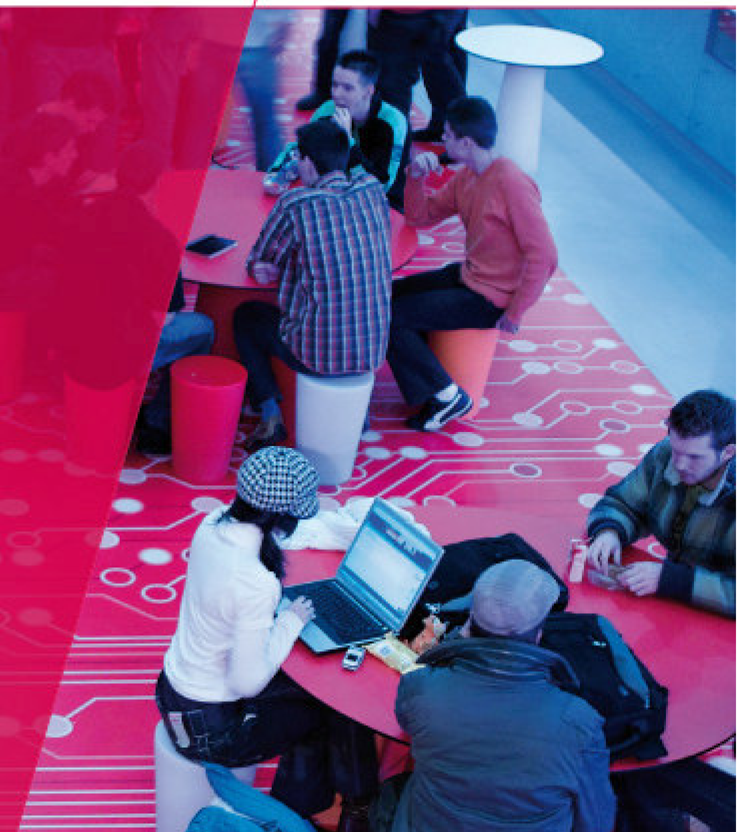
Classification
Ensemble Learning



Toon Calders

Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

Sheets are based on the those provided by Tan, Steinbach, and Kumar. *Introduction to Data Mining*

What happened before …

- Classification:
  - Learning a model on labeled data for prediction.
- Models:
  - Decision trees (Hunt’s algorithm)
  - Naïve Bayes Classifier
  - Nearest Neighbor Classifier
- Evaluation of models and classifiers

This lecture

- Combining classifiers
  - Bagging
  - Boosting
    - AdaBoost
  - Random Forest

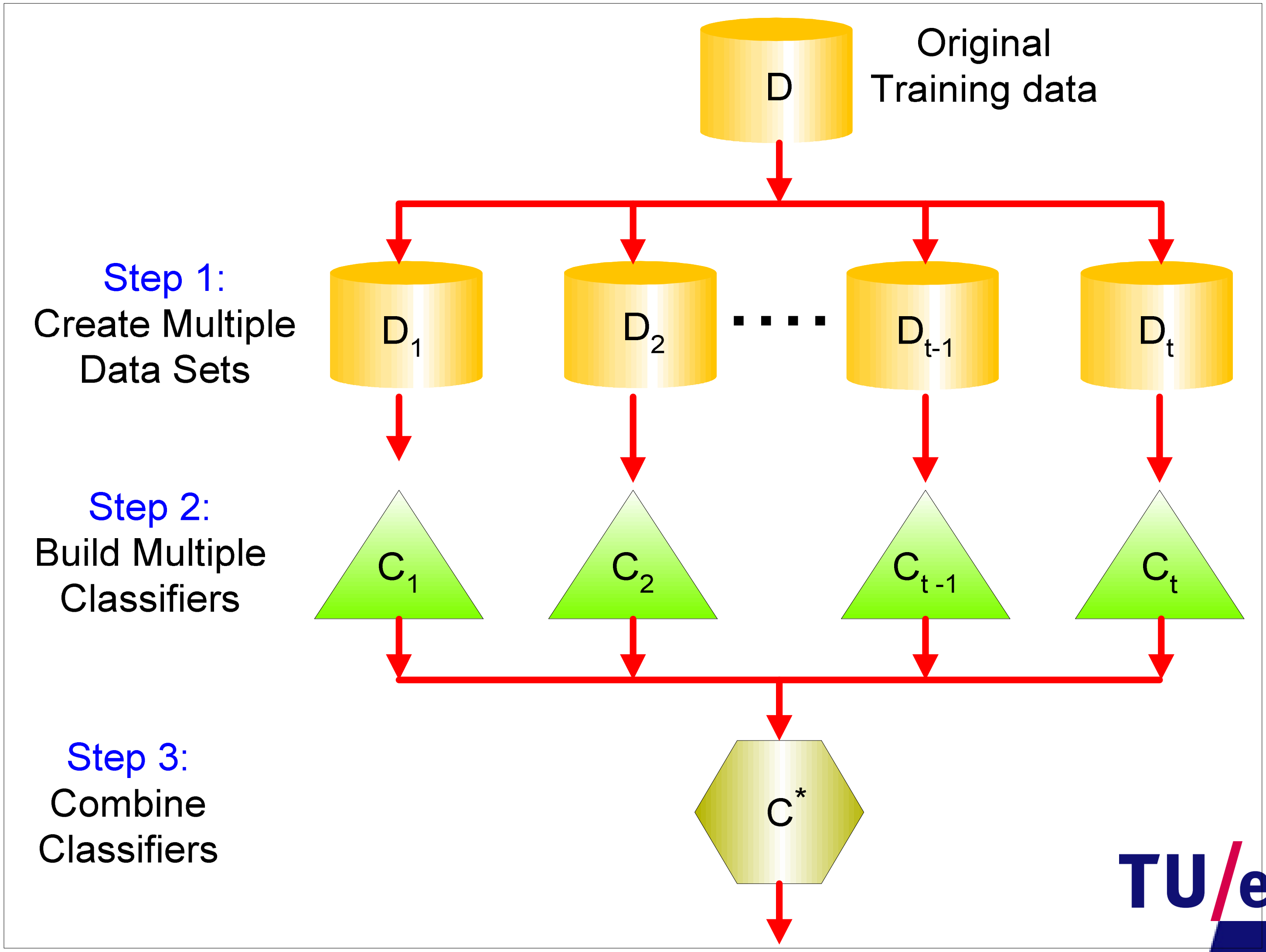
- Conclusion

- Exercises

Ensemble Methods

- Construct a set of classifiers from the training data
- Predict class label of previously unseen records by aggregating predictions made by multiple classifiers

