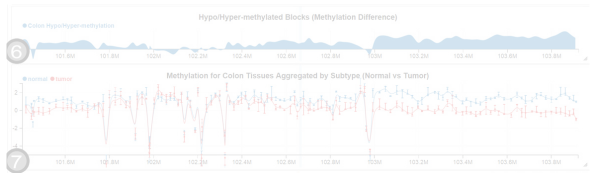


Ensemble Methods

Héctor Corrada Bravo

University of Maryland, College Park, USA

CMSC 643: 2018-10-23

Ensemble Learning

In studying the Random Forest algorithm we have seen some advantages of aggregating models to improve our predictions.

Ensemble Learning

In studying the Random Forest algorithm we have seen some advantages of aggregating models to improve our predictions.

This idea of combining models is called *Ensemble Learning* in general.

Ensemble Learning

In studying the Random Forest algorithm we have seen some advantages of aggregating models to improve our predictions.

This idea of combining models is called *Ensemble Learning* in general.

At a high level, *Ensemble Learning* is *the aggregation of predictions of multiple weak learners with the goal of improving prediction performance.*

Ensemble Learning

We will see three general approaches to this methodology:

Ensemble Learning

We will see three general approaches to this methodology:

bagging: uses ensembles to reduce the variability of single ML models,

Ensemble Learning

We will see three general approaches to this methodology:

bagging: uses ensembles to reduce the variability of single ML models,

stacking: uses ensembles to capture different characteristics of task, learning how to combine them

Ensemble Learning

We will see three general approaches to this methodology:

bagging: uses ensembles to reduce the variability of single ML models,

stacking: uses ensembles to capture different characteristics of task, learning how to combine them

boosting: uses ensembles of ML models each capturing a specific subspace of predictor space.

The story of the Netflix Prize

In October 2006 Netflix announced an ML prize around their movie recommendation engine.

The story of the Netflix Prize

In October 2006 Netflix announced an ML prize around their movie recommendation engine.

Supervised learning task:

- Dataset of users and their ratings, (1,2,3,4 or 5 stars), of movies they have rated.
- Build an ML model that given predicts a specific user's rating to a movie they have not rated.

The story of the Netflix Prize

In October 2006 Netflix announced an ML prize around their movie recommendation engine.

Supervised learning task:

- Dataset of users and their ratings, (1,2,3,4 or 5 stars), of movies they have rated.
- Build an ML model that given predicts a specific user's rating to a movie they have not rated.

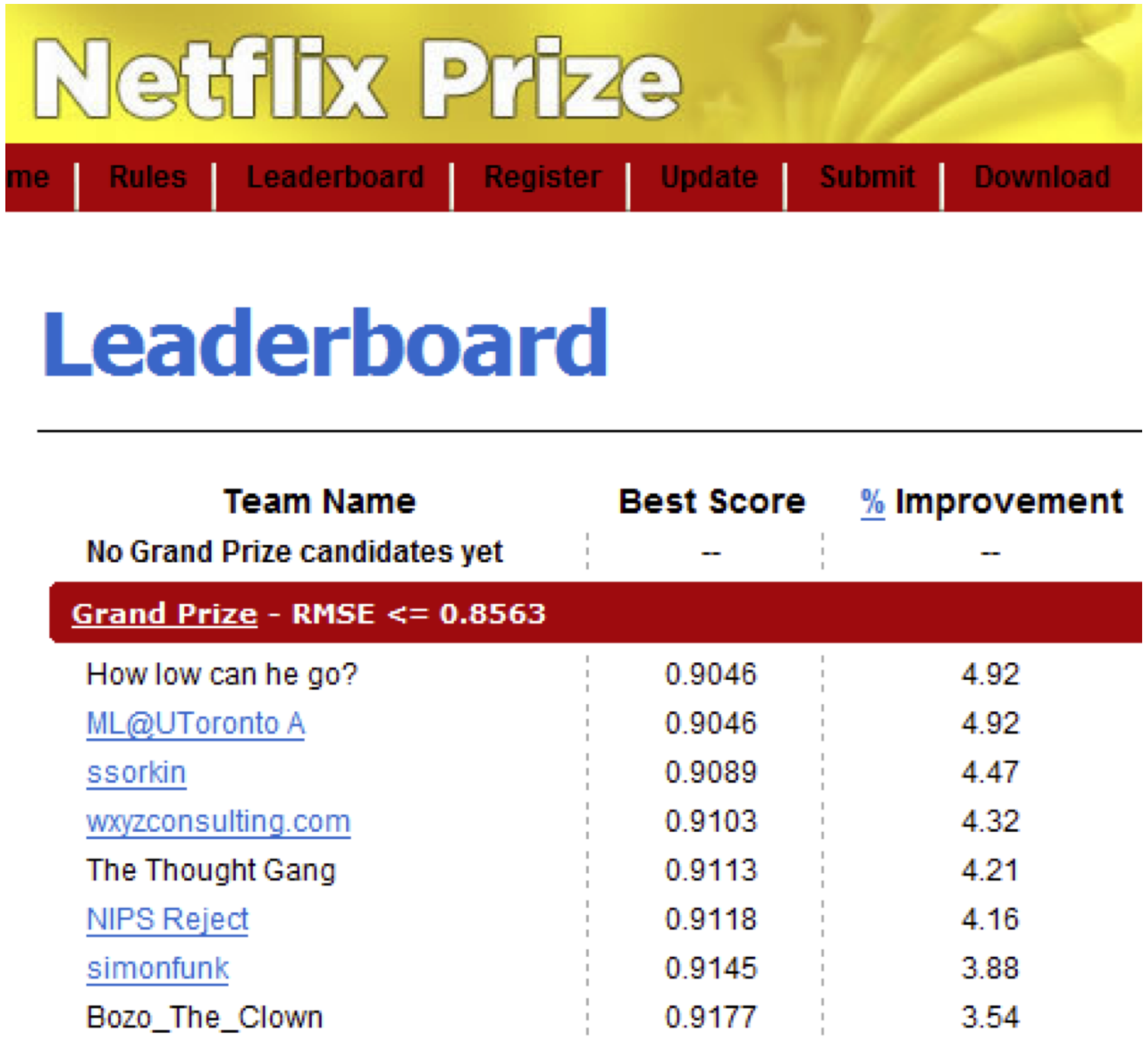
The idea is that they can then recommend movies to those users if they predict they would rate them highly.

