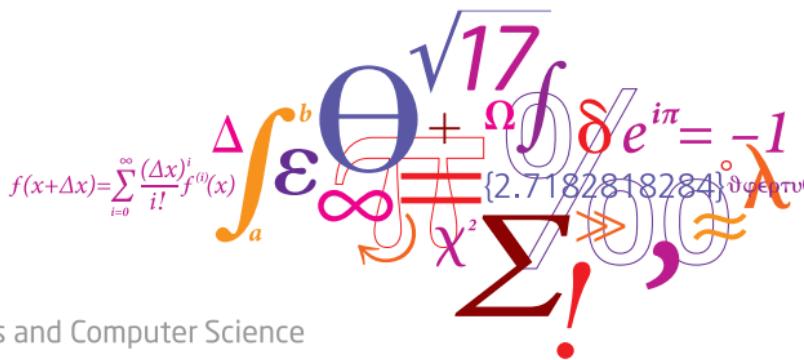


02450: Introduction to Machine Learning and Data Mining

AUC and ensemble methods

Jes Frellsen

DTU Compute, Technical University of Denmark (DTU)



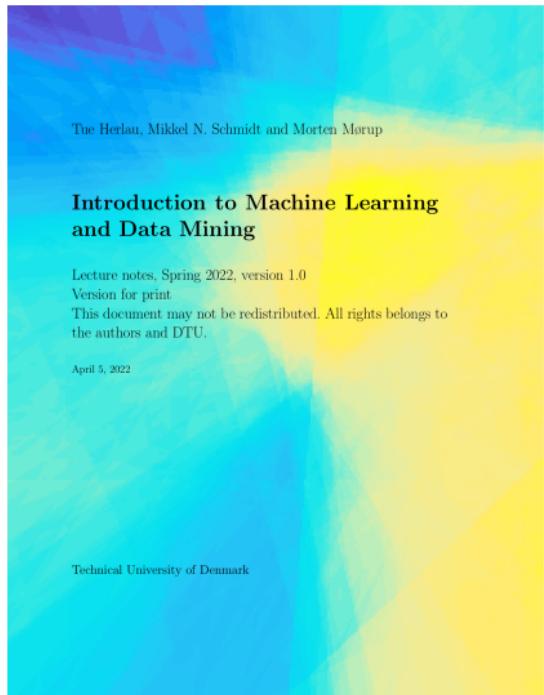
Today

Feedback Groups of the day:

Frederik Lindencrone Lundin, Andrés Martinez Banda, Jacob Øager Møller, Viktor Skovgaard Ørum, Johan Wohlfahrt Veje, Nicolas Triebold, Jacqueline Villalobos, Finnur Helgi Malmquist, Mads Rumle Nordstrøm, Hossein Soleimani, Simon Therkildsen, Jacob Østergaard Pedersen, Wei Kan Teo, Lucas Pedersen, Marcus Stürup, Erkin Najaflı, Spyridon Pikoulas, Jakub Maciej Wisniewski, Aleksandra Ludwiniak, Mario Medoni, Nina Kuna Peuker, Ask Espensønn Øren, Xiaopeng Mao, Martin Nguyen, Magnús Sigurðarson, Lukas Alexander Guldstøv Mittun, Julian Dominik Sempruch-Szachowicz, Hans Forum Møller, Naglis Naslenas, Athanasios Nichoritis, Sune Vang Nielsen, Søren Meyer Nielsen, Sifat E Noor, Dagfinnur Ari Normann, Julius Olander, Jens Helmer Olesen, Safiyye Nur Özer

Reading material:

