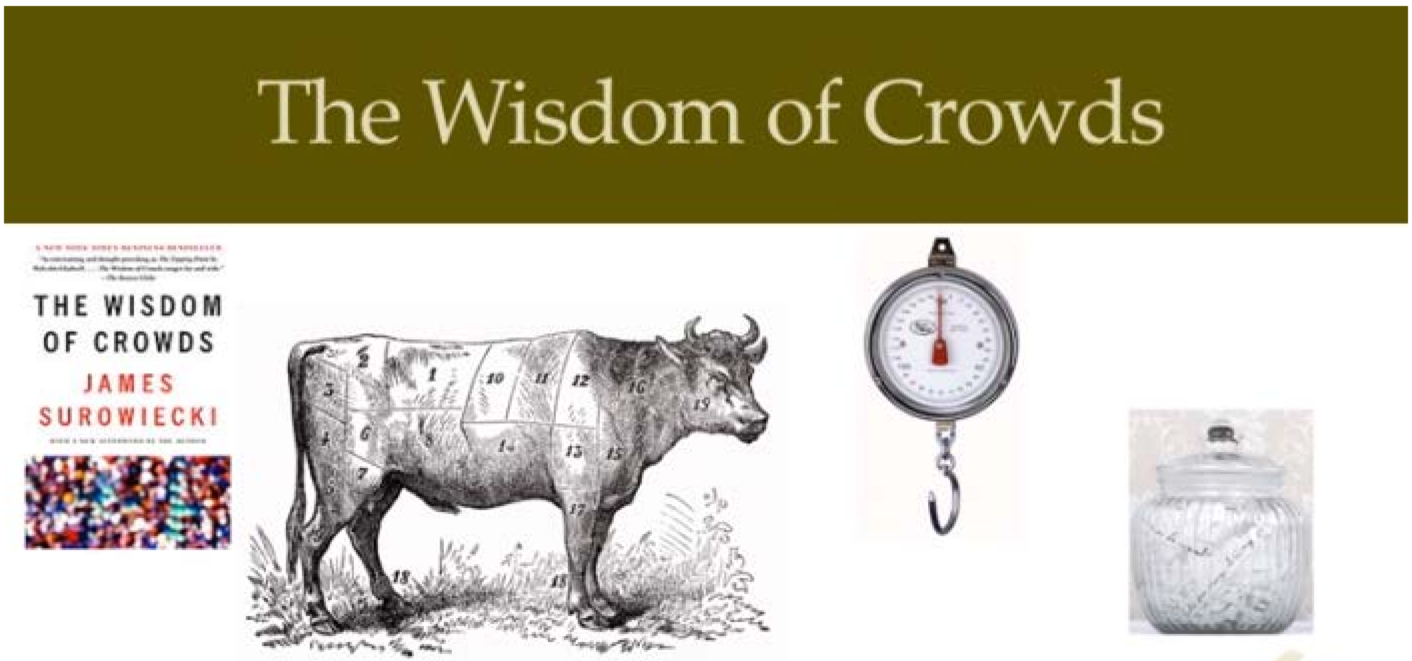
Ensemble Methods

averaging, bagging, boosting, random forests

“Wisdom of Crowds” (Francis Galton)

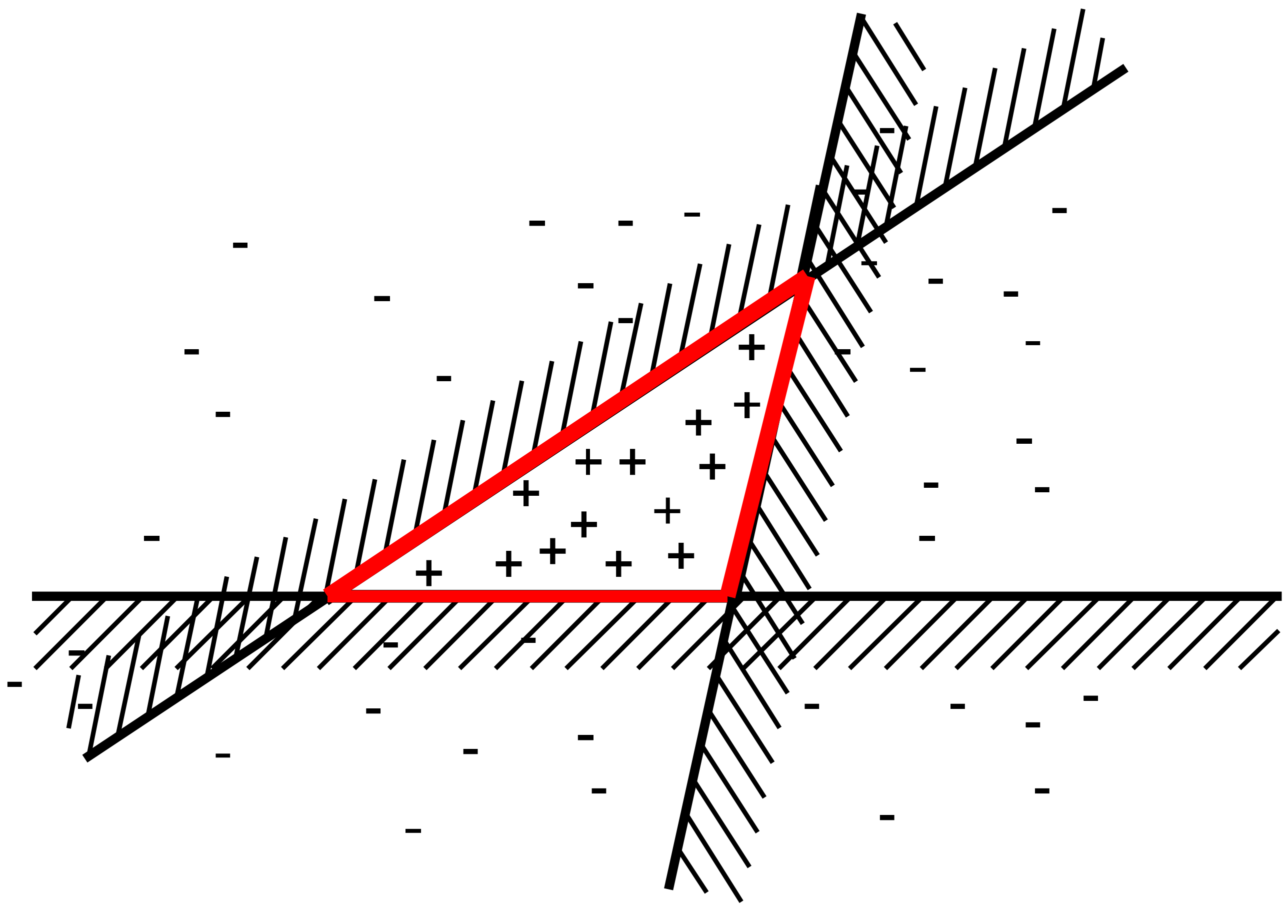
http://en.wikipedia.org/wiki/Wisdom_of_the_crowd

- Many idiots (“weak learners”) are often better than one expert



http://www.npr.org/2015/08/20/432978431/wighty-issue-cow-guessing-game-helps-to-explain-the-stock-market

Combination of several “decision stumps”



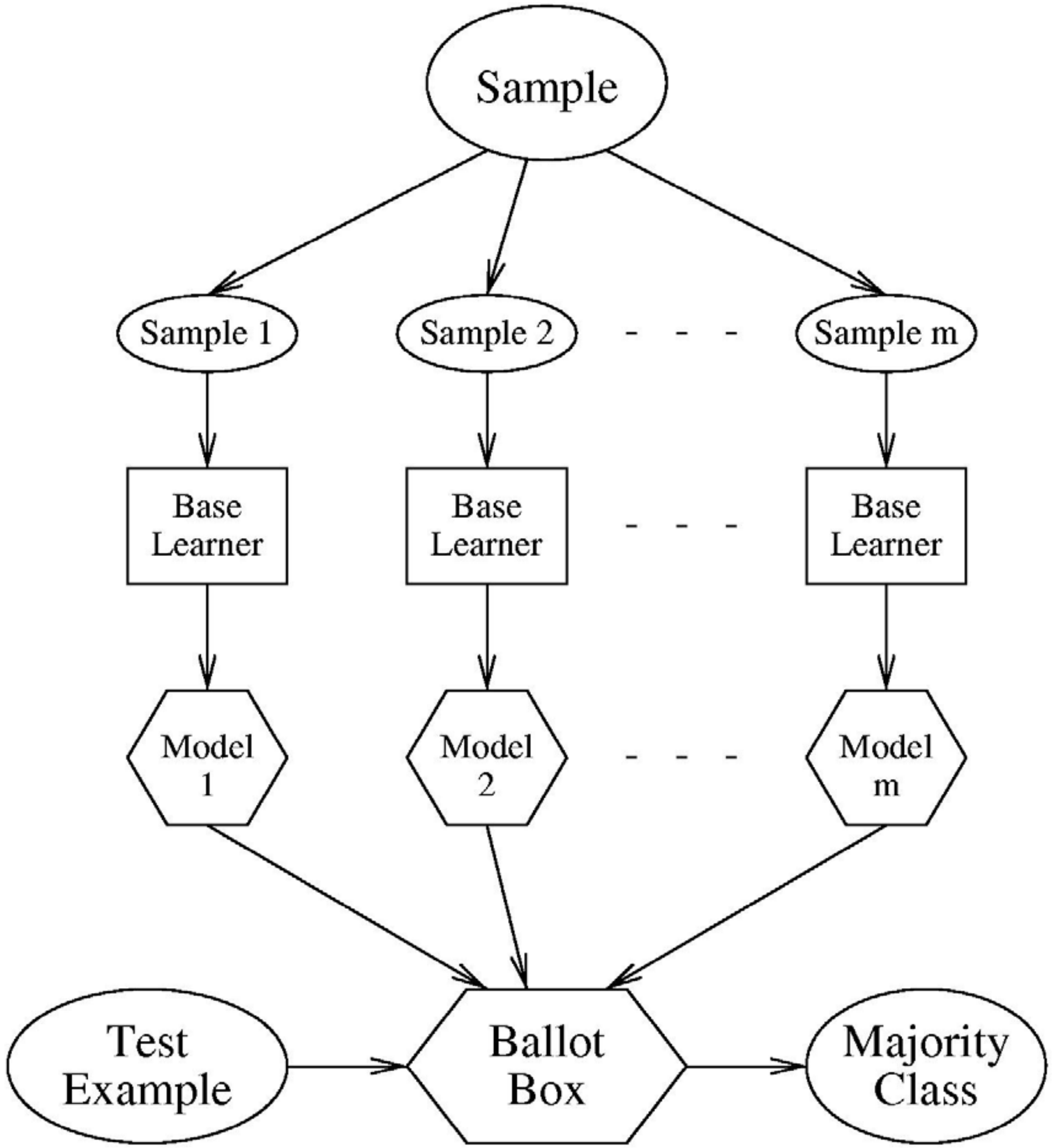
Ensemble Methods

- Instead of learning one model, learn several and combine. Different ways to get a set of models:
  - Averaging
    - **Randomize each model** (e.g. random initialization for gradient descent)
  - Bagging (Bootstrap Aggregation)
    - **Randomize the dataset** fed to a model
  - Random Forests
    - Do both
  - Boosting
    - Specialize each model for a subset of examples
- All can be applied on top of any “weak learner”, but particularly popular with decision trees/stumps

Bagging

- Generate "bootstrap" replicates of training set by sampling with replacement
- Learn one model on each replicate
- Combine by uniform voting

Q: How much data of the original dataset are in each replica?
A: About 63%

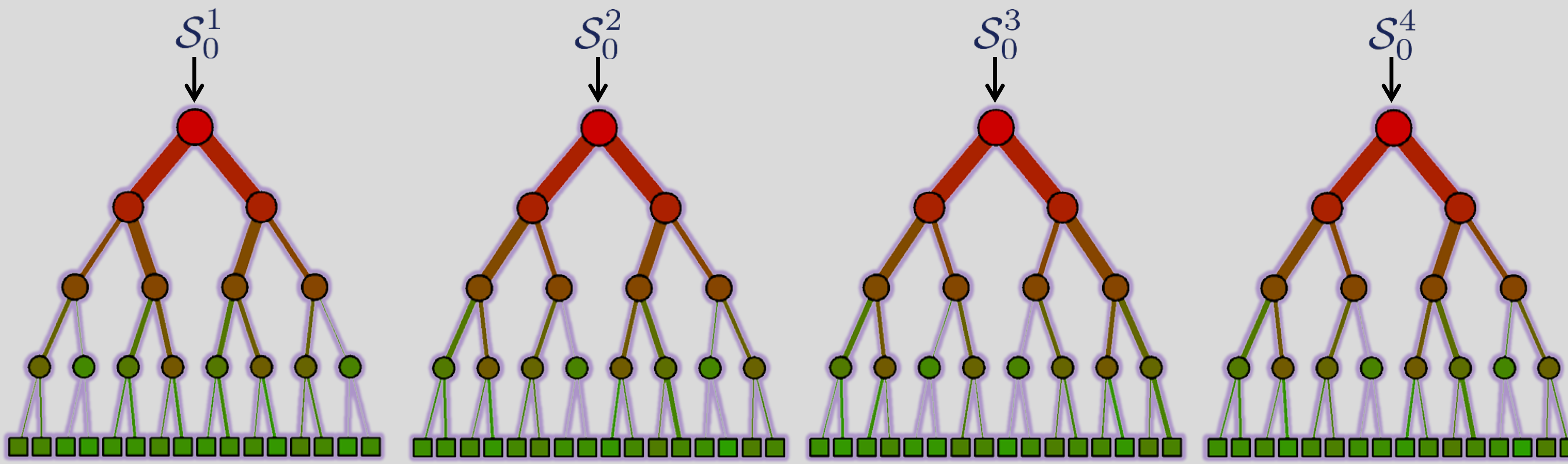

Sample
Sample 1
Sample 2
Sample m
Base Learner
Base Learner
Base Learner
Model 1
Model 2
Model m
Test Example
Ballot Box
Majority Class

