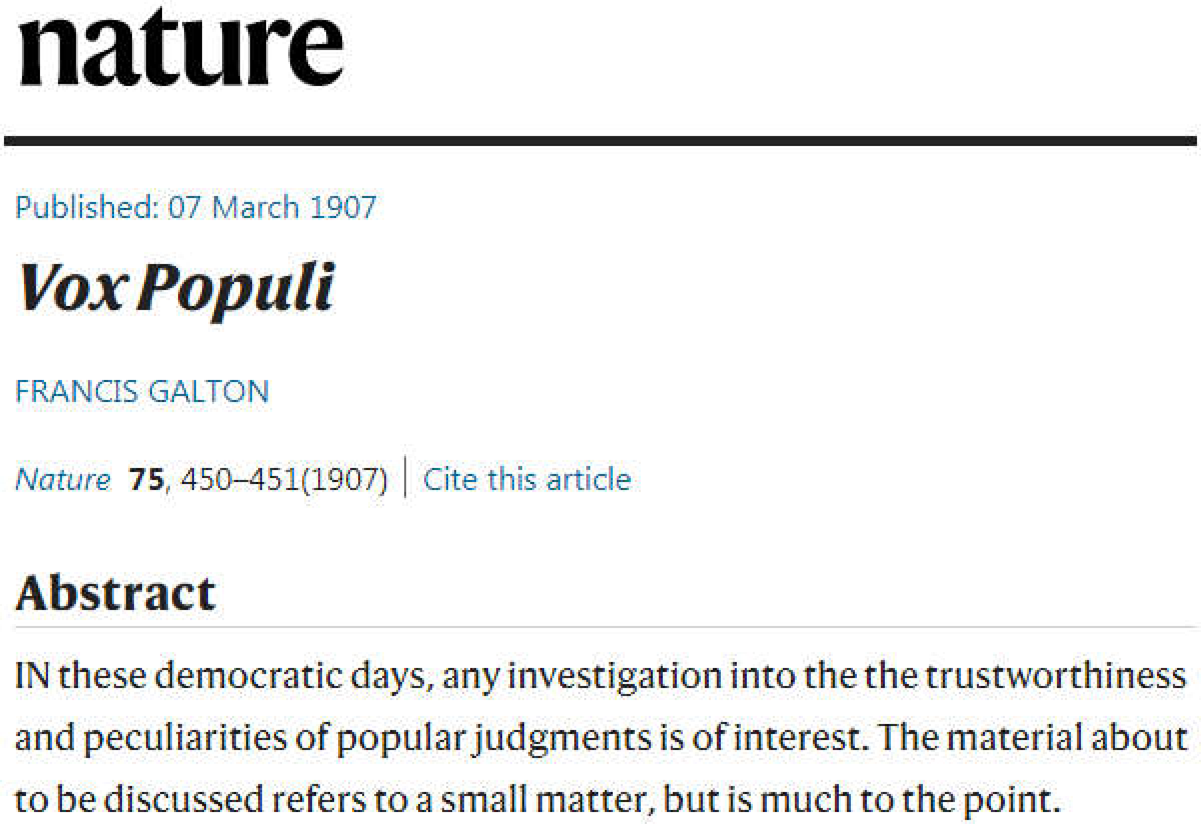
Ensembles of Classifiers

Nazerfard, Ehsan
nazerfard@aut.ac.ir

The Wisdom of Crowds



nature

Published: 07 March 1907

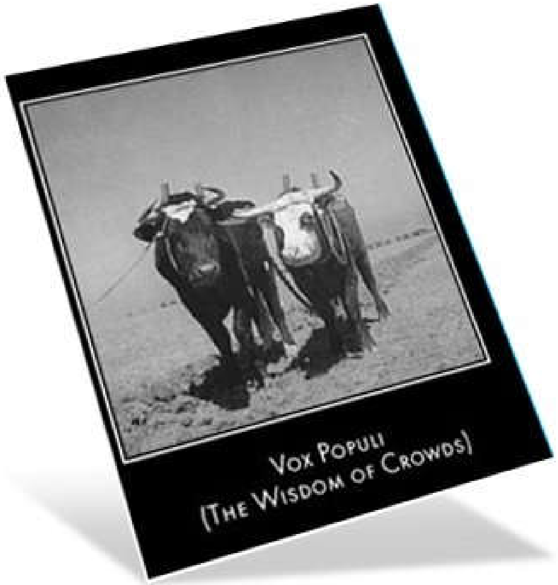
Vox Populi

FRANCIS GALTON

Nature **75**, 450–451(1907) | Cite this article

Abstract

IN these democratic days, any investigation into the the trustworthiness and peculiarities of popular judgments is of interest. The material about to be discussed refers to a small matter, but is much to the point.



Ask the Audience

- In Wikipedia's current logo, depicting a spherical jigsaw puzzle, which letter is seen on the piece immediately to the right of the one bearing an "Ω"?

A. W

B. X

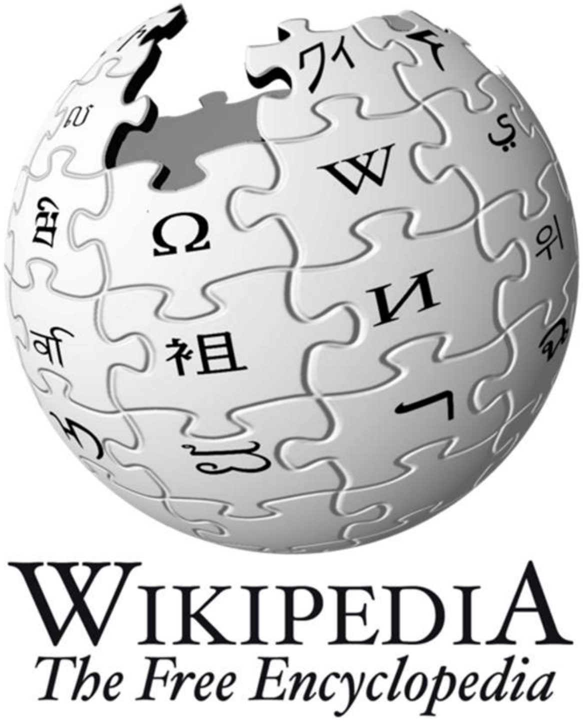
C. Y

D. Z

Ask the Audience (cont.)

- In Wikipedia’s current logo, depicting a spherical jigsaw puzzle, which letter is seen on the piece immediately to the right of the one bearing an “Ω”?

A. W B. X
C. Y D. Z



© Who wants to be a millionaire

Ensemble of Classifiers

- Goal
  - Improve accuracy of supervised learning task
- Approach
  - Use an ensemble of classifiers, rather than just one
- Challenges
  - How to construct ensemble
  - How to use individual hypotheses of ensemble to produce a classification

Ensemble of Classifiers (cont.)

