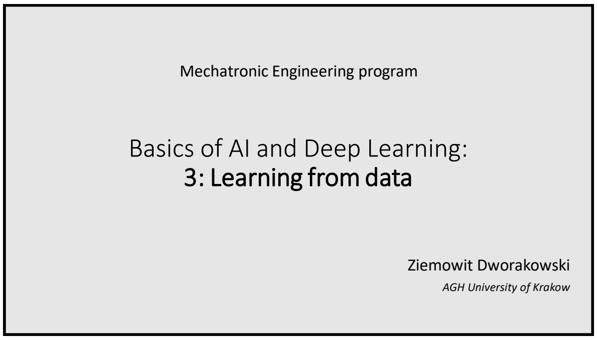


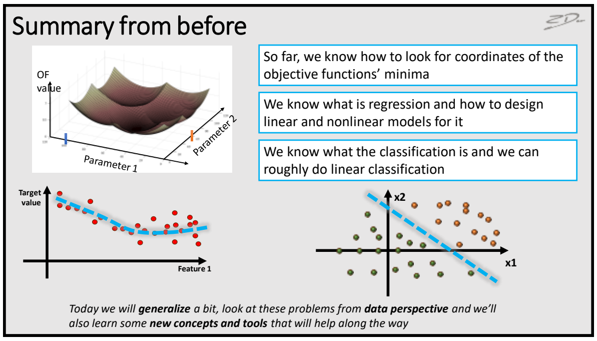
Mechatronic Engineering program

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

Ziemowit Dworakowski

AGH University of Krakow

1



Summary from before

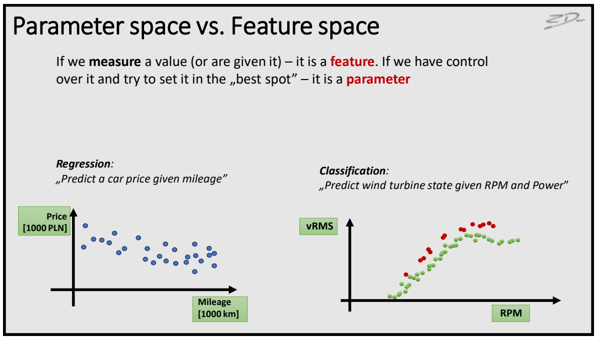
So far, we know how to look for coordinates of the objective functions' minima

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

*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



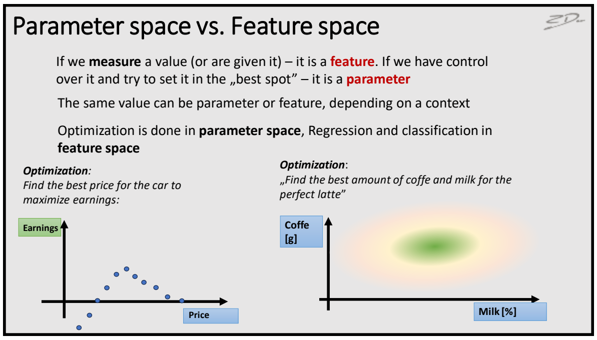
Parameter space vs. Feature space

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"

***Classification**:*
„Predict wind turbine state given RPM and Power"

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**

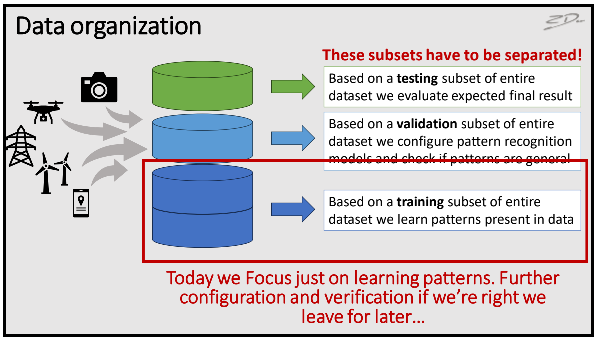
The same value can be parameter or feature, depending on a context

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!

