Ensemble Methods

Outline

- The Netflix Prize
  - Success of ensemble methods in the Netflix Prize
- Why Ensemble Methods Work
- Algorithms
  - Bagging
  - Random forests
  - AdaBoost

Definition

Ensemble Classification

Aggregation of predictions of multiple classifiers with the goal of improving accuracy.

Teaser: How good are ensemble methods?

*Let's look at the Netflix Prize Competition…*

Netflix Prize *Began October 2006*

- Supervised learning task
  - Training data is a set of users and ratings (1,2,3,4,5 stars) those users have given to movies.
  - Construct a classifier that given a user and an unrated movie, correctly classifies that movie as either 1, 2, 3, 4, or 5 stars

- $1 million prize for a 10% improvement over Netflix's current movie recommender/classifier (MSE = 0.9514)

Just three weeks after it began, at least 40 teams had bested the Netflix classifier.
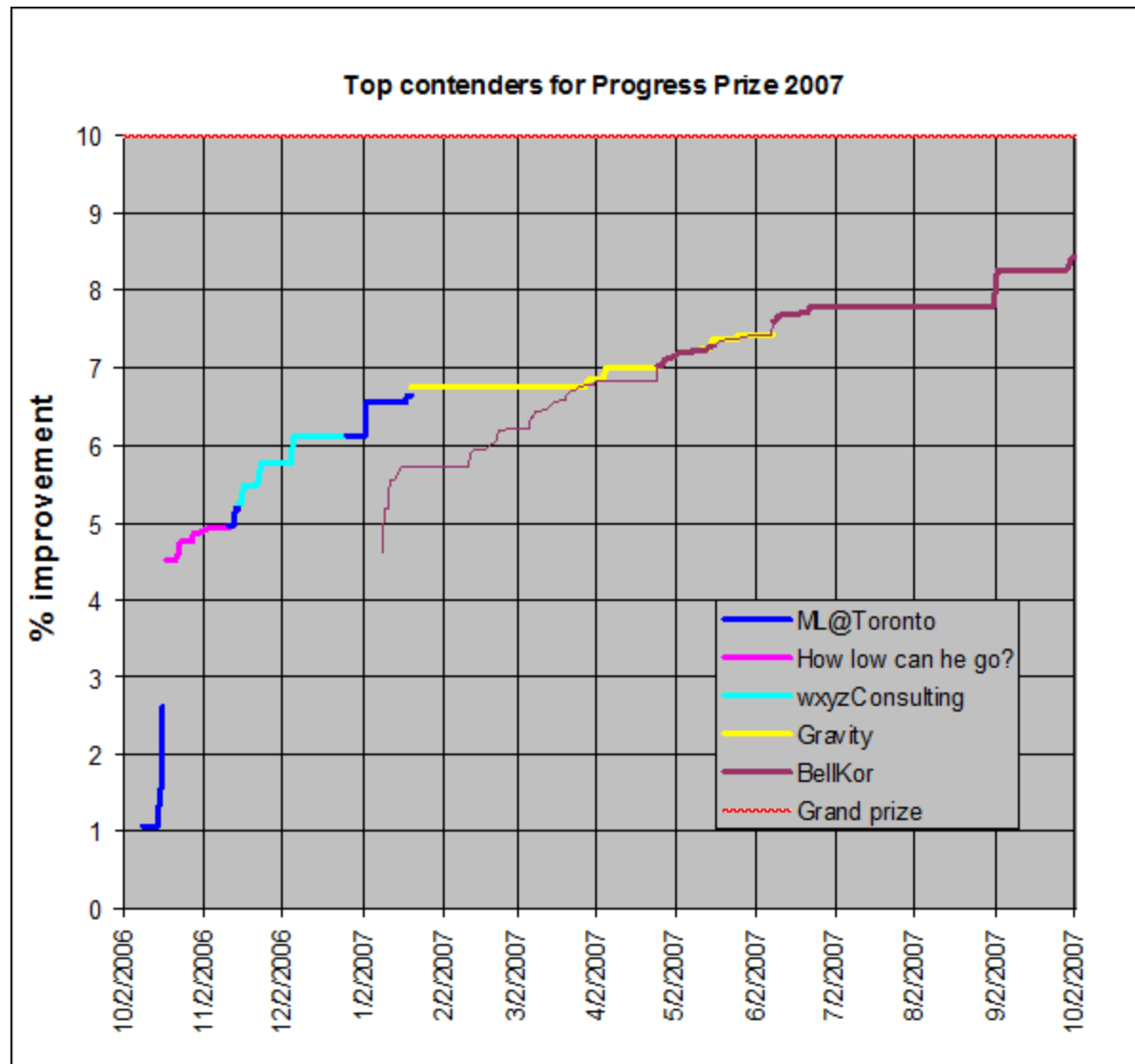
Top teams showed about 5% improvement.

