

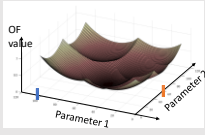
Mechatronic Engineering program

Basics of AI and Deep Learning:
3: Learning from data

Ziemowit Dworakowski
AGH University of Krakow

1

Summary from before SD



OF value

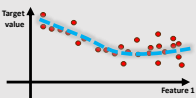
Parameter 1

Parameter 2

So far, we know how to look for coordinates of the objective functions' minima

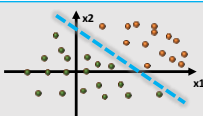
We know what is regression and how to design linear and nonlinear models for it

We know what the classification is and we can roughly do linear classification



Target value

Feature 1



x2

x1

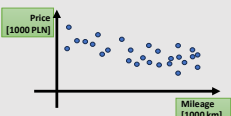
Today we will **generalize** a bit, look at these problems from **data perspective** and we'll also learn some **new concepts and tools** that will help along the way

2

Parameter space vs. Feature space SD

If we **measure** a value (or are given it) – it is a **feature**. If we have control over it and try to set it in the „best spot” – it is a **parameter**

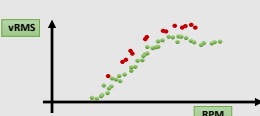
Regression:
„Predict a car price given mileage”



Price [10000 PLN]

Mileage [10000 km]

Classification:
„Predict wind turbine state given RPM and Power”



vRMS

RPM

3

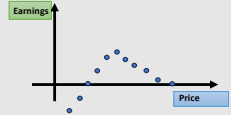
Parameter space vs. Feature space

If we **measure** a value (or are given it) – it is a **feature**. If we have control over it and try to set it in the „best spot“ – it is a **parameter**


The same value can be parameter or feature, depending on a context

Optimization is done in **parameter space**, Regression and classification in **feature space**

Optimization:
Find the best price for the car to maximize earnings:

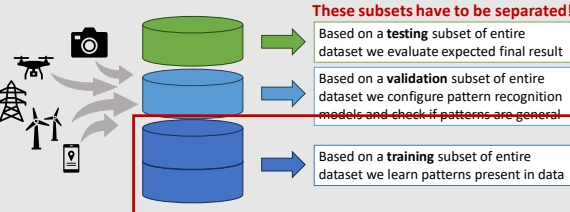


Optimization:
„Find the best amount of coffe and milk for the perfect latte“



4

Data organization

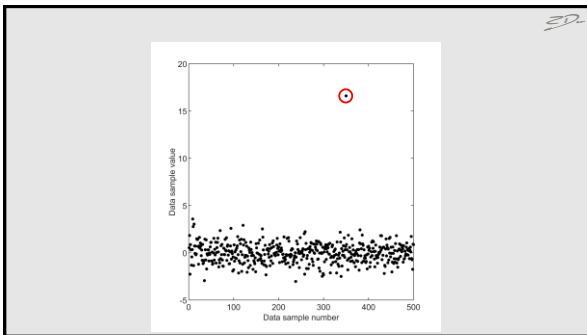


These subsets have to be separated!

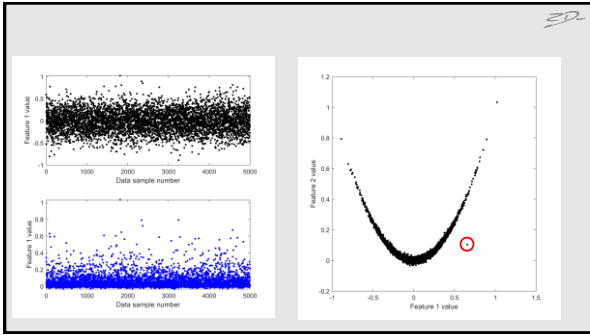
- Based on a **testing** subset of entire dataset we evaluate expected final result
- Based on a **validation** subset of entire dataset we configure pattern recognition ~~models and check if patterns are general~~
- Based on a **training** subset of entire dataset we learn patterns present in data

Today we Focus just on learning patterns. Further configuration and verification if we're right we leave for later...

