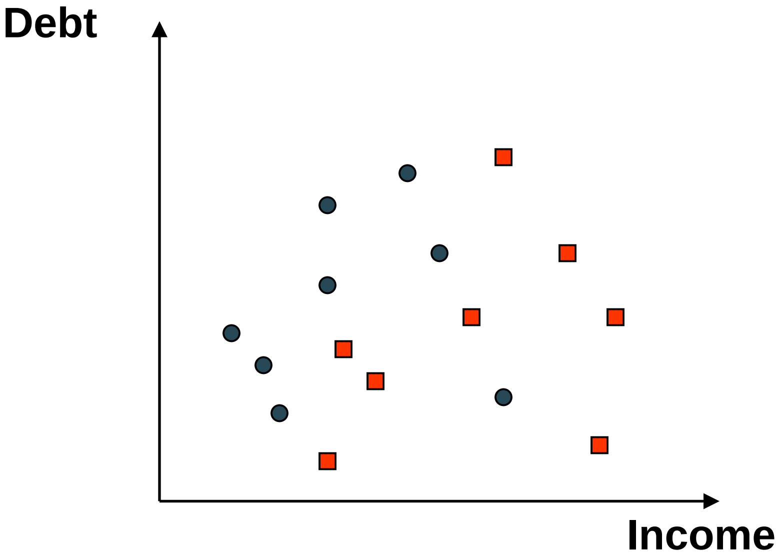
Ensembles

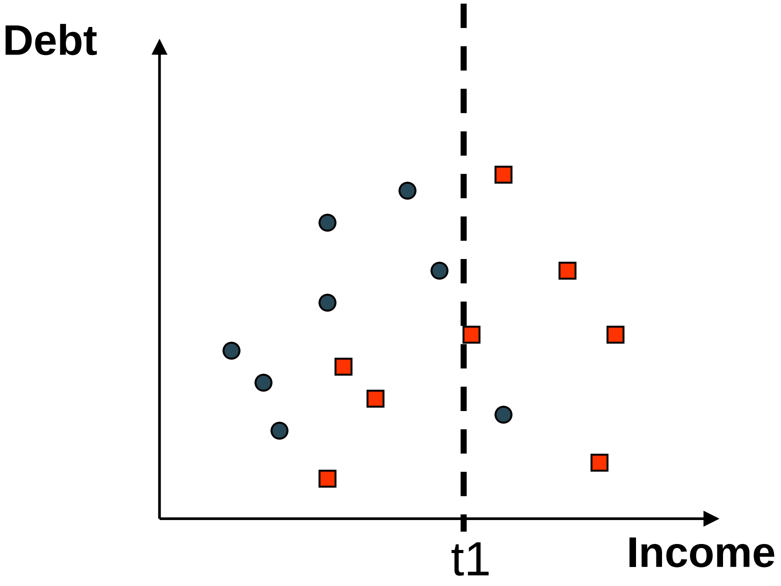
Coneixement, Raonament i Incertesa.

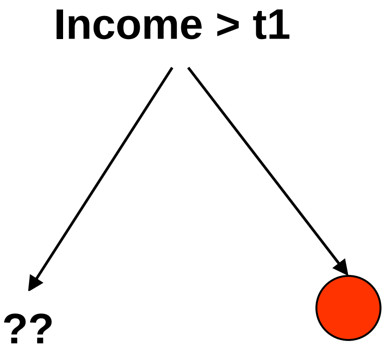
http://www.cs.utexas.edu/~ear/nsc110/ScienceAndSociety/Lectures/AI-long.ppt
http://www.cs.utexas.edu/users/ear/nsc110/Mirrors/DSMirrorsArtificialIntelligence.ppt
http://decisiontrees.net/decision-trees-tutorial/
Credit: Fernando Vilariño

