Ensembles of models - how to get the most benefits

Anna Kozak, Hubert Ruczyński

MI2.AI Seminar

Rich Caruana

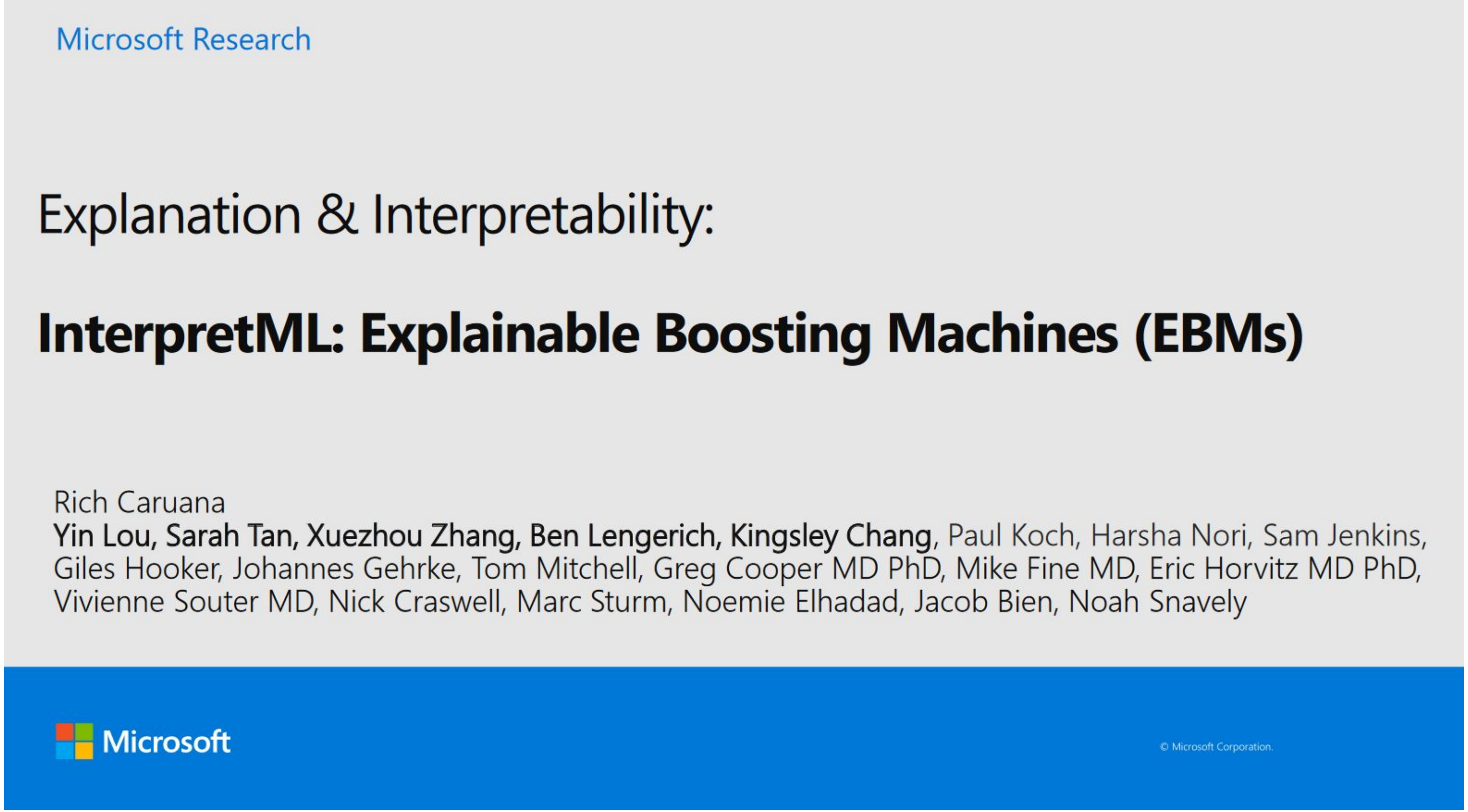
Microsoft Research

PAST

Department of Computer Science, Cornell University, USA

- Caruana, Rich & Niculescu-Mizil, Alexandru & Crew, Geo & Ksikes, Alex, " *Ensemble Selection from Libraries of Models*"
- Caruana, Rich & Munson, Art & Niculescu-Mizil, Alexandru, "*Getting the Most Out of Ensemble Selection*"