- Given ensemble of L classifiers $h_1, h_2, \ldots, h_L$ (called base classifiers)

- Decision based on combination of $h_l's$ $(l = 1..L)$
  - E.g., weighted or unweighted voting

- How to construct ensemble whose accuracy is better than any individual classifier?

Ensemble of Classifiers (cont.)

- Ensemble requirements
  - Individual classifiers disagree
  - Each classifier's error < 0.5
  - Classifiers' errors uncorrelated
- Then, ensemble will outperform any h_l

Ensemble Methods

- Two main categories:

  I. Bagging (bootstrap aggregation)

  II. Boosting → Adaboost

- Combining a set of heterogeneous classifiers: Stacking, blending, voting, …
- Graph-based cross-validated committees ensembles
- …

Bagging

- Given m training examples
- Construct L random samples of size m with replacement (bootstrap step)
  - Each sample called a bootstrap replicate
  - On average, each replicate contains 63.2% of training data
- Learn a classifier h_l for each of the L samples
- Average over the learned classifiers to produce the final classifier (aggregation step)

Bagging (cont.)

- How Bagging combines classifiers:

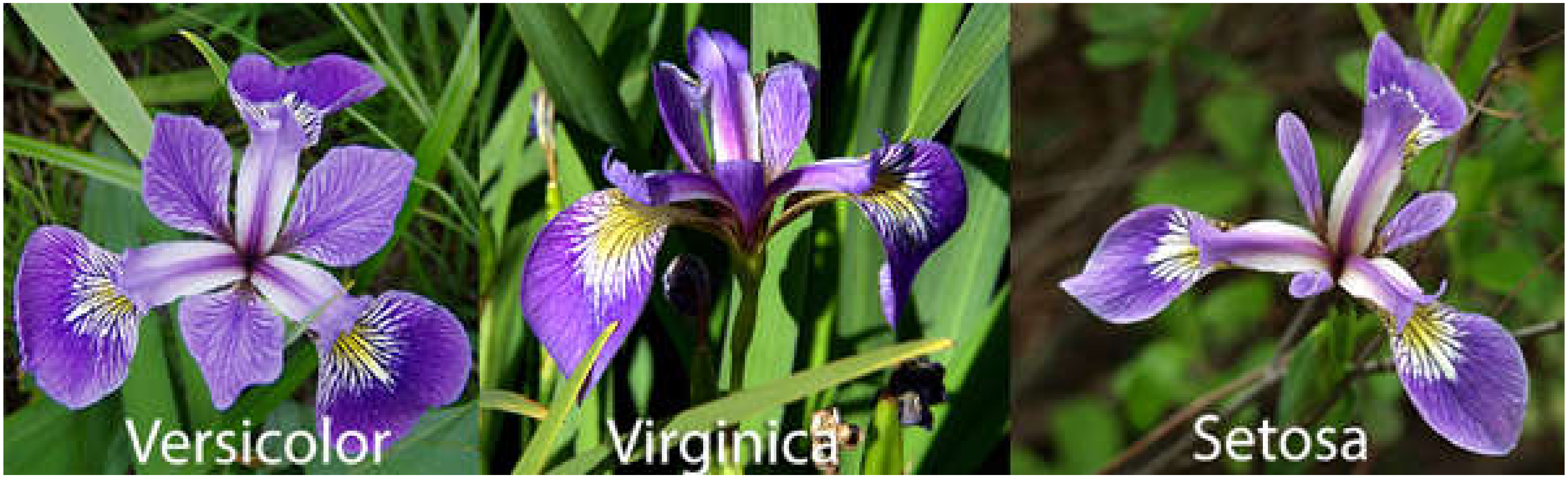
$$h: X \to \{-1, +1\}$$

  - Unweighted voting

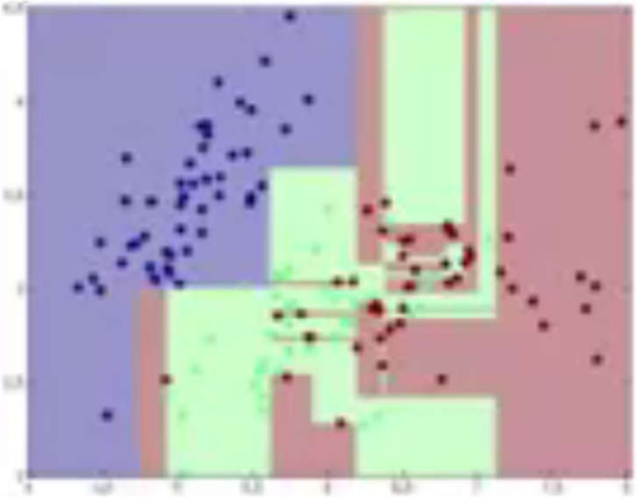
$$h(\boldsymbol{x}_{test}) = \text{sgn}[h_1(\boldsymbol{x}_{test}) + h_2(\boldsymbol{x}_{test}) + \cdots + h_L(\boldsymbol{x}_{test})]$$

- The main idea behind Bagging is to reduce overfitting.
  - It works well for unstable/ low bias/ high variance models, such as Decision Trees.
  - It doesn’t work for linear models. Why?