Decision Tree Example

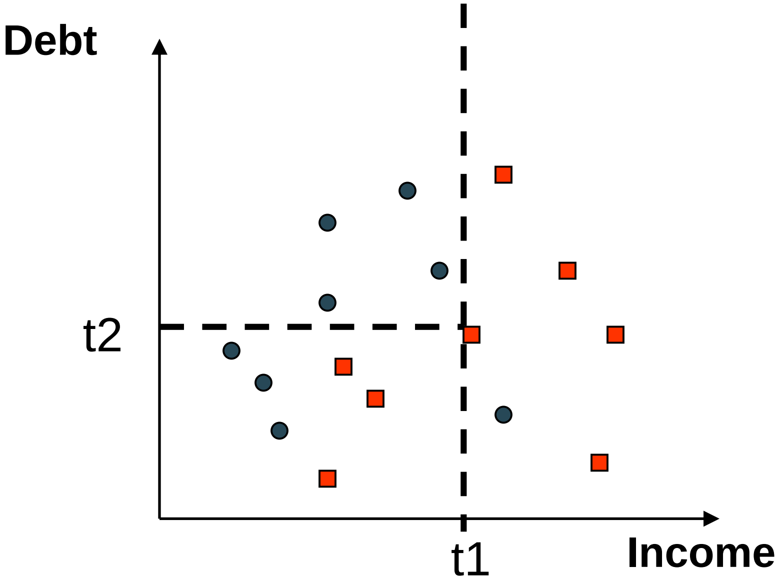


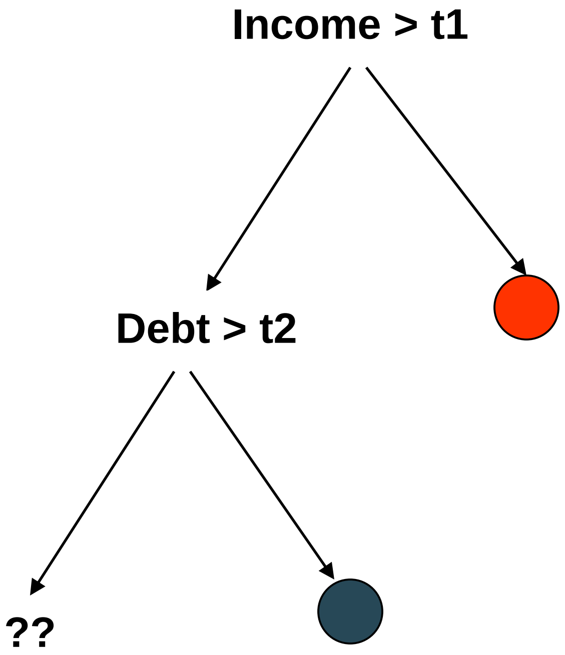
Decision Tree Example



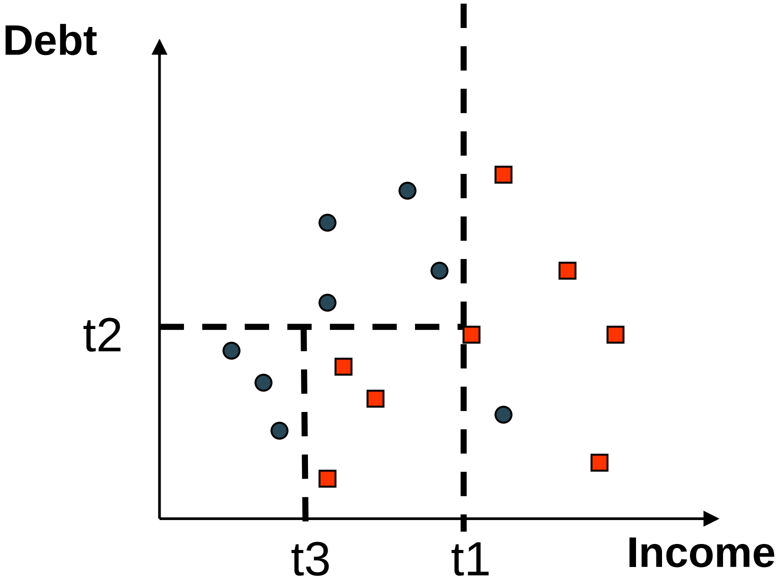


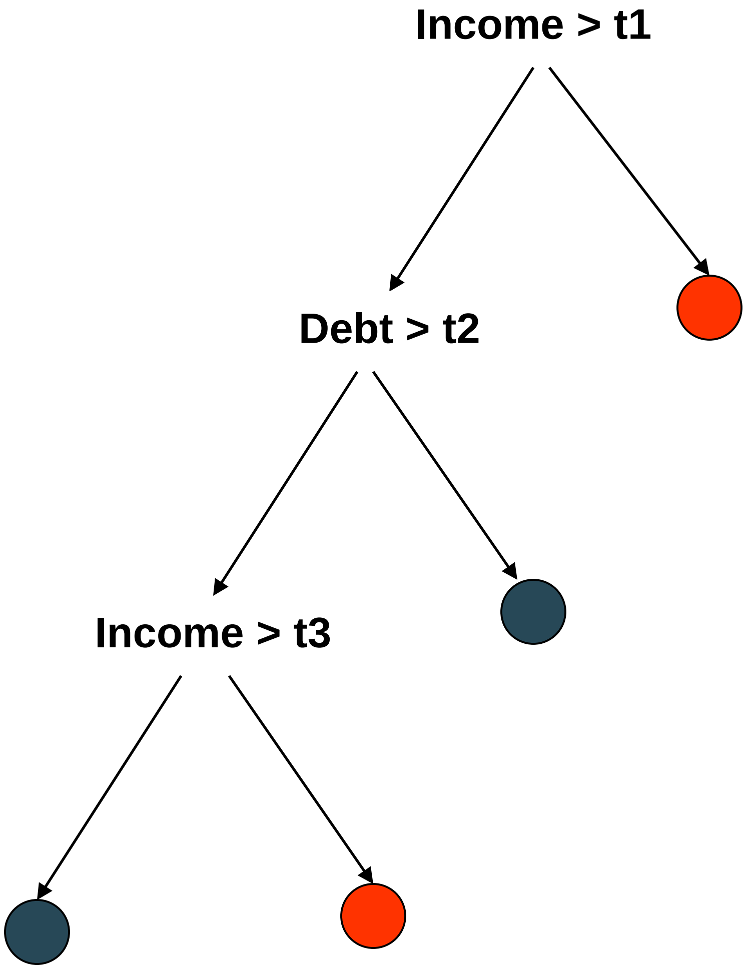
Decision Tree Example





Decision Tree Example



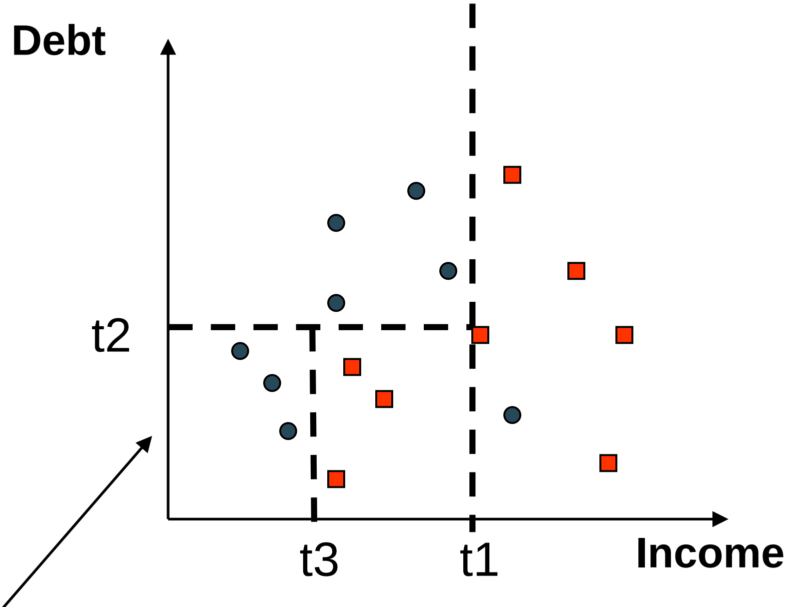


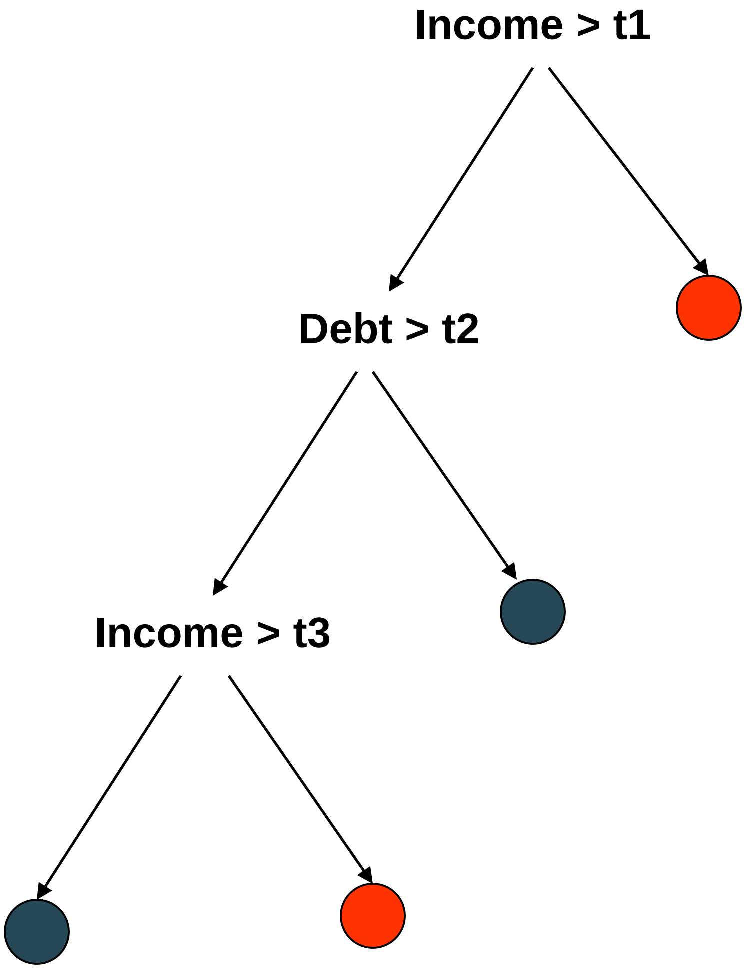
Income > t1

Debt > t2

Income > t3

Decision Tree Example