Bagging on Trees

1) Bagging (randomizing the training set)

$\mathcal{S}_0$	The full training set
$\mathcal{S}_0^t \subset \mathcal{S}_0$	The randomly sampled subset of training data made available for the tree *t*

Forest training



Efficient training

Random Forests

- With bagging, often the trees look very correlated. Why?
- All trees pick the same (very good) splits
  - The trees become correlated, so averaging doesn't buy as much
- What can we do? Add more randomness:
  - at each node, only allow a random subset of ρ splits
  - Typically $\rho = \sqrt{|\mathcal{T}|}$

Decision forest model: the randomness model

2) Randomized node optimization (RNO)

$\mathcal{T}$ The full set of all possible node test parameters

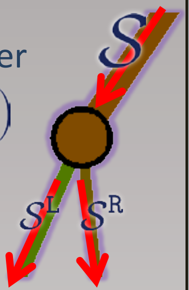
$\mathcal{T}_j \subset \mathcal{T}$ For each node the set of randomly sampled features

$\rho = |\mathcal{T}_j|$ Randomness control parameter.
For $\rho = |\mathcal{T}|$ no randomness and maximum tree correlation.
For $\rho = 1$ max randomness and minimum tree correlation.

Node training

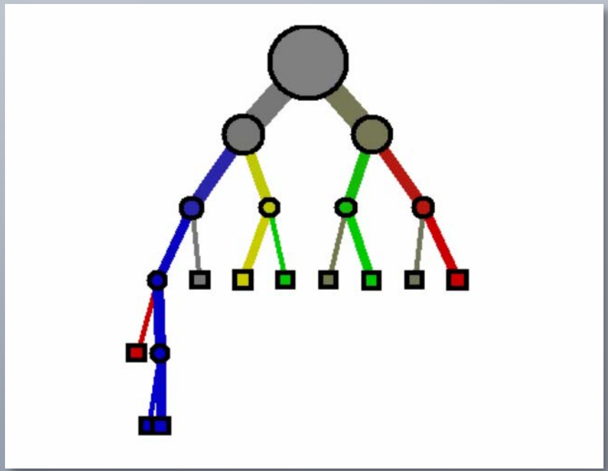
Node weak learner
$h(\mathbf{v}, \boldsymbol{\theta}_j)$

Node test params
$\boldsymbol{\theta} \in \mathcal{T}_j$

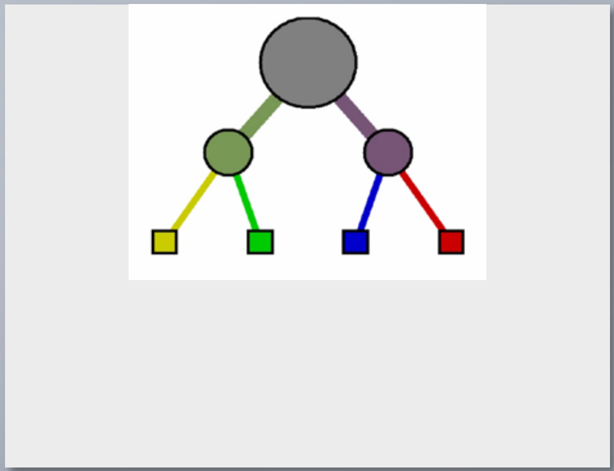


The effect of ρ

Small value of ρ; little tree correlation.



Large value of ρ; large tree correlation.



Classification forest: the weak learner model

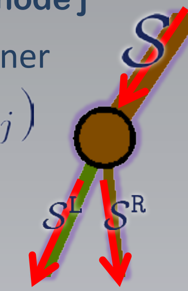
Splitting data at node j

Node weak learner

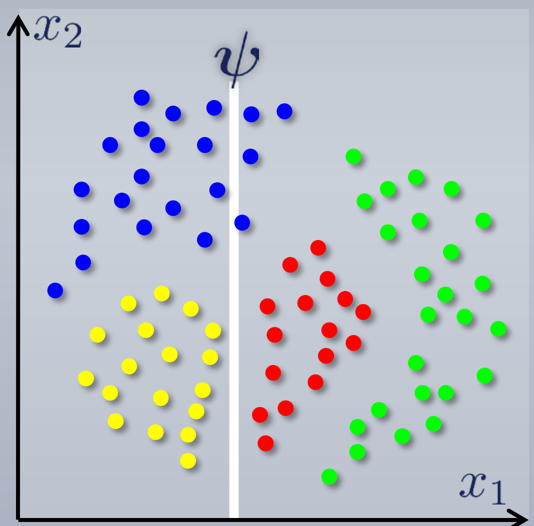
$$h(\mathbf{v}, \boldsymbol{\theta}_j)$$

Node test params

$$\boldsymbol{\theta} \in \mathcal{T}_j$$



Examples of weak learners

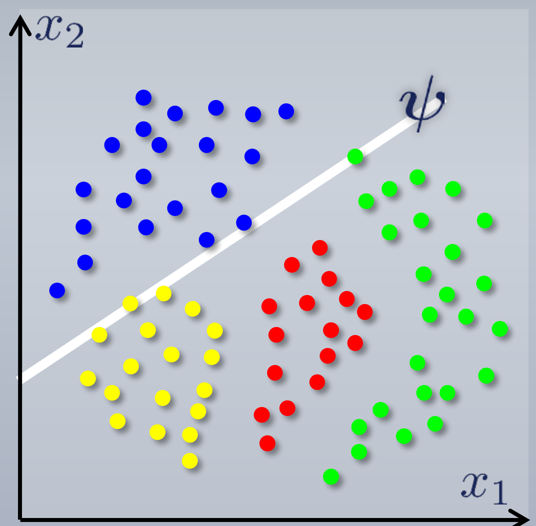


Weak learner: axis aligned

$$h(\mathbf{v}, \boldsymbol{\theta}) = [\tau_1 > \boldsymbol{\phi}(\mathbf{v}) \cdot \boldsymbol{\psi} > \tau_2]$$

Feature response for 2D example. $\boldsymbol{\phi}(\mathbf{v}) = (x_1 \ x_2 \ 1)^\top$

With $\boldsymbol{\psi} = (1 \ 0 \ \psi_3)$ or $\boldsymbol{\psi} = (0 \ 1 \ \psi_3)$

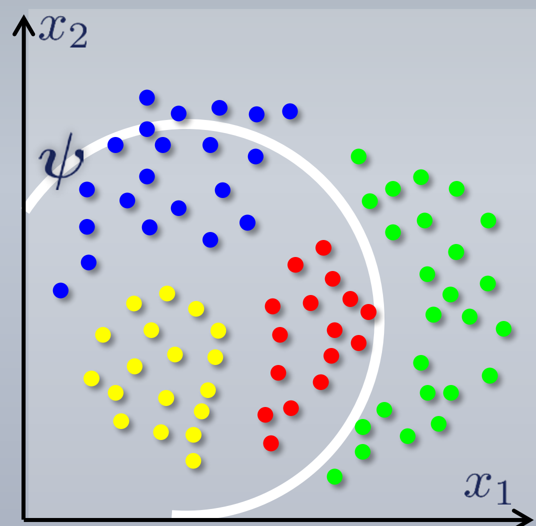


Weak learner: oriented line

$$h(\mathbf{v}, \boldsymbol{\theta}) = [\tau_1 > \boldsymbol{\phi}(\mathbf{v}) \cdot \boldsymbol{\psi} > \tau_2]$$

Feature response for 2D example. $\boldsymbol{\phi}(\mathbf{v}) = (x_1 \ x_2 \ 1)^\top$

With $\boldsymbol{\psi} \in \mathbb{R}^3$ a generic line in homog. coordinates.



Weak learner: conic section

$$h(\mathbf{v}, \boldsymbol{\theta}) = \left[\tau_1 > \boldsymbol{\phi}^\top(\mathbf{v}) \ \boldsymbol{\psi} \ \boldsymbol{\phi}(\mathbf{v}) > \tau_2\right]$$

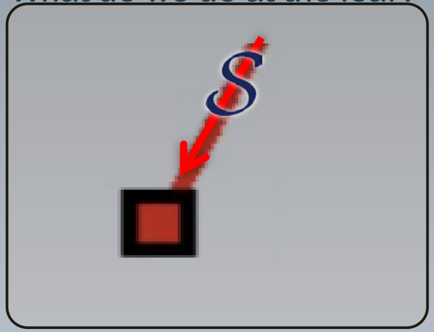
Feature response for 2D example. $\boldsymbol{\phi}(\mathbf{v}) = (x_1 \ x_2 \ 1)^\top$

With $\boldsymbol{\psi} \in \mathbb{R}^{3\times 3}$ a matrix representing a conic.

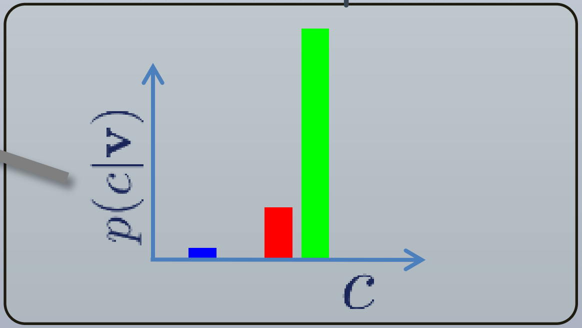
In general $\boldsymbol{\phi}$ may select only a very small subset of features $\boldsymbol{\phi}(\mathbf{v}) : \mathbb{R}^d \rightarrow \mathbb{R}^{d'+1}, \quad d' << d$

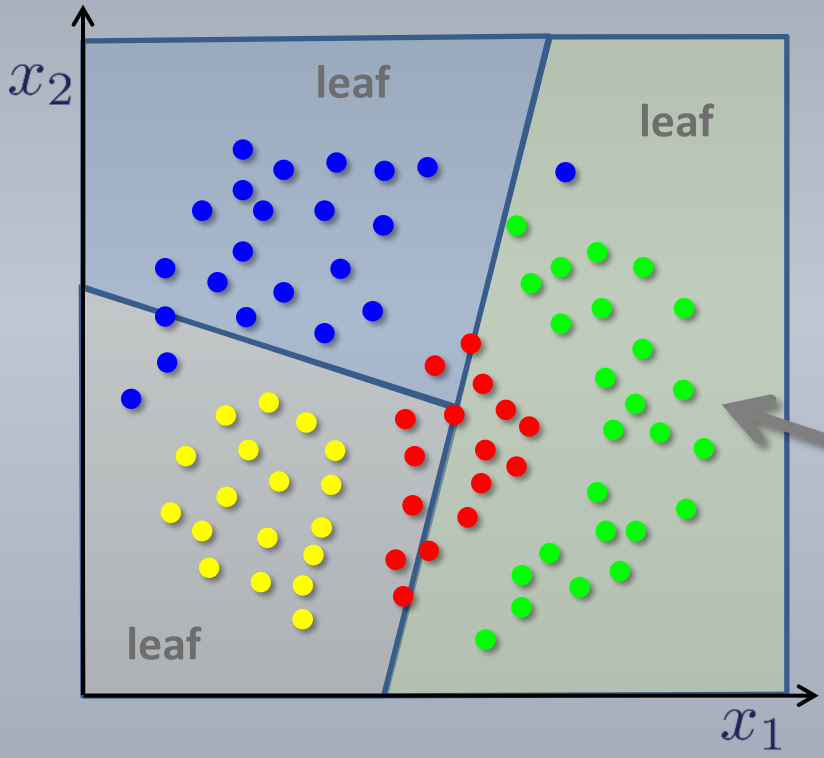
Classification forest: the prediction model

What do we do at the leaf?

