11.10.2024

Statistical Methods in AI (CS7.403)

Lecture-19: Ensemble Methods (Bagging, Boosting, Stacking)

Ravi Kiran (ravi.kiran@iiit.ac.in)

https://ravika.github.io

@vikataravi

Center for Visual Information Technology (CVIT)
IIIT Hyderabad

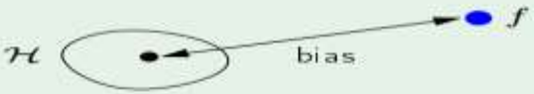
$$\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x})\right)^2\right] = \underbrace{\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x})\right)^2\right]}_{\text{var}(\mathbf{x})} + \underbrace{\left(\bar{g}(\mathbf{x}) - f(\mathbf{x})\right)^2}_{\text{bias}(\mathbf{x})}$$

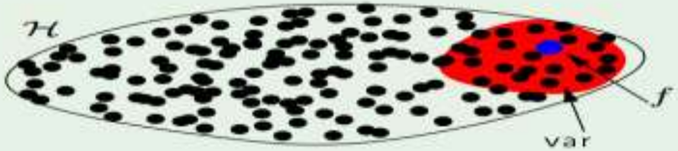
$$\mathbb{E}_{\mathbf{x}}[\text{bias}(\mathbf{x}) + \text{var}(\mathbf{x})]$$

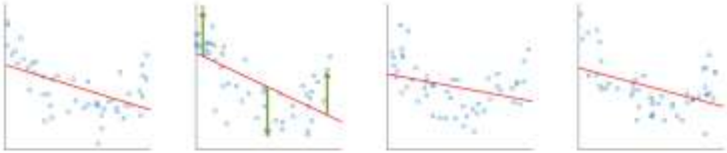
The tradeoff

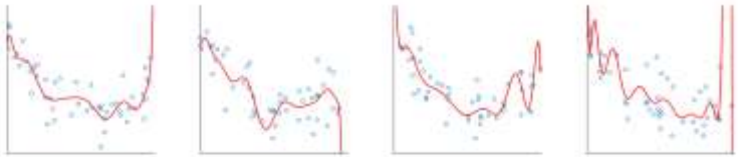
$$\text{bias} = \mathbb{E}_{\mathbf{x}}\left[\left(\bar{g}(\mathbf{x}) - f(\mathbf{x})\right)^2\right]$$

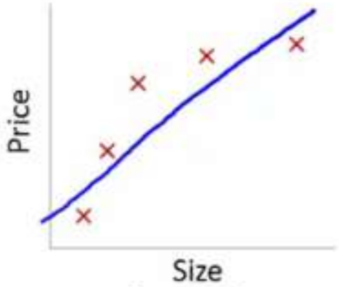
$$\text{var} = \mathbb{E}_{\mathbf{x}}\left[\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x})\right)^2\right]\right]$$





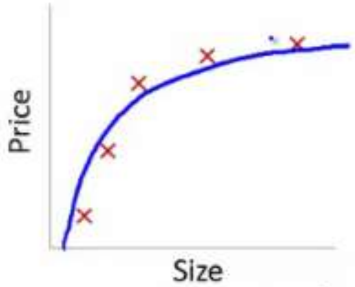






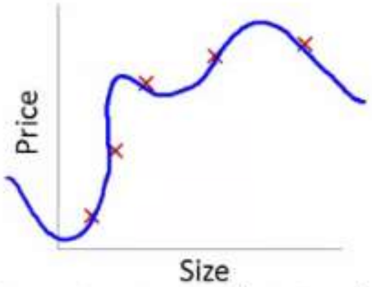
$\theta_0 + \theta_1 x$

High bias
(underfit)



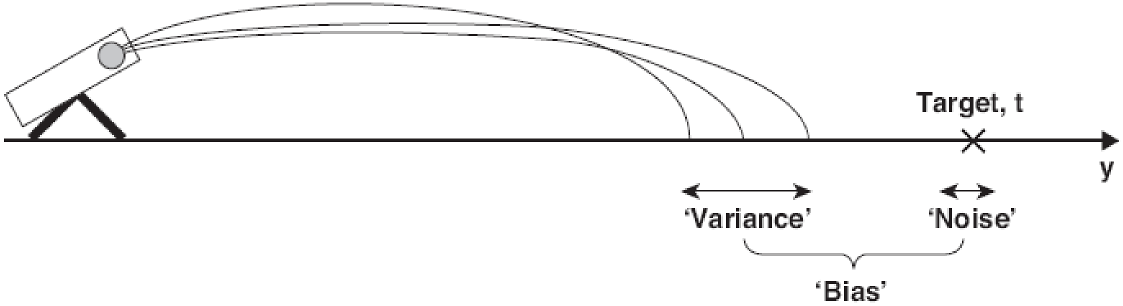
$\theta_0 + \theta_1 x + \theta_2 x^2$

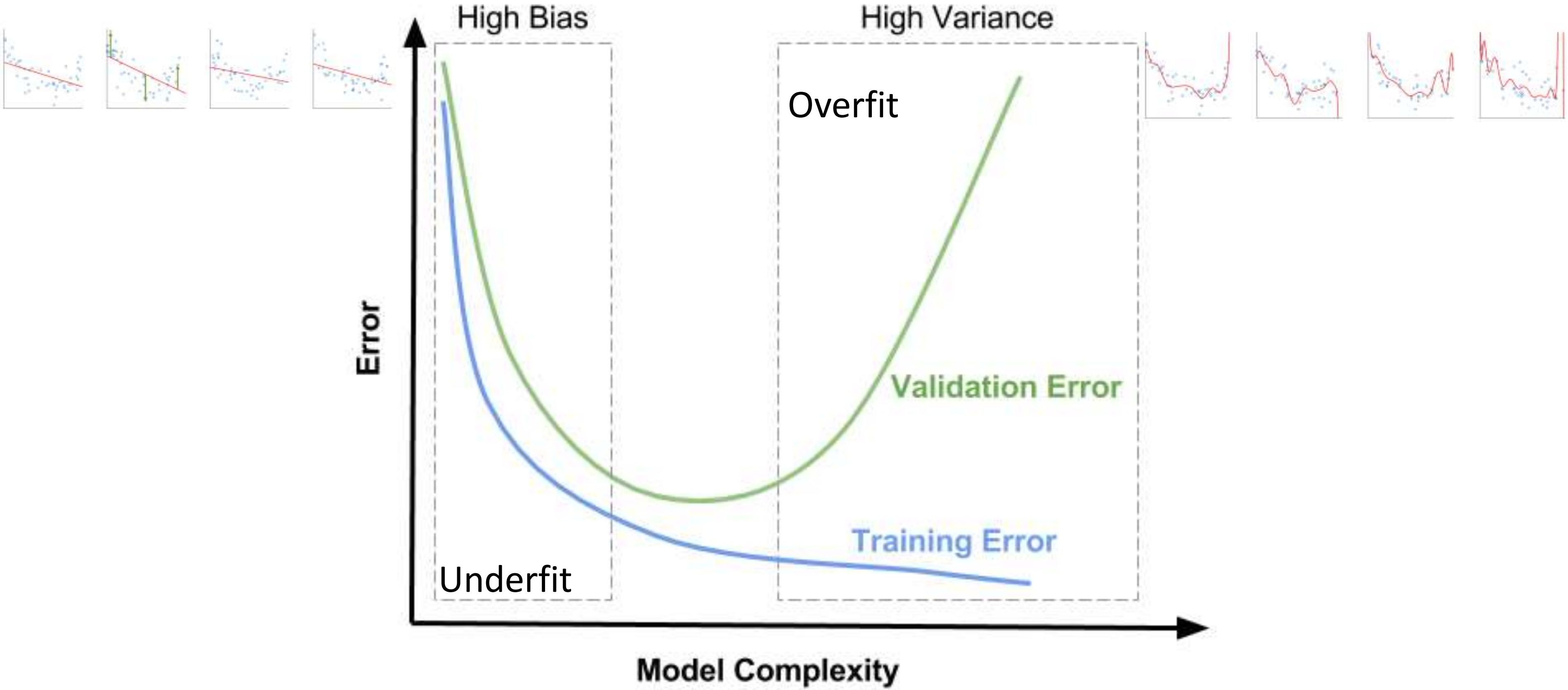
“Just right”