NOW



Microsoft Research

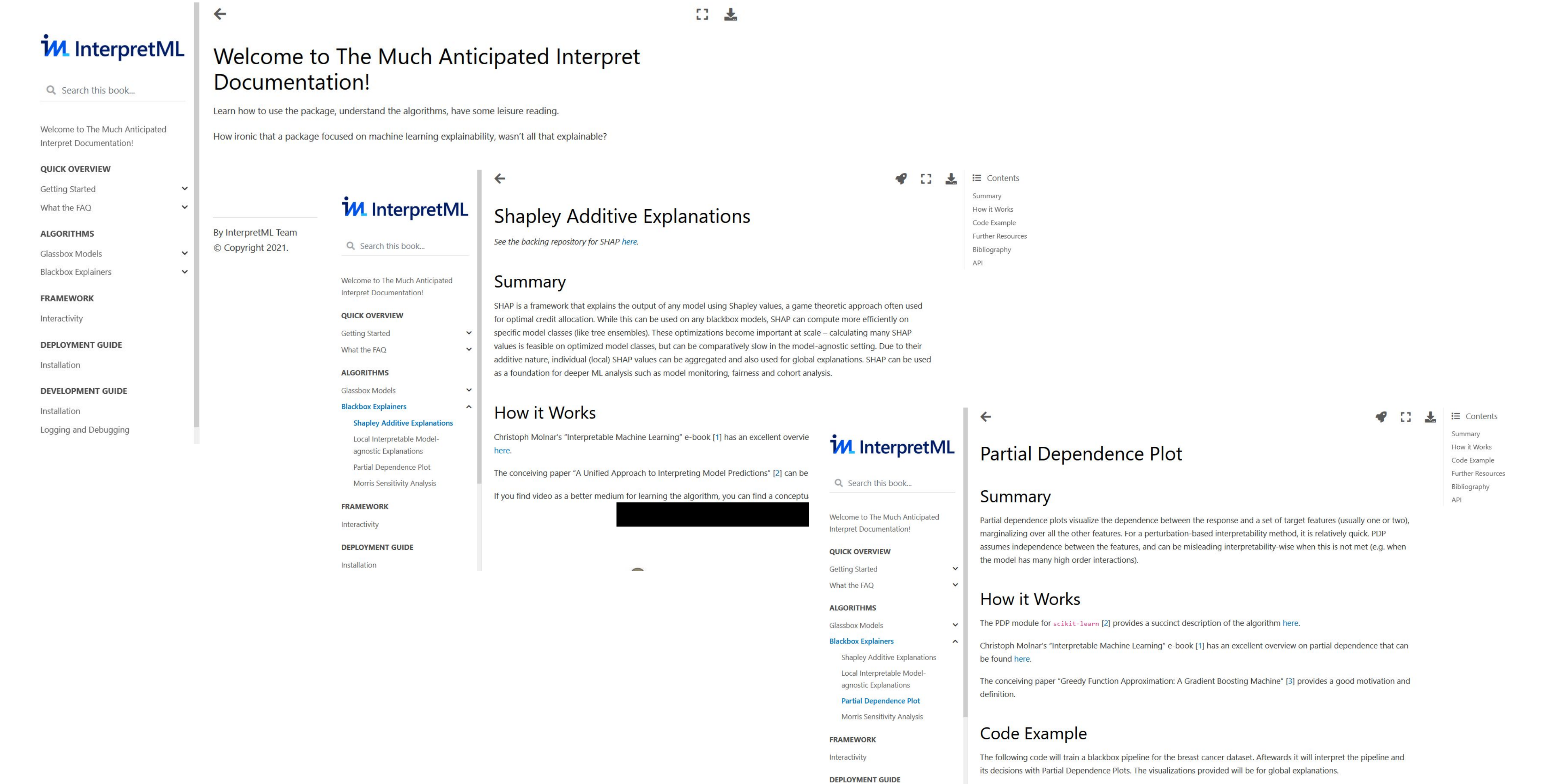
Explanation & Interpretability:

InterpretML: Explainable Boosting Machines (EBMs)

Rich Caruana
Yin Lou, Sarah Tan, Xuezhou Zhang, Ben Lengerich, Kingsley Chang, Paul Koch, Harsha Nori, Sam Jenkins, Giles Hooker, Johannes Gehrke, Tom Mitchell, Greg Cooper MD PhD, Mike Fine MD, Eric Horvitz MD PhD, Vivienne Souter MD, Nick Craswell, Marc Sturm, Noemie Elhadad, Jacob Bien, Noah Snavely

Microsoft

© Microsoft Corporation.



InterpretML

Search this book...

Welcome to The Much Anticipated Interpret Documentation!

QUICK OVERVIEW

Getting Started

What the FAQ

ALGORITHMS

Glassbox Models

Blackbox Explainers

FRAMEWORK

Interactivity

DEPLOYMENT GUIDE

Installation

DEVELOPMENT GUIDE

Installation

Logging and Debugging

Welcome to The Much Anticipated Interpret Documentation!

Learn how to use the package, understand the algorithms, have some leisure reading.

How ironic that a package focused on machine learning explainability, wasn't all that explainable?

By InterpretML Team
© Copyright 2021.

InterpretML

Search this book...

Welcome to The Much Anticipated Interpret Documentation!

QUICK OVERVIEW

Getting Started

What the FAQ

ALGORITHMS

Glassbox Models

Blackbox Explainers

Shapley Additive Explanations

Local Interpretable Model-agnostic Explanations

Partial Dependence Plot

Morris Sensitivity Analysis

FRAMEWORK

Interactivity

DEPLOYMENT GUIDE

Installation

Shapley Additive Explanations

See the backing repository for SHAP here.