Chapter 16, Chapter 17



Lecture Schedule

1 Introduction

31 January: C1

Data: Feature extraction, and visualization

2 Data, feature extraction and PCA

7 February: C2, C3

3 Measures of similarity, summary statistics and probabilities

14 February: C4, C5

4 Probability densities and data visualization

21 February: C6, C7

Supervised learning: Classification and regression

5 Decision trees and linear regression

28 February: C8, C9

6 Overfitting, cross-validation and Nearest Neighbor

7 March: C10, C12 (Project 1 due before 13:00)

7 Performance evaluation, Bayes, and Naive Bayes

14 March: C11, C13

8 Artificial Neural Networks and Bias/Variance

21 March: C14, C15

9 AUC and ensemble methods

28 March: C16, C17

Unsupervised learning: Clustering and density estimation

10 K-means and hierarchical clustering

11 April: C18

11 Mixture models and density estimation

18 April: C19, C20 (Project 2 due before 13:00)

12 Association mining

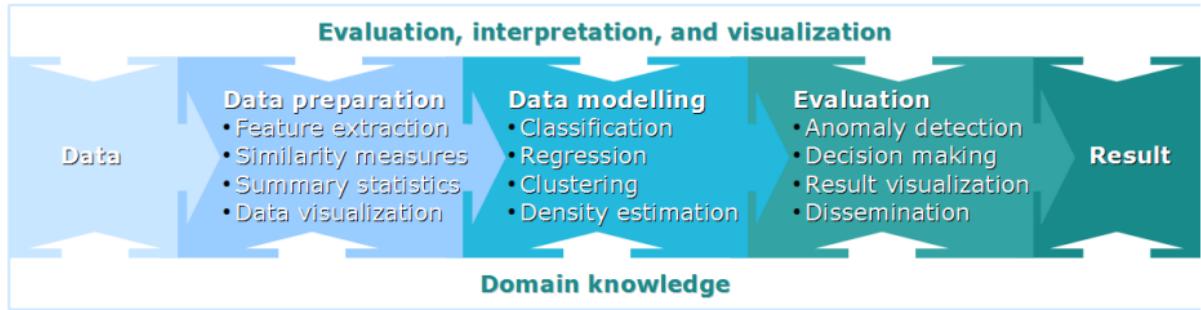
25 April: C21

Recap

13 Recap and discussion of the exam

2 May: C1-C21

Online 24/7 help: Discussion Forum/Piazza
Streaming & Videos: <https://panopto.dtu.dk/>
Online exercises: MS Teams

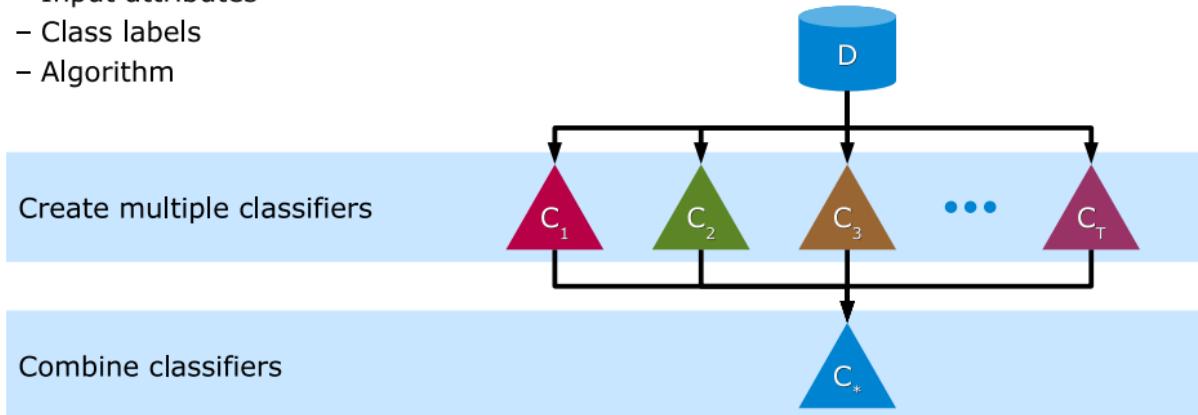


Learning Objectives

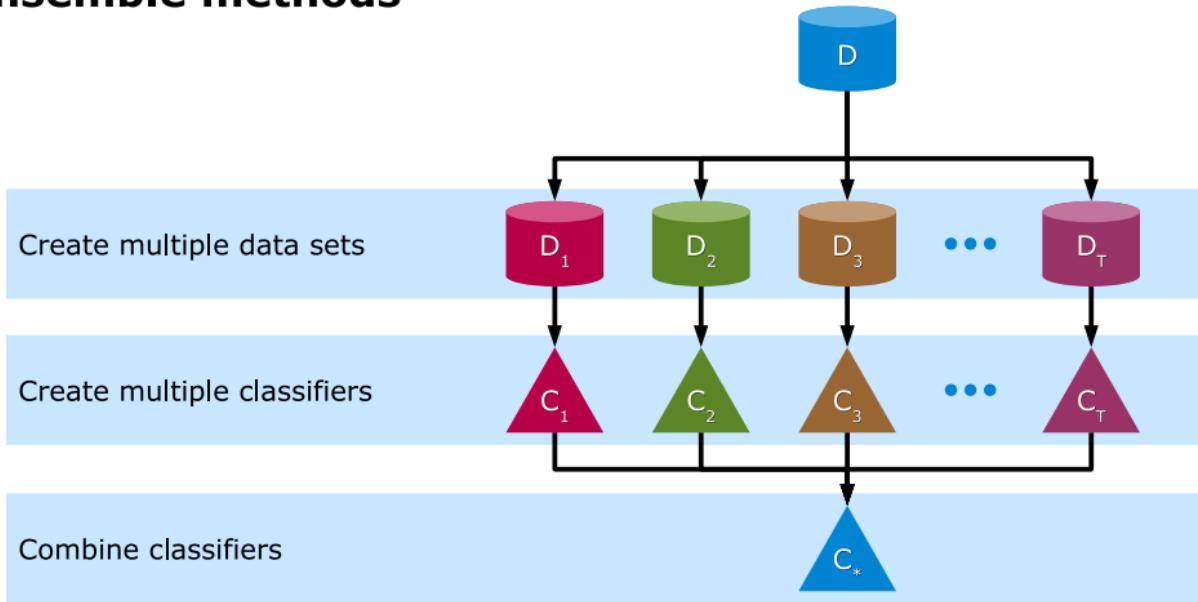
- Explain the principle behind boosting and bagging and apply it to improve classifiers
- Be able to address issues of class-imbalance and resampling
- Understand the definition of Precision, Recall, ROC, and AUC

