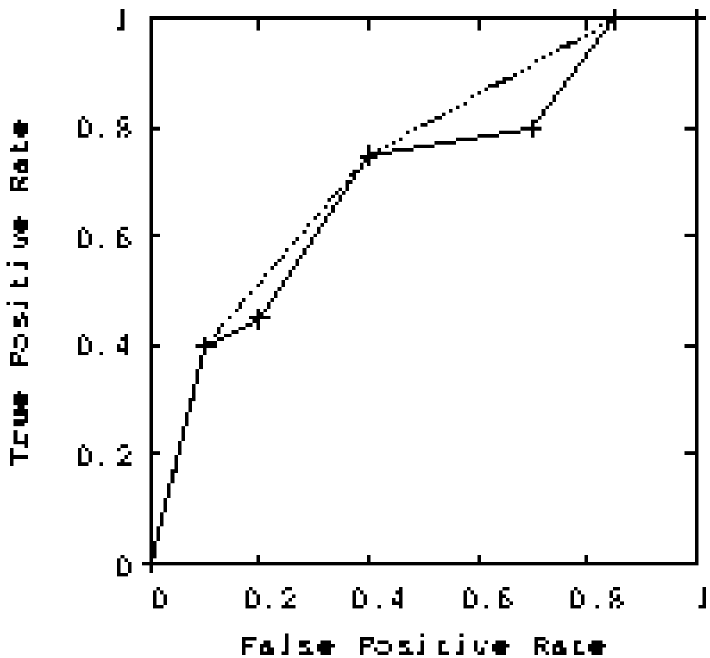
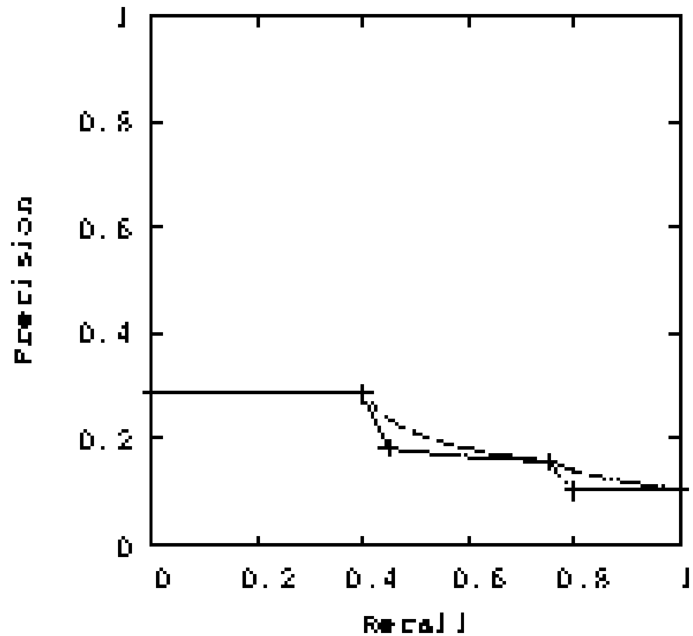
In Precision/Recall graphs, linear interpolation of classifiers does *not* correspond to linear interpolation of points in the plot.



ROC convex hull



Translation to P/R curve

Disadvantage of ROC

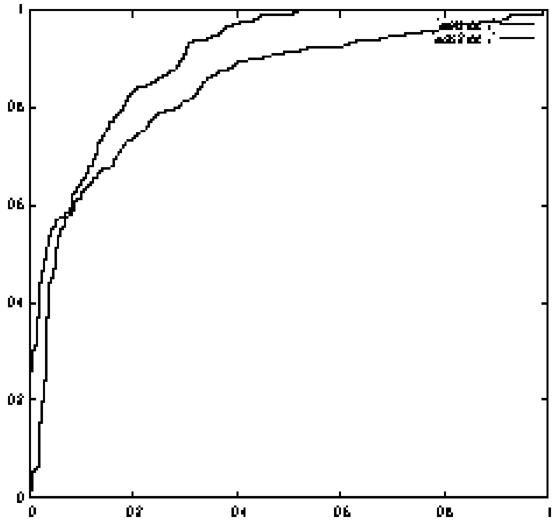
- If the TNR is high, any system can easily achive good FPR or ER by biasing towards the negative class.
- High TNR problems are typically those where one tries to pick out a few interesting points against a large background class (e.g. face detection).