Netflix Prize

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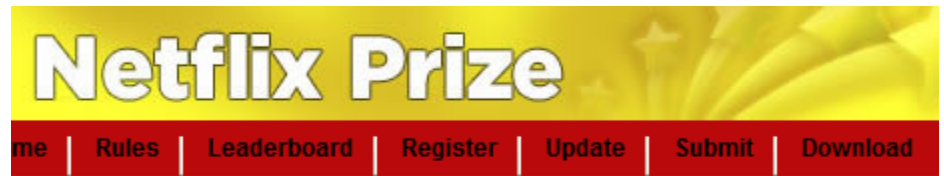
Leaderboard

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
Elliptic Chaos	0.9179	3.52
datcracker	0.9183	3.48
Foreseer	0.9214	3.15
bsdfish	0.9229	3.00
Three Blind Mice	0.9234	2.94
Bocsimacko	0.9238	2.90
Remco	0.9252	2.75
karmatics	0.9301	2.24
Chapelator	0.9314	2.10
Flmod	0.9325	1.99
mthrox	0.9328	1.96

However, improvement slowed…

Top contenders for Progress Prize 2007

Legend:
- ML@Toronto
- How low can he go?
- wxyzConsulting
- Gravity
- BellKor
- Grand prize

from http://www.research.att.com/~volinsky/netflix/

Today, the top team
has posted
a 8.5% improvement.

**Ensemble methods
are the best
performers…**

--	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
21	Rookies	0.8872	6.75
22	Paul Harrison	0.8872	6.75
23	ATTEAM	0.8873	6.74
24	wxyzconsulting.com	0.8874	6.73
25	ICMLsubmission	0.8875	6.72
26	Efratko	0.8877	6.70
27	Kitty	0.8881	6.65
28	SecondaryResults	0.8884	6.62
29	Birgit Kraft	0.8885	6.61

Rookies

"Thanks to Paul Harrison's collaboration, a simple mix of our solutions improved our result from 6.31 to 6.75"

--	**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
21	Rookies	0.8872	6.75
22	Paul Harrison	0.8872	6.75
23	ATTEAM	0.8873	6.74
24	wxyzconsulting.com	0.8874	6.73
25	ICMLsubmission	0.8875	6.72
26	Efratko	0.8877	6.70
27	Kitty	0.8881	6.65
28	SecondaryResults	0.8884	6.62
29	Birgit Kraft	0.8885	6.61

Arek Paterek

"My approach is to **combine the results of many methods** (also two-way interactions between them) using linear regression on the test set. The best method in my ensemble is regularized SVD with biases, post processed with kernel ridge regression"

http://rainbow.mimuw.edu.pl/~ap/ap_kdd.pdf

--	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
21	Rookies	0.8872	6.75
22	Paul Harrison	0.8872	6.75
23	ATTEAM	0.8873	6.74
24	wxyzconsulting.com	0.8874	6.73
25	ICMLsubmission	0.8875	6.72
26	Efratko	0.8877	6.70
27	Kitty	0.8881	6.65
28	SecondaryResults	0.8884	6.62
29	Birgit Kraft	0.8885	6.61

U of Toronto

"When the predictions of **multiple** RBM models and **multiple** SVD models are linearly combined, we achieve an error rate that is well over 6% better than the score of Netflix's own system."