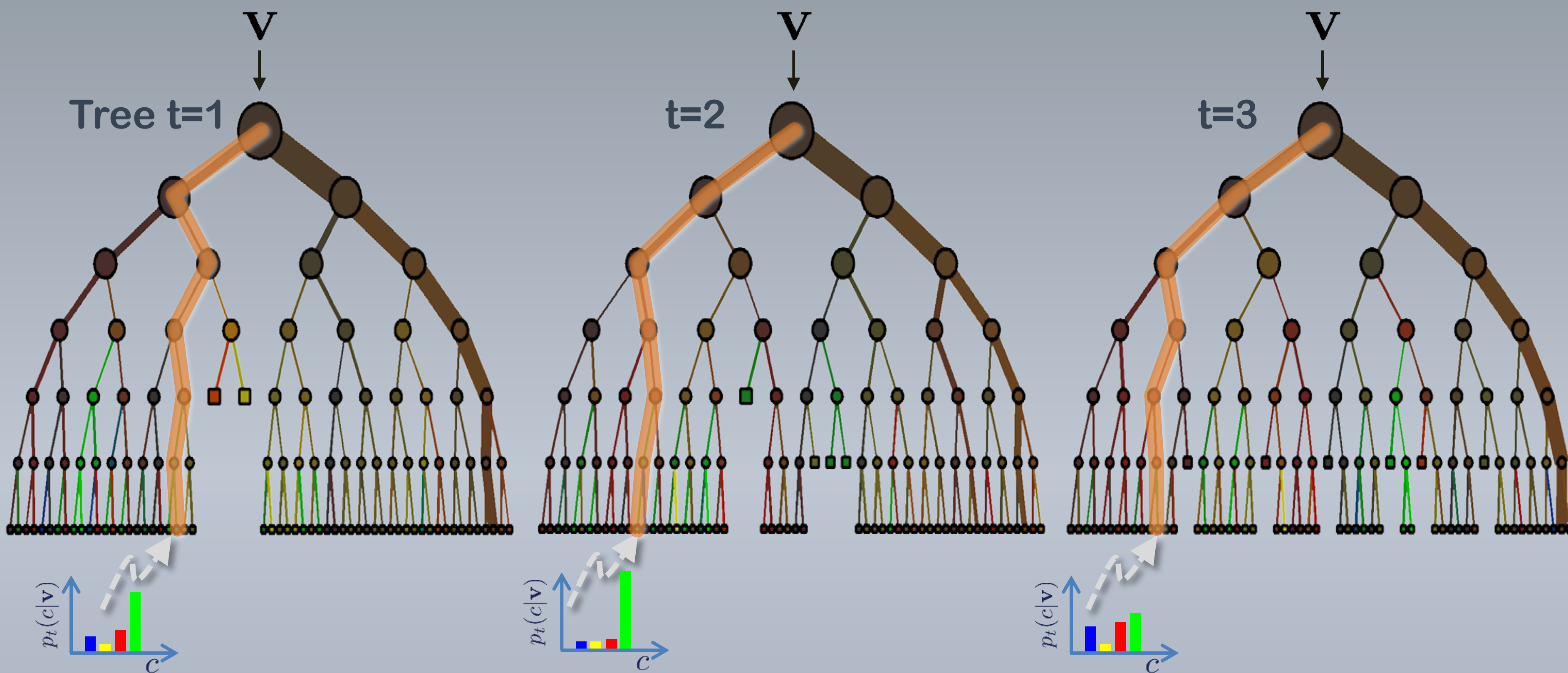


Prediction model: probabilistic



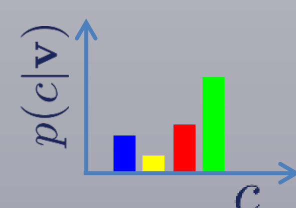


Classification forest: the ensemble model

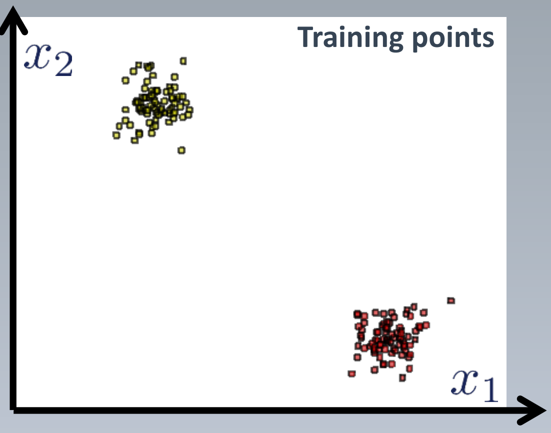


The ensemble model

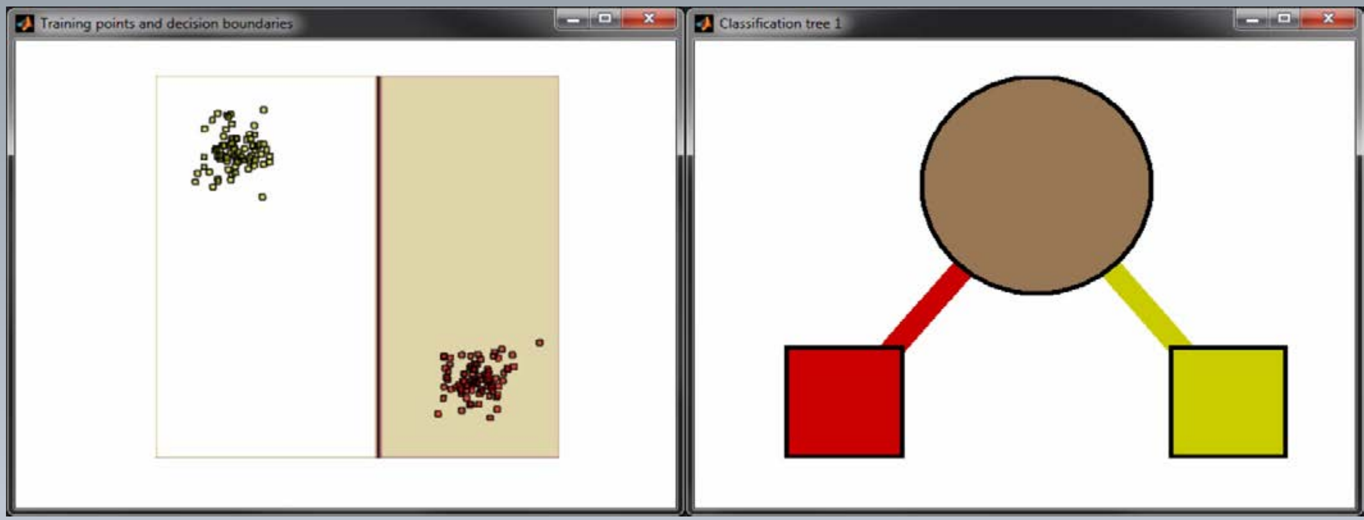
Forest output probability $$p(c|\mathbf{v}) = \frac{1}{T}\sum_{t}^{T} p_t(c|\mathbf{v})$$



Classification forest: effect of the weak learner model



Training different trees in the forest



Testing different trees in the forest

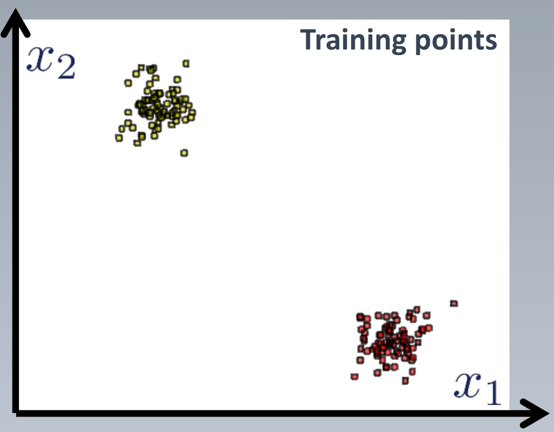
Three concepts to keep in mind:

- "Accuracy of prediction"
- "Quality of confidence"
- "Generalization"

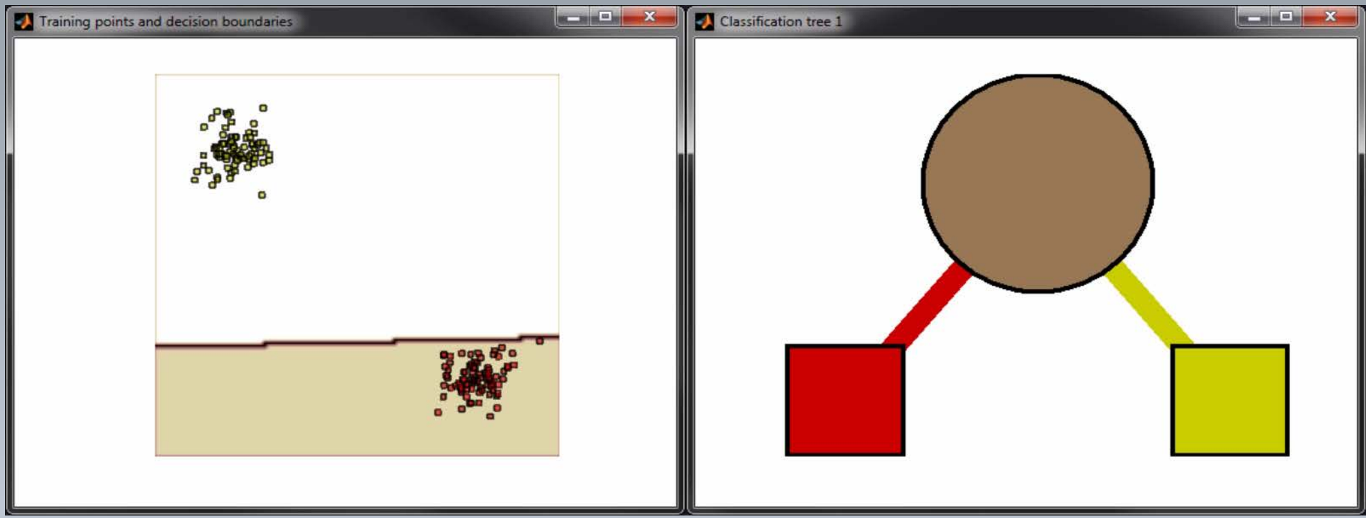
(2 videos in this page)

Parameters: T=200, D=2, weak learner = aligned, leaf model = probabilistic

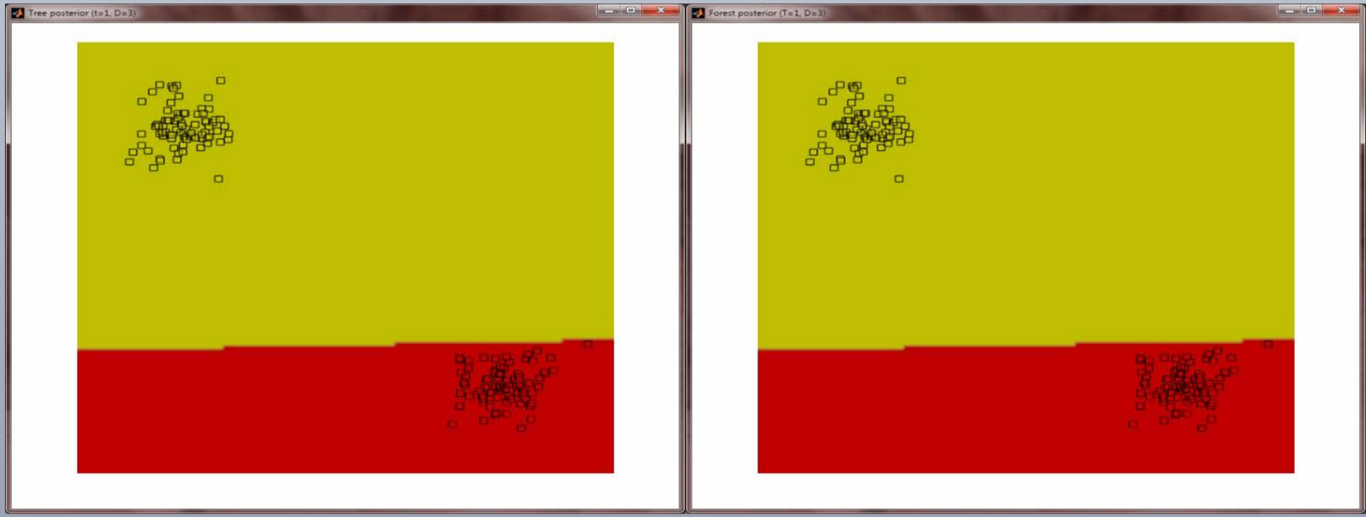
Classification forest: effect of the weak learner model