Example

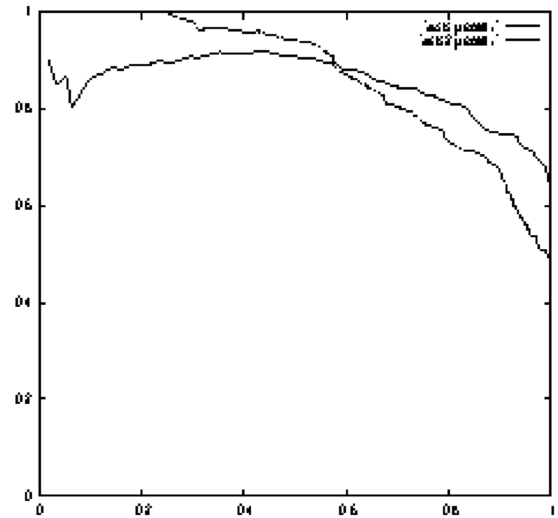
- Two classes are given. Increase the size of the negative class by a factor 10.
- The TP value of a given classifier and # Positives in training data do not depend on the negative class, so the TPR does not change.
- Since FP increases roughly by a factor ten, the FPR does not change either:

$$\mathrm{FPR}_{\mathrm{new}} \approx \frac{10 \cdot \mathrm{FP}_{\mathrm{old}}}{10 \cdot \#\,\mathrm{Negatives}_{\mathrm{old}}} = \mathrm{FPR}_{\mathrm{old}}$$

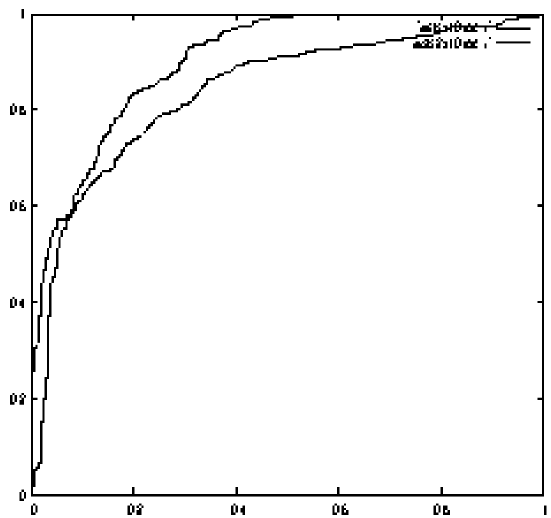
- Consequence: The ROC curve does not change, up to small fluctuations.



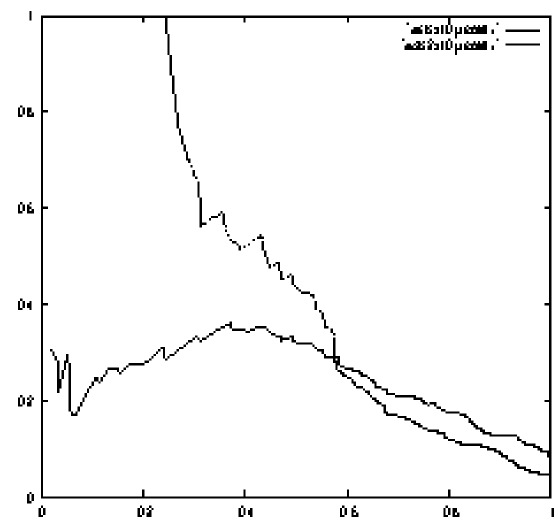
ROC (original classes)



P/R (original classes)



ROC (negative class $\times$ 10)



P/R (negative class $\times$ 10)