Three Iris Types



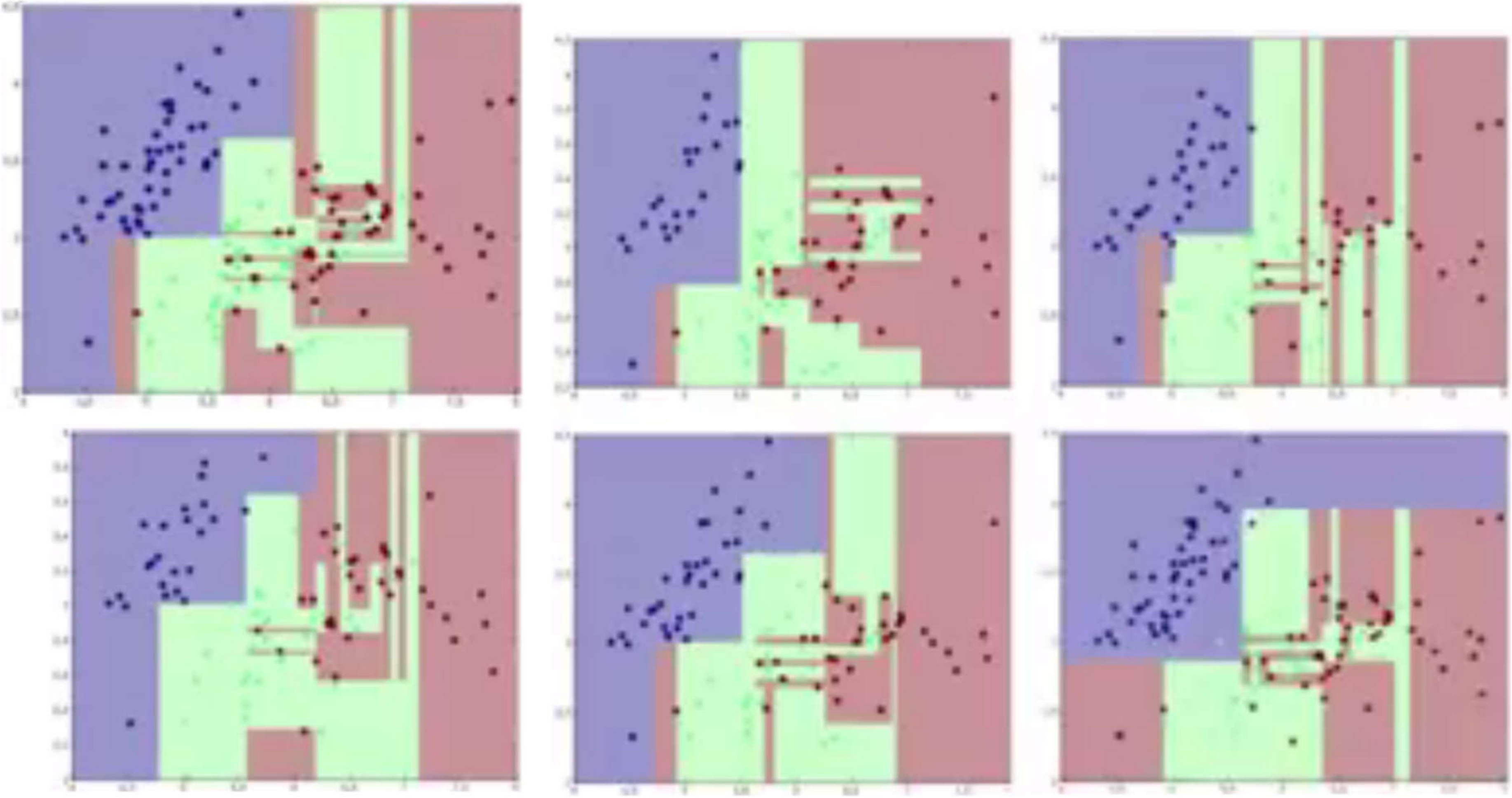
Decision Tree (DT) – decision boundaries



The full Iris.2D dataset

Figure © Alex Ihler, Lecture on Ensemble Methods, UCI

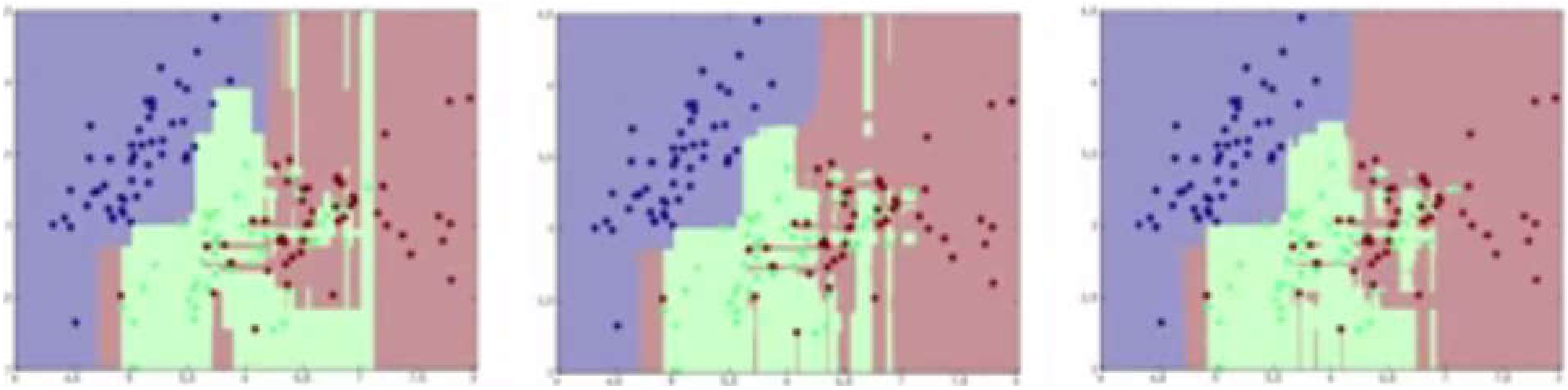
Learned DTs (Bootstrap step)



Figures © Alex Ihler, Lecture on Ensemble Methods, UCI

Testing/validation step

- Aggregating (averaging) over the learned Bagged Trees
  - e.g. majority vote/unweighted average (classification)



Average of 5 trees

Average of 25 trees

Average of 100 trees

- The Random Forest Algorithm

Figures © Alex Ihler, Lecture on Ensemble Methods, UCI

Boosting

- Boosting is based on the question posed by Michael Kearns (1988) [2]: "Can a set of weak learners create a single strong learner?"
  - Boosting is an ensemble meta-algorithm for primarily reducing bias, and also variance.

Adaboost Training Phase

$$\begin{cases} w_i^1 = \dfrac{1}{m} \\ \epsilon^1 = \sum\limits_i w_i^1 \end{cases}$$

Subject to: i=1..m

Subject to: i_{th} example is incorrectly classified

Total error at step t: $$\epsilon^t = \sum_i w_i^t$$

Incorrectly
classified
examples

Adaboost Final Classifier

- How Adaboost combines classifiers:

$$h\colon X \to \{-1,+1\}$$

  - Weighted voting

$$h(\boldsymbol{x}_{\boldsymbol{test}}) = \operatorname{sgn}[\alpha^1 h^1(\boldsymbol{x}_{\boldsymbol{test}}) + \alpha^2 h^2(\boldsymbol{x}_{\boldsymbol{test}}) + \cdots]$$