Contents

Summary

How it Works

Code Example

Further Resources

Bibliography

API

Summary

SHAP is a framework that explains the output of any model using Shapley values, a game theoretic approach often used for optimal credit allocation. While this can be used on any blackbox models, SHAP can compute more efficiently on specific model classes (like tree ensembles). These optimizations become important at scale – calculating many SHAP values is feasible on optimized model classes, but can be comparatively slow in the model-agnostic setting. Due to their additive nature, individual (local) SHAP values can be aggregated and also used for global explanations. SHAP can be used as a foundation for deeper ML analysis such as model monitoring, fairness and cohort analysis.

How it Works

Christoph Molnar's "Interpretable Machine Learning" e-book [1] has an excellent overvie here.

The conceiving paper "A Unified Approach to Interpreting Model Predictions" [2] can be

If you find video as a better medium for learning the algorithm, you can find a conceptu

InterpretML

Search this book...

Welcome to The Much Anticipated Interpret Documentation!

QUICK OVERVIEW

Getting Started

What the FAQ

ALGORITHMS

Glassbox Models

Blackbox Explainers

Shapley Additive Explanations

Local Interpretable Model-agnostic Explanations

Partial Dependence Plot

Morris Sensitivity Analysis

FRAMEWORK

Interactivity

DEPLOYMENT GUIDE

Partial Dependence Plot

Contents

Summary

How it Works

Code Example

Further Resources

Bibliography

API

Summary

Partial dependence plots visualize the dependence between the response and a set of target features (usually one or two), marginalizing over all the other features. For a perturbation-based interpretability method, it is relatively quick. PDP assumes independence between the features, and can be misleading interpretability-wise when this is not met (e.g. when the model has many high order interactions).

How it Works

The PDP module for `scikit-learn` [2] provides a succinct description of the algorithm here.

Christoph Molnar's "Interpretable Machine Learning" e-book [1] has an excellent overview on partial dependence that can be found here.

The conceiving paper "Greedy Function Approximation: A Gradient Boosting Machine" [3] provides a good motivation and definition.

Code Example

The following code will train a blackbox pipeline for the breast cancer dataset. Aftewards it will interpret the pipeline and its decisions with Partial Dependence Plots. The visualizations provided will be for global explanations.

Ensemble Selection from Libraries of Models

Rich Caruana CARUANA@CS.CORNELL.EDU
Alexandru Niculescu-Mizil ALEXN@CS.CORNELL.EDU
Geoff Crew GC97@CS.CORNELL.EDU
Alex Ksikes AK107@CS.CORNELL.EDU
Department of Computer Science, Cornell University, Ithaca, NY 14853 USA

> “A necessary and sufficient condition for an ensemble of classifiers to be more accurate than any of its individual members is if the classifiers are accurate and diverse.”

Dietterich, T. G. (2000). Ensemble methods in machine learning. First International Workshop on Multiple Classifier Systems, 1–15.

Main contributions

- method for constructing ensembles from libraries of thousands of models
- experiment which demonstrated the benefit of ensemble selection

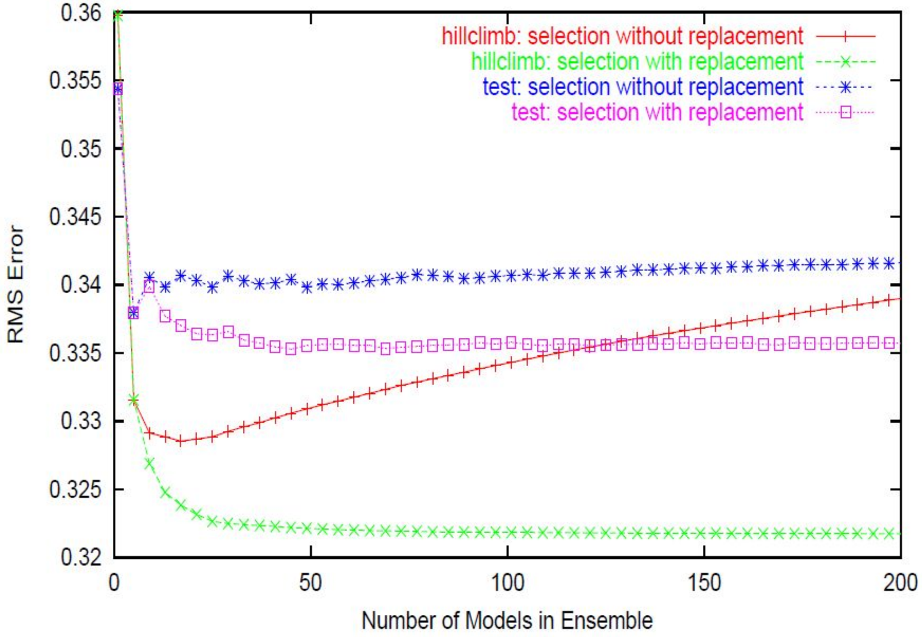
The basic ensemble selection procedure is very simple:

1. Start with the empty ensemble.
2. Add to the ensemble the model in the library that maximizes the ensemble's performance to the error metric on a validation set.
3. Repeat Step 2 for a fixed number of iterations or until all the models have been used.
4. Return the ensemble from the nested set of ensembles that has maximum performance on the validation set.