$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

High variance
(overfit)

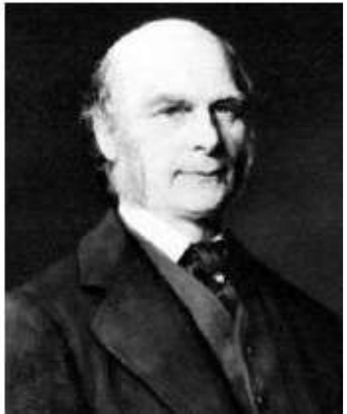




High Bias
High Variance
Overfit
Validation Error
Training Error
Underfit
Error
Model Complexity

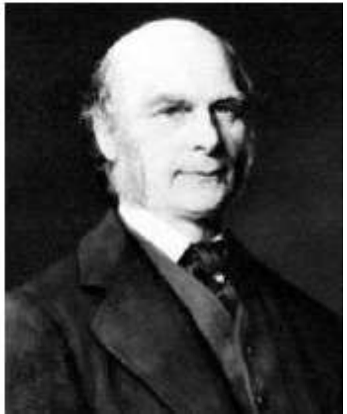
Francis Galton

- Galton promoted statistics and invented the concept of correlation.
- In 1906 Galton visited a livestock fair and stumbled upon an intriguing contest.
- An ox was on display, and the villagers were invited to guess the animal's weight.
- Nearly 800 gave it a go and, not surprisingly, not one hit the exact mark: 1,198 pounds.



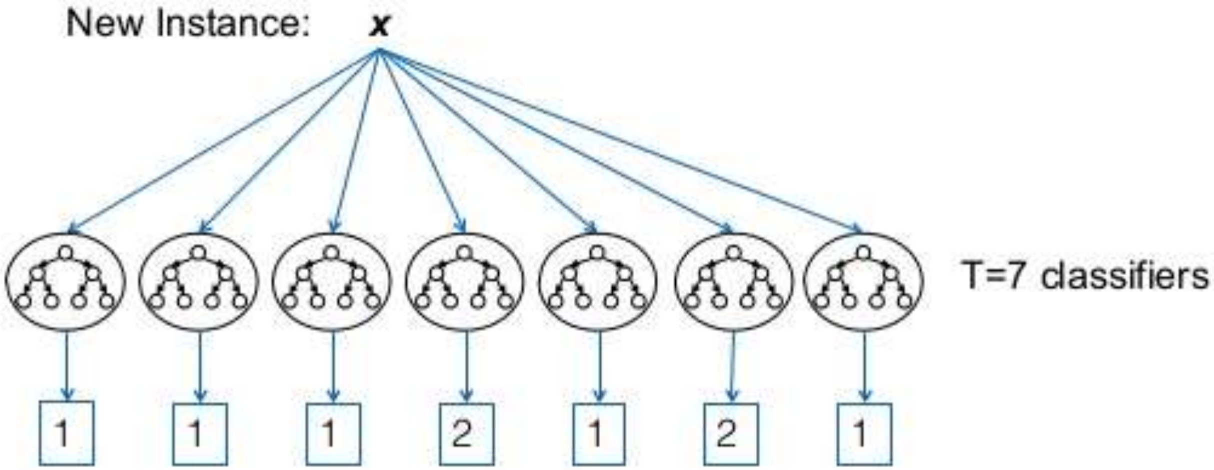
Francis Galton

- Galton promoted statistics and invented the concept of correlation.
- In 1906 Galton visited a livestock fair and stumbled upon an intriguing contest.
- An ox was on display, and the villagers were invited to guess the animal's weight.
- Nearly 800 gave it a go and, not surprisingly, not one hit the exact mark: 1,198 pounds.
- Astonishingly, however, the average of those 800 guesses came close - very close indeed. It was 1,197 pounds.



Ensemble Learning

- An ensemble is a combination of classifiers that output a final classification.



General idea

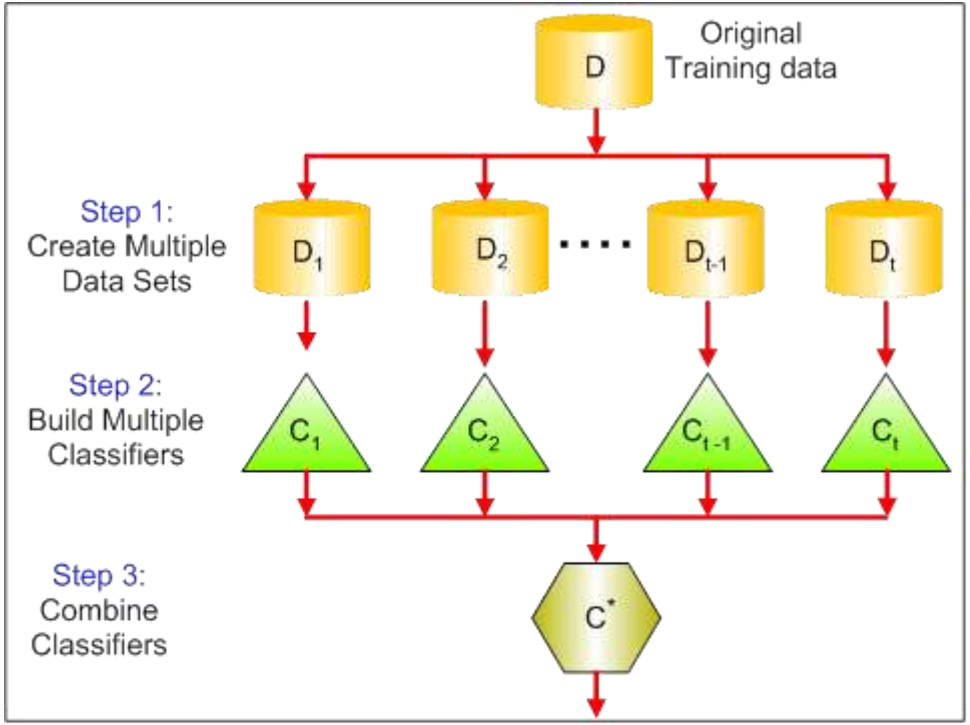
- Generate many classifiers and combine them to get a final classification
- They perform very good. In general better than any of the single learners they are composed of
- The classifiers should be different from one another
- It is important to generate diverse classifiers from the available data

How to build them?