http://www.cs.toronto.edu/~rsalakhu/papers/rbmcf.pdf

--	**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
21	Rookies	0.8872	6.75
22	Paul Harrison	0.8872	6.75
23	ATTEAM	0.8873	6.74
24	wxyzconsulting.com	0.8874	6.73
25	ICMLsubmission	0.8875	6.72
26	Efratko	0.8877	6.70
27	Kitty	0.8881	6.65
28	SecondaryResults	0.8884	6.62
29	Birgit Kraft	0.8885	6.61

Gravity

Table 5: Best results of single approaches and their combinations

Method/Combination	RMSE
MF	0.9190
NB	0.9313
CL	0.9606
NB + CL	0.9275
MF + CL	0.9137
MF + NB	0.9089
MF + NB + CL	0.9089

home.mit.bme.hu/~gtakacs/download/**gravity**.pdf

--	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
21	Rookies	0.8872	6.75
22	Paul Harrison	0.8872	6.75
23	ATTEAM	0.8873	6.74
24	wxyzconsulting.com	0.8874	6.73
25	ICMLsubmission	0.8875	6.72
26	Efratko	0.8877	6.70
27	Kitty	0.8881	6.65
28	SecondaryResults	0.8884	6.62
29	Birgit Kraft	0.8885	6.61

When Gravity and Dinosaurs Unite

"Our common team blends the result of team Gravity and team Dinosaur Planet."

Might have guessed from the name...

--	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
21	Rookies	0.8872	6.75
22	Paul Harrison	0.8872	6.75
23	ATTEAM	0.8873	6.74
24	wxyzconsulting.com	0.8874	6.73
25	ICMLsubmission	0.8875	6.72
26	Efratko	0.8877	6.70
27	Kitty	0.8881	6.65
28	SecondaryResults	0.8884	6.62
29	Birgit Kraft	0.8885	6.61

BellKor / KorBell

And, yes, the top team which is from AT&T…

"Our final solution (RMSE=0.8712) consists of blending 107 individual results. "

--	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
21	Rookies	0.8872	6.75
22	Paul Harrison	0.8872	6.75
23	ATTEAM	0.8873	6.74
24	wxyzconsulting.com	0.8874	6.73
25	ICMLsubmission	0.8875	6.72
26	Efratko	0.8877	6.70
27	Kitty	0.8881	6.65
28	SecondaryResults	0.8884	6.62
29	Birgit Kraft	0.8885	6.61

Leaderboard

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				

The winner was an ensemble of ensembles (including BellKor).

Some Intuitions on Why Ensemble Methods Work...

Intuitions

- Utility of combining diverse, independent opinions in human decision-making
  - Protective Mechanism (e.g. stock portfolio diversity)

- **Violation of Ockham's Razor**
  - Identifying the best model requires identifying the proper "model complexity"

    See Domingos, P. Occam's two razors: the sharp and the blunt. KDD. 1998.

Intuitions

Majority vote

Suppose we have 5 completely independent classifiers...

- If accuracy is 70% for each
  - $10 (.7^3)(.3^2)+5(.7^4)(.3)+(.7^5)$
  - **83.7% majority vote accuracy**
- 101 such classifiers
  - **99.9% majority vote accuracy**

Strategies

Bagging

– Use different samples of observations and/or predictors (features) of the examples to generate diverse classifiers

– Aggregate classifiers: average in regression, majority vote in classification

Boosting

– Make examples currently misclassified more important (or less, in some cases)

Bagging (Constructing for Diversity)

1. Use random samples of the examples to construct the classifiers

2. Use random feature sets to construct the classifiers
   - Random Decision Forests

- Bagging: **B**ootstrap **Agg**regation

Leo Breiman

- Bootstrap: consider the following situation:
  - A random sample $\mathbf{x} = (x_1, \ldots, x_N)$ from unknown probability distribution *F*
  - We wish to estimate parameter $\theta = t(F)$
  - We build estimate $\hat{\theta} = s(\mathbf{x})$
  - What is the s.d. of $\hat{\theta}$?

