Weekly Report
04/04/2015

Jingyu Deng

This Week

- Improved our model to adapt to the case when n_estimators is really small.
- Implemented a new model combining random forest with svm.
- Tested random forest model, SVM model, combination model, and Wang's model on data sets.

Improvement

- When n_estimator is 1, our algorithm doesn't execute normalization on weight vector. Otherwise the only weight is 1.0 forever and the only tree has no possibility to be replaced.

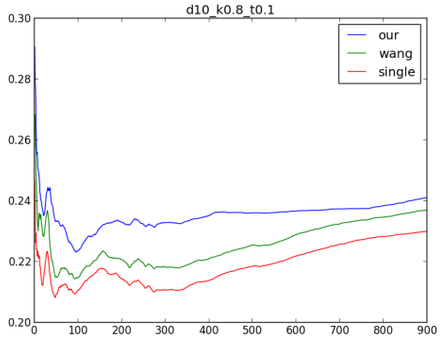
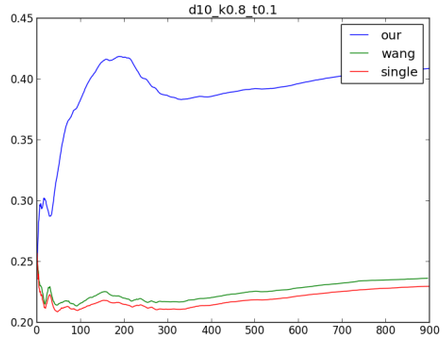
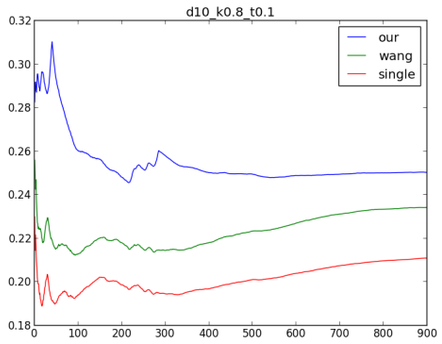
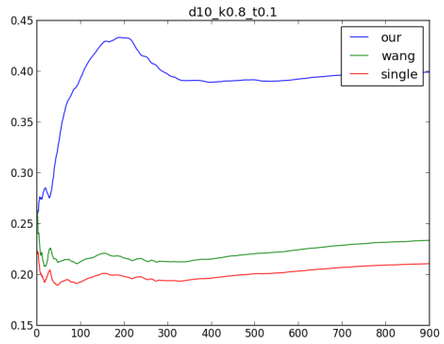
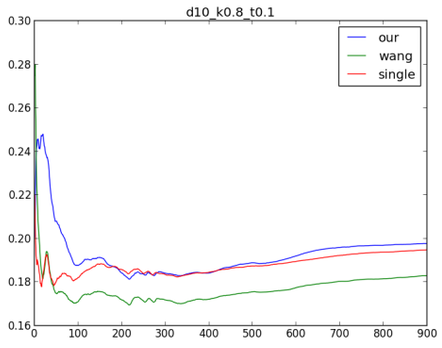
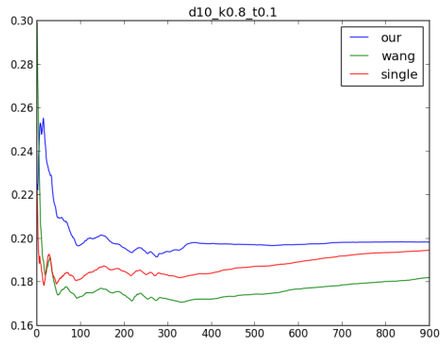
- When n_estimators is larger than 1, the threshold is a function about n_estimators:

    - threshold = 1.0 / n_trees * (0.3 / (n_trees * 2 - 3) + 0.4

- Thus in the case of small n_estimators, trees are still have chance to be replaced.