Improving Ensemble Selection

1. Selection with Replacement

- performance drops because the best models in the library have been used and selection must now add models that hurt the ensemble,
- the loss in performance can be significant if the peak is missed.



Improving Ensemble Selection

2. Sorted Ensemble Initialization

- forward selection sometimes overfits early in selection when ensembles are small
- starting with empty ensemble, sort the models in the library by their performance, and put the best N models in the ensemble
- we have 5-25 of the best models in ensemble before greedy stepwise selection begins

Improving Ensemble Selection

3. Bagged Ensemble Selection

- the number of models in a library increases, the chances of finding combinations of models that overif the validation set increases - *bagging can minimize this problem*
- random sample of models from the library, combination of M models overfits, the probability of those M models being in a random bag of models in less than (1-p)^M for p the fraction of models in the bag

Experiment

Datasets:

- ADULT, **COVER_TYPE, LETTER.p1, LETTER.p2**, MEDIS, SLAC, **HYPER_SPECT**
- binary classification problems
- large enough to allow moderate size train and validation sets, and still have data left for large final test sets
- train: 4000, valid: 1000, test: ~20000

Experiment

Performance metrics:

- accuracy (ACC)
- RMSE,
- mean cr0ss-entropy (MXE),
- LIFT,
- precision/recall break-even point (BEP),
- precision/recall F-score (FSC),
- average precision (APR),
- ROC,
- measure of probability calibration (CAL),
- SAR = (ACC + ROC + 1(-RMSE))/3

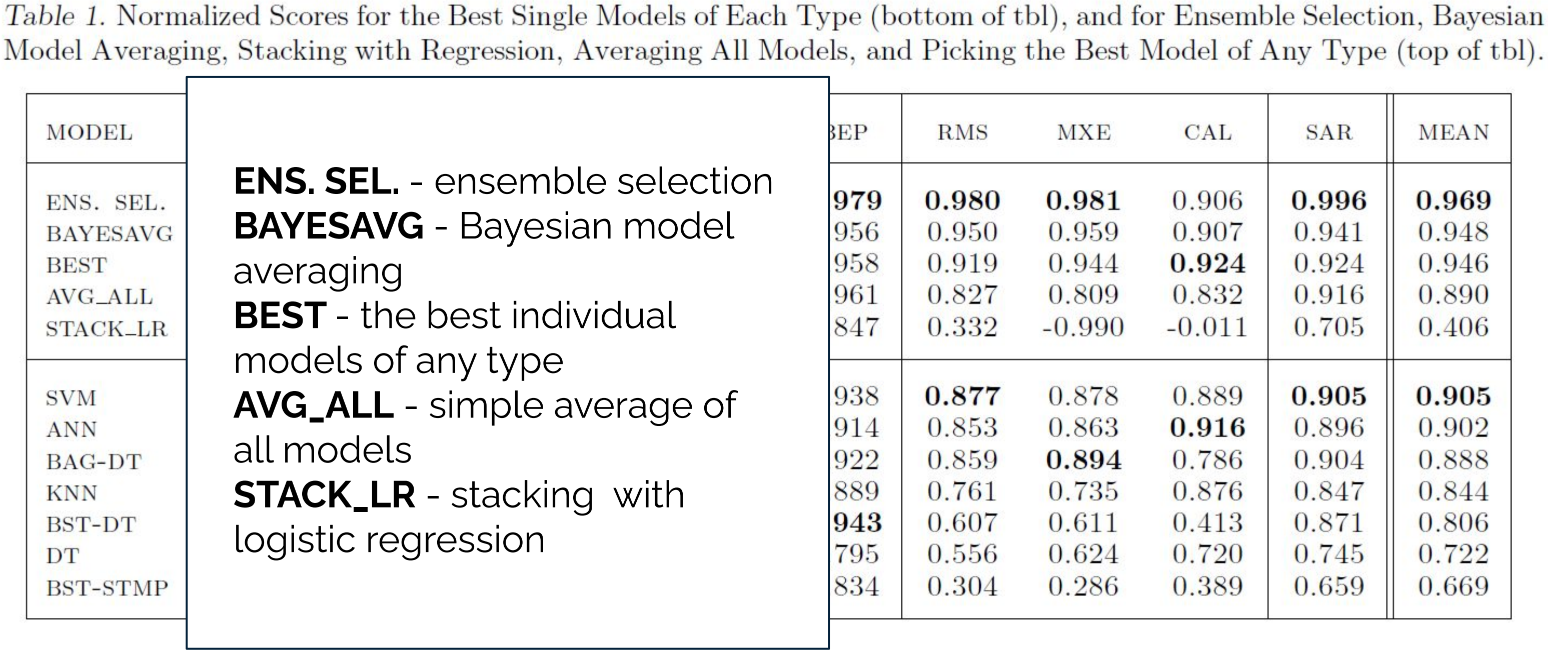


Table 1. Normalized Scores for the Best Single Models of Each Type (bottom of tbl), and for Ensemble Selection, Bayesian Model Averaging, Stacking with Regression, Averaging All Models, and Picking the Best Model of Any Type (top of tbl).