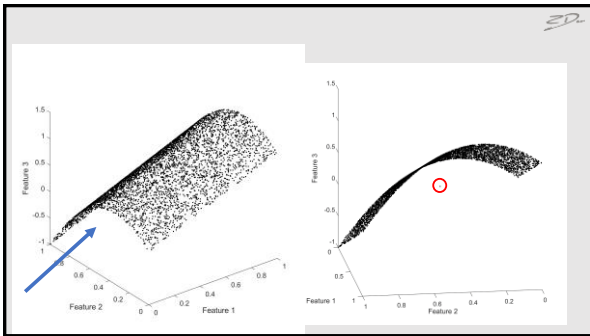
5



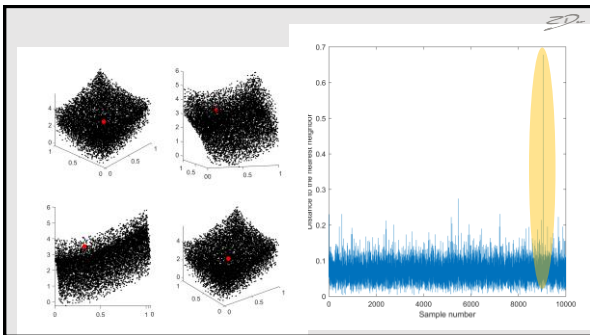
6



7



8



9

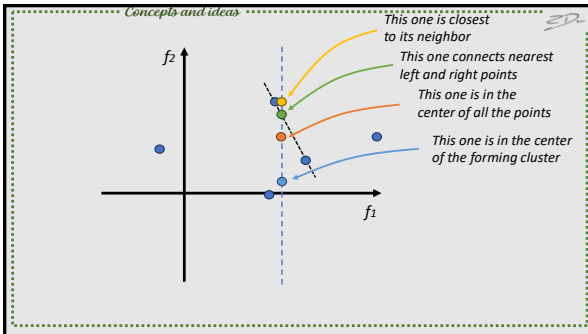
SD

So – we want to think about some ways of representing data, to build models that would be able to recognize patterns in multidimensional feature space

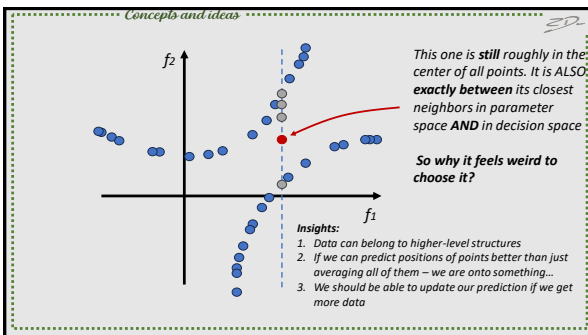
↓

- Learn rules that govern where points are expected to appear
- Organize points by separating them into sub-categories

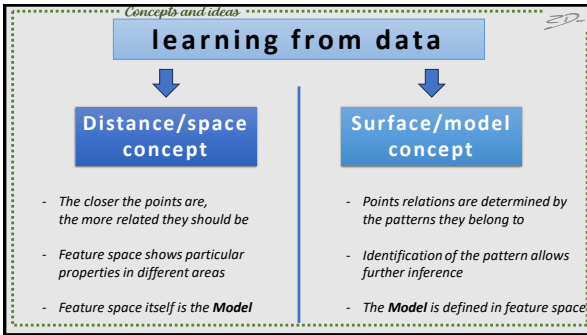
10



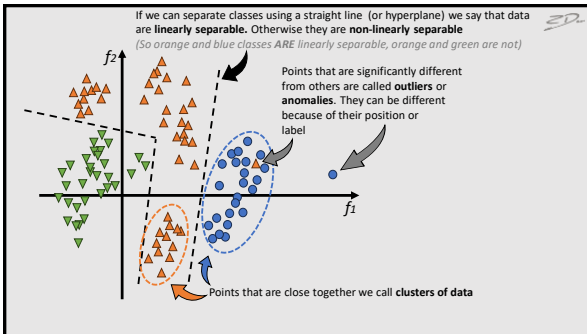
11



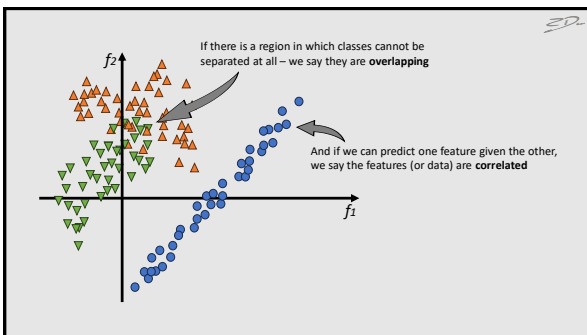
12



13



14



15

k Nearest Neighbor classifier (kNN)

1. Find k (odd) closest neighbors of the point
2. Assign the most common label

How will this method react to outliers?
 ↓
 We want $k \geq 3$

+ Intuitive
 + Simple in implementation
 + easy in configuration

- Require large memory
 - Computationally demanding
 - Poor scalability to high dimensional feature spaces

