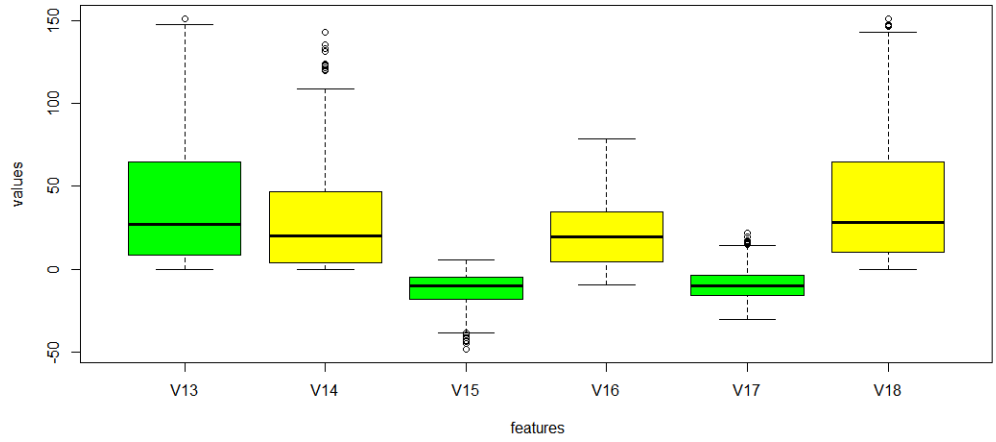

Figure 4: Boxplot of features.

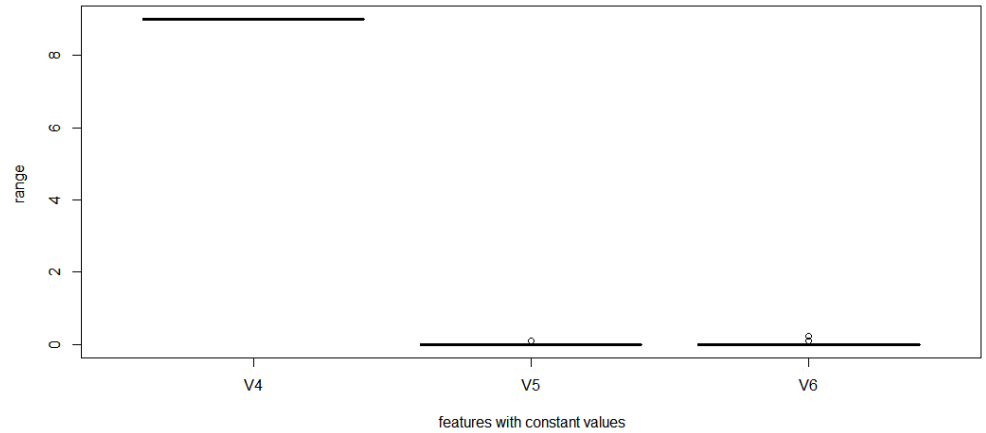

Figure 5: Boxplot of features with constant values.