Parametrization by a threshold τ

- Many classifiers we have seen can be written as comparing a function g to a threshold τ.
- The classification result $f(\mathbf{x})$ is then computed as

$$f(\mathbf{x}) = \begin{cases} +1 & g(\mathbf{x}) \geq \tau \\ -1 & g(\mathbf{x}) < \tau \end{cases}$$

For example

f	$g(\mathbf{x})$	τ
linear classifier	$\langle \mathbf{v}, \mathbf{x} \rangle - c$	$\tau = 0$
logistic regression	$\sigma(\langle \mathbf{v}, \mathbf{x} \rangle - c)$	$\tau = \frac{1}{2}$
one gaussian density p per class	$p_{+1}(\mathbf{x}) - p_{-1}(\mathbf{x})$	$\tau = 0$

Varying τ

- We can denote the classifier f above as f_τ for a given value of τ, and vary that value.
- As τ changes, the values of TP, FN, etc change.
- For a larger value of τ, fewer points are classified as positive, so we expect fewer false positives and more false negatives.
- If we regard τ as the parameter θ above, we can draw a ROC curve or Precision/Recall diagram for f, where each point correspond to a value of τ.

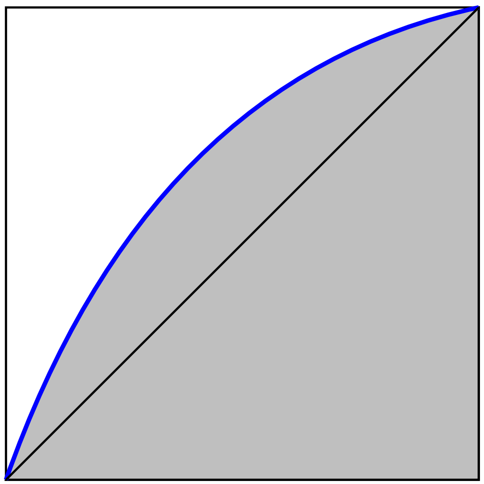
If you see a ROC or P/R curve reported for a single classifier, this is usually what it means.

AUC

Definition

The **Area Under the Curve** (AUC) the area under an ROC curve. Note this is a value between 0 and 1.