The story of the Netflix Prize

Netflix would award $1M for the first ML system that provided a 10% improvement to their existing system

The story of the Netflix Prize

Existing system had a 0.9514 mean squared error


Netflix Prize
me | Rules | Leaderboard | Register | Update | Submit | Download

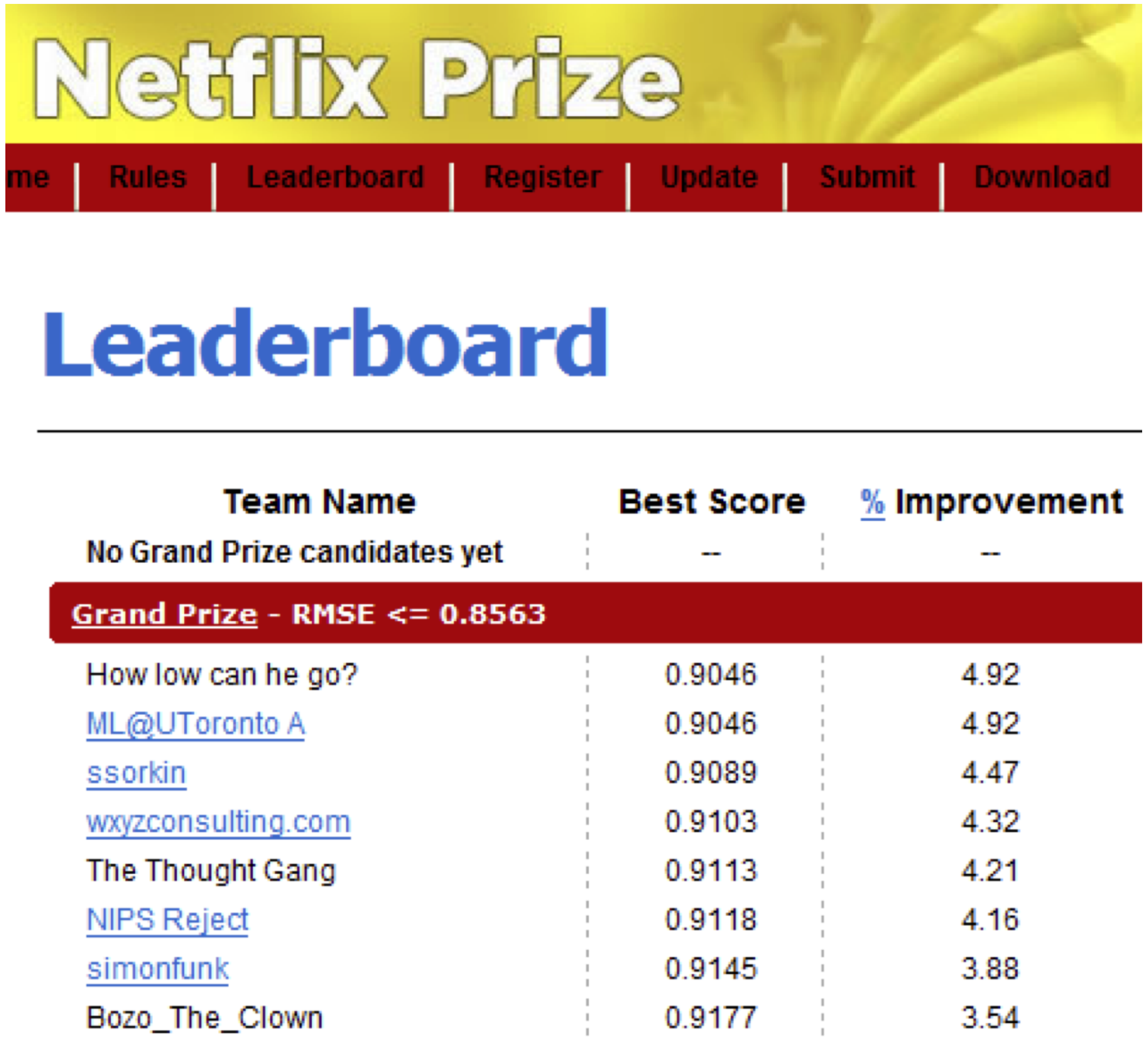
Leaderboard

Team Name	Best Score	% Improvement
No Grand Prize candidates yet	--	--
Grand Prize - RMSE <= 0.8563		
How low can he go?	0.9046	4.92
ML@UToronto A	0.9046	4.92
ssorkin	0.9089	4.47
wxyzconsulting.com	0.9103	4.32
The Thought Gang	0.9113	4.21
NIPS Reject	0.9118	4.16
simonfunk	0.9145	3.88
Bozo_The_Clown	0.9177	3.54

The story of the Netflix Prize

Within three weeks, at least 40 teams had improved upon the existing Netflix system.

The top teams were showing improvement over 5%.

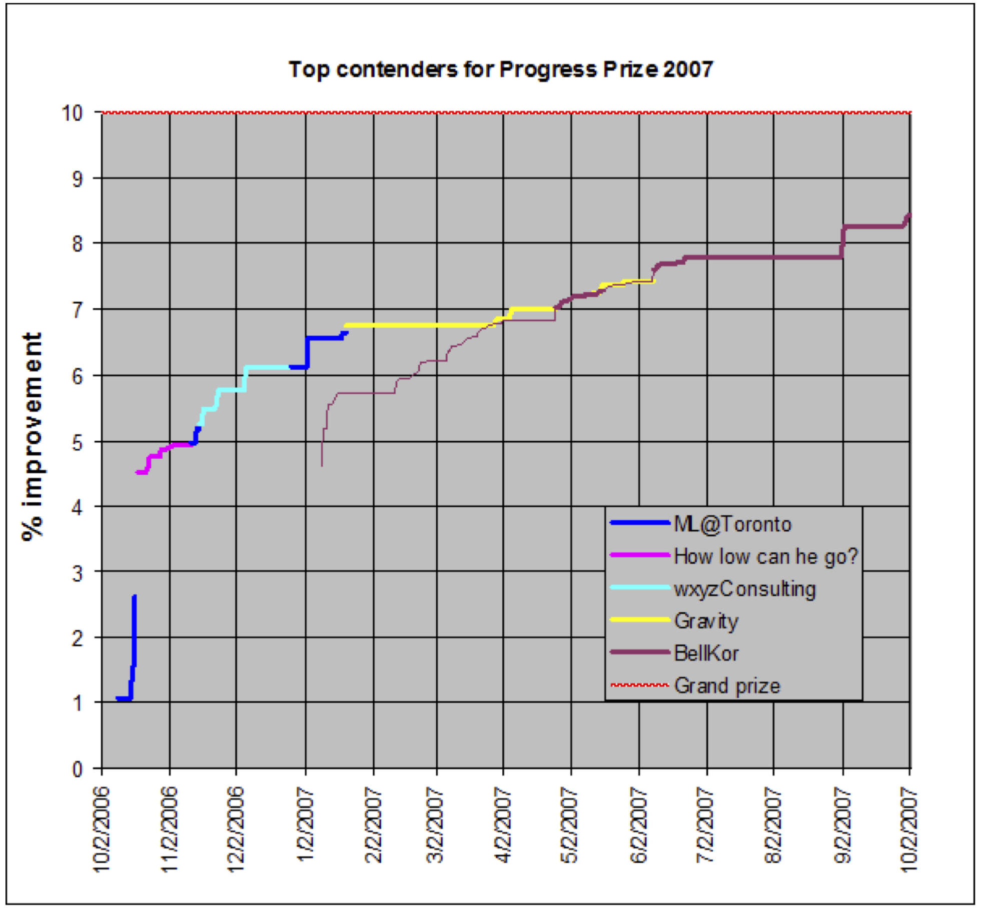

Netflix Prize
me | Rules | Leaderboard | Register | Update | Submit | Download

Leaderboard

Team Name	Best Score	% Improvement
No Grand Prize candidates yet	--	--
Grand Prize - RMSE <= 0.8563		
How low can he go?	0.9046	4.92
ML@UToronto A	0.9046	4.92
ssorkin	0.9089	4.47
wxyzconsulting.com	0.9103	4.32
The Thought Gang	0.9113	4.21
NIPS Reject	0.9118	4.16
simonfunk	0.9145	3.88
Bozo_The_Clown	0.9177	3.54