Examples:
  1) estimate mean and sd of *expected prediction error*

  2) estimate point-wise confidence bands in smoothing
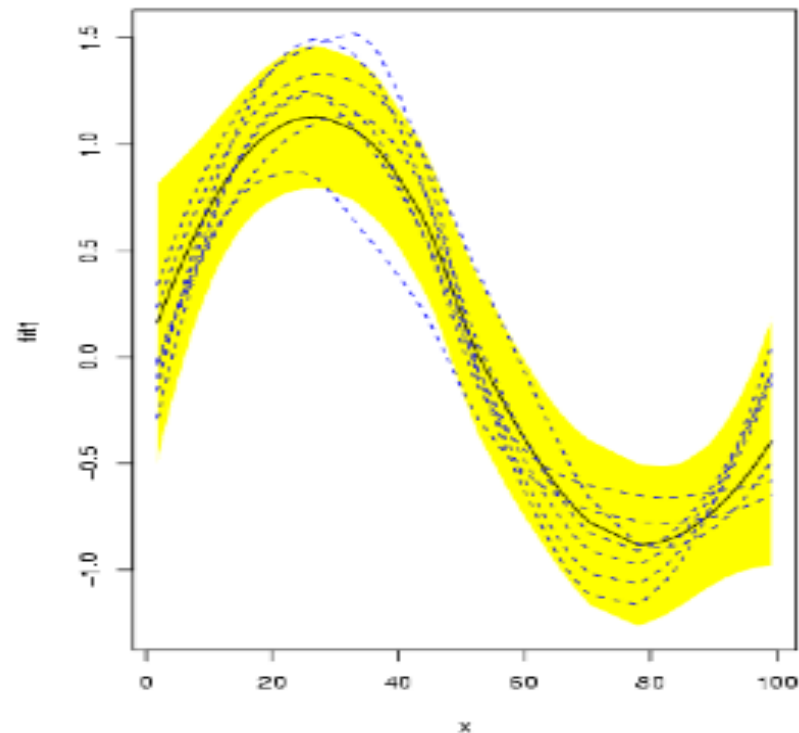
- Bootstrap:
  - It is completely automatic
  - Requires no theoretical calculations
  - Not based on asymptotic results
  - Available regardless of how complicated the estimator $\hat{\theta}$

- Bootstrap algorithm:

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

  2. Evaluate the bootstrap replication corresponding to each bootstrap sample

  $$\hat{\theta}^*(b) = s(\mathbf{x}_b^*), \; b = 1, \ldots, B$$

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

- Bagging: use bootstrap to improve *predictions*
  1. Create bootstrap samples, estimate model from each bootstrap sample
  2. Aggregate predictions (average if regression, majority vote if classification)
- This works best when perturbing the training set can cause significant changes in the estimated model
- For instance, for least-squares, can show variance is decreased while bias is unchanged

Random forests

- At every level, choose a random subset of the variables (predictors, not examples) and choose the best split among those attributes

Random forests

- Let the number of training points be *M*, and the number of variables in the classifier be *N*.

For each tree,

1. Choose a training set by choosing *N* times with replacement from all *N* available training cases.

2. For each node, randomly choose *m* variables on which to base the decision at that node.

Calculate the best split based on these.

Random forests

- Grow each tree as deep as possible *no pruning!*

- *Out-of-bag* data can be used to estimate cross-validation error

  – For each training point, get prediction from averaging trees where point is not included in bootstrap sample

