Ensemble Selection

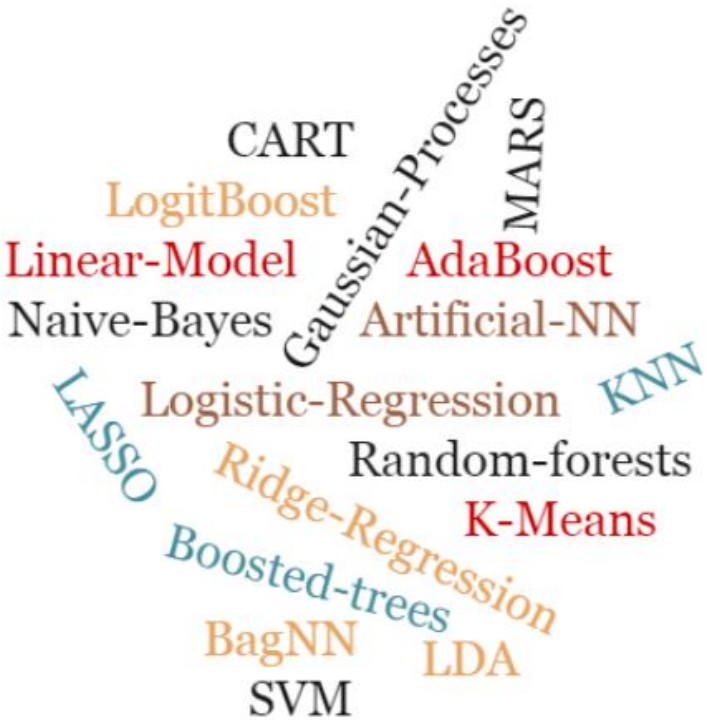
Shikhar Srivastava

Numerical Introductory Course
Humboldt–Universität zu Berlin

Motivation: Dawn of forecasting

- Medical Diagnosis
- Natural Language Processing
- Weather Predictions
- Economics and Finance
- Image Recognition
- and many more...

Tools of forecasting



Each of these algorithms is an effective prediction techniques.

A search space odyssey!

- Multiple hyper-parameters usually a large search choices of combinations.
- There is usually no rule of thumb.
- Researchers claiming a champion algorithm are often victimized by over-tuning.
- And this problem persists for each of these predictive algorithms

How do you make final predictions?

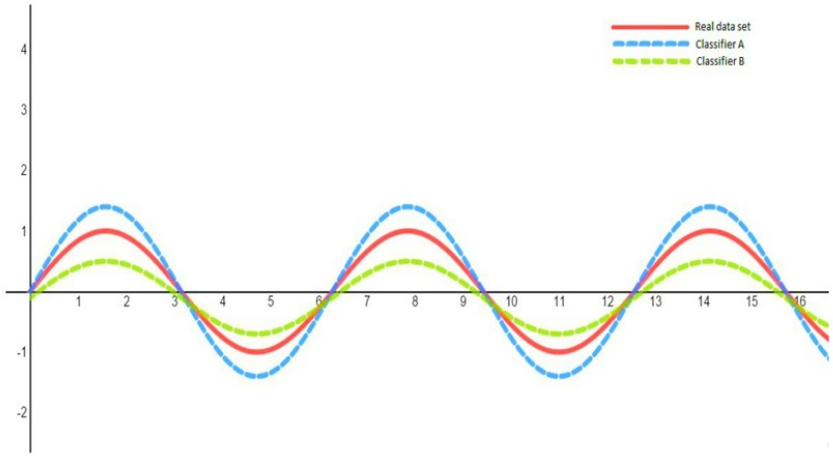
Ensemble Selection

Ensemble Modeling

Ensemble modeling is a method of combining different models and predicting an outcome that is more accurate than the outcomes from each individual model.

Diversity

The success of an ensemble system depends on the diversity of the classifiers. This means that we have *error* diversity in our ensemble.