MODEL	BEP	RMS	MXE	CAL	SAR	MEAN
ENS. SEL.	**979**	**0.980**	**0.981**	0.906	**0.996**	**0.969**
BAYESAVG	956	0.950	0.959	0.907	0.941	0.948
BEST	958	0.919	0.944	**0.924**	0.924	0.946
AVG_ALL	961	0.827	0.809	0.832	0.916	0.890
STACK_LR	847	0.332	-0.990	-0.011	0.705	0.406
SVM	938	**0.877**	0.878	0.889	**0.905**	**0.905**
ANN	914	0.853	0.863	**0.916**	0.896	0.902
BAG-DT	922	0.859	**0.894**	0.786	0.904	0.888
KNN	889	0.761	0.735	0.876	0.847	0.844
BST-DT	**943**	0.607	0.611	0.413	0.871	0.806
DT	795	0.556	0.624	0.720	0.745	0.722
BST-STMP	834	0.304	0.286	0.389	0.659	0.669

ENS. SEL. - ensemble selection
BAYESAVG - Bayesian model averaging
BEST - the best individual models of any type
AVG_ALL - simple average of all models
STACK_LR - stacking with logistic regression

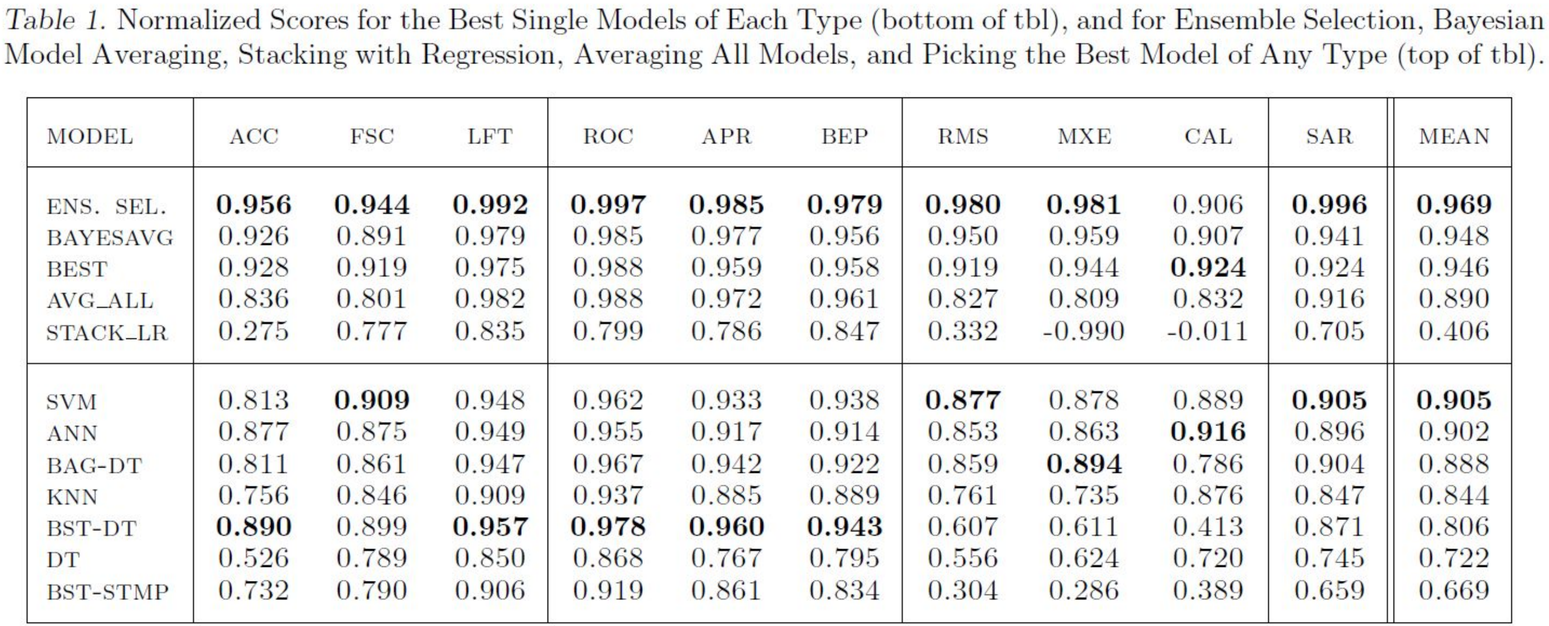


Table 1. Normalized Scores for the Best Single Models of Each Type (bottom of tbl), and for Ensemble Selection, Bayesian Model Averaging, Stacking with Regression, Averaging All Models, and Picking the Best Model of Any Type (top of tbl).