The story of the Netflix Prize

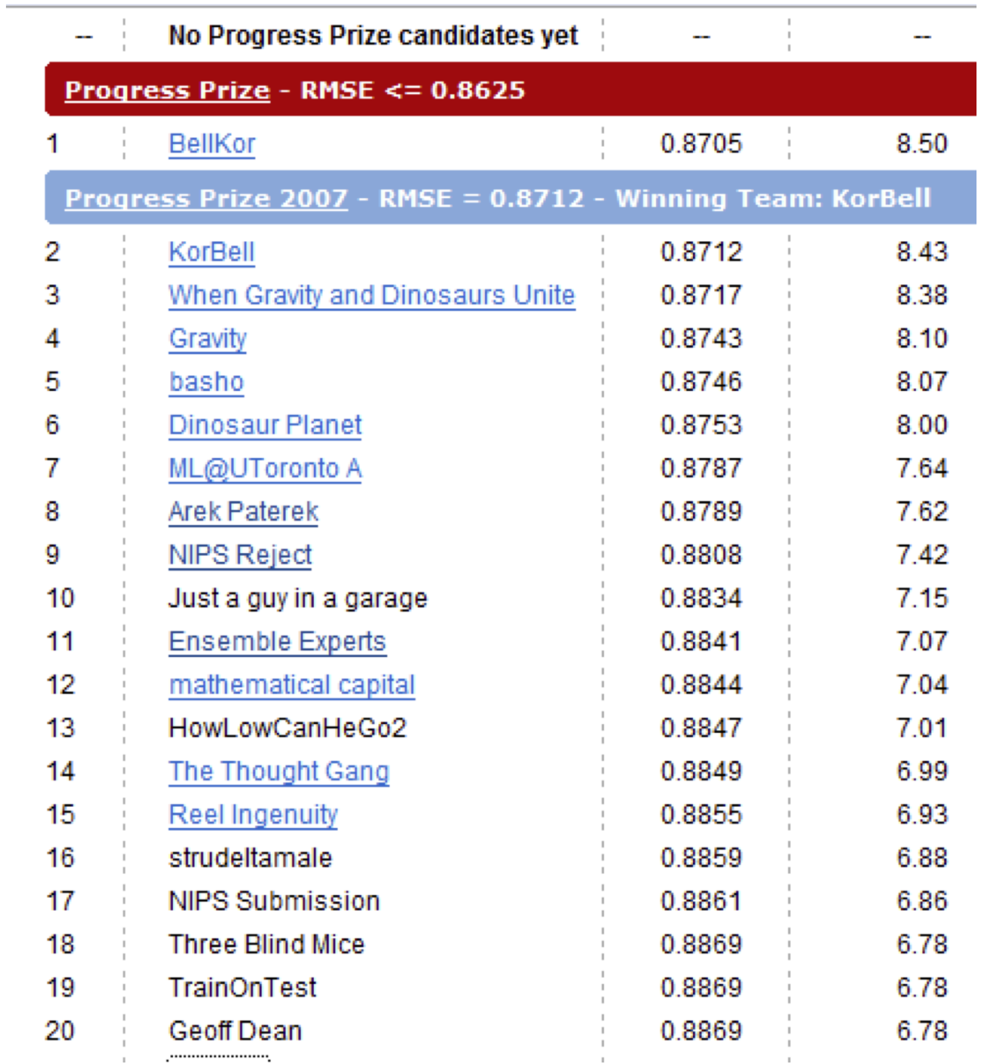
Progress soon slowed and teams were stuck at around 8-9% improvement over the existing system.



The story of the Netflix Prize

Along the way, progress prizes were awarded based on the top team at each challenge anniversary.

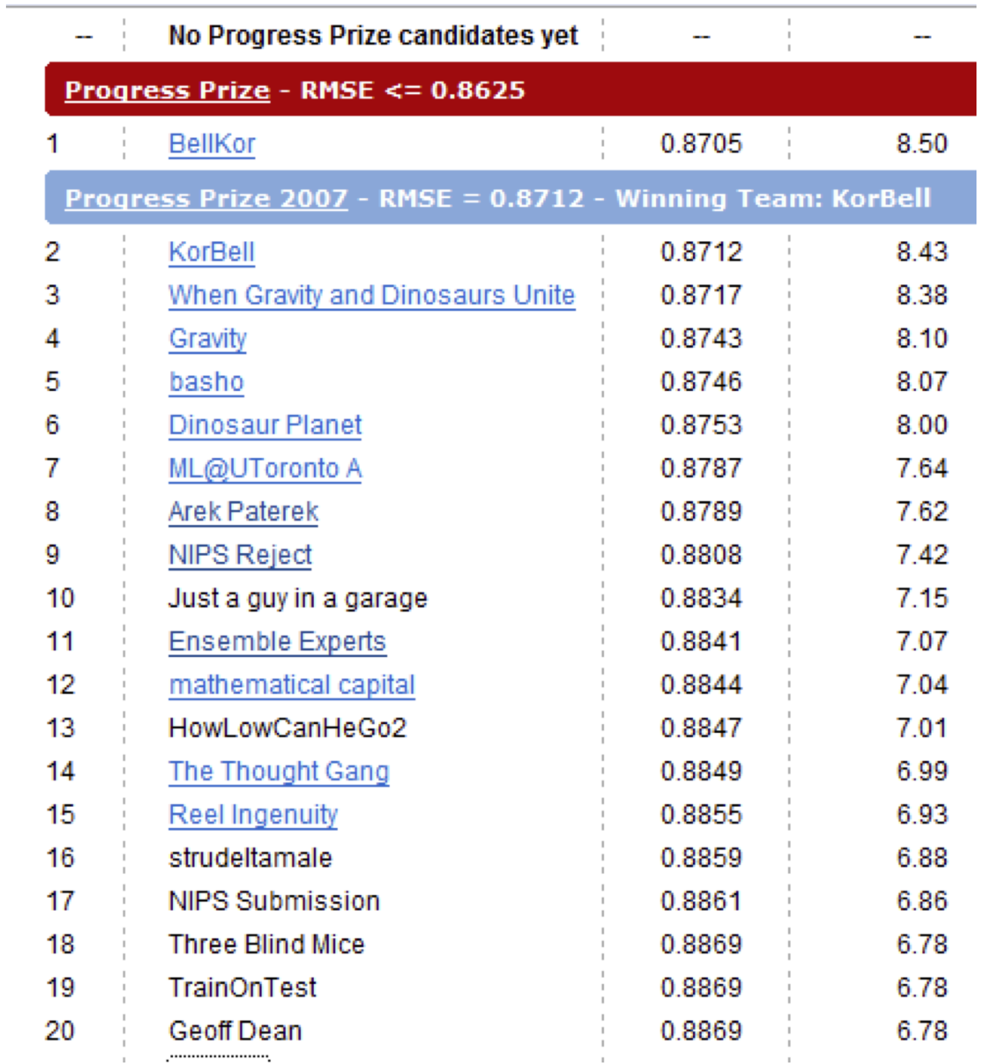
The top teams were using ensemble methods.