- Reduces bias-variance trade-off.
- Confirms ambiguity decomposition.

Real World Example

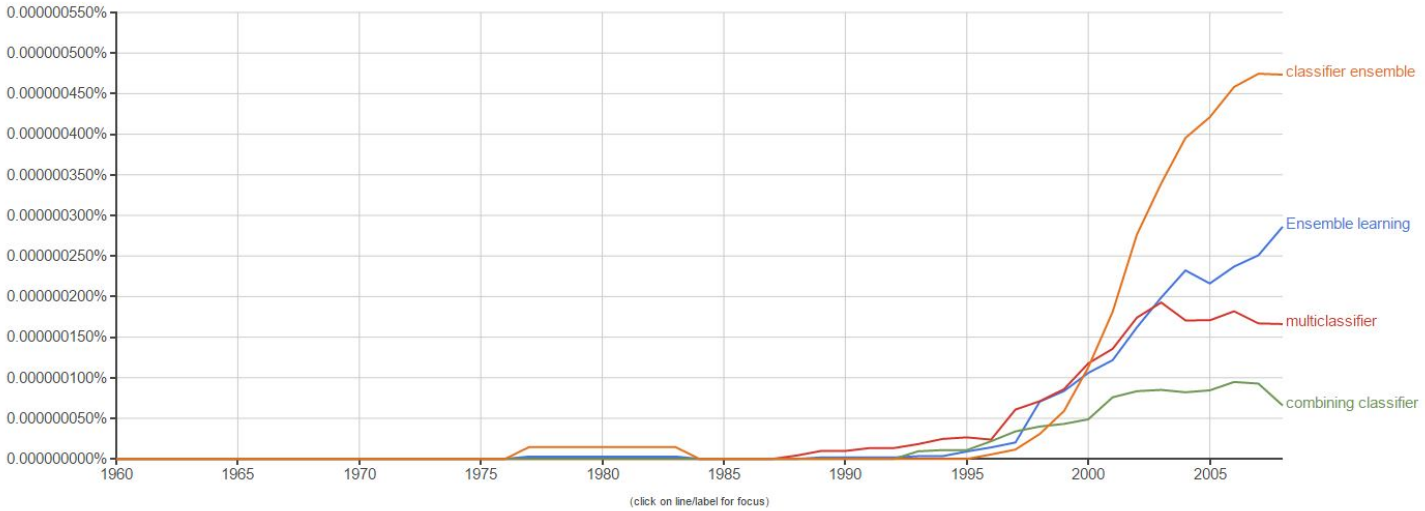
Lessons from the Netflix Prize Challenge

Robert M. Bell and Yehuda Koren
AT&T Labs – Research
180 Park Ave, Florham Park, NJ
{rbell,yehuda}@research.att.com