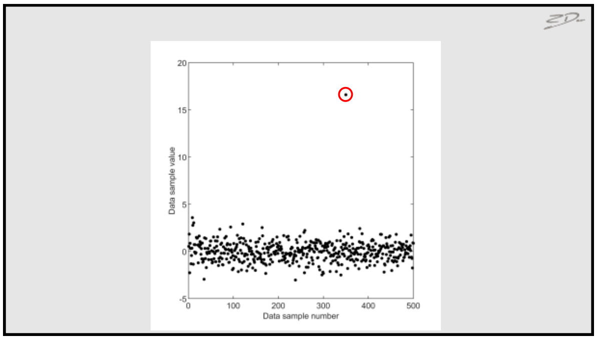
Based on a **testing** subset of entire dataset we evaluate expected final result

Based on a **validation** subset of entire dataset we configure pattern recognition models and check if patterns are general

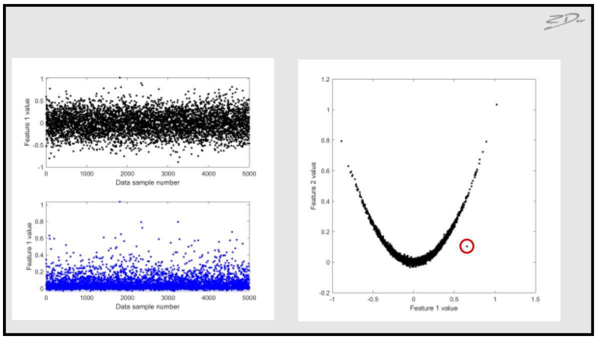
Based on a **training** subset of entire dataset we learn patterns present in data

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

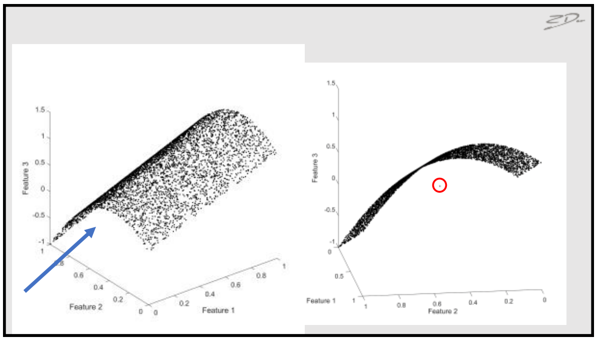
5



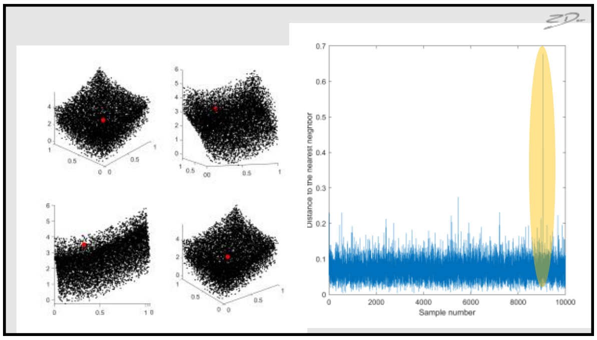
6


Feature 1 value
Data sample number
Feature 1 value
Data sample number
Feature 2 value
Feature 1 value

