Bagging Ensemble Selection for Regression

Quan Sun and Bernhard Pfahringer

Machine Learning Group
Computer Science Department
The University of Waikato

12/2012

A Brief Background: Ensemble Strategies for Regression

A linear form:

$$F(\mathbf{x_i}) = \sum_{j=1}^{k} w_j f_j(\mathbf{x_i}), \tag{1}$$

where w_j is the weight of base model f_j.

- Gradient Boosting
- Stochastic Gradient Boosting
- Bagging
- MultiBoosting
- Ensemble Selection (ES) from a library of models ...

Ensemble Selection from a Library of Models (ES)

ES (Caruana, ICML'04) basic procedure:

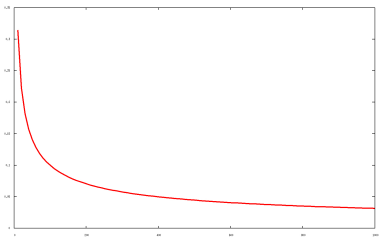
1. build models in library
2. start with an empty ensemble
3. add to the ensemble the model that maximizes the ensemble's performance using some given error metrics on a hillclimb set, then update the model's weight (i.e., weight++)
4. repeat Step 3 until all models have been examined
5. return that subset of models (with >0 weight)

In 2009, IBM Research has won the KDD Cup using ES with models produced from a diverse model library (incl. Weka, LibLinear, LibSVM models). Already won after the first 3 submissions.

(KDDCup'09: 6k+ submssions made from 500+ teams)

ES: Rate of Convergence

No theoretical work on examining the convergence property of ES, but if we see the step-wise ensemble selection procedure used in ES as a "greedy" feature selection algorithm, and see the predictions of each base regression model as the 'features'.



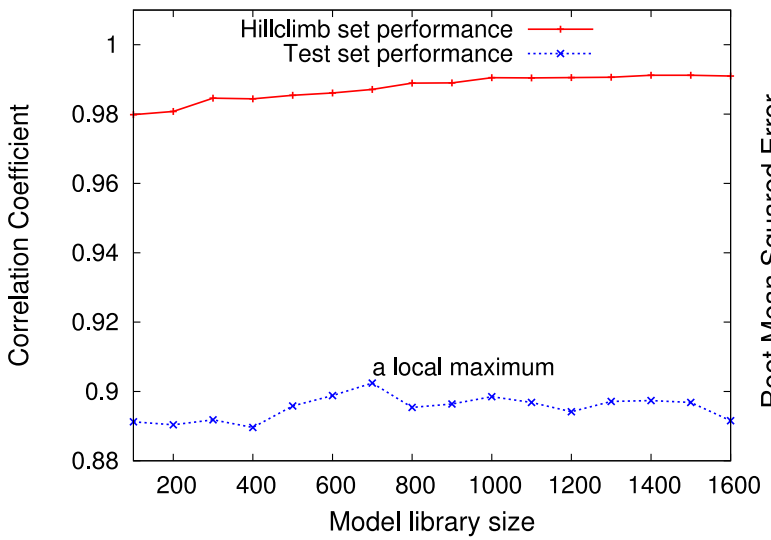
The theoretical convergence rate is sublinear at $m^{-\frac{1}{2}}$, m is the number of base models, based on results in (Bühlmann and Geer, 2011)

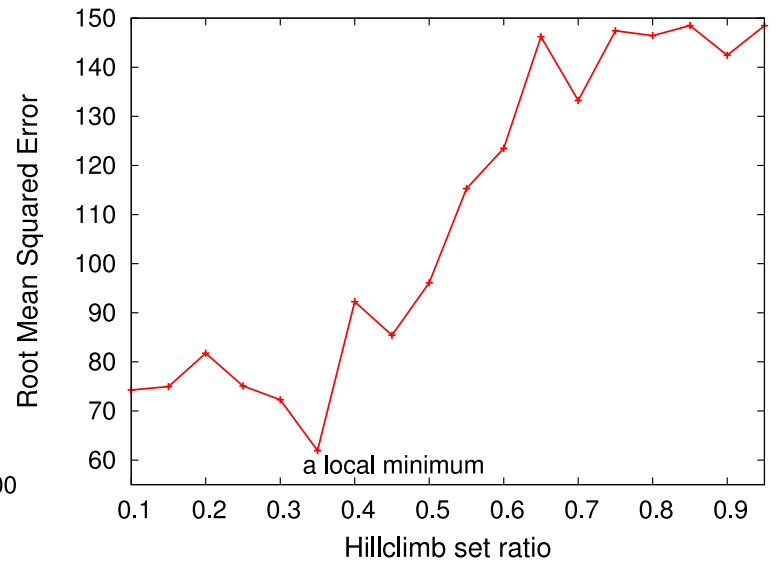
[a direction for future work]