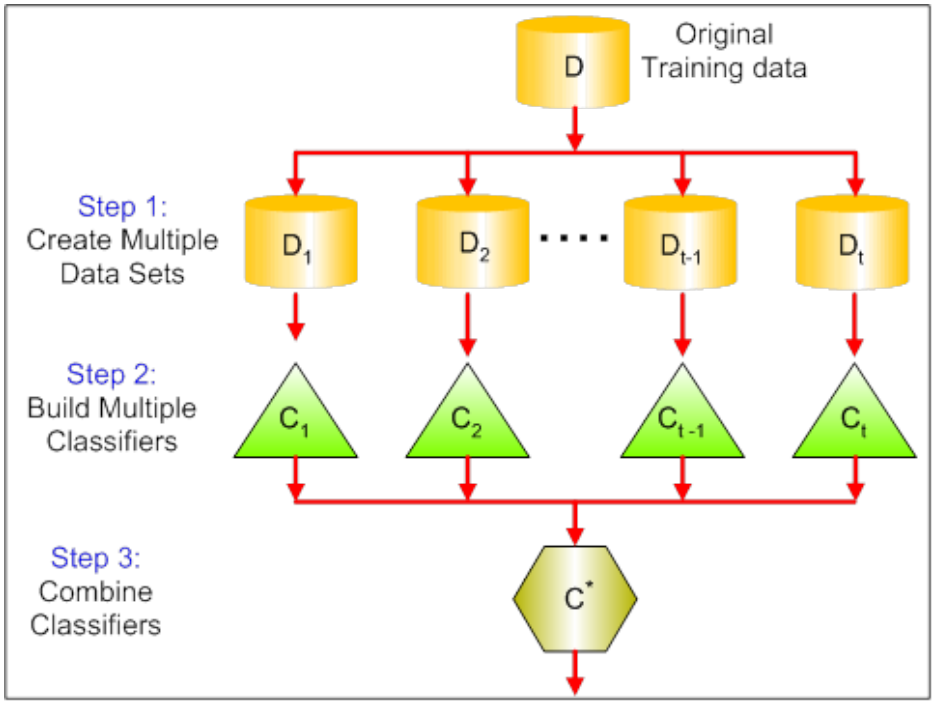
2. UTILIZING A COMPLEMENTARY SET OF MODELS

We found no perfect model. Instead, our best results came from combining predictions of models that complemented each other. While our winning entry, a linear combination of many prediction sets, achieved an improvement over Cinematch of 8.43%, the best single set of predictions reached only 6.57%. Even that method was a hybrid based on apply-

Popularity


0.000000550%
0.000000500%
0.000000450%
0.000000400%
0.000000350%
0.000000300%
0.000000250%
0.000000200%
0.000000150%
0.000000100%
0.000000050%
0.000000000%
1960
1965
1970
1975
1980
1985
1990
1995
2000
2005
(click on line/label for focus)
classifier ensemble
Ensemble learning
multiclassifier
combining classifier

Producing the Models



Producing the Models

- Tweaking classifier's hyper-parameters
- Leave one out or K-fold sampling.
- Bagging (bootstrap aggregation).

Creating an Ensemble

- **Ranking**: rank each model 1, ..., n based on diversity or other measure and delete similar or underperforming models.
- **Clustering**: group the models that make similar prediction and remove the most/least accurate models in each cluster.
- **Optimization**: search-based methods that find the subset of the original ensemble that optimizes performance.

- Tsoumakas (2009)

What we did.

- Analyzed Australian credit dataset.
- Applied k-folding and bagging methods to generate training samples.
- Used ANN, RF and LogR classifiers for building ensemble candidates.
- Used RF and LogR on top of these candidates as stacking agents.
- Area under the ROC curve (AUC) was used as goodness of fit.

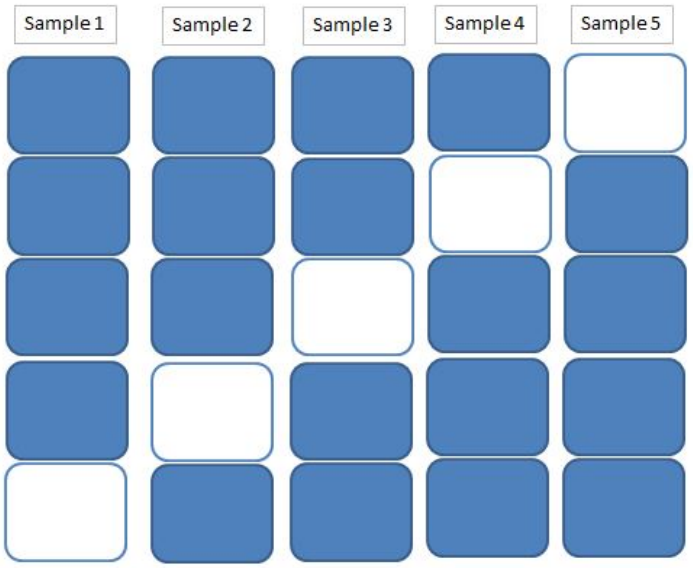
Data Split

- Train : 60% , Validation : 20% and Test : 20%
- Validation set acted as training set for Ensemble modeling.
- Train data is sub-sampled into 30 sets.