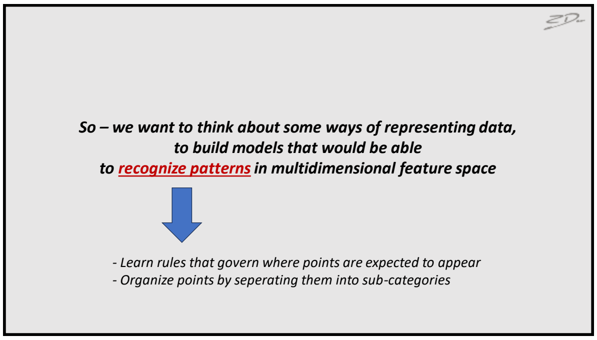
7


Feature 3
Feature 2
Feature 1
Feature 3
Feature 1
Feature 2

8


Distance to the nearest neighbor
Sample number

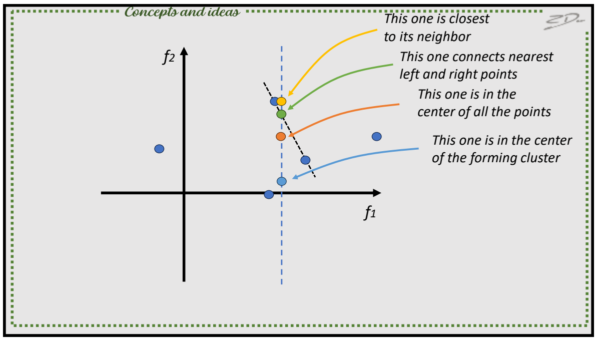
9



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

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

10



Concepts and ideas

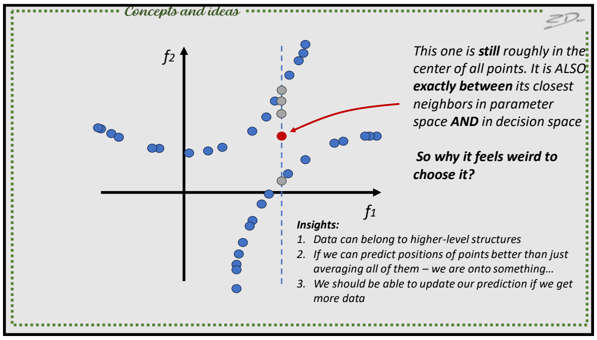
This one is closest to its neighbor

This one connects nearest left and right points

This one is in the center of all the points

This one is in the center of the forming cluster

11



Concepts and ideas

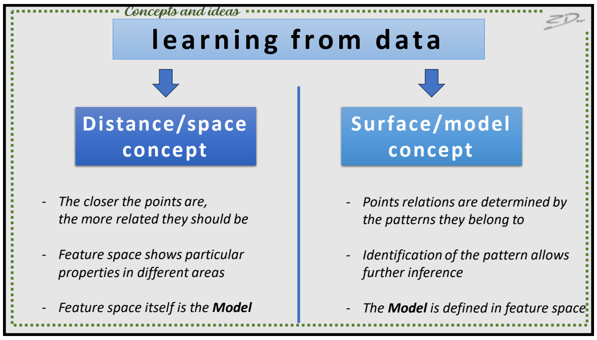
This one is ***still*** *roughly in the center of all points. It is ALSO* ***exactly between*** *its closest neighbors in parameter space* ***AND*** *in decision space*

So why it feels weird to choose it?

Insights:

1. *Data can belong to higher-level structures*
2. *If we can predict positions of points better than just averaging all of them – we are onto something…*
3. *We should be able to update our prediction if we get more data*

12



Concepts and ideas

learning from data

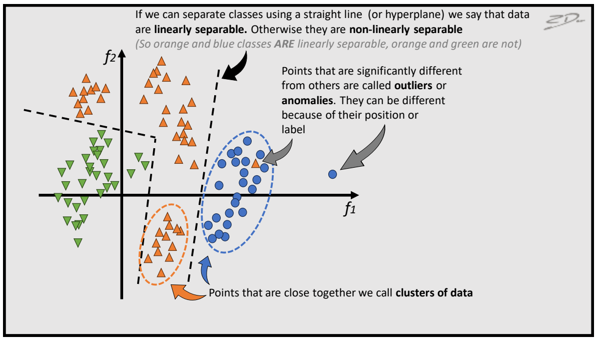
Distance/space concept

- The closer the points are, the more related they should be
- Feature space shows particular properties in different areas
- Feature space itself is the **Model**

Surface/model concept

- Points relations are determined by the patterns they belong to
- Identification of the pattern allows further inference
- The **Model** is defined in feature space