General Idea



TU/e Technische Universiteit Eindhoven University of Technology

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:

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

This lecture

- Combining classifiers
  - Bagging
  - Boosting
    - AdaBoost
    - Random Forest

- Exercises

Bagging

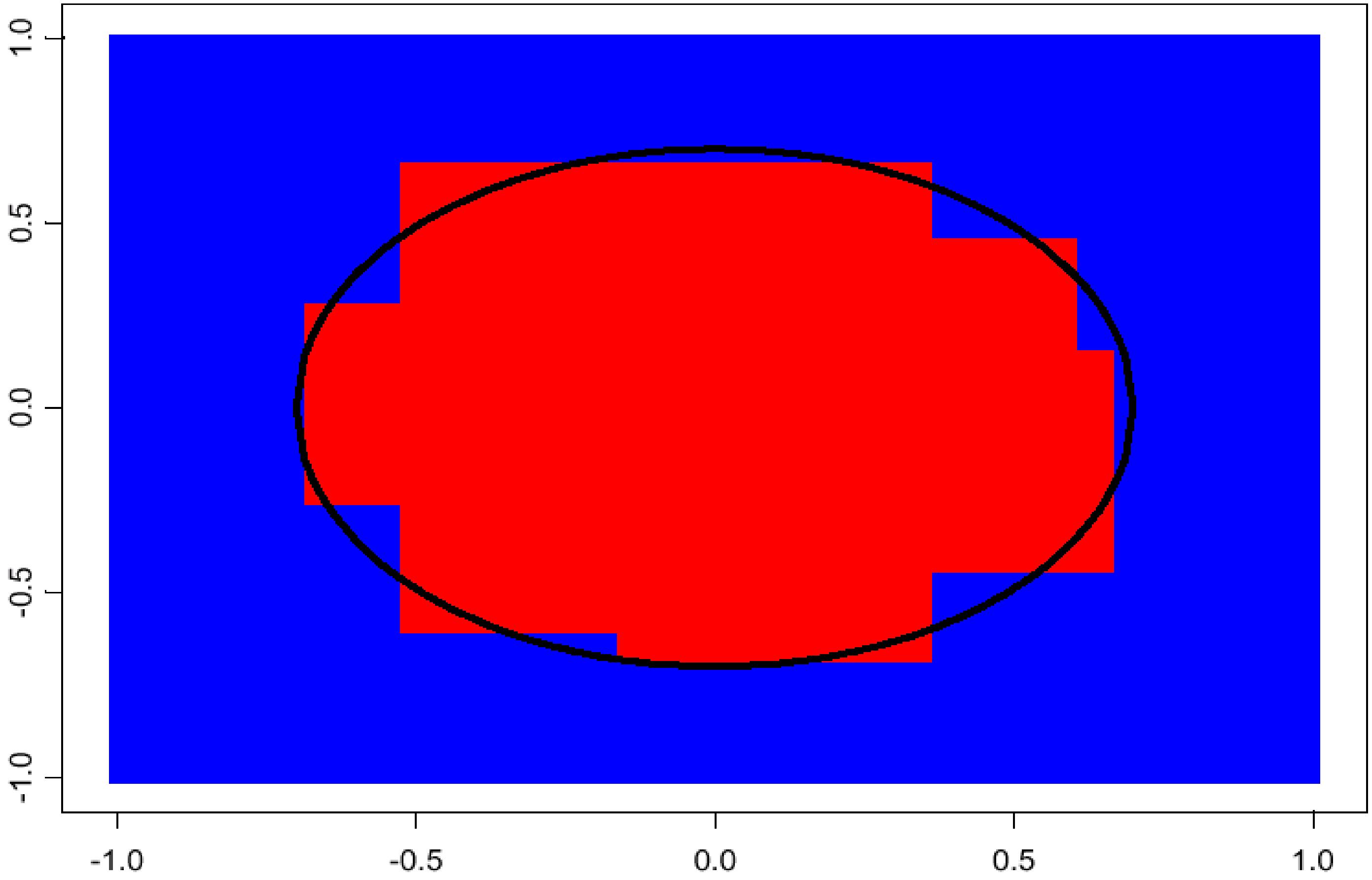
- **Sampling with replacement**

Original Data	1	2	3	4	5	6	7	8	9	10
Bagging (Round 1)	7	8	10	8	2	5	10	10	5	9
Bagging (Round 2)	1	4	9	1	2	3	2	7	3	2
Bagging (Round 3)	1	8	5	10	5	5	9	6	3	7