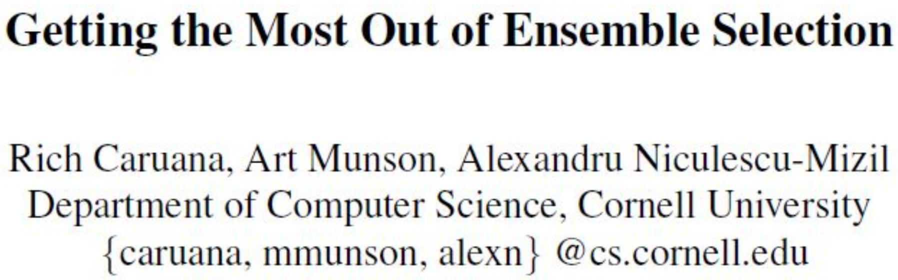
MODEL	ACC	FSC	LFT	ROC	APR	BEP	RMS	MXE	CAL	SAR	MEAN
ENS. SEL.	**0.956**	**0.944**	**0.992**	**0.997**	**0.985**	**0.979**	**0.980**	**0.981**	0.906	**0.996**	**0.969**
BAYESAVG	0.926	0.891	0.979	0.985	0.977	0.956	0.950	0.959	0.907	0.941	0.948
BEST	0.928	0.919	0.975	0.988	0.959	0.958	0.919	0.944	**0.924**	0.924	0.946
AVG_ALL	0.836	0.801	0.982	0.988	0.972	0.961	0.827	0.809	0.832	0.916	0.890
STACK_LR	0.275	0.777	0.835	0.799	0.786	0.847	0.332	-0.990	-0.011	0.705	0.406
SVM	0.813	**0.909**	0.948	0.962	0.933	0.938	**0.877**	0.878	0.889	**0.905**	**0.905**
ANN	0.877	0.875	0.949	0.955	0.917	0.914	0.853	0.863	**0.916**	0.896	0.902
BAG-DT	0.811	0.861	0.947	0.967	0.942	0.922	0.859	**0.894**	0.786	0.904	0.888
KNN	0.756	0.846	0.909	0.937	0.885	0.889	0.761	0.735	0.876	0.847	0.844
BST-DT	**0.890**	0.899	**0.957**	**0.978**	**0.960**	**0.943**	0.607	0.611	0.413	0.871	0.806
DT	0.526	0.789	0.850	0.868	0.767	0.795	0.556	0.624	0.720	0.745	0.722
BST-STMP	0.732	0.790	0.906	0.919	0.861	0.834	0.304	0.286	0.389	0.659	0.669

Conclusion

Using a variety of learning algorithms and parameter settings appears to be effective for generating libraries of diverse, high quality models.

Ensemble selection's most important feature is that it can optimize ensemble performance to any easily computed performance metric.

The performance metrics show that ensemble selection consistently finds ensembles that outperform all other models, including models trained with bagging, boosting, and Bayesian model averaging.



Getting the Most Out of Ensemble Selection

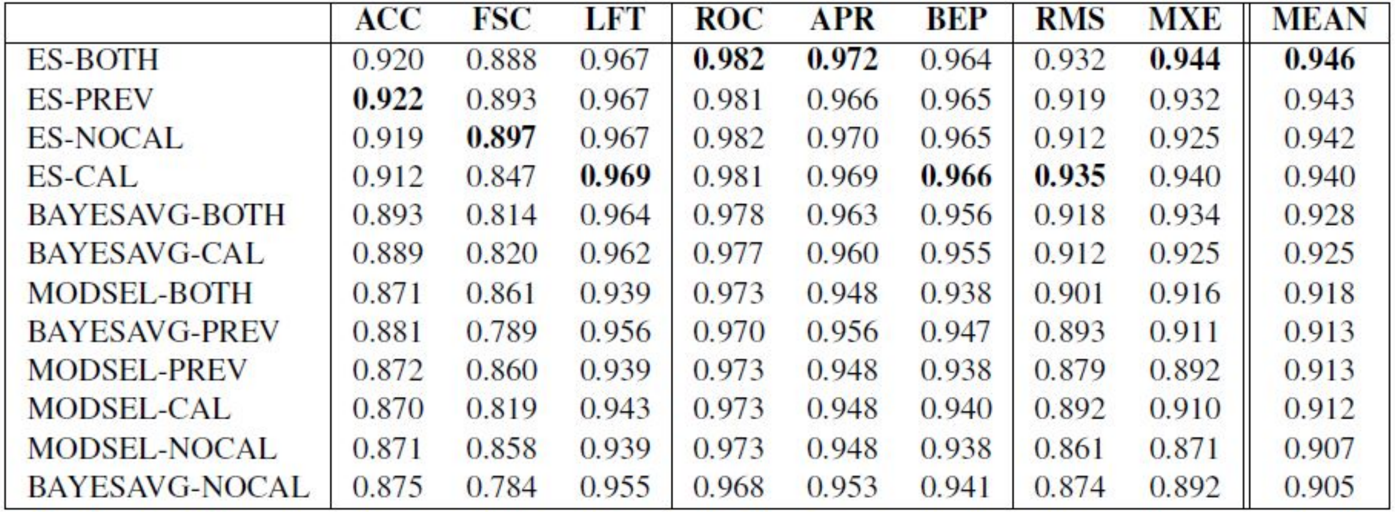
Rich Caruana, Art Munson, Alexandru Niculescu-Mizil
Department of Computer Science, Cornell University
{caruana, mmunson, alexn} @cs.cornell.edu

Main contributions

- Adjusting the models predictions into the canonical scale,
- Exploration of performance for different hillclimbing dataset sizes,
- Quantification of the models ability to optimize to arbitrary metric,
- Measuring the impact of pruning the models selection choice for the ES (Ensemble Selection) algorithm.

