Weekly Report
04/27/2015

Jingyu Deng

This Week

- Modified package to support parallelization.

- Finished some experiments

New API

- When declaring a new ensemble, we can set n_jobs to make our program run in parallel.

- n_jobs is the number of jobs to run in parallel when building new regressors.

Experiment

- Data set: Anil's challenging data set
- Environment: NYU HPC
- Different parameters tested:
  - n_jobs: 1, 8
  - chunk_size: 200, 500
  - n_trees: 200, 500, 1000
  - regressor category: random tree, SVM, combination

Speed-up

- Because running all data points on a node with single processor is really slow, I tested 800 data points when n_jobs=1. When 8 processors are used, I tested 8000 data points.

Speed-up

chunk_size	n_trees	regressor	n_jobs=1, time (800)	n_jobs=1, time(8000)	n_jobs=8, time(8000)	speedup
200	200	RF	36.4775938988	364.775938988	334.794417858	1.0895520341164
200	200	SVM	82.7814319134	827.814319134	368.938416004	2.24377371188428
200	200	COM	37.580878973	375.80878973	311.685477018	1.20573083265056
200	500	RF	86.4029800892	864.029800892	727.069844007	1.18837248995254
200	500	SVM	202.545316935	2025.45316935	951.409502029	2.12889735180326
200	500	COM	94.6864309311	946.864309311	763.640388966	1.23993482140605
200	1000	RF	168.116439104	1681.16439104	1387.26568508	1.21185466426574
200	1000	SVM	407.930876017	4079.30876017	1880.50671101	2.16926041065764
200	1000	COM	189.790000916	1897.90000916	1858.45618296	1.02122397426512
500	200	RF	40.3172161579	403.172161579	305.975580931	1.31766123411632
500	200	SVM	154.45838809	1544.5838809	365.16496706	4.22982492908804
500	200	COM	49.157173872	491.57173872	311.314589977	1.5790192767911
500	500	RF	96.2175331116	962.175331116	719.766896963	1.33678741711493
500	500	SVM	385.431233168	3854.31233168	948.369683027	4.06414544946
500	500	COM	119.959914207	1199.59914207	933.54113698	1.28499869427361
500	1000	RF	193.260294914	1932.60294914	1381.92772698	1.39848337319595
500	1000	SVM	772.139837027	7721.39837027	1923.50974202	4.01422368787239
500	1000	COM	240.18727684	2401.8727684	1435.71747303	1.67294249287848

Speed-up

- From the results, we can find that multi-processors helps a lot when we use SVM as our regressor type. In addition, as chunk_size increases, the speed-up (especially for SVM) will also increase.

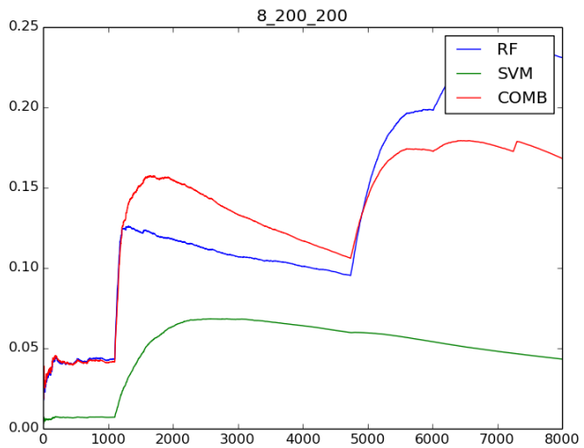
Incoming Data Rate

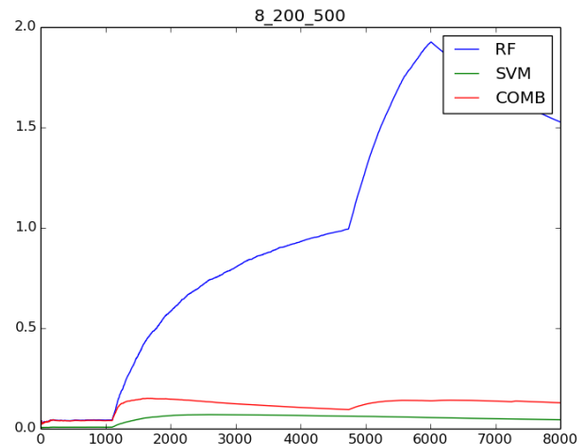
- The dimension of this data set is 50. Our incoming data rate (how many data points can be processed per second) is shown below when we use 8 processors:

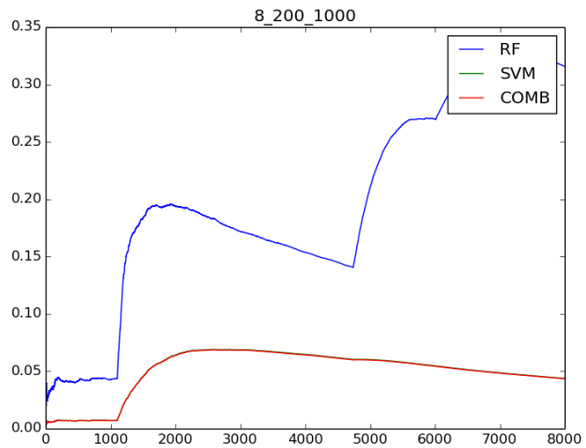
chunk_size	n_trees	regressor	n_jobs=8, time(8000)	incoming data rate
200	200	RF	334.794417858	23.8952610117685
200	200	SVM	368.938416004	21.6838357107091
200	200	COM	311.685477018	25.6669000960157
200	500	RF	727.069844007	11.003069465666
200	500	SVM	951.409502029	8.40857694077997
200	500	COM	763.640388966	10.4761352537054
200	1000	RF	1387.26568508	5.76673962748431
200	1000	SVM	1880.50671101	4.25417253400988
200	1000	COM	1858.45618296	4.30464816623131
500	200	RF	305.975580931	26.1458773136673
500	200	SVM	365.16496706	21.9079066220652
500	200	COM	311.314589977	25.6974785556663
500	500	RF	719.766896963	11.1147095452088
500	500	SVM	948.369683027	8.43552903807053
500	500	COM	933.54113698	8.56952059539653
500	1000	RF	1381.92772698	5.78901475367516
500	1000	SVM	1923.50974202	4.15906393673821
500	1000	COM	1435.71747303	5.57212693324436

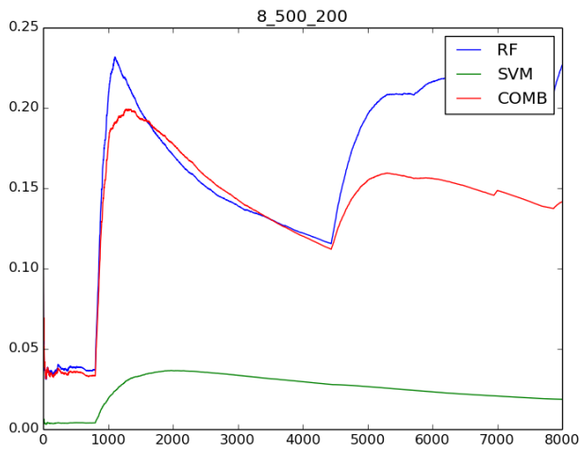
Performance of Prediction

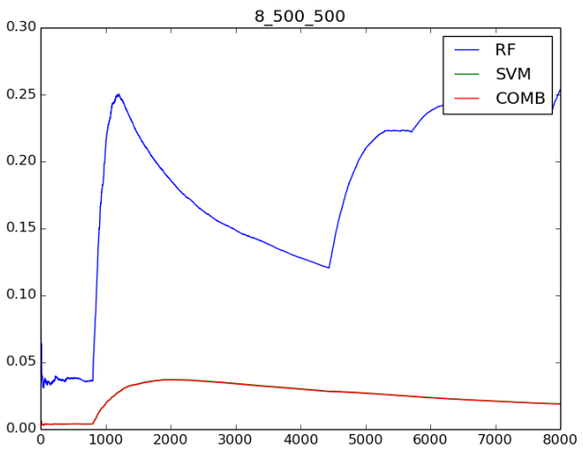
- All plots are listed here. Title 8_200_200 means that this is the plot for n_jobs=8, chunk_size=200, n_trees=200.