- *Variable importance* measures are easy to calculate

Gini

Randomization

Variable Importance
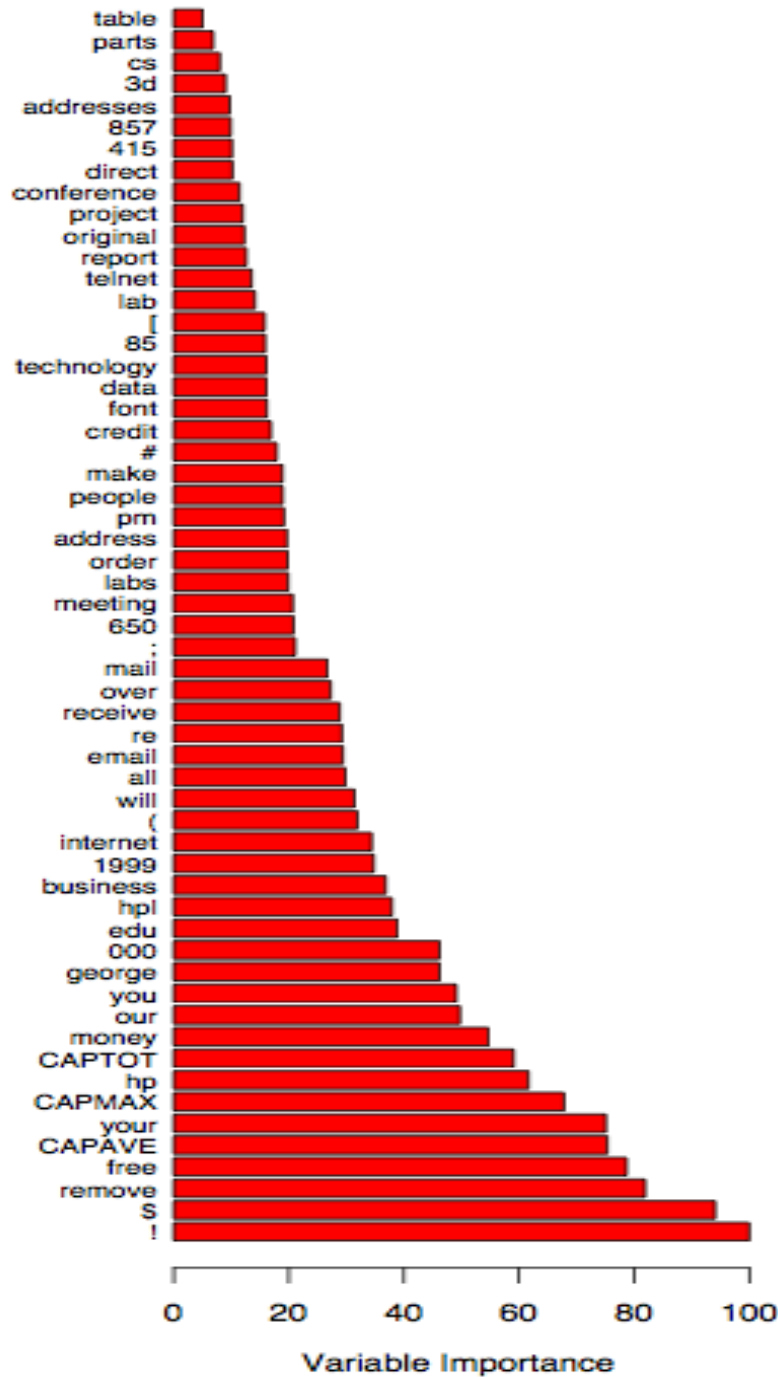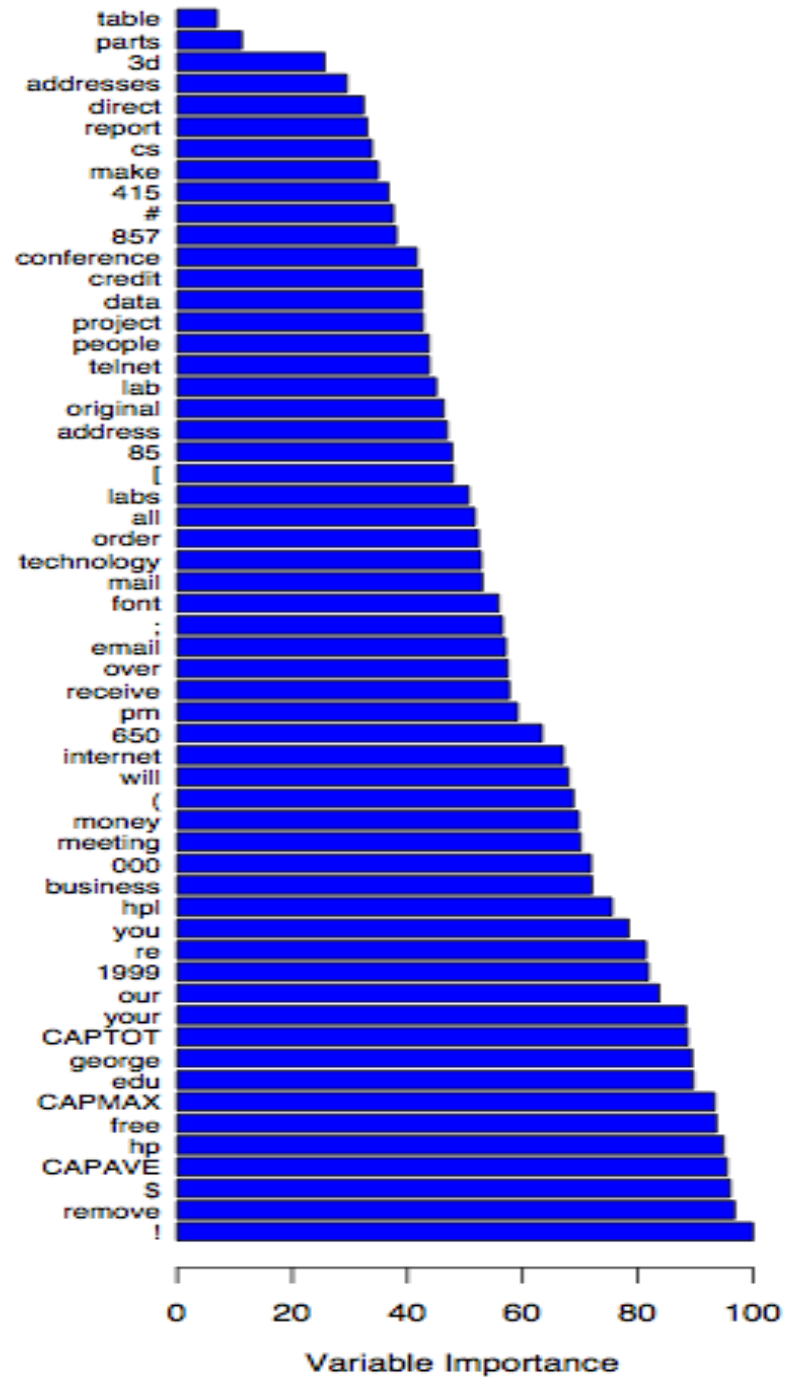
Boosting