- There are several techniques to build diverse base learners in an ensemble:
  - Use modified versions of the training set to train the base learners

- Modifications of the training set can be generated by
    - Resampling the dataset. By bootstrap sampling (e.g. bagging), weighted sampling (e.g. boosting).

Bootstrap Aggregating (Bagging)



"Bagging Predictors" [Leo Breiman, 1994]
http://statistics.berkeley.edu/sites/default/files/tech-reports/421.pdf

Why does it work?

- Suppose there are 25 base classifiers
  - Each classifier has error rate, $\varepsilon = 0.35$
  - Assume classifiers are independent
  - Probability that the ensemble classifier makes a wrong prediction:

Why does it work?

- Suppose there are 25 base classifiers
  - Each classifier has error rate, ε = 0.35
  - Assume classifiers are independent
  - Probability that the ensemble classifier makes a wrong prediction: $\sum_{i=13}^{25} \binom{25}{i} \varepsilon^i (1-\varepsilon)^{25-i} = 0.06$

```
Input:
  Dataset L
  Ensemble size T

for t=1 to T:
    sample = BootstrapSample(L)
    h_t = TrainClassifier(sample)
```

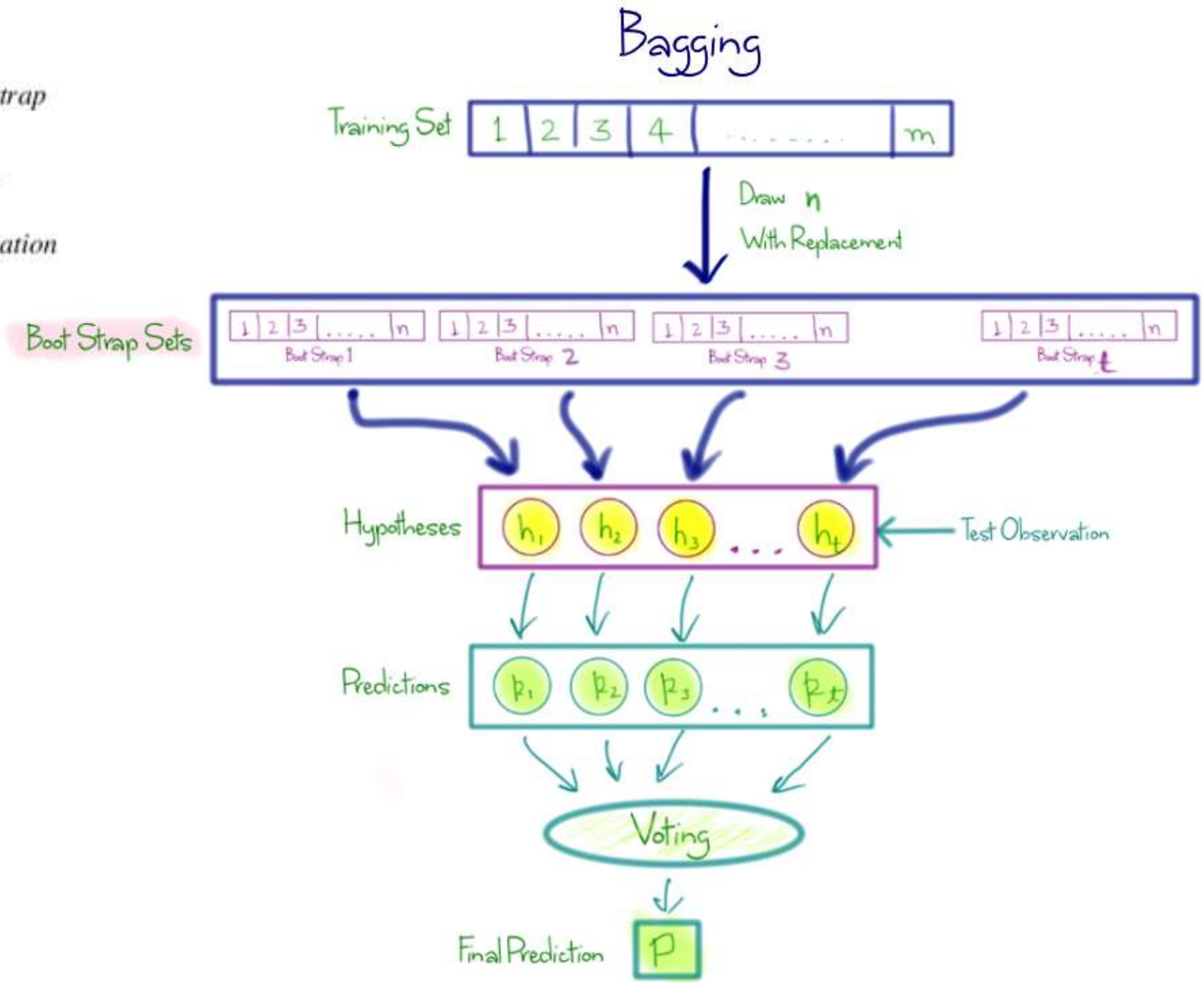
Bootstrap

+

Output:

$$H(\mathbf{x}) = \underset{j}{\operatorname{argmax}}\left(\sum_{t=1}^{T} I(h_t(\mathbf{x}) = j)\right)$$

Aggregation



```
Input:
  Dataset L
  Ensemble size T

for t=1 to T:
    sample = BootstrapSample(L)
    h_t = TrainClassifier(sample)
```