- Caruana (2006)

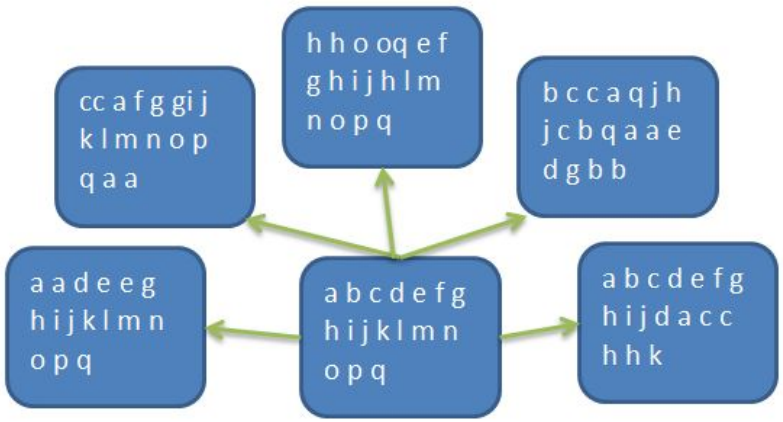
Sub-Sampling

- Using K-fold method 5 samples are created



Sub-Sampling

- Each of these sets were used to create bags of repeated random samples



Conclusion

- Stacking uses one of the classifier as generalizer over models.
- The improvement over base candidate performance was minor and inconclusive.
- The project showed scope for a meaningful research.

More Generalization!

Table 1: Datasets for comparison

Dataset	Observations	Attributes	Description
Australian Credit Approval	690	14	credit card application data
German Credit Data	1000	24	credit history information
Wisconsin Breast Cancer	680	10	classifying breast cancer as benign or malignant
Pima Indian Diabetes	770	9	forecasts the onset of diabetes mellitus among Pima Indian women using various health measures

- All the datasets are gathered from UCI data library for Machine Learning.

More Diversity!

Table 2: Classifiers used

Algorithm		Parametric Combinations
RF	Random Forests	54
GB	Gradient Boosting	96
SVM-l	Support Vector Machine (linear)	16
SVM-r	Support Vector Machine (radial)	44
NB	NaÃ¯ve Bayes	10
LogR	Logistic Regression	26
ELM	Extreme Learning Machine	39
ANN	Artificial Neural Networks	72

- Each model was made for every candidate which in turn increased the search space to 10710.

Backtracking Search Algorithm

Backtracking Search Optimization Algorithm (BSA) is a new evolutionary algorithm (EA) for solving real-valued numerical optimization problems as proposed by Pinar Civicioglu (2013)

- EA uses mechanisms inspired by biological evolution.
- Performs well in approximating solutions to all types optimization problems.