13



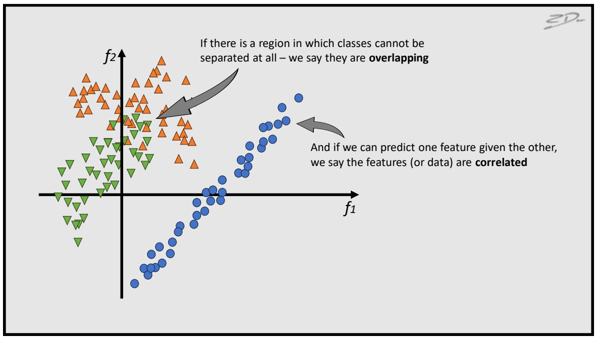
If we can separate classes using a straight line (or hyperplane) we say that data are **linearly separable.** Otherwise they are **non-linearly separable**

(So orange and blue classes ***ARE*** *linearly separable, orange and green are not)*

Points that are significantly different from others are called **outliers** or **anomalies**. They can be different because of their position or label

Points that are close together we call **clusters of data**

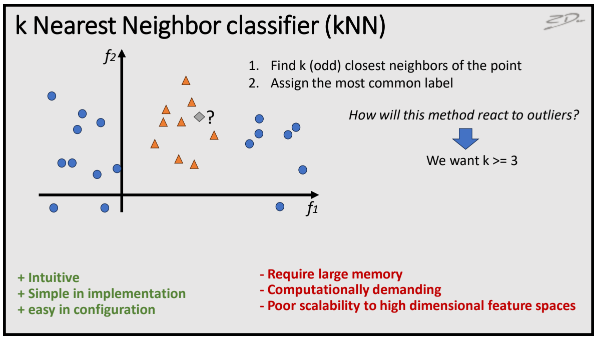
14



If there is a region in which classes cannot be separated at all – we say they are **overlapping**

And if we can predict one feature given the other, we say the features (or data) are **correlated**

15



k Nearest Neighbor classifier (kNN)

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

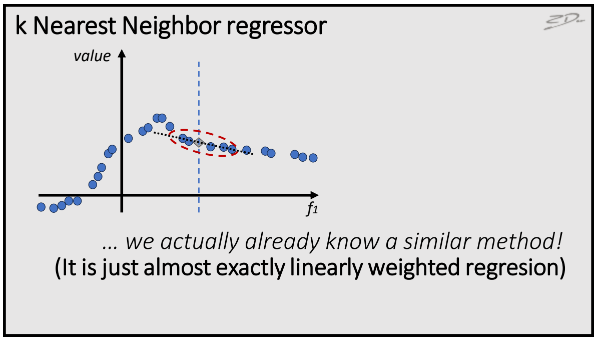
How will this method react to outliers?

We want k >= 3

+ Intuitive
+ Simple in implementation
+ easy in configuration

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

16

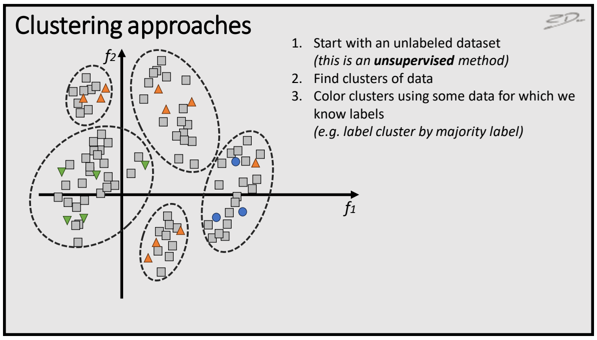


k Nearest Neighbor regressor

… we actually already know a similar method!