Training different trees in the forest



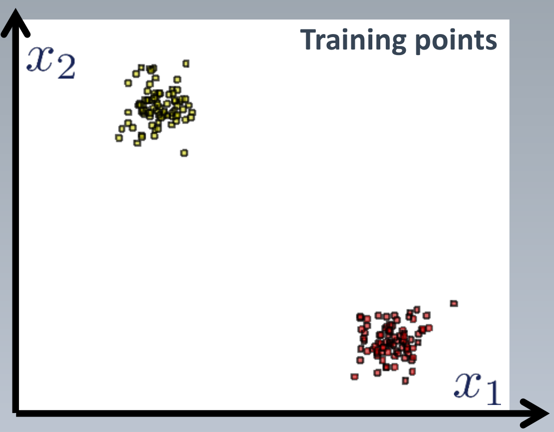
Testing different trees in the forest



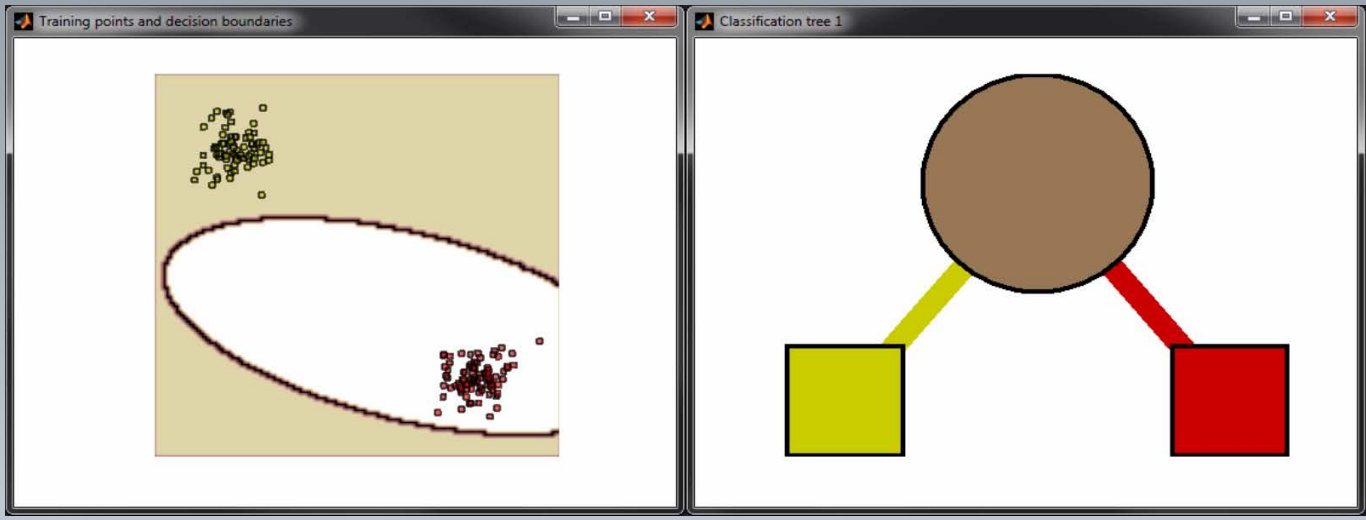
(2 videos in this page)

Parameters: T=200, D=2, weak learner = linear, leaf model = probabilistic

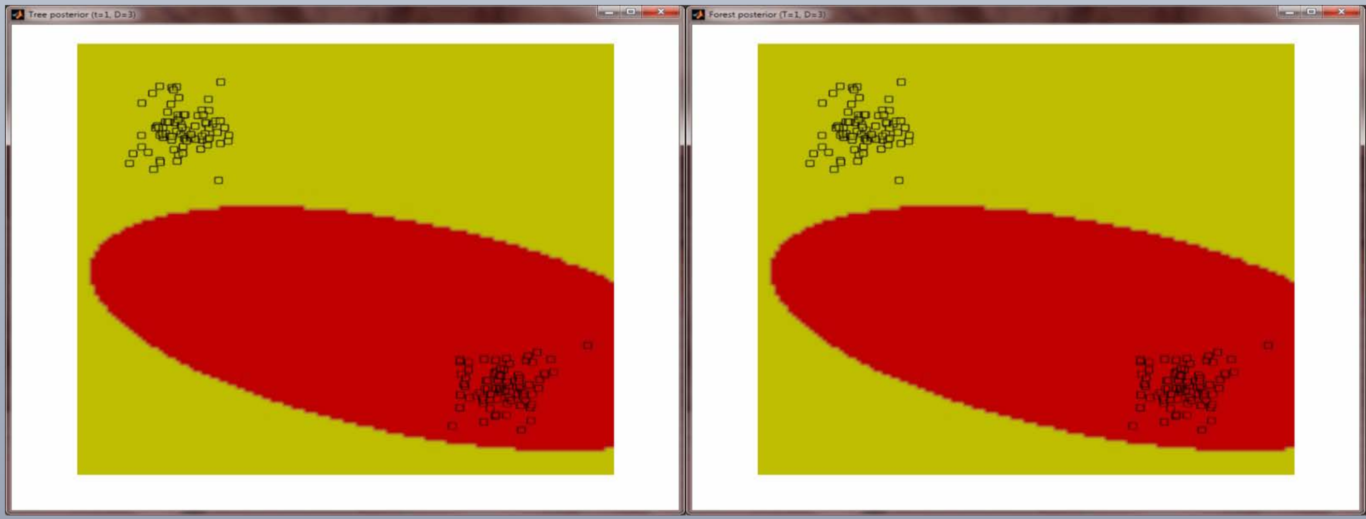
Classification forest: effect of the weak learner model



Training different trees in the forest



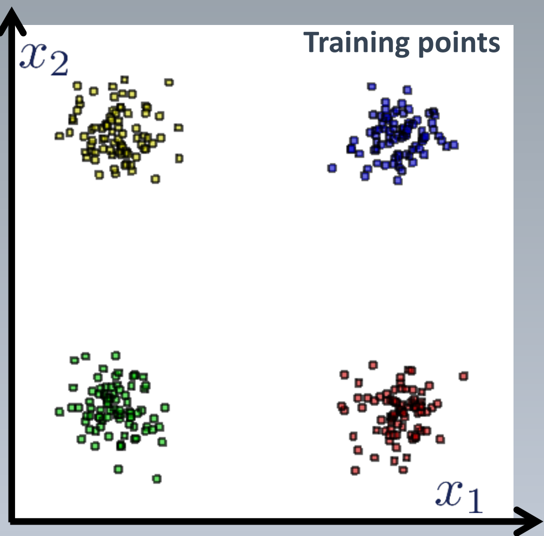
Testing different trees in the forest



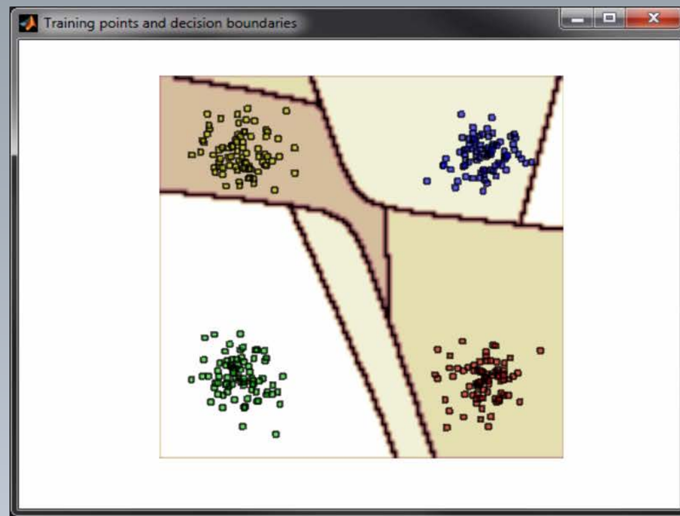
(2 videos in this page)

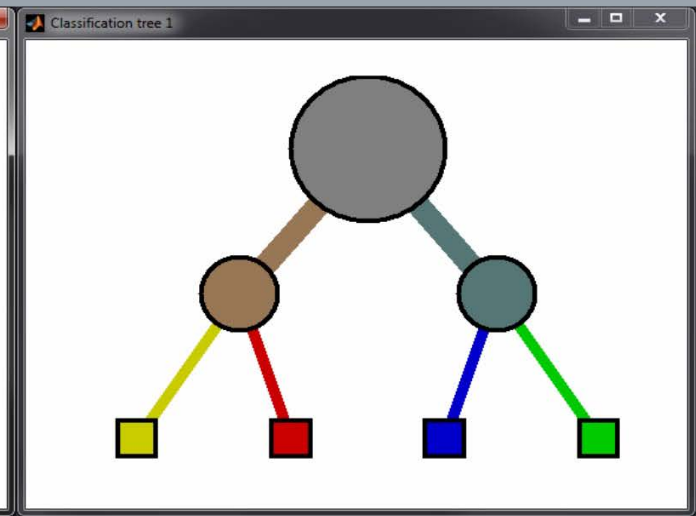
Parameters: T=200, D=2, weak learner = conic, leaf model = probabilistic

Classification forest: with >2 classes

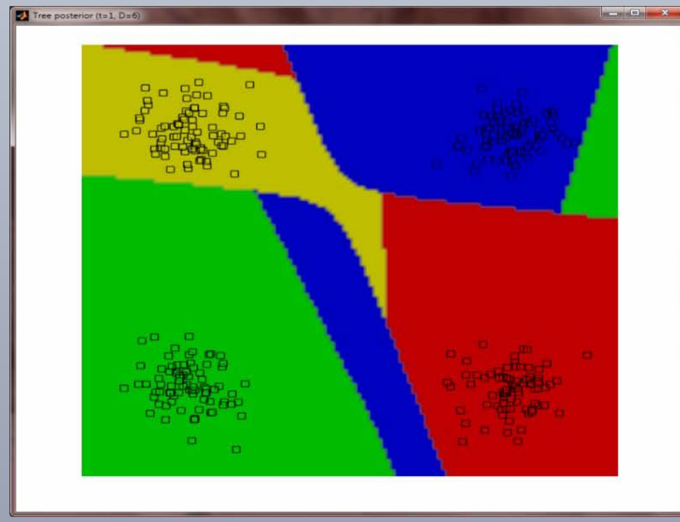


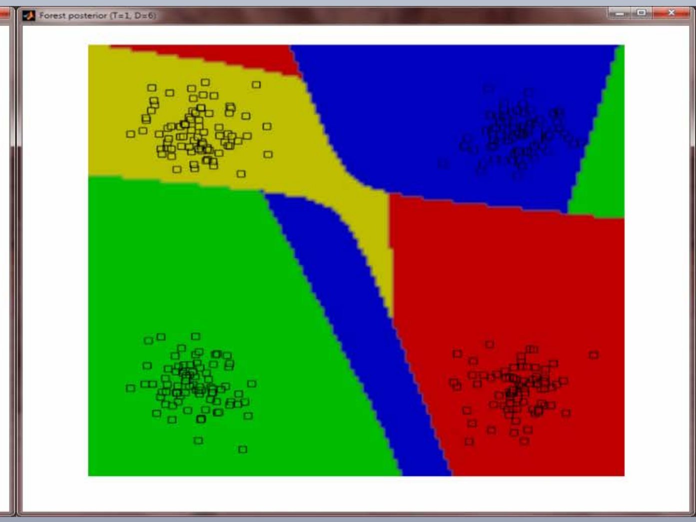
Training different trees in the forest





Testing different trees in the forest

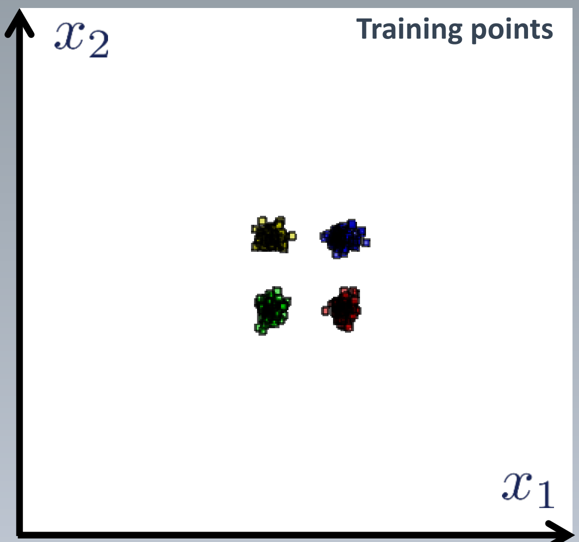




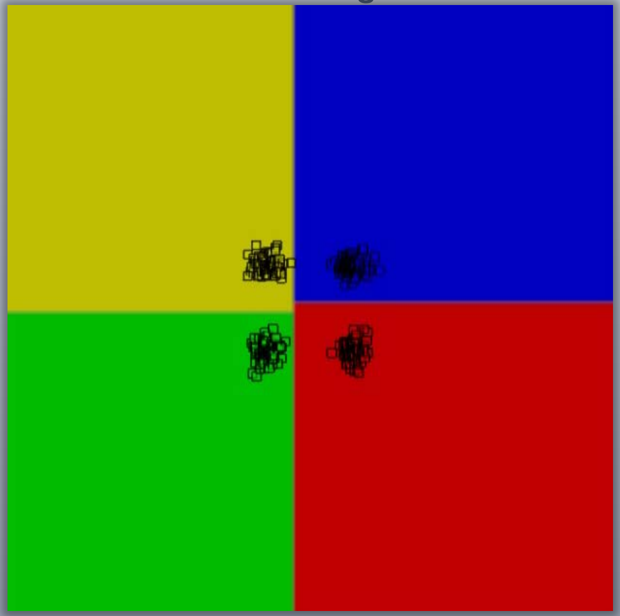
(2 videos in this page)