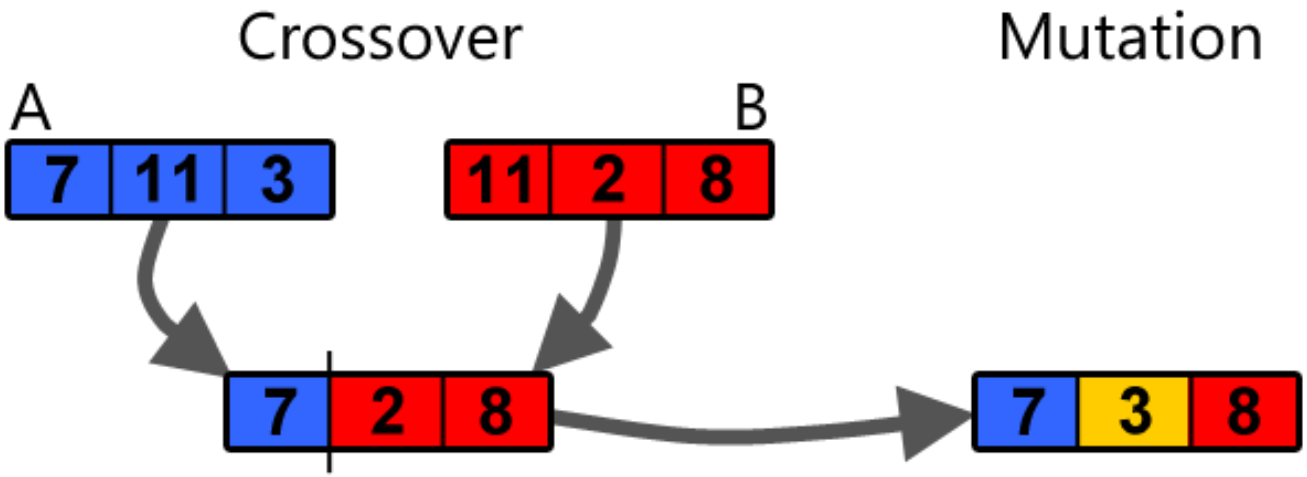
Evolutionary Algorithm

Inspired by nature where environmental pressures cause natural selection (survival of the fittest).

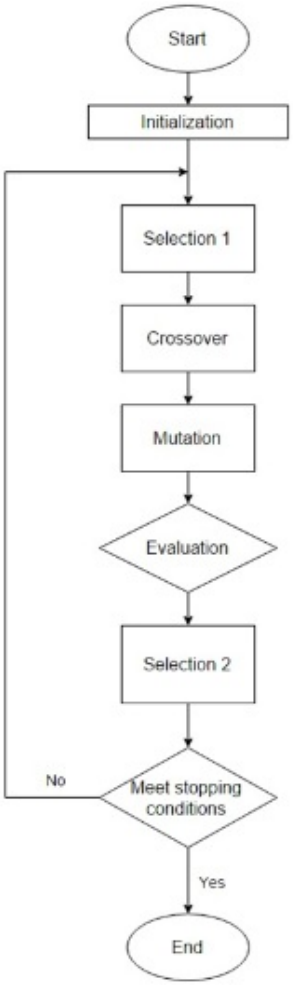
- We maximize our objective function on an abstract fitness measure.
- Intuition lies in the belief of nature's way of working.

Crossover and Mutation

The reason we exist!



Algorithm - Flowchart



Before going in

	Parent/Offspring Matrix				
	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	RF1	RF2	RF3	RF4	RF5
[2,]	LR1	LR2	LR3	LR4	LR5
[3,]	ANN1	ANN2	ANN3	ANN4	ANN5
[4,]	SVM1	SVM2	SVM3	SVM4	SVM5
[5,]	ELM1	ELM2	ELM3	ELM4	ELM5

- A matrix of $n \times m$ size is defined where $n \times m$ is the number of candidates.
- Each value in the matrix represents one candidate

Population Generation

- Two parents are created by assigning random values using following function.

```
GeneratePopulation <- function(popsize,dim,low,
  up){
    pop = matrix(1,popsize,dim)
    for (i in 1:popsize)    {
        for (j in 1:dim)        {
            pop[i,j]=runif(1,0,1)*(up[j]-low[j])
              +low[j]
        }
    }
    pop
}
```

Population Generation

	Parent P1				
	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	0.568	0.583	0.450	0.447	0.559
[2,]	0.616	0.979	0.856	0.198	0.570
[3,]	0.866	0.444	0.346	0.642	0.397
[4,]	0.397	0.521	0.232	0.165	0.966
[5,]	0.165	0.019	0.538	0.347	0.939

	Parent P2				
	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	0.862	0.089	0.498	0.746	0.222
[2,]	0.527	0.139	0.257	0.938	0.521
[3,]	0.791	0.908	0.303	0.561	0.841
[4,]	0.834	0.728	0.431	0.643	0.761
[5,]	0.335	0.998	0.681	0.682	0.636

