Project Milestone 3 – Tuning the parameter

Xue Yang, Wei WAN

1. Tuning the Random Forest classifier learner.

1.1. Classifier background

Random forests[1] operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes output by individual trees. It is a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest.

1.2. Parameters detail

Table 1 shows the parameters related to algorithm in random forest classifier and its meaning.

-I <number of trees>	Number of trees to build
-K <number of features>	Number of features to consider
-depth <num>	The maximum depth of the trees, 0 for unlimited.(default 0)

Table 1 Random Forest Parameters

1.2.1. Number of trees

As the number of trees in the forest becomes large, there is a limitation of generalization errors that is why random forests do not overfit as more trees are added. Meanwhile, more trees means the longer learning time, in that case. Table 2 shows our results of tuning number of trees which definitely support the analysis. We selected the points based on the size of dataset. Since both the learning time and error ratio should be taken in to consideration. We finally chose 20 as our parameter.

1.2.2. Number of features

As reported by the paper, number of features selected to split each node has less influence on results, especially in small dataset. Our experiments confirm this argument. Based on int(logM+1), We try the sequence as in table 3. It can be seen, learning time slightly increases as the number of features goes up and maximum and

[1] Breiman, Leo. "Random forests." *Machine learning* 45.1 (2001): 5-32.

average fluctuates randomly. In that case,

# of trees	Maximum Error Ratio	Average Error Ratio	Average Learning Time
1	1.685185	1.125676	23.67
5	1.477778	0.8747	74.00
10	1.466667	0.813265	141.25
10	1.422222	0.809792	149.33
20	**1.422222**	**0.779088**	**275.08**
30	1.411111	0.774429	402.83
40	1.431481	0.761966	546.25
50	1.401852	0.753692	672.42
100	1.416667	0.748531	1355.67
150	1.401852	0.74943	2021.08
200	1.435185	0.752242	2726.92
250	1.392593	0.7539	3435.42
300	1.409259	0.745816	4141.92

Table 2 Results of tuning number of trees

# of features	Maximum Error Ratio	Average Error Ratio	Average Learning Time
0	**1.412963**	**0.779901**	**288.42**
1	1.293750	0.819873	256.92
2	1.337037	0.790721	257.00
3	1.403704	0.796162	260.33
4	1.448148	0.784325	277.83
5	1.435185	0.783654	305.17
6	1.470370	0.786886	311.58
7	1.487037	0.795928	321.08
8	1.437037	0.806102	327.83
9	1.446296	0.803592	333.92

Table 3 Results of tuning number of features

default value 0 (automatically tuning) is using in our final parameter set.

1.2.3. Maximum depth

The maximum depth of the trees is the limitation of depth in each tree. It can be seen from our results, as the maximum depth limitation increases, the average learning time increase quickly, at the same time, Average error ratio decreases and converges to about 0.8. Noted that default value 0 (automatically tuning) did really good job in balance the learning time with error ratio. So default value is used in our final setting.

1.3. Final setting

Maximum Depth	Maximum Error Ratio	Average Error Ratio	Average Learning Time
0	**1.412963**	**0.779901**	**311.08**
1	3.080435	1.437221	55.83
2	2.269565	1.181159	90.25
3	1.446296	1.004635	129.75
4	1.312500	0.890042	167.92
5	1.305769	0.833910	208.25
6	1.294231	0.797854	235.67
7	1.320370	0.774912	259.67
8	1.370370	0.790165	276.33
9	1.425926	0.785428	279.83

Table 4 Results of tuning maximum depth

After tuning the parameter one by one, # of trees is 20 and default value 0 is used for maximum depth and # of features. Under the optimal parameter setting, the results are shown as follows,

	Maximum Error Ratio	Average Error Ratio	Average Learning Time
Cross Validation	1.412963	0.779901	274.58
Test Set	1.137931	0.762843	28.00

Table 5 Test of Random Forest

2. Tuning the LogitBoost classifier learner.

2.1. Classifier background

The LogitBoost algorithm is a variation of boosting procedure. Instead of squared error loss, LogitBoost directly optimizes the binomial log-likelihood. For multiclass situation, it uses quasi-Newton steps for fitting an additive symmetric logistic model by maximum-likelihood.