Parameters: T=200, D=3, weak learner = conic, leaf model = probabilistic

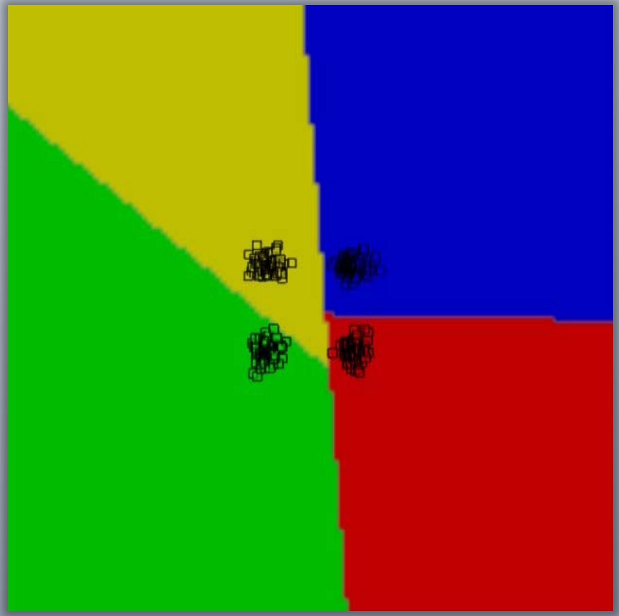
Classification forest: analysing generalization



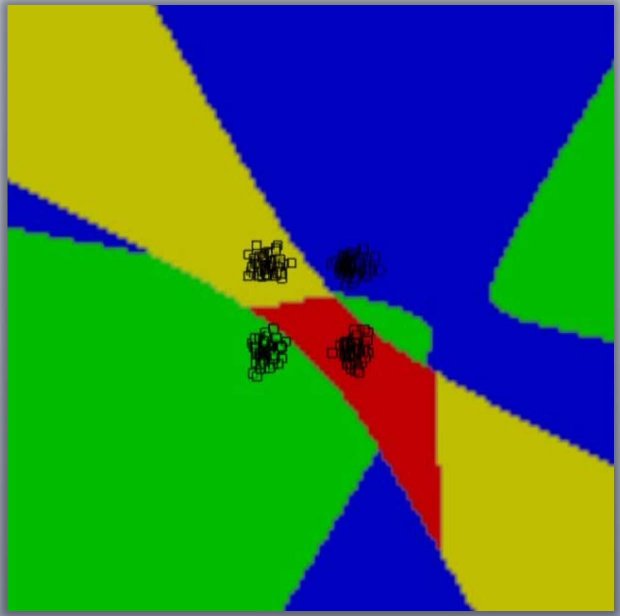
Weak learner: axis aligned



Weak learner: oriented line



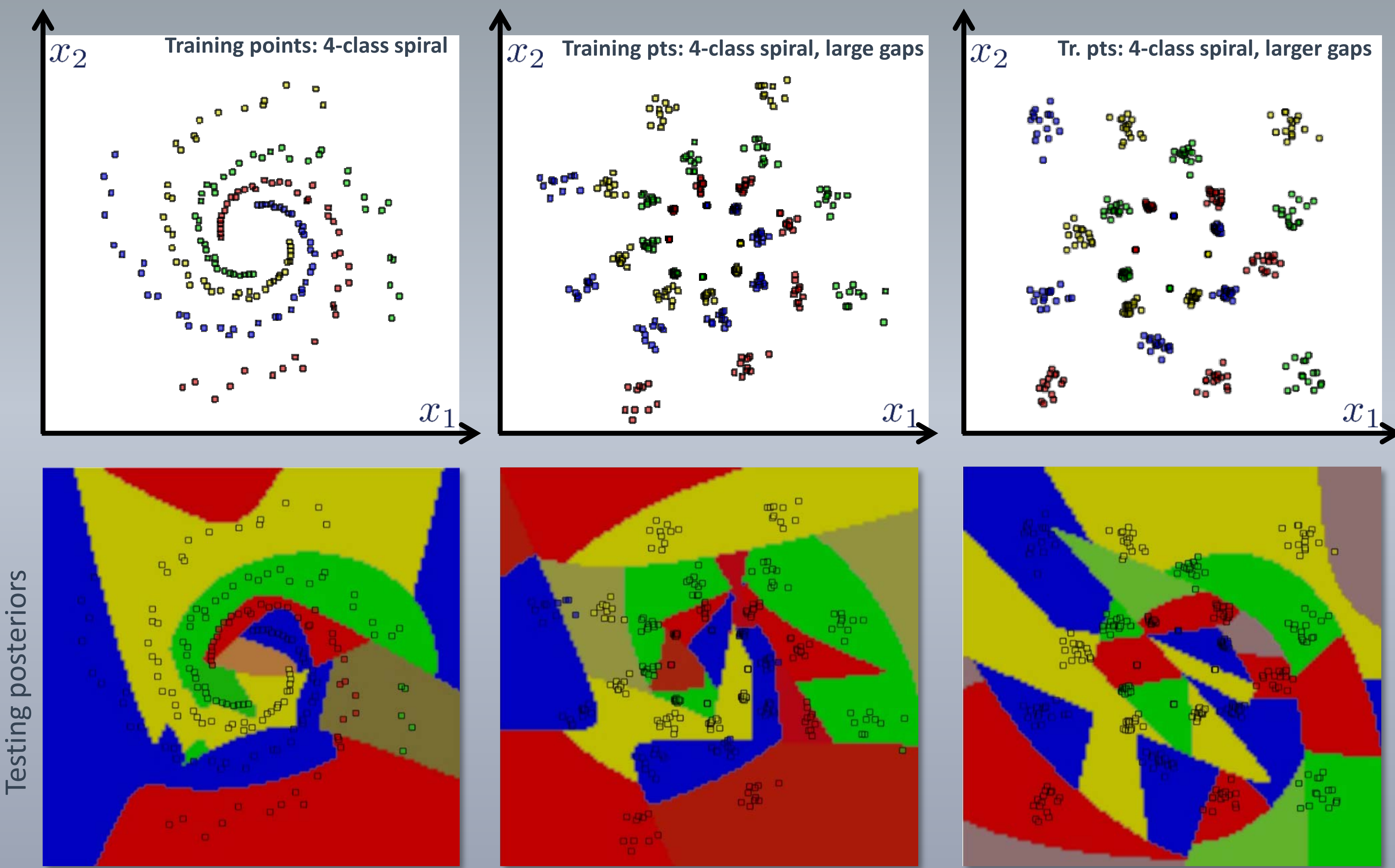
Weak learner: conic section



(3 videos in this page. Increasing T)

Parameters: T=200, D=3, leaf model = probabilistic

Classification forest: analysing generalization

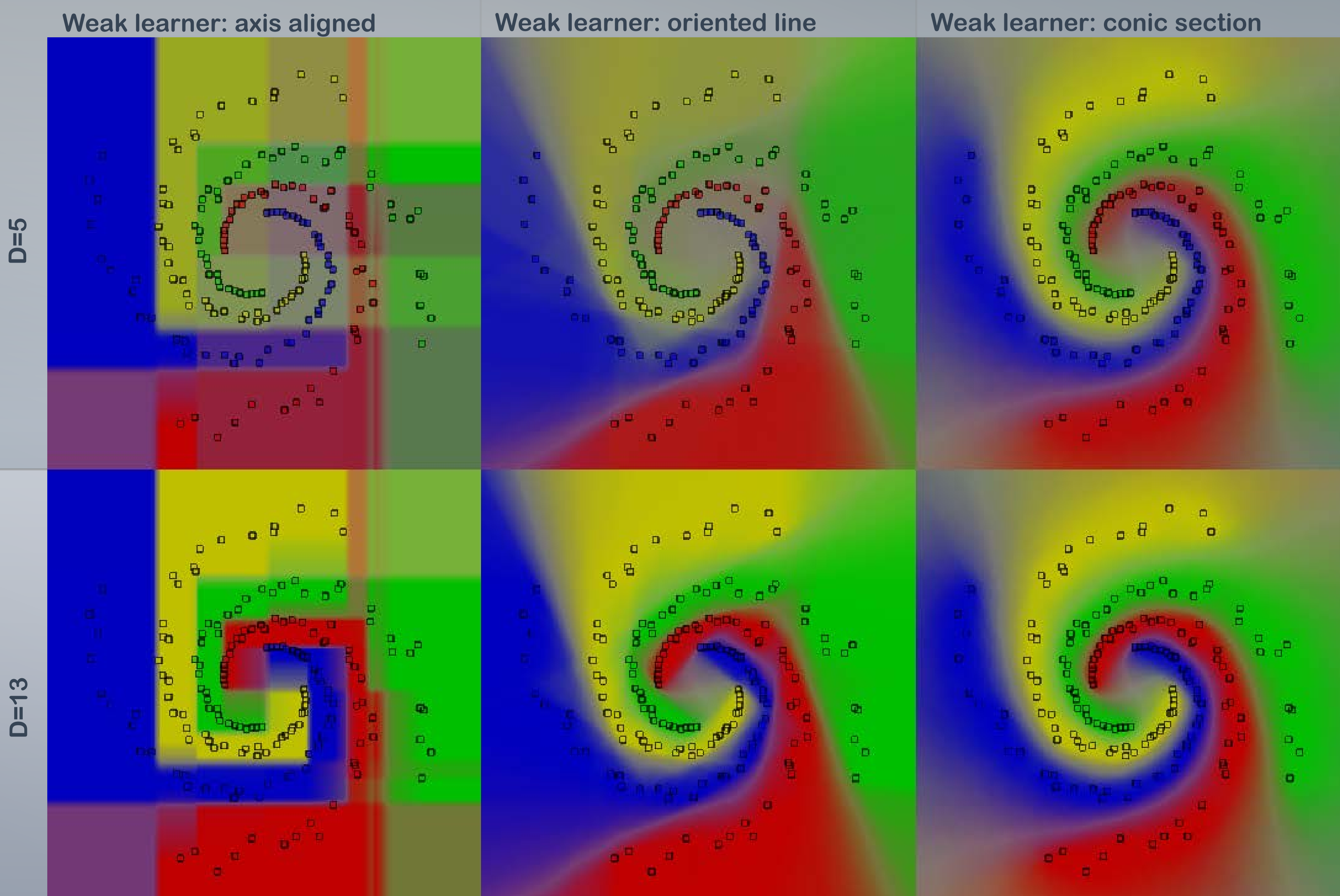


(3 videos in this page)

Parameters: T=200, D=13, w. l. = conic, predictor = prob.

Classification forest: effect of weak learner model and randomness

Testing posteriors

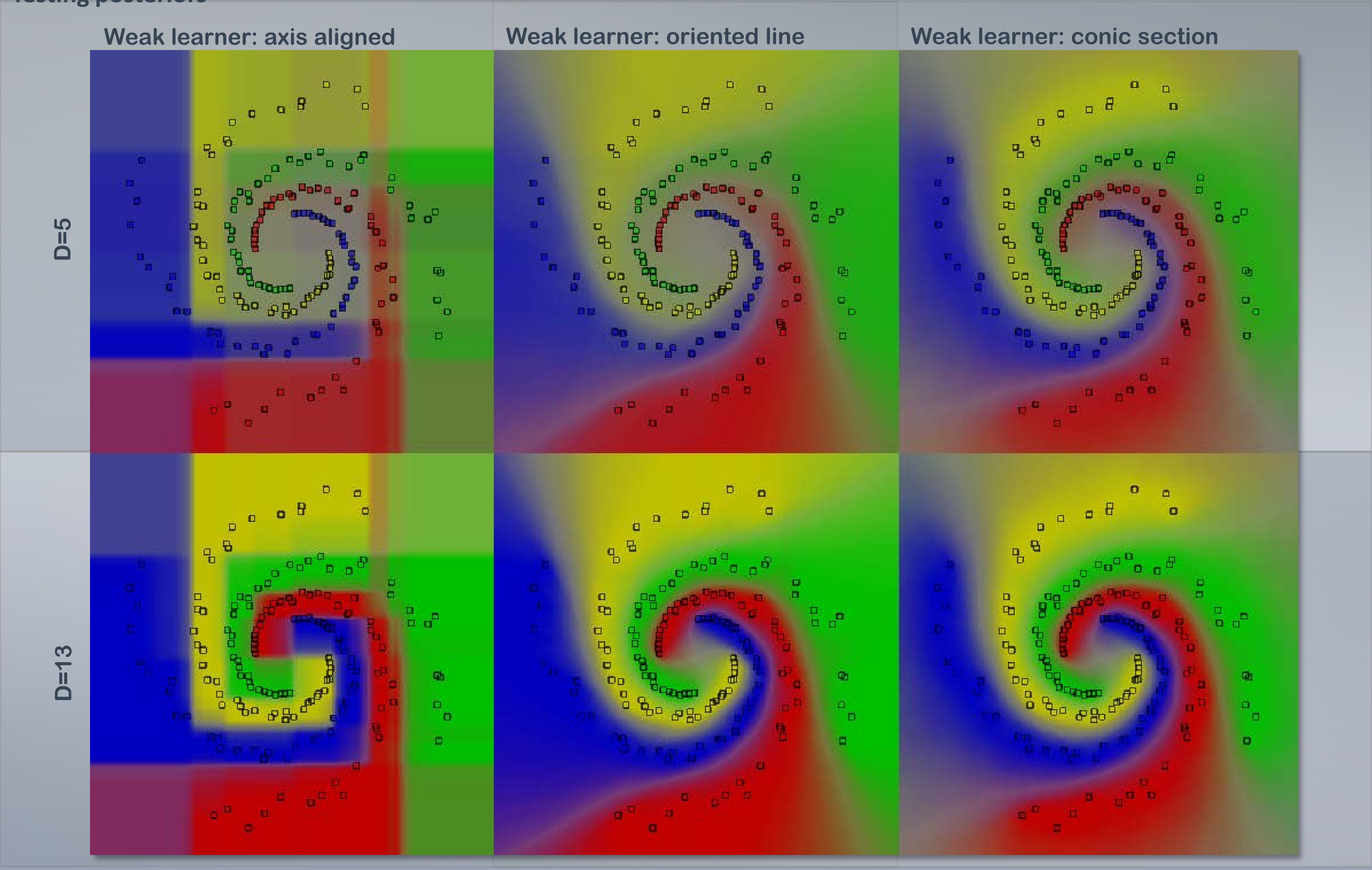


Randomness: $\rho = 500$

Parameters: T=400 predictor model = prob.