(It is just almost exactly linearly weighted regresion)

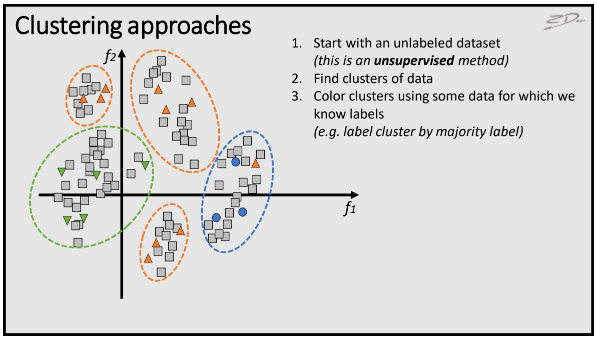
17



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)

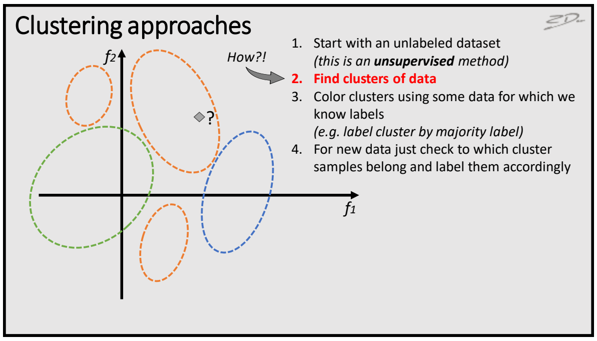
18



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)

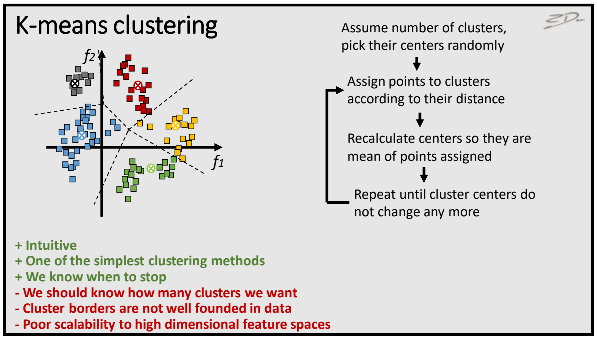
19



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

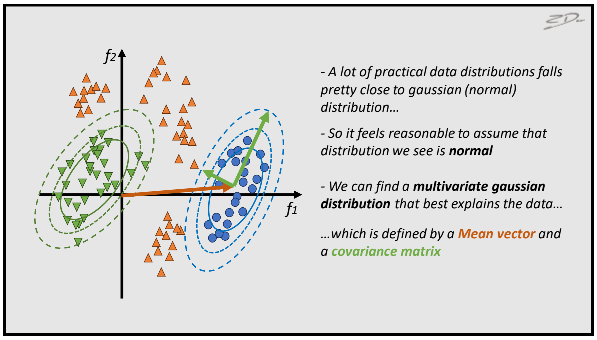
Assign points to clusters according to their distance

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



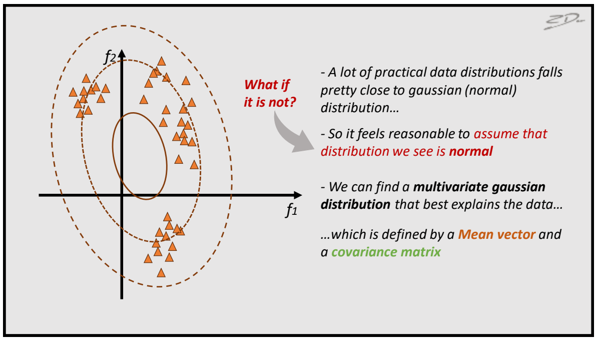
- *A lot of practical data distributions falls pretty close to gaussian (normal) distribution…*

- *So it feels reasonable to assume that distribution we see is* ***normal***

- *We can find a* ***multivariate gaussian distribution*** *that best explains the data…*

…which is defined by a ***Mean vector*** *and a* ***covariance matrix***

22



What if it is not?