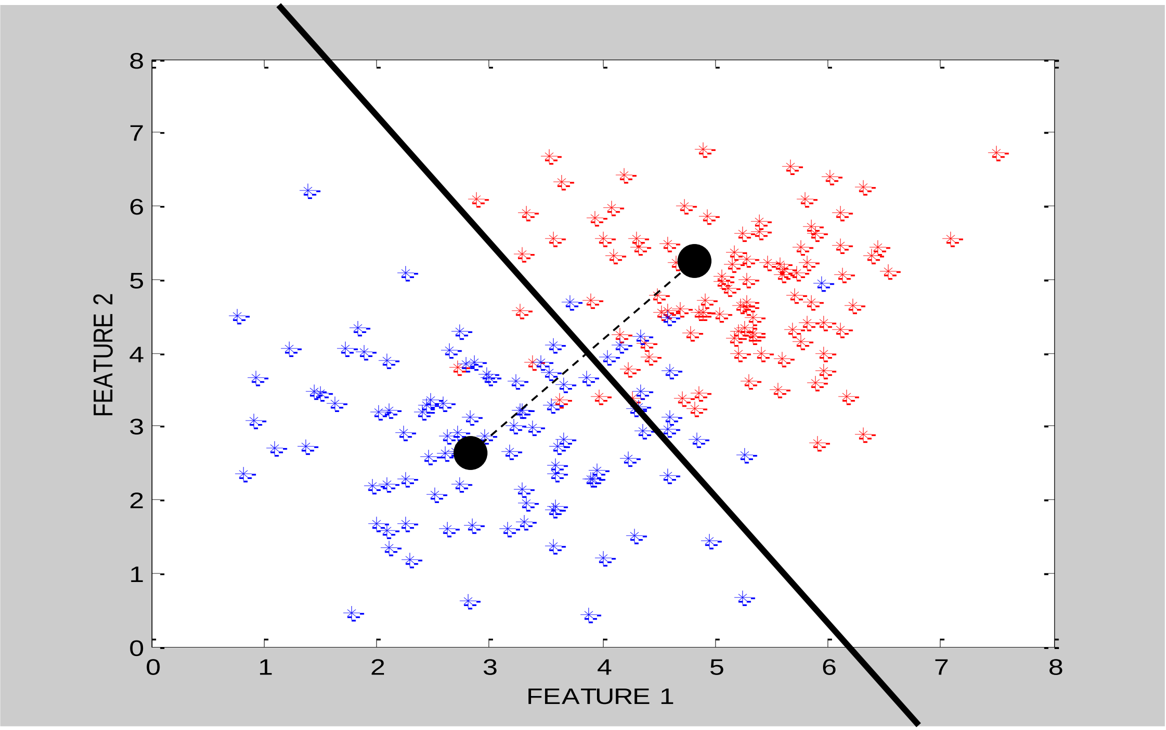

Income > t1

Debt > t2

Income > t3

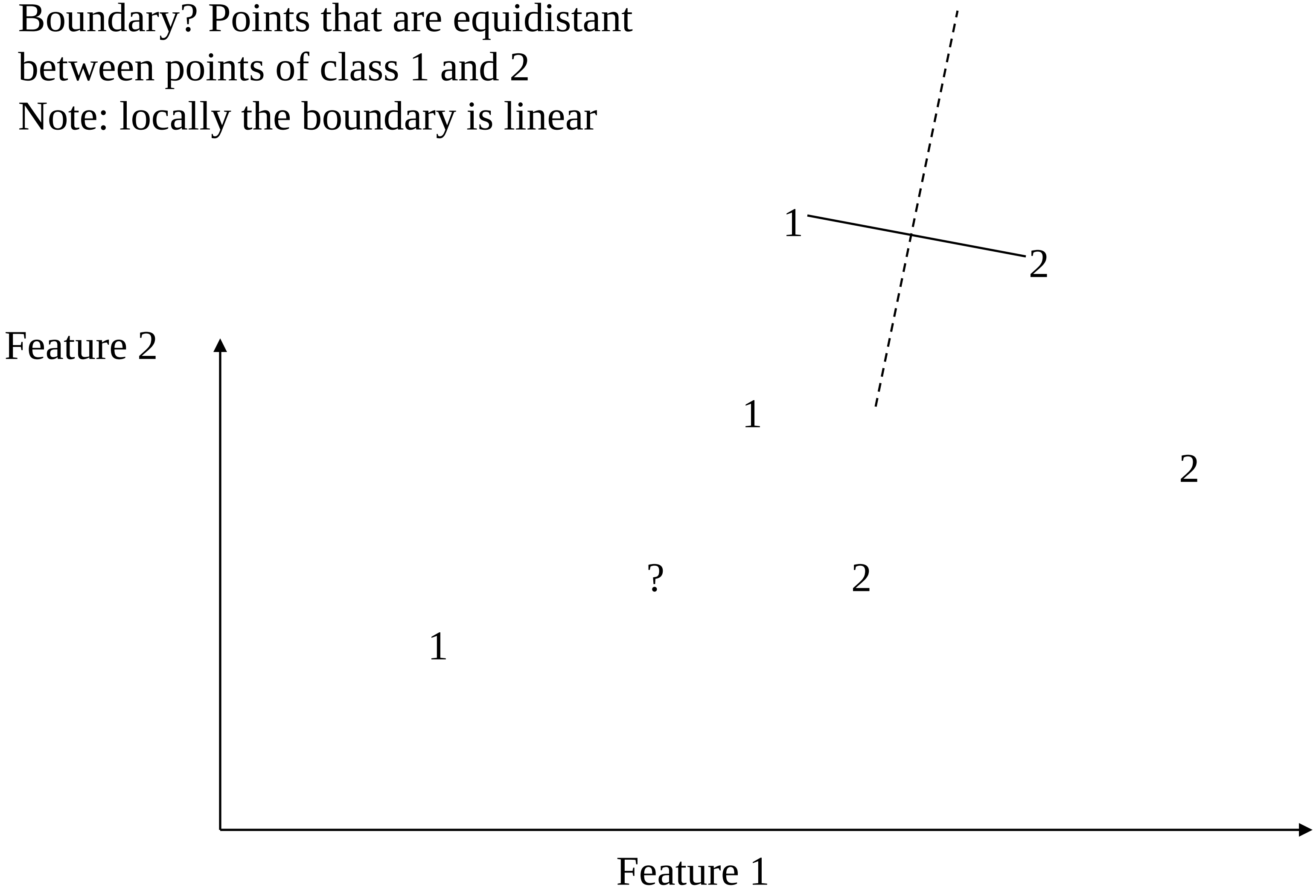
Note: tree boundaries are linear and axis-parallel

Minimum Distance Classifier

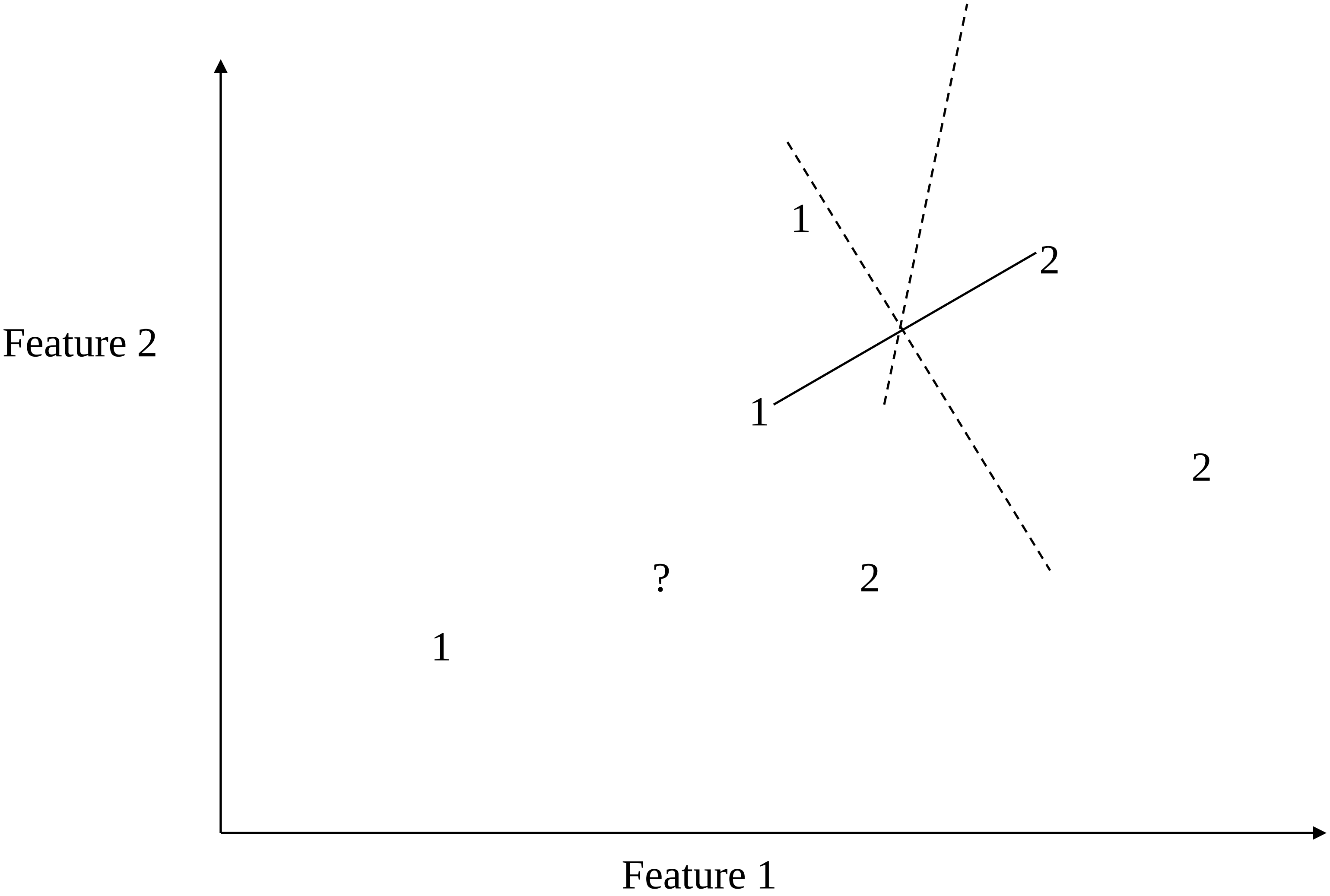


Local Decision Boundaries

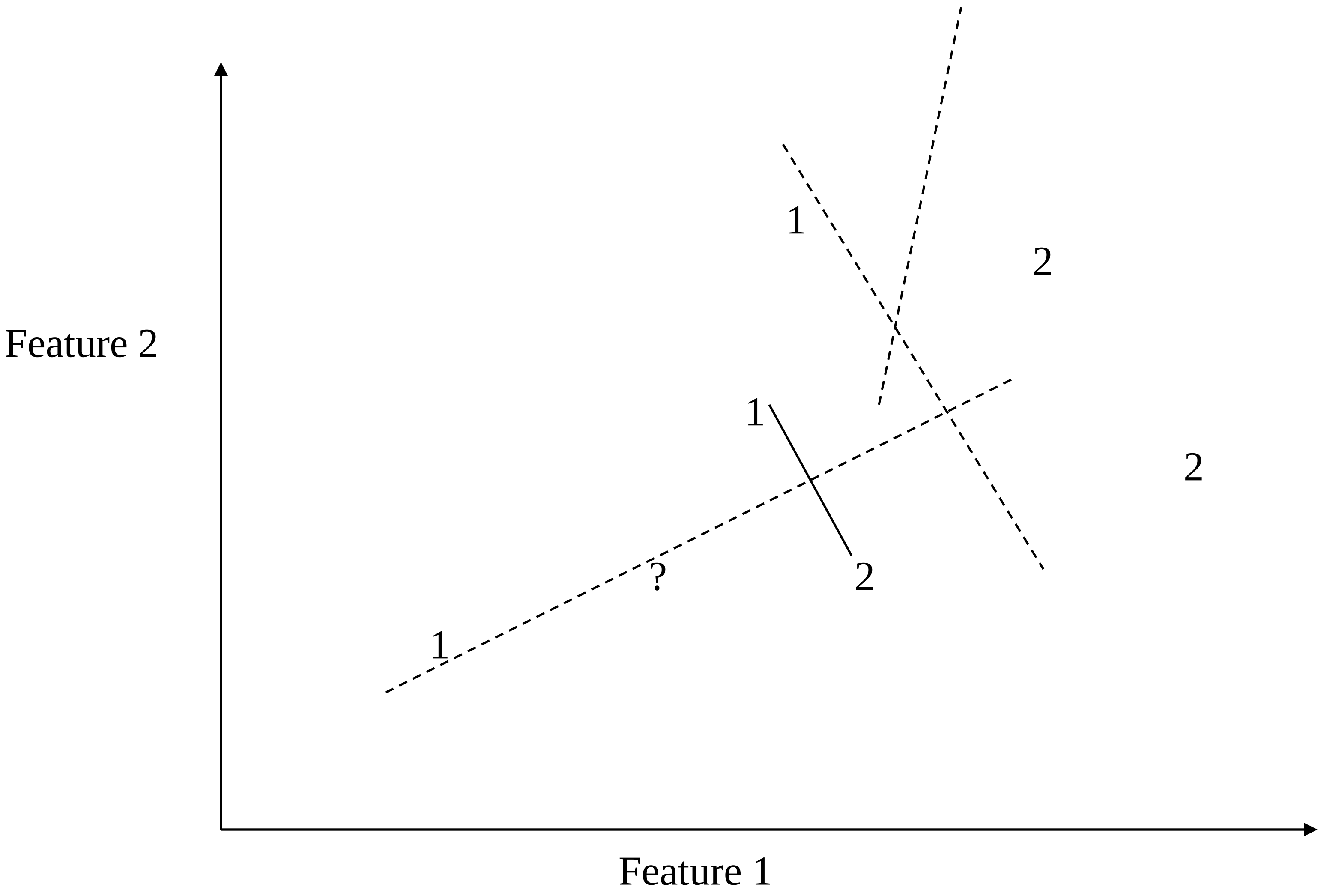
Boundary? Points that are equidistant
between points of class 1 and 2
Note: locally the boundary is linear



