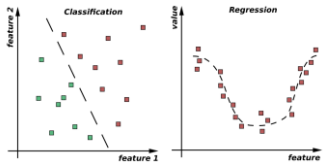
Signal processing and identification-AI module, Lecture 3

Classification and regression

Ziemowit Dworakowski
AGH University of Science and Technology,
Department of Robotics and Mechatronics

1

Classification and regression



Classification –
assigning a **class** label to the object

(Binary problem – object either belongs or does not belong to particular class)

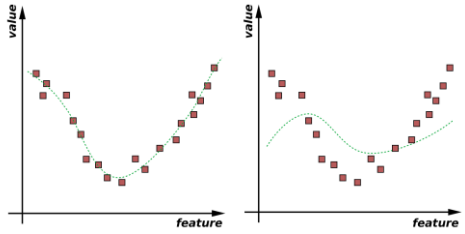
Regression –
assigning a **value** to the object

(A continuous problem – values can be floating point numbers)

- Both are usually multidimensional
- Both are based on **features**
- Both are very similar (in terms of issues and methods used)

2

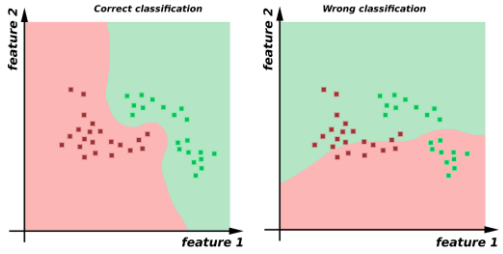
Classification and regression



Training of a regressor: Minimization of a **regression error** over selected set of **training data**

3

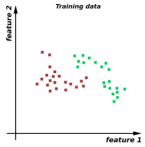
Classification and regression



Training of a classifier: Minimization of a **classification error** over selected set of **training data**

4

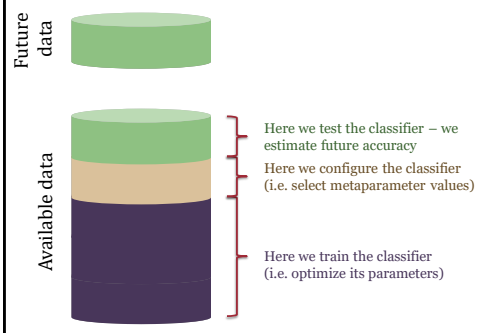
Training of a classifier in general



Training data

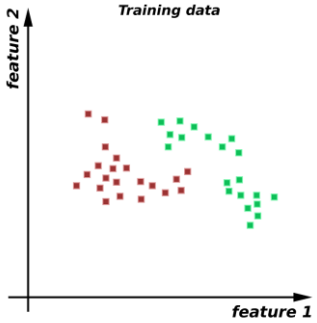
5

Testowanie metod



6

Classification and regression: clusters of data



Usually data are grouped in **clusters**

It is an advantage
(e.g. easy reduction of data amount, easier interpretation of data)

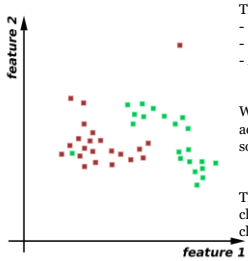
It can also be a disadvantage
(e.g. more difficult training, 'statistical' methods are more tricky to use)

7

Classification and regression: outliers

Single observations that are significantly different than rest of our data are called **outliers**

They are caused usually by:
- Measurement errors
- Labeling errors
- Abnormal situations in data acquisition

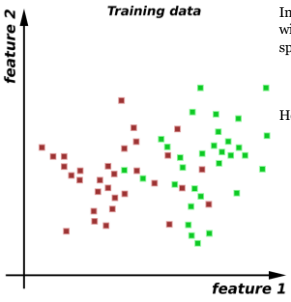


We usually filter them out during data acquisition and initial processing. However, sometimes it can be a bit tricky

There are ways to deal with outliers at the classification stage as well – using particular classifiers

8

Overlapping of classes



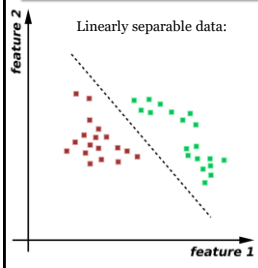
In many practical situations classess overlap with each other -there exist areas in feature space occupied by more than one class

How to deal with this problem?

