02450: Introduction to Machine Learning and Data Mining

AUC and ensemble methods

DTU Compute
Department of Applied Mathematics and Computer Science

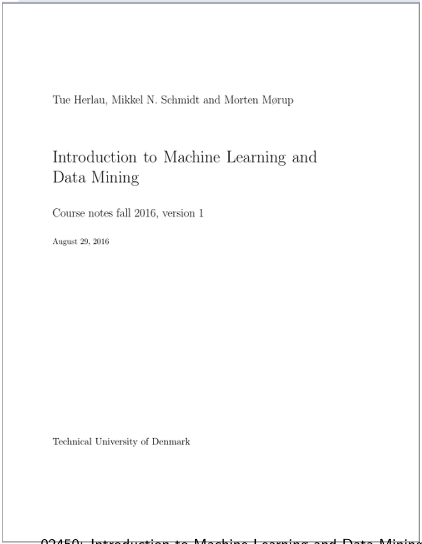
Reading Material

Reading material:

C14, C15

Feedback Groups of the day:

- Tobias Brasch, Sigbjørn Hokland
- Philip Bendixen Larsen, Hans-Christian Thorsen-Meyer
- Jarrett Taylor, Jesper Nissen
- Caroline Harder Hovgesen, Patrick Janowski
- Georg Thomassen, John Johannesen
- Niels Beuschau, Søren Norge Andreassen
- Matthias Scharl, Henry Bliemel, Daniel Santaella
- Mette Kyhn Larsen, Simon Mørup Carlsson


Tue Herlau, Mikkel N. Schmidt and Morten Mørup

Introduction to Machine Learning and Data Mining

Course notes fall 2016, version 1

August 29, 2016

Technical University of Denmark

Lecture Schedule

1. Introduction
   30 August: C1

Data: Feature extraction, and visualization

2. Data and feature extraction
   6 September: C2, C3
3. Measures of similarity and summary statistics
   13 September: C4
4. Data Visualization and probability
   20 September: C5, C6

Supervised learning: Classification and regression

5. Decision trees and linear regression
   27 September: C7, C8 **(Project 1 due before 13:00)**
6. Overfitting and performance evaluation
   4 October: C9
7. Nearest Neighbor, Bayes and Naive Bayes
   11 October: C10, C11
8. Artificial Neural Networks and Bias/Variance
   25 October: C12, C13
9. **AUC and ensemble methods**
   1 November: C14, C15

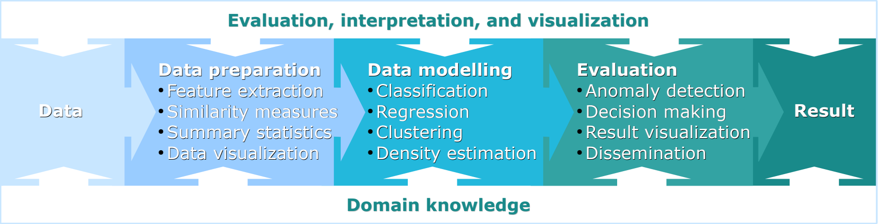
Unsupervised learning: Clustering and density estimation

10. K-means and hierarchical clustering
    8 November: C16 **(Project 2 due before 13:00)**
11. Mixture models and density estimation
    15 November: C17, C18
12. Association mining
    22 November: C19

Recap

13. Recap and discussion of the exam
    29 November: C1-C19 **(Project 3 due before 13:00)**

Data modeling framework

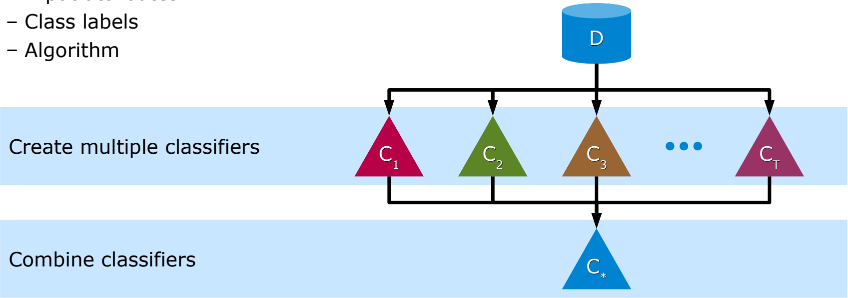


After today you should be able to:
Explain the principle behind boosting and bagging and apply it to improve classifiers
Be able to address issues of class-imbalances by resampling
Understand the definition of Precision, Recall, ROC and AUC