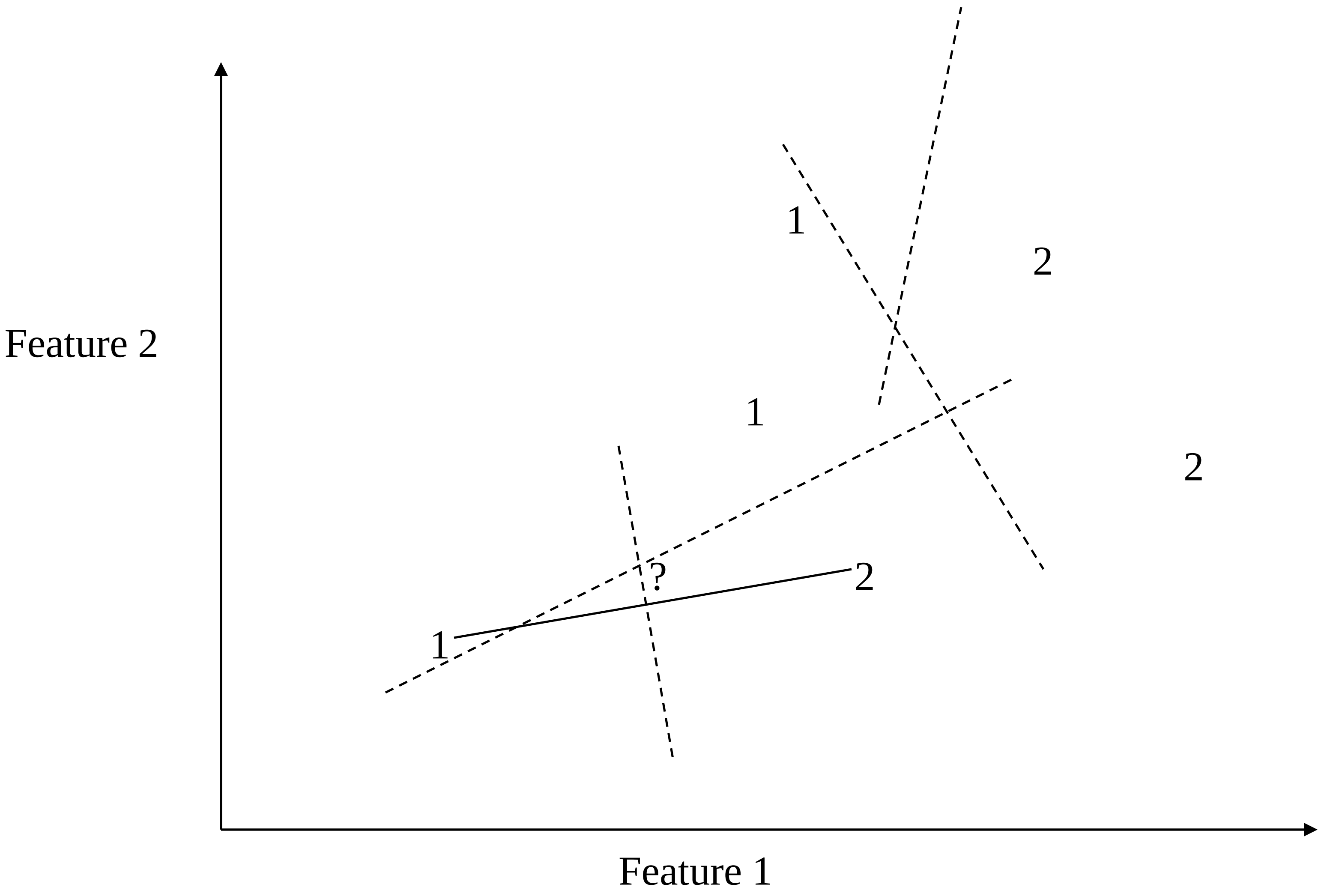
Finding the Decision Boundaries



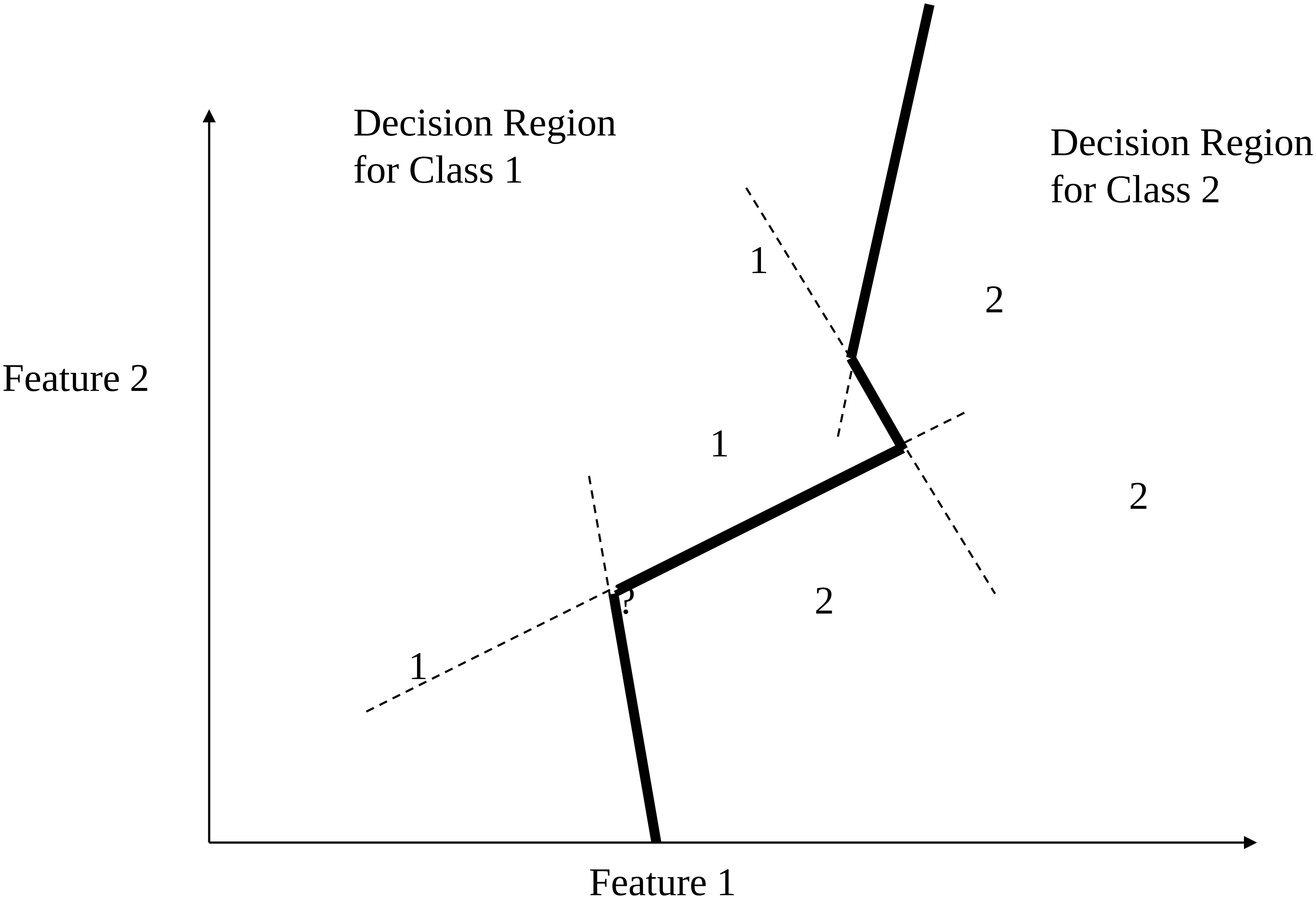
Finding the Decision Boundaries



Finding the Decision Boundaries



Overall Boundary = Piecewise Linear

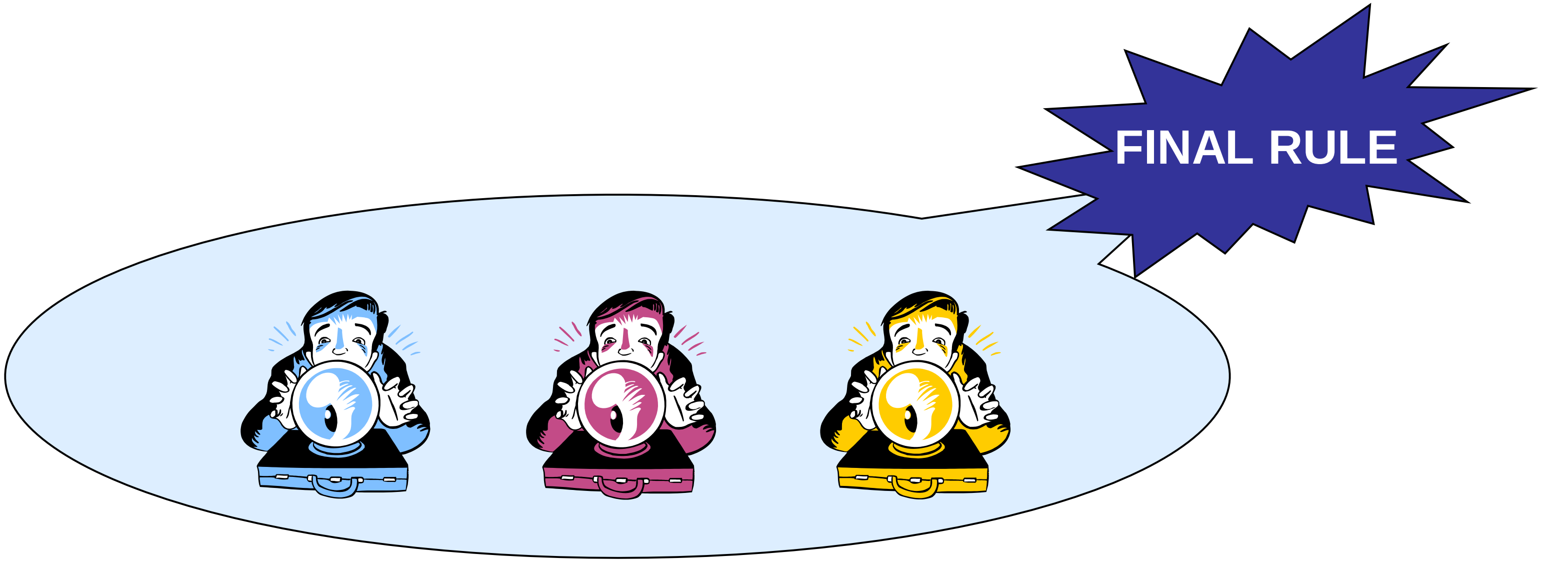


Decision Region for Class 1

Decision Region for Class 2

- Bagging and Boosting

➔Aggregating Classifiers



Breiman (1996) found gains in accuracy by aggregating predictors built from reweighed versions of the learning set

Bagging and Boosting: Aggregating Classifiers

⇨ *3 questions:*

? How to **reweigh** ?

? How to **aggregate** ?

? Which type of **gain**

in accuracy ?

Bagging

- *Bagging* = Bootstrap Aggregating

- Reweighing of the learning sets is done by drawing at random with replacement from the learning sets

- Predictors are aggregated by plurality voting

The Bagging Algorithm

- B bootstrap samples
- From which we derive:
  - ***B* Classifiers** $\in \{-1,1\}: c^1, c^2, c^3, \ldots, c^B$