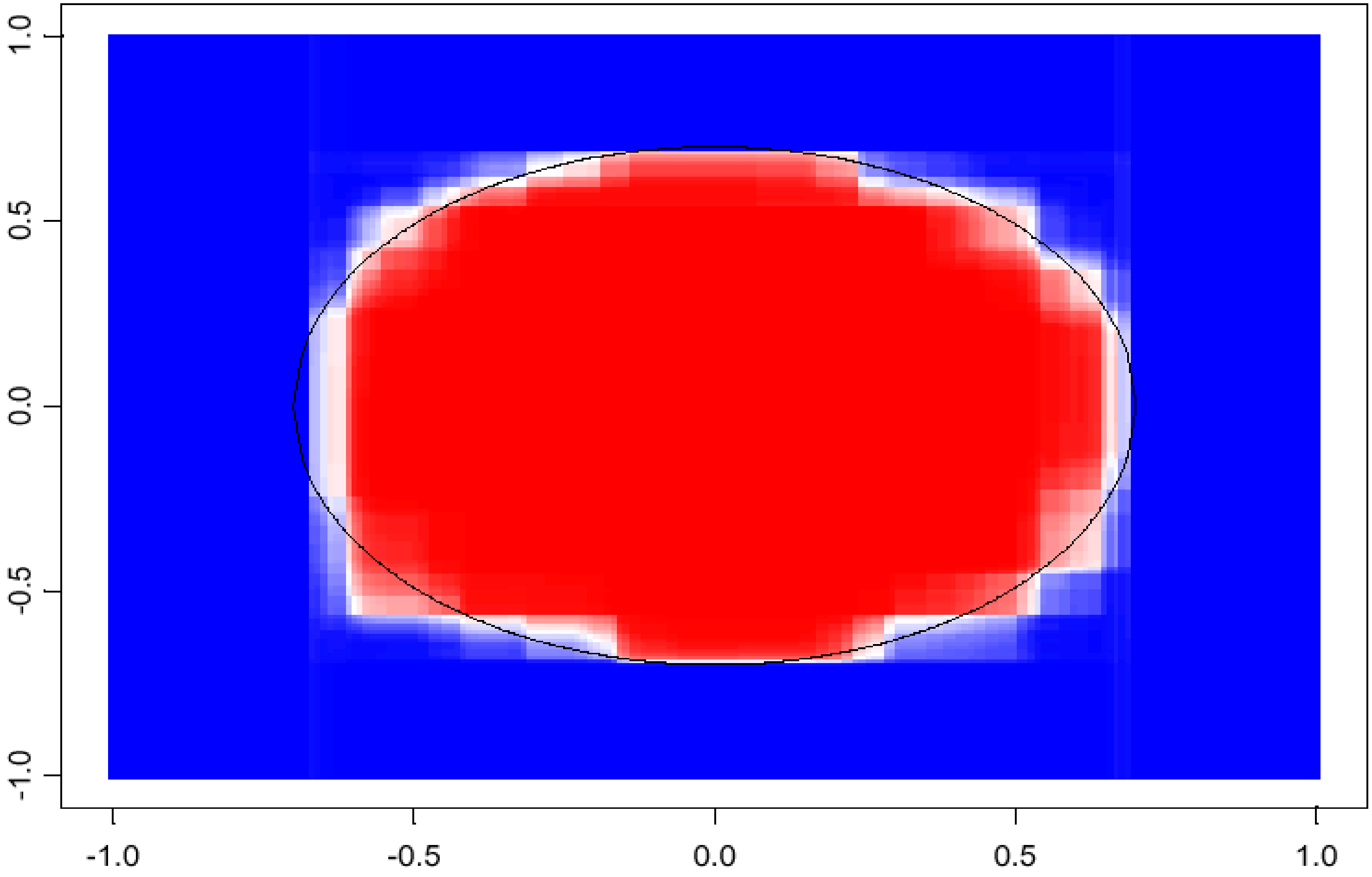
- **Build classifier on each bootstrap sample**

- **Each sample has probability $1 - (1 - 1/n)^n$ of being selected**

CART decision boundary



100 bagged trees



TU/e Technische Universiteit Eindhoven University of Technology

This lecture

- Combining classifiers
  - Bagging
  - Boosting
    - AdaBoost
  - Random Forest

- Conclusion

- Exercises

Boosting

- An iterative procedure to adaptively change distribution of training data by focusing more on previously misclassified records
  - Initially, all N records are assigned equal weights
  - Unlike bagging, weights may change at the end of boosting round

Boosting

- **Records that are wrongly classified will have their weights increased**
- **Records that are classified correctly will have their weights decreased**

Original Data	1	2	3	4	5	6	7	8	9	10
Boosting (Round 1)	7	3	2	8	7	9	4	10	6	3
Boosting (Round 2)	5	4	9	4	2	5	1	7	4	2
Boosting (Round 3)	4	4	8	10	4	5	4	6	3	4

- Example 4 is hard to classify
- Its weight is increased, therefore it is more likely to be chosen again in subsequent rounds

Example: AdaBoost

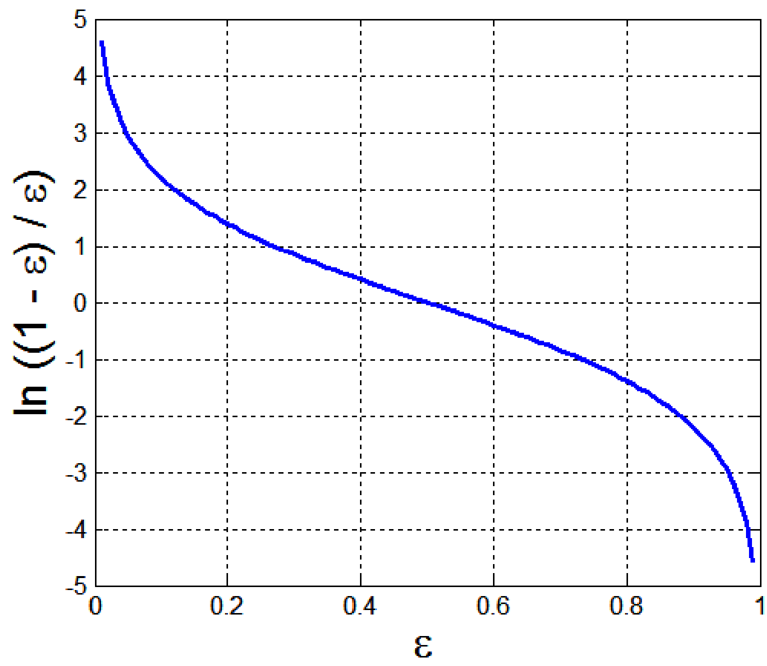
- **Base classifiers: C_1, C_2, …, C_T**

- **Error rate:**

$$\varepsilon_i = \frac{1}{N}\sum_{j=1}^{N} w_j \delta\left(C_i(x_j) \neq y_j\right)$$

- **Importance of a classifier:**

$$\alpha_i = \frac{1}{2}\ln\left(\frac{1-\varepsilon_i}{\varepsilon_i}\right)$$



Example: AdaBoost

- **Weight update:**

$$w_i^{(j+1)} = \frac{w_i^{(j)}}{Z_j} \begin{cases} \exp^{-\alpha_j} & \text{if } C_j(x_i) = y_i \\ \exp^{\alpha_j} & \text{if } C_j(x_i) \neq y_i \end{cases}$$

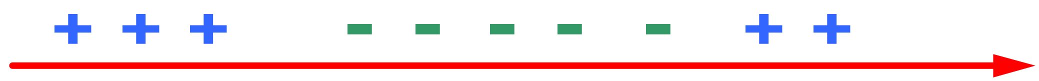
where Z_j is the normalization factor

- **If any intermediate rounds produce error rate higher than 50%, the weights are reverted back to 1/n and the resampling procedure is repeated**
- **Classification:**