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

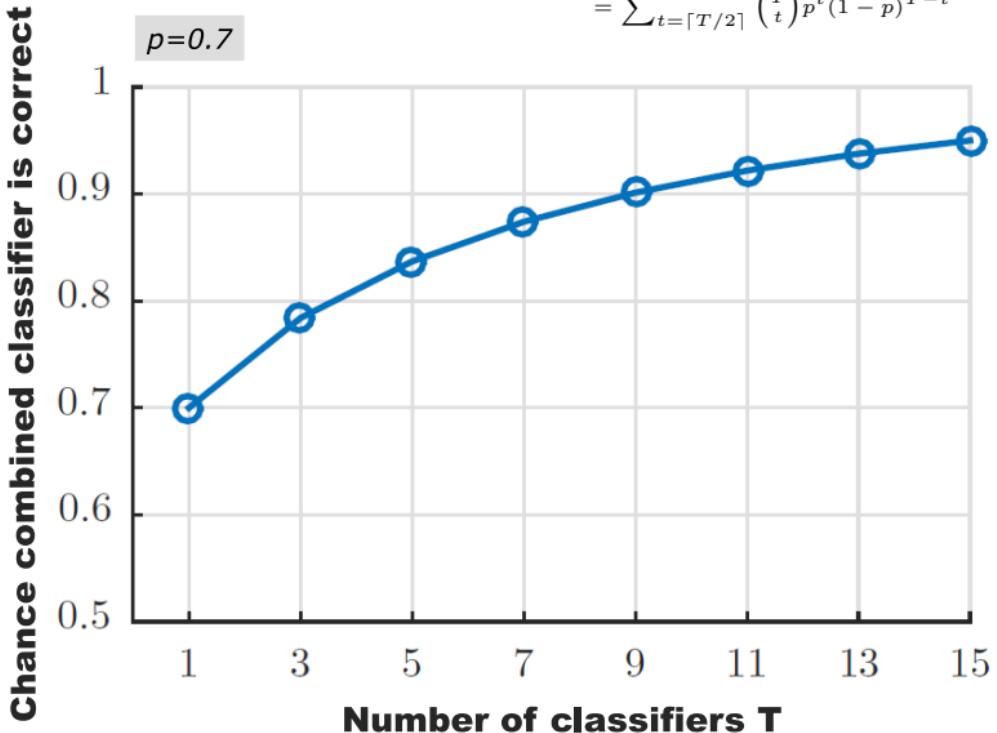


Why ensemble methods?

- 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 probability a classifier which use majority voting is correct is then given by:

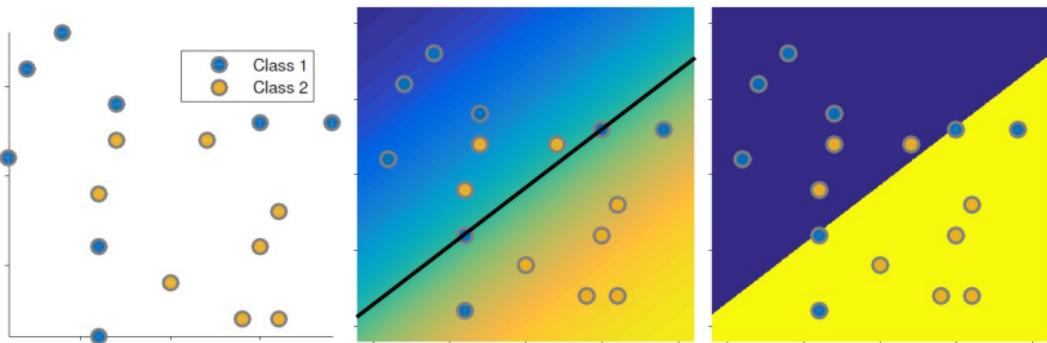
$$\begin{aligned} P(\text{Majority voting is correct}) &= \sum_{t=\lceil T/2 \rceil}^T P(\{t \text{ classifiers are correct}\}) \\ &= \sum_{t=\lceil T/2 \rceil}^T \binom{T}{t} p^t (1-p)^{T-t} \end{aligned}$$

$$\begin{aligned} P(\text{Majority voting is correct}) &= \sum_{t=\lceil T/2 \rceil}^T P(\{t \text{ classifiers are correct}\}) \\ &= \sum_{t=\lceil T/2 \rceil}^T \binom{T}{t} p^t (1-p)^{T-t} \end{aligned}$$



