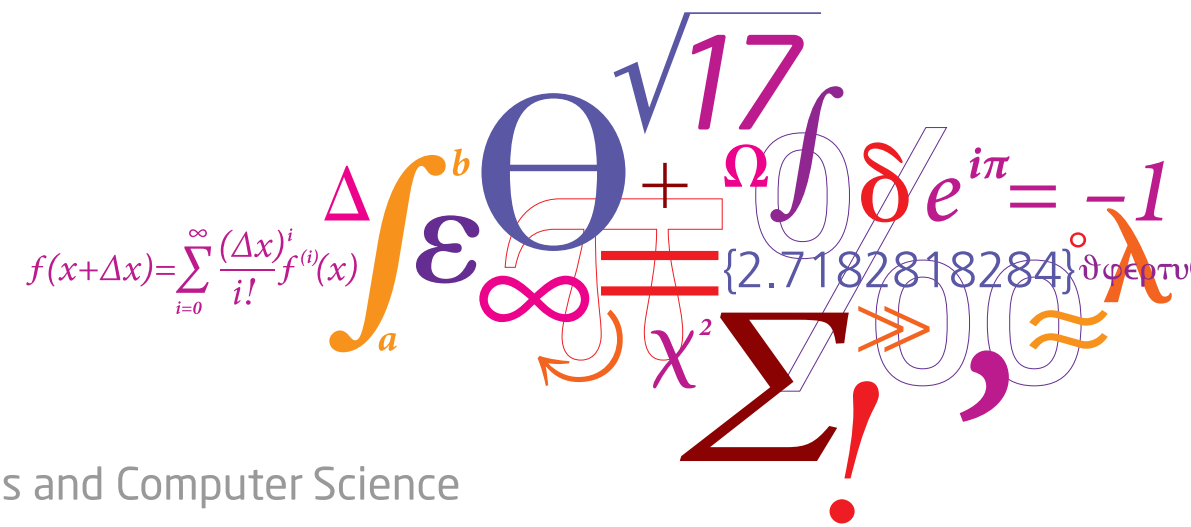
02450: Introduction to Machine Learning and Data Mining

AUC and ensemble methods

Morten Mørup and Mikkel N. Schmidt

DTU Compute, Technical University of Denmark (DTU)



DTU Compute
Department of Applied Mathematics and Computer Science

Lecture Schedule

1. Introduction
   7 October: C1

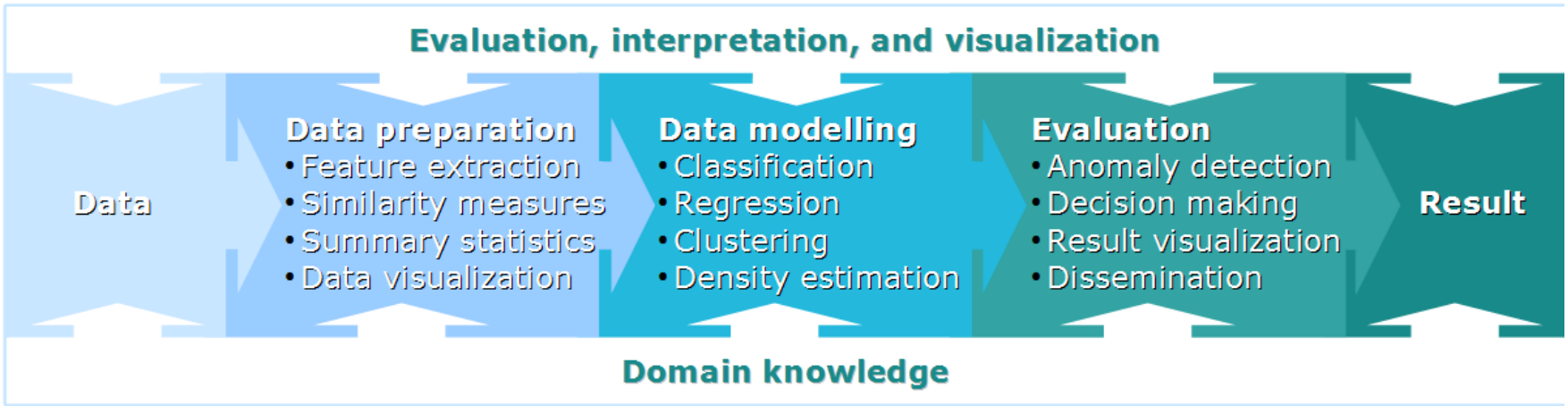
   Data: Feature extraction, and visualization
2. Data, feature extraction and PCA
   7 October: C2, C3
3. Measures of similarity, summary statistics and probabilities
   7 October: C4, C5
4. Probability densities and data Visualization
   7 October: C6, C7

   Supervised learning: Classification and regression
5. Decision trees and linear regression
   8 October: C8, C9
6. Overfitting, cross-validation and Nearest Neighbor
   8 October: C10, C12
7. Performance evaluation, Bayes, and Naive Bayes
   9 October: C11, C13
8. Artificial Neural Networks and Bias/Variance
   9 October: C14, C15
9. **AUC and ensemble methods**
   10 October: C16, C17

   Unsupervised learning: Clustering and density estimation
10. K-means and hierarchical clustering
    10 October: C18
11. Mixture models and density estimation
    11 October: C19, C20
12. Association mining
    11 October: C21

    Recap
13. Recap
    11 October: C1-C21

Piazza online help: `https://piazza.com/dtu.dk/fall2019/october2019`

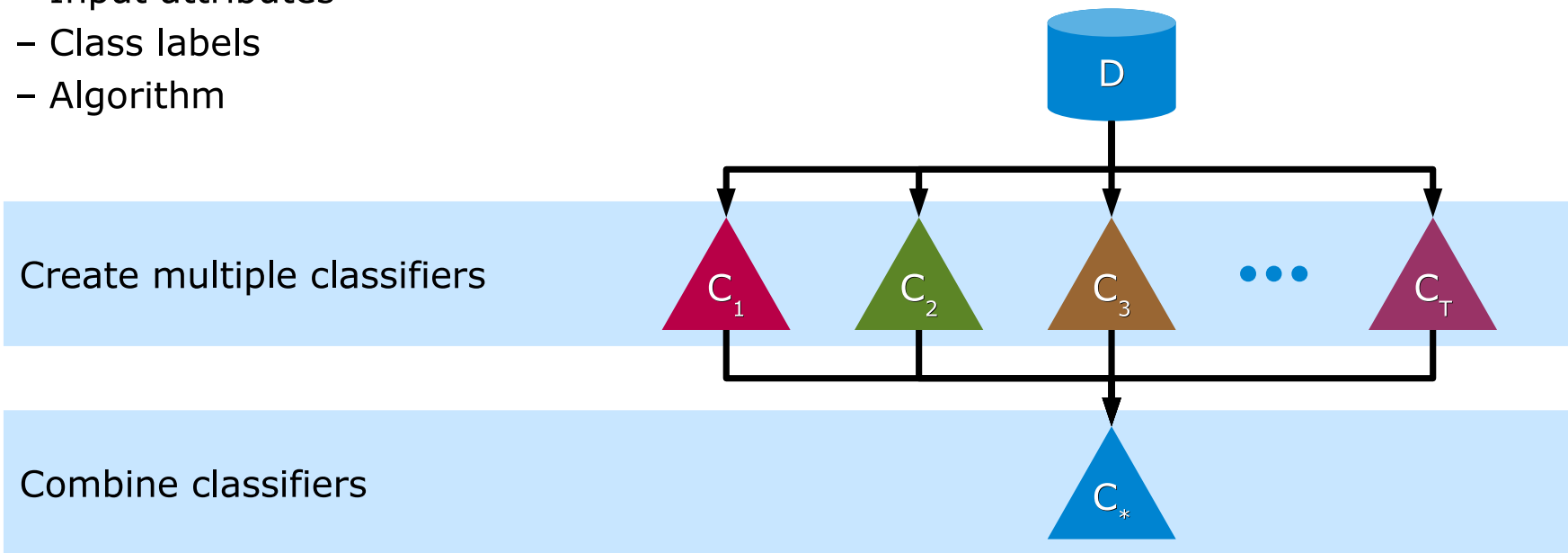


Learning Objectives

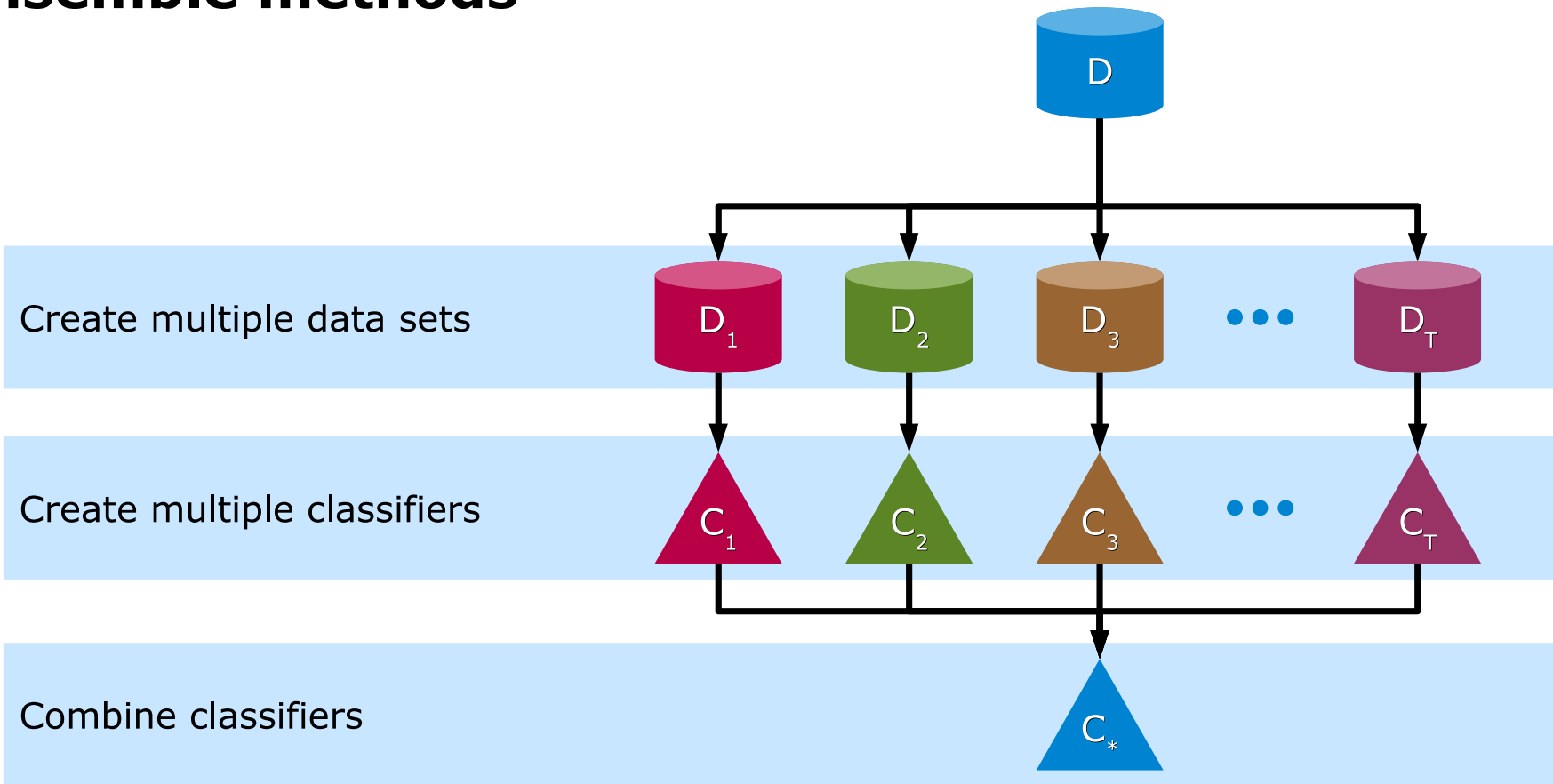
- Explain the principle behind boosting and bagging and apply it to improve classifiers
- Be able to address issues of class-imbalance and resampling
- Understand the definition of Precision, Recall, ROC, and AUC