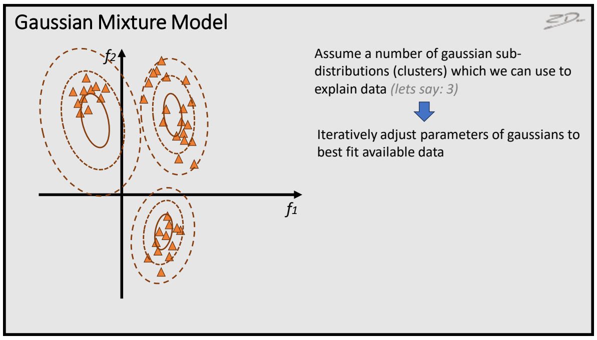
- *A lot of practical data distributions falls pretty close to gaussian (normal) distribution…*

- *So it feels reasonable to assume that distribution we see is* ***normal***

- *We can find a* ***multivariate gaussian distribution*** *that best explains the data…*

…which is defined by a ***Mean vector*** *and a* ***covariance matrix***

23

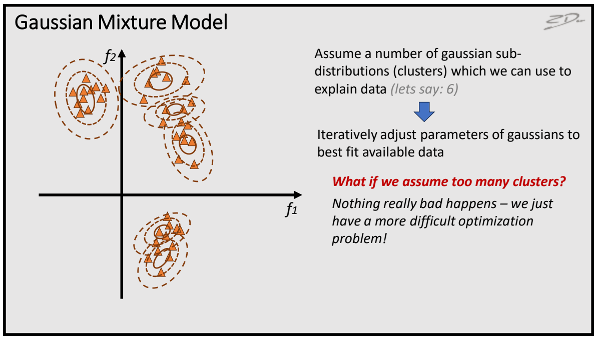


Gaussian Mixture Model

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

24



Gaussian Mixture Model

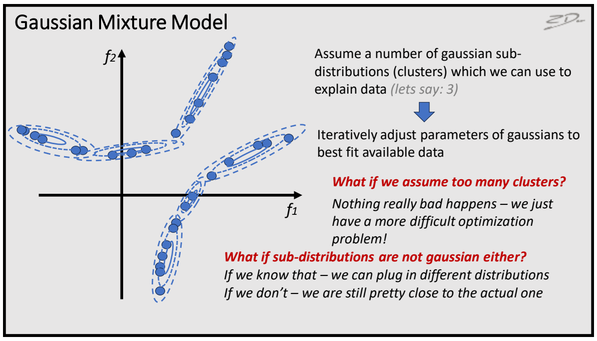
Assume a number of gaussian sub-distributions (clusters) which we can use to explain data *(lets say: 6)*

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

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

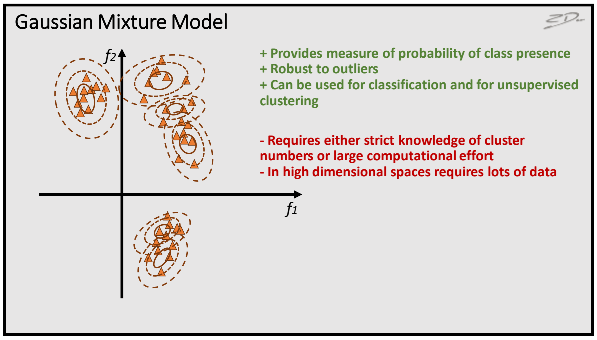
What if we assume too many clusters?

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

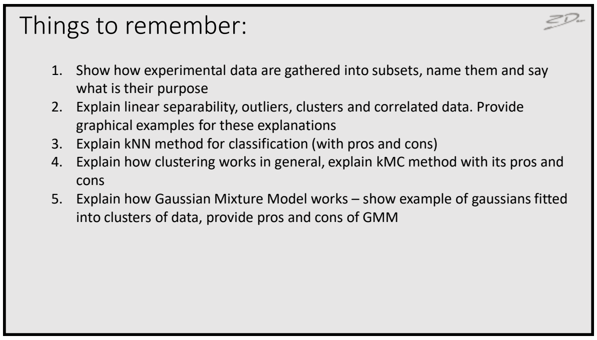


Gaussian Mixture Model

+ 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:

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

28