Illustration

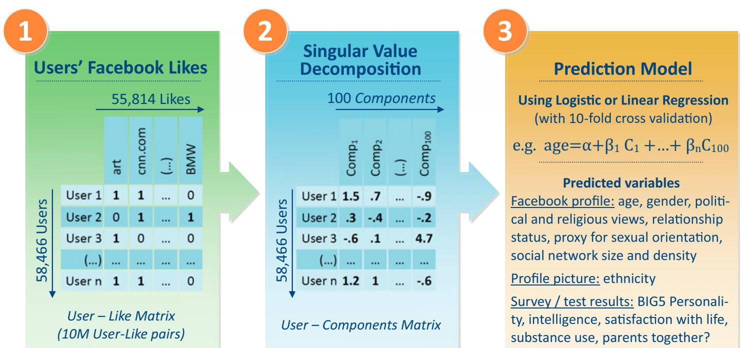


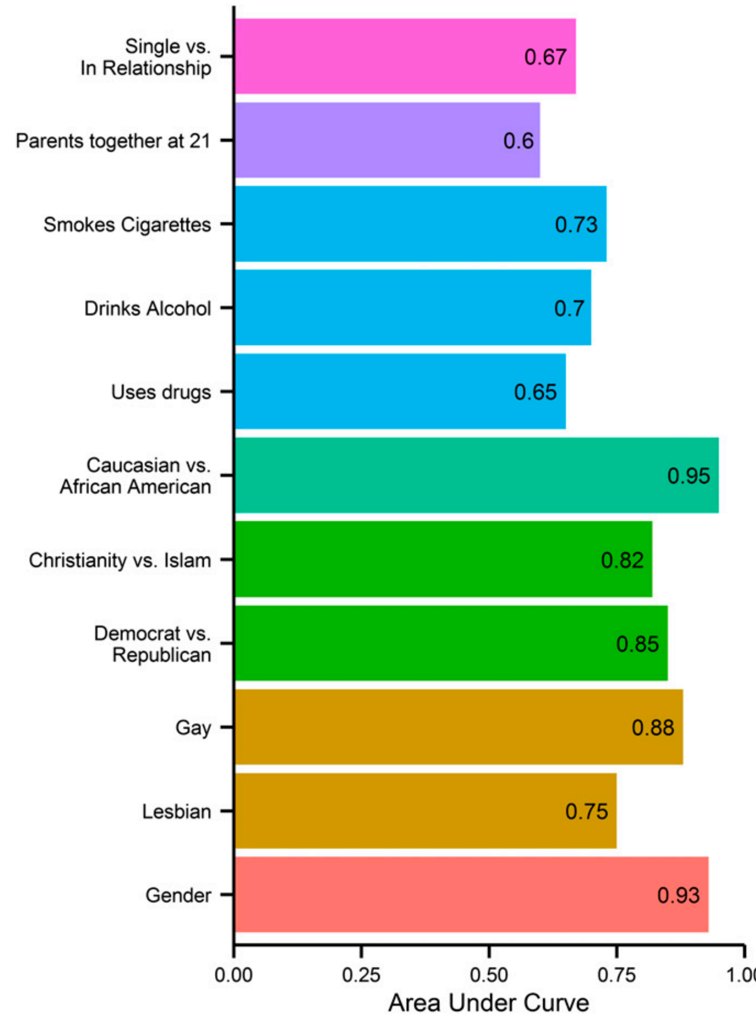
- The blue curve is an ROC curve.
- The AUC value is the size of the area shaded in gray.
- AUC is a summary statistic that summarizes a ROC diagram in a single number.

AUC of a classifier

When AUC is reported for a single classifier, it typically refers to the AUC defined by the ROC diagram obtained by varying a threshold τ as above.

EXAMPLE



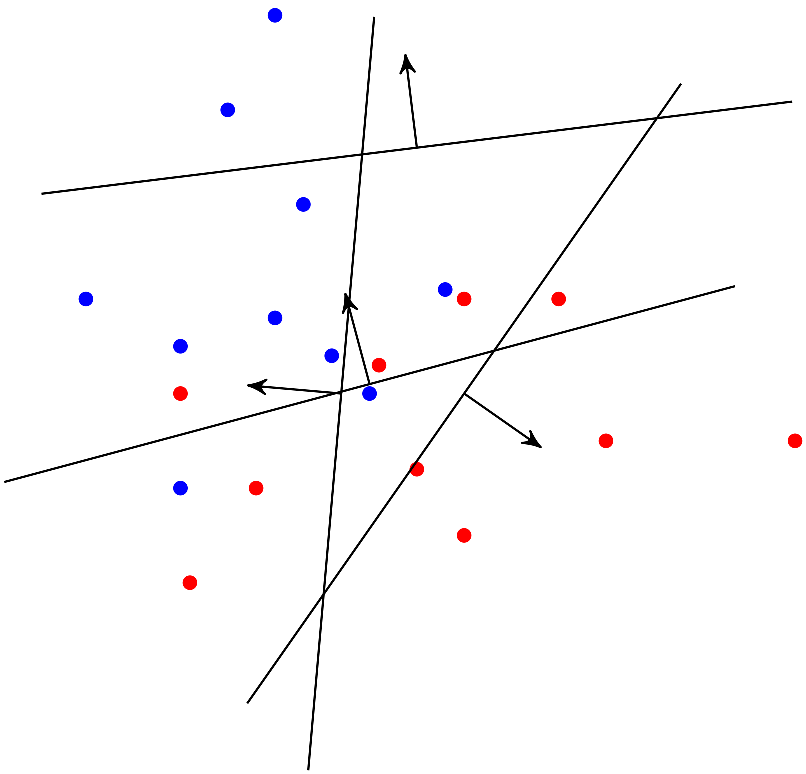


From: Kosinski, Stillwell, Graepel: PNAS, Vol 110, No 15, 2013.