  - ***B* Estimated probabilities** $\in [0,1]: p^1, p^2, p^3, \ldots, p^B$

The aggregate classifier becomes:

$$c_{bag}(x) = sign\left(\frac{1}{B}\sum_{b=1}^{B} c^b(x)\right)$$

or

$$p_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B} p^b(x)$$

Bagging Example (Opitz, 1999)

Original	1	2	3	4	5	6	7	8
Training set 1	2	7	8	3	7	6	3	1
Training set 2	7	8	5	6	4	2	7	1
Training set 3	3	6	2	7	5	6	2	2
Training set 4	4	5	1	4	6	4	3	8

Aggregation

Sign

Classifier 1

+

Classifier 2

+

Classifier 3

+

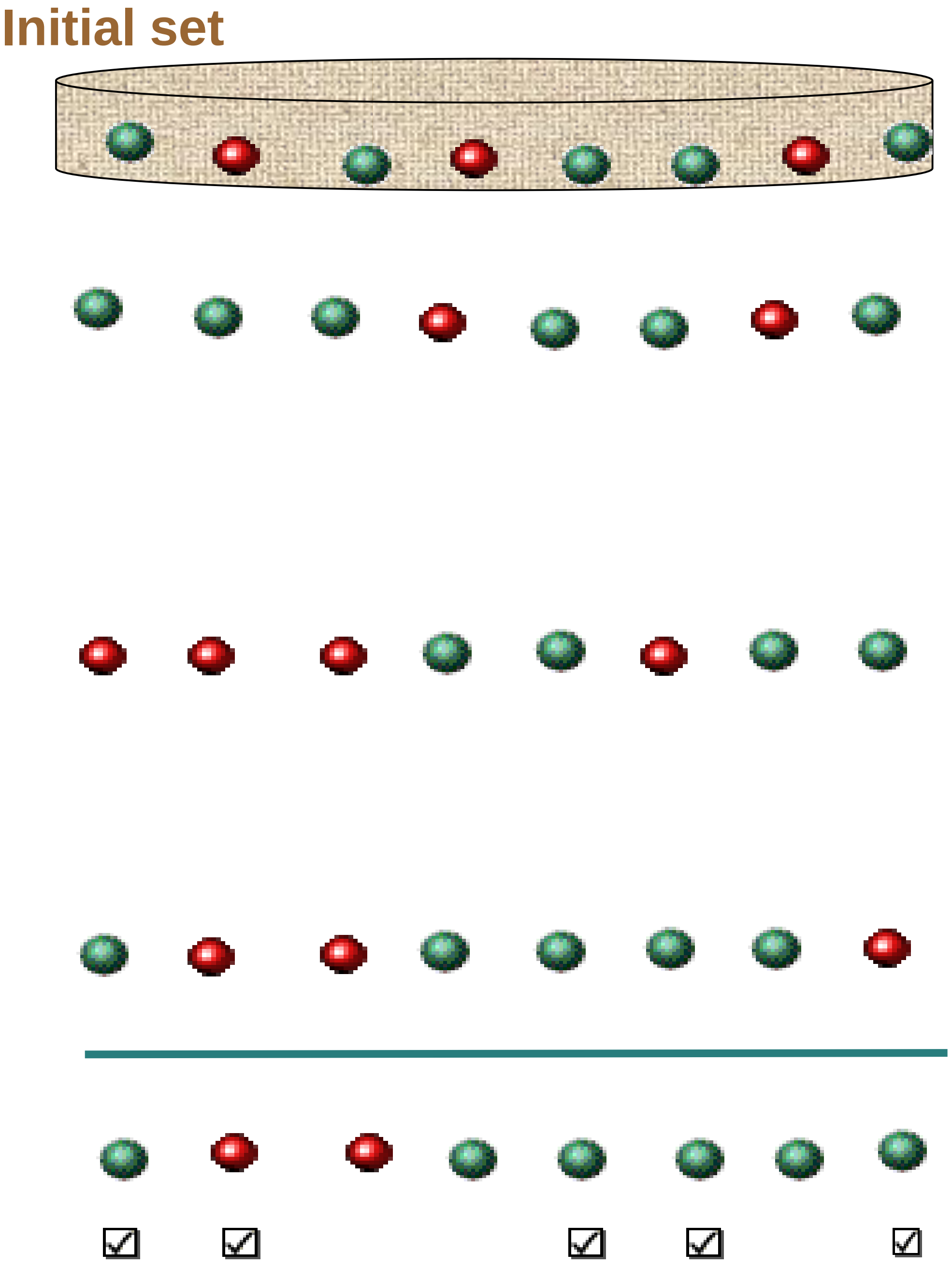
…

+

Classifier T

=

Final rule



Boosting

- Freund and Schapire (1997), Breiman (1998)
- Data *adaptively resampled*

Previously misclassified observations ➔ weights ⬆

Previously wellclassified observations ➔ weights ⬇

Predictor aggregation done by *weighted voting*

AdaBoost

$$y_i \in \{-1, +1\}$$

- Initialize weights: $w_i^1 = 1/N$
- Fit a classifier with these weights
- Give predicted probabilities to observations according to this classifier