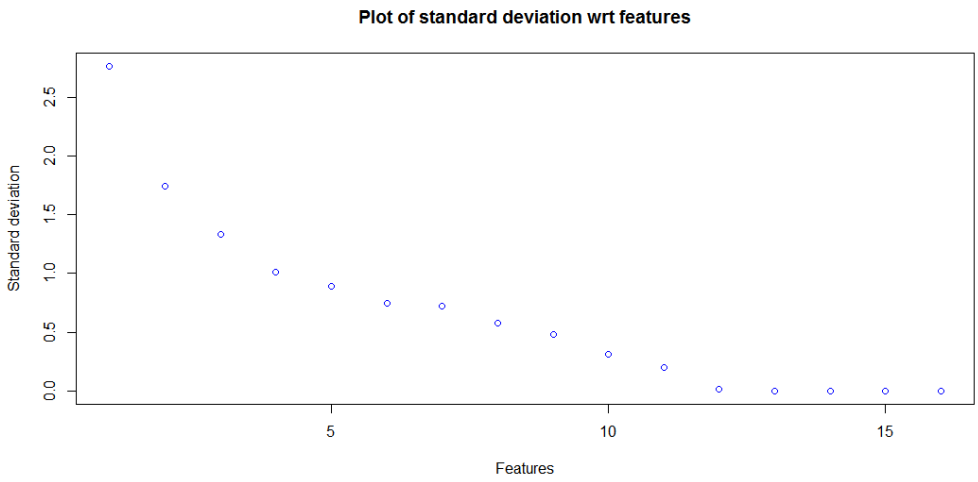

Figure 6: Plot of standard deviation for features using PCA

the data most accurately and thus also reduce the dimensionality of excessively correlated and redundant data.We see that the deviation among the values of the predictors is not much after the ninth or tenth feature point.Thus the first nine to ten principal components show maximum variance and are the most essential components for prediction.[1]

Choosing Classifiers

The image dataset is moderately long and has two independent sets for test and train. The attributes for each record are all numerical continuous values and the classes are categorical in nature , each being a class depicting part of an outdoor imagery or surface. While choosing classifiers that would help us learn the dependencies of the train set and then predict classes for the unknown test data , we look for classifiers that could predict with high accuracy and are not very complex in performance.

Naive Bayes Classifier would be an appropriate choice as they are good for medium length data sets with high bias and low variance.Because of conditional independence assumption,they will help us converge faster and if correlated data is not part of the set, they are efficient.

Support Vector Machines are also a good choice as they are capable of setting complex geometric boundaries for the hyperplane that help separate the data into various classes seamlessly in a high dimensional space.They tend to generalize well without the risk of overfitting and as we have seven classes here they would be a good choice of classifier.

Decision trees work well for datasets of the size of the image dataset and are fast and work well with non linear data. For our image dataset decision tree classifier would be great as it would help classify the categorical data without worrying about outliers.

We could further use ensemble methods like the Ada boost classifier which help strengthen weak classifiers and help pick out the most optimal classifier and also Bagging to increase the classification accuracy.We can also use k Nearest Neighbours, although they are more sensitive to outliers and need a huge amount of data for efficient learning.

	class.test						
results	1	2	3	4	5	6	7
1	248	6	2	0	0	0	5
2	13	253	8	0	32	2	19
3	0	9	52	0	12	0	10
4	0	0	0	296	0	0	0
5	0	7	0	0	254	0	0
6	0	0	1	0	0	298	0
7	39	25	237	4	2	0	266

Figure 7: Confusion Matrix for Naive Bayes Model

	class.test						
classes	1	2	3	4	5	6	7
1	271	2	2	0	0	0	8
2	17	282	19	0	7	0	38
3	11	3	255	0	0	0	59
4	0	0	2	297	0	0	0
5	0	3	0	3	293	0	0
6	0	0	1	0	0	300	0
7	1	10	21	0	0	0	195

Figure 8: Confusion Matrix for Decision Tree Model

The classes are numbered as follows : brickface(Class 1),cement(Class 2),foliage (Class 3), grass (Class 4), path (Class 5), sky (Class 6) and window (Class 7).

Naive Bayes

The Naive Bayes model is used to learn the patterns of the training set of 210 records by implementing conditional independence among the features.The prior and posterior are calculated for each of the seven classes for all values of the predictor variables and then used to learn the model. We use the naiveBayes model in R from the e1071 package. We create the model from the training data and then test it on the test data of 2100 records.The accuracy of the classifer is around 79.4 percent and the confusion matrix for the predictions is shown in Figure 7.[2]

Decision Trees

We create a classification decision tree for the training data using the rpart() function in R.It automatically uses Gini impurities to select splits and the complexity parameter is used to stop growing the tree if the complexity does not decrease with further splits.The decision tree performs extremely well when it is tested for classification on test data and the accuracy is 90.2 percent.The confusion matrix for predictions is given in Figure 8.