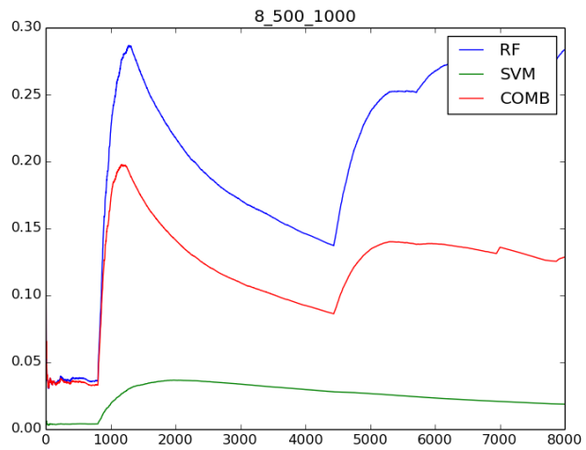


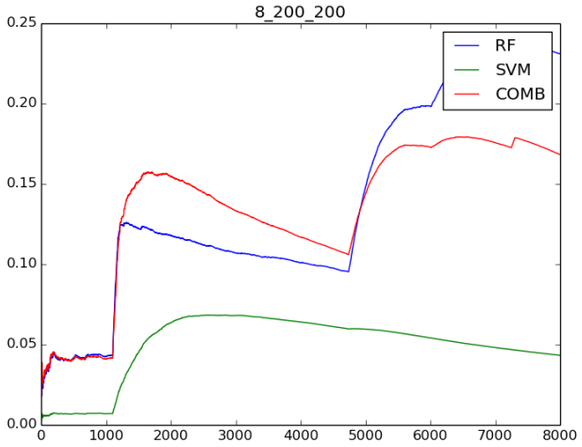


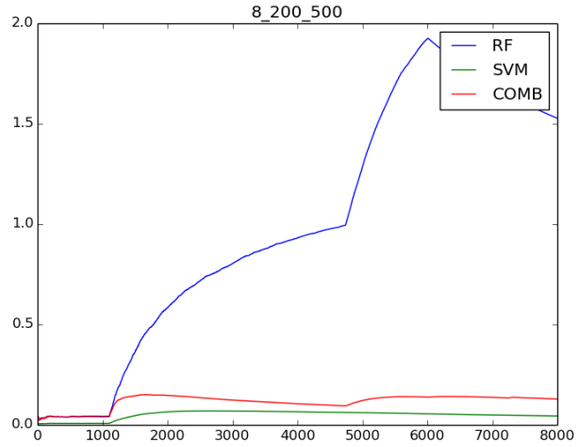


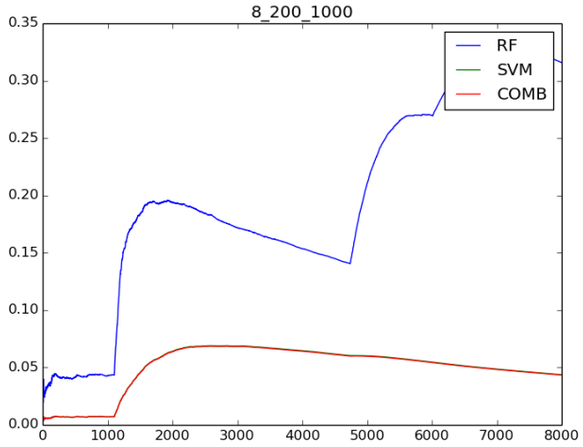


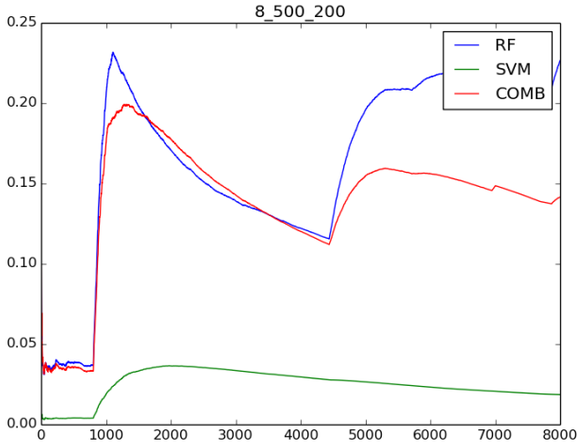


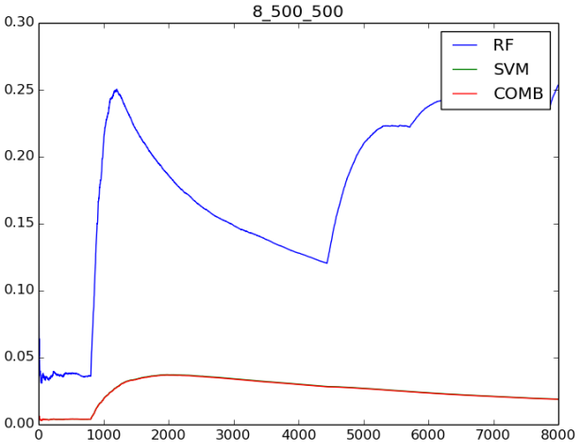


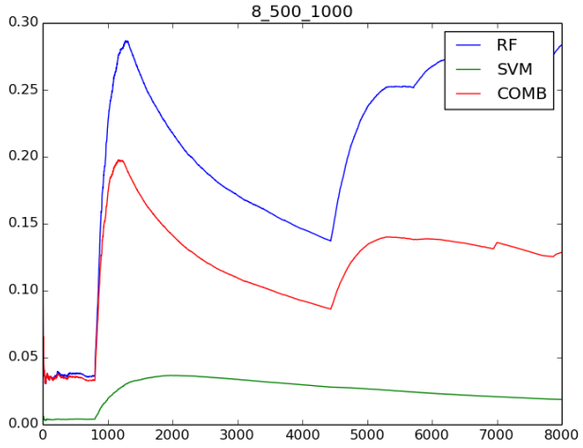




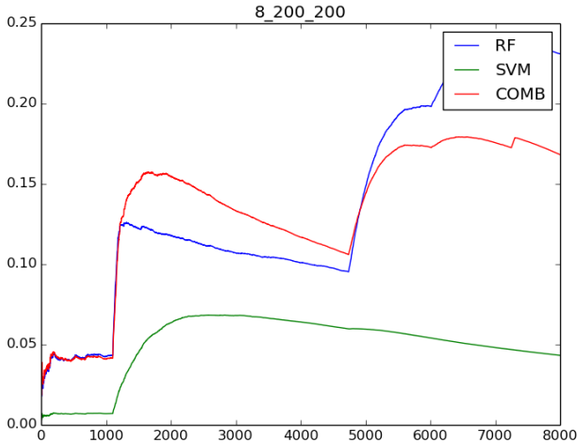