Ensemble methods

- Combine multiple (weak) classifiers into one (strong) classifier
- Each classifier trained using different variations of
  - Data set
  - Input attributes
  - Class labels
  - Algorithm



Ensemble methods



Why ensemble methods?

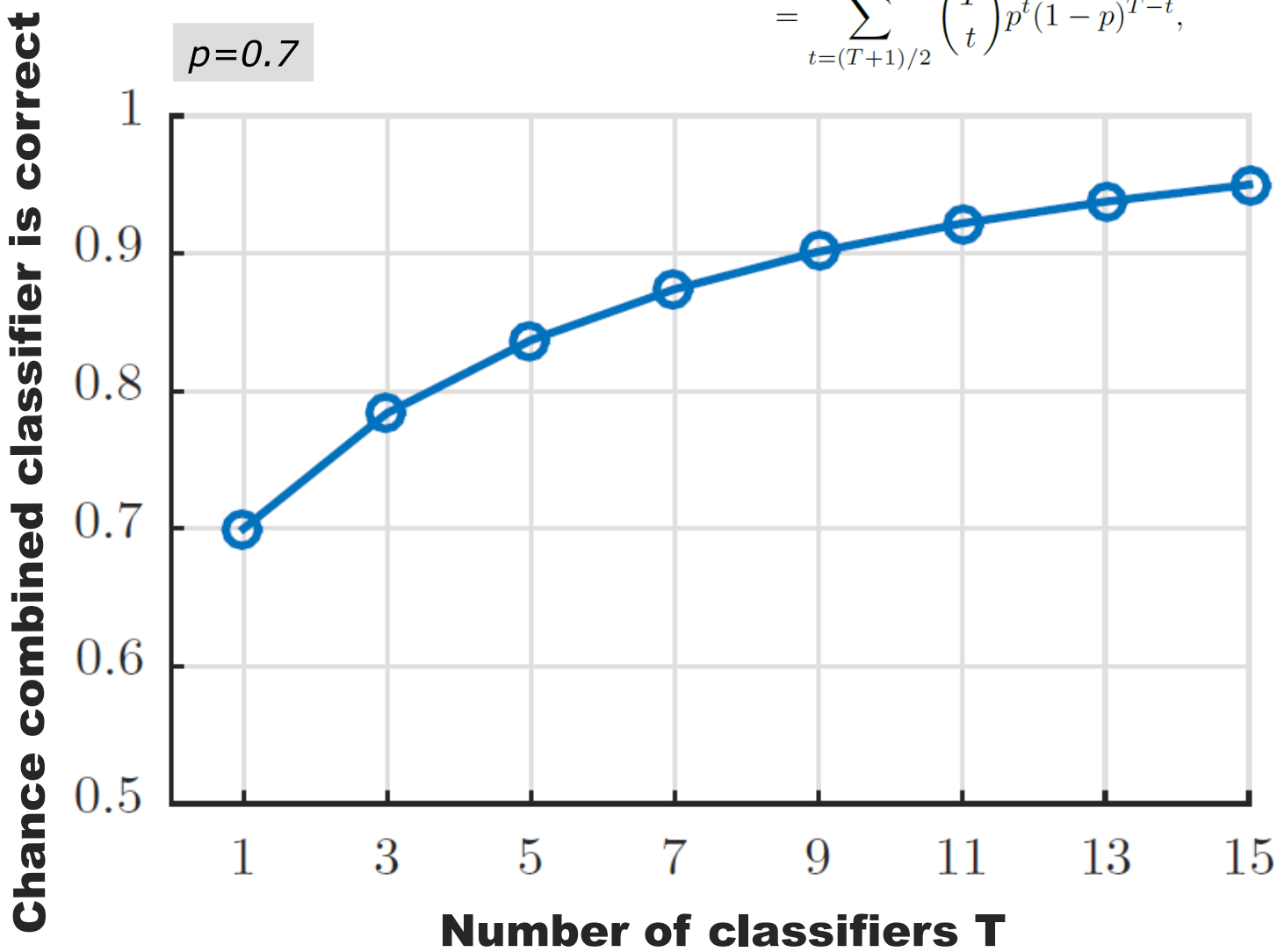
- Can improve classification algorithms in terms of
  - Better classification accuracy
  - Increased stability
  - Reduced variance
  - Less overfitting
- Consider T independent classifiers for binary classification, each with accuracy p. The probabilty a classifier which use majority voting is correct is then given by:

$$P(\text{Majority voting is correct}) = \sum_{t=(T+1)/2}^{T} \{t \text{ of the classifiers are correct}\}$$

$$= \sum_{t=(T+1)/2}^{T} \binom{T}{t} p^t (1-p)^{T-t},$$

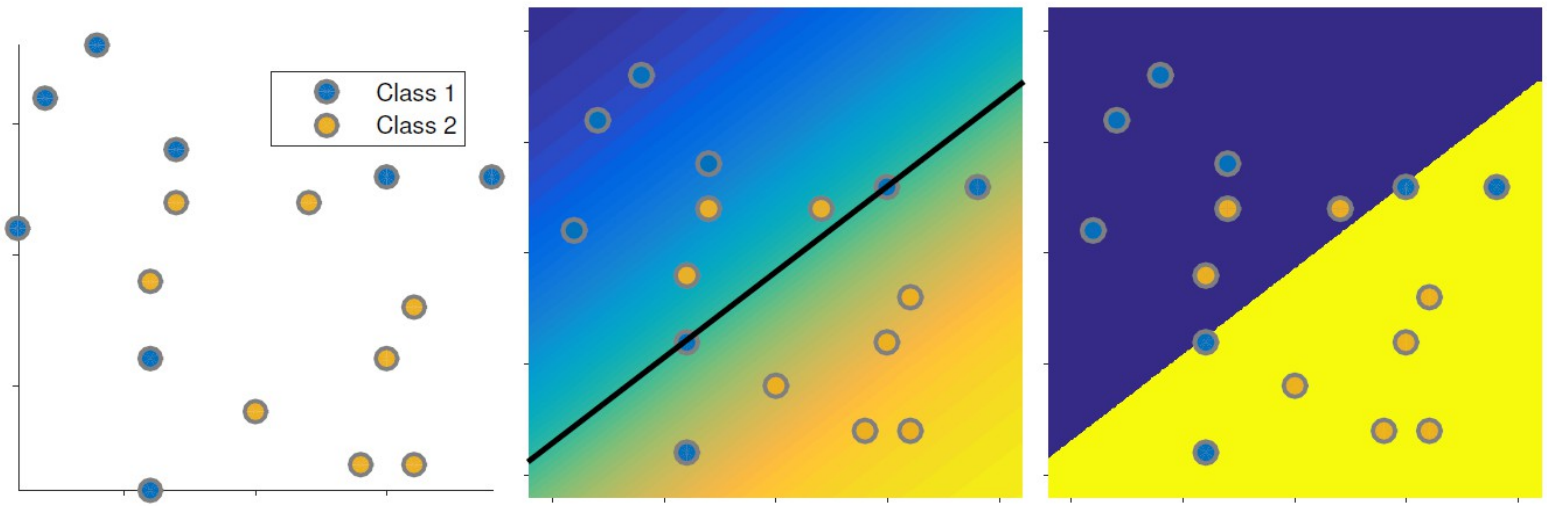
$$P(\text{Majority voting is correct}) = \sum_{t=(T+1)/2}^{T} \{t \text{ of the classifiers are correct}\}$$