1. Create a sequence of classifiers, giving higher influence to more accurate classifiers
2. At each iteration, make examples currently misclassified more important (get larger weight in the construction of the next classifier).
3. Then combine classifiers by weighted vote (weight given by classifier accuracy)

AdaBoost Algorithm

1. Initialize Weights: each case gets the same weight:

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

2. Construct a classifier using current weights. Compute its error:

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

3. Get classifier *influence,* and update example weights

$$\alpha_m = \log\left(\frac{1 - \varepsilon_m}{\varepsilon_m}\right) \quad w_i \leftarrow w_i \times \exp\{\alpha_m I\{y_i \neq g_m(x_i)\}\}$$
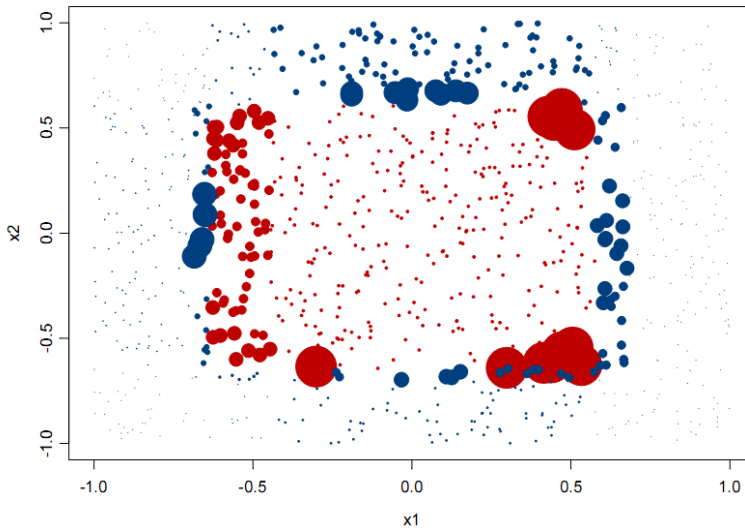
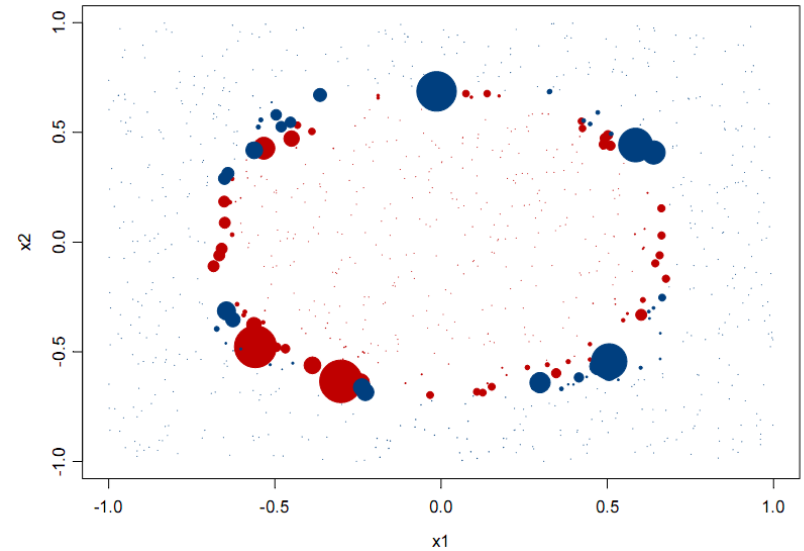4. Goto step 2…

