Report of Random Forest

Yuliang Li
GTID# 903012703

This project completes a random forest learner for data regression. In this project, 60% of given data are used as training data, while 40% of those are used as test data. In my learner, I accomplished “bagging”, namely randomly choose 60% of training data are used to build each tree.

To evaluate the random forest learner’s effect, we compare random forest learner with KNN learner. The index of evaluation are the RMS error and correlation coefficient between results of learning and “correct” data.

We test two data sets for learners. One is data-classification-prob.csv (We called Classification data in this report), and another one is data-ripple-classification-prob (Ripple data).

Fig.1 shows the RMS Error between the result of learning and “correct” data.

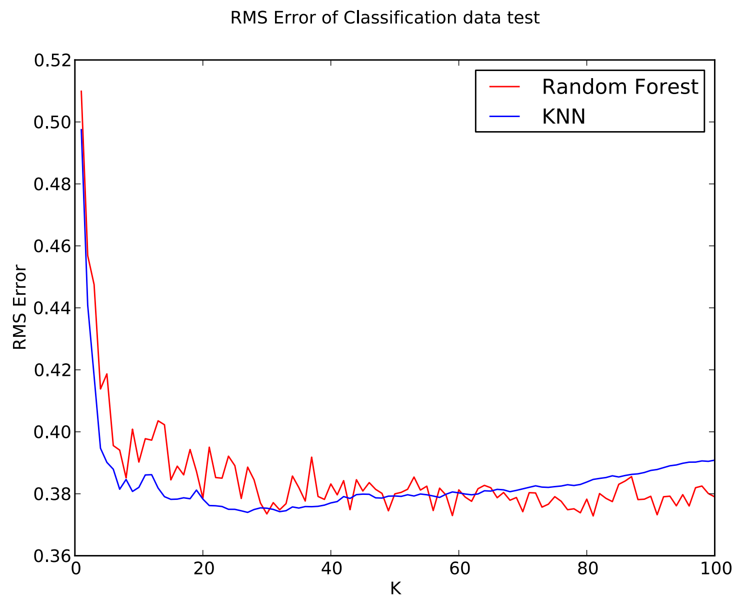

Fig. 1.1 RMS Error of testing Classification data with Random Forest learner and KNN learner

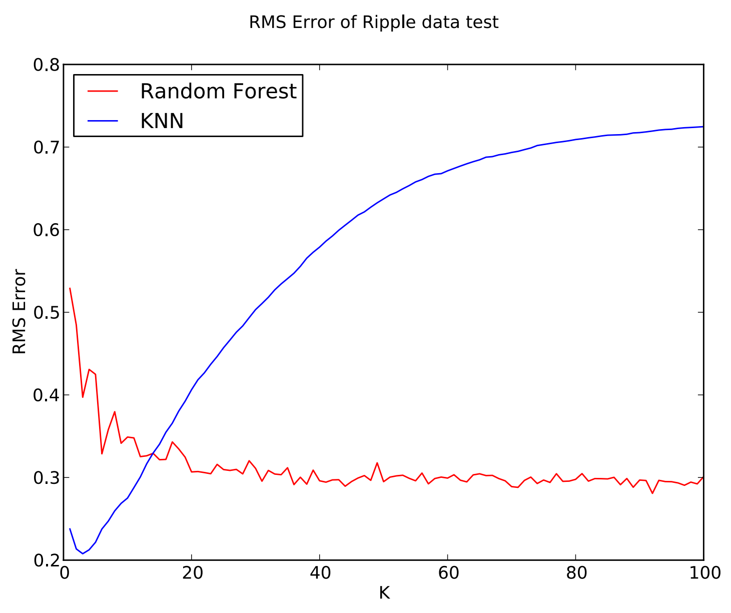

Fig. 1.2 RMS Error of testing Ripple data with Random Forest learner and KNN learner

Fig.2 Shows the Correlation Coefficient of two data sets' learning result and "correct" data.