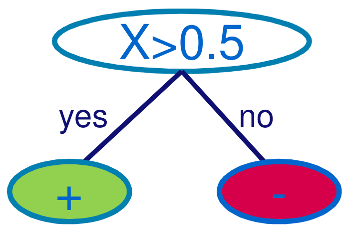
$$C^*(x) = \arg\max_y \sum_{j=1}^{T} \alpha_j \delta\left(C_j(x) = y\right)$$

Illustrating AdaBoost

- **One-dimensional input data:**

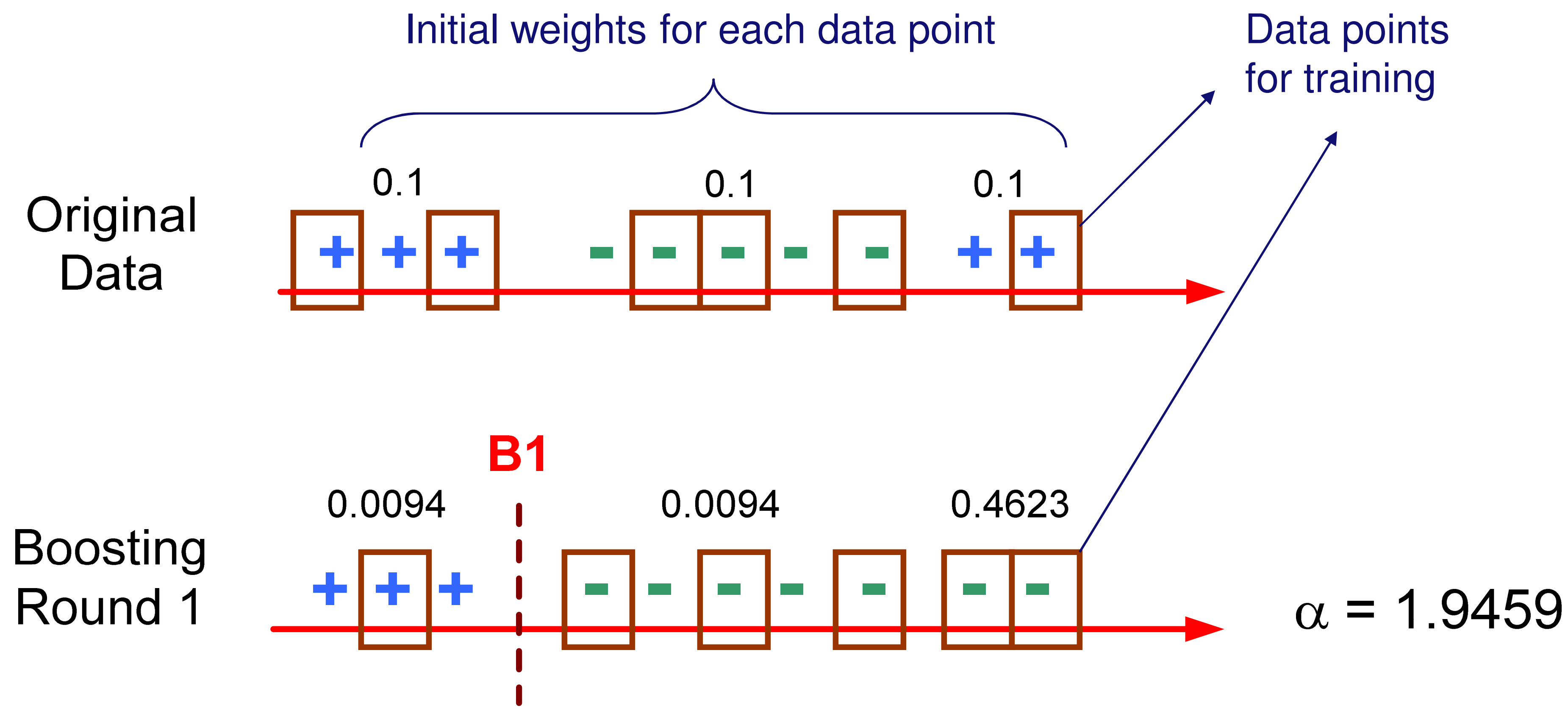


- **Base classifiers: *decision stumps***
  - **Decision trees of height two, with one split**