Motivations: Improving ES for Regression

Two problems:

- when using ES for regression problems, the algorithm sometimes overfits the hillclimb set
- user has to find the optimal hillclimb set ratio





Left: Boston housing-price data **Right**: CPU performance data

Bagging Ensemble Selection with the Out-of-Bag Sample (BES-OOB)

To overcome these problems and improve the predictive performance of the ES strategy for regression, we propose the **BES-OOB** strategy:

- BES-OOB is an ensemble of ES ensembles, each base-level ES can be seen as an unstable base model trained on one bootstrap sample.
- In general, the out-of-bag (oob) sample is used as the validation set, in BES-OOB we use the oob sample as the "hillclimb" set for base-level ES ensemble construction.

Bagging Ensemble Selection with the Out-of-Bag Sample (BES-OOB)

BES-OOB(S, E, T)

S is the training set
E is the Ensemble Selection regressor
T is the number of bootstrap samples

1: $H \leftarrow$ empty ensemble
2: for $i \leftarrow 1$ to T {
3: $S_b \leftarrow$ bootstrap sample from S
4: $S_{oob} \leftarrow$ out of bag sample
5: train base regressors in E on S_b
6: $E_i \leftarrow$ do ensemble construction based on base regressors' performance on S_{oob}
7: add E_i to H
8: }
9: return H

Experiment 1: Comparison to Other Ensemble Strategies

Compare BES-OOB to:

- Stochastic Gradient Boosting (SGB),
- Standard Bagging (BG)
- MultiBoosting: bagged Gradient Boosting (BSGB)

42 regression data sets. 10 $\times$ 10-fold CV estimations

1,500 REPTrees (a CART-like regression tree) each strategy

[details are in Sec 3.1]