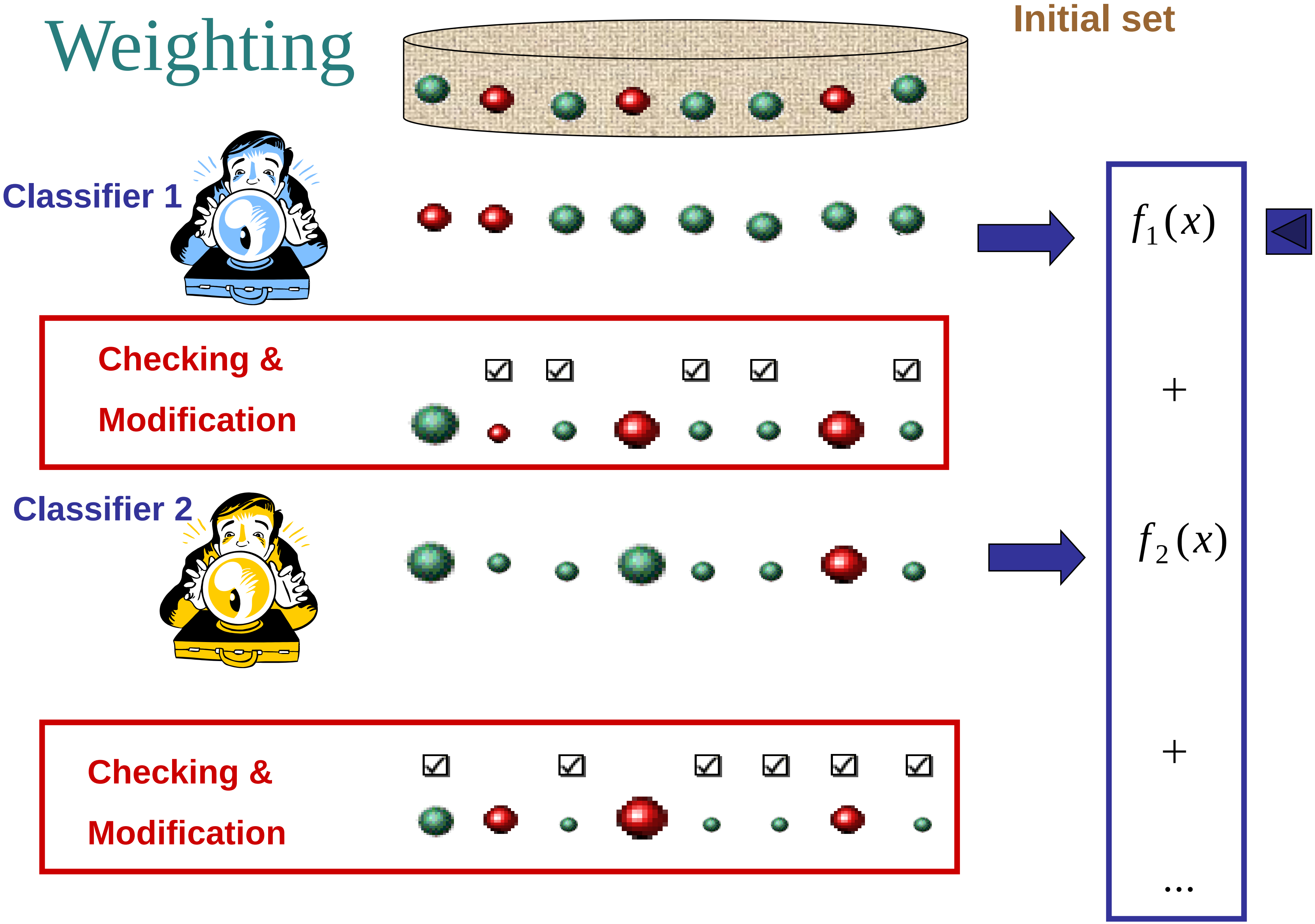
$$p_b(x) = \hat{P}_w(y = 1 | x) \in [0,1]$$

- Compute “pseudo probabilities”: $f_b(x) = \frac{1}{2}\log\left(\frac{p_b(x)}{1 - p_b(x)}\right) \in \Re$
- Get new weights: $w_i^{b+1} = w_i^b \exp\left[-y_i f_b(x_i)\right]$

& “Normalize” it (i.e., rescale so that it sums to 1)

- Combine the “pseudo probabilities”: $c_{Boost} = sign\left[\sum_{b=1}^{B} f_b(x)\right]$

Weighting



Initial set

Classifier 1

Checking &

Modification

Classifier 2

Checking &

Modification

$$f_1(x)$$

$$+$$

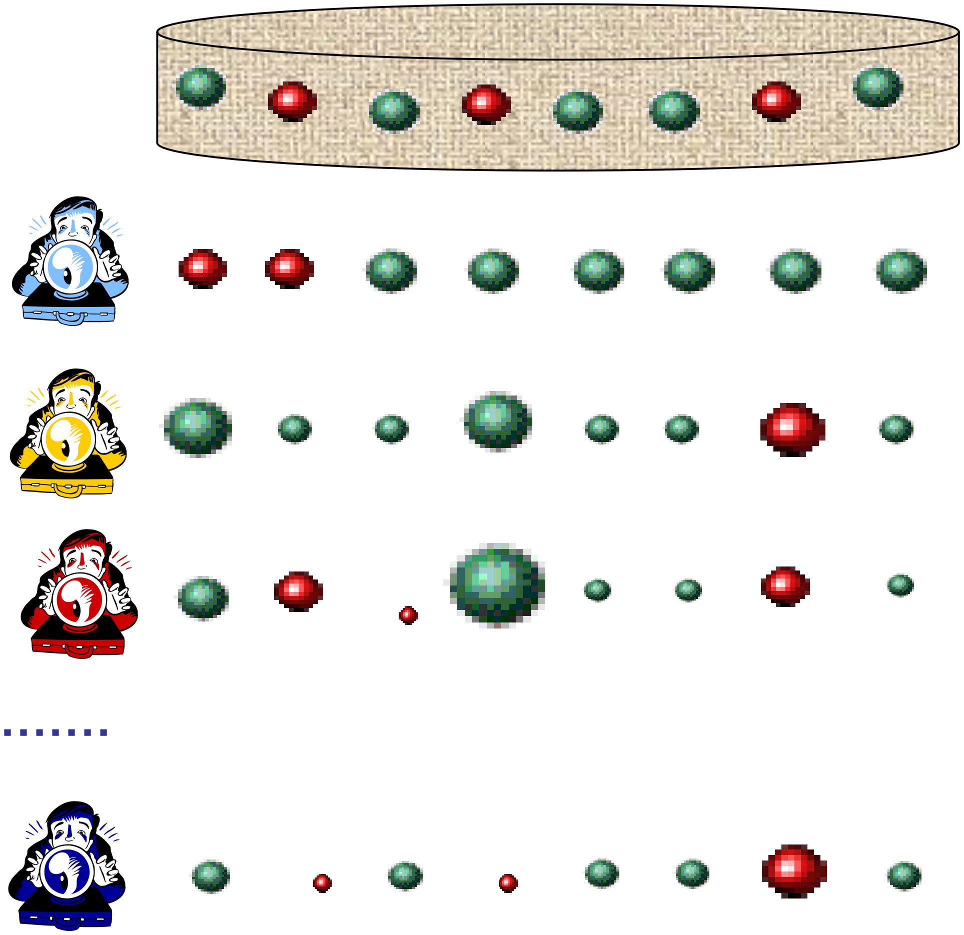
$$f_2(x)$$

$$+$$

$$\ldots$$

Aggregation

Initial set



Classifier 1

Classifier 2

Classifier 3

Classifier ....

Classifier B

$$\text{Sign}\left[f_1(x) + f_2(x) + f_3(x) + \ldots + f_B(x) \right] = \textbf{Final rule}$$

Boosting

- Definition of Boosting:

  Boosting refers to a general method of producing a very accurate prediction rule by combining rough and moderately inaccurate rules-of-thumb.

- Intuition:

  1) No learner is always the best;

  2) Construct a set of base-learners which when combined achieves higher accuracy

Boosting

3) Different learners may:

--- Be trained by different algorithms

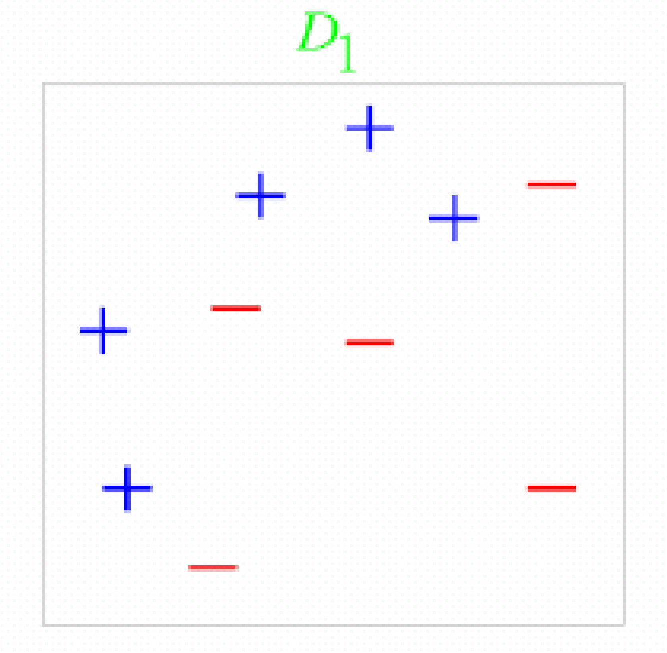
--- Use different modalities(features)

