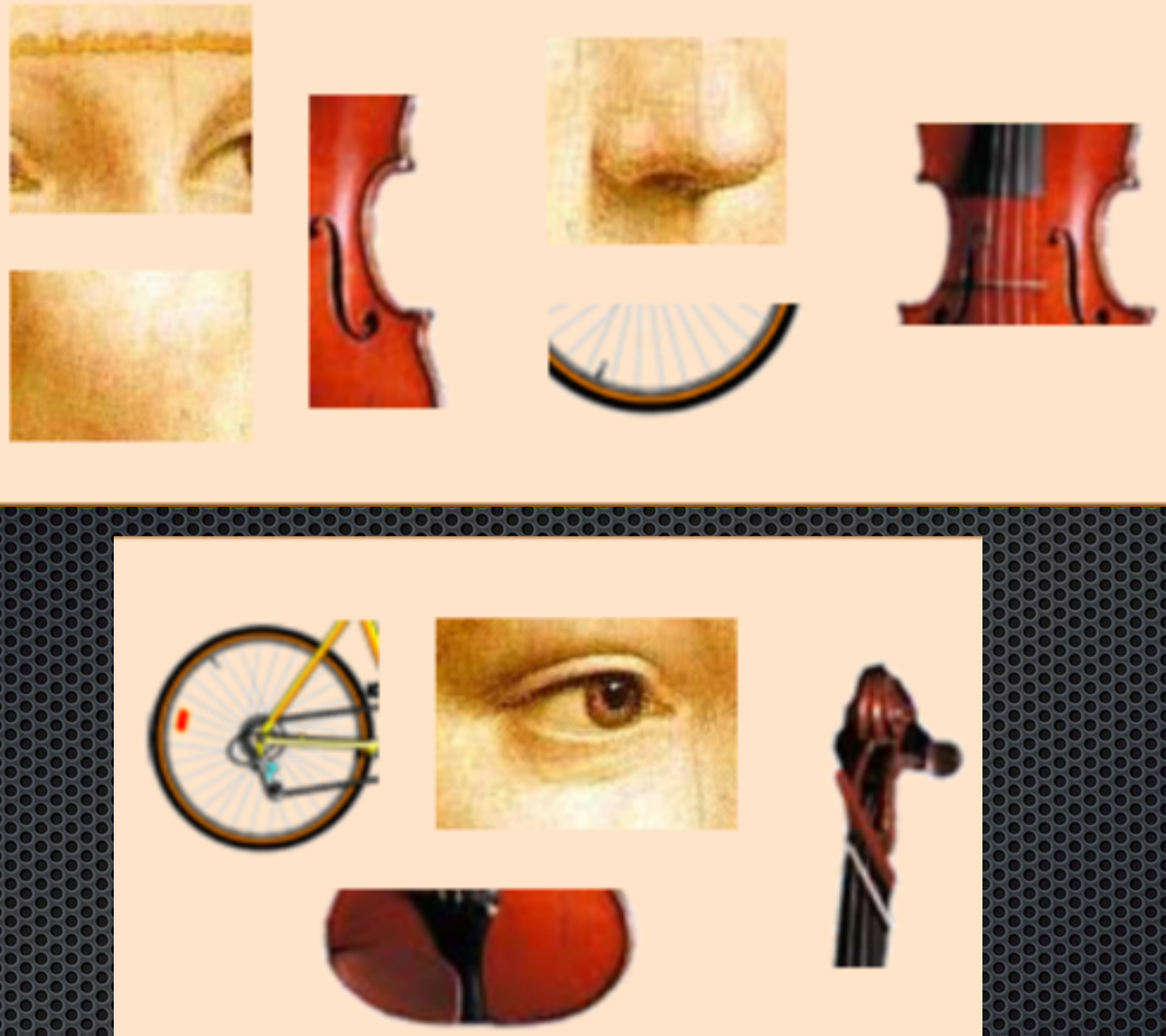
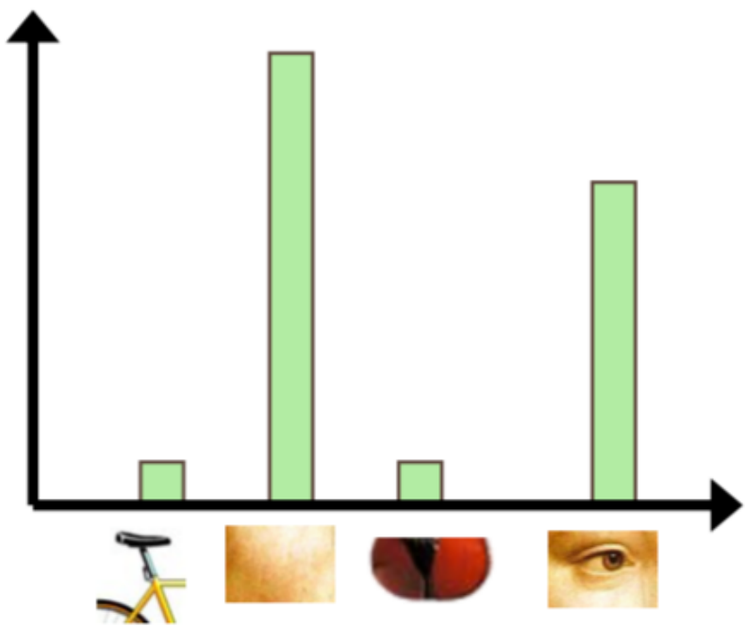
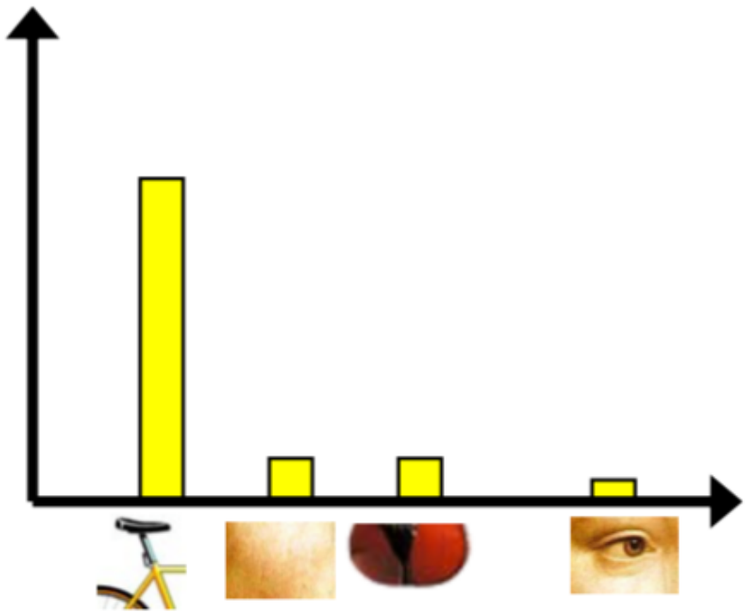
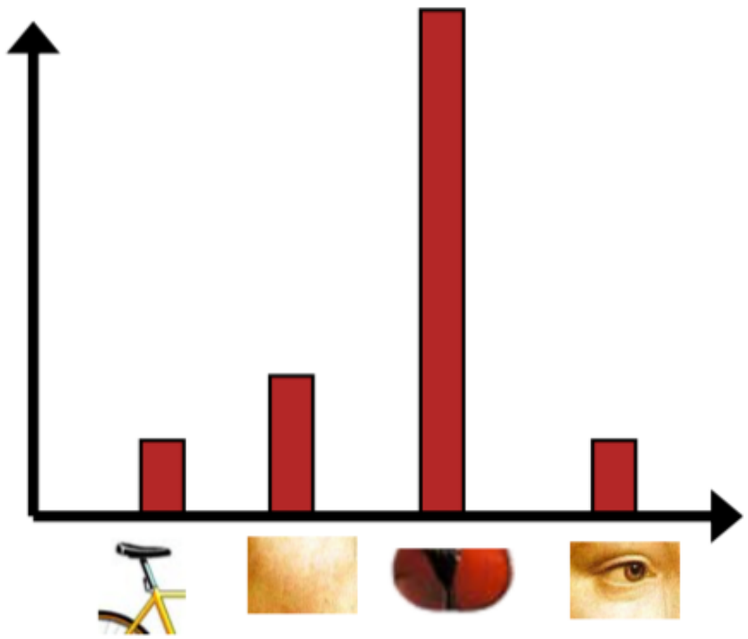
Machine learning
Computer Vision

There are two major learning types. Supervised and unsupervised.









Now, how to frame a model of these features, such that we can group together images of similar objects?

Let’s represent pets on a number line, based on say cuteness

Figure 2. A single feature does not result in a perfect separation of our training data.

Let’s add intelligence as another feature