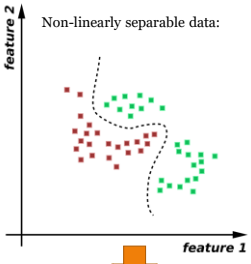
9

Linear and non-linear data separability

Linearly separable data:



Non-linearly separable data:



Require more advanced classifiers (usually with more parameters = more degrees of freedom)

10

Overfitting

Overfitting means that classifier or regressor obtains very high accuracy on training dataset but much lower on a testing dataset

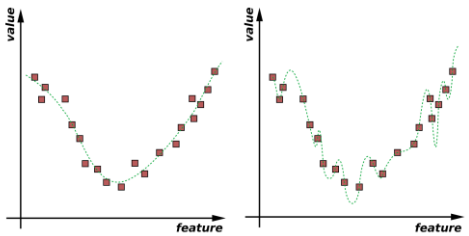
In other words:

System „memorizes" data and cannot **generalize**

11

Overfitting

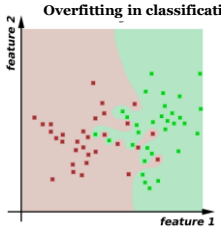
Overfitting in regression:



12

Overfitting

Overfitting in classification:



How to avoid overfitting (How to **generalize** efficiently) ?

- A subset of data only for detecting the overfitting
- Enough training data with respect to adjustable parameters of the classifier
- Early stopping of training, regularization

13

Curse of dimensionality

Level of difficulty rises exponentially with rise of problem dimensions (e.g. parameters to optimize, features to classify)

(Exponentially more data is required to fill the space reasonably well)

How to deal with the Curse of Dimensionality?

- A trivial solution: gather more data

- Pick features carefuly (Only the most important)

- (in the last resort) Generate new data based on training examples

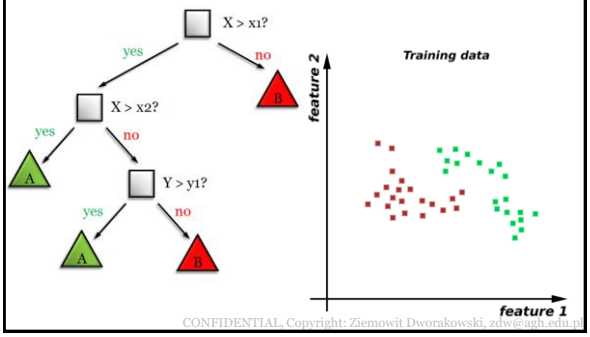
14

Basic classification algorithms

15

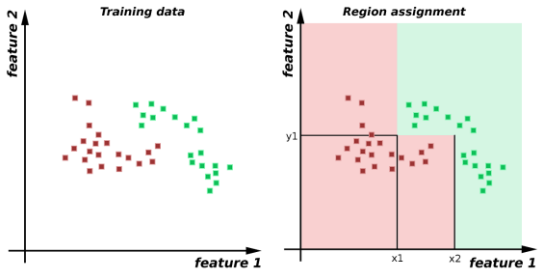
Decision tree

X and Y are features, A and B are classes



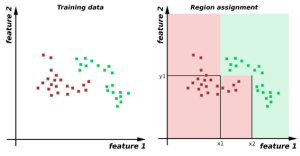
16

Decision tree



17

Decision tree

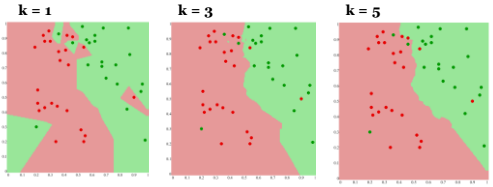


+ Easy, intuitive, fast
+ Easy to configure for simple 2D or 3D problems

- Very hard to manualy configure in high dimensional spaces
- Autonomous configuration algorithms are complex
- Sensitive to outliers and overfitting

18

k Nearest Neighbors, kNN



In order to label a new sample:

1. **Find its k (odd) closest neighbors**
2. **Pick a most popular label among them**
3. **Assign a sample with a picked label**

19

k Nearest Neighbors, kNN

+ Intuitive
+ Easy to implement
+ Robust to outliers (if k > 3)
+ Robust to overfitting

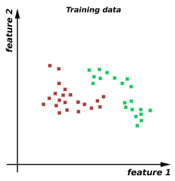
- Very sensitive to dimensionality, does not scale well
- Computationally demanding, slow, requires lots of memory

20

Clustering-based classification

We know that kNN works OK but cannot handle lots of data...,

We know that data are often in clusters ...