Comparison with former version

- Here are some plots of now model and former model. The chunk_size is 1000 and k is 0.8

	current	former
n=1		
n=2		
n=4		

Combination model

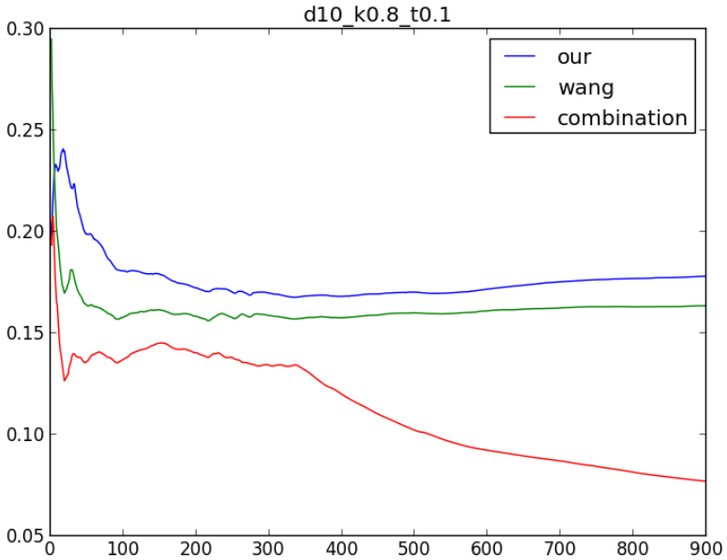
- As professor Shasha suggested, I added Lefteris' SVM into the model. So when inserting with new estimator, there is 1/4 possibility to insert with a SVM instead of a decision tree. The prediction are made by decision and SVM together. Our program considers results from both estimators.

- I tried to maintain a model with decision trees only and a model with SVM only at the same time. In 1/4 of the time we used SVM model. But maintaining a model with SVM costs really long time. So I choose the method above.

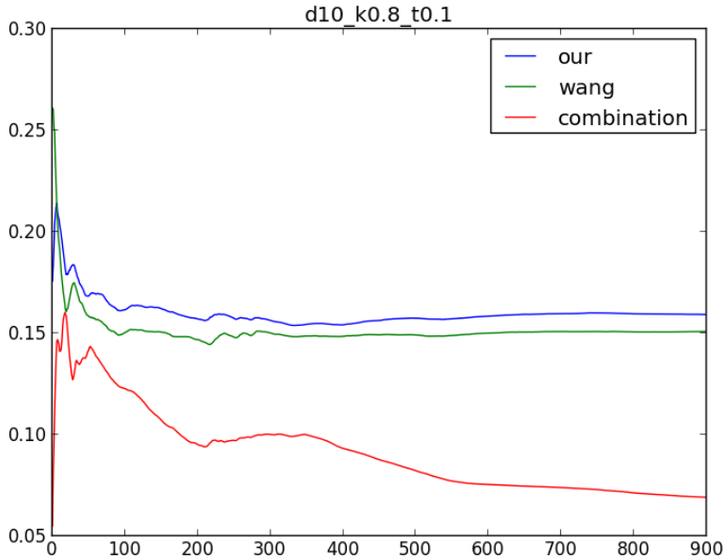
Some experimental results

- I picked some results with parameter k=0.8 and chunk_size=1000. In this data set, it is more difficult to predict and Wang's algorithm won't be influenced by the chunk_size. From these results, we know that a combination model is much better than others.