16

k Nearest Neighbor regressor

... we actually already know a similar method!
 (It is just almost exactly linearly weighted regression)

17

Clustering approaches

1. Start with an unlabeled dataset (this is an **unsupervised** method)
2. Find clusters of data
3. Color clusters using some data for which we know labels (e.g. label cluster by majority label)

18

Clustering approaches

1. Start with an unlabeled dataset (this is an **unsupervised method**)
2. Find clusters of data
3. Color clusters using some data for which we know labels (e.g. label cluster by majority label)

19

Clustering approaches

1. Start with an unlabeled dataset (this is an **unsupervised method**)
2. **Find clusters of data**
3. Color clusters using some data for which we know labels (e.g. label cluster by majority label)
4. For new data just check to which cluster samples belong and label them accordingly

20

K-means clustering

Assume number of clusters, pick their centers randomly

↓

Assign points to clusters according to their distance

↓

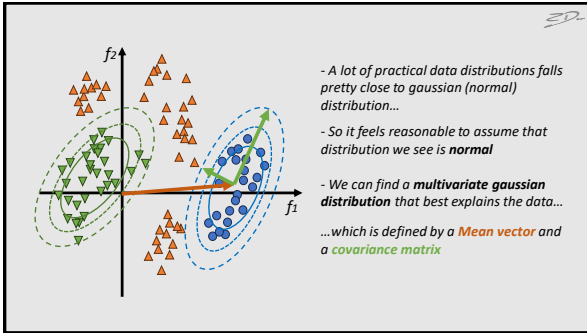
Recalculate centers so they are mean of points assigned

↓

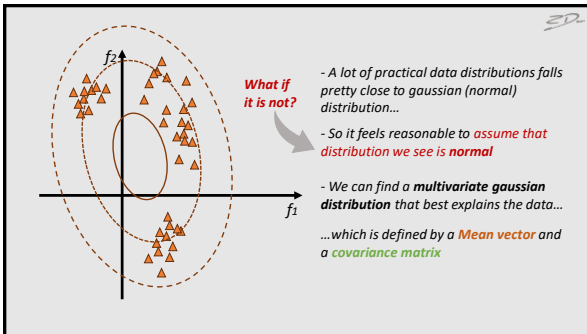
Repeat until cluster centers do not change any more

- + Intuitive
- + One of the simplest clustering methods
- + We know when to stop
- We should know how many clusters we want
- Cluster borders are not well founded in data
- Poor scalability to high dimensional feature spaces

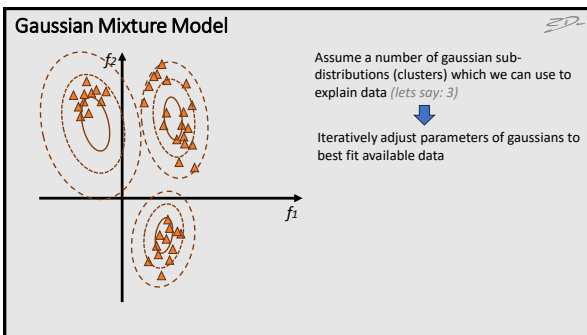
21



22



23



24

Gaussian Mixture Model SD

Assume a number of gaussian sub-distributions (clusters) which we can use to explain data (lets say: 6)

Iteratively adjust parameters of gaussians to best fit available data

What if we assume too many clusters?
Nothing really bad happens – we just have a more difficult optimization problem!

25

Gaussian Mixture Model SD

Assume a number of gaussian sub-distributions (clusters) which we can use to explain data (lets say: 3)

Iteratively adjust parameters of gaussians to best fit available data

What if we assume too many clusters?
Nothing really bad happens – we just have a more difficult optimization problem!

What if sub-distributions are not gaussian either?
If we know that – we can plug in different distributions
If we don't – we are still pretty close to the actual one

26

Gaussian Mixture Model SD

- + Provides measure of probability of class presence
- + Robust to outliers
- + Can be used for classification and for unsupervised clustering

- Requires either strict knowledge of cluster numbers or large computational effort
- In high dimensional spaces requires lots of data

27

Things to remember:

SD

1. Show how experimental data are gathered into subsets, name them and say what is their purpose
2. Explain linear separability, outliers, clusters and correlated data. Provide graphical examples for these explanations
3. Explain kNN method for classification (with pros and cons)
4. Explain how clustering works in general, explain KMC method with its pros and cons
5. Explain how Gaussian Mixture Model works – show example of gaussians fitted into clusters of data, provide pros and cons of GMM