Selection of "genes"

	Parent P2				
	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	0.862	0.089	0.498	0.746	0.222
[2,]	0.527	0.139	0.257	0.938	0.521
[3,]	0.791	0.908	0.303	0.561	0.841
[4,]	0.834	0.728	0.431	0.643	0.761
[5,]	0.335	0.998	0.681	0.682	0.636

After crossover and mutation

	Offspring				
	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	**0.870**	0.583	0.450	**0.183**	0.559
[2,]	0.616	**0.340**	**0.977**	0.198	**0.681**
[3,]	**0.444**	0.444	0.346	0.642	0.397
[4,]	0.397	**0.157**	0.232	0.165	0.966
[5,]	0.165	0.019	0.538	**0.112**	0.939

	Parent P1				
	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	0.568	0.583	0.450	0.447	0.559
[2,]	0.616	0.979	0.856	0.198	0.570
[3,]	0.866	0.444	0.346	0.642	0.397
[4,]	0.397	0.521	0.232	0.165	0.966
[5,]	0.165	0.019	0.538	0.347	0.939

- After crossover and mutation the performance of offspring is compared against Parent P1

After crossover and mutation

	New Parent P1				
	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	**0.870**	0.583	0.450	**0.183**	0.559
[2,]	0.616	**0.340**	**0.977**	0.198	**0.681**
[3,]	**0.444**	0.444	0.346	0.642	0.397
[4,]	0.397	**0.157**	0.232	0.165	0.966
[5,]	0.165	0.019	0.538	**0.112**	0.939

- Process is repeated till stopping conditions are met.

Change in Dimension Rate

Offspring

Candidate	Weight
1	0.93
2	0.33
3	0.21
4	0.23
5	0.56
6	0.44
7	0.60
8	0.50
9	0.77
10	0.34

Dimension rate
0.2

Dimension rate
0.6

Offspring

Candidate	Weight
1	0.02
2	0.29
3	0.21
4	0.87
5	0.56
6	0.44
7	0.24
8	0.03
9	0.98
10	0.82

Parent

Candidate	Weight
1	0.93
2	0.16
3	0.21
4	0.23
5	0.56
6	0.44
7	0.24
8	0.50
9	0.77
10	0.34