$$= \sum_{t=(T+1)/2}^{T} \binom{T}{t} p^t (1-p)^{T-t},$$



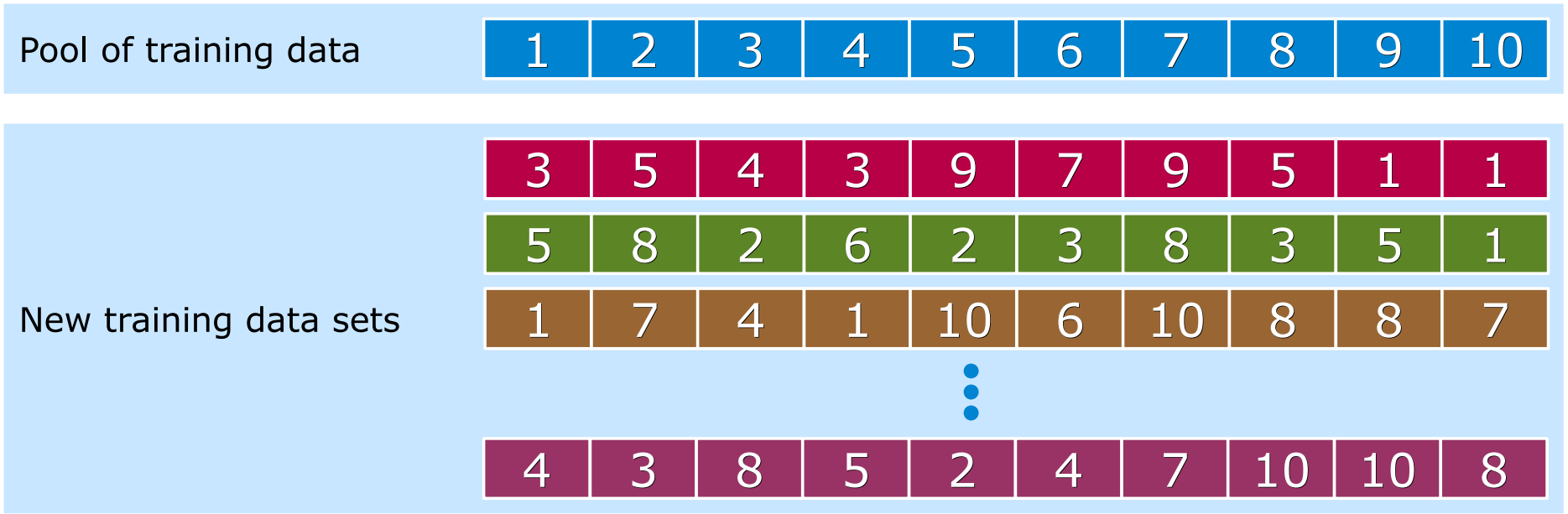
Data example

- Classification using logistic regression

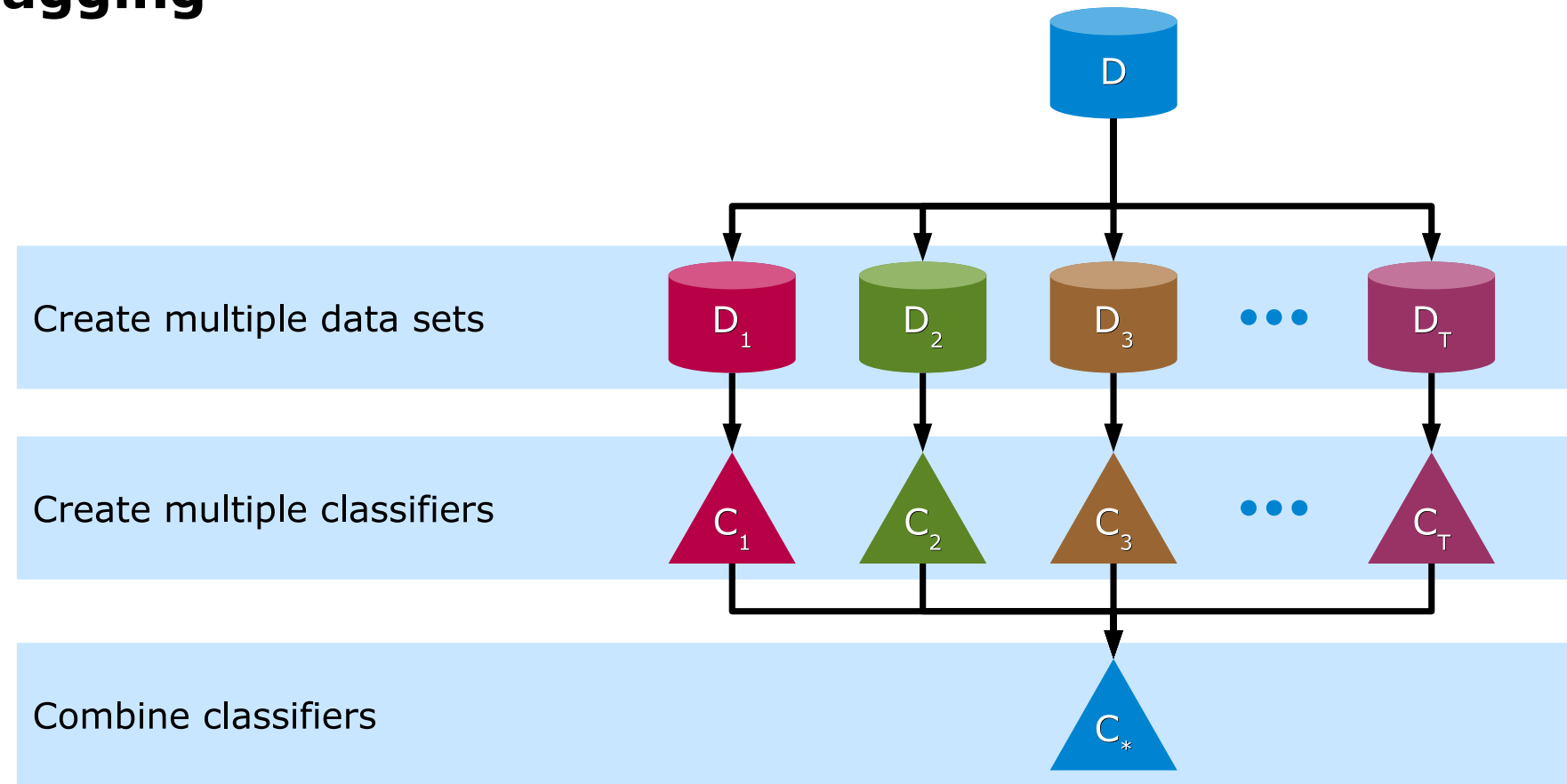


Bagging

- New training data sets drawn randomly from pool with replacement

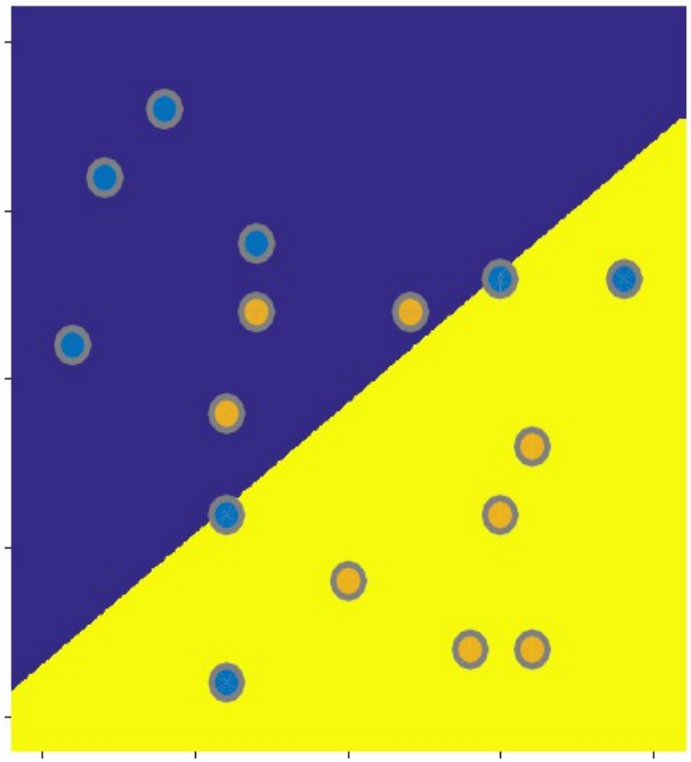


Bagging

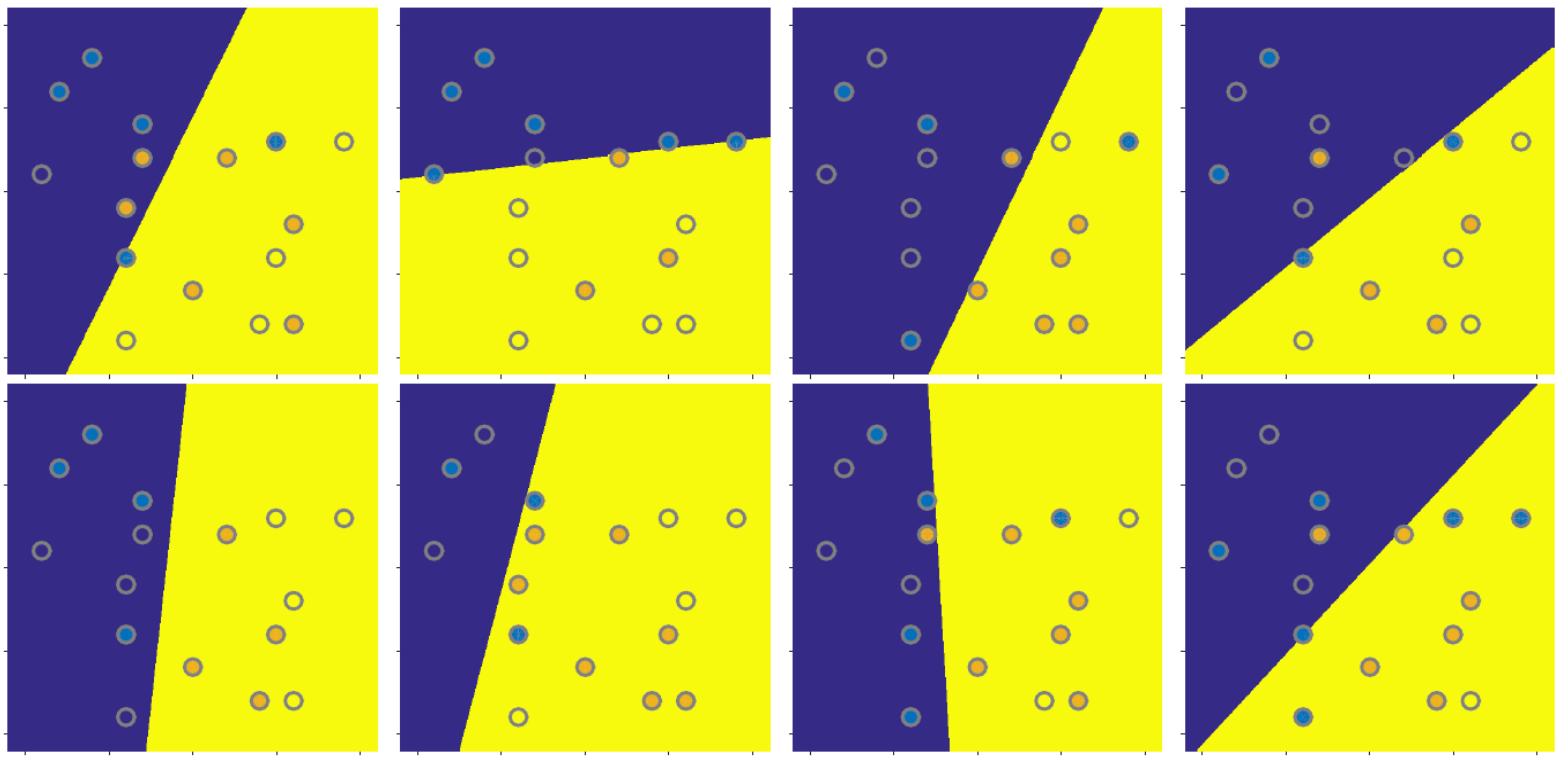

Create multiple data sets
Create multiple classifiers
Combine classifiers
D
D_1
D_2
D_3
D_T
C_1
C_2
C_3
C_T
C_*