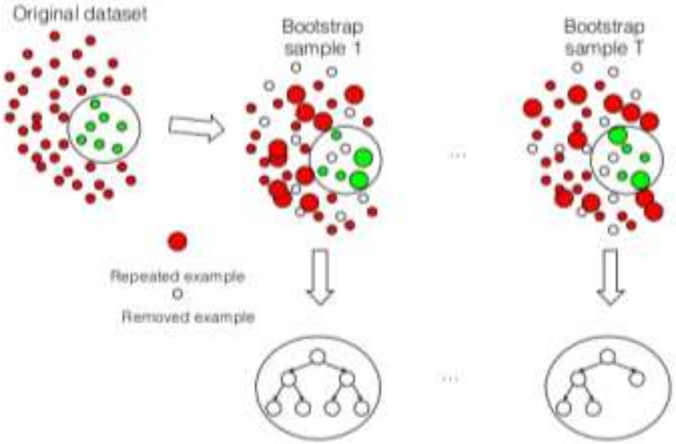
Bootstrap

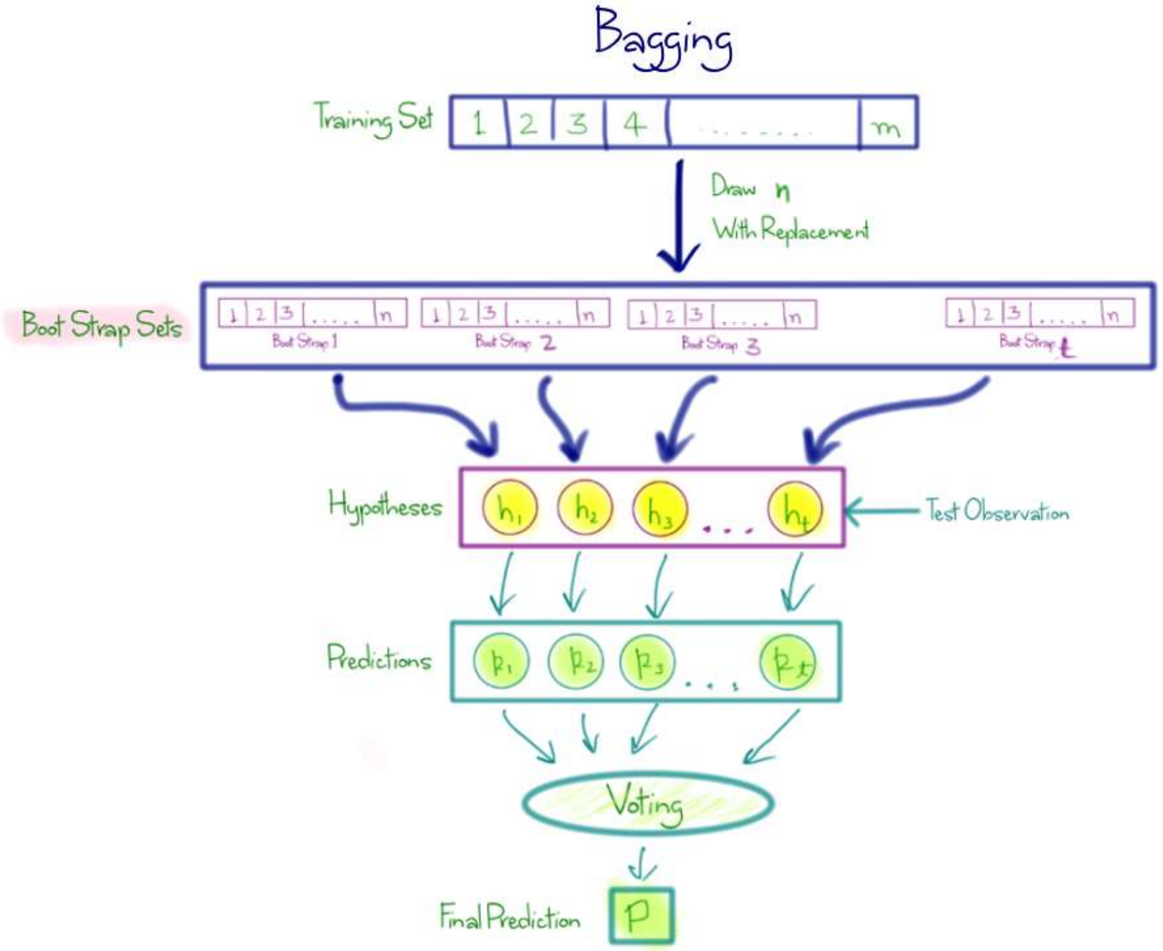
\+

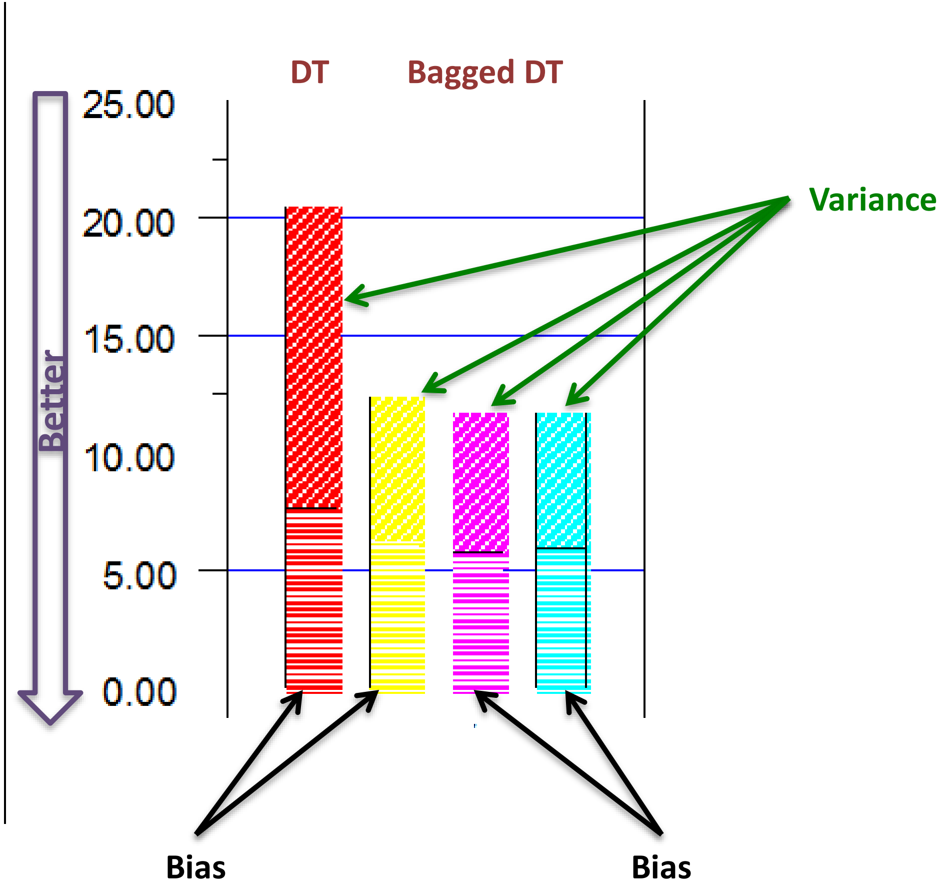
Output:

$$H(\mathbf{x}) = \underset{j}{\operatorname{argmax}}\left(\sum_{t=1}^{T} I(h_t(\mathbf{x}) = j)\right)$$