Final prediction is weighted vote, with weight α_m
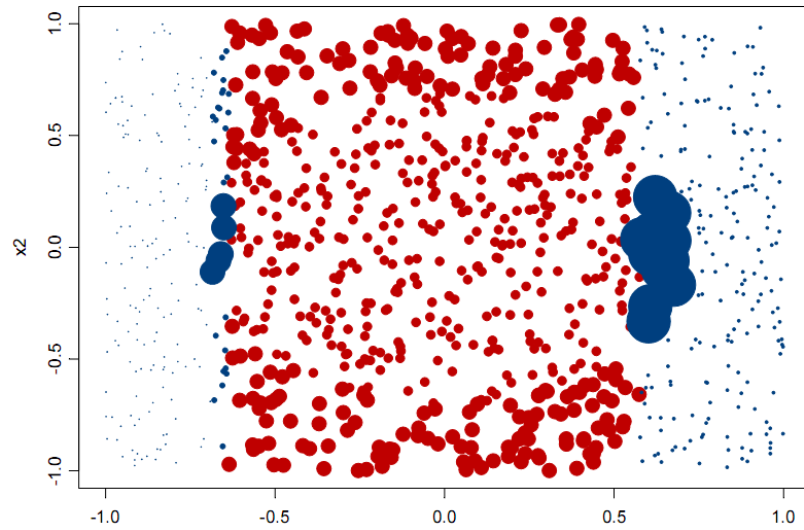
Classifications (colors) and
Weights (size) after *1 iteration*
Of AdaBoost

3 iterations

20 iterations

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

AdaBoost

- Advantages
  - Very little code
  - Reduces variance

- Disadvantages
  - Sensitive to noise and outliers. Why?

How was the Netflix prize won?

Gradient boosted decision trees…

Details: http://www.netflixprize.com/assets/GrandPrize2009_BPC_BellKor.pdf

Sources

- David Mease. Statistical Aspects of Data Mining. Lecture. http://video.google.com/videoplay?docid=-4669216290304603251&q=stats +202+engEDU&total=13&start=0&num=10&so=0&type=search&plindex=8

- Dietterich, T. G. Ensemble Learning. In The Handbook of Brain Theory and Neural Networks, Second edition, (M.A. Arbib, Ed.), Cambridge, MA: The MIT Press, 2002. http://www.cs.orst.edu/~tgd/publications/hbtnn-ensemble-learning.ps.gz

- Elder, John and Seni Giovanni. From Trees to Forests and Rule Sets - A Unified Overview of Ensemble Methods. KDD 2007 http://Tutorial. videolectures.net/kdd07_elder_ftfr/

- Netflix Prize. http://www.netflixprize.com/

- Christopher M. Bishop. Neural Networks for Pattern Recognition. Oxford University Press. 1995.