Bagging

- **Single classifier**
  - Logistic regression
  - Two features, (x,y)



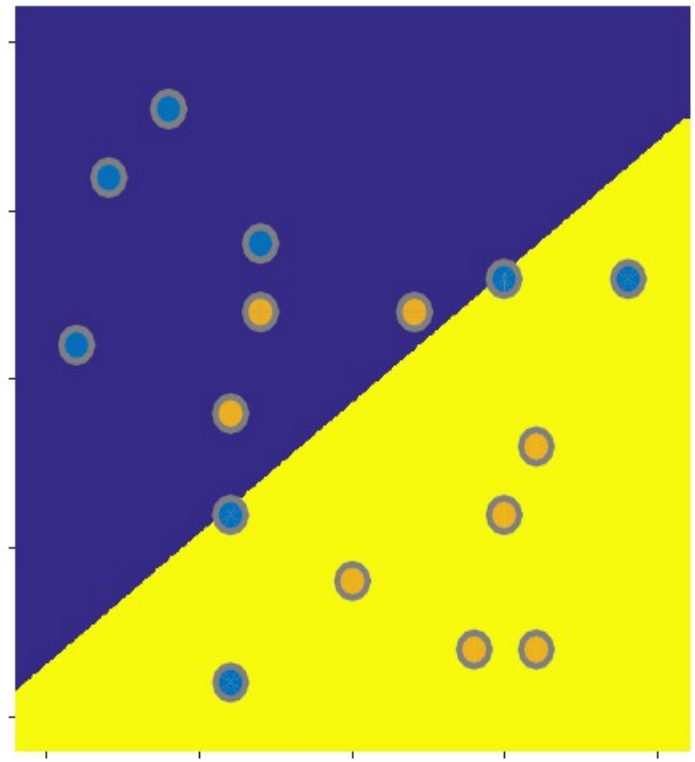
Bagging



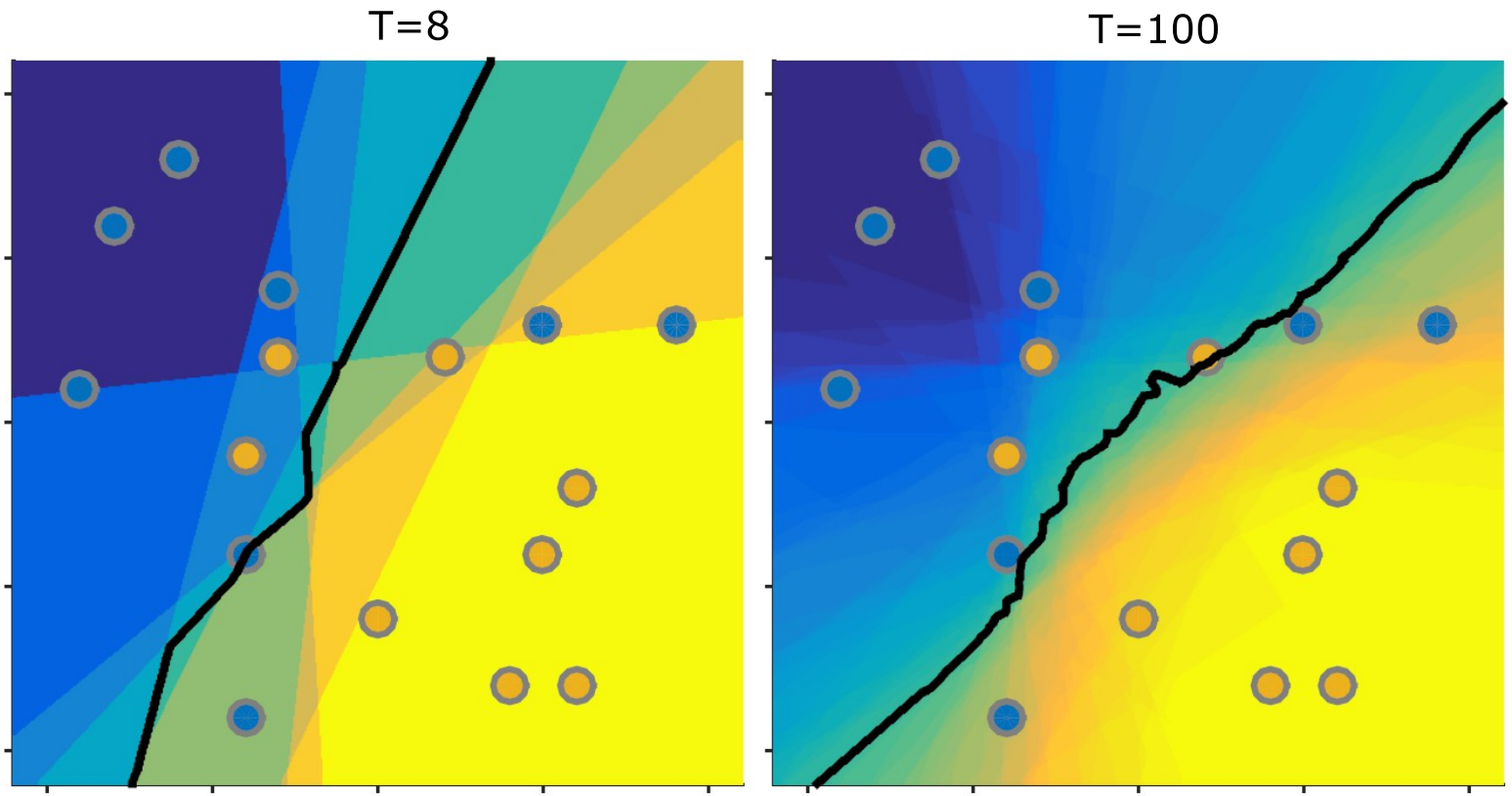
Notice, hollow dots are observations not included in bagging round

Bagging

- Single classifier



Bagging

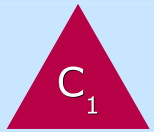


Boosting

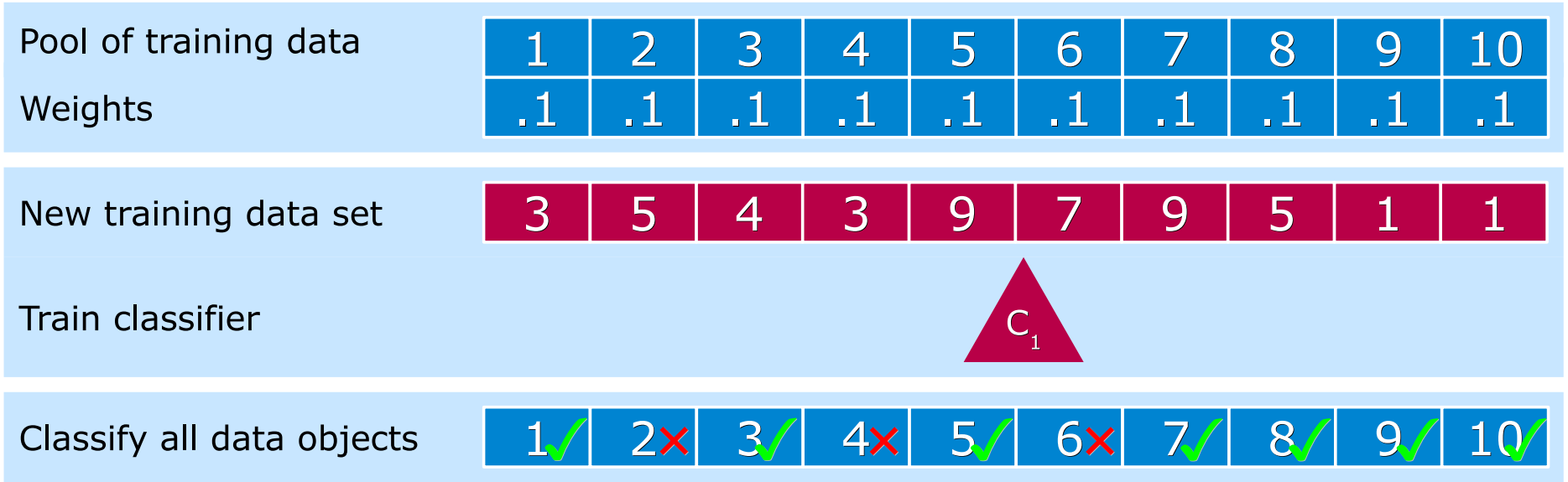
Pool of training data	1	2	3	4	5	6	7	8	9	10
Weights	.1	.1	.1	.1	.1	.1	.1	.1	.1	.1

New training data set	3	5	4	3	9	7	9	5	1	1

Train classifier



Boosting

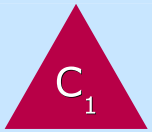

Pool of training data
1 2 3 4 5 6 7 8 9 10
Weights
.1 .1 .1 .1 .1 .1 .1 .1 .1 .1
New training data set
3 5 4 3 9 7 9 5 1 1
Train classifier
C1
Classify all data objects
1 2 3 4 5 6 7 8 9 10