2.2. Parameters detail

Table 6 shows the parameters related to algorithm in LogitBoost classifier and its meaning.

-Q	Use resampling instead of reweighting for boosting.
-P <percent>	Percentage of weight mass to base training on. (default 100, reduce to around 90 speed up)
-L <num>	Threshold on the improvement of the likelihood.

	(default -Double.MAX_VALUE)
-H <num>	Shrinkage parameter. (default 1)
-I <num>	Number of iterations. (default 10)

Table 6 LogitBoost Parameters

Note that, LogitbBoost supports internal cross validation and can tune number of iterations automatically. However, cross validation is time consuming. As the result, we decide to tune all the parameters manually.

2.2.1. Threshold on improvement of the likelihood

Threshold on improvement of likelihood seems has no influence on algorithm. Default value is simply selected.

Number of Iterations	Maximum Error Ratio	Average Error Ratio	Average Learning Time
-1	1.46875	0.737901	315.83
-1E+20	1.30625	0.712141	309.33
-1E+40	1.3875	0.724926	309.42
-1E+60	1.375	0.721295	308.58
-1E+80	1.4375	0.730277	308.08
-1E+100	1.35	0.724486	308.33
-1E+120	1.45625	0.72679	309.33
-1E+140	1.30625	0.715063	307.92
-1E+160	1.39375	0.723	309.00
-1E+180	1.375	0.726806	308.83
-1E+200	1.325	0.724896	309.25
-1E+220	1.35	0.722471	307.08
-1E+240	1.45625	0.741115	309.25
-1E+260	1.4125	0.731298	308.17
-1E+280	1.302702703	0.715205	309.42
-1E+300	1.3125	0.710809	308.25

Table 10 Results of tuning likelihood threshold

2.2.2. Percentage

Percentage of weight mass to base training on is designed to reduce the computation time. This indicated the percent of training sample should remain in data set after deleting the large fraction of observations with very low weight. According to original paper, 0.9 is a normal setting to speed up. In that case, we tune this parameter from 0.8 to 1.0 and obtain the results in table 8.

Percentage	Maximum Error Ratio	Average Error Ratio	Average Learning Time
81	1.58750	0.786574	182.58
82	1.43750	0.764716	179.92
83	1.38125	0.750068	181.75
84	1.50625	0.765335	183.92
85	1.53125	0.762911	186.00
86	1.37500	0.750493	189.83
87	1.46875	0.756396	193.25
88	1.43125	0.752737	196.67
89	1.47500	0.761991	198.83
90	1.36250	0.74605	202.67
91	1.48125	0.760728	205.25
92	1.52500	0.761143	209.50
93	1.48125	0.759648	213.25
94	1.38750	0.742594	216.42
95	**1.36250**	**0.739478**	**226.92**
96	1.35000	0.741665	233.08
97	1.41875	0.753390	238.83
98	1.46875	0.757169	245.25
99	1.43125	0.752955	255.67
100	1.46250	0.751999	268.58

Table 8 Results of tuning percentage

It can be seen from table 7, as the percentage goes up, it takes more time to learn, meanwhile, classifier are more robust as more data are considered. We chose 95 as a balance of time and error ratio.

2.2.3. Number of iterations

Number of iterations is the number of turns we use data. As the iterations increase, average error ratio may converge to a relatively low level. However, learning time is proportional to number of iterations. Our experiments fully support this argument. Finally 15 is using as our setting.

Number of Iterations	Maximum Error Ratio	Average Error Ratio	Average Learning Time
1	3.04074	1.287273	40.08
3	1.98889	0.882737	88.00
5	1.57500	0.835058	135.83
7	1.50625	0.782189	179.58
10	1.40000	0.745385	230.67
11	1.50625	0.758045	254.33
12	1.49375	0.748701	264.83

13	1.27500	0.722099	278.67
14	1.43125	0.738344	293.75
15	**1.39375**	**0.717257**	**305.58**
20	1.34375	0.714549	372.58
30	1.33750	0.711743	497.50
50	1.43784	0.722444	713.75
80	1.40811	0.711927	1021.17

Table 9 Results of tuning iterations

2.2.4. Shrinkage parameter

Shrinkage is a parameter whose value controls the amount of shrinkage in update which computed in the usual manner by the boosting algorithm. It is an important ingredient to the success of boosting in the regression context. As reported by the paper, such shrinkage dramatically improves the accuracy of the target function estimate when measured by likelihood, squared error or absolute error.