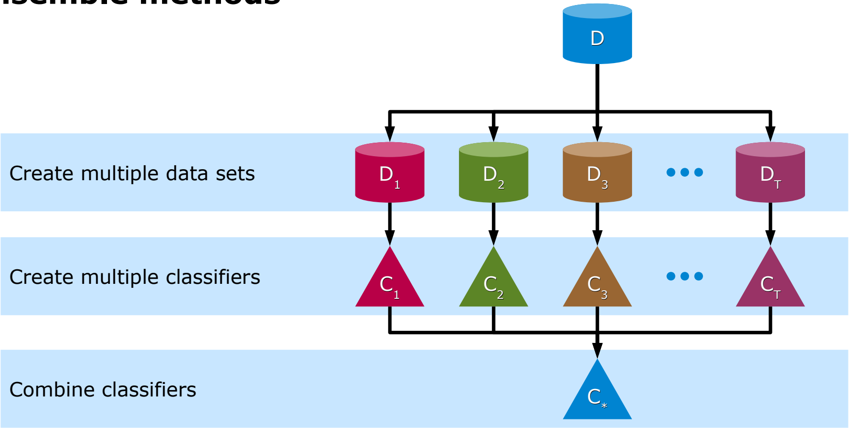
Report 2 due at next lecture before 13:00. Please upload the report as a single PDF file to campusnet. You do not have to hand in a paper copy. Remember to answer all questions asked in the report.

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

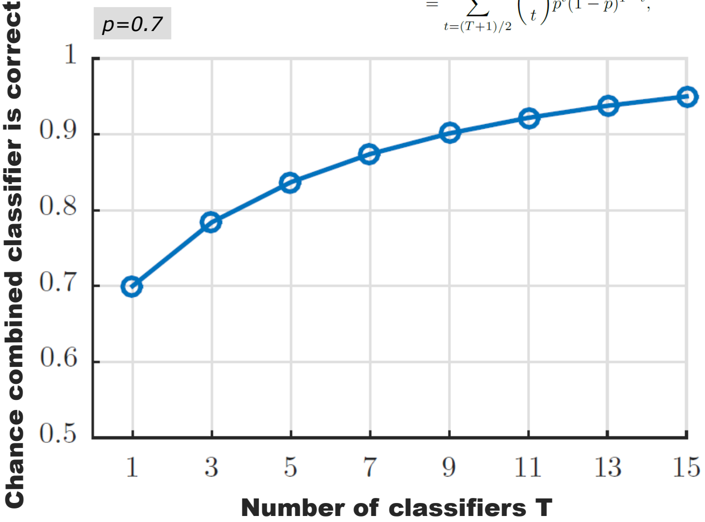


Why ensemble methods?

- 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:

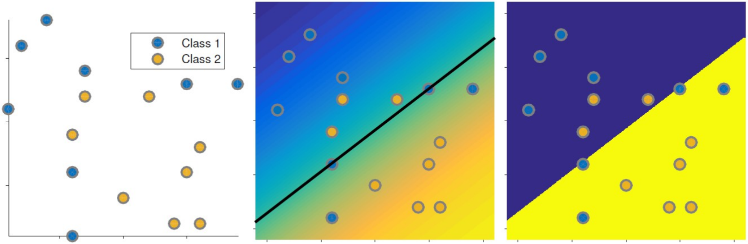
$$
\begin{aligned}
P(\text{Majority voting is correct}) &= \sum_{t=(T+1)/2}^{T} \{t \text{ of the classifiers are correct}\} \\
&= \sum_{t=(T+1)/2}^{T} \binom{T}{t} p^t (1-p)^{T-t},
\end{aligned}
$$