Aggregation



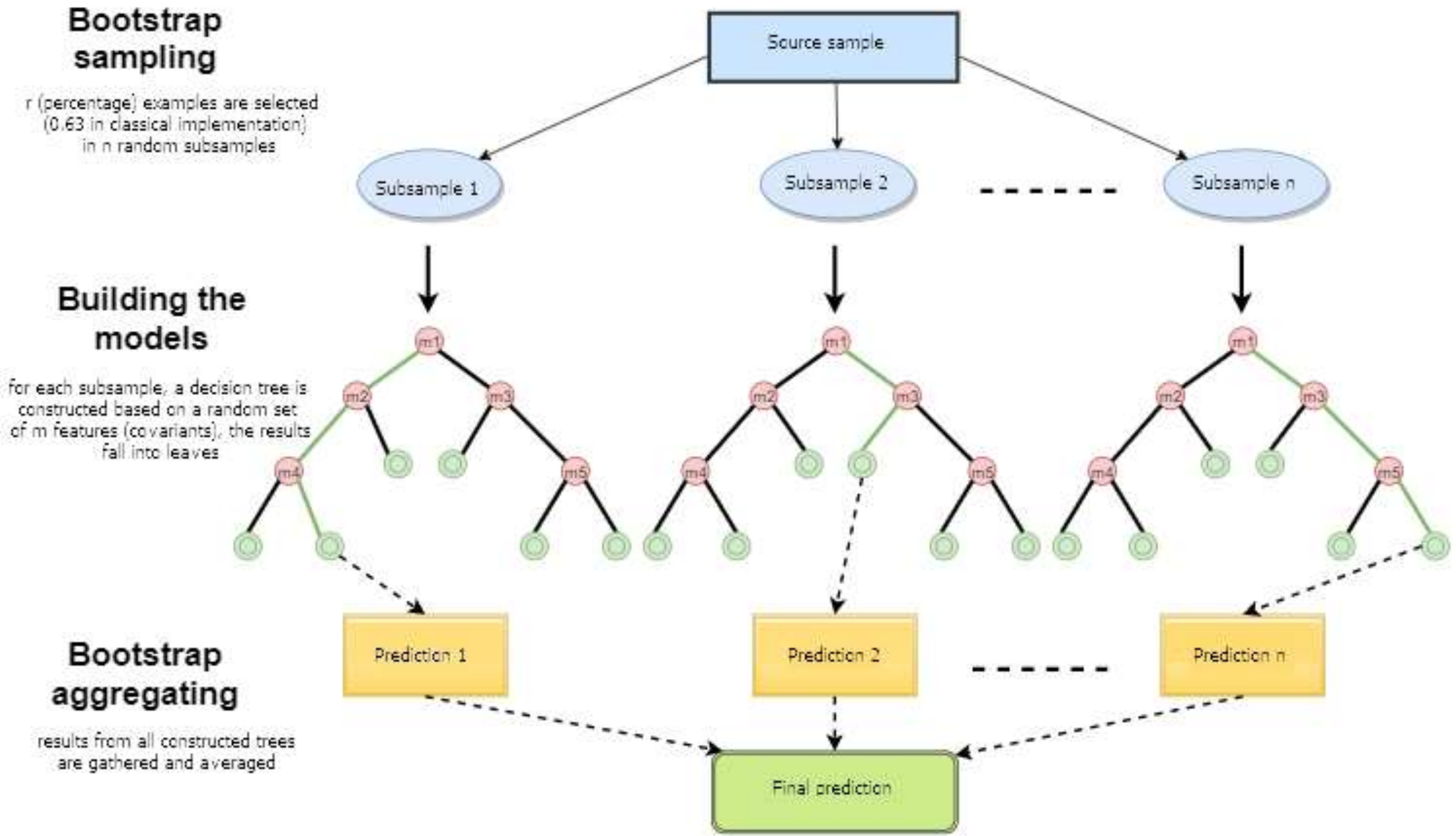




“An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants”
Eric Bauer & Ron Kohavi, Machine Learning 36, 105–139 (1999)

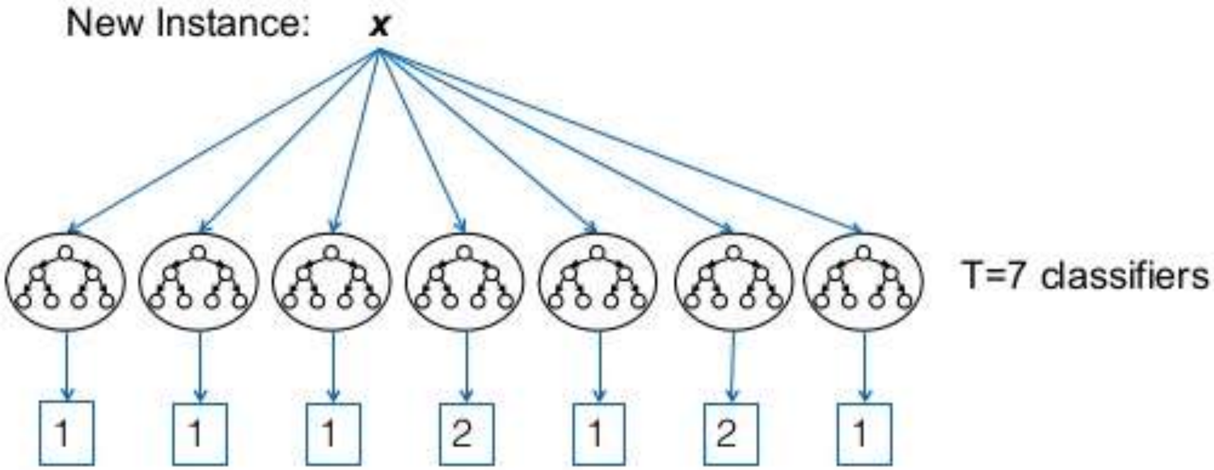
- Modifications of the training set can be generated by
  - Resampling the dataset. By bootstrap sampling (e.g. bagging), weighted sampling (e.g. boosting).
  - Altering the attributes: The base learners are trained using different feature subsets (e.g Random subspaces)

Random Forests



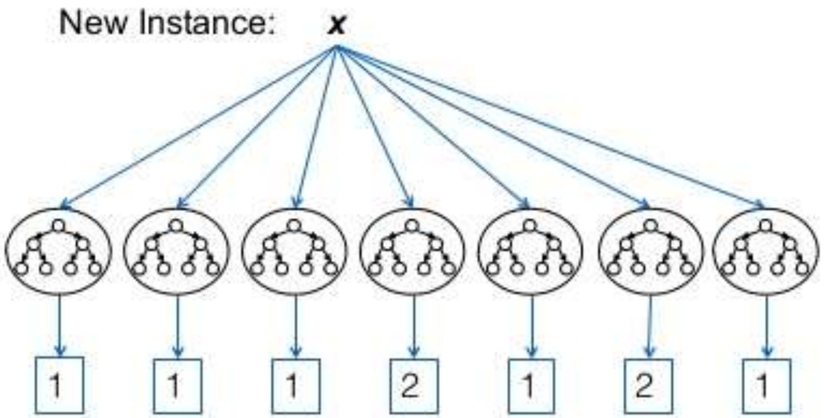
Ensemble Learning

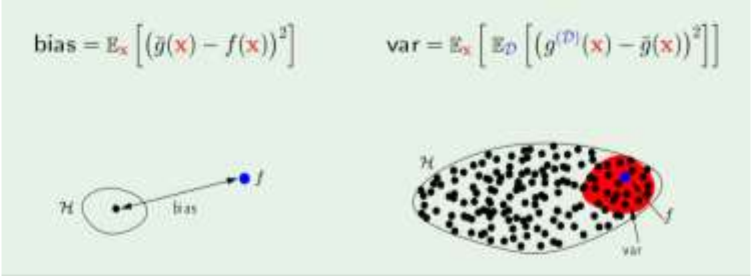
- An ensemble is a combination of classifiers that output a final classification.