Data example

- Classification using logistic regression



Bagging

- New training data sets drawn randomly from pool with replacement

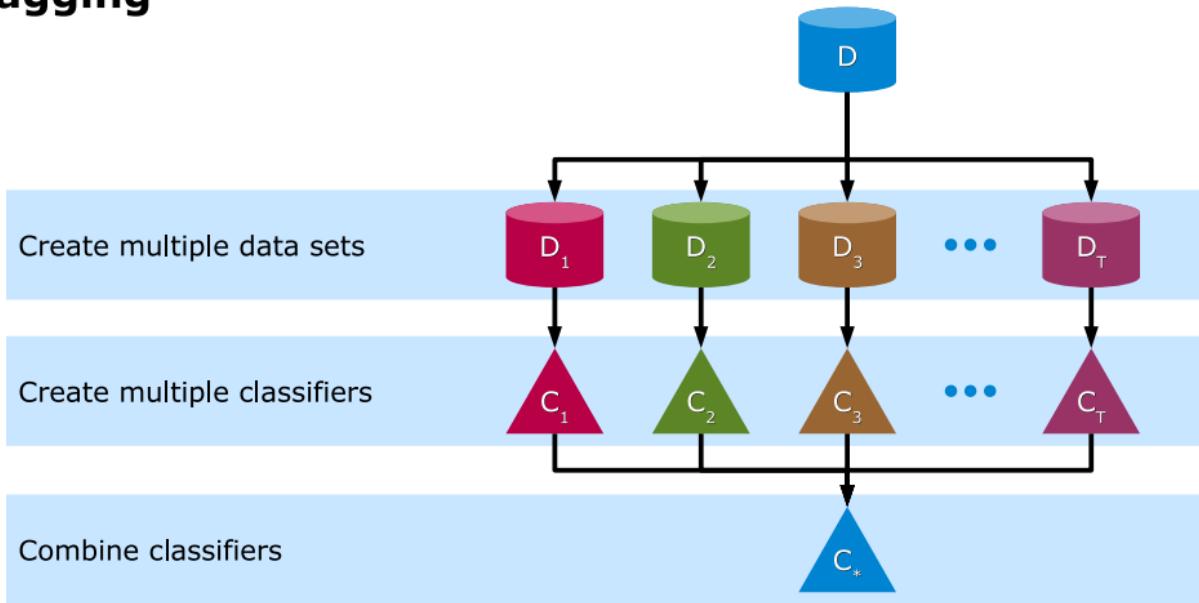
Pool of training data

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

New training data sets

3	5	4	3	9	7	9	5	1	1
5	8	2	6	2	3	8	3	5	1
1	7	4	1	10	6	10	8	8	7
⋮									
4	3	8	5	2	4	7	10	10	8

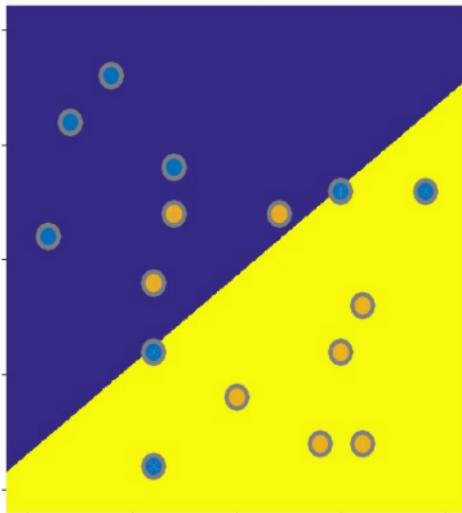
Bagging



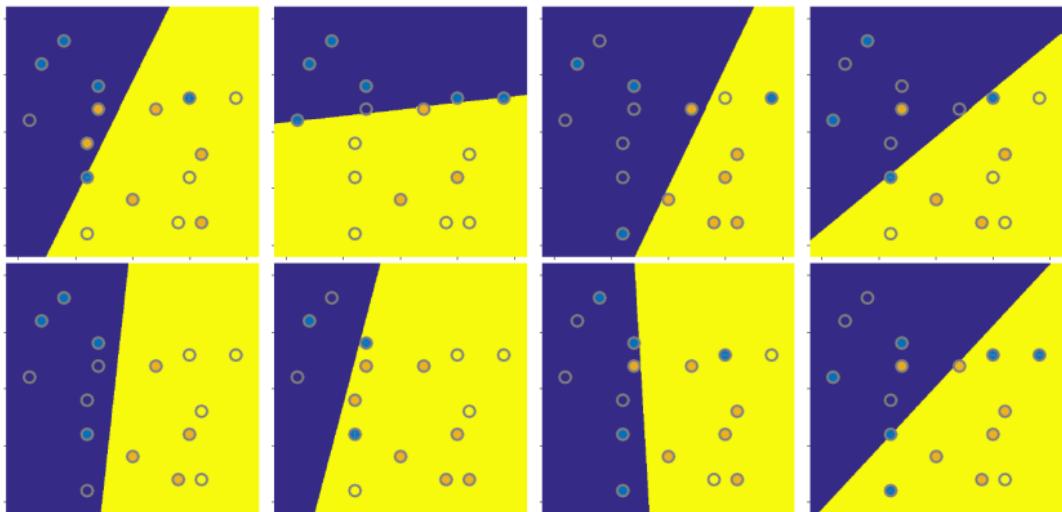
Bagging

- **Single classifier**

- Logistic regression
- Two features, (x, y)