Adaboost: Components and Steps

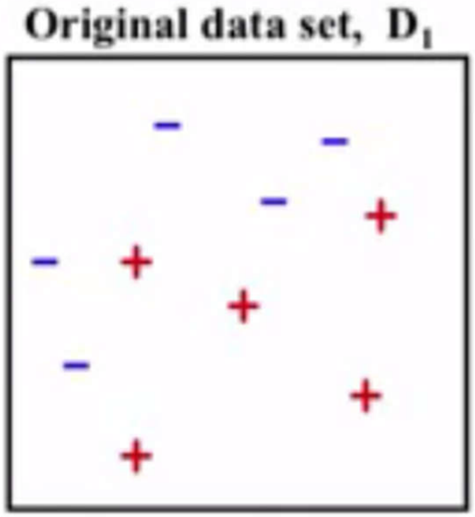
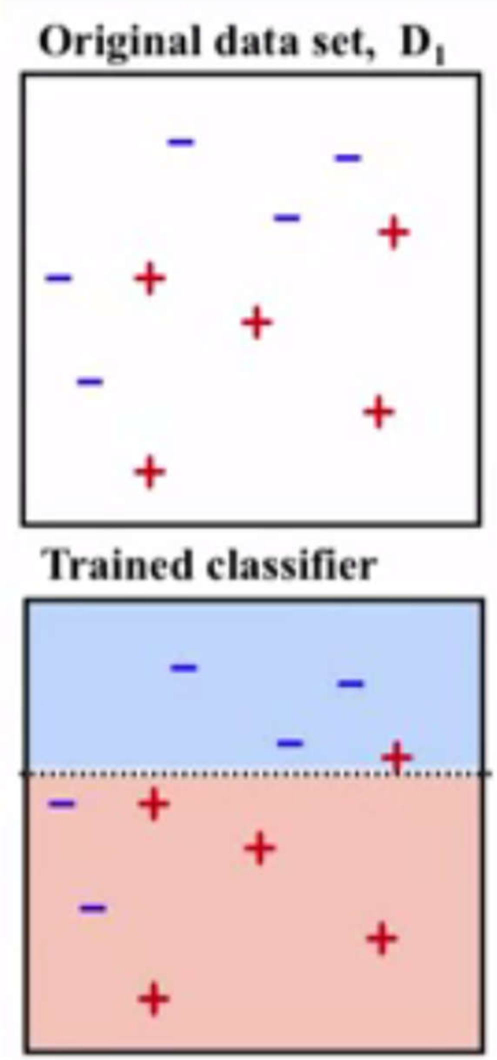


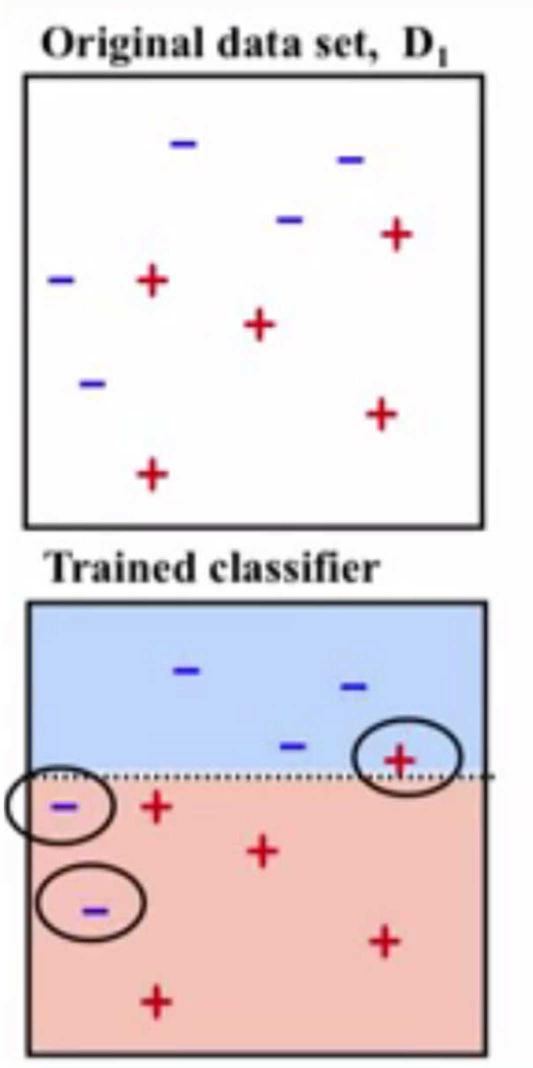
Figure © Alex Ihler, Lecture on Ensemble Methods, UCI

Base Classifier: Decision Stump



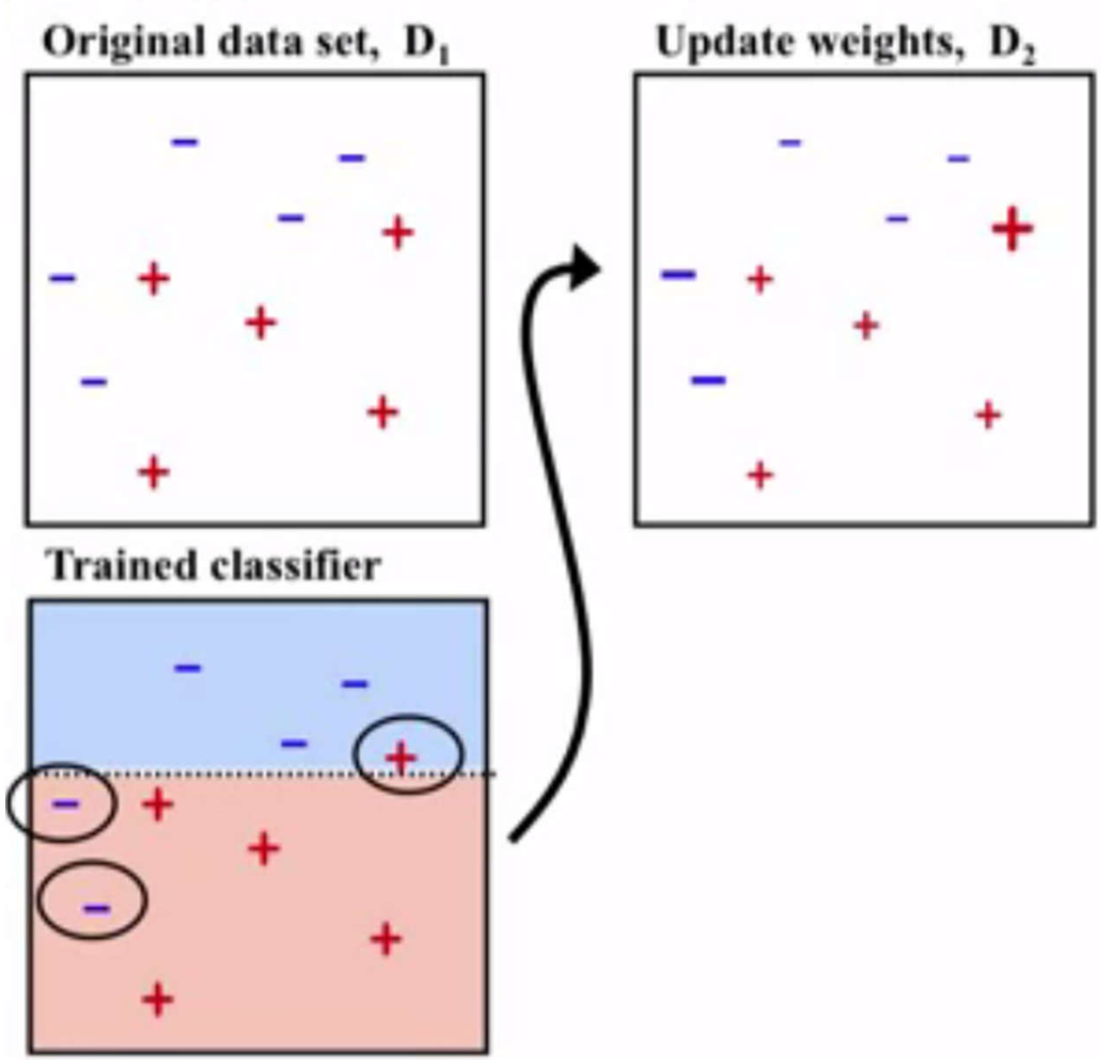
Figures © Alex Ihler, Lecture on Ensemble Methods, UCI

