Boosting many low capacity classifiers compared to boosting fewer high capacity classifiers in a context of multiclass classifications

Jonathan Guymont, Marzieh Mehdizad, Léa Ricard, Jeff Sylvestre Decary & Joseph D. Viviano

Université de Montréal and Polytechnique Montréal

Introduction

- Investigate whether there is any benefit to using Adaboost while keeping ensemble capacity fixed.
- Compares models in the same family: a single strong learner, or multiple weaker learners such that the number of trained parameters remains roughly the same.
- Hypothesize that using more learners with lower individual capacity will outperform a single learner with equivalent capacity.

Algorithms

Decision Tree. Let d be the number of features. At a particular node of the tree, $\sqrt{d}$ features were randomly selected to be tested. At each node, the data is split according to 1

$$\theta_m = \arg\max_{\theta} H(D_m) - H(D_m|\theta). \quad (1)$$

Each node is divided in two child until either the maximal depth of the tree $\mathcal{T}$ is reached or the minimum number of sample at a node $\min_{sample}$ is reached. $\mathcal{T}$ and $\min_{sample}$ are two hyperparameters that we used to control the capacity.

Logistic Regression.

$$p(y = k|\mathbf{x}) = \frac{e^{\mathbf{W}_{k,}\mathbf{x}+\mathbf{b}_k}}{\sum_{i=1}^{m} e^{\mathbf{W}_{i,}\mathbf{x}+\mathbf{b}_i}} \quad (2)$$

Multi-Layer Perceptrons.

$$\mathbf{h} = \sigma(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}), \quad \mathbf{y} = \text{softmax}(\mathbf{W}^{(2)}\mathbf{h} + \mathbf{b}^{(2)}) \quad (3)$$

We change the dimension of $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(2)}$ (i.e. the size of the hidden layer) to vary the capacity.

Datasets

We used two data sets:

- **Wine Quality**: Using 12 quality features measured from white and red wine, predict how these wines are rated by experts (5 classes).
- **Balanced Forest Cover Type**: Using cartographic variables, predict the correct forest cover type (7 classes).

Methodology

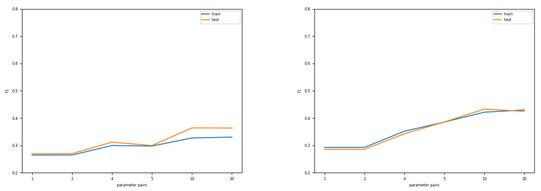
Data pre-processing: All data was normalized before training. For the wine quality data set, the extremely poor and good wines were merged (respectively) as those classes were very rare.

Hyper-parameter search: We used the training set to learn the parameters of the model and the validation set to do hyper-parameter selection using randomized search with 50 iterations, and 10 fold inner-loop cross validation.

Performance evaluation: We used macro F_1 score for both hyperparameter tuning and test-set reporting due to unbalanced classes.

Capacity variation: We decrease the capacity of decision tree by reducing the maximum depth of the tree and MLP model by reducing the number of hidden units while proportionally-increasing the number of classifiers.

Logistic Regression Results

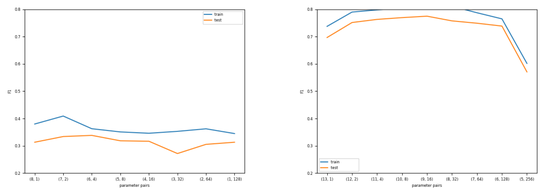


(a) (b)

Figure 2: Performance on test and training set by parameter pairs (tree maximum depth, number of tree boosted) on (a) Wine Quality data and (b) Forest Cover Type boosted with Decision Tree classifiers

- **Adding more learners increases classification performance** for both datasets in the absence of capacity reduction.
- Neither model shows evidence of overfitting.

Decision Trees Results

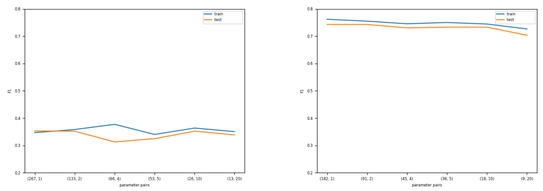


(a) (b)

Figure 1: Performance on test and training set by parameter pairs (tree maximum depth, number of tree boosted) on (a) Wine Quality data and (b) Forest Cover Type boosted with Decision Tree classifiers

- The individual capactiy / number of learner tradeoff showed **no difference between model performance in most cases**.
- **Not a precise method for controlling the number of parameters**, may explain drop in performance with very shallow trees.

Multi-Layer Perceptrons Results



(a) (b)

Figure 3: Performance on test and training set by parameter pairs (number of neurons, number of MLP boosted) on (a) Wine Quality data and (b) Forest Cover Type boosted with MLP classifiers

- When reducing the capacity of the model and proportionally adding more learners, we observe **no differences in model performance**.
- Easier to **precisely control the number of parameters** in the model via the hidden layer size.

Results Summary

Table 1: Test performance of models Wine Quality

Wine Quality (boosted)						
	Best result		Highly boosted		Not boosted	
Model	Param.	F_1	Param.	F_1	Param.	F_1
LR	(-,10)	0.36	(-,20)	0.36	(-,1)	0.26
Dec. Tree	(6,4)	0.34	(1,128)	0.32	(8,1)	0.32
MLP	(133,2)	0.35	(13,20)	0.34	(267,1)	0.35

Table 2: Test performance of models Forest Cover Type

Forest Cover Type (boosted)						
	Best result		Highly boosted		Not boosted	
Model	Param.	F_1	Param.	F_1	Param.	F_1
LR	(-,10)	0.43	(-,20)	0.42	(-,1)	0.29
Dec. Tree	(11,64)	0.53	(10,128)	0.51	(17,1)	0.41
MLP	(91,2)	0.74	(9,20)	0.70	(182,1)	0.74

Discussion and Conclusion

- Adding additional classifiers to balance out the loss of per-model capacity recovers the capacity of the stronger individual model model.
- Could repeat experiments to obtain good confidence intervals on the F_1 scores.
- Could comparing bagging and boosting methods for datasets with noisy labels (wine dataset) [1].
- Evaluating the red and white datasets separately as the features may not correspond across classes.

References

[1] Dietterich, T. (2000). An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization. Mach. Learn.. 40. 10.1023

Université de Montréal

POLYTECHNIQUE MONTRÉAL
LE GÉNIE EN PREMIÈRE CLASSE