--	No Progress Prize candidates yet	--	--
Progress Prize - RMSE <= 0.8625			
1	BellKor	0.8705	8.50
Progress Prize 2007 - RMSE = 0.8712 - Winning Team: KorBell			
2	KorBell	0.8712	8.43
3	When Gravity and Dinosaurs Unite	0.8717	8.38
4	Gravity	0.8743	8.10
5	basho	0.8746	8.07
6	Dinosaur Planet	0.8753	8.00
7	ML@UToronto A	0.8787	7.64
8	Arek Paterek	0.8789	7.62
9	NIPS Reject	0.8808	7.42
10	Just a guy in a garage	0.8834	7.15
11	Ensemble Experts	0.8841	7.07
12	mathematical capital	0.8844	7.04
13	HowLowCanHeGo2	0.8847	7.01
14	The Thought Gang	0.8849	6.99
15	Reel Ingenuity	0.8855	6.93
16	strudeltamale	0.8859	6.88
17	NIPS Submission	0.8861	6.86
18	Three Blind Mice	0.8869	6.78
19	TrainOnTest	0.8869	6.78
20	Geoff Dean	0.8869	6.78

The story of the Netflix Prize

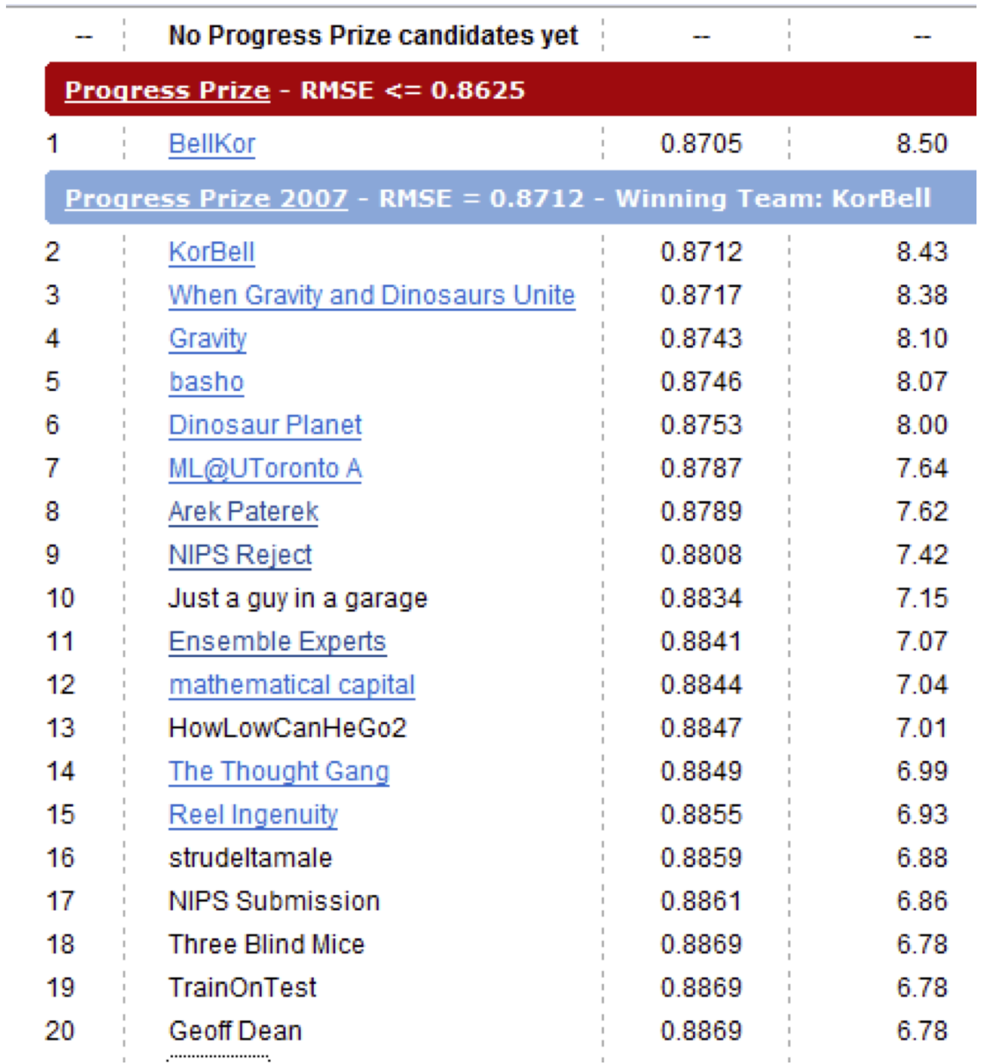
- **Arek Paterek**: "combine the results of many methods"
- **U of Toronto**: "when the predictions of multiple RBM and SVD models are linearly combined..."



--	No Progress Prize candidates yet	--	--
Progress Prize - RMSE <= 0.8625			
1	BellKor	0.8705	8.50
Progress Prize 2007 - RMSE = 0.8712 - Winning Team: KorBell			
2	KorBell	0.8712	8.43
3	When Gravity and Dinosaurs Unite	0.8717	8.38
4	Gravity	0.8743	8.10
5	basho	0.8746	8.07
6	Dinosaur Planet	0.8753	8.00
7	ML@UToronto A	0.8787	7.64
8	Arek Paterek	0.8789	7.62
9	NIPS Reject	0.8808	7.42
10	Just a guy in a garage	0.8834	7.15
11	Ensemble Experts	0.8841	7.07
12	mathematical capital	0.8844	7.04
13	HowLowCanHeGo2	0.8847	7.01
14	The Thought Gang	0.8849	6.99
15	Reel Ingenuity	0.8855	6.93
16	strudeltamale	0.8859	6.88
17	NIPS Submission	0.8861	6.86
18	Three Blind Mice	0.8869	6.78
19	TrainOnTest	0.8869	6.78
20	Geoff Dean	0.8869	6.78