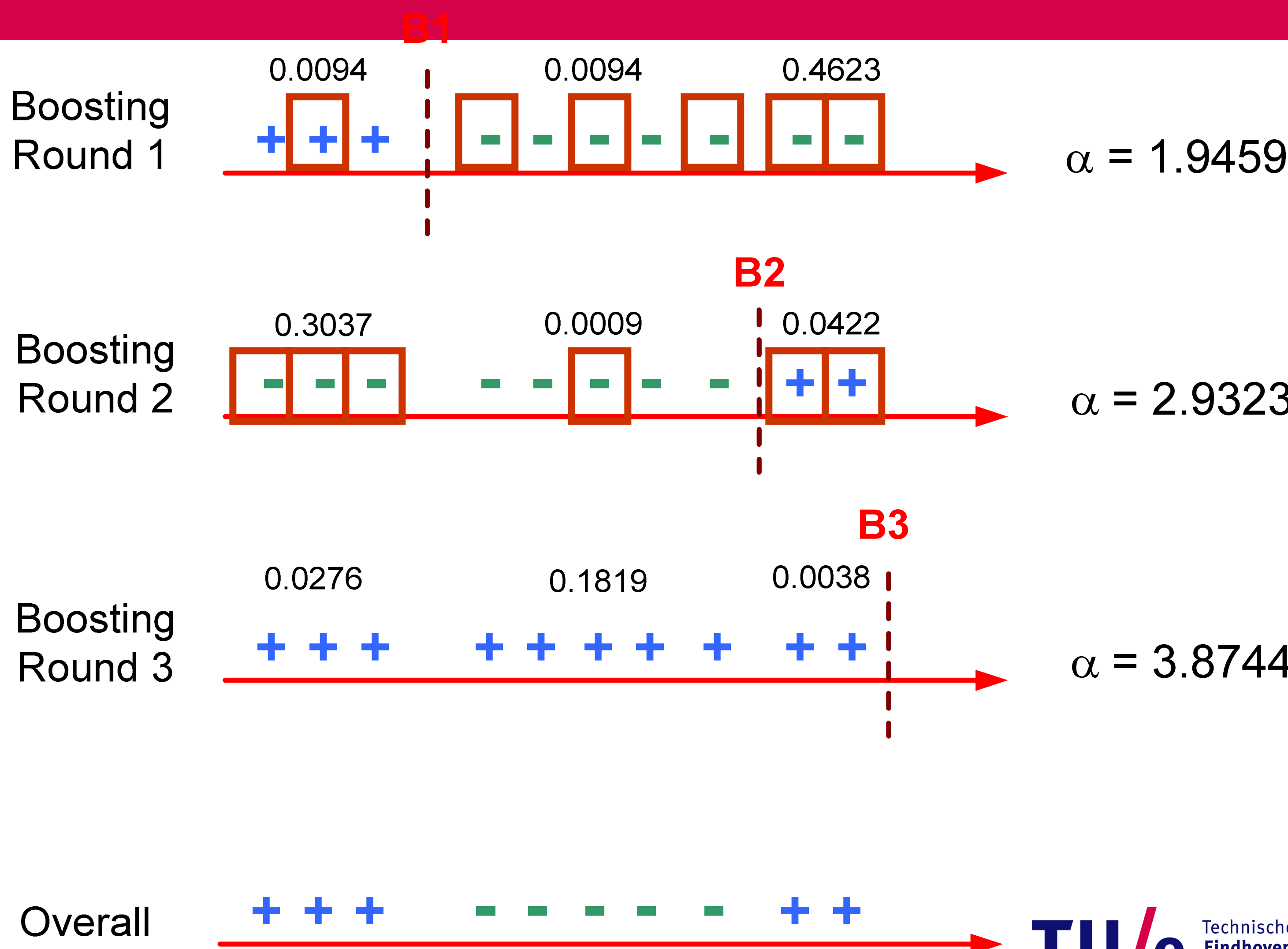


- **Maximal attainable accuracy: 80%**

Illustrating AdaBoost



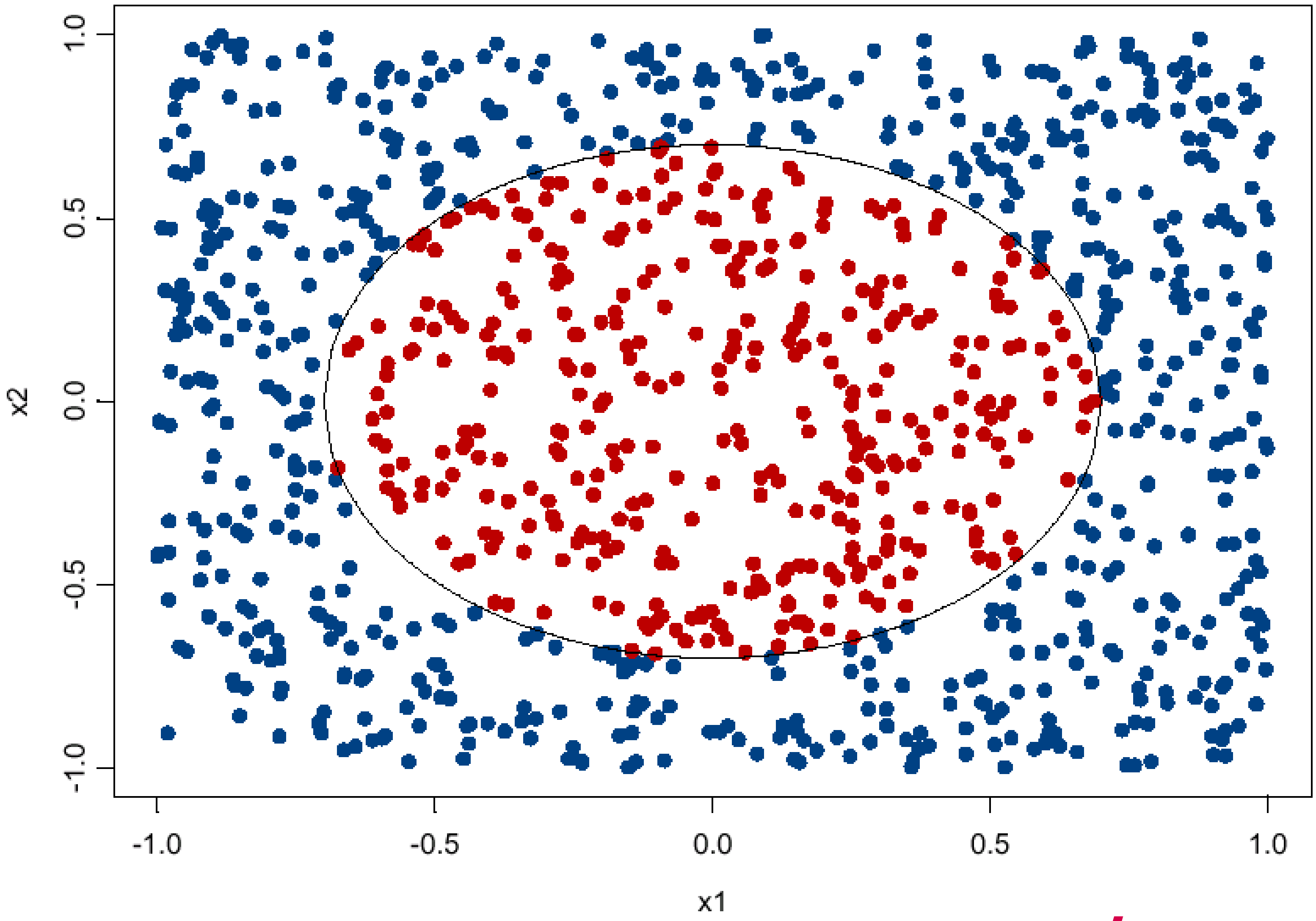
Illustrating AdaBoost



Boosting Example