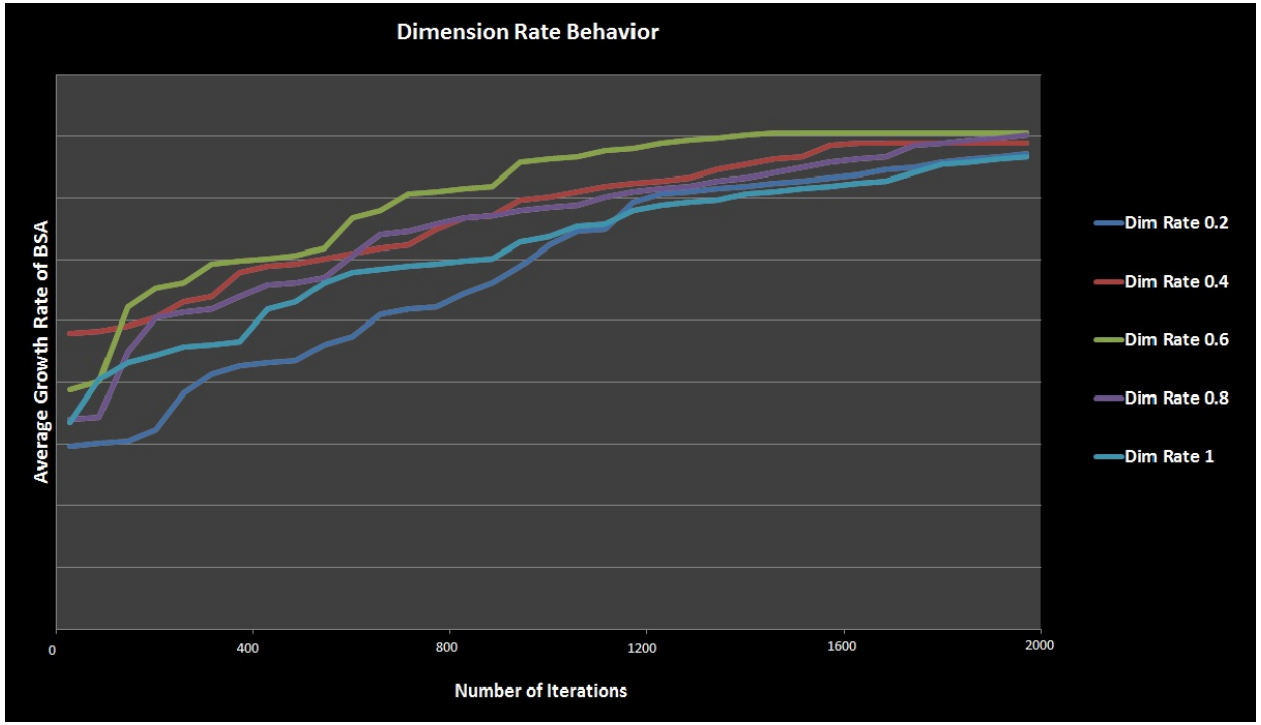
Candidate	Weight
1	0.86
2	0.16
3	0.15
4	0.23
5	0.96
6	0.44
7	0.24
8	0.29
9	0.77
10	0.34

0.4
Dimension rate

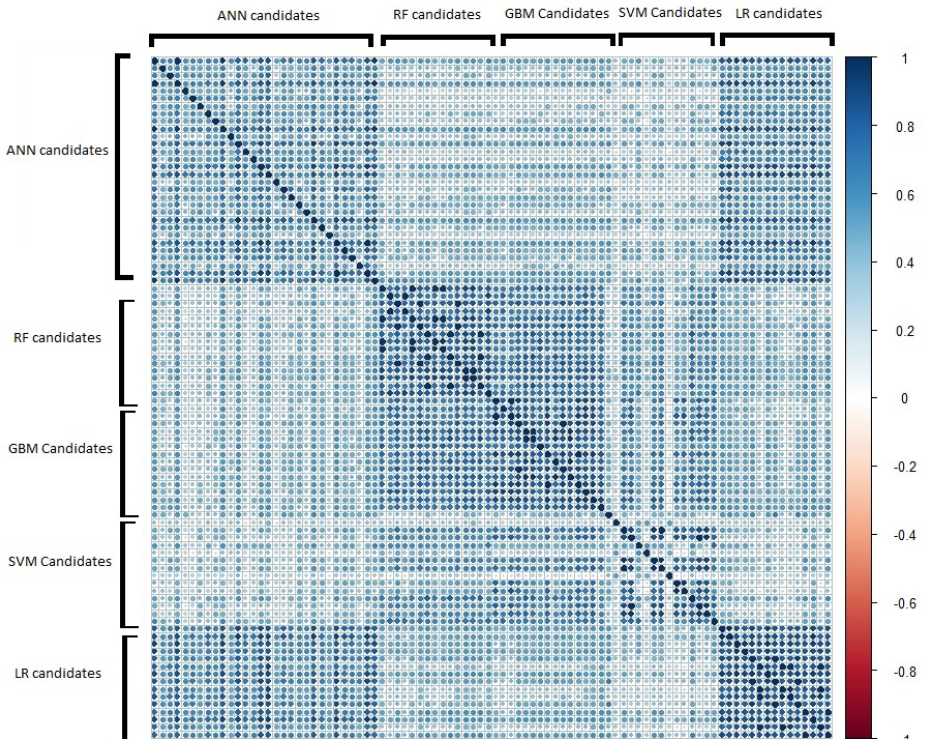
0.8
Dimension rate

Candidate	Weight
1	0.37
2	0.48
3	0.21
4	0.23
5	0.89
6	0.48
7	0.93
8	0.77
9	1.00
10	0.45

Change in Dimension Rate



Pruning the dataset



Pruning the dataset

- Started with 10710 candidates.
- After removing candidates with 85% correlation, only 1042 candidates were left.