Ensembles of calibrated models

Table 1. Performance with and without model calibration. The best score in each column is bolded.



	ACC	FSC	LFT	ROC	APR	BEP	RMS	MXE	MEAN
ES-BOTH	0.920	0.888	0.967	**0.982**	**0.972**	0.964	0.932	**0.944**	**0.946**
ES-PREV	**0.922**	0.893	0.967	0.981	0.966	0.965	0.919	0.932	0.943
ES-NOCAL	0.919	**0.897**	0.967	0.982	0.970	0.965	0.912	0.925	0.942
ES-CAL	0.912	0.847	**0.969**	0.981	0.969	**0.966**	**0.935**	0.940	0.940
BAYESAVG-BOTH	0.893	0.814	0.964	0.978	0.963	0.956	0.918	0.934	0.928
BAYESAVG-CAL	0.889	0.820	0.962	0.977	0.960	0.955	0.912	0.925	0.925
MODSEL-BOTH	0.871	0.861	0.939	0.973	0.948	0.938	0.901	0.916	0.918
BAYESAVG-PREV	0.881	0.789	0.956	0.970	0.956	0.947	0.893	0.911	0.913
MODSEL-PREV	0.872	0.860	0.939	0.973	0.948	0.938	0.879	0.892	0.913
MODSEL-CAL	0.870	0.819	0.943	0.973	0.948	0.940	0.892	0.910	0.912
MODSEL-NOCAL	0.871	0.858	0.939	0.973	0.948	0.938	0.861	0.871	0.907
BAYESAVG-NOCAL	0.875	0.784	0.955	0.968	0.953	0.941	0.874	0.892	0.905

Analysis of training type

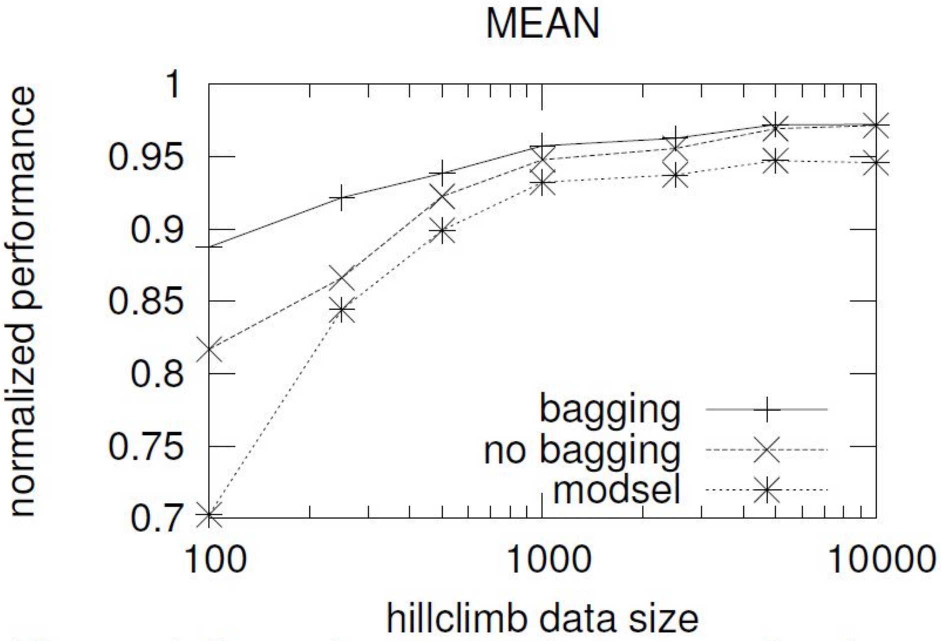


Figure 1. Learning curve for ens. selection.