--- Focus on different subproblems

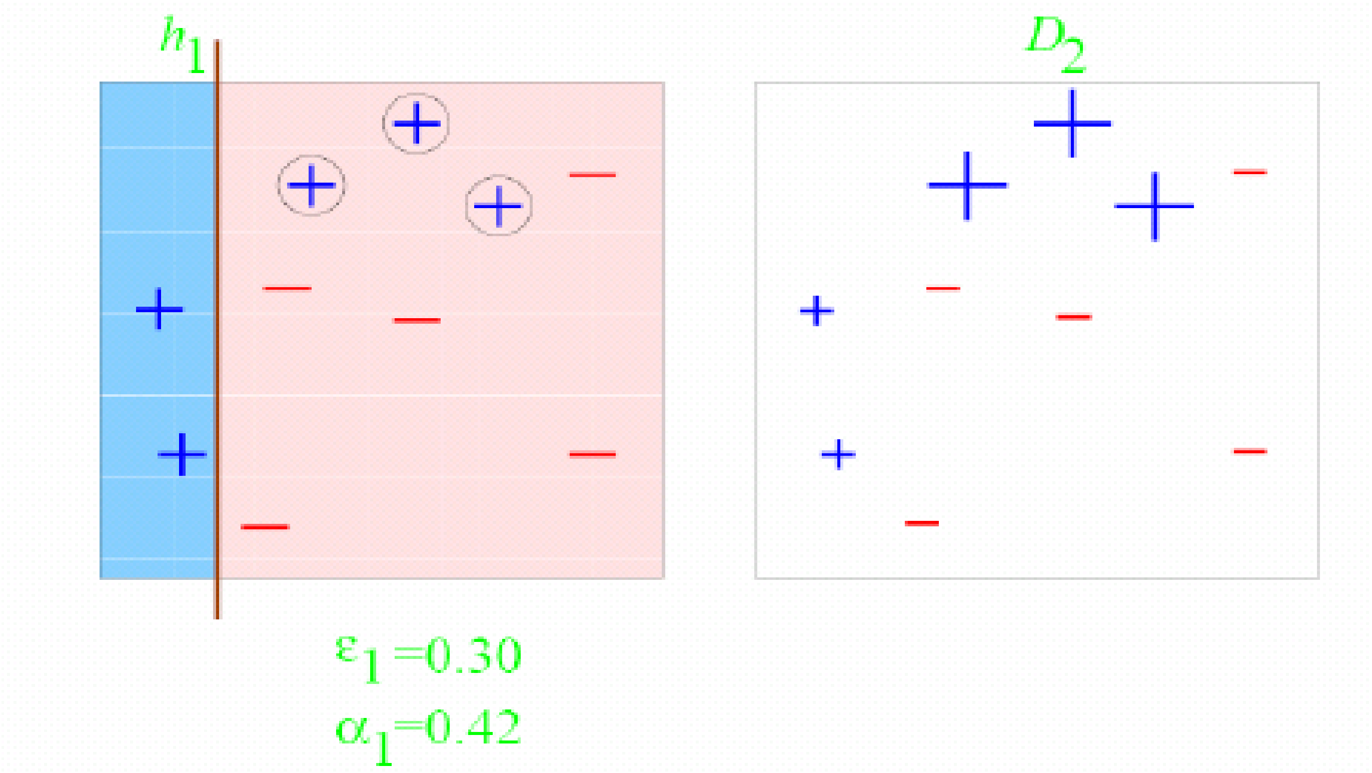
--- ……

4) A week learner is “rough and moderately inaccurate” predictor but one that can predict better than chance.

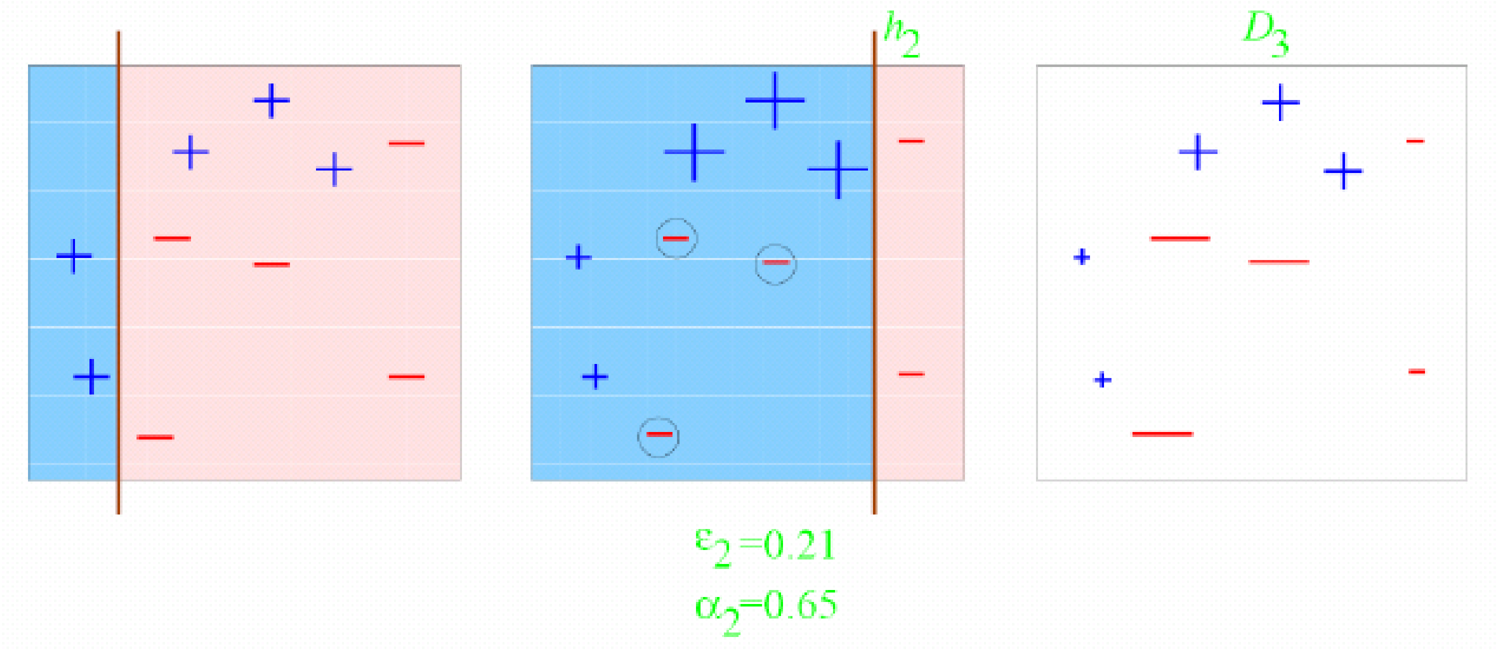
A toy example



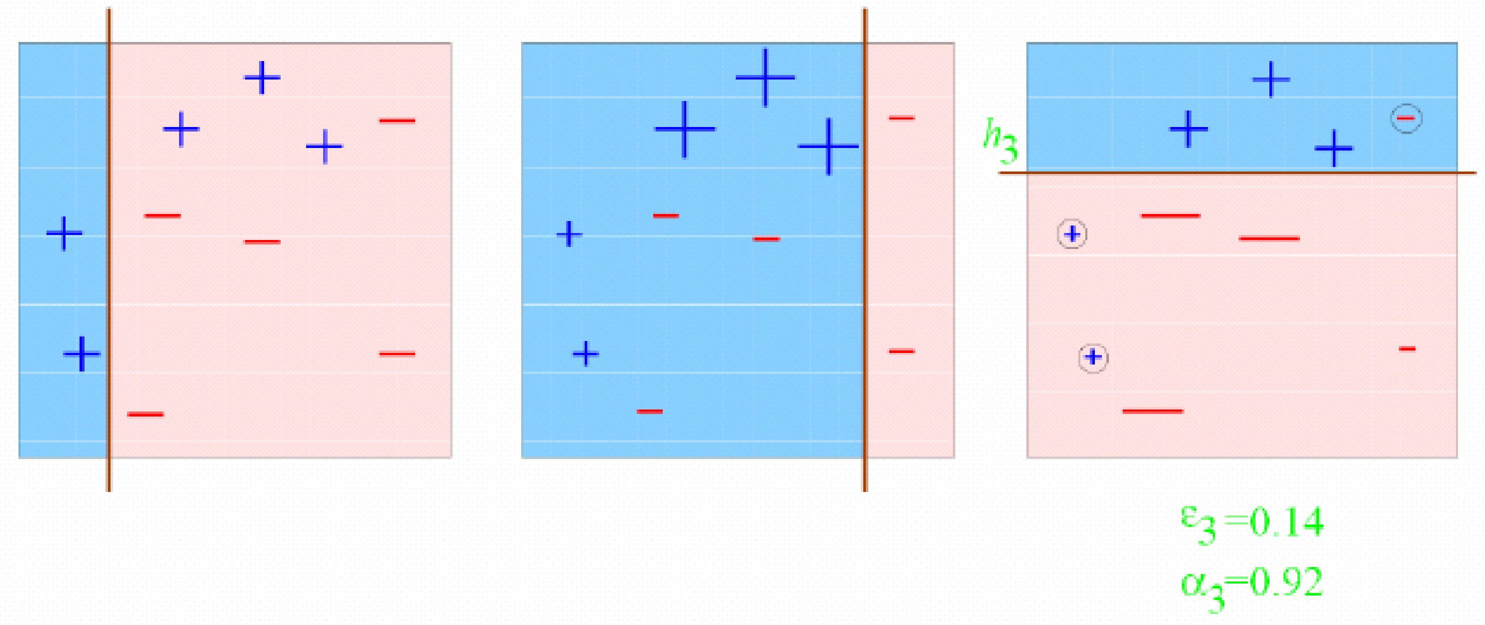
A toy example(cont'd)