Incorrectly Classified Examples



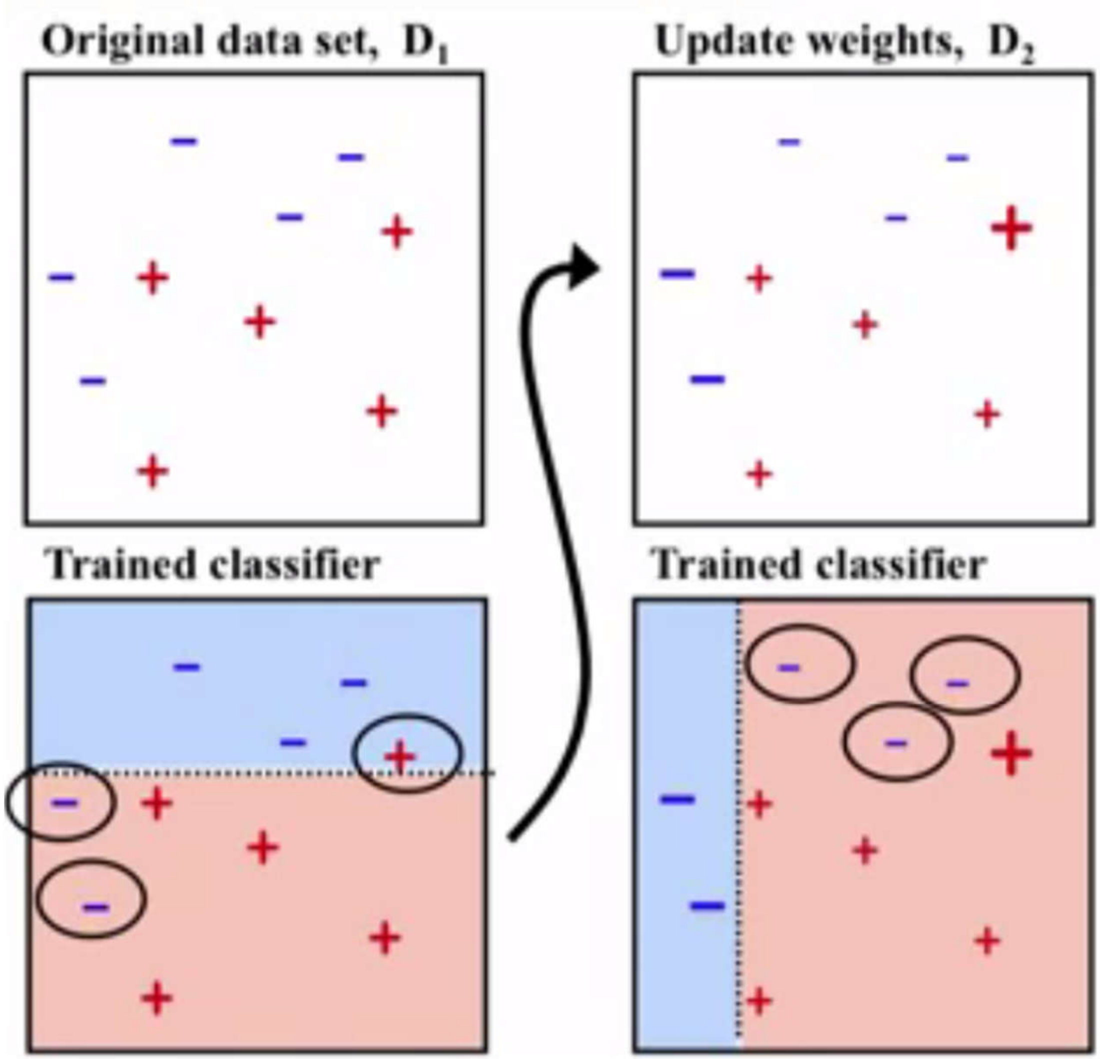
Figures © Alex Ihler, Lecture on Ensemble Methods, UCI