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



Data example

- Classification using logistic regression

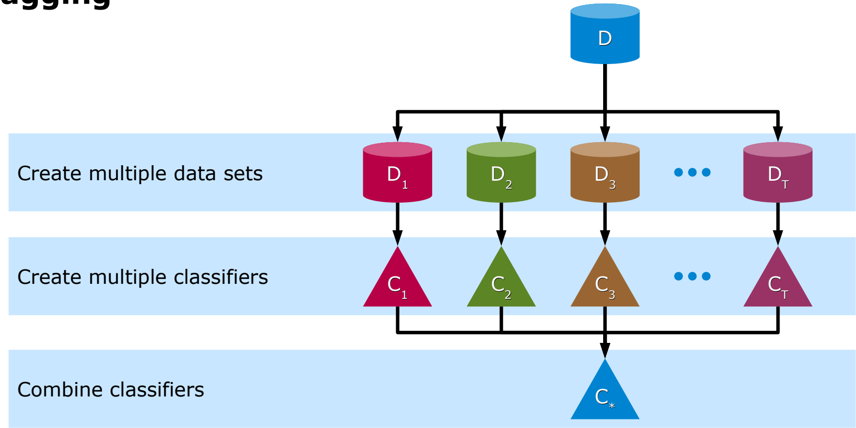


Bagging

- New training data sets drawn randomly from pool with replacement

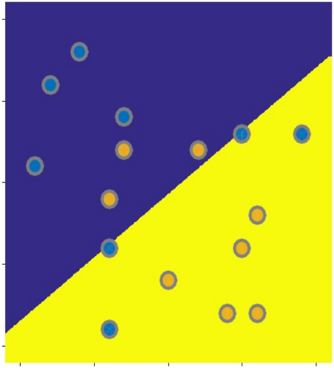
Pool of training data	1	2	3	4	5	6	7	8	9	10
New training data sets	3	5	4	3	9	7	9	5	1	1
	5	8	2	6	2	3	8	3	5	1
	1	7	4	1	10	6	10	8	8	7
	⋮									
	4	3	8	5	2	4	7	10	10	8

Bagging

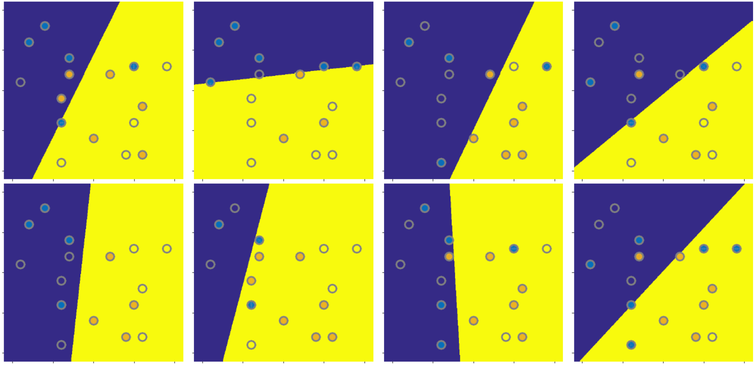

D
Create multiple data sets
D1
D2
D3
DT
Create multiple classifiers
C1
C2
C3
CT
Combine classifiers
C*

Bagging

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



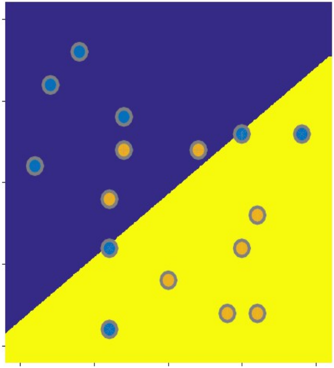
Bagging



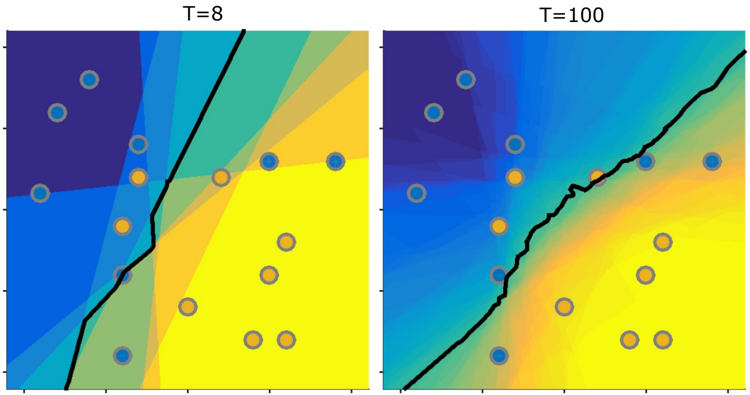
Notice, hollow dots are observations not included in bagging round

Bagging

- Single classifier



Bagging

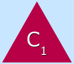


Boosting

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						C_1				
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						C_1				
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						C_1				
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						C_2				

