Letter Recognition using Gradient Boosting Machines

Yuan Wang

December, 2016

1 Data Overview

In this part, we apply gradient boosting to the *Letter Image Recognition Data Set* to solve a multiclass classification problem. This data set is available from UCI Machine Learning Repository. The original data contains information of 26 capital letters (A-Z) with different fonts. Each observation is one image of black-and-white rectangular pixels, which displays as one of the 26 capital letters. Here we only use a subset of letter B, C, D, G, O and Q, and the goal is to distinguish the six letters.

The data set includes one letter category (B,C,D,G,O,Q) and 16 numeric features, with a sample size of 4616 (B 766, C 736, D 805, G 773, O 753, Q 783). All the predictor variables (statistical moments and edge counts) have been scaled to fit into a range of integer values from 0 to 15.

Table 1: Variable Information

Variable	Description
Class	capital letter(B, C, D, G, O, Q)
x-box	horizontal position of box
y-box	vertical position of box
width	width of box
high	height of box
onpix	total # on pixels
x-bar	mean x of on pixels in box
y-bar	mean y of on pixels in box
x2bar	mean x variance
y2bar	mean y variance
xybar	mean x y correlation
x2ybr	mean of x * x * y
xy2br	mean of x * y * y
x-ege	mean edge count left to right
xegvy	correlation of x-ege with y
y-ege	mean edge count bottom to top
yegvx	correlation of y-ege with x

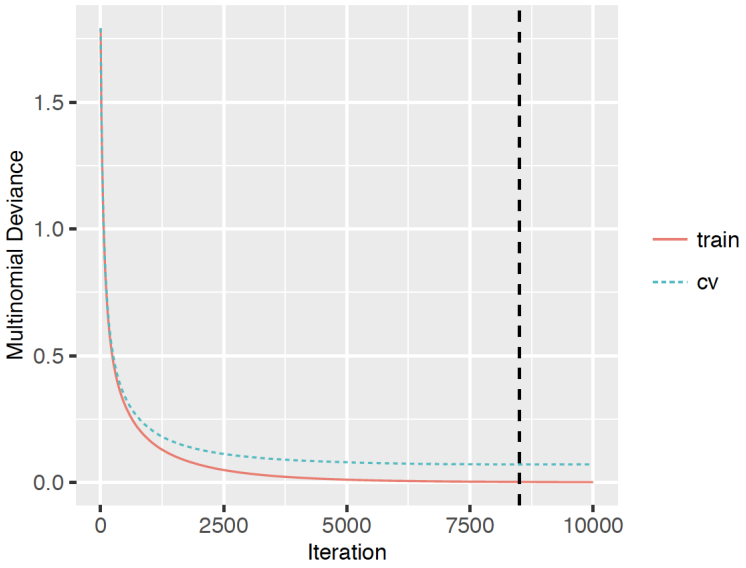
2 Building the Model

We divided the data into a training set(80%) and test set(20%) randomly, using $J = 4$ node trees with a learning rate $\nu = 0.01$, sub-sampling rate 50% and iteration time $M = 10000$.

Then we use the 10-fold cross validation to estimate the optimal number of iterations after the gbm model has been fit.

Figure 1 plots the deviance of train (red curve) and cross validation (blue curve) at each iteration. The deviance of training set is monotonic decreasing. The deviance of cross validation decreases until additional iterations cause it to increase and over-fitting exists. Thus, we choose the number of iterations that gives minimal cross validation deviance to reduce the risk of over-fitting.

Figure 1 Train and cv error at each iteration



From the method, we find that the optimal iteration time is 8508, which is shown by the black vertical line in Figure 1.

3 Model Interpretation

Figure 2 Relative importance of predictors

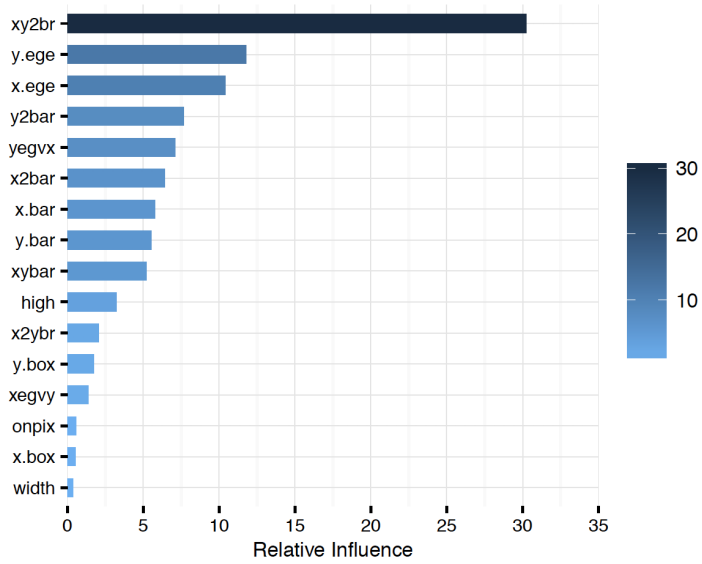


Figure 2 shows the relative importance of each of the 16 predictors, as averaged over all classes. The importance is defined to be the reduction of loss for each variable under all iterations. It is a useful way to measure the influence of each variable and to conduct variable selection.

xy2br is the most relevant predictor, which measures the correlation of the vertical variance with the horizontal position of the rectangular pixels [6]. Predictors related with mean edge count like *y.ege* and *x.ege* also have great importance, while *onpix*, *x.box* and *width* are much less influential.