Experiment 1: Comparison to Other Ensemble Strategies

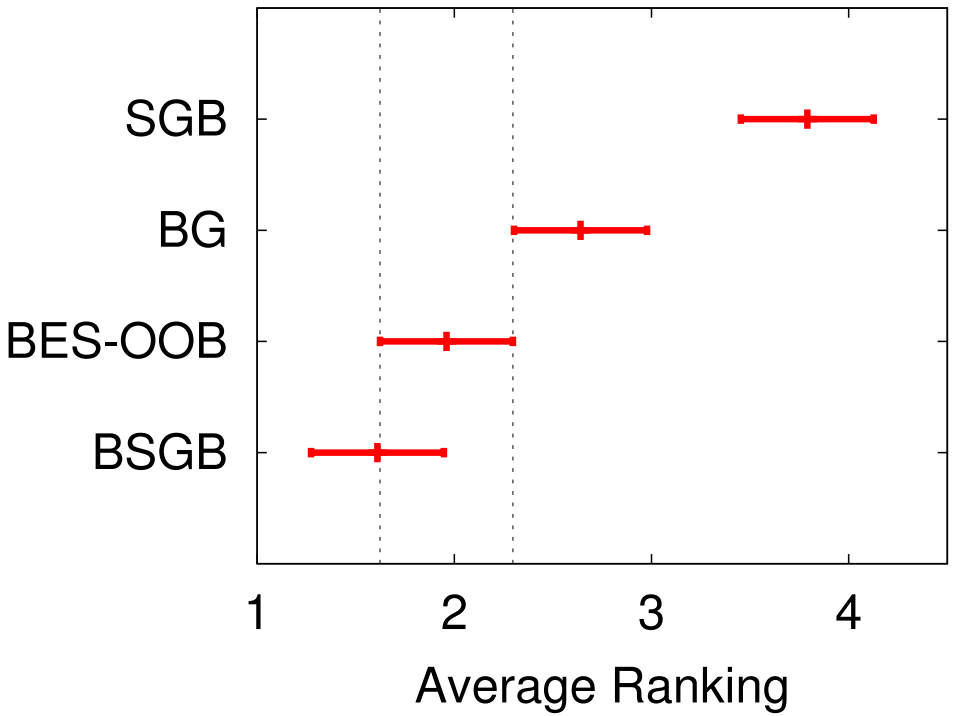


Figure: Friedman-test results for the four strategies with REPTree as base learners over 42 data sets

Experiment 2: In-Model Diverse Model Libraries

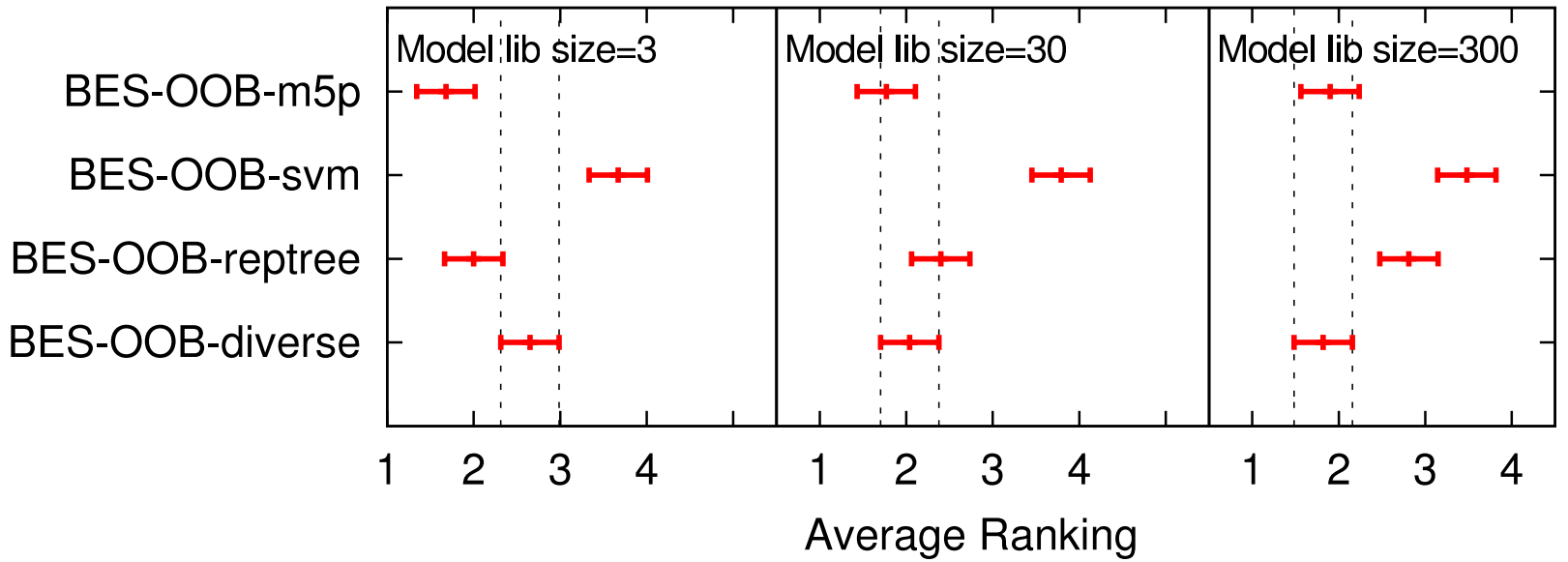
BES-OOB can use different types of base learners. In this exp, BES-OOB with a diverse model library consisting of three types of base learners (REPTree, SVM regression and M5P model trees), is compared to BES-OOB with only one of the three base learners.