BOOSTING

Ensembles

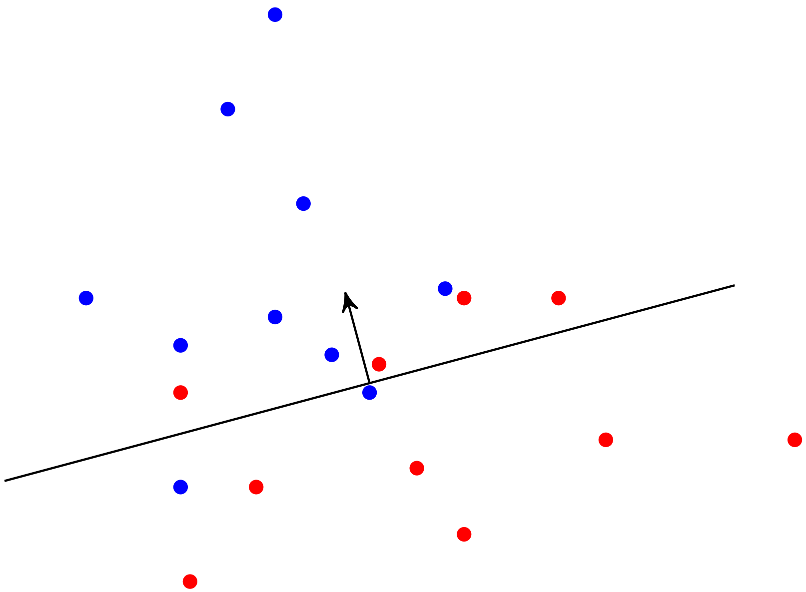
Suppose we are given a data source with two classes, and manage to generate a *random* hyperplane classifier with *expected* error of 0.5 (i.e. 50%).



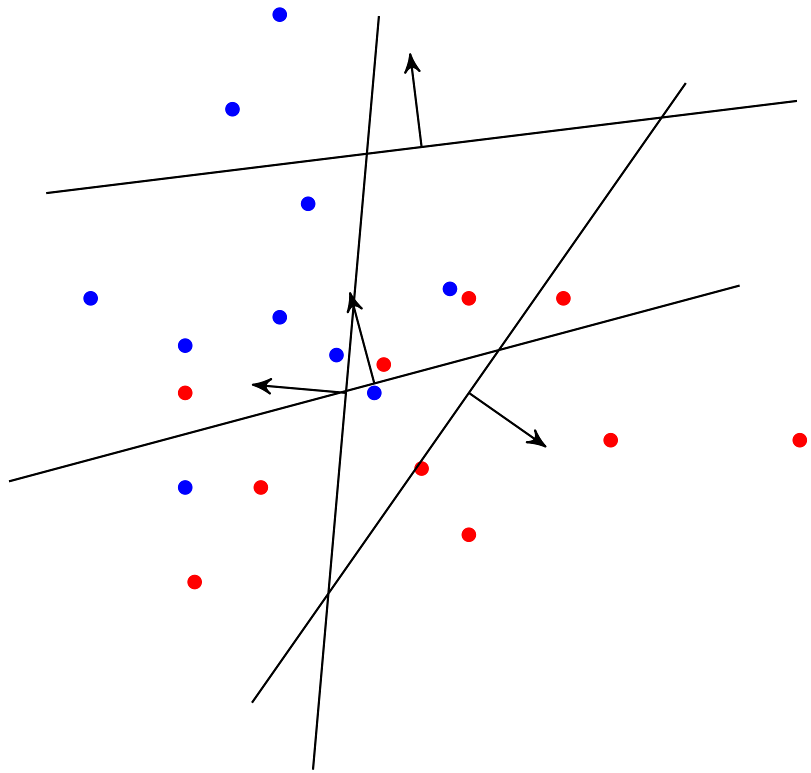
(Informally, think of this as not knowing the data source and generating a "uniformly distributed classifier".)

ENSEMBLES

A *randomly* chosen hyperplane classifier has an *expected* error of 0.5 (i.e. 50%).