The story of the Netflix Prize

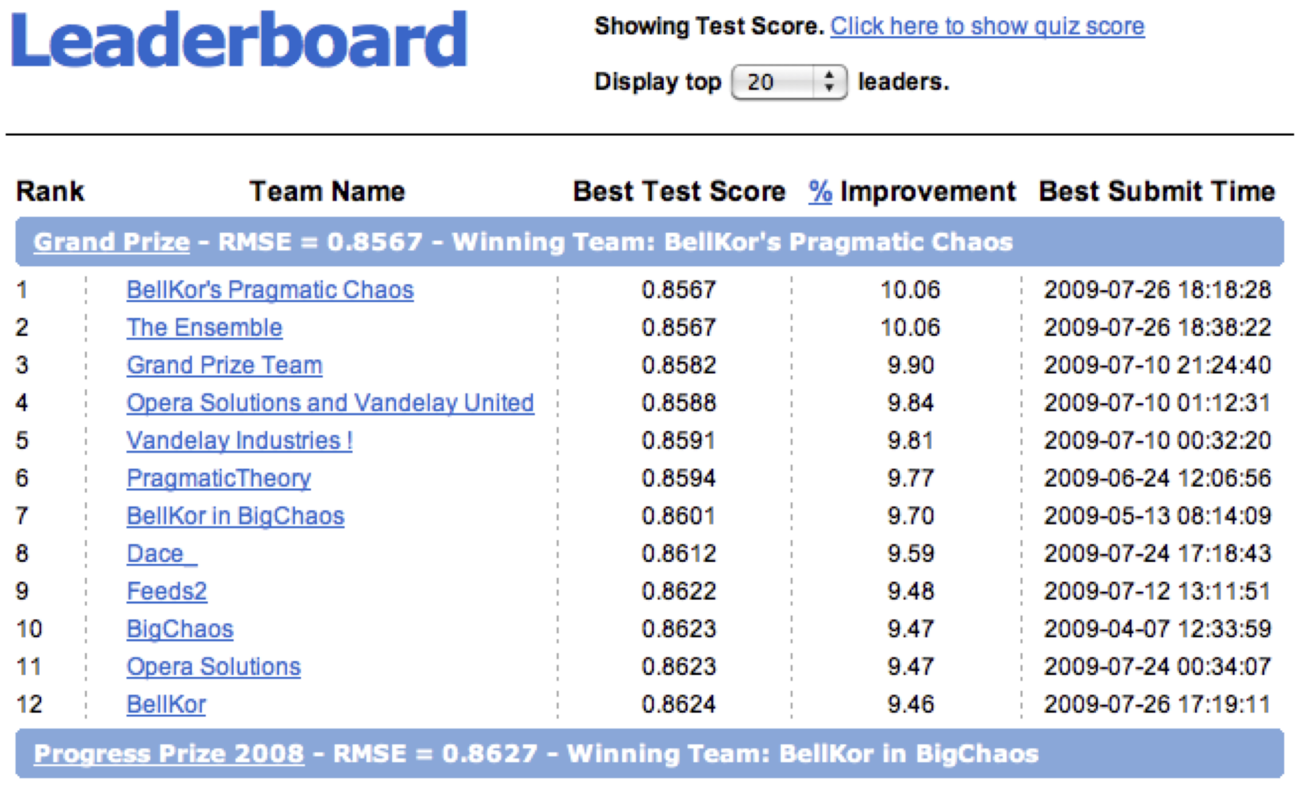
- **When Gravity and Dinosaurs Unite**: "Our common team blends the result of team Gravity and team Dinosaur Planet"
- **BellKor**: "Our final solution consists of blending 107 individual resluts"



--	No Progress Prize candidates yet	--	--
Progress Prize - RMSE <= 0.8625			
1	BellKor	0.8705	8.50
Progress Prize 2007 - RMSE = 0.8712 - Winning Team: KorBell			
2	KorBell	0.8712	8.43
3	When Gravity and Dinosaurs Unite	0.8717	8.38
4	Gravity	0.8743	8.10
5	basho	0.8746	8.07
6	Dinosaur Planet	0.8753	8.00
7	ML@UToronto A	0.8787	7.64
8	Arek Paterek	0.8789	7.62
9	NIPS Reject	0.8808	7.42
10	Just a guy in a garage	0.8834	7.15
11	Ensemble Experts	0.8841	7.07
12	mathematical capital	0.8844	7.04
13	HowLowCanHeGo2	0.8847	7.01
14	The Thought Gang	0.8849	6.99
15	Reel Ingenuity	0.8855	6.93
16	strudeltamale	0.8859	6.88
17	NIPS Submission	0.8861	6.86
18	Three Blind Mice	0.8869	6.78
19	TrainOnTest	0.8869	6.78
20	Geoff Dean	0.8869	6.78

The story of the Netflix Prize

Ultimately, the challenge was won when multiple top teams allied to combine their own ensemble models as ensembles.



Leaderboard

Showing Test Score. Click here to show quiz score

Display top 20 leaders.

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos				
1	BellKor's Pragmatic Chaos	0.8567	10.06	2009-07-26 18:18:28
2	The Ensemble	0.8567	10.06	2009-07-26 18:38:22
3	Grand Prize Team	0.8582	9.90	2009-07-10 21:24:40
4	Opera Solutions and Vandelay United	0.8588	9.84	2009-07-10 01:12:31
5	Vandelay Industries !	0.8591	9.81	2009-07-10 00:32:20
6	PragmaticTheory	0.8594	9.77	2009-06-24 12:06:56
7	BellKor in BigChaos	0.8601	9.70	2009-05-13 08:14:09
8	Dace_	0.8612	9.59	2009-07-24 17:18:43
9	Feeds2	0.8622	9.48	2009-07-12 13:11:51
10	BigChaos	0.8623	9.47	2009-04-07 12:33:59
11	Opera Solutions	0.8623	9.47	2009-07-24 00:34:07
12	BellKor	0.8624	9.46	2009-07-26 17:19:11
Progress Prize 2008 - RMSE = 0.8627 - Winning Team: BellKor in BigChaos				

Intuition behind ensemble methods

So what is the intuition behind the success of these ensemble models?

Intuition behind ensemble methods

So what is the intuition behind the success of these ensemble models?

First, there is the general protective mechanism of diversification.

Intuition behind ensemble methods

So what is the intuition behind the success of these ensemble models?

First, there is the general protective mechanism of diversification.

For instance, combining diverse independent opinions in human decision-making.

Intuition behind ensemble methods

So what is the intuition behind the success of these ensemble models?

First, there is the general protective mechanism of diversification.

For instance, combining diverse independent opinions in human decision-making.

Averting risk by diversifying a stock portfolio.

Intuition behind ensemble methods

So what is the intuition behind the success of these ensemble models?

Second, it is generally difficult to establish precisely the type and complexity of model required for a specific learning task.

Intuition behind ensemble methods

So what is the intuition behind the success of these ensemble models?

Second, it is generally difficult to establish precisely the type and complexity of model required for a specific learning task.

A combination of models of diverse types and complexities can alleviate this challenge.

Intuition behind ensemble methods

Here is another, mathematical intuition behind model combination.

Intuition behind ensemble methods

Here is another, mathematical intuition behind model combination.

Suppose we have completely independent classifiers, each having 70% accuracy on a specific task.

Intuition behind ensemble methods

Here is another, mathematical intuition behind model combination.

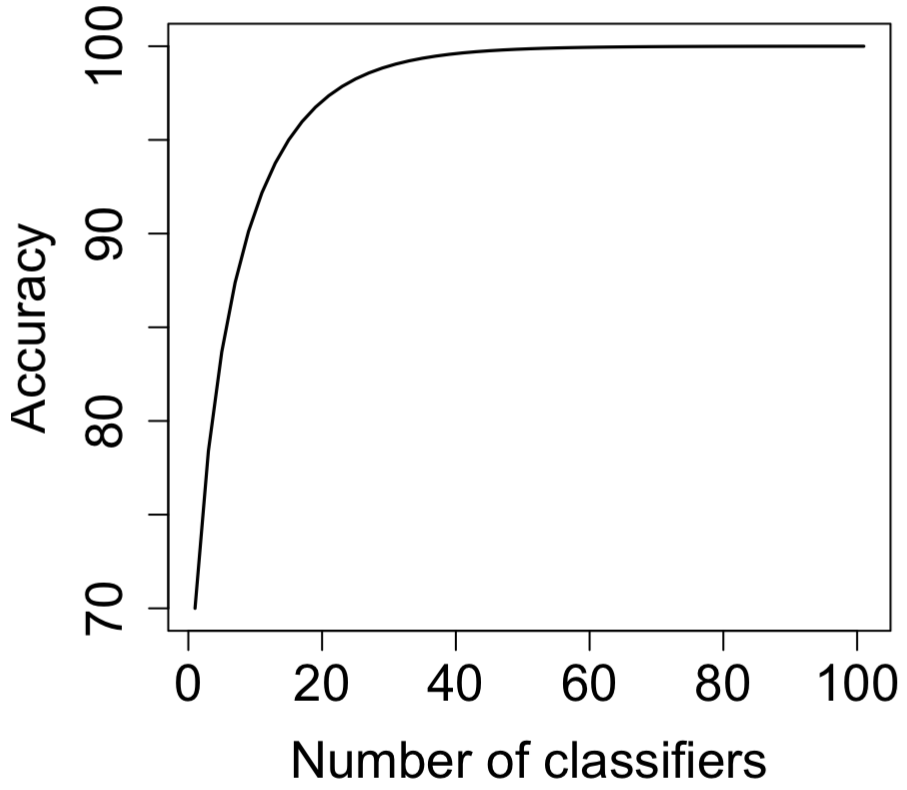
Suppose we have completely independent classifiers, each having 70% accuracy on a specific task.