Shrinkage	Maximum Error Ratio	Average Error Ratio	Average Learning Time
0.01	2.49815	1.21318	418.25
0.05	2.25741	1.09011	413.83
0.1	1.83889	0.91623	409.58
0.15	1.49444	0.83174	408.42
0.2	1.39375	0.77618	400.83
0.25	1.40625	0.75910	396.33
0.3	1.36875	0.74855	388.42
0.35	1.41250	0.73379	384.08
0.4	1.35625	0.72429	379.00
0.45	1.33750	0.70868	366.92
0.5	1.39375	0.71034	359.17
0.55	1.39375	0.71032	353.50
0.6	1.43750	0.70788	345.58
0.65	1.35625	0.69920	340.00
0.7	1.31875	0.70747	334.75
0.75	1.31250	0.71069	325.17
0.8	1.31250	0.71070	320.67
0.85	1.37500	0.71492	314.67
0.9	1.40000	0.72501	307.75
0.95	1.35000	0.72090	305.58

Table 7 Results of tuning shrinkage

Based on our experiments, 0.45~0.7 are similar. But shrinkage should be couple with number of iterations. In that case, we varied shrinkage from 0.45~0.7 and

enlarge iteration number to make it converges. Finally, the best setting pair for shrinkage and number of iterations is 0.5 and 40.

2.2.5. Resampling or reweighting

Resampling or reweighting switch designs for deciding which approach is used for update. Based on our experiments, it seems to be no influence on algorithm. Default value is simply selected.

2.3. Final setting

After tuning the parameter one by one, we set shrinkage to 0.5, percentage as 96 and number of iterations to 40. Under the optimal parameter setting, the results are shown as follows,

	Maximum Error Ratio	Average Error Ratio	Average Learning Time
Cross Validation	1.26250	0.687233	764.58
Test Set	1.00000	0.666878	82.00

Table 8 Test of Logitboost

3. Tuning the AODEsr classifier learner.

3.1. Classifier background

AODEsr^2 (Weka class name), whose normal name is Averaged One-Dependence Estimator with Subsumption Resolution, is a semi-naïve Bayesian algorithm. It reduces the error by relaxing the attribute independence assumption. And the way it relaxes the attribute independence assumption is by deleting the generalization attributes.

3.2. Parameters detail

Table 6 shows the parameters related to algorithm in AODEsr classifier and its meaning.

-F	Impose a frequency limit for superParents
-M	Weight value for m-estimation
-C	Impose a critical value for specialization-generalization relationship
-L	Using Laplace estimation

Table 9 AODEsr Parameters

F-For AODE, a single attribute is selected as the parent of all other attributes. F is the limit of frequency of the value for classified object of the parent attribute in the training data. Large F value will increase error so default is 1.

M-M-estimate is used in Naïve Bayes when calculating conditional probabilities to avoid the case where the times of an event that happens in training data is 0.

C reflects the relationship between a parent attribute and other attributes. It's the probability that an attribute is true while the parent attribute is true.

L-Whether to estimate base probabilities with Laplace estimation.

3.3. Problems

This classifier had difficulties in dealing numerical attributes. When I tried multiclassifier class, as what we did in milestone2, I can evaluate the test set correctly. However, cross validation of multiclassifier leads to the same exception. Since we cannot using cross validation as we did in the other classifiers. We ignored this AODEsr in this milestone.

4. Summary

Among the classifiers we used, the best robustness was obtained by LogitBoost, which was naturally quite robust without parameter tuning. However, by using optimal settings robustness from 0.73 to 0.67, and improve it's maximum robustness from 1.04 to 1.00. It is even better then selected the best classifier for each dataset without parameter tuning.