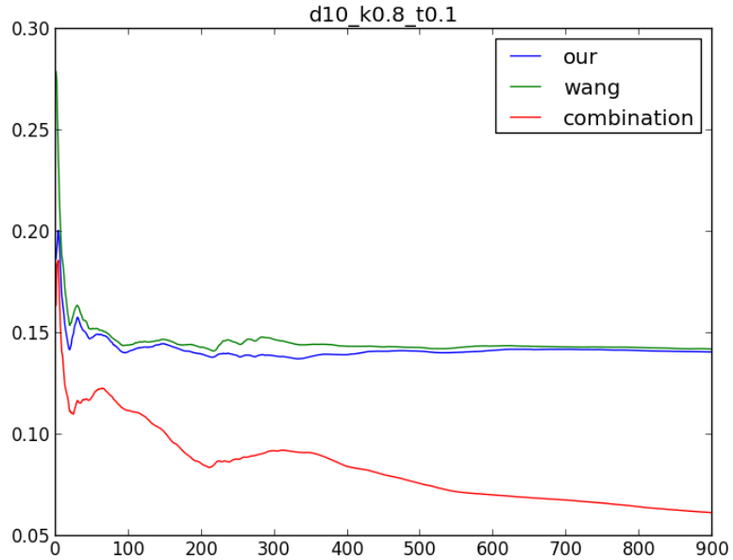
n=6



n=8



n=10



Execution Time

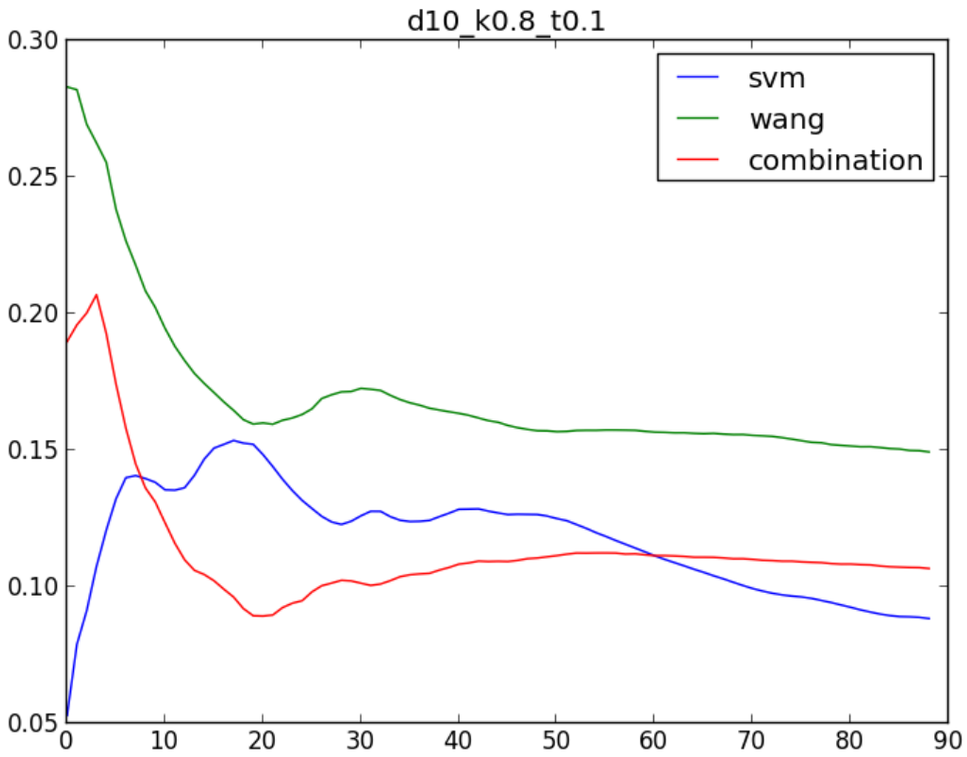
- I recorded execution time during the tests. Following table shows consumed time for the tests in the previous slide.

	n=6	n=8	n=10
decision tree	670s	909s	1094s
wang	596s	827s	976s
combination	1635s	2290s	2952s

- From this table, we know that combination model is slower than other two. And this gap will be larger when n increases.

Pure SVM Model

- I tried a model containing SVM only before but it took a really long time even when compare it with combination model. Here is the performance of SVM only model when k=0.8, n=8, chunk_size=1000



Time(estimated)	
SVM	3987s
Wang	917s
Combination	2365s

- As Lefteris mentioned, grid search procedure takes a lot of time when inserting a svm. As n_estimators increases, it will take much much more time to finish.