Now, suppose we combine them using majority vote, where each classifiers predicts a label for an instance, and we make a final prediction based on what the majority of classifiers predicted.

Intuition behind ensemble methods

In this case, the accuracy of the ensemble increases as the number of classifiers increase.

In this case, we would reach 99% accuracy with about 40 classifiers.



Ensemble Models

We will look at three strategies for building ensembles.

Bagging: The goal is to construct diverse models.

Ensemble Models

We will look at three strategies for building ensembles.

Bagging: The goal is to construct diverse models.

Use different samples of instances and/or attributes to independently train diverse classifiers.

Ensemble Models

We will look at three strategies for building ensembles.

Bagging: The goal is to construct diverse models.

Use different samples of instances and/or attributes to independently train diverse classifiers.

Then, aggregate the classifiers using majority vote (classification) or averaging (regression).

Ensemble Models

Stacking: Build ensembles in parallel

Ensemble Models

Stacking: Build ensembles in parallel

Different model types, different features, lots of diversity

Ensemble Models

Stacking: Build ensembles in parallel

Different model types, different features, lots of diversity

Learn how to combine models via a simple blending model

Ensemble Models

Boosting: Build the ensemble sequentially, using the performance of the ensemble to train the next classifier in the ensmeble.

Bagging

The name Bagging comes from combining the *bootstrap* to generate different samples of instances and *aggregation* used to combine predictions.

Bagging

The name Bagging comes from combining the *bootstrap* to generate different samples of instances and *aggregation* used to combine predictions.

This method increases diversity of the weak learners using randomness in two ways

1. using bootstrap sampling to generate the training set for each weak learner
2. using random subsets of features to train each weak learner

Bagging

The general bootstrap algorithm was designed to estimate variability of estimates when we do not have support for a specific data generating model.

Bagging

The general bootstrap procedure is as follows for parameter θ and estimator s:

1. Select B independent bootstrap samples $\mathbf{x}_1^*, \ldots, \mathbf{x}_B^*$ each consisting of N data values drawing *with replacement* from training set $\mathbf{x}$.

2. Evaluate each bootstrap replication to estimate $\hat{\theta}(b) = s(\mathbf{x}_b^*)$, for $b = 1, \ldots, B$.

3. Estimate standard error of $\hat{\theta}$ using the sample standard error of the B $\hat{\theta}$ estimates.

Bagging

In the ensemble learning case, s is an ML training algorithm

Bagging

In the ensemble learning case, s is an ML training algorithm

$\hat{\theta}_b$ is interpreted as the predictions made by the ML algorithm trained using bootstrap sample b.

Bagging

In the ensemble learning case, s is an ML training algorithm

$\hat{\theta}_b$ is interpreted as the predictions made by the ML algorithm trained using bootstrap sample b.

In this case we are not necessarily interested in the inferential task.

Bagging

In the ensemble learning case, s is an ML training algorithm

$\hat{\theta}_b$ is interpreted as the predictions made by the ML algorithm trained using bootstrap sample b.

In this case we are not necessarily interested in the inferential task.

Instead, we use aggregation of these multiple predictions to make final predictions.

Bagging

Bagging works best when perturbing the training data can cause significant changes in the estimated model.

Bagging

Bagging works best when perturbing the training data can cause significant changes in the estimated model.

This is specially notable in decision trees.

Bagging

Bagging works best when perturbing the training data can cause significant changes in the estimated model.

This is specially notable in decision trees.

For some models, like linear regression models, we can show anatically that this strategy decreases variance without increasing bias.

Stacking

A related idea is stacking (also known as blending)

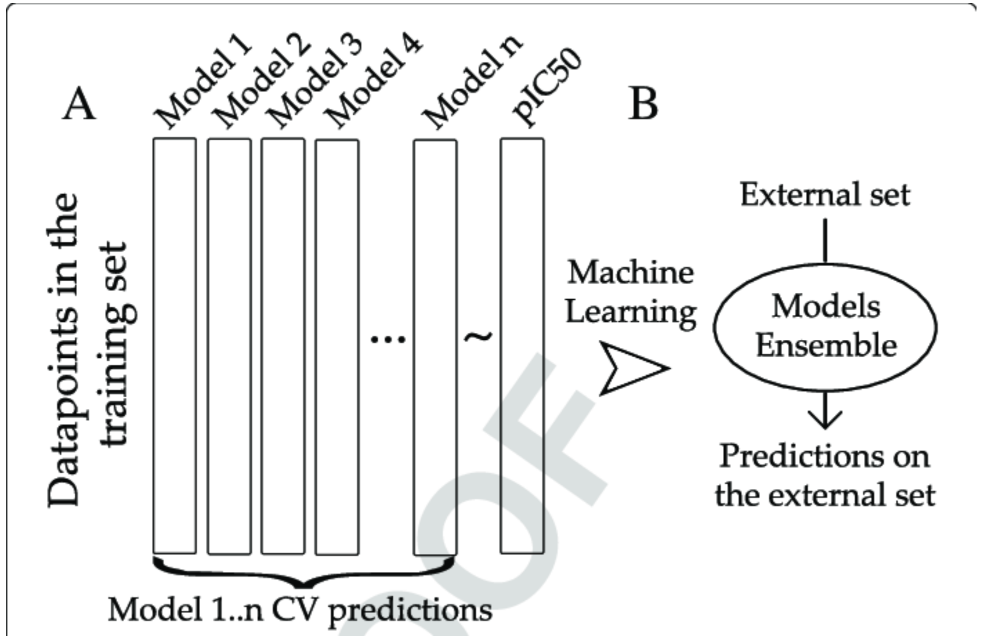
This is what the Netflix teams did as they combined predictors near the end of the challenge.

Stacking

The basic idea is to train a set of diverse predictors as done in bagging with some differences:

Train each model (e.g. SVM, Decision Tree, KNN) on the entire set of training observations (instead of bootstrapped samples)

Train a simple model (e.g., linear regression or logistic regression) using the *predictions* made by the predictors as *features*



Geometric Representation of Classification Problems

Let's consider again little bit how we think of training data. For each instance i:

- predictors (covariates) x_i,
- *qualitative* outcomes (or classes) g_i, which can take values from a discrete set G.