Results - on Pruned Data

Table 3: AUC values from Test dataset

Algorithm	Dataset			
	Australian Credit	German Credit	Indian Pima Diabetes	Breast cancer
BSA	0.9107	0.9926	0.8106	0.9889
Stack-RF	0.9289	0.9888	**0.8423**	0.9972
RF	0.9194	0.9869	0.8116	0.9947
GBM	0.9026	**0.9938**	0.8163	0.9977
SVM-L	**0.9301**	0.9818	0.8308	**0.9985**
SVM-R	0.9298	0.9849	0.8224	0.9883
LogR	0.9242	0.9870	0.8363	0.9982
ANN	0.9300	0.9866	0.8163	0.9899
ELM	0.9021	0.9806	0.8359	0.9883
NB	0.9164	0.9787	0.8196	0.9975

Results - on Full Data

Table 4: AUC values from Test dataset

Algorithm	Dataset			
	Australian Credit	German Credit	Indian Pima Diabetes	Breast cancer
BSA	**0.9372**	**0.9952**	**0.8444**	0.9970
Stack-RF	0.9072	0.9757	0.8116	0.9972
RF	0.9194	0.9869	0.8123	0.9947
GBM	0.9026	0.9938	0.8163	0.9977
SVM-L	0.9301	0.9818	0.8308	**0.9985**
SVM-R	0.9298	0.9849	0.8224	0.9883
LogR	0.9242	0.9870	0.8363	0.9982
ANN	0.9300	0.9866	0.8163	0.9899
ELM	0.9021	0.9806	0.8359	0.9883
NB	0.9164	0.9787	0.8196	0.9975

Conclusion

- BSA outperforms when provided with larger search space.
- However, there is no free lunch. BSA took 10x more processing to find the optimal results.
- The test in itself is not conclusive of algorithm's real-world application
- However, it definitely proved to be a strong contender...

Thank You

Appendix

Sources I

- G. Tsoumakas, I. Partalas, I. Vlahavas, "An Ensemble Pruning Primer", in: O. Okun, G. Valentini Eds. *Applications of Supervised and Unsupervised Ensemble Methods*, Springer, Berlin, pp. 1-13. 2009.
- Opitz, David; Maclin, Richard: "Popular ensemble methods: An empirical Study" in: *Journal of Artificial Intelligence Research Journal*, 1999,Volume 11, pages 169-198.
- Caruana, Rich; Munson, Art; Niculescu-Mizil, Alexandru, "Getting the Most Out of Ensemble Selection", Department of Computer Science Cornell University 2006.

Sources II

- Civicioglu, Pinar, “Backtracking Search Optimization Algorithm for numerical optimization problems” Erciyes University, College of Aviation, Dept. of Aircraft Electrics and Electronics, Kayseri, Turkey, 2013.
- Lin, David; Mackey, Lester; Sill, Joseph; Takacs, Gabor, “Feature-Weighted Linear Stacking”, CoRR Journal, pp.1-17, October 2009
- Dasarathy, B.V. Sheela, “Composite classifier system design: concepts and methodology,“ Proceedings of the IEEE, vol. 67, no. 5, pp. 708-713. 1979.

Sources III

- Hansen, L.; Salamon, P., "Neural network ensembles," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 10, pp. 993-1001, 1990.
- Schapire, R.E., "The Strength of Weak Learnability," Machine Learning, vol. 5, no. 2, pp. 197-227, 1990.