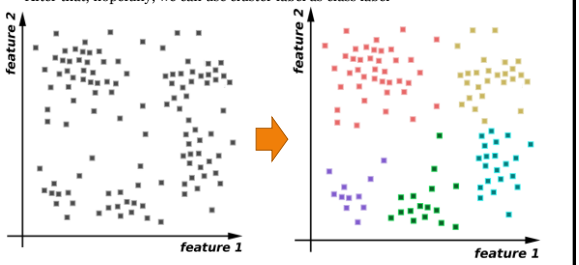
Why not use each cluster as a single data point?

21

Clustering-based classification

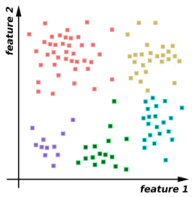
Clustering requires assigning each sample with a **cluster label** (instead of class label) – using an **unsupervised** algorithm

After that, hopefully, we can use cluster label as class label



22

Clustering-based classification

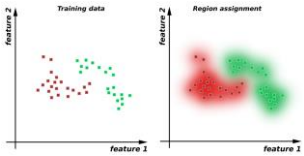


+ Saves memory
+ Efficient if data are organized in clusters
+ Intuitive
+ Robust to outliers
+ Robust to overfitting

- Hard optimization
- Detection of cluster numbers is challenging
- Works poorly for overlapping classes
- Does not allow for a complex classification marigin

23

Statistical – based classifiers

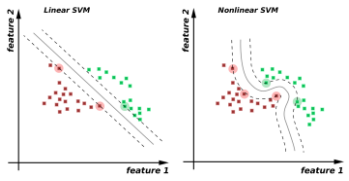


+ Intuitive
+ Robust to outliers
+ Robust to overfitting
+ Provide measure of probability of classification

- Sensitive to dimensionality
- Computationally and (sometimes) memory demanding – even more than kNN!

24

Support Vector Machine (SVM)

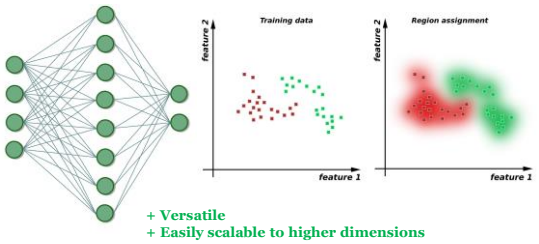


+ (Very!) fast learning
+ Reasonable selection of classification marigin
+ Scales well to higly dimensional feature spaces

- Requires experience in configuration
- Ignores data density

25

Neural networks



+ Versatile
+ Easily scalable to higher dimensions
+ Good measure of classification certainty
+ Takes data density into account

- Requires experience in configuration
- Prone to overfitting
- non-deterministic

26

Classifiers and regressors comparison

Classifier	Required memory	Difficulty of training	Operational speed	Classification and regression?	Sensitivity to dimensionality	Configuration	Overfitting	Overall efficiency
Drzewo dec.								
kNN								
Statystyczne								
Klasteryzacja								
SVM								
ANNs								
...								
...								

27

Working with data

28

Data amount reduction

Do we want to have many samples?

Do we want to have many dimensions?

29

Samples number reduction

In general *„You can't have TOO MANY data"*...
(theoretically speaking...)

However, due to finite memory and computational resources sometimes reduction of number of data samples is worthwile. We can do it by random drawing, clusterization or calculation of probability density of samples in feature space

30

Dimensionality reduction

„***Curse of dimensionality***" provides a soft constraint on the number of dimensions (features). More features require much more samples.

Even if we have vast amount of data, it is usually good to reduce number of samples as much as possible as it allows usage of more (and easier) algorithms, higher efficiency, shorter computation etc.

31

Dimensionality reduction

General rules:

- **Features should be significant**
(Each should contribute to the result. If discarding one of the features does not diminish the result – it was not necessary)

- **Features should be uncorrelated**

- **The number of features should be as small as possible**

32

General rules

- Training, validation and testing datasets must not overlap (so we could detect overfitting both in training and in metaparameters' choice)

- If we want to artificially generate data for training, we always do it **AFTER** division of data into training, validation and testing subsets – so new data generated from one particular sample would always fall into the same subset

- Non-deterministic methods should be tested multiple times, using statistics

- We assess general accuracy, but also accuracy with respect to particular classes

33

Signal processing and identification-AI module, Lecture 3

Classification and regression

1) **Basic vocabulary: clusters of data, linear separability, outliers, overfitting, curse of dimensionality etc.**
2) **General procedure of training a classifier**
3) **Examples of classification algorithms**
4) **How to test algorithms**
5) **How to work with data**

34