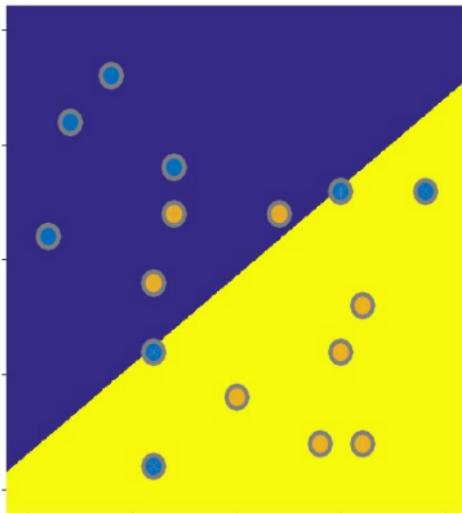
Bagging



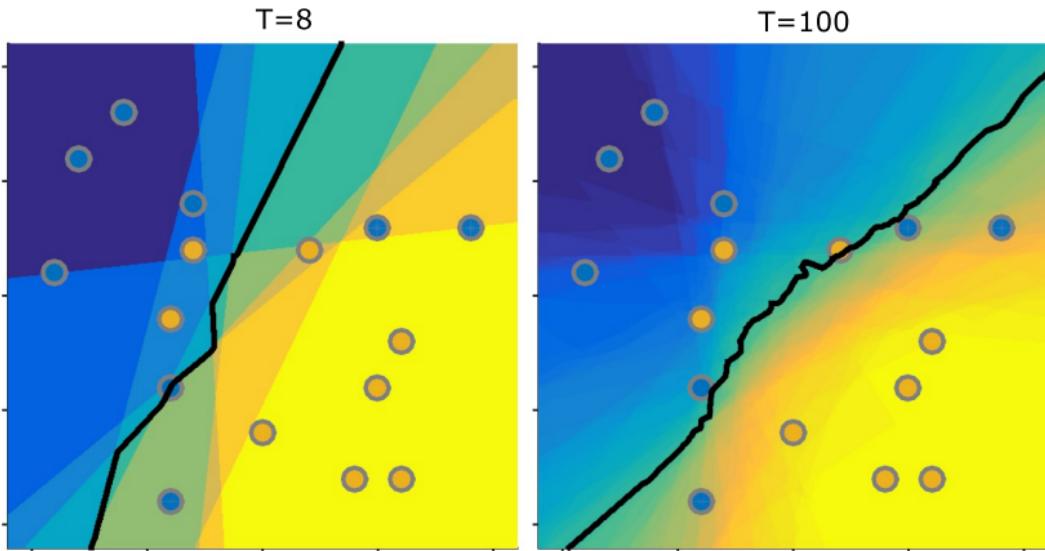
Notice, hollow dots are observations not included in bagging round

Bagging

- Single classifier



Bagging



Boosting

Pool of training data

1	2	3	4	5	6	7	8	9	10
.1	.1	.1	.1	.1	.1	.1	.1	.1	.1

Weights

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										
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
.1	.1	.1	.1	.1	.1	.1	.1	.1	.1

Weights

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
.1	.1	.1	.1	.1	.1	.1	.1	.1	.1

Weights

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✓
.07	.17	.07	.17	.07	.17	.07	.07	.07	.07

Update weights

New training data set

6	4	7	3	2	4	10	2	5	6
---	---	---	---	---	---	----	---	---	---

Train classifier



AdaBoost

Algorithm 6: AdaBoost algorithm

- 1: Initialize $w_i(1) = \frac{1}{N}$ for $i = 1, \dots, N$
- 2: **for** $t = 1, \dots, T$ **do**
- 3: Create \mathcal{D}_t by sampling (with replacement) from \mathcal{D} according to $\mathbf{w}(t)$
- 4: Let f_t be the classifier *trained* on \mathcal{D}_t
- 5: $\epsilon_t = \sum_{i=1}^N w_i(t) (1 - \delta_{f_t(\mathbf{x}_i), y_i})$ (*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):