Cross-validated ensemble selection

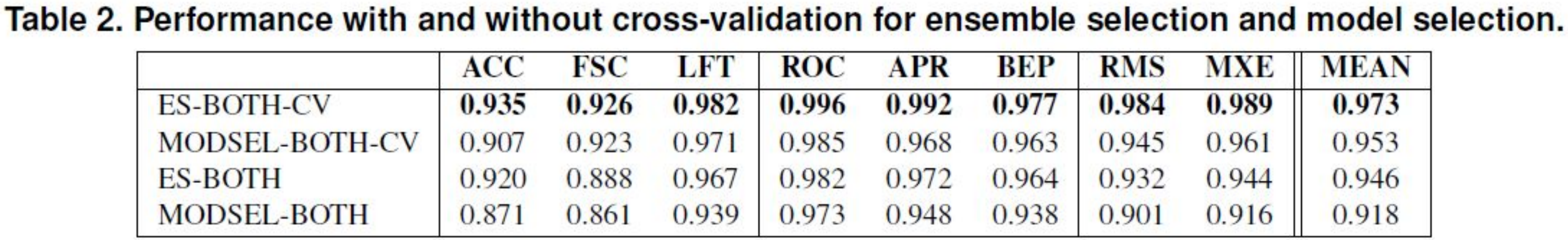


Table 2. Performance with and without cross-validation for ensemble selection and model selection.

	ACC	FSC	LFT	ROC	APR	BEP	RMS	MXE	MEAN
ES-BOTH-CV	**0.935**	**0.926**	**0.982**	**0.996**	**0.992**	**0.977**	**0.984**	**0.989**	**0.973**
MODSEL-BOTH-CV	0.907	0.923	0.971	0.985	0.968	0.963	0.945	0.961	0.953
ES-BOTH	0.920	0.888	0.967	0.982	0.972	0.964	0.932	0.944	0.946
MODSEL-BOTH	0.871	0.861	0.939	0.973	0.948	0.938	0.901	0.916	0.918

Direct metric optimisation

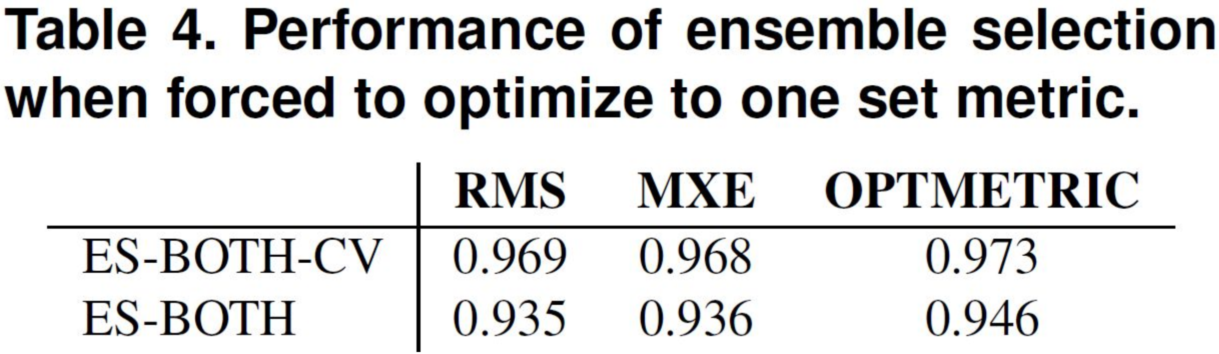


Table 4. Performance of ensemble selection when forced to optimize to one set metric.

	RMS	MXE	OPTMETRIC
ES-BOTH-CV	0.969	0.968	0.973
ES-BOTH	0.935	0.936	0.946

Model library pruning

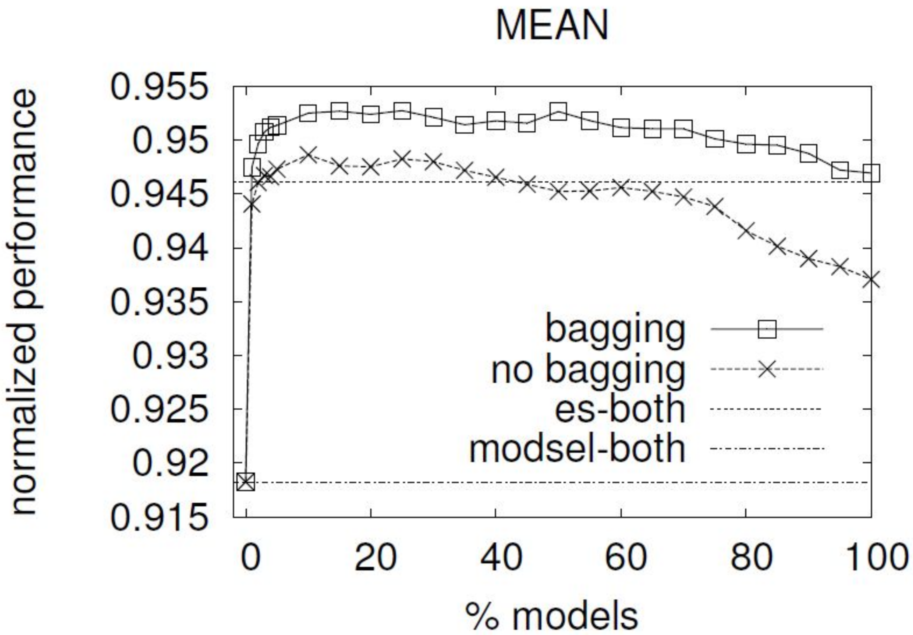


Figure 2. Pruned ens. selection performance.

Thank you for your attention!