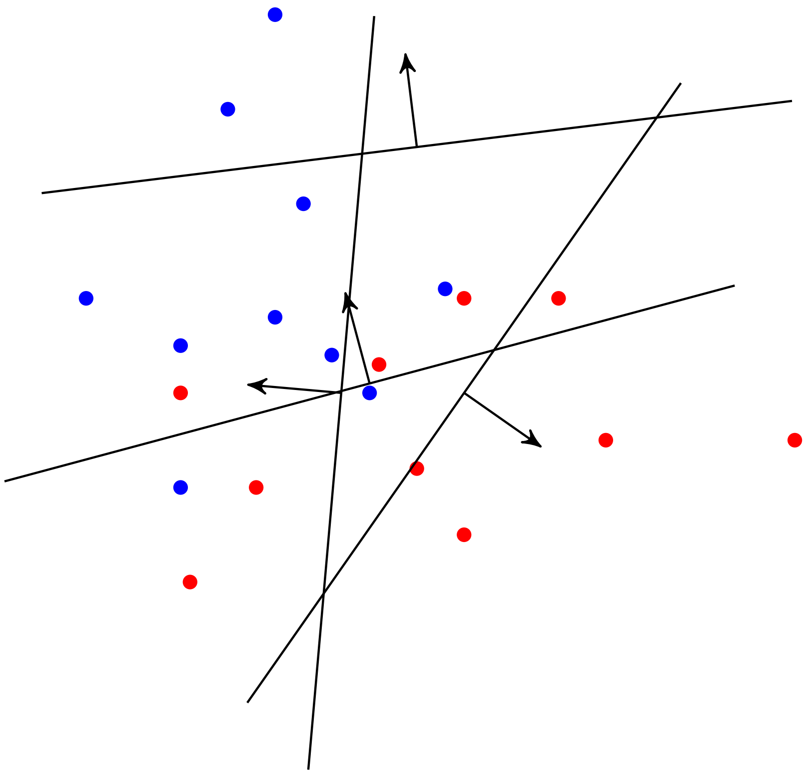


A *randomly* chosen hyperplane classifier has an *expected* error of 0.5 (i.e. 50%).



ENSEMBLES

A *randomly* chosen hyperplane classifier has an *expected* error of 0.5 (i.e. 50%).



- Many random hyperplanes combined by majority vote: Still 0.5.
- A single classifier slightly better than random: $0.5 + \varepsilon$.
- What if we use m such classifiers and take a majority vote?

Decision by majority vote

- m individuals (or classifiers) take a vote. m is an odd number.
- They decide between two choices; one is correct, one is wrong.
- After everyone has voted, a decision is made by simple majority.

Note: For two-class classifiers $f_1, \ldots, f_m$ (with output ± 1):

$$\text{majority vote} = \operatorname{sgn}\Big(\sum_{j=1}^{m} f_j\Big)$$

Assumptions

Before we discuss ensembles, we try to convince ourselves that voting can be beneficial. We make some simplifying assumptions:

- Each individual makes the right choice with probability $p \in [0, 1]$.
- The votes are *independent*, i.e. stochastically independent when regarded as random outcomes.