Boosting

Pool of training data	1	2	3	4	5	6	7	8	9	10
Weights	.1	.1	.1	.1	.1	.1	.1	.1	.1	.1

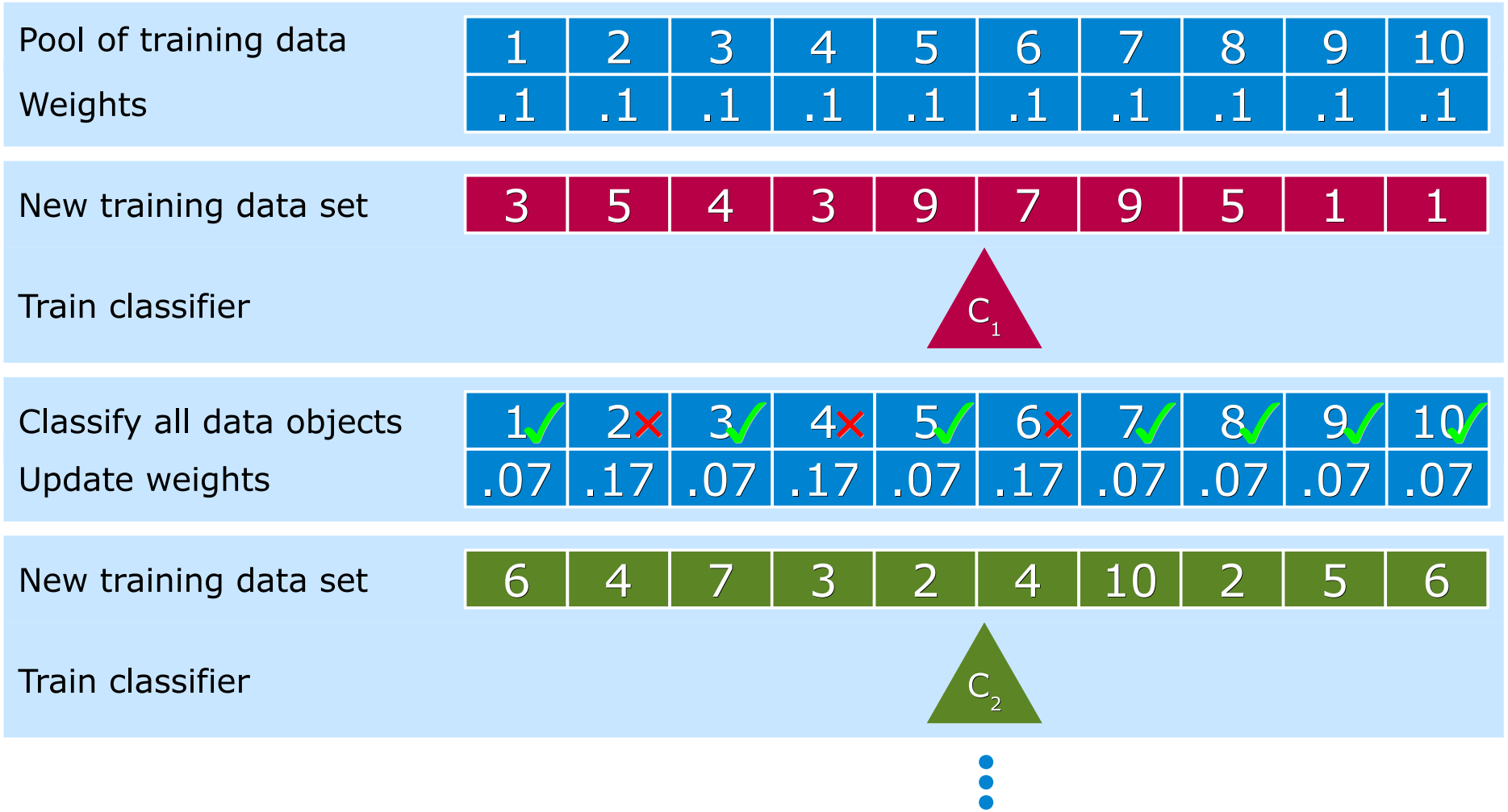
New training data set	3	5	4	3	9	7	9	5	1	1

Train classifier



Classify all data objects	1✓	2✗	3✓	4✗	5✓	6✗	7✓	8✓	9✓	10✓
Update weights	.07	.17	.07	.17	.07	.17	.07	.07	.07	.07

Boosting


Pool of training data
1 2 3 4 5 6 7 8 9 10
Weights
.1 .1 .1 .1 .1 .1 .1 .1 .1 .1
New training data set
3 5 4 3 9 7 9 5 1 1
Train classifier
C1
Classify all data objects
1✓ 2✗ 3✓ 4✗ 5✓ 6✗ 7✓ 8✓ 9✓ 10✓
Update weights
.07 .17 .07 .17 .07 .17 .07 .07 .07 .07
New training data set
6 4 7 3 2 4 10 2 5 6
Train classifier
C2

AdaBoost

Algorithm 6: AdaBoost algorithm

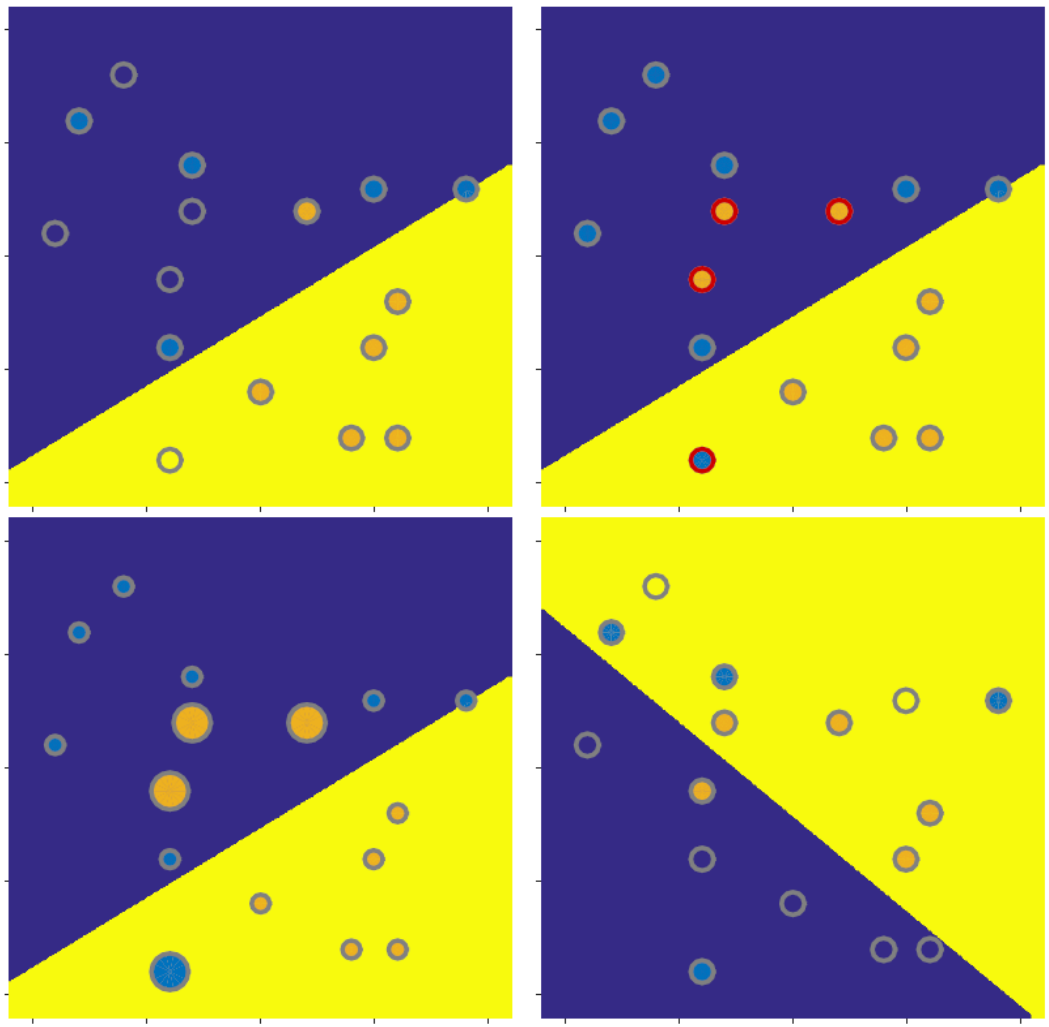
1: Initialize $w_i(1) = \frac{1}{N}$ for $i = 1, \ldots, N$
2: **for** $t = 1, \ldots, T$ **do**
3: Create $\mathcal{D}_t$ by sampling (with replacement) from $\mathcal{D}$ according to $\boldsymbol{w}(t)$
4: Let f_t be the classifier *trained* on $\mathcal{D}_t$
5: $\epsilon_t = \sum_{i=1}^{N} w_i(t) \left(1 - \delta_{f_t(\boldsymbol{x}_i), y_i}\right)$ *(weighted error of f_t on all data).*
6: $\alpha_t = \frac{1}{2} \log \frac{1-\epsilon_t}{\epsilon_t}$
7: For each i update weights using eq. (15.7):

$$w_i(t+1) = \frac{\tilde{w}_i(t+1)}{\sum_{j=1}^{N} \tilde{w}_j(t+1)}, \quad \tilde{w}_i(t+1) = \begin{cases} w_i(t) e^{-\alpha_t} & \text{if } f_t(\boldsymbol{x}_i) = y_i \\ w_i(t) e^{\alpha_t} & \text{if } f_t(\boldsymbol{x}_i) \neq y_i. \end{cases}$$

8: **end for**
9: $f^*(\boldsymbol{x}) = \arg\max_{y=1,2} \sum_{t=1}^{T} \alpha_t \delta_{f_t(\boldsymbol{x}), y}$ *(Majority voting classifier)*

Boosting

A:
A dataset is sampled with replacement and a classifier trained.