http://www.cs.ucsd.edu/~yfreund/adaboost/index.html

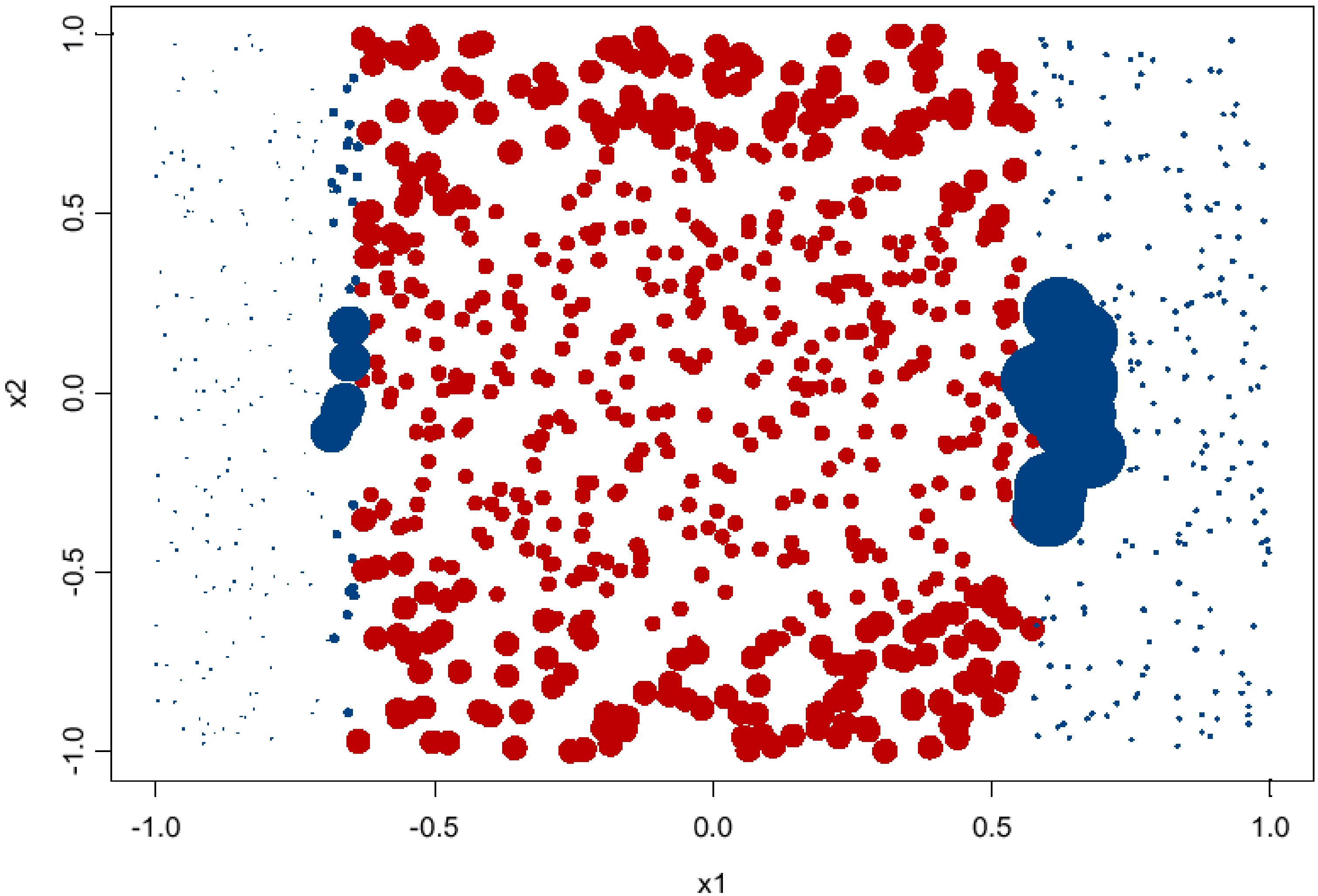
Boosting Example



siteit
nology

After one iteration

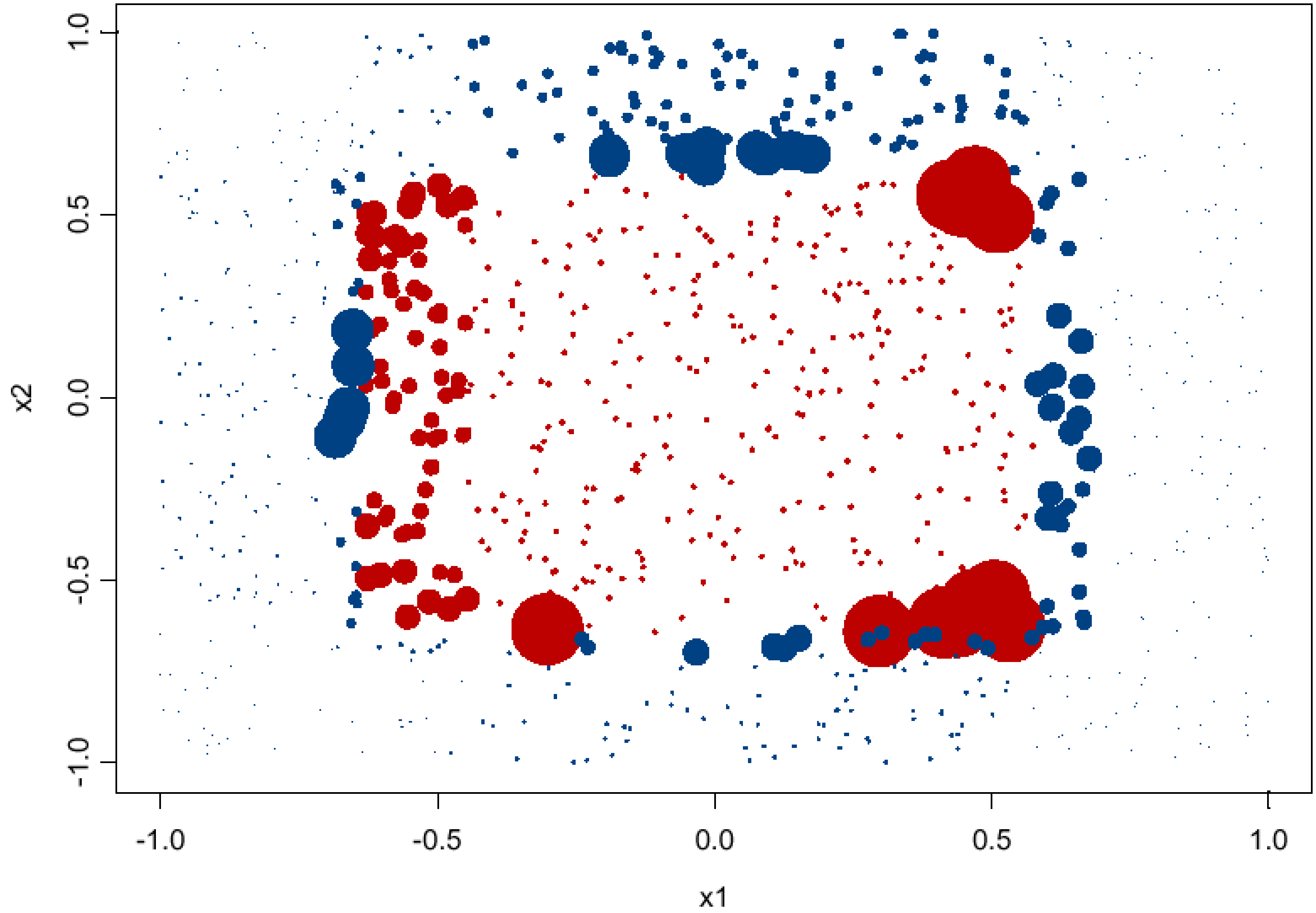
CART splits, larger points have great weight