4 Model Evaluation

Now we evaluate our model on the test set with a size of 924. Using the model to classify the test set, we get an error rate of 2.38%, indicating that GBM performs pretty well on this data set.

Figure 4 Confusion matrix of the test set

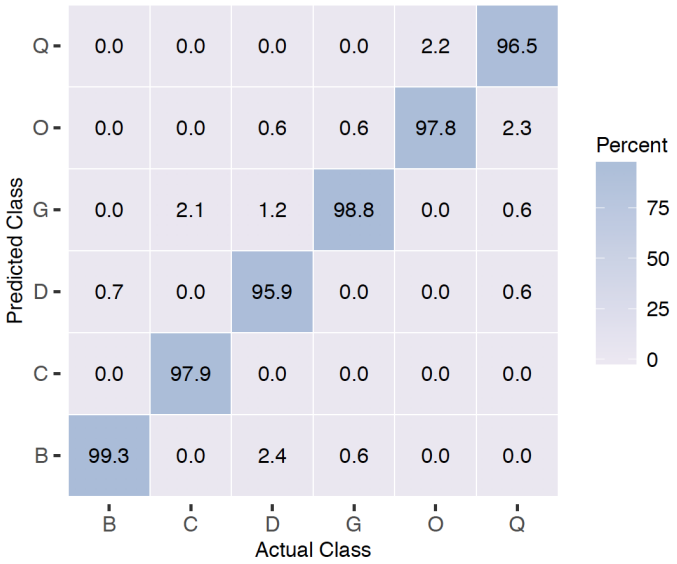


Figure 4 is the confusion matrix of the test set, also showing that GBM has achieved reasonably-high accuracy on this problem. B is best separated from other letters, having a correct classification rate of 99.3%. It is noticeable that {B,D}, {C,G}, and {O,Q} are three pairs that are relatively hard to classify.

5 Model Comparison and Summary

Next, we compare the performance of the obtained GBM model with other machine learning methods. We will validate the GBM performance with logistic regression, support vector machine and random forest.

Table 2: Comparison of different methods

Method	Running time (s)	Test error rates (%)
Logistic regression	1.14	16.23
Support vector machine - RBF kernel	1.2	1.52
Random forest	69.56	2.63
Gradient boosting, trees	267.98	2.38

Among all the methods, logistic regression has the highest error rate on test set. It is reasonable because logistic regression doesn't attempt to maximize classification accuracy, but to find coefficients that minimize the cross-entropy cost.

Among the other three methods, error rate is lowest for SVM (with Radial Basis kernel), intermediate for GBM and highest for RF. RF also learns ensembles of trees, but it trains multiple trees in parallel, taking it less time than GBM.

Since all predictors are numeric with no missing values, SVM performs pretty well. But when the inputs are mixtures of quantitative, binary and categorical, or many missing values exist in the data set, SVM may not

have such a good performance.

We can see that GBM takes the longest time to run. GBM is not the best for this particular problem, however, it tends to keep the good performance in more complex data sets. GBM is robust to outliers and missing data, having the ability to model interactions. It takes advantages of traditional boosting, tree-based models, as well as bagging. Although it may take more time to fit GBM, we can tune parameters to make a balance between predictive performance and computational cost.

References

[1] Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics* 29(5): 1189-1232.

[2] Natekin, A., & Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, 7.

[3] Friedman, J., Hastie, T., & Tibshirani, R. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd Edition). *Springer Series in Statistics*.

[4] Gradient boosting. (2016, December 8). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:09, December 8, 2016, from https://en.wikipedia.org/w/index.php?title=Gradient_boosting&oldid=753674492

[5] Lee, S., Lin, S., & Antonio, K. (2015). Delta Boosting Machine and its Application in Actuarial Modeling.

[6] Frey, P. W., & Slate, D. J. (1991). Letter recognition using Holland-style adaptive classifiers. *Machine learning*, 6(2), 161-182.

[7] Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4), 367-378.

[8] Ridgeway, G. (2007). Generalized Boosted Models: A guide to the gbm package. *Update*, 1(1), 2007.

[9] Ogutu, J. O., Piepho, H. P., & Schulz-Streeck, T. (2011, May). A comparison of random forests, boosting and support vector machines for genomic selection. In *BMC proceedings* (Vol. 5, No. 3, p. 1). BioMed Central.