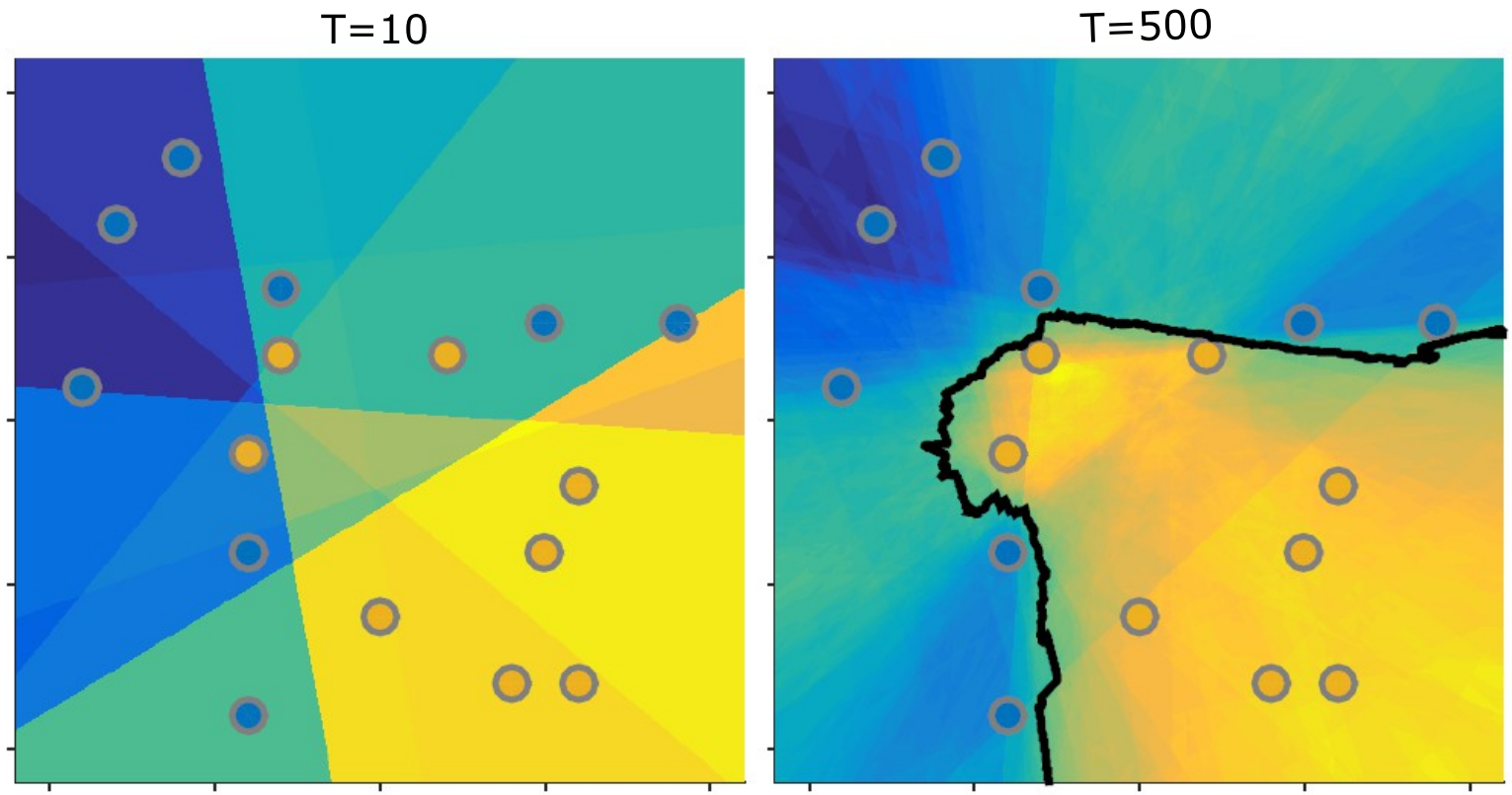


B:
Mis-classified observations are identified.

C:
Weighs are updated such that more emphasis is given to these mis-classified observations.

New round:
Based on the updated weights a new dataset is sampled and a classifier trained (shown), mis-classified observations identified and given more emphasis...

Boosting

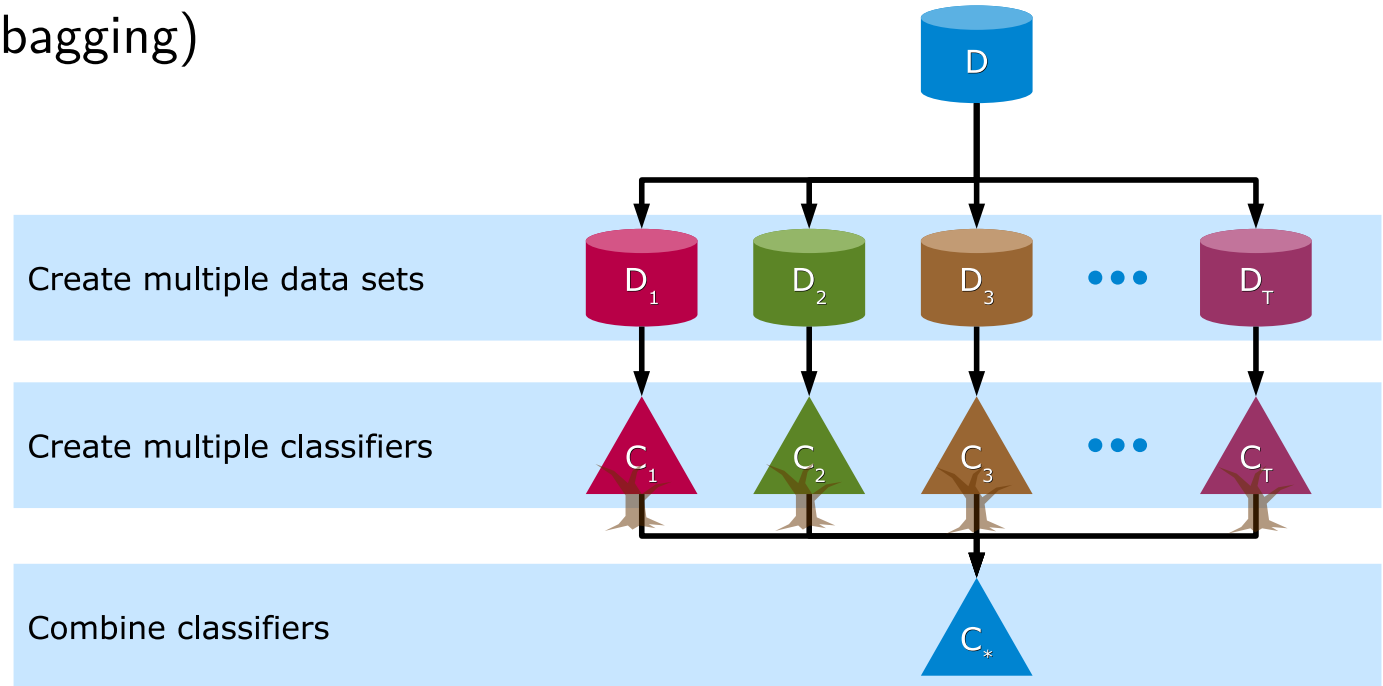


Bagging example: Random forest

Each tree is generated as follows:

- Sample dataset with replacement
- When generating each node in the tree, randomly select a subset of the features and only consider splits using these features

A large number of trees are generated and the trees are combined using majority voting (bagging)



Quiz 01 (please answer on Piazza): Adaboost (Spring 2016)

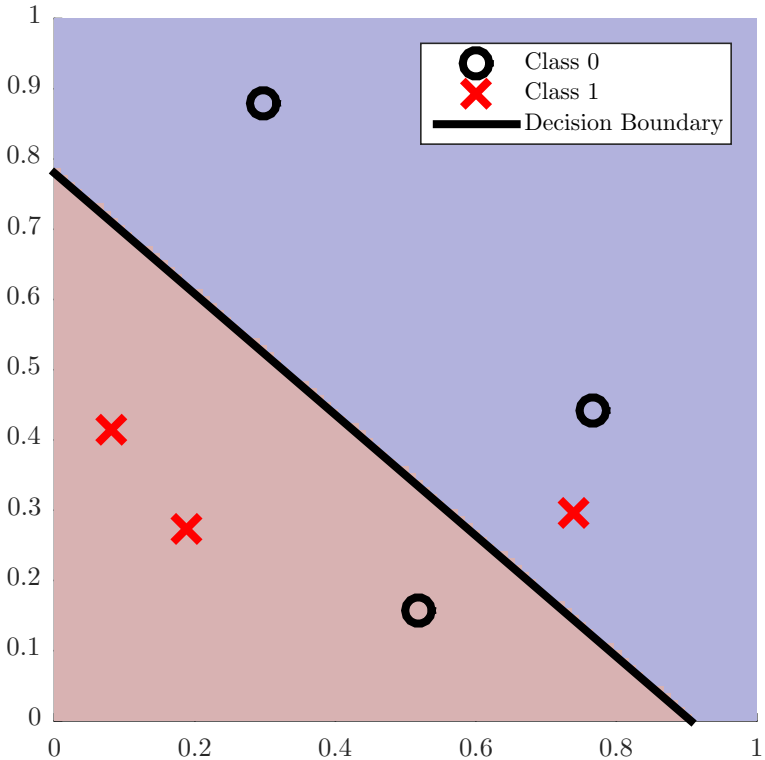


Figure 1: A binary classification problem and the decision boundary obtained by logistic regression. Observations left of the boundary are classified as belonging to the positive class 1 (red crosses) and observations right of the boundary to the negative class 0 (black circles)

We wish to apply a logistic regression model to the binary classification problem shown in Figure 1. We attempt to improve the performance by applying AdaBoost. AdaBoost works by first sampling a new dataset with replacement, then training a classifier on the dataset and then proceeding with the subsequent steps of the AdaBoost algorithm.

Suppose in the first iteration of the AdaBoost algorithm the classification boundary of the trained classifier is as indicated by the black line (i.e. observations left of the black line are classified as in the positive class). What is the resulting value of the weights $\boldsymbol{w}$?

A. $\boldsymbol{w} = \begin{bmatrix} 0.1 & 0.250 & 0.125 & 0.125 & 0.125 & 0.250 \end{bmatrix}$

B. $\boldsymbol{w} = \begin{bmatrix} 0.026 & 0.447 & 0.026 & 0.026 & 0.026 & 0.447 \end{bmatrix}$

C. $\boldsymbol{w} = \begin{bmatrix} 0.235 & 0.029 & 0.235 & 0.235 & 0.235 & 0.029 \end{bmatrix}$

D. $\boldsymbol{w} = \begin{bmatrix} 0.1 & 0.3 & 0.1 & 0.1 & 0.1 & 0.3 \end{bmatrix}$

E. Don't know.

(Hint: First compute ε_1, then α_1, then the weights)

Solution:

The classifier classifies two out of $N = 6$ observations incorrectly. We therefore have:

$$\varepsilon_i = \left[\sum_{j=1}^{N} w_j I\left(\hat{y}_j \neq y_j\right)\right]$$

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

and accordingly $\varepsilon_1 = \frac{1}{N} \times 2 = \frac{1}{3}$. This gives

$$\alpha_1 = \frac{1}{2}\log\frac{1-\frac{1}{3}}{\frac{1}{3}} = \frac{1}{2}\log 2$$

and so for $\boldsymbol{w}$ we get

$$\boldsymbol{w} \propto \begin{bmatrix} e^{-\alpha_1} & e^{\alpha_1} & e^{-\alpha_1} & e^{-\alpha_1} & e^{-\alpha_1} & e^{\alpha_1} \end{bmatrix}$$

Simplifying by moving $\frac{1}{\sqrt{2}}$ outside the vector:

$$\boldsymbol{w} \propto \begin{bmatrix} 1 & 2 & 1 & 1 & 1 & 2 \end{bmatrix}$$

and normalizing:

$$\boldsymbol{w} = \frac{1}{8}\begin{bmatrix} 1 & 2 & 1 & 1 & 1 & 2 \end{bmatrix}$$

accordingly option A is correct.