Emphasizing Incorrectly Classified Examples



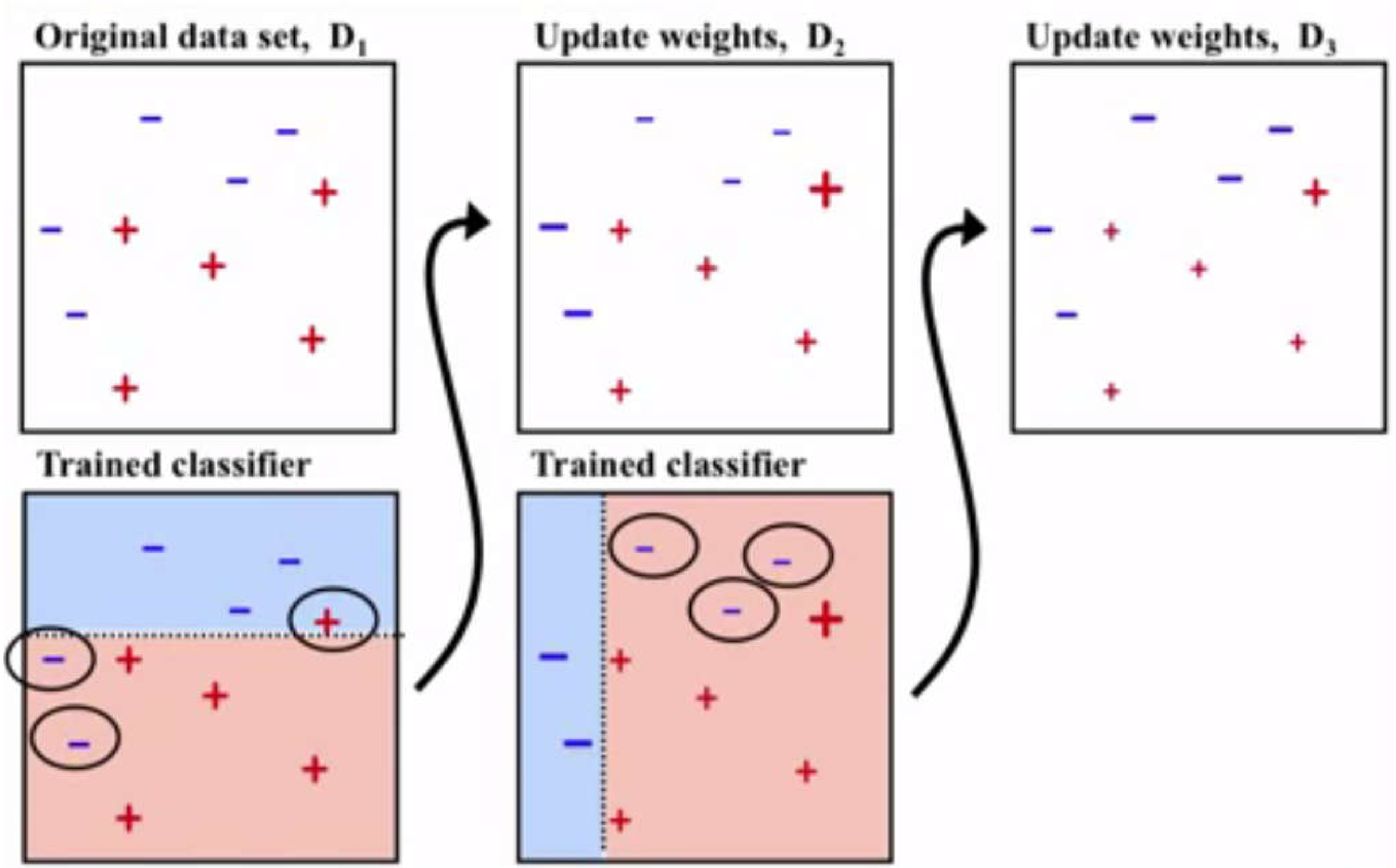
Figures © Alex Ihler, Lecture on Ensemble Methods, UCI

Incorrectly Classified Examples



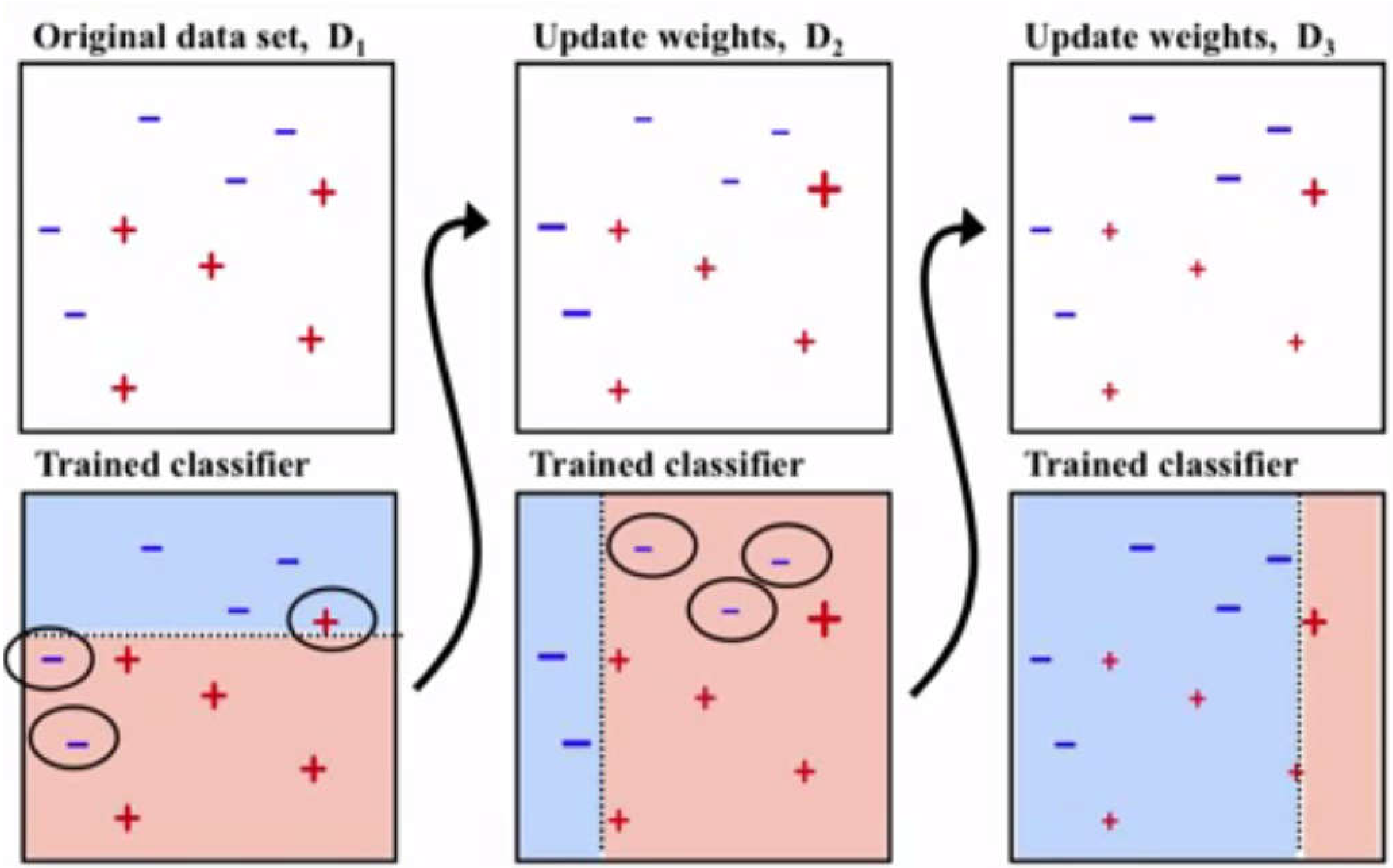
Figures © Alex Ihler, Lecture on Ensemble Methods, UCI