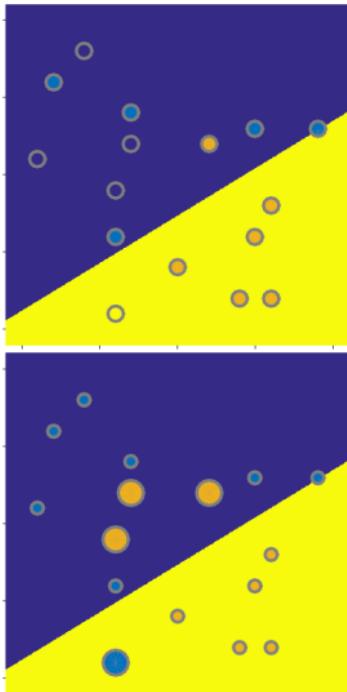
$$w_i(t+1) = \frac{\tilde{w}_i(t+1)}{\sum_{j=1}^N \tilde{w}_j(t+1)}, \quad \tilde{w}_i(t+1) = \begin{cases} w_i(t)e^{-\alpha_t} & \text{if } f_t(\mathbf{x}_i) = y_i \\ w_i(t)e^{\alpha_t} & \text{if } f_t(\mathbf{x}_i) \neq y_i. \end{cases}$$

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

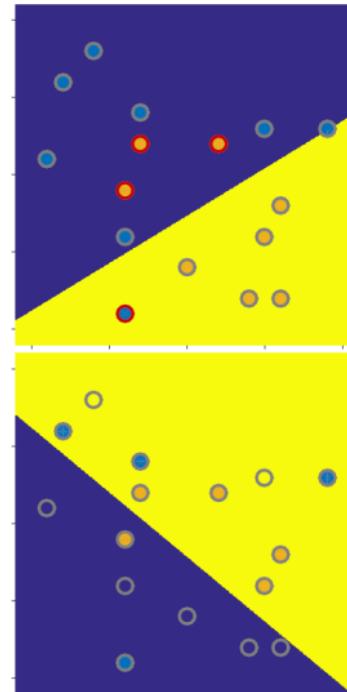
Boosting

A:

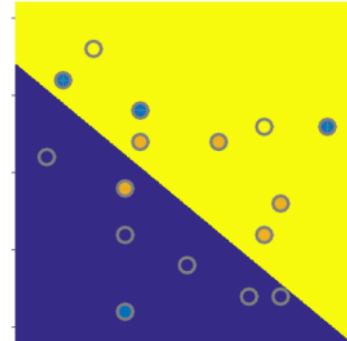
A dataset is sampled with replacement and a classifier trained.

**C:**

Weights are updated such that more emphasis is given to these mis-classified observations.

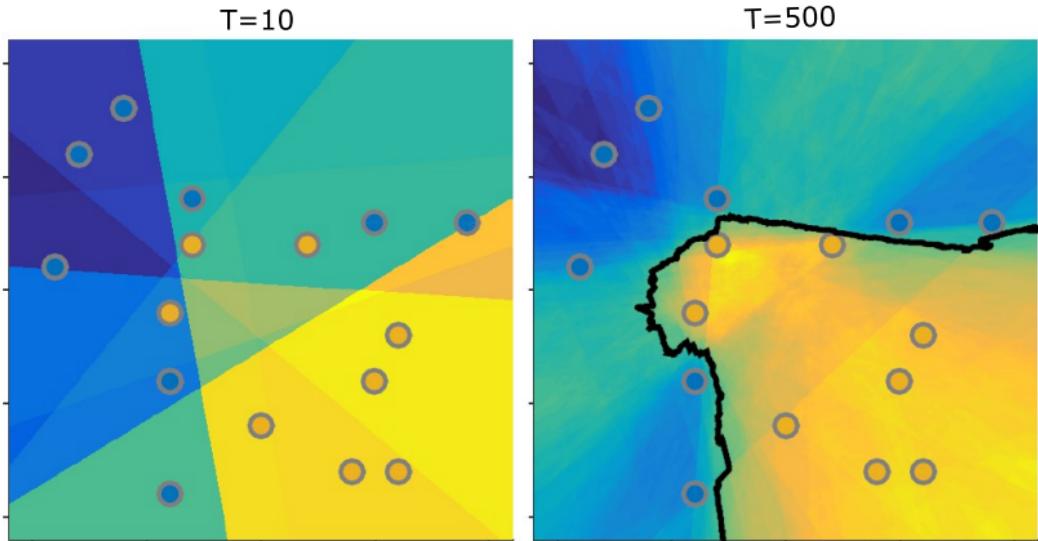
**B:**

Mis-classified observations are identified.