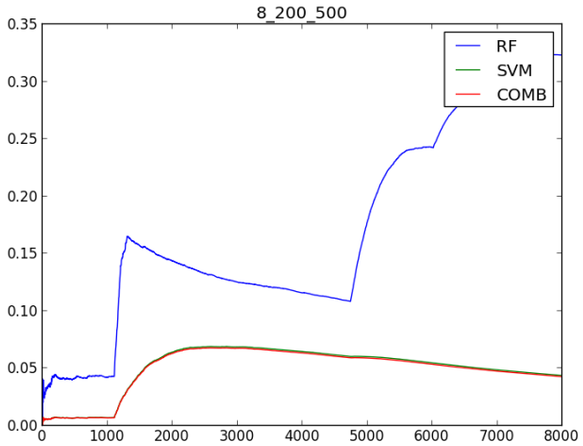


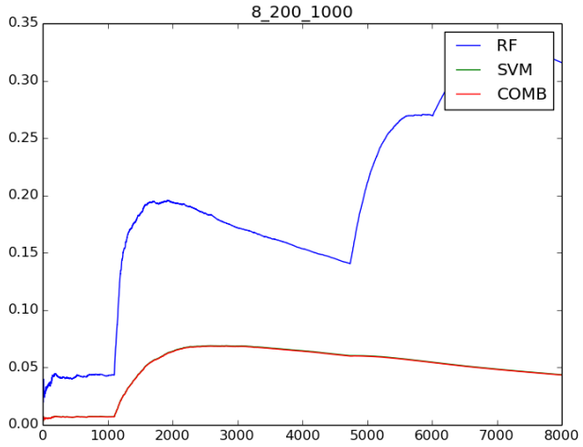


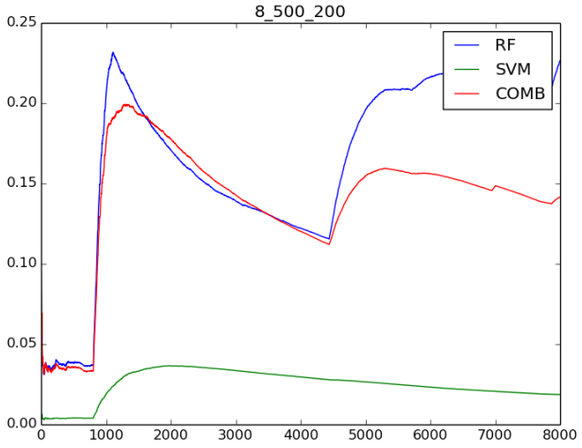


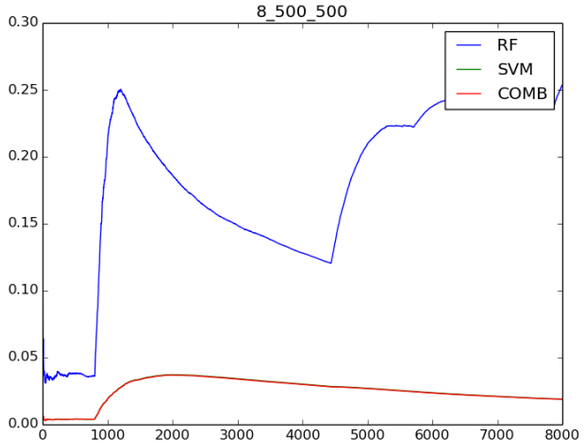
- We can see that Anil's algorithm does adjust weight to adapt to new environment while concept drift occurs.

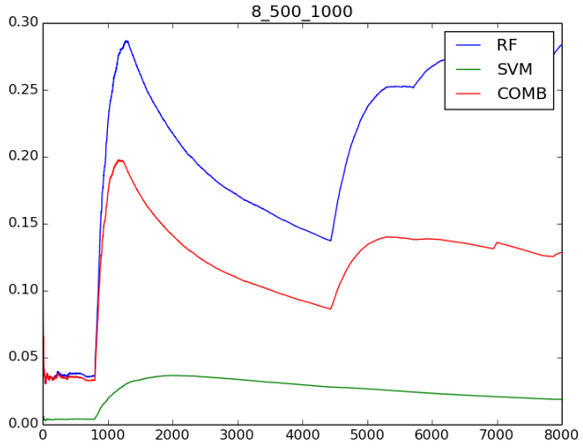












- N_trees and Chunk_size doesn't make influence on the performance of prediction.

- In some case (8_200_500), pure random_forest ensemble may fail to adapt. I wanted to make sure that it did happened so I tested this case again and again. But I got normal plot instead of that weird plot… Maybe it was an accident.