Classification forest: effect of weak learner model and randomness

Testing posteriors



Randomness: $\rho = 50$

Parameters: T=400 predictor model = prob.

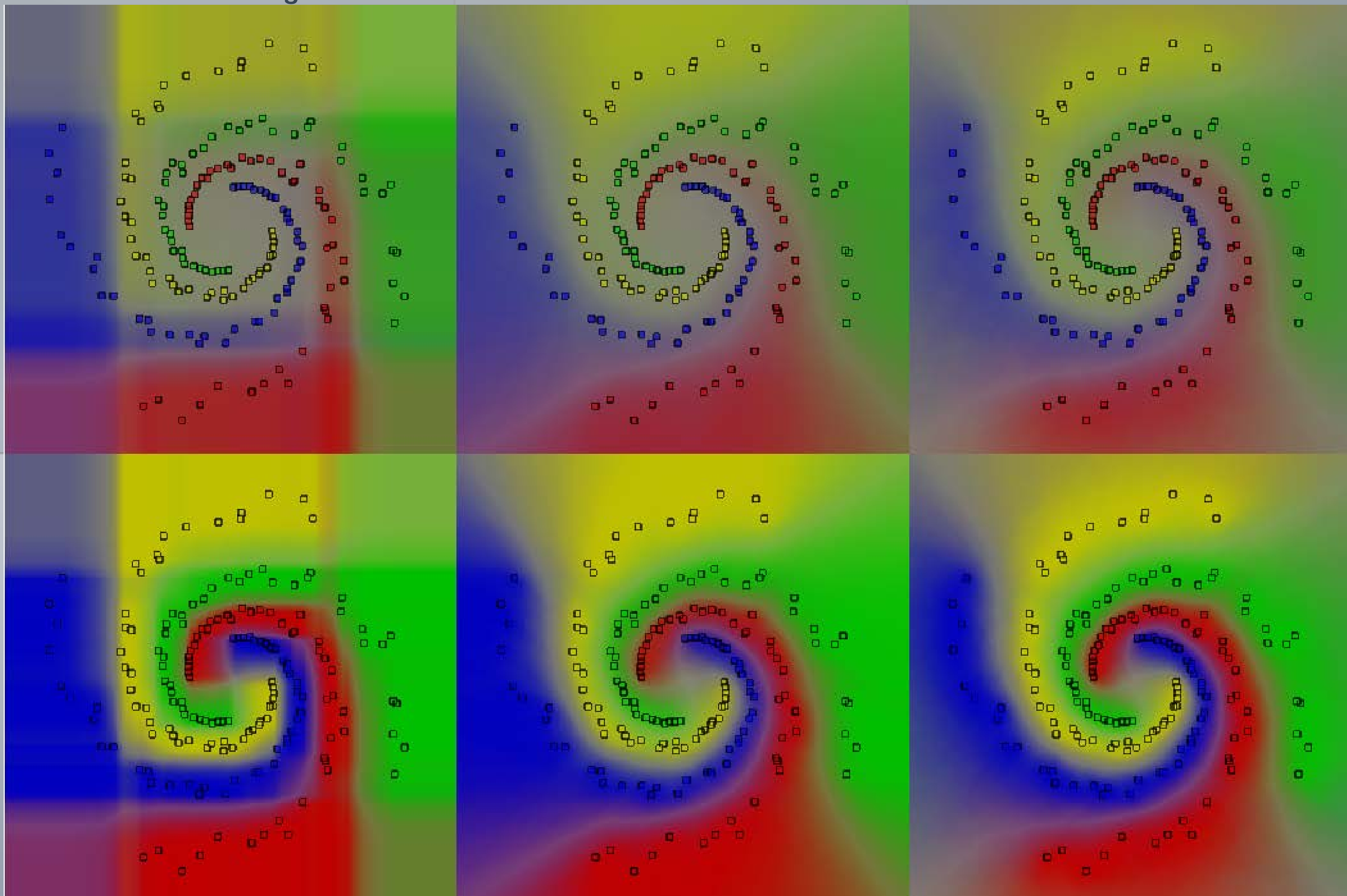
Classification forest: effect of weak learner model and randomness

Testing posteriors

Weak learner: axis aligned | Weak learner: oriented line | Weak learner: conic section

D=5

D=13

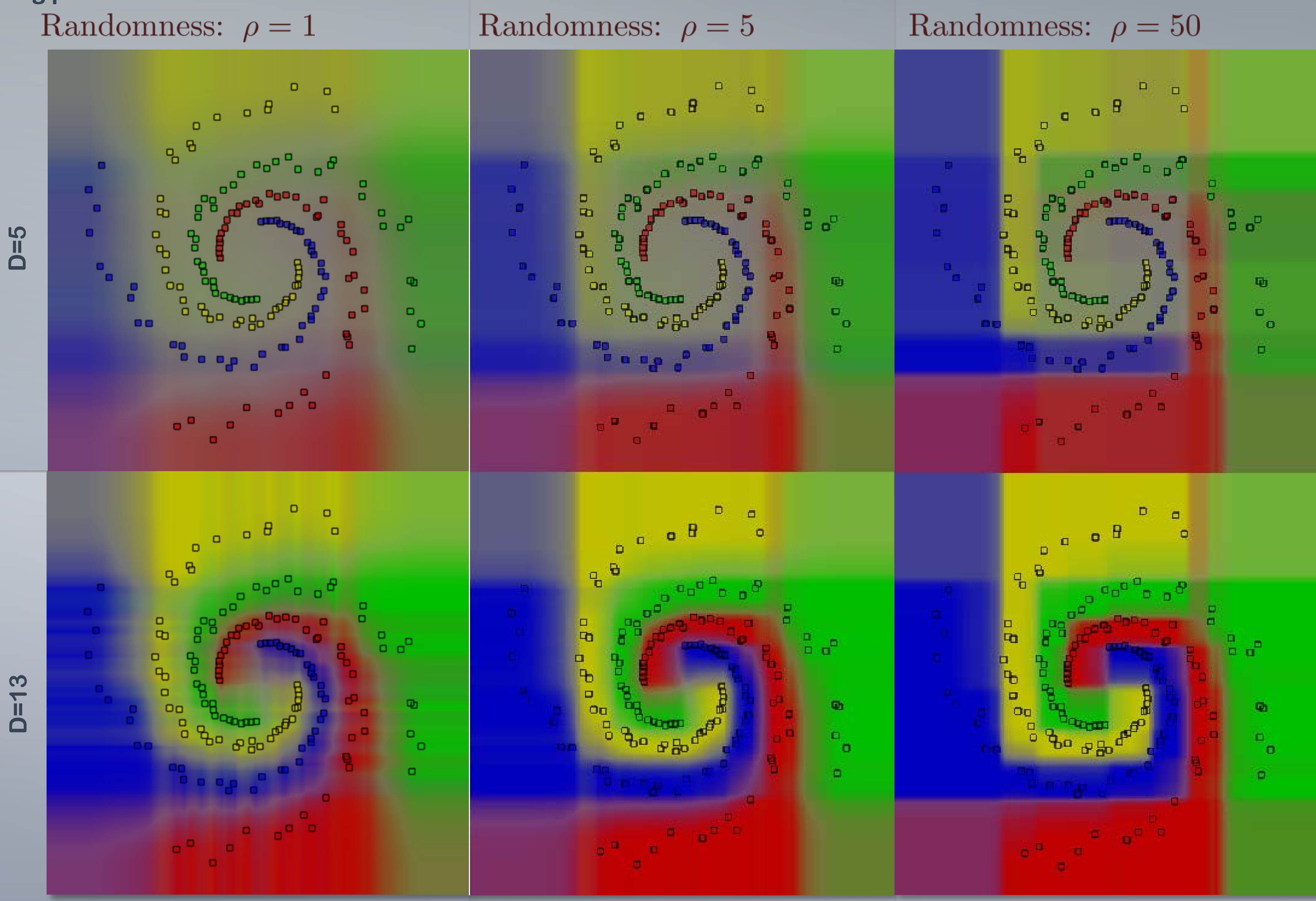


Randomness: $\rho = 5$

Parameters: T=400 predictor model = prob.

Classification forest: effect of randomness

Testing posteriors



Weak learner: axis aligned

Parameters: T=400 predictor model = prob.

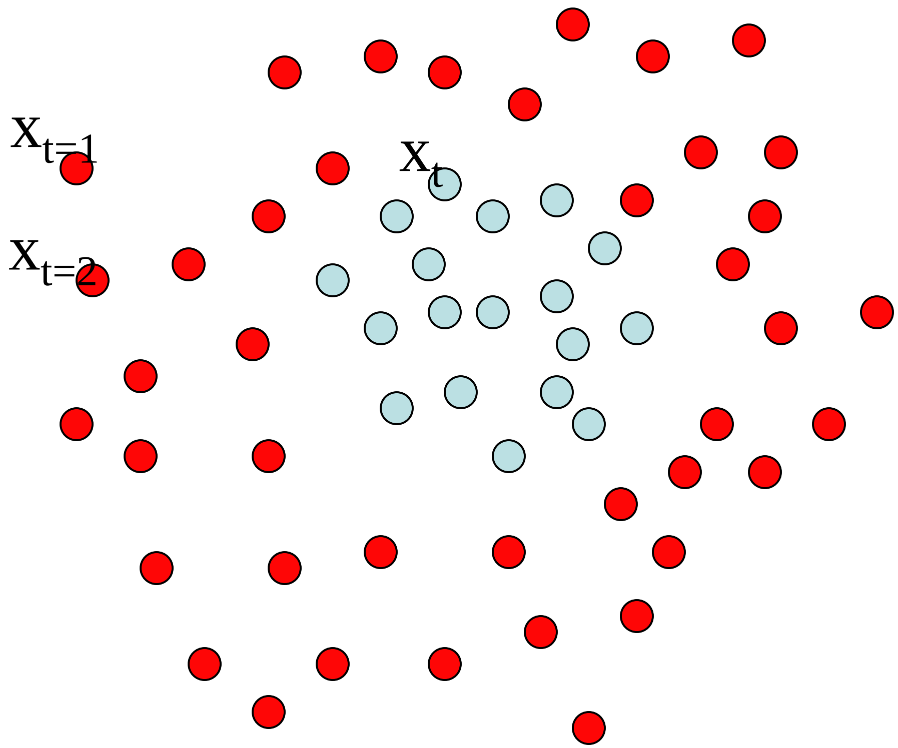
Boosting

- Defines a classifier using an additive model:

$$\underbrace{F}_{\text{Strong classifier}}(\underbrace{x}_{\text{Features vector}}) = \underbrace{\alpha_1}_{\text{Weight}}\underbrace{f_1}_{\text{Weak classifier}}(x) + \alpha_2 f_2(x) + \alpha_3 f_3(x) + \ldots$$

Boosting

- It is a sequential procedure:



Each data point has a class label:

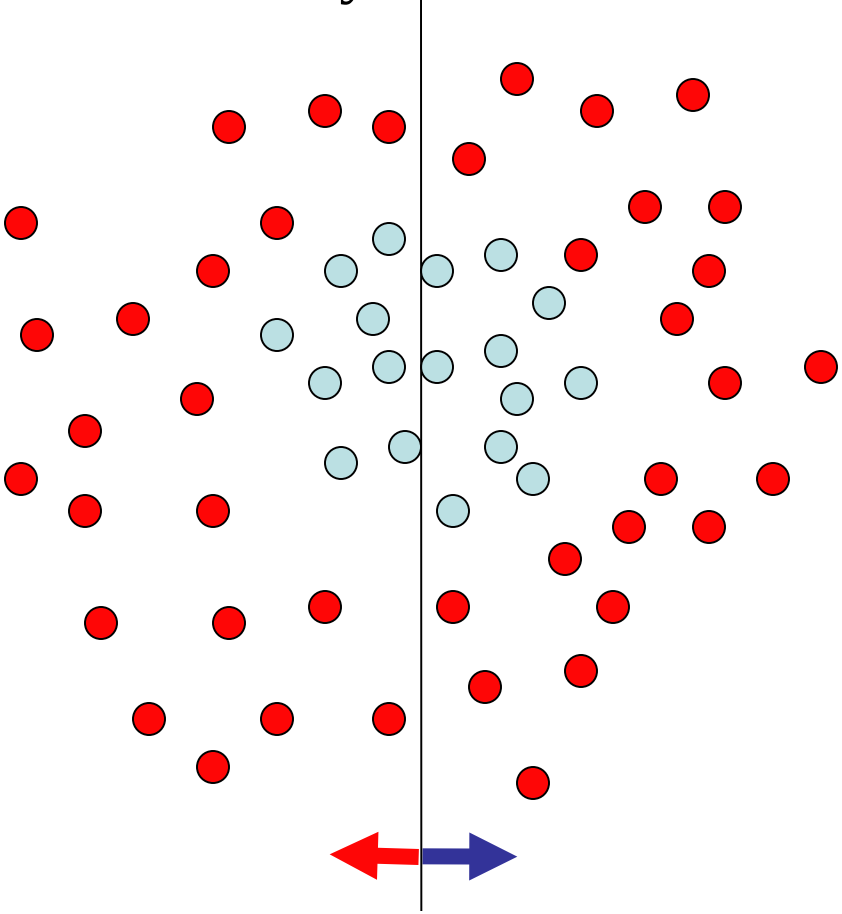
$$y_t = \begin{cases} +1 & (\;) \\ -1 & (\;) \end{cases}$$

and a weight:

$w_t = 1$

Toy example

Weak learners from the family of lines



Each data point has a class label:

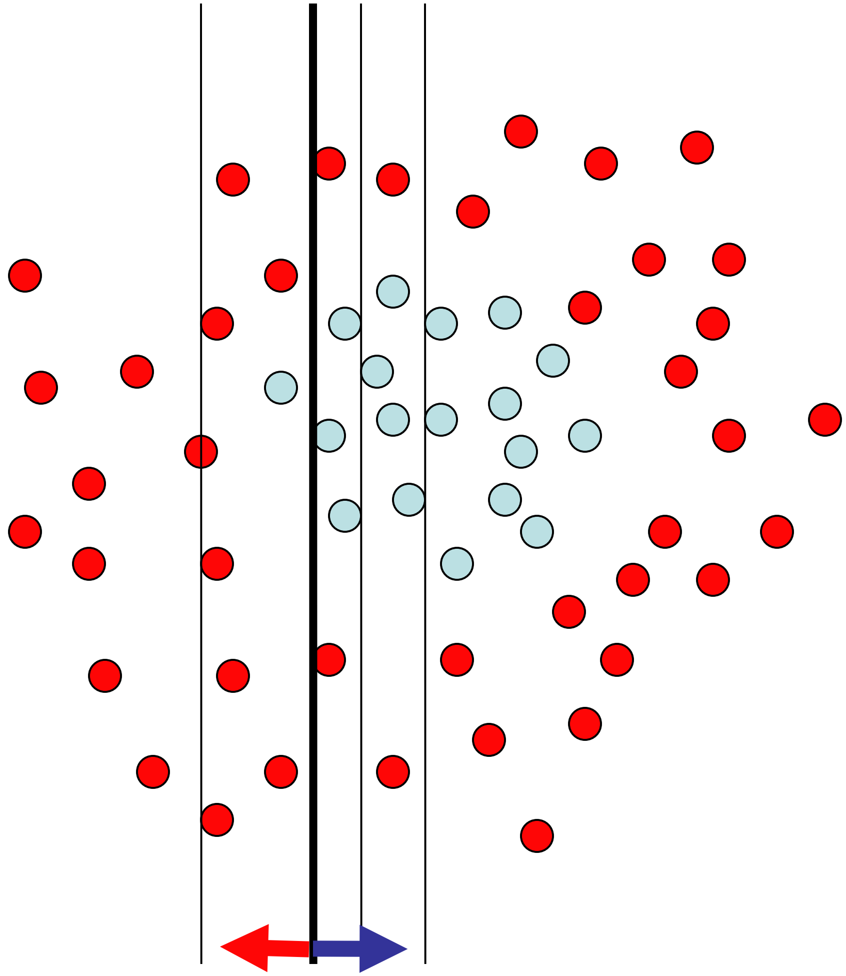
$$y_t = \begin{cases} +1 \; (\;) \\ -1 \; (\;) \end{cases}$$

and a weight:

$w_t = 1$

h => p(error) = 0.5 it is at chance

Toy example



Each data point has a class label:

$$y_t = \begin{cases} +1 \ (\text{red}) \\ -1 \ (\text{blue}) \end{cases}$$

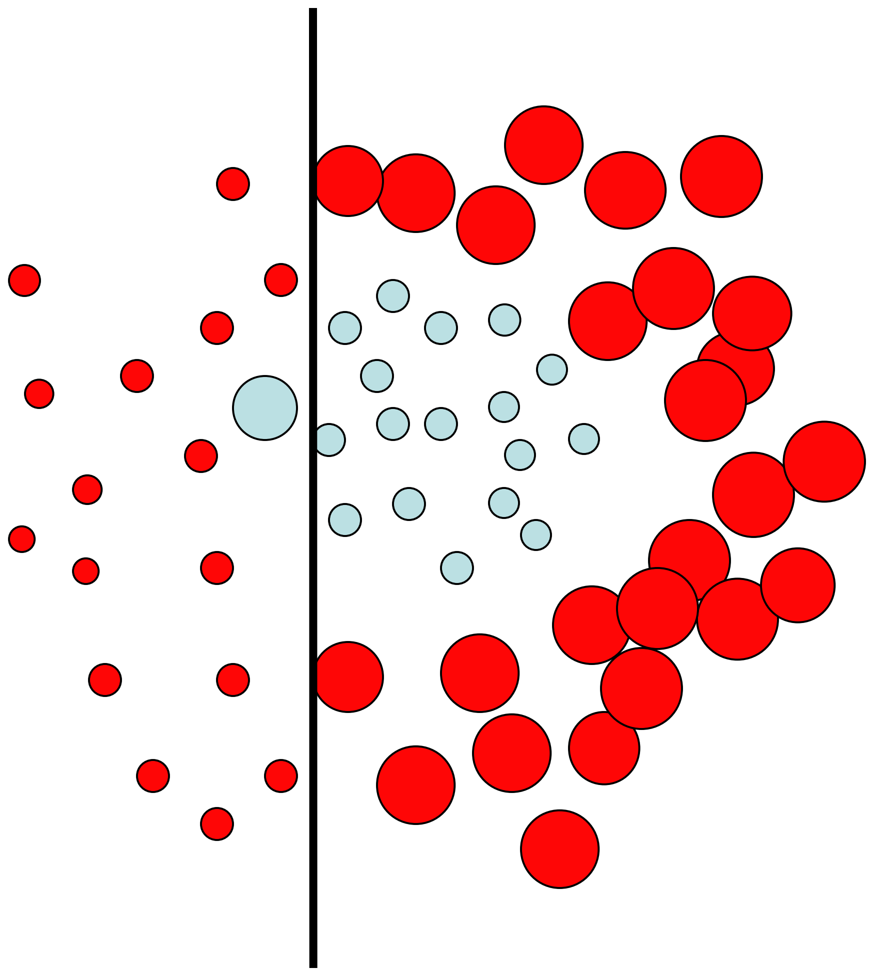
and a weight:

$w_t = 1$

This one seems to be the best

This is a '**weak classifier**': It performs slightly better than chance.

Toy example



Each data point has a class label:

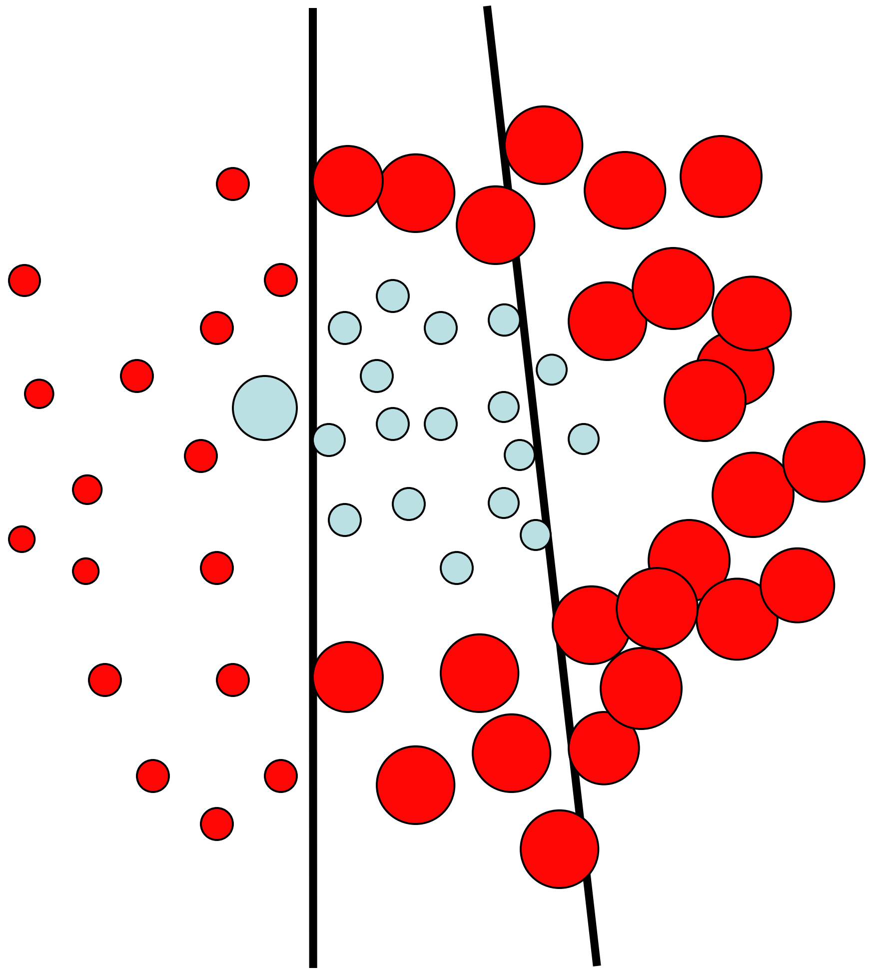
$$y_t = \begin{cases} +1 \; (\;\;) \\ -1 \; (\;\;) \end{cases}$$

We update the weights:

$$w_t \leftarrow w_t \exp\{-y_t H_t\}$$

We set a new problem for which the previous weak classifier performs at chance again

Toy example



Each data point has a class label:

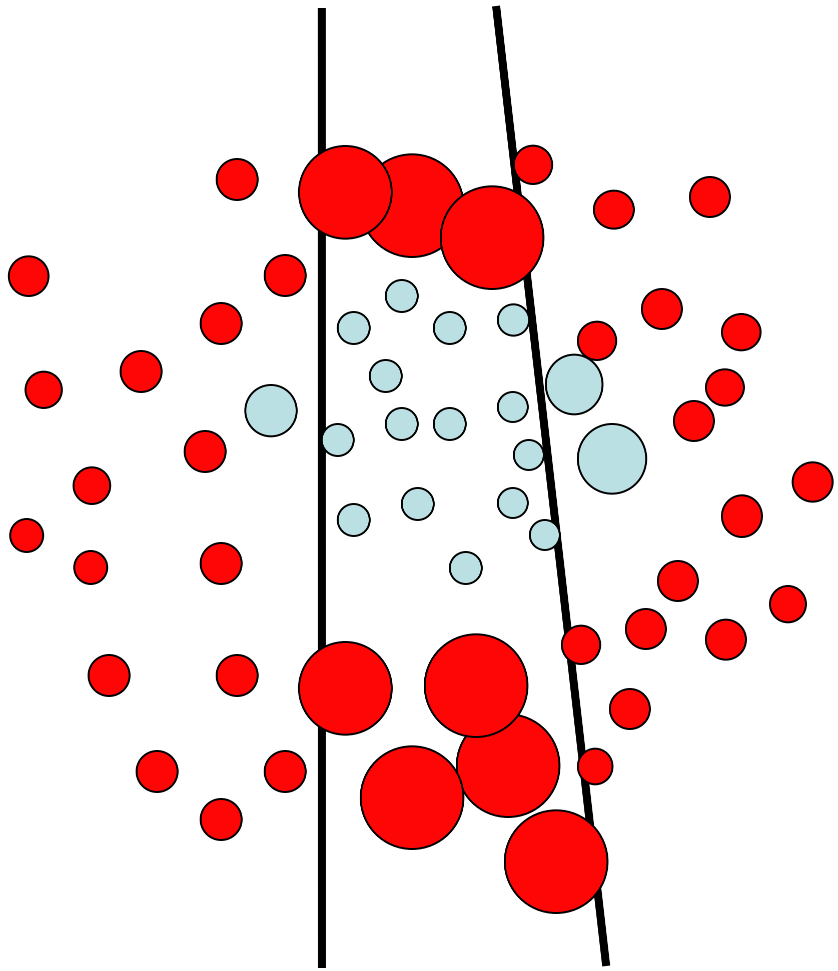
$$y_t = \begin{cases} +1 \ (\ \ ) \\ -1 \ (\ \ ) \end{cases}$$