⋮

AdaBoost

Algorithm 6: AdaBoost algorithm

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

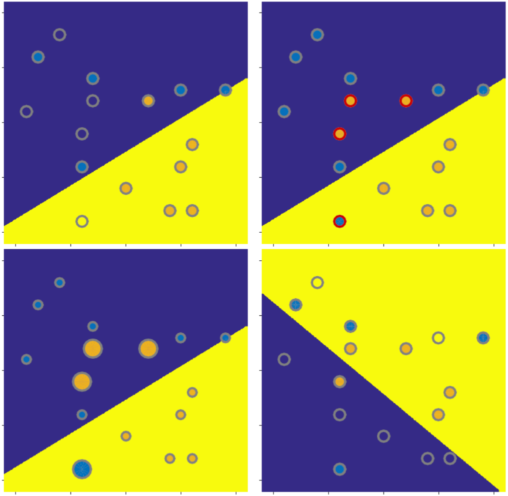
$$w_i(t+1) = \frac{\tilde{w}_i(t+1)}{\sum_{j=1}^{N} \tilde{w}_j(t+1)}, \quad \tilde{w}_j(t+1) = \begin{cases} w_j(t) e^{-\alpha_t} & \text{if } f_t(\boldsymbol{x}_i) = y_i \\ w_j(t) e^{\alpha_t} & \text{if } f_t(\boldsymbol{x}_i) \neq y_i. \end{cases}$$

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

Boosting

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

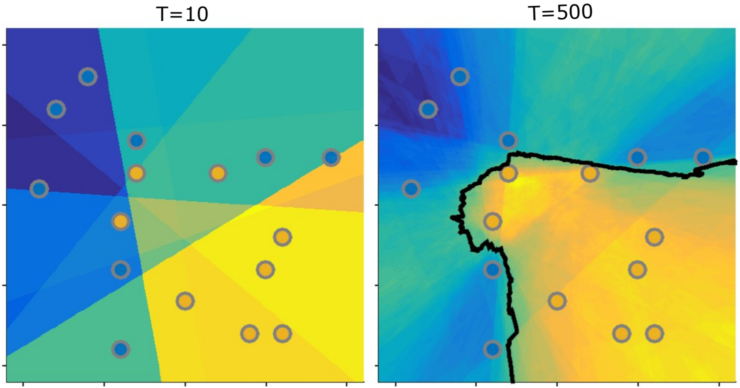
B:
Mis-classified observations are identified.



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



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

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

Precision and recall

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

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

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

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

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

Group exercise

- You consider two different classifiers, on a test set with 20 positive objects
  - **Classifier 1** detects 54 positives of which 18 are actually positive
  - **Classifier 2** detects 16 positives of which 14 are actually positive
- Compute the **precision** and **recall** for the two classifiers
- Which classifier (if any) is the best?
- Which would you use if the objective is to detect credit card fraud
  *(consider what is most costly – **missing** or **falsely detecting** a positive)*

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

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

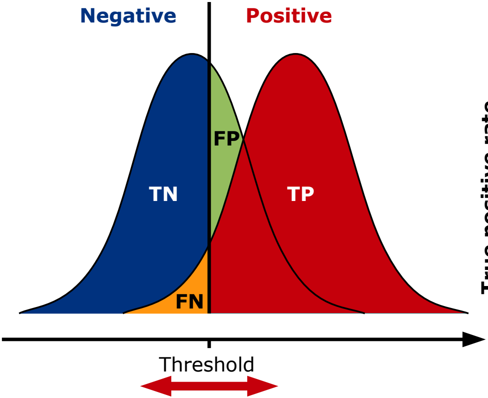
- **Recall**
  - Fraction of objects predicted to be positive among all positive objects

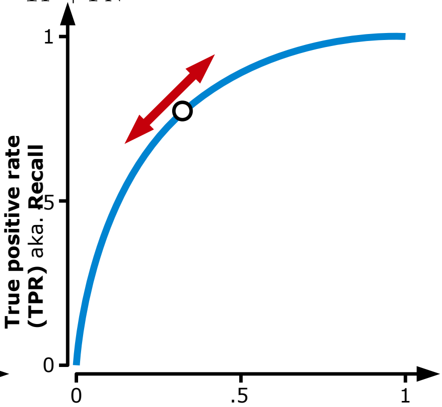
$$r = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

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

Receiver operating characteristic

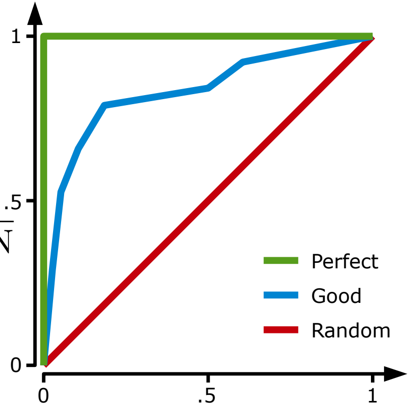
$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$





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

Receiver operating characteristic



True positive rate
aka. **Recall**

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

False positive rate

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