Emphasizing Incorrectly Classified Examples

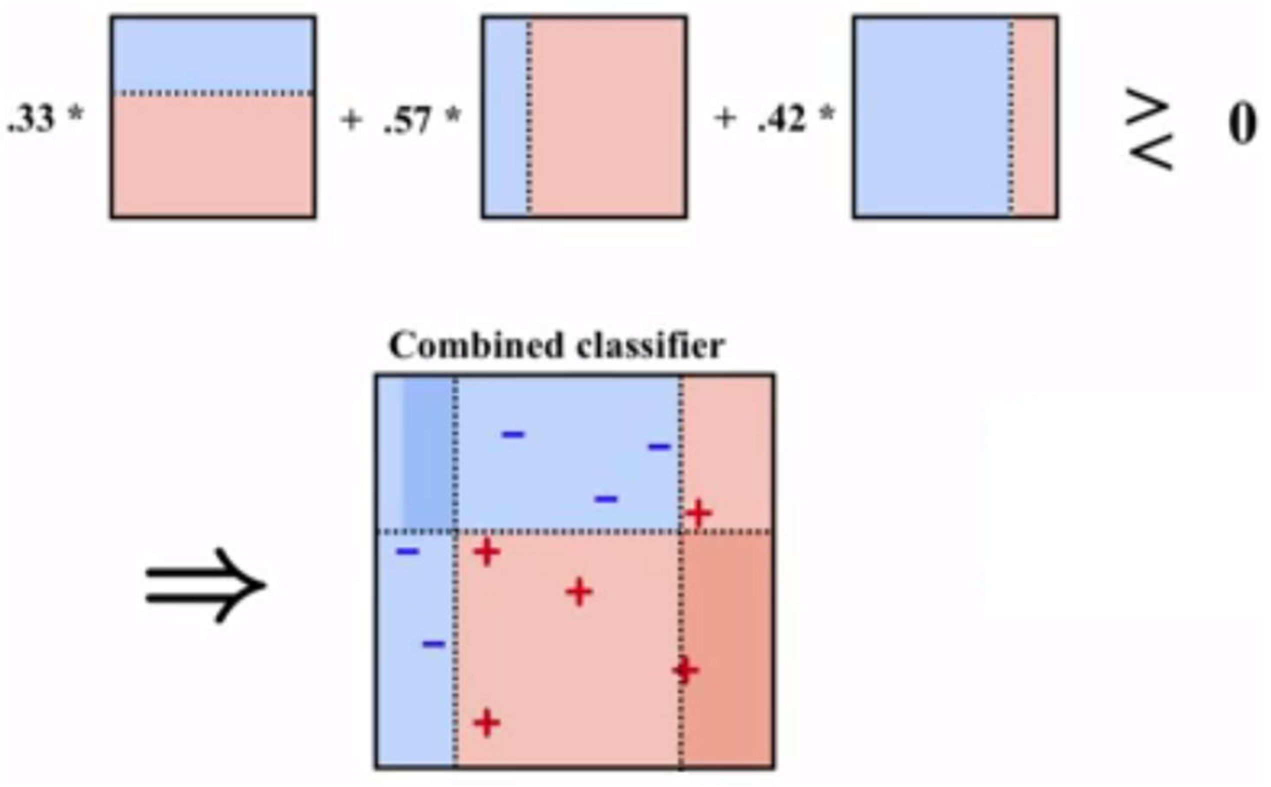


Figures © Alex Ihler, Lecture on Ensemble Methods, UCI

Training Done



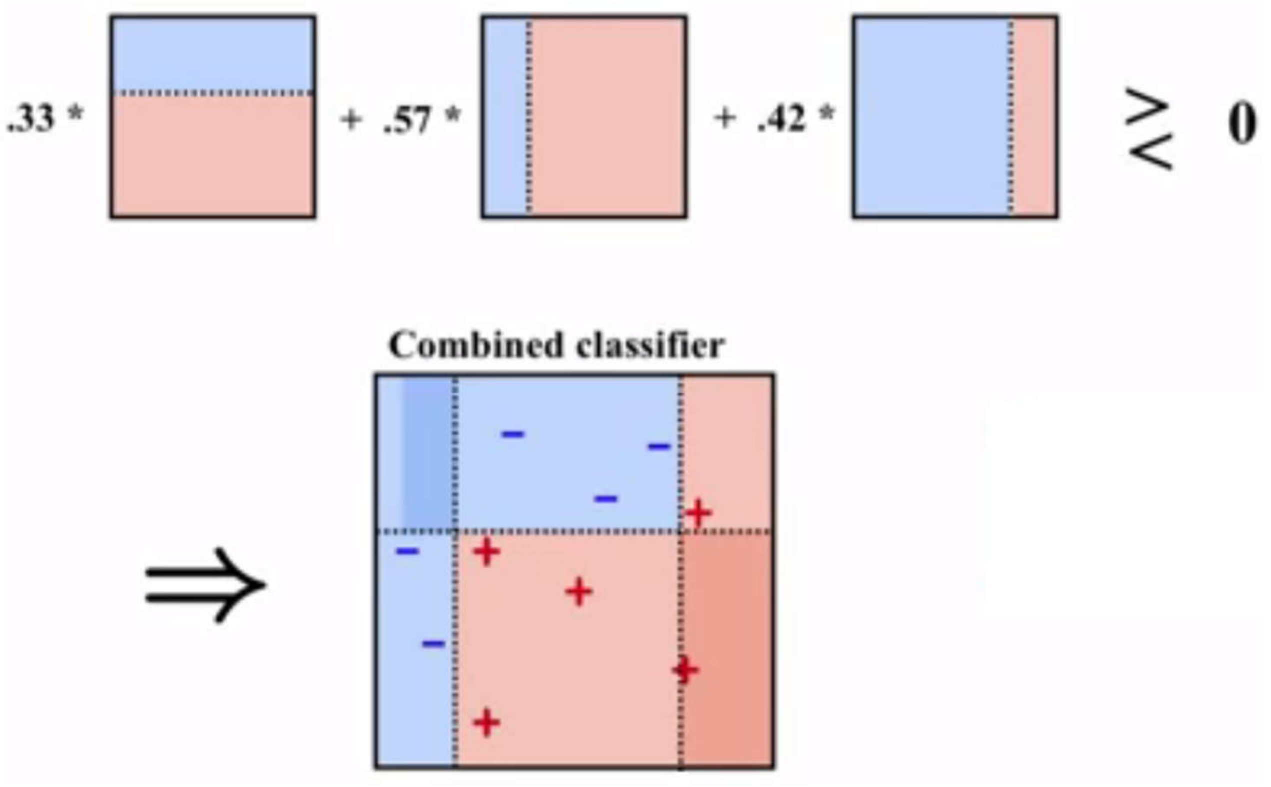
Final Classifier: Weighted Voting



- ❑ Adaboost (Adaptive Boosting)
  - ○ Yoav Freund and Schapire, 97 [1]

Figures © Alex Ihler, Lecture on Ensemble Methods, UCI

Final Classifier: Weighted Voting



- AdaBoost (Gödel Prize, 2003)
  - This paper [3] (simplified version)

Figures © Alex Ihler, Lecture on Ensemble Methods, UCI

Adaboost Details

- How to update weights at each step:

$$
\begin{cases}
w_i^{t+1} = \dfrac{w_i^t}{Z^t} \times \exp[-\alpha^t h^t(\boldsymbol{x_i}) y(\boldsymbol{x_i})] \\
\alpha^t = \dfrac{1}{2} \ln \dfrac{1-\epsilon^t}{\epsilon^t}
\end{cases}
$$

Adaboost Details (cont.)