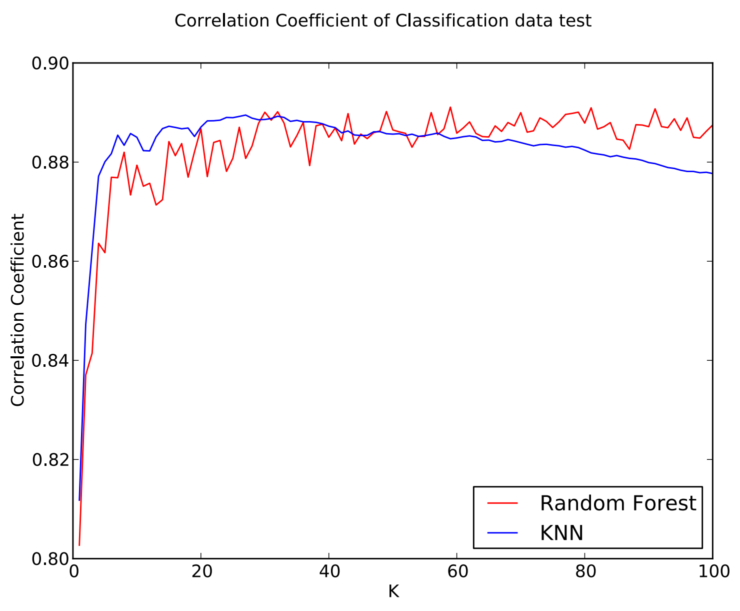

Fig. 2.1 Correlation Coefficient of testing Classification data with Random Forest learner and KNN learner

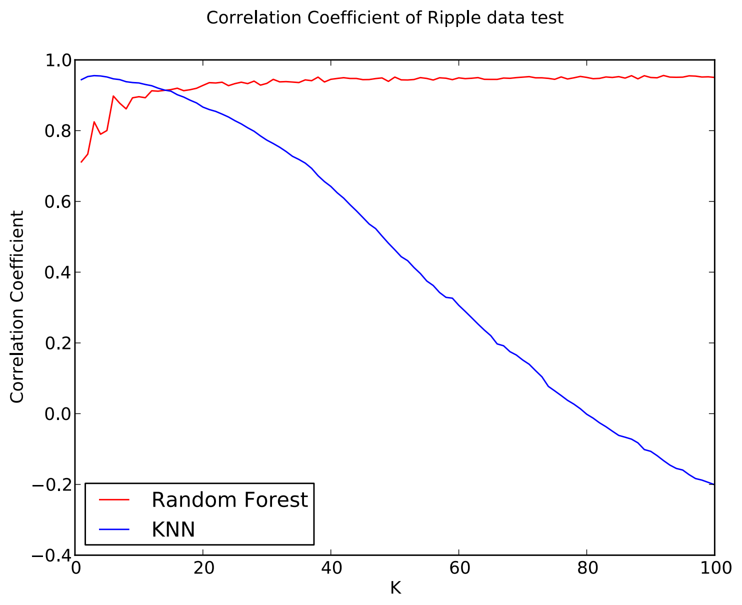

Fig. 2.2 Correlation Coefficient of testing Ripple data with Random Forest learner and KNN learner

For both data sets, the RMS Error decreases and correlation coefficient increases while K is increasing until K reaches a certain value. For Classification data, when K reaches 20, the RMS Error is hardly decreases, and the correlation error is also hardly increases. For Ripple data, when K reaches 22, both indexes are hardly improves.

For Classification data, when the number of trees (the K) in forest increases after 15, the RMS Error and correlation coefficient improves not smoothly as those of Ripple data. RMS Error and correlation coefficient fluctuate heavily for Classification data.

Thus, we can conclude that Random Forest learner is better for Ripple data sets than Classification data. That's because the Y values of Classification data are discrete, which only contains -1, 1 and 0. It's hardly can get the accurate discrete value for we average the training values of Y when we construct the trees. But Ripple data are continuous value, so its result of learning are better.

Comparing with KNN learner, each data sets' performances are different. For Classification data sets, when K is small, Random Forest learner has no advantages over KNN learner. When K exceeds 50, the former learner becomes better than KNN learner, because KNN learner is a fit learner for process discrete data set. However, for Ripple data set, when K exceeds 18, Random Forest learner has large advantages over KNN learner. With K's increasing, the RMS Error of Random Forest learner's results decreases while that of KNN learner's results increases. Thus, for continuous data set like Ripple data, Random Forest learner is much more fit operating it.

Compared between Random Forest learner, KNN learner and Linear Regression, Linear Regression has the worst performance on two data sets. Based the discussion above, without consideration on over-fitting, Random Forest learner is better to process continuous data sets while KNN learner is better to process discrete data sets. Random Forest learner has no over-fitting, but KNN learner costs less time than Random Forest learner. To save time of Random Forest learner, we can use “bagging” to reduce the size of trees.