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

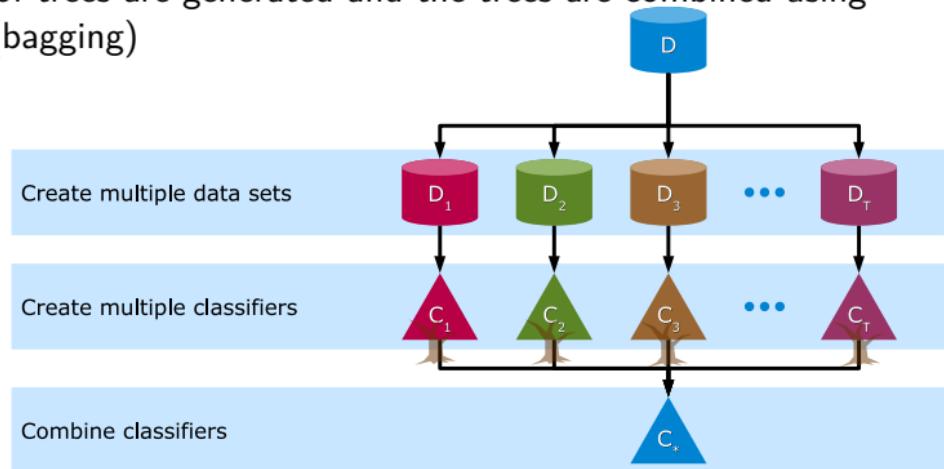


Bagging example: Random forest

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 1 (please answer on Piazza): Adaboost (Spring 2016)

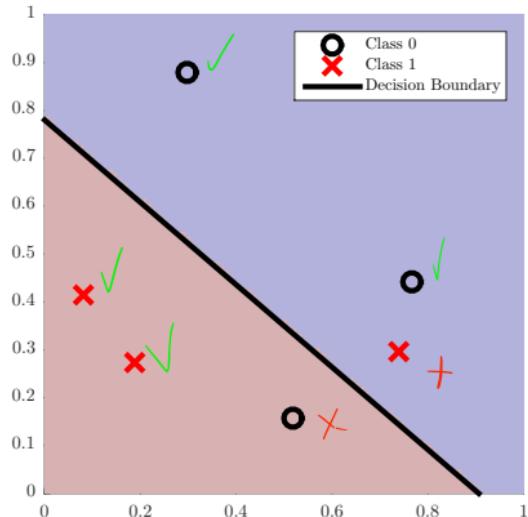


Figure 1: A binary classification problem and the decision boundary obtained by logistic regression. Observations left of the boundary are classified as belonging to the positive class 1 (red crosses) and observations right of the boundary to the negative class 0 (black circles)

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 \mathbf{w} ?

- A. $\mathbf{w} = [0.125 \ 0.250 \ 0.125 \ 0.125 \ 0.125 \ 0.250]$
- B. $\mathbf{w} = [0.026 \ 0.447 \ 0.026 \ 0.026 \ 0.026 \ 0.447]$
- C. $\mathbf{w} = [0.235 \ 0.029 \ 0.235 \ 0.235 \ 0.235 \ 0.029]$
- D. $\mathbf{w} = [0.1 \ 0.3 \ 0.1 \ 0.1 \ 0.1 \ 0.3]$
- E. Don't know.

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

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...**

Imbalanced training data	1	2	3	4	5	6	7	8	9	10
Oversampling	1	2	3	4	5	7	9	10	6	6
	6	6	8	8	8	8				
Undersampling	3	5	6	8						
Somewhere in between	3	5	4	3	9	6	6	8	8	8

Confusion matrix

		<i>Predicted</i>	
		<i>positive</i>	<i>negative</i>
<i>Actual</i>	<i>positive</i>	TP True Positive	FN False Negative
	<i>negative</i>	FP False Positive	TN True Negative

Precision and recall

- **Precision**

- Fraction of true positive among objects predicted to be positive

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

- **Recall**

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

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

		<i>Predicted</i>	
		<i>positive</i>	<i>negative</i>
<i>Actual</i>	<i>positive</i>	TP	FN
	True Positive	False Negative	
	<i>negative</i>	FP	TN
	False Positive	True Negative	