- How to update weights at each step:

$$\begin{cases} w_i^{t+1} = \dfrac{w_i^t}{Z^t} \times \exp[-\alpha^t h^t(\boldsymbol{x_i}) y(\boldsymbol{x_i})] \\ \alpha^t = \dfrac{1}{2} \ln \dfrac{1-\epsilon^t}{\epsilon^t} \end{cases}$$

$$\Rightarrow w_i^{t+1} = \frac{w_i^t}{Z^t} \times \begin{cases} \sqrt{\dfrac{\epsilon^t}{1-\epsilon^t}} & h^t(\boldsymbol{x_i}) = y_i \\ \sqrt{\dfrac{1-\epsilon^t}{\epsilon^t}} & h^t(\boldsymbol{x_i}) \neq y_i \end{cases}$$

Adaboost Details (cont.)

- Normalization factor:

$$Z^t = \underset{\text{Correctly classified}}{\sum_i} w_i^t \times \sqrt{\frac{\epsilon^t}{1-\epsilon^t}} + \underset{\text{Incorrectly classified}}{\sum_i} w_i^t \times \sqrt{\frac{1-\epsilon^t}{\epsilon^t}} = 2\sqrt{\epsilon^t(1-\epsilon^t)}$$

Adaboost Details (cont.)

- Normalization factor:

$$Z^t = \sum_{i} w_i^t \times \sqrt{\frac{\epsilon^t}{1-\epsilon^t}} + \sum_{i} w_i^t \times \sqrt{\frac{1-\epsilon^t}{\epsilon^t}} = 2\sqrt{\epsilon^t(1-\epsilon^t)}$$

Correctly classified

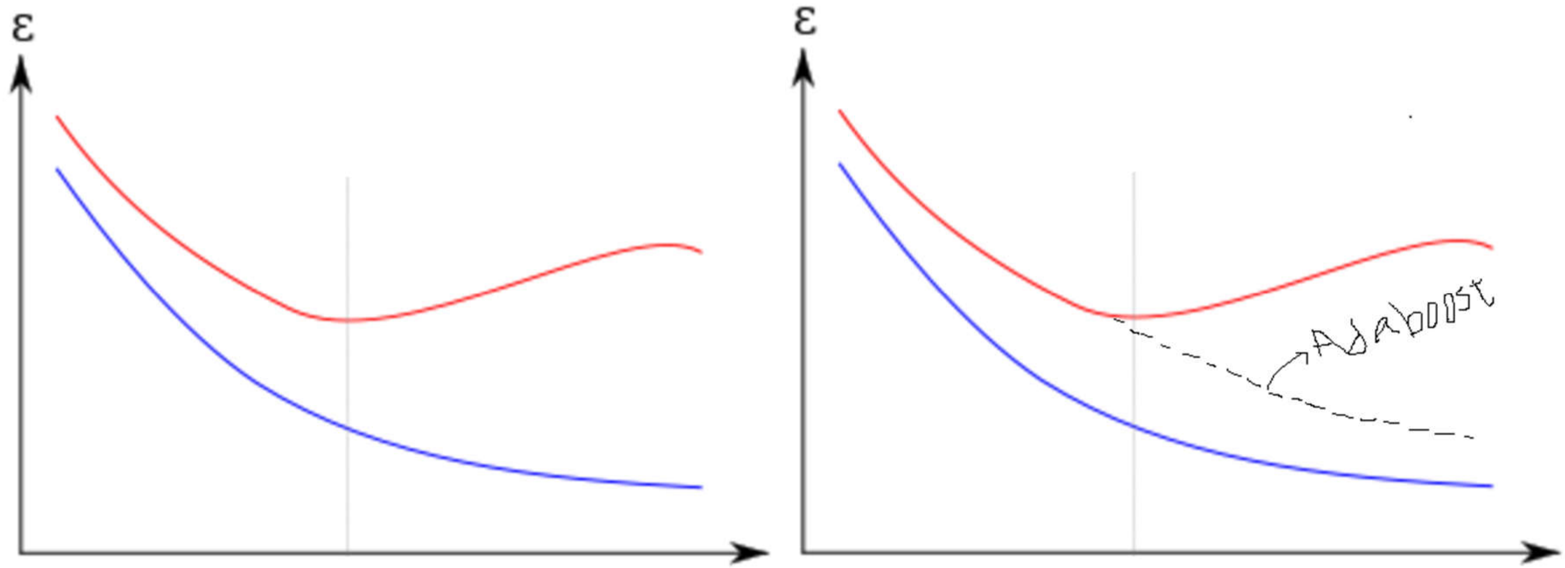
Incorrectly classified

- Weight updating rule:

$$\begin{cases} w_i^{t+1} = \dfrac{w_i^t}{2} \times \dfrac{1}{1-\epsilon^t} & h^t(\boldsymbol{x_i}) = y_i \\ w_i^{t+1} = \dfrac{w_i^t}{2} \times \dfrac{1}{\epsilon^t} & h^t(\boldsymbol{x_i}) \neq y_i \end{cases}$$

Adaboost Properties

- Adaboost is sensitive to noisy data & outliers, less sensitive to overfitting. (Why?)
- Bias-variance Tradeoff



Adaboost Properties (cont.)

- Adaboost is sensitive to noisy data & outliers, less sensitive to overfitting.

- Adaboost (with decision stump/trees as the weak classifier) is often referred to as the best out-of-the-box classifier.

Further Reading

- T. Dietterich, Ensemble Methods in Machine Learning, International Workshop on Multiple Classifier Systems, Lecture Notes in Computer Science, pp.1-15, 2000.

- J. H. Friedman, Greedy Function Approximation: A Gradient Boosting Machine, 1999.

- J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression: a statistical view of boosting, Annals of Statistics. 28 (2): 337–407, 2000. → LogitBoost Algorithm

- Random Forest Regression

References

1. Yoav Freund and Robert E. Schapire, A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting, computer and system sciences 55, 119-139,1997.
2. Michael Kearns, Thoughts on Hypothesis Boosting, Unpublished manuscript (ML class project), 1988.
3. Yoav Freund, Robert E. Schapire, A Short Introduction to Boosting, Journal of Japanese Society for Artificial Intelligence, 14(5):771-780, 1999 (translated).
4. Patrick Winston, Lecture on Boosting, MIT