Class imbalance problem

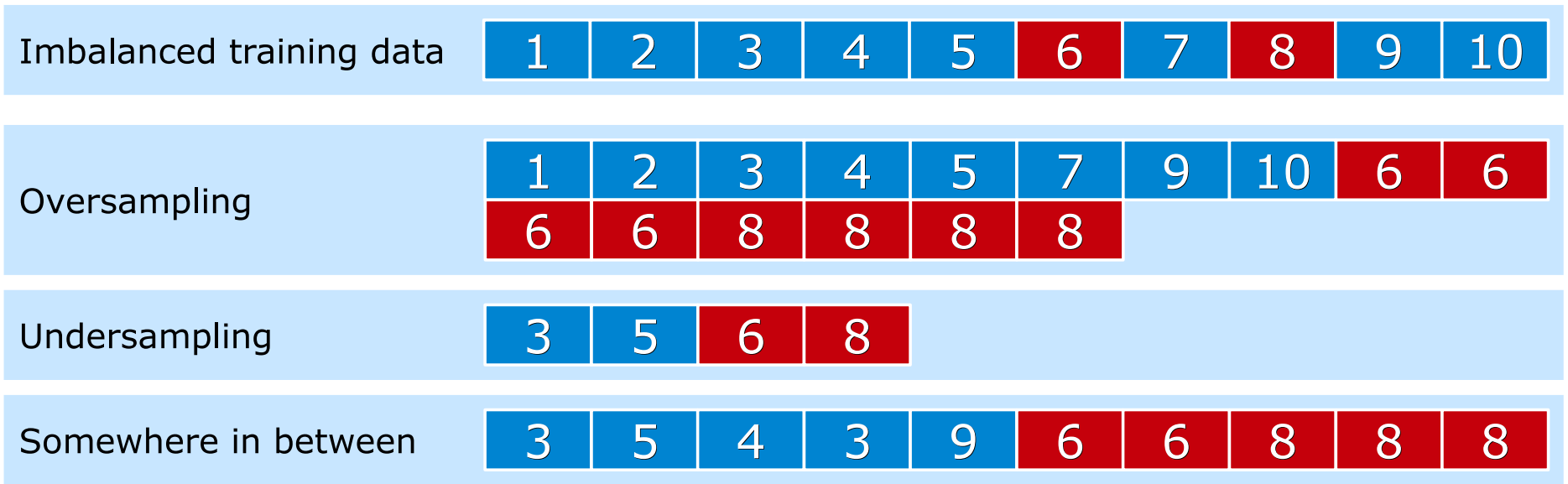
- Many data sets have **imbalanced class distributions**
  - Example: Detection of defects that only occur rarely (e.g. 1/1,000,000)
  - Danger: Algorithm that says nothing is defect will be 99.999% correct

- **Solution approaches**
  - Resample to balance data sets
  - Modify existing classification algorithms
  - Measure performance in a way that takes balance into account

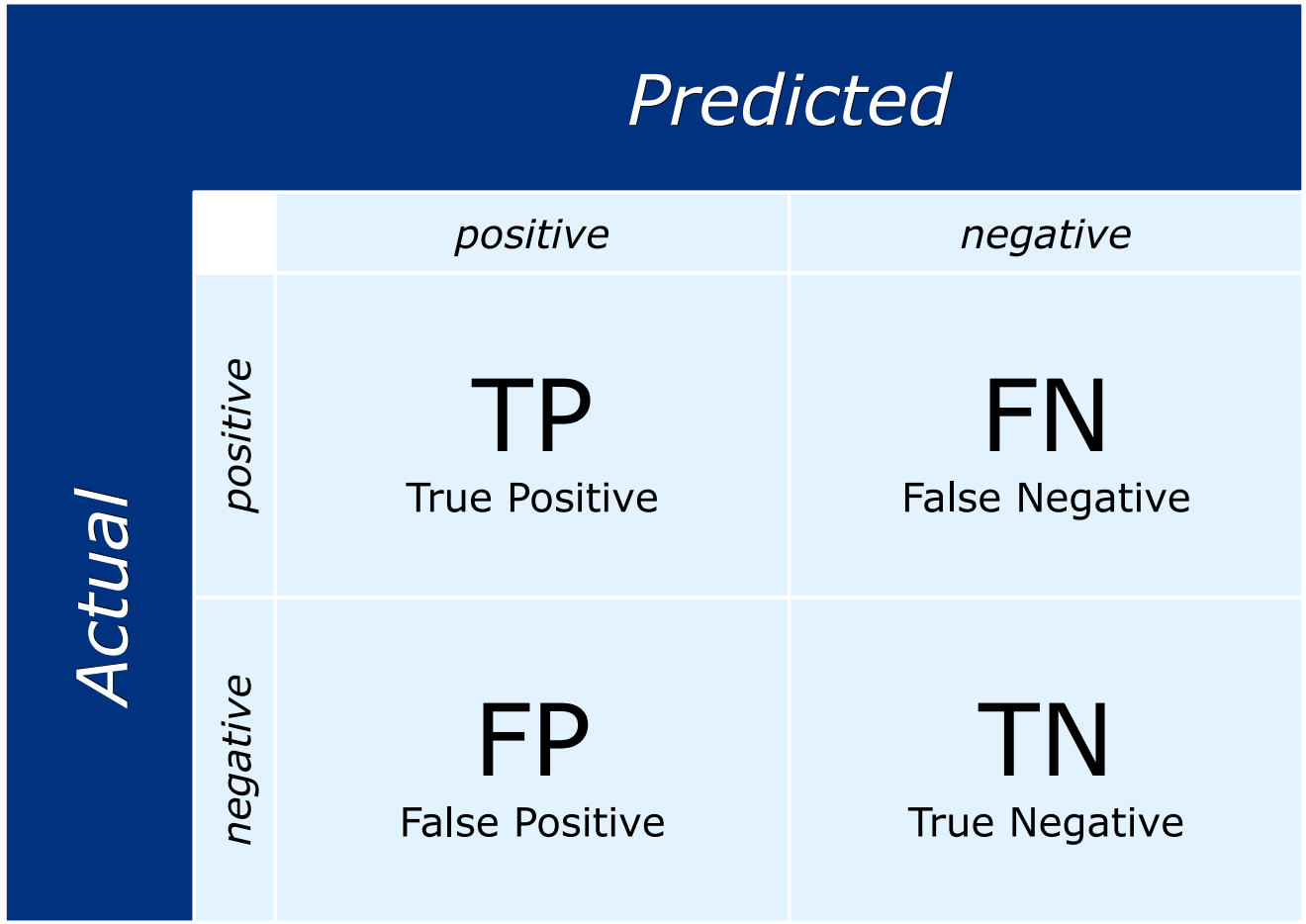
Resampling balanced data

- New sample has equal number of data objects from each class

- **Approaches**
  - **Undersampling** majority class: Throws out potentially useful data
  - **Oversampling** minority class: Increase data size and computational burden
  - **Somewhere in between**...



Confusion matrix



Precision and recall

- **Precision**
  - Fraction of true positive among objects predicted to be positive

$$p = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

- **Recall**
  - Fraction of objects predicted to be positive among all positive objects

$$r = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

Actual \ Predicted	*positive*	*negative*
positive	TP True Positive	FN False Negative
negative	FP False Positive	TN True Negative

Quiz 02 (please answer on Piazza): Precision/Recall

Consider two different classifiers, and suppose on a test set with 20 positive observations:

- Classifier 1 detects 54 positive of which 18 are actually positive
- Classifier 2 detects 16 positive of which 14 are actually positive

Actual \ Predicted	positive	negative
positive	TP True Positive	FN False Negative
negative	FP False Positive	TN True Negative

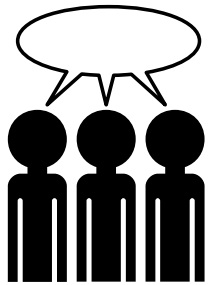
What is the precision and recall of the two classifiers?

A. Classifier 1: $p_1 = \frac{2}{3}, r_1 = \frac{7}{10}$
Classifier 2: $p_2 = \frac{1}{3}, r_2 = \frac{1}{3}$

B. Classifier 1: $p_1 = \frac{1}{3}, r_1 = \frac{9}{10}$
Classifier 2: $p_2 = \frac{2}{3}, r_2 = \frac{9}{10}$

C. Classifier 1: $p_1 = \frac{2}{3}, r_1 = \frac{7}{10}$
Classifier 2: $p_2 = \frac{1}{3}, r_2 = \frac{9}{10}$

D. Classifier 1: $p_1 = \frac{1}{3}, r_1 = \frac{9}{10}$
Classifier 2: $p_2 = \frac{7}{8}, r_2 = \frac{7}{10}$



Which classifier would you use if the objective was to detect credit-card fraud (the positive class corresponds to fraud)

- **Precision**
  - Fraction of true positive among objects predicted to be positive

$$p = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

- **Recall**
  - Fraction of objects predicted to be positive among all positive objects

$$r = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

Solution:

The precision is the number of true positives divided by the number *predicted* to be positive. I.e. for the two classifiers

$$p_1 = \frac{18}{54} = \frac{1}{3}, \quad p_2 = \frac{14}{16} = \frac{7}{8}.$$

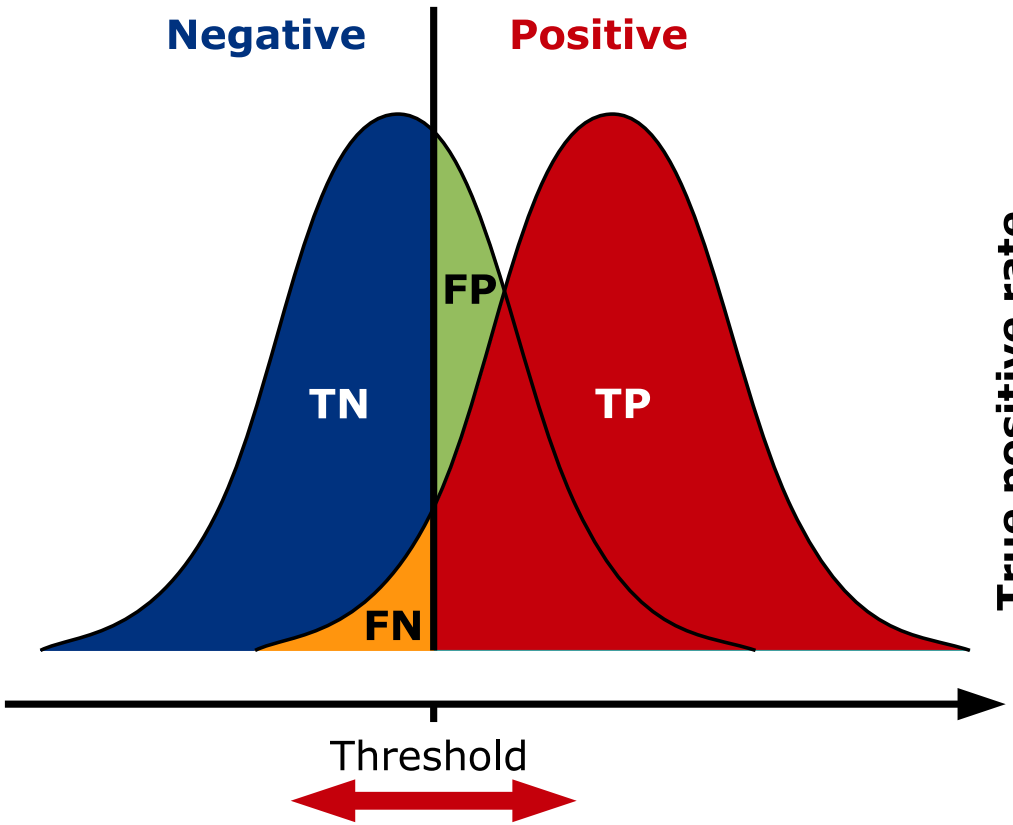
The recall is the number of true positives divided by the number of *actual* positives:

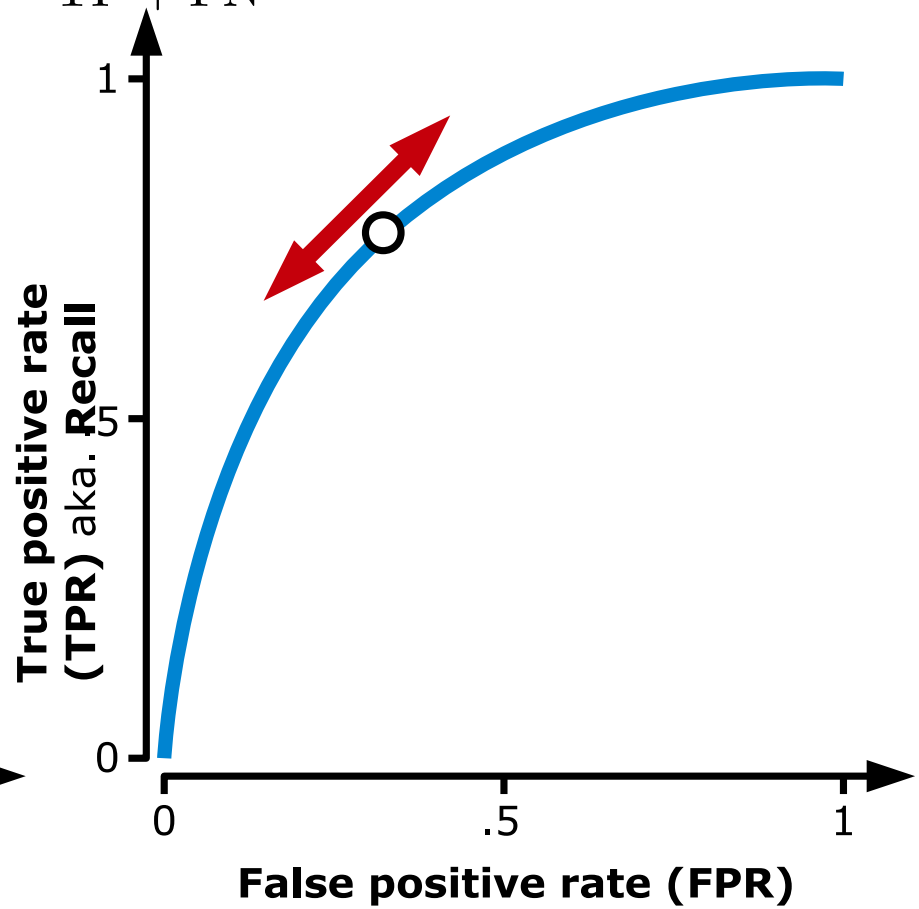
$$r_1 = \frac{18}{20} = \frac{9}{10}, \quad r_2 = \frac{14}{20} = \frac{7}{10}.$$

If the classifiers are supposed to detect credit-card fraud, we probably want a classifier which does not incorrectly *miss* many fraudulent transactions, i.e. it should have a low value of FN. In that case a high recall is desirable, and we should go with classifier 1.

Receiver operating characteristic

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$



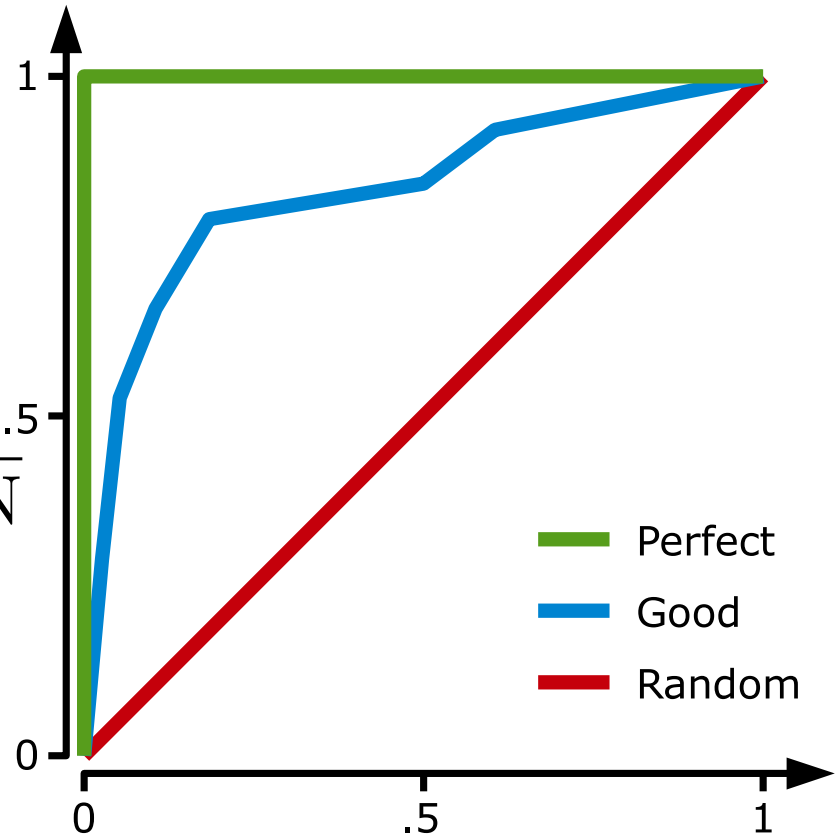


$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{TN} + \mathrm{FP}}$$

Receiver operating characteristic

True positive rate
aka. **Recall**

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$



$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{TN} + \mathrm{FP}}$$

Quiz 03 (please answer on Piazza): AUC (Spring 2017)

	3 gears ($x_5 = 3$)	4 gears ($x_5 = 4$)	5 gears ($x_5 = 5$)
Low mpg ($y = 0$)	13	2	2
High mpg ($y = 1$)	2	10	3

Table 1: Number of low mpg and high mpg cars (i.e. $y = 0$ and $y = 1$) according to the number of gears, i.e. $x_5 = 3$, $x_5 = 4$, or $x_5 = 5$.

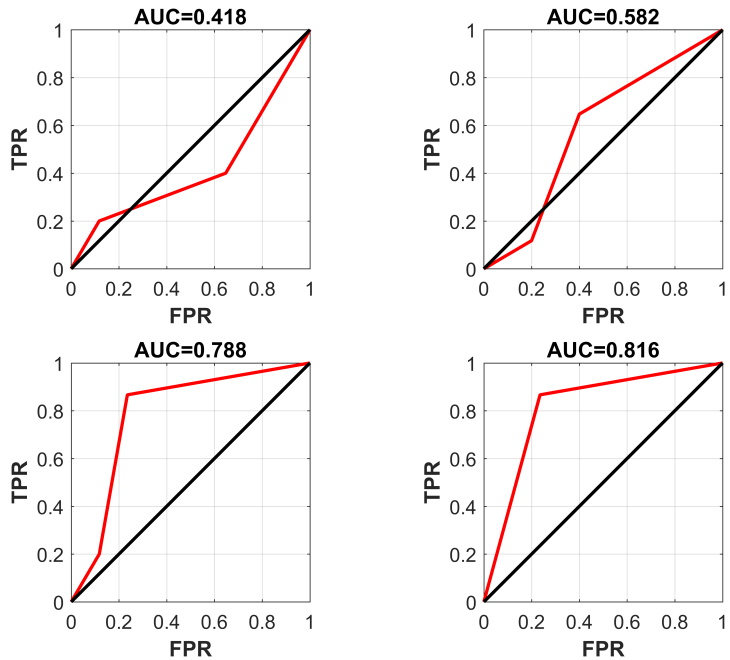


Figure 1: Four different receiver operator characteristic (ROC) curves and their area under curve (AUC) value.

A dataset representing cars contain an attribute x_5 corresponding to the number of gears. We wish to evaluate how well the number of gears predict low mpg, ($y = 0$, considered the negative class) from high mpg, ($y = 1$, considered the positive class) based on the data given in Table 1. For this purpose, we will evaluate the area under curve (AUC) of the receiver operator characteristic (ROC) using the feature x_5. Which one of the ROC curves given in Figure 1 corresponds to using x_5 to discriminate between low mpg ($y = 0$) and high mpg ($y = 1$)?

A. The curve having AUC=0.418

B. The curve having AUC=0.582

C. The curve having AUC=0.788

D. The curve having AUC=0.816

E. Don't know.

(Hint: Select a value e.g. $x_5 = 4$. We then predict cars with 4 or more gears as being in the positive class and otherwise negative. Compute the FPR and TPR using this prediction and use the (FPR, TPR) values to discriminate between the curves)

Solution:

The ROC curve can be calculated by lowering the threshold, as no cars have more than 5 forward gears a threshold above 5 will result in the point (0,0). Lowering the threshold we find at the value 5 that 2/17 of the low mpg cars (FPR) are at 5 and 3/15 of the high mpg cars (TPR) are at 5 corresponding to the point (2/17,3/15). When lowering to a threshold of 4 gears or more we are at the point (4/17,13/15) and at a threshold at 3 gears or more we have (1,1). Thus, this curve corresponds to the curve having AUC=0.788.

Resources