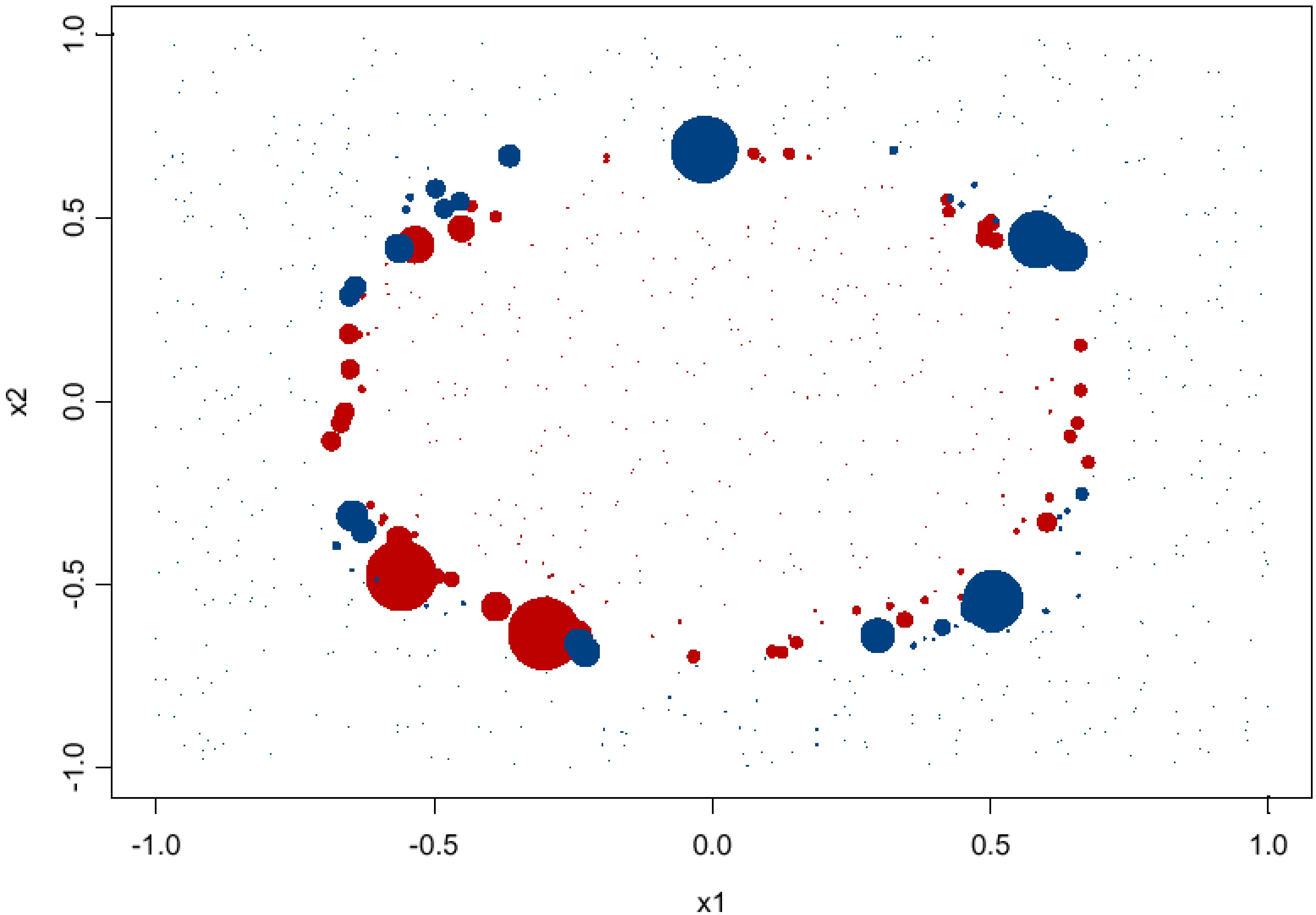
versiteit

echnology

After 3 iterations

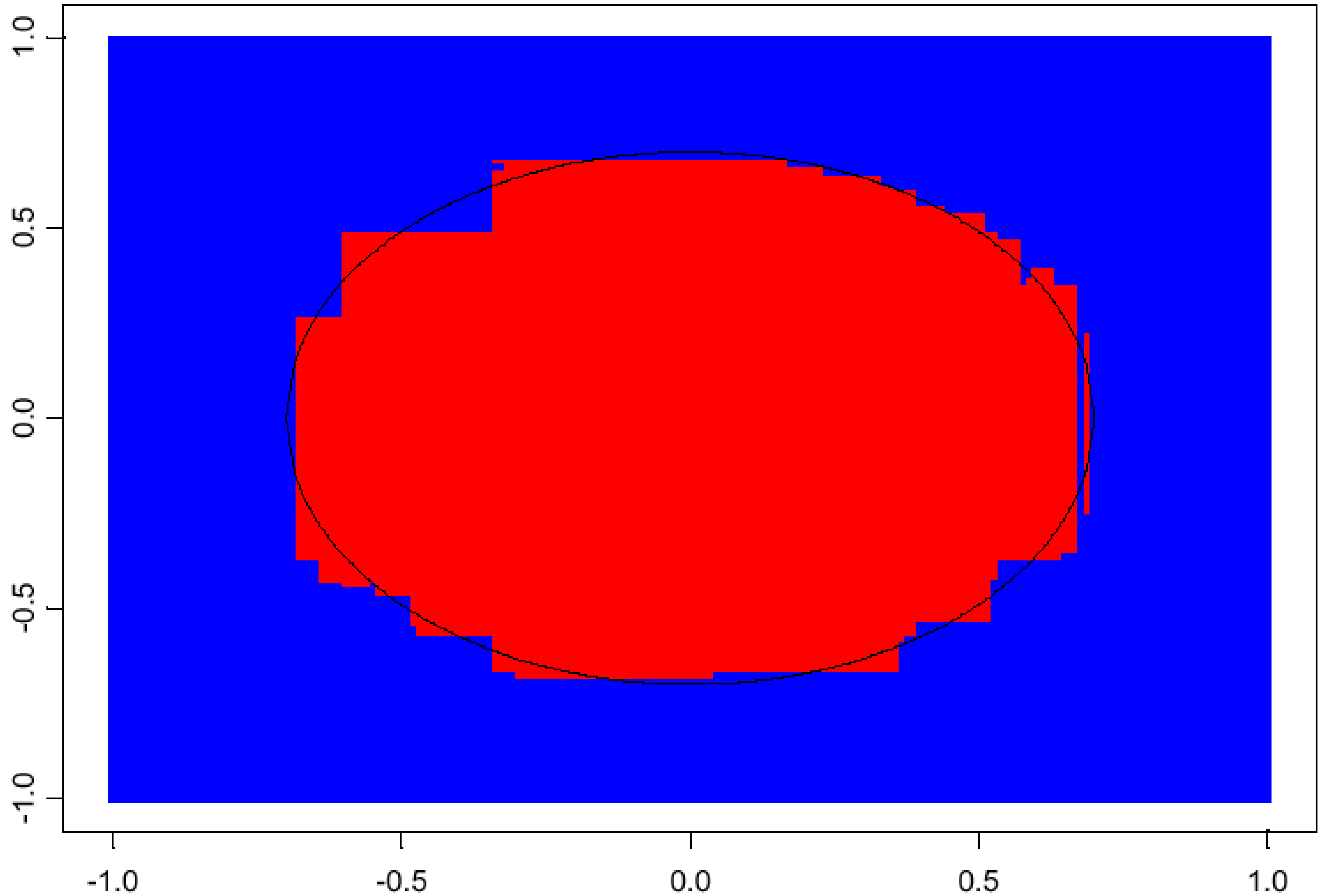


After 20 iterations



ersiteit
hnology

Decision boundary after 100 iterations



Theoretical Bounds

- It can be shown *on training data*:
  - Let ε_t denote the error of the t-th base classifier (on the modified data)
  - Let $\gamma_t = ½ - \varepsilon_t$

the training error is bounded by $\exp\left(-2\sum_t \gamma_t^2\right)$

- Hence, decreasing *exponentially fast*

This lecture

- Combining classifiers
  - Bagging
  - Boosting
    - AdaBoost
  - Random Forest

- Conclusion

- Exercises

Random Forest

- Ensemble of decision trees
- Input set:
  - N tuples, M attributes
- Each tree is learned on a reduced training set
  - Randomly select F<<M attributes
  - Sample training data
    - with replacement
    - Only keep randomly selected attributes
- State-of-the-art technique

This lecture

- Combining classifiers
  - Bagging
  - Boosting
    - AdaBoost
  - Random Forest
- Conclusion
- Exercises

Conclusion

- Ensembles to combine classifiers
  - On which data learn the classifier
  - How to combine the final classifiers
  - Weak base classifiers combined into one strong one
- Different choices lead to different *meta-learners*
  - Bagging
  - Boosting
  - Random Forrest
- Over-fitting of base classifiers not always bad

Overfitting in Ensembles

- Not that much research has been done into this topic
- A surprising recent finding:
  - ensembles of overfitting base classifiers are in many cases better than the ensembles of non-overfitting base classifiers
- This is related most probably to the fact that in that case the ensemble diversity is much higher

This lecture

- Combining classifiers
  - Bagging
  - Boosting
    - AdaBoost
  - Random Forest
- Conclusion
- Exercises

Exercises

- **Decision tree:** p. 198, Chapter 4, ex. 2
- **Naïve Bayes:** p. 318, Chapter 5, ex 7
- **Ensembles:**
  - Why is it important to have weak base classifiers?
  - Think of examples where the combination of strong base classifiers can be useful

Table 4.1. Data set for Exercise 2.

Customer ID	Gender	Car Type	Shirt Size	Class
1	M	Family	Small	C0
2	M	Sports	Medium	C0
3	M	Sports	Medium	C0
4	M	Sports	Large	C0
5	M	Sports	Extra Large	C0
6	M	Sports	Extra Large	C0
7	F	Sports	Small	C0
8	F	Sports	Small	C0
9	F	Sports	Medium	C0
10	F	Luxury	Large	C0
11	M	Family	Large	C1
12	M	Family	Extra Large	C1
13	M	Family	Medium	C1
14	M	Luxury	Extra Large	C1
15	F	Luxury	Small	C1
16	F	Luxury	Small	C1
17	F	Luxury	Medium	C1
18	F	Luxury	Medium	C1
19	F	Luxury	Medium	C1
20	F	Luxury	Large	C1

Compute Gini-indices

What is the best split?

Why is it a bad idea to split on CustomerID?

Table 5.1. Data set for Exercise 7.

Record	A	B	C	Class
1	0	0	0	+
2	0	0	1	−
3	0	1	1	−
4	0	1	1	−
5	0	0	1	+
6	1	0	1	+
7	1	0	1	−
8	1	0	1	−
9	1	1	1	+
10	1	0	1	+

Give the model the Naïve Bayesian classifier learns

a) Without m-estimate
b) With m-estimate; p=1/2, m=4
c) Predict in both cases the class of (A=0, B=1, C=0)