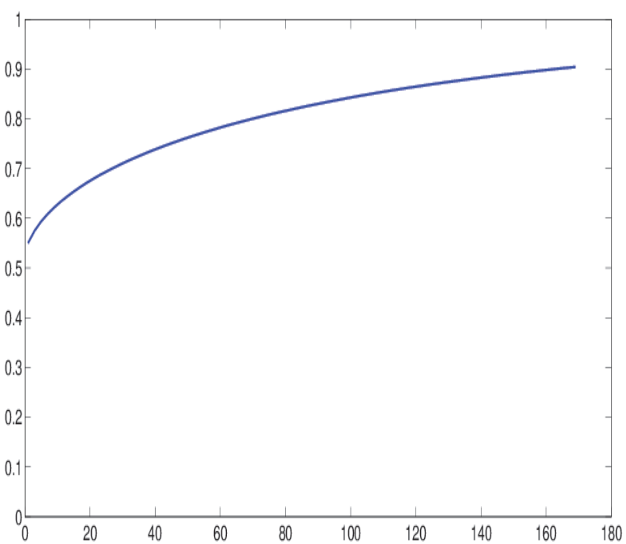
Condorcet's rule

If the individual votes are independent, the answer is

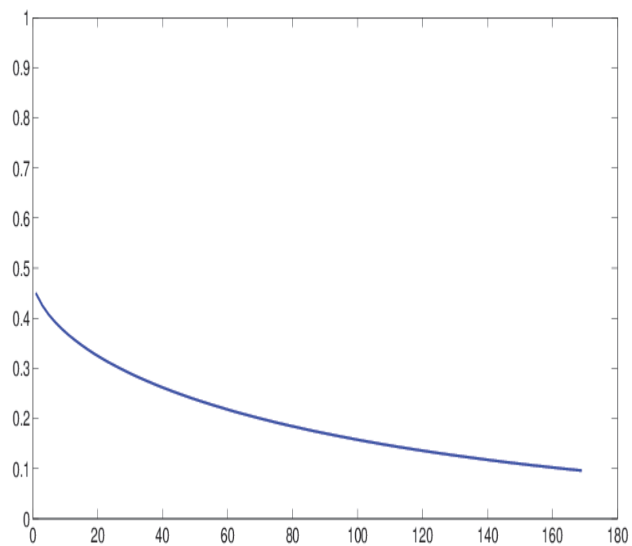
$$\Pr\{\text{ majority makes correct decision }\} = \sum_{j=\frac{m+1}{2}}^{m} \frac{m!}{j!(m-j)!} p^j (1-p)^{m-j}$$

This formula is known as **Condorcet's jury theorem**.

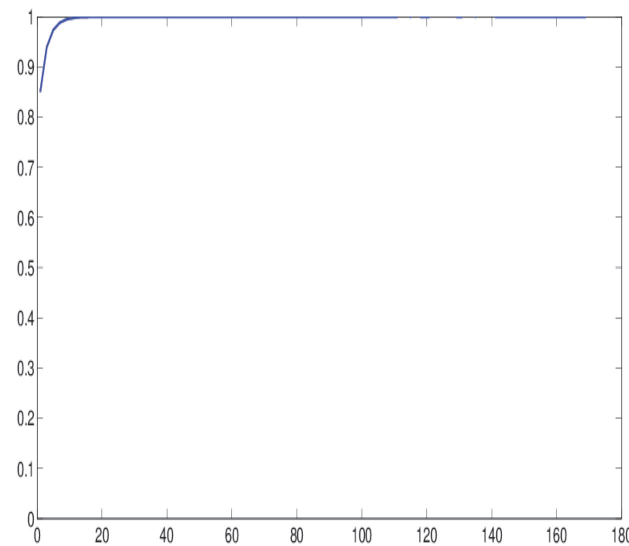
Probability as function of the number of votes



$p = 0.55$



$p = 0.45$



$p = 0.85$

Terminology

- An **ensemble method** makes a prediction by combining the predictions of many classifiers into a single vote.
- The individual classifiers are usually required to perform only slightly better than random. For two classes, this means slightly more than 50% of the data are classified correctly. Such a classifier is called a **weak learner**.

Strategy

- We have seen above that if the weak learners are random and independent, the prediction accuracy of the majority vote will increase with the number of weak learners.
- Since the weak learners all have to be trained on the training data, producing random, independent weak learners is difficult.
- Different ensemble methods (e.g. Boosting, Bagging, etc) use different strategies to train and combine weak learners that behave relatively independently.

METHODS WE WILL DISCUSS

Boosting

- After training each weak learner, data is modified using weights.
- Deterministic algorithm.

Bagging

- Each weak learner is trained on a random subset of the data.

Random forests

- Bagging with tree classifiers as weak learners.
- Uses an additional step to remove dimensions in $\mathbb{R}^d$ that carry little information.