Quiz 2 (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

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

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

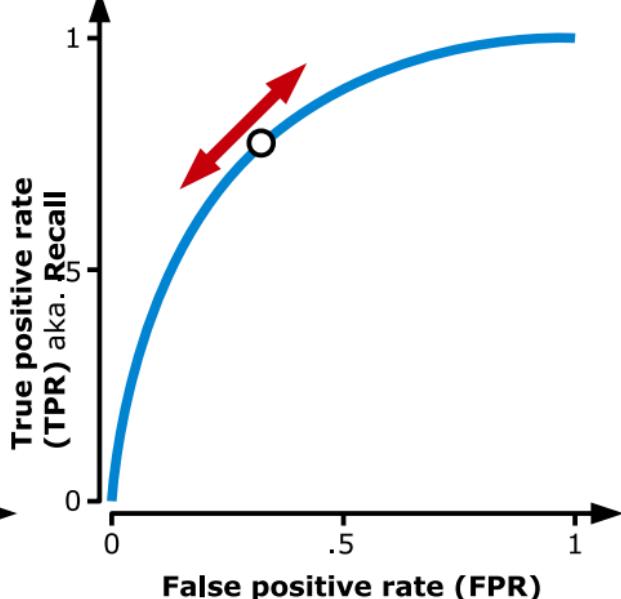
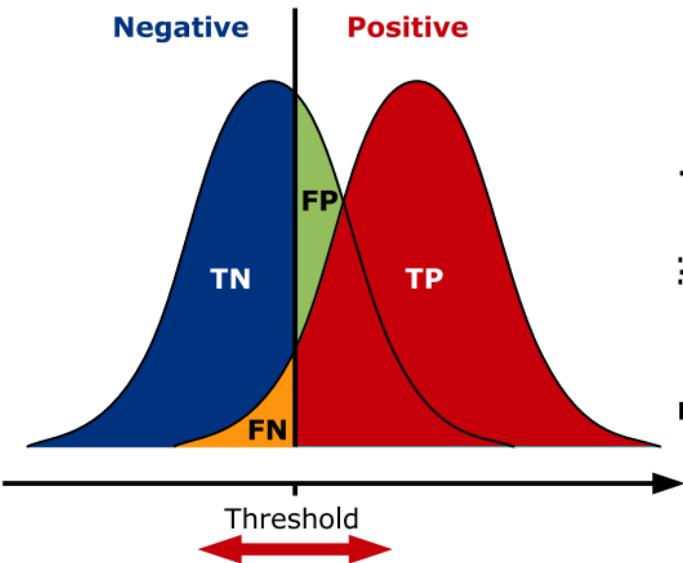
• Recall

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

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

Receiver operating characteristic

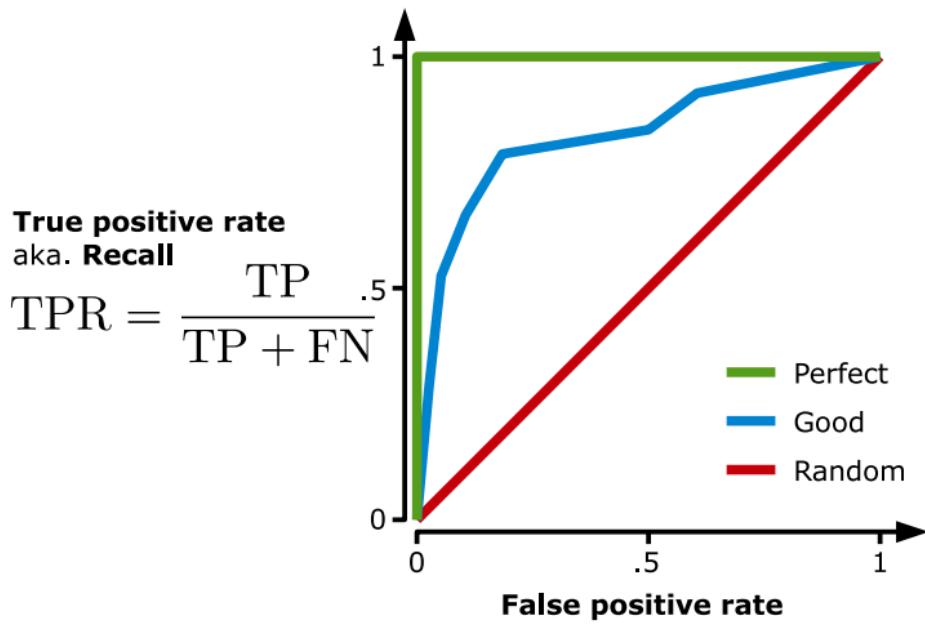
$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$



False positive rate (FPR)

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

Receiver operating characteristic



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

Quiz 3 (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 $x_5 = 5$.

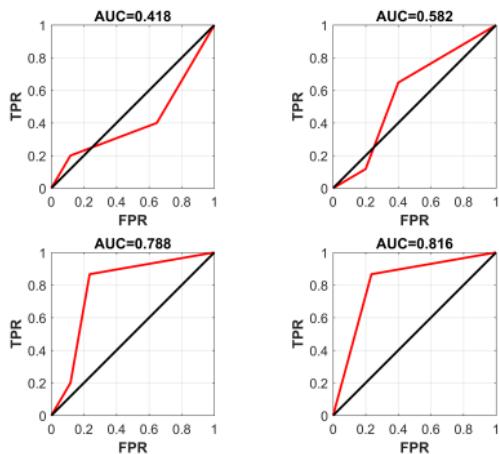


Figure 1: Four different receiver operator characteristic (ROC) curves and their area under curve (AUC) value.

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)

Resources

<https://www.youtube.com> Video tutorial on ROC curve and AUC

(<https://www.youtube.com/watch?v=0Al6eAyP-yo>)

<https://towardsdatascience.com> More in-depth discussion of the Random Forrest algorithm and parameter choices

(<https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd>)

<https://www.datacamp.com> Practical use of the random forest algorithm in python

(<https://www.datacamp.com/community/tutorials/random-forests-classifier-python>)

<https://citeseerx.ist.psu.edu> Justification for the AdaBoost algorithm (technical) (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.51.9525>)