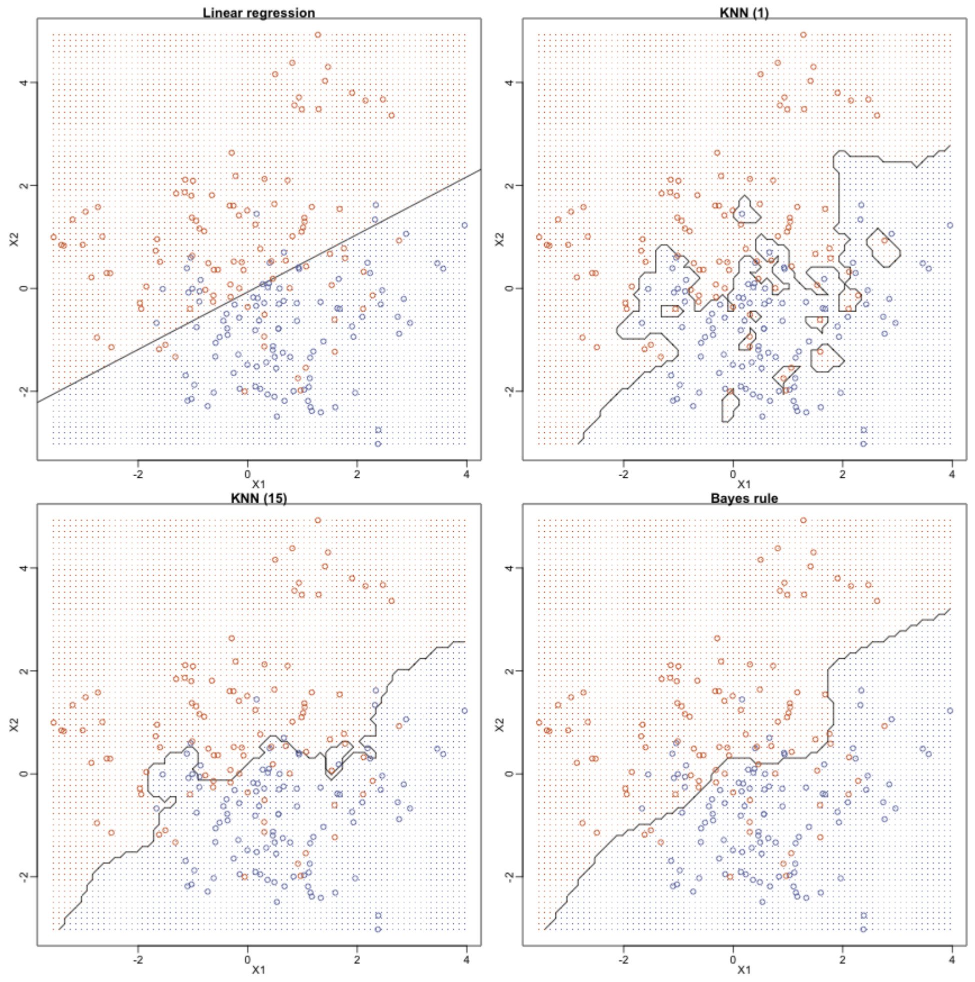
Geometric Representation of Classification Problems

Let's consider again little bit how we think of training data. For each instance i:

- predictors (covariates) x_i,
- *qualitative* outcomes (or classes) g_i, which can take values from a discrete set G.

We can always divide the input space into a collection of regions taking the same predicted values.

Boundaries can be linear or non-linear depending on the *decision* function.



Linear expansions

It will be helpful throughout our discussion to represent classification in terms of decision functions

$$f(\mathbf{x}_i) = \sum_{m=1}^{M} h_m(\mathbf{x}_i)$$

For binary classification a single decision function is needed, the sign of $f(\mathbf{x}_i)$ is used to choose **positive** or **negative**

MARS (multivariate adaptive regression spline)

Make each term a simple form:

$$h_m(\mathbf{x}_i) = \beta_m(x_{im} - t_m)_+$$

(and products of pairs these)

- Still (somewhat interpretable)
- Which terms to add?

  Algorithmic Search

Forward Stagewise Additive Modeling

Algorithm

(1) Initialize $f_0(\mathbf{x}) = 0$

(2) For $m = 1, \ldots, M$:

(a) Compute

$$(\beta_m, t_m) = \arg\min_{\beta,t} \sum_{i=1}^{N} L(y_i, f_{m-1}(x_i) + \beta b(x_i; t))$$

(b) Set $f_m = f_{m-1}(x) + \beta_m b_m(x; t_m)$

Forward Stagewise Additive Modeling

Computing the next term:

Suppose we use *least squares* as the loss function:

$$L(y_i, f_{m-1}(x_i) + \beta b(x_i;t)) = (y_i - (f_{m-1}(x_i) + \beta b(x_i;t)))^2$$

Then we are minimizing

$$\sum_{i=1}^{N} (r_i - \beta b(x_i;t))^2$$

where r_i is the **residual** of the model at stage $m-1$.

Forward Stagewise Additive Modeling

For other loss functions (e.g., negative log binomial likelihood, i.e., logistic regression)

Minimize

$$\sum_{i=1}^{N} (-g_i - \beta b(x_i; t))^2$$

where g_i is the *gradient* of the loss function for instance i

$$g_i = \left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \right]_{f(x_i) = f_{m-1}(x_i)}$$

Boosting

Create an ensemble of classifiers sequentially.

At each iteration of the sequence we modify the loss function so that instances that are incorrectly classified by the current set of classifiers are given higher weight.

Boosting

Create an ensemble of classifiers sequentially.

At each iteration of the sequence we modify the loss function so that instances that are incorrectly classified by the current set of classifiers are given higher weight.

Here diversity is injected into the ensemble by sequentially "re-weighting" training instances.

Boosting

Final prediction from the ensemble is also given by aggregation (majority decision or averaging)

Boosting

Final prediction from the ensemble is also given by aggregation (majority decision or averaging)

Predictions are weighted based on each classifiers accuracy.

Adaboost

The Adaboost algorithm is probably the best known example of boosting. The algorithm is as follows:

- Initialize weights: each instance gets the same weight

$$w_i = 1/N, i = 1, \ldots, N$$

- Construct a classifier m using current weights (Decision Trees, SVM, linear/logistic regression). Compute the error of the new classifier

$$\epsilon_m = \frac{\sum_i w_i I y_i \neq g_m(x_i)}{\sum_i w_i}$$

Adaboost

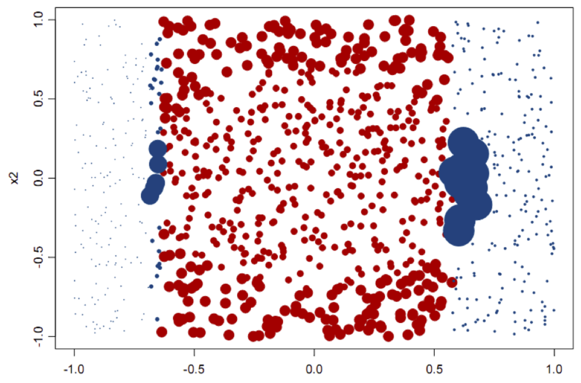
- Get the *influence* of the new classifier and update training instance weights

$$\alpha_m = \log\left(\frac{1-\epsilon_i}{\epsilon_i}\right)$$
$$w_i \leftarrow w_i \exp \alpha_m I y_i \neq g_m(x_i)$$

- Goto step 2

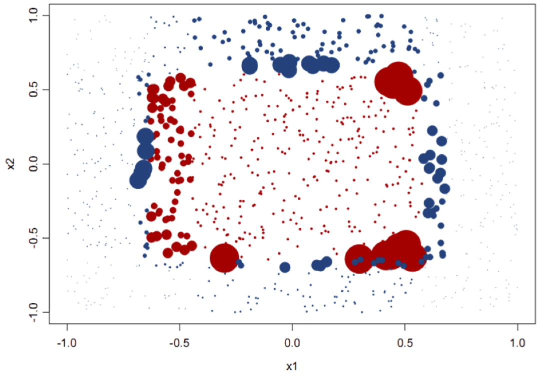
For final prediction, average predictions from each classifier with weight α_m.