Num. Bagging iteration of BES-OOB: 30

Three ES model library sizes: 3, 30 and 300 (corresponding to 90, 900, 9000 base models)

[details are in Sec. 3.2]

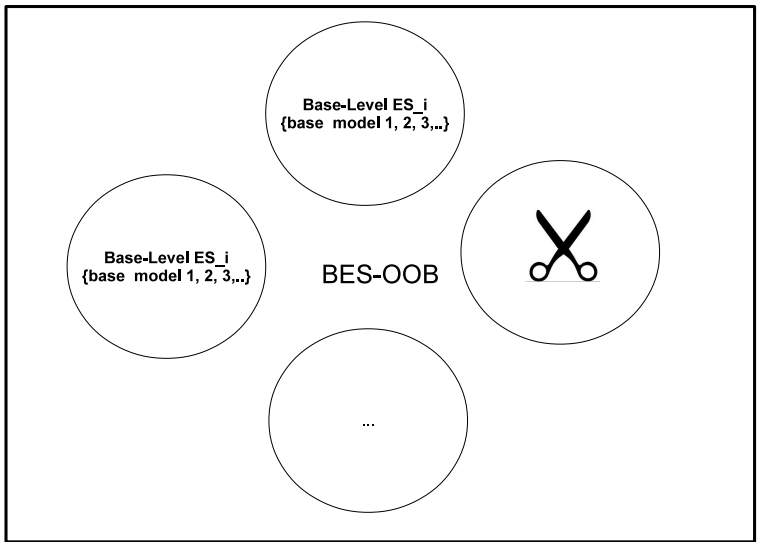
Experiment 2: In-Model Diverse Model Libraries



- BES-OOB-diverse's avg. rank improves when the model library size increases (from the 3rd to the 1st)
- the advantage of using a diverse model library becomes clear when the model library size is relatively large

Experiment 3: Pruning an Ensemble of ES Ensembles

In this experiment, we consider methods for BES-OOB pruning (Pruning an Ensemble of ES Ensembles).



Two main reasons:

- reduce prediction cost without sacrificing too much predictive power
- (ideally) a more accurate model

Experiment 3: BES-OOB Pruning

Compare simple averaging (no pruning) to two pruning methods:

1. pruning with the cocktail ensemble (CE) algorithm [Sec 3.3]
2. pruning with the stacking strategy using the non-negative least-squares (NNLS) algorithm as the meta-level learner

Experiment 3: BES-OOB Pruning

Stacking with NNLS for BES-OOB pruning: use NNLS as the meta-level learner, predictions of base-level ES ensembles are used as "features".

The final ensemble consists only of ES ensembles with greater than zero weight.

Basic form of NNLS optimisation:

$$\min_{w \geq 0} \|Xw - y\|_2^2. \tag{2}$$

Advantages of using the NNLS algorithm:

- fast even for thousands of (numeric) features
- final ensemble is usually small
- guaranteed to find the global optimal solution (for squared loss)

Experiment 3: BES-OOB Pruning

An experiment for BES-OOB-avg (no pruning) vs.

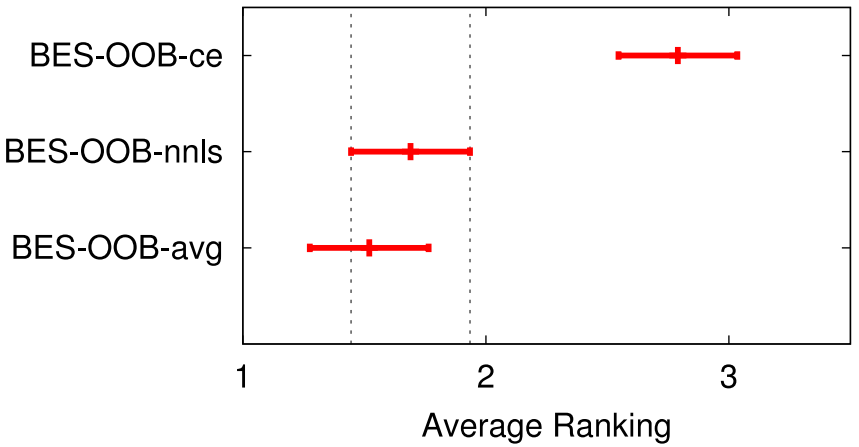
- BES-OOB-ce (cocktail ensemble based pruning)
- BES-OOB-nnls (stacking with NNLS based pruning)