We update the weights:

$$w_t \leftarrow w_t \exp\{-y_t H_t\}$$

We set a new problem for which the previous weak classifier performs at chance again

Toy example



Each data point has a class label:

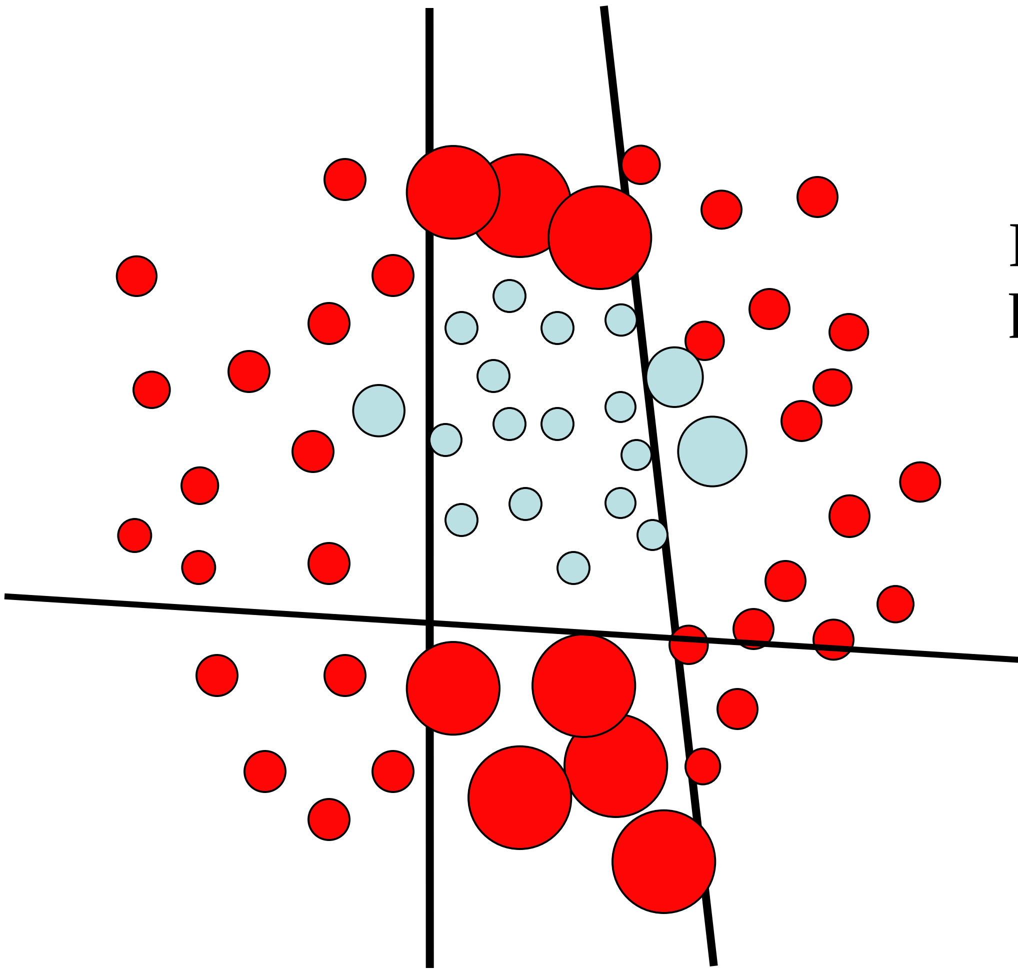
$$y_t = \begin{cases} +1 \; (\quad) \\ -1 \; (\quad) \end{cases}$$

We update the weights:

$$w_t \leftarrow w_t \exp\{-y_t H_t\}$$

We set a new problem for which the previous weak classifier performs at chance again

Toy example



Each data point has a class label:

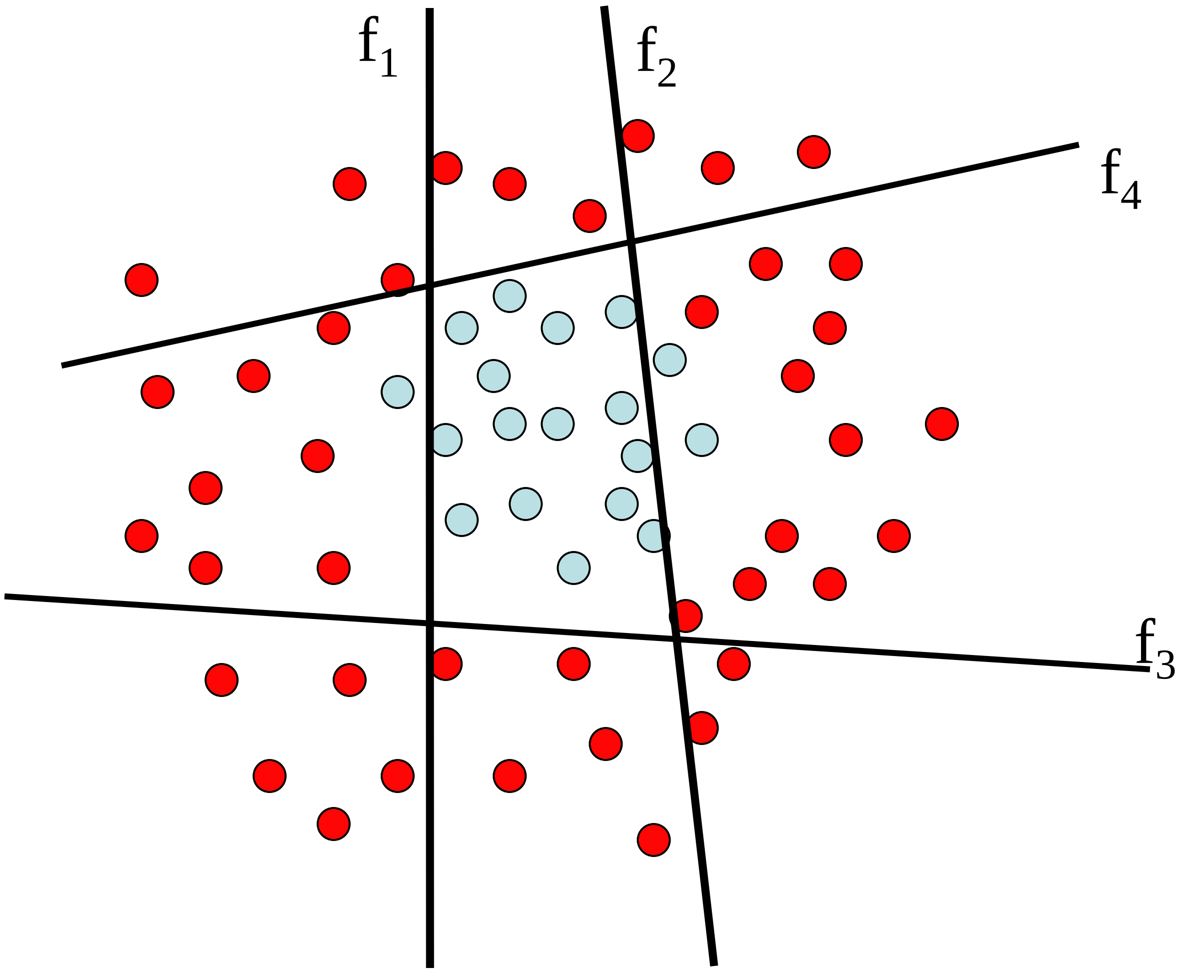
$$y_t = \begin{cases} +1 \; (\bullet) \\ -1 \; (\circ) \end{cases}$$

We update the weights:

$$w_t \leftarrow w_t \exp\{-y_t H_t\}$$

We set a new problem for which the previous weak classifier performs at chance again

Toy example



The strong (non- linear) classifier is built as the combination of all the weak (linear) classifiers.

AdaBoost Algorithm

Given: m examples $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in X, y_i \in Y = \{-1, +1\}$

Initialize $D_1(i) = 1/m$

For t = 1 to T

1. Train learner $\boldsymbol{h_t}$ with min error $\varepsilon_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i]$
2. Compute the hypothesis weight $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$
3. For each example i = 1 to m

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases}$$

Output

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

The goodness of h_t is calculated over D_t and the bad guesses.

The weight **Ada**pts. The bigger ε_t becomes the smaller α_t becomes.

Boost example if incorrectly predicted.

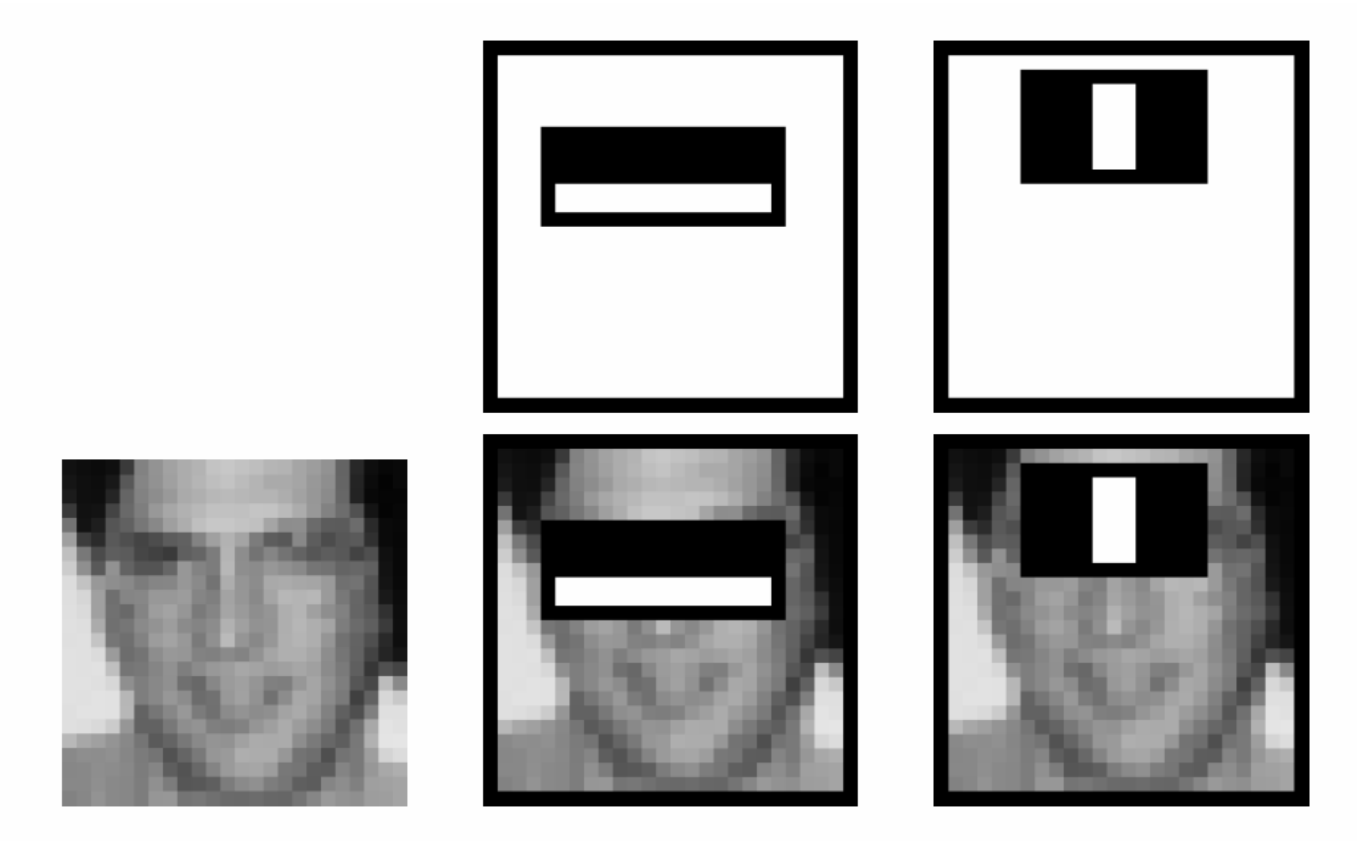
Z_t is a normalization factor.

Linear combination of models.

Details: http://www.yorku.ca/gisweb/eats4400/boost.pdf

Boosting for face detection

- First two features selected by boosting:



This feature combination can yield 100% detection rate and 50% false positive rate

Random Forest vs. Boosting

What are the pros and cons?

- Boosting:
  - +
  - -
- Random Forest:
  - +
  - -

Who wins?