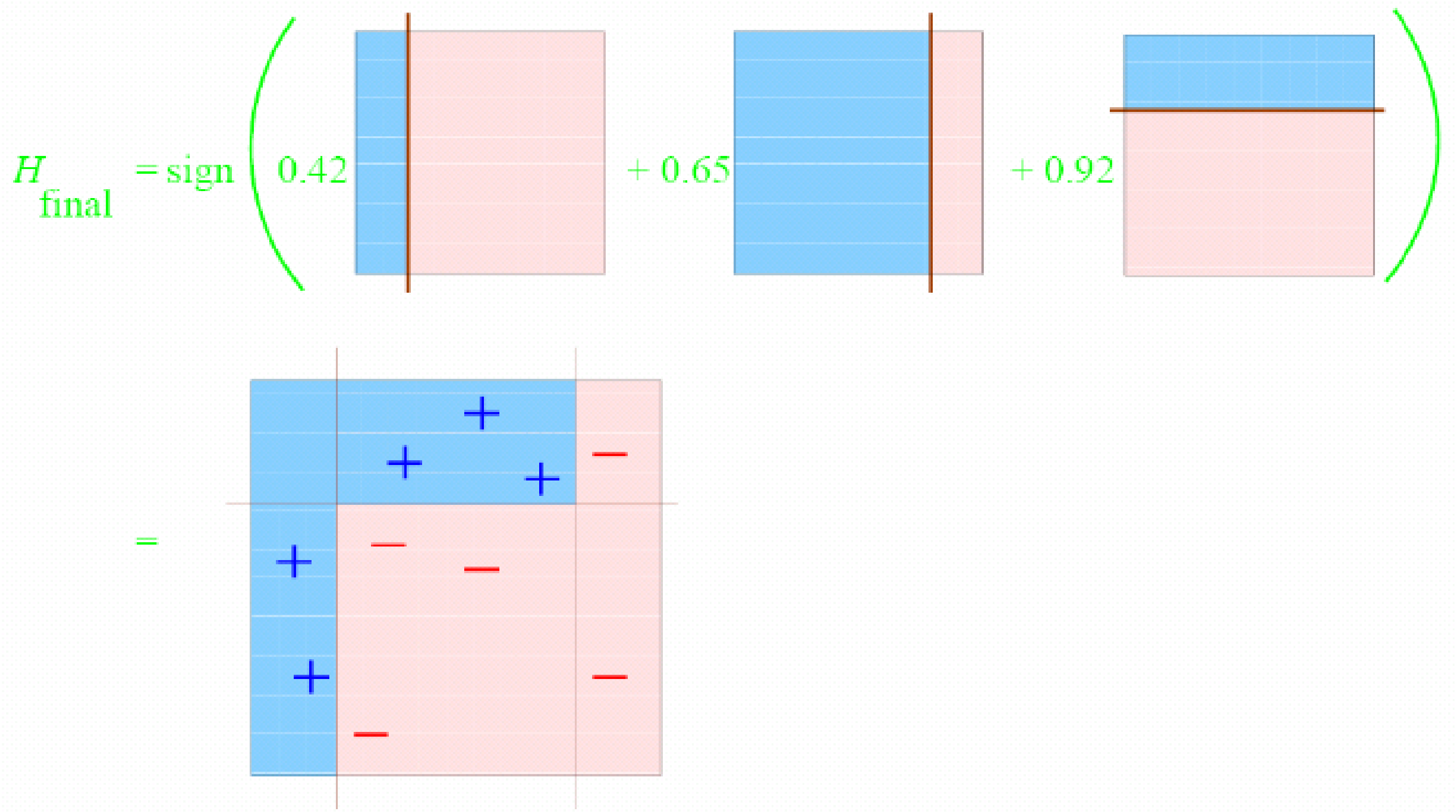
A toy example(cont’d)



A toy example(cont’d)



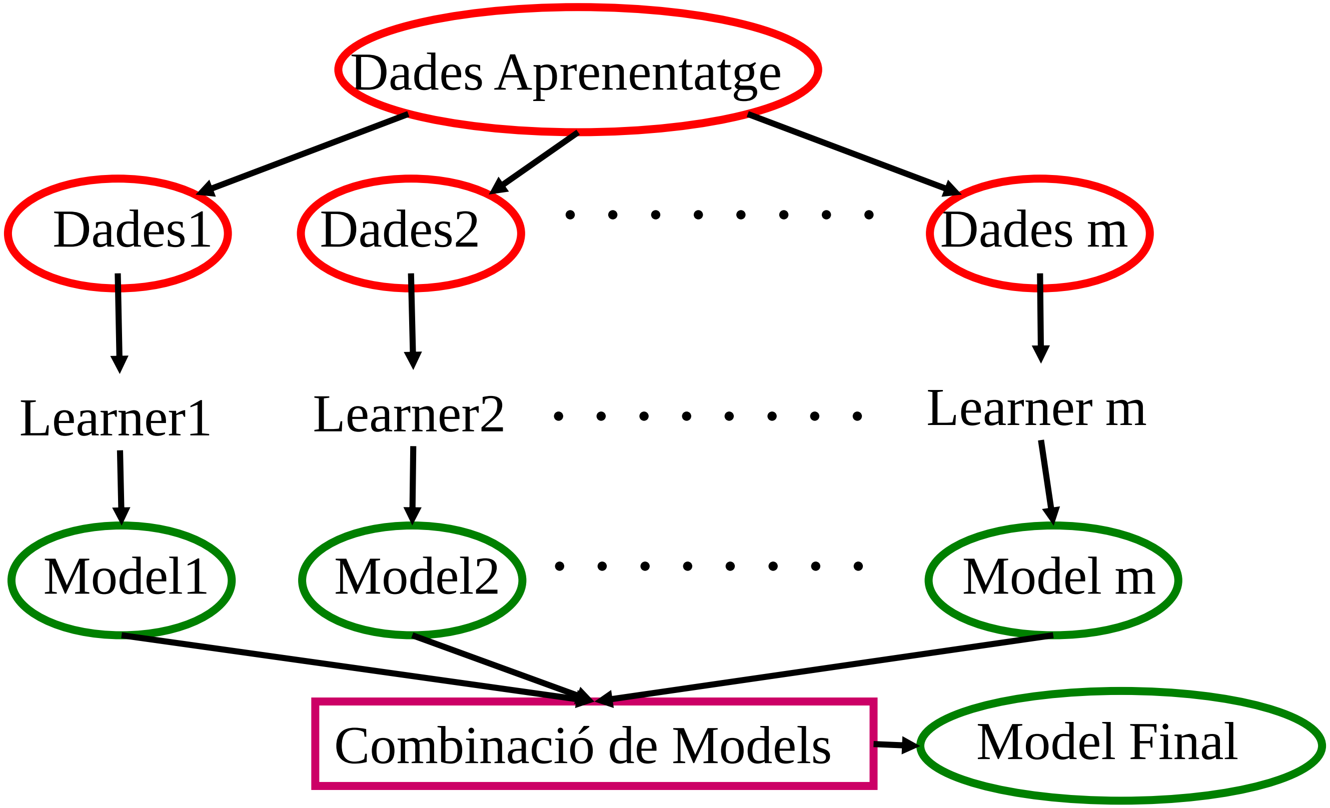
A toy example(cont’d)



Ensembles Methods

Funcionament:

- Aprendre multiples definicions alternatives d’un concepte usant **diferents dades d’aprenentatge** o **diferents algorismes d’aprenentatge**.
- **Combinar** les decisions de multiples definicions, p.ex. Usant el vot pesat.

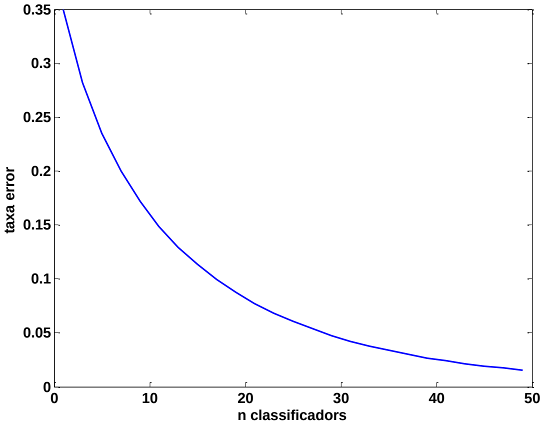


Perque funcionen?

Suposem que tenim 25 classificadors base

- Cada classificador té un taxa d’error, ε = 0.35
- Suposem que els classificadors són independents
- La probabilitat que el ‘ensemble classifier’ faci una predicció erronia (si s’equivoca en 13 de les 25 prediccions) :

$$\sum_{i=13}^{25} \binom{25}{i} \varepsilon^{i} (1-\varepsilon)^{25-i} = 0.06$$



Valor dels ‘Ensembles’

- Quan combinem múltiples decisions ***independents*** i ***diverses*** cada un de les cuals és millor que l’atzar, els errors deguts a atzar es cancel·len els uns als altres, i les decisions correctes es reforcen.

Ensembles Homogenis

Utilitzar un **únic, algorisme d'aprenentatge arbitrari** però **manipular les dades d'aprenentatge** per a fer-lo aprendre multiples models.

- Data1 ≠ Data2 ≠ … ≠ Data m
- Learner1 = Learner2 = … = Learner m
- Model 1 ≠ Model 2 ≠ … ≠ Model m

Mètodes per canviar les dades d'aprenentatge:

- Bagging: Re-mostreigar les dades d'aprenentatge
- Boosting: Re-pesar les dades d'aprenentatge
- Decorate: Afegir dades d'aprenentatge adicionals artificials