Num. Bagging iterations is set to 30

10 REPTrees per base-level ES (in total 300 trees per strategy)

[Sec 3.3]

Experiment 3: BES-OOB Pruning



- Friedman-test shows BES-OOB-avg and BES-OOB-nnls has no significant differences [Fig].
- [Table 3 Sec 3.3] BES-OOB-nnls **significantly** outperforms BES-OOB-avg on 5 data sets indicating that BES-OOB-nnls could be used for further improving the predictive performance of BES-OOB

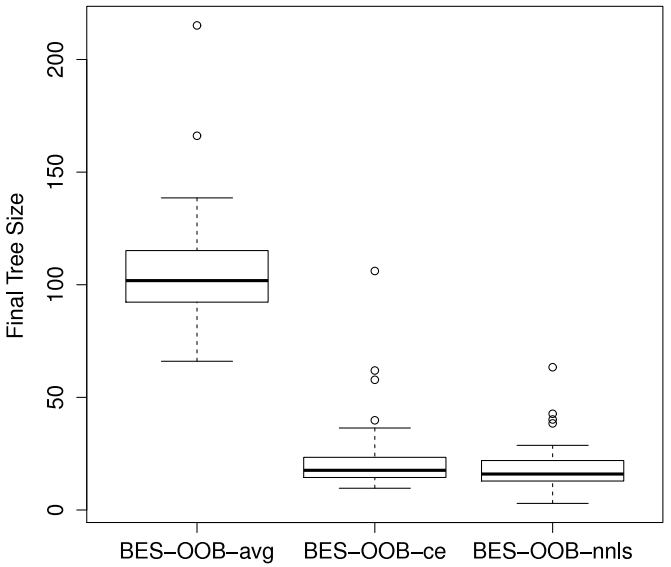
Experiment 3: BES-OOB Pruning

Final Ensemble Size of BES-OOB

Strategy	**Avg. final ensemble size**	**Avg. Reduction**
BES-OOB-avg (no pruning)	30	-
BES-OOB-ce	6.2	70%
BES-OOB-nnls	5.2	83%

Experiment 3: BES-OOB Pruning

Final Number of Trees in the BES-OOB ensemble



Strategy	Avg. final #.Trees	Avg. Reduction
BES-OOB-avg (no pruning)	106.7 (36% of 300)	-
BES-OOB-ce	23.2	78%
BES-OOB-nnls	19.0	82%

Conclusions

- [regression setting] BES-OOB is competitive to MultiBoosting, and is superior to Bagging and Gradient Boosting when using CART-like regression trees as the base learners
- Large and diverse model library could further boost BES-OOB's predictive performance
- CE and NNLS work well for BES-OOB pruning. Particularly, the latter method may produce a better BES-OOB model

When to use BES-OOB?

- accuracy is critical (many domains)
- don't need to understand the model (some domains)
- training cost is NOT critical (access to low cost clusters)
- need to optimize a special error metric or a combination of metrics
- data mining competitions :-)

Thank you!

Questions?

Supplementary material: NNLS Algorithm

A WEKA implementation can be downloaded from:

http://www.cs.waikato.ac.nz/~qs12/bagging_es/

Supplementary material: CE an example

Cocktail Ensemble: given two ensembles f_1 and f_2, a linear ensemble of ensembles f_1 and f_2 can be expressed as:

$$f^c = wf_1 + (1-w)f_2, \quad wrt \quad w \in [0,1]$$

where the optimal pairwise weight w^* can be estimated from data.

An example:

Assuming a BES-OOB ensemble has 3 base-level ES models: E_1, E_2, E_3

Estimate pairwise ensemble performance for each pair $\{E_{12}, E_{13}, E_{23}\}$

Say we found (pairwise-) ensemble $E_{12} = E_1 w^* + E_2(1-w^*)$ gives the best estimated performance among all pairs, then we use E_{12} as a new ensemble

If combining E_{12} and E_3 is better than E_{12}, then return $E_{E_{12}E3}$ as the final ensemble, Otherwise return E_{12} as the final ensemble (E_3 is pruned)