	class.test						
classes	1	2	3	4	5	6	7
1	271	2	2	0	0	0	8
2	17	282	19	0	7	0	38
3	11	3	255	0	0	0	59
4	0	0	2	297	0	0	0
5	0	3	0	3	293	0	0
6	0	0	1	0	0	300	0
7	1	10	21	0	0	0	195

Figure 9: Confusion Matrix for Support Vector Machine Model

Support Vector Machines

SVM model is created by using the svm package in R and it creates a hyperplane with the widest margin that will effectively classify the records in to one of the seven classes.The model is created on the training set and we cross validate the data to obtain an accurate model.We use C type classification and it has a radial kernel.When tested on the test set , it provides a very decent accuracy rate of 89 percent. The corresponding confusion matrix is shown in Figure 9.[3]

Ada Boost Classifier

We also try to implement one of the boosting methodologies using Ada Boost Classifier uses a combination of weak learning classifiers and manipulates them to be sensitive to wrongly classified data and to outliers. We eventually obtain an optimal classifier. Ada boost performed well on the image segmentation data giving an accuracy of 92.4 percent.[4]

Bagging

Bagging obtains subsets of the dataset and trains the algorithm on each set separately and gets an average of the predictions at the end for majority voting. These separate sets are said to be independent of each other.Bagging has a pretty low accuracy in this case and it gives an accuracy value of around 16.4 percent.

K Nearest Neighbours

KNN is very efficient at classification by calculating the distance metrics between points to efficiently cluster the most closest points into one class. They invariably provide good results when the data does not have too much variance and has low bias.They however require a larger training set. In our case however the error rate for the classifier seems to increase with the value of k. The results are poor with KNN when the value of k is 100 and give only about 20 percent accuracy.

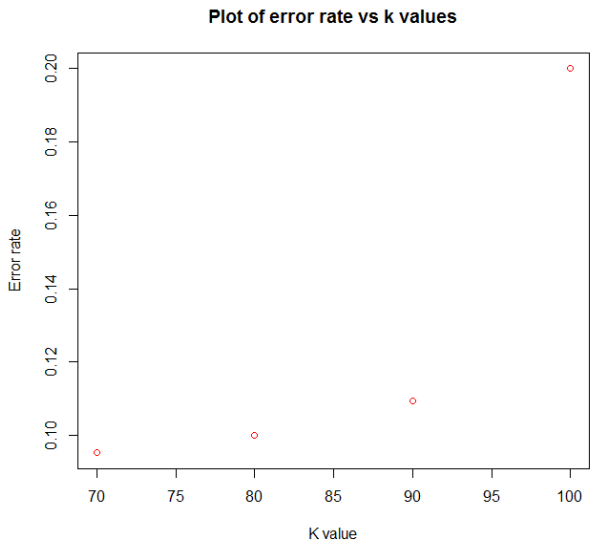

Figure 10: Plot of KNN error rate versus k values

Results

We see that the performance of the image dataset with models like Naive Bayes,SVM and Decision trees are pretty good .The data holds well with conditions of independence and is simple with continuous numerical feature variables and a categorical labeled class. Out of all the classifiers, Ada Boost had the best results with a classification accuracy of 92.4 percent.

Conclusion

Thus the image segmentation data from the UCI Machine Learning Repository was analyzed and used for creating models of classification using different classification algorithms and was tested on an independent test data set. The accuracy of the model was best when learning happened with Naive Bayes ,SVM and Decision trees and also worked well with ensemble methods like the Ada Boost Classifier.

References

[1] https://tgmstat.wordpress.com/2013/11/28/computing-and-visualizing-pca-in-r/.

[2] http://ugrad.stat.ubc.ca/R/library/e1071/html/naiveBayes.html.

[3] https://cran.r-project.org/web/packages/e1071/e1071.pdf.

[4] https://cran.r-project.org/web/packages/adabag/adabag.pdf.

[5] Vision Group, U. O. M. Image segmentation data. http://archive.ics.uci.edu/ml/datasets/Image+Segmentation, November, 1990.