Adaboost

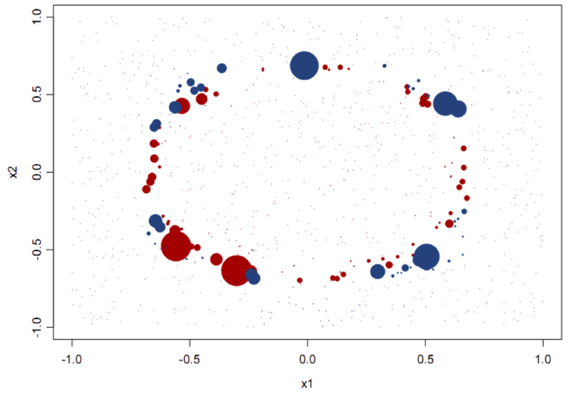


Classifications (colors) and Weights (size) after *1 iteration* Of AdaBoost

3 iterations



20 iterations



from Elder, John. From Trees to Forests and Rule Sets - A Unified Overview of Ensemble Methods. 2007.

Adaboost

Adaboost has the advantage of its simplicity

Adaboost

Adaboost has the advantage of its simplicity

Like bagging, adaboost reduces the variability of individual weak learners.

Adaboost

Adaboost has the advantage of its simplicity

Like bagging, adaboost reduces the variability of individual weak learners.

Unlike bagging it is sensitive to noise and outliers.

Gradient Boosted Trees

Uses the sequential additive modeling idea we saw previously. At each iteration learn a regression tree $T(x)$ to minimize

$$\sum_{i=1}^{N} (-g_i - T(x_i))^2$$

Very close to Adaboost if loss is "exponential": $L(y_i, f_i) = \exp\{-y_i f_i\}$.

Not a robust loss function to use, prefer binomial deviance (i.e., logistic regression negative log likelihood)

Gradient Boosted Trees

Excellent software XGBoost: most commonly generally

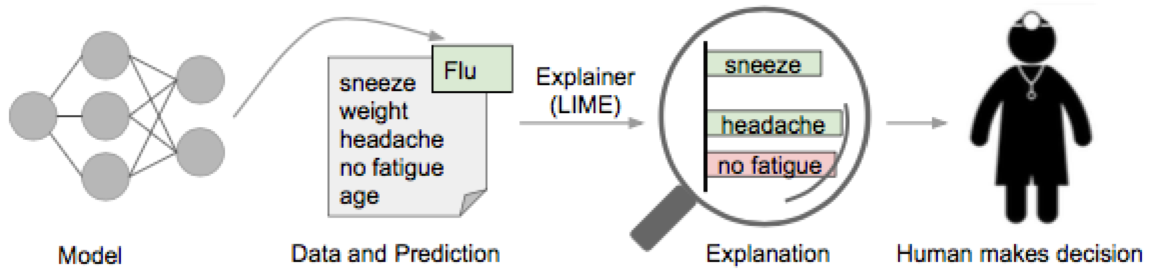
Actually minimizes a regularized loss function: no need to prune trees

Can still compute variable importance:

- compute variable importance for each tree
- add variable importance across trees

No simple interpretation by inspection, but there are options...

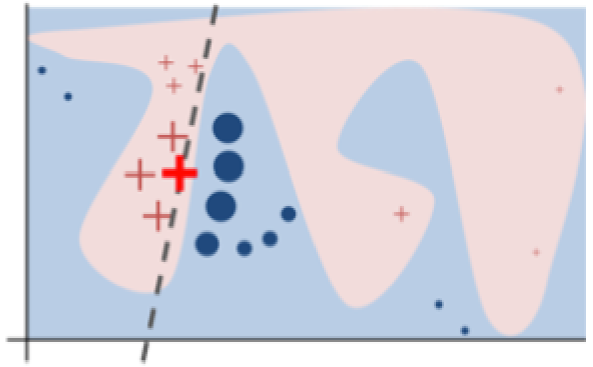
Model explanation



- Surrogate models
- Local explanatory models

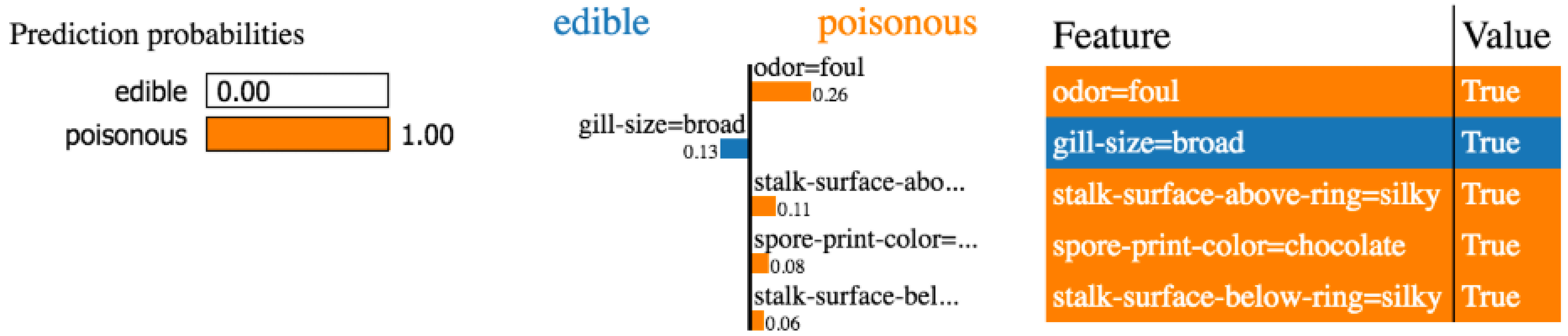
Model explanation

Use a *sparse* simple model (e.g., logistic regression) to explain the prediction of a complicated model for a specific instance i



LIME:
https://github.com/marcotcr/lime
ELI5:
http://eli5.readthedocs.io/en/latest/index.html

Model explanation



Feature	Value
odor=foul	True
gill-size=broad	True
stalk-surface-above-ring=silky	True
spore-print-color=chocolate	True
stalk-surface-below-ring=silky	True

LIME: https://github.com/marcotcr/lime

ELI5: http://eli5.readthedocs.io/en/latest/index.html

Summary

Ensemble methods seek to

1. reduce variance of individual weak learners by aggregating their predictions.

2. improve performance by exploiting prediction diversity

Summary

Ensemble methods seek to

1. reduce variance of individual weak learners by aggregating their predictions.

2. improve performance by exploiting prediction diversity

Bagging, boosting and stacking are alternative methods of training ensembles while providing diversity to each of the trained weak learners.