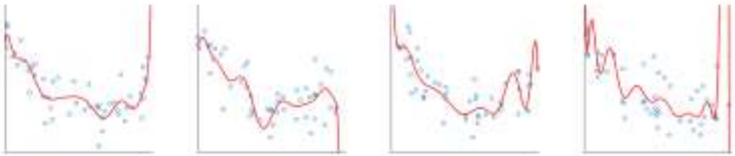


- **Ensemble methods that minimize variance**
  - Classifier Bagging
  - Classifier + Feature Bagging (e.g. Random Forests)

$$\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - f(\mathbf{x})\right)^2\right] = \underbrace{\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x})\right)^2\right]}_{\text{var}(\mathbf{x})} + \underbrace{\left(\bar{g}(\mathbf{x}) - f(\mathbf{x})\right)^2}_{\text{bias}(\mathbf{x})}$$



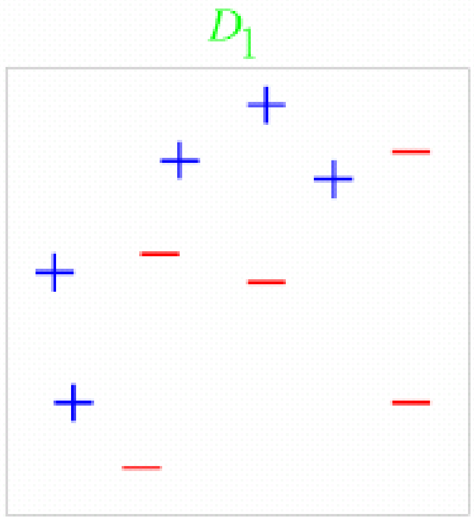




How to build them?

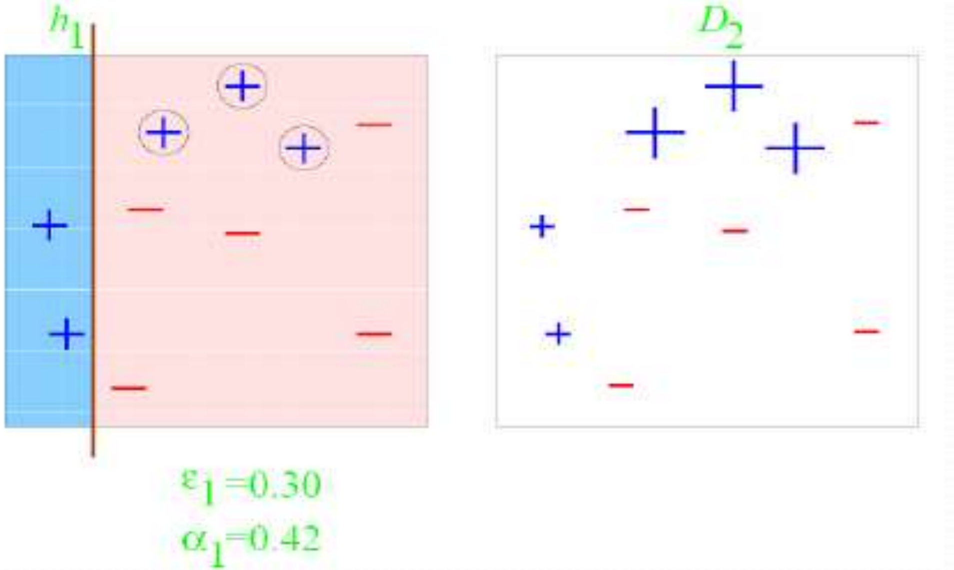
- There are several techniques to build diverse base learners in an ensemble:
  - Use modified versions of the training set to train the base learners
  - Introduce changes in the learning algorithms

A toy example[2]



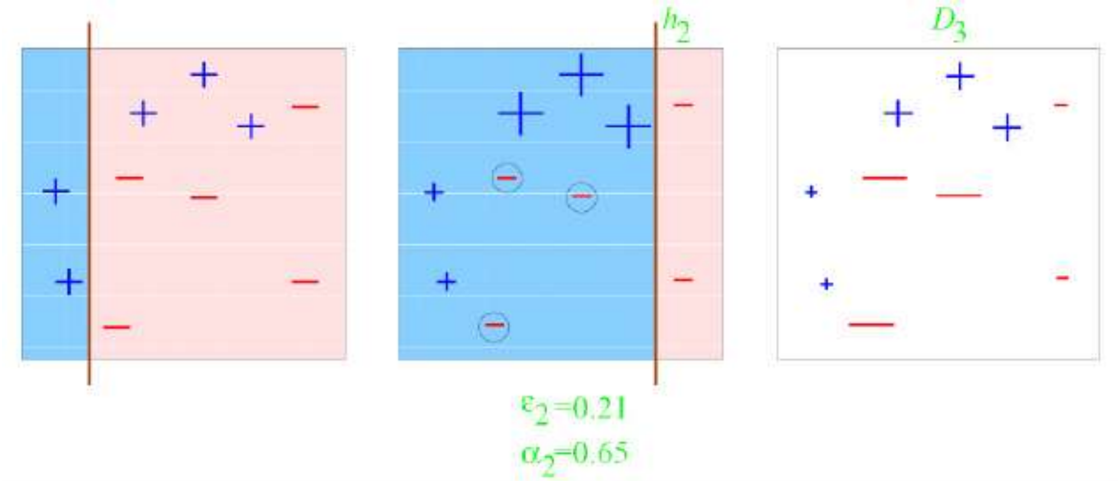
Training set: 10 points
(represented by plus or minus)
Original Status: Equal Weights
for all training samples

A toy example(cont'd)



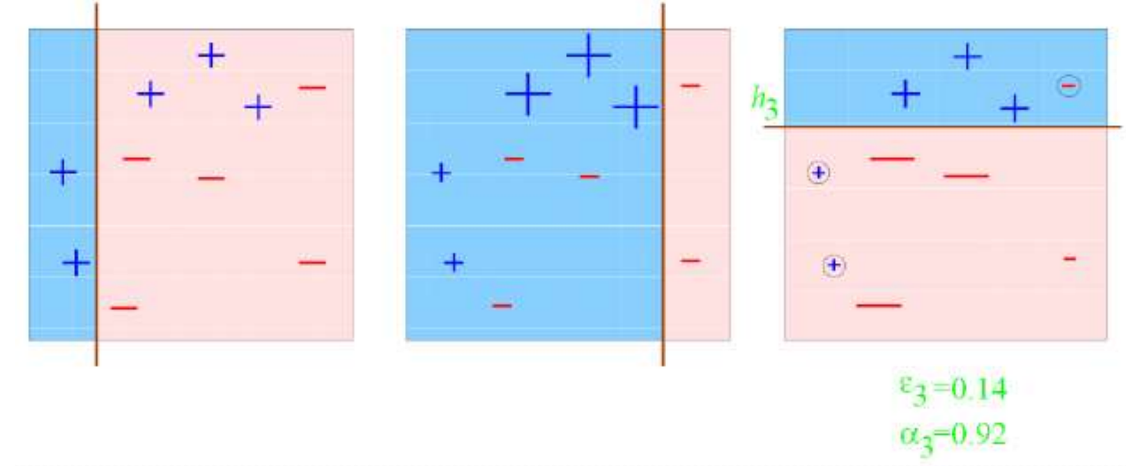
Round 1: Three "plus" points are not correctly classified;
They are given higher weights.

A toy example(cont'd)



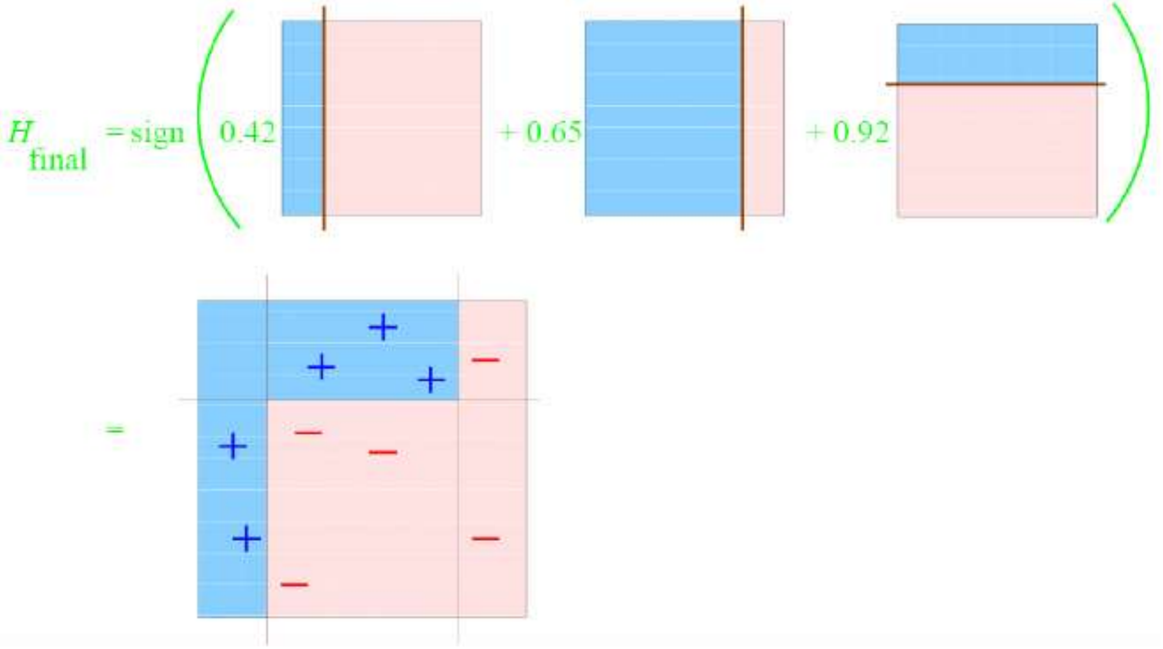
Round 2: Three "minus" points are not correctly classified;
They are given higher weights.

A toy example(cont'd)



Round 3: One "minus" and two "plus" points are not correctly classified;
They are given higher weights.

A toy example(cont'd)



Final Classifier: integrate the three "weak" classifiers and obtain a final strong classifier.

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in \mathscr{X}$, $y_i \in \{-1, +1\}$.
Initialize $\boxed{D_1(i) = 1/m \text{ for } i = 1, \ldots, m.}$ Initial Distribution of Data
For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t.
- Get weak hypothesis $h_t : \mathscr{X} \to \{-1, +1\}$.

Train model

- Aim: select h_t with low weighted error:

$$\boxed{\varepsilon_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i].}$$

Error of model

- Choose $\boxed{\alpha_t = \frac{1}{2} \ln\left(\frac{1 - \varepsilon_t}{\varepsilon_t}\right)}$.

Coefficient of model

- Update, for $i = 1, \ldots, m$:

$$\boxed{D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}}$$

Update Distribution

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

$$\boxed{H(x) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right).}$$

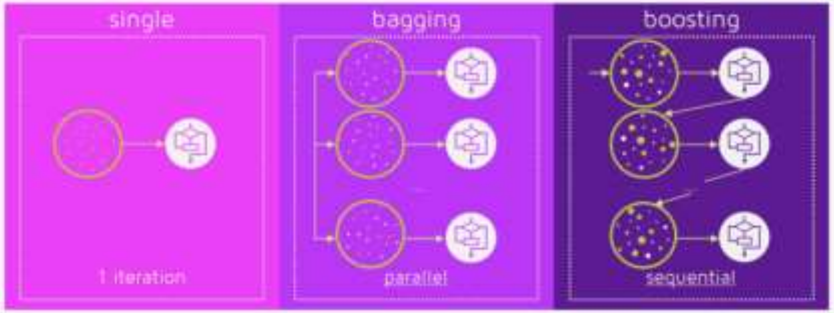
Final average

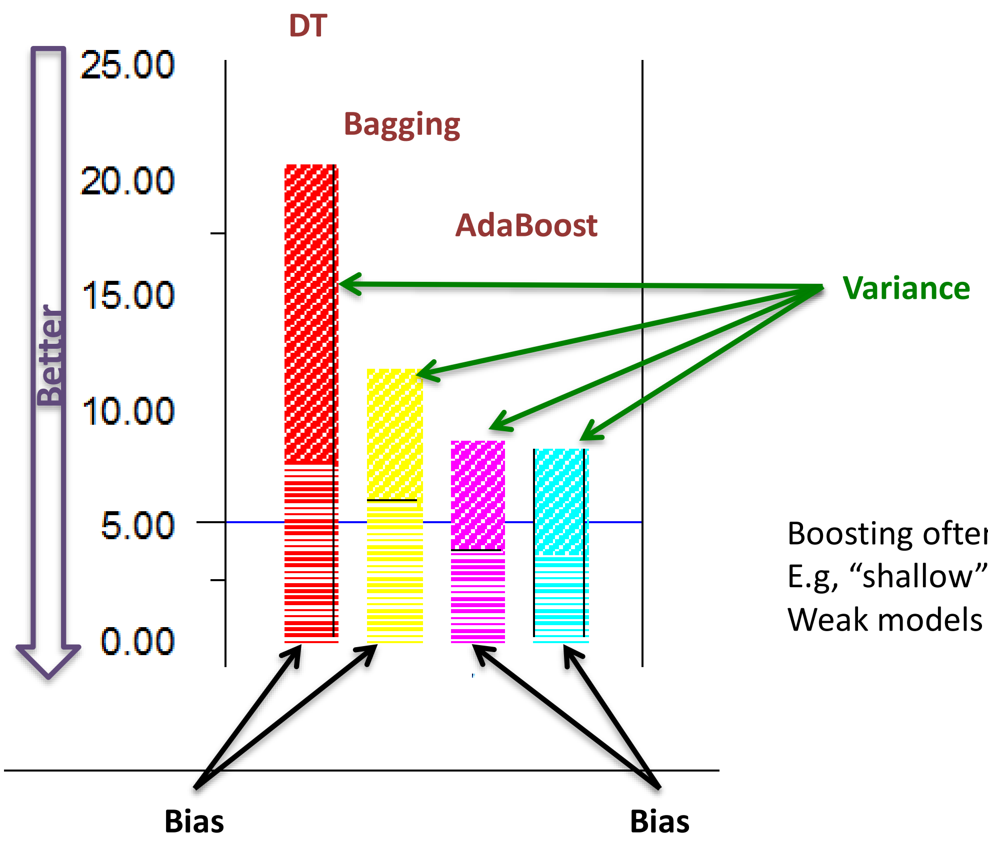
Theorem: training error drops exponentially fast

https://www.cs.princeton.edu/~schapire/papers/explaining-adaboost.pdf

Bagging *vs* Boosting

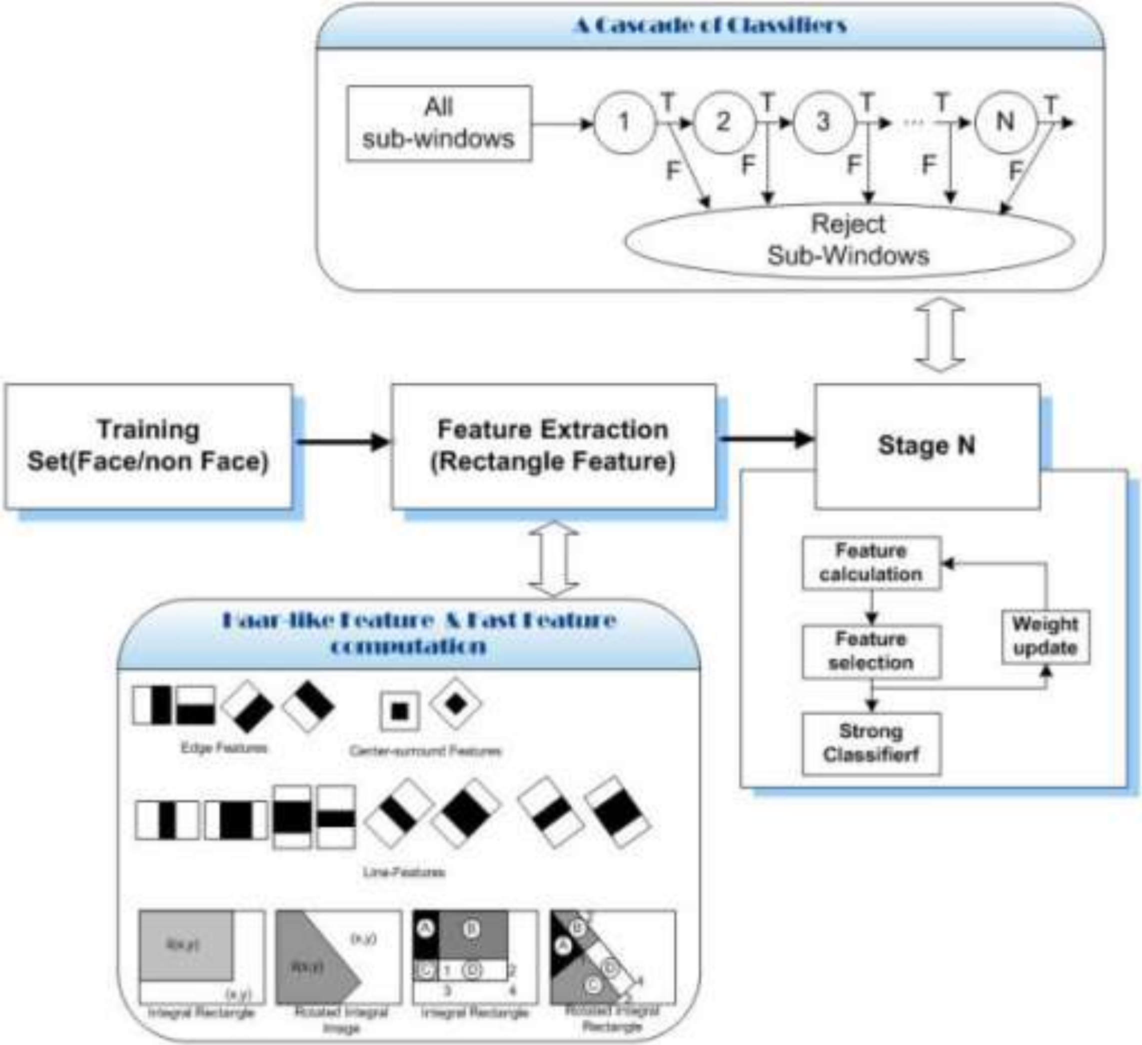
- Bagging
  - Construction of <u>complementary</u> base-learners is left to chance
  - .. and to the unstability of the learning methods.
- Boosting
  - Actively seek to generate complementary base-learner
  - Training the next base-learner based on the mistakes of the previous learners.

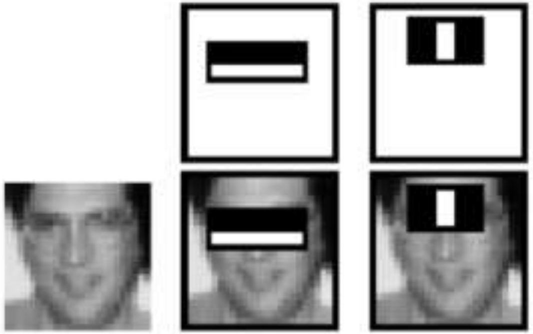


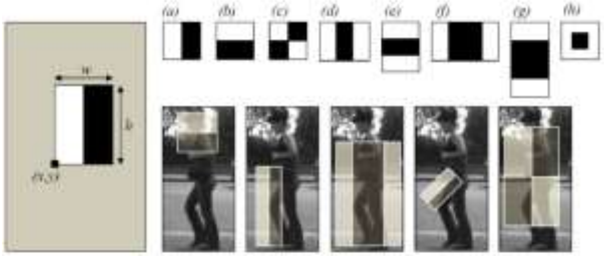


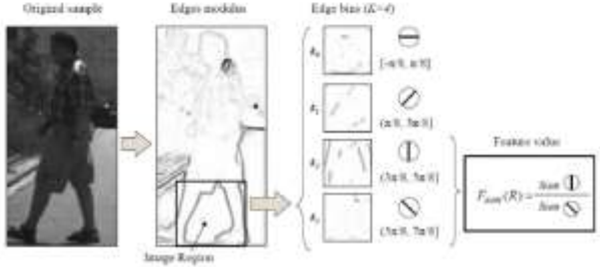
Boosting often uses weak models
E.g, “shallow” decision trees
Weak models have lower variance

“An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants”
Eric Bauer & Ron Kohavi, Machine Learning 36, 105–139 (1999)


A Cascade of Classifiers
All sub-windows
1
2
3
N
T
F
Reject Sub-Windows
Training Set(Face/non Face)
Feature Extraction (Rectangle Feature)
Stage N
Feature calculation
Feature selection
Weight update
Strong Classifierf
Haar-like Feature & Fast Feature computation
Edge Features
Center-surround Features
Line Features
Integral Rectangle
Rotated Integral Image
Integral Rectangle
Rotated Integral Rectangle




(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)


Original sample
Edges modulus
Feature value
Image Region

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in \mathscr{X}$, $y_i \in \{-1, +1\}$.

Initialize $\boxed{D_1(i) = 1/m \text{ for } i = 1, \ldots, m.}$ Initial Distribution of Data

For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t.
- Get weak hypothesis $h_t : \mathscr{X} \to \{-1, +1\}$.

Train model

- Aim: select h_t with low weighted error:

$$\boxed{\varepsilon_t = \Pr_{i \sim D_t}\left[h_t(x_i) \neq y_i\right].}$$

Error of model

- Choose $\boxed{\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \varepsilon_t}{\varepsilon_t} \right)}$. Coefficient of model
- Update, for $i = 1, \ldots, m$:

$$\boxed{D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}}$$

Update Distribution

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

$$\boxed{H(x) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x) \right).}$$

Final average

Adaboost: Additive model with exponential loss function

$$\min_{\alpha_{n=1:N}, \beta_{n=1:N}} L\left(y, \sum_{n=1}^{N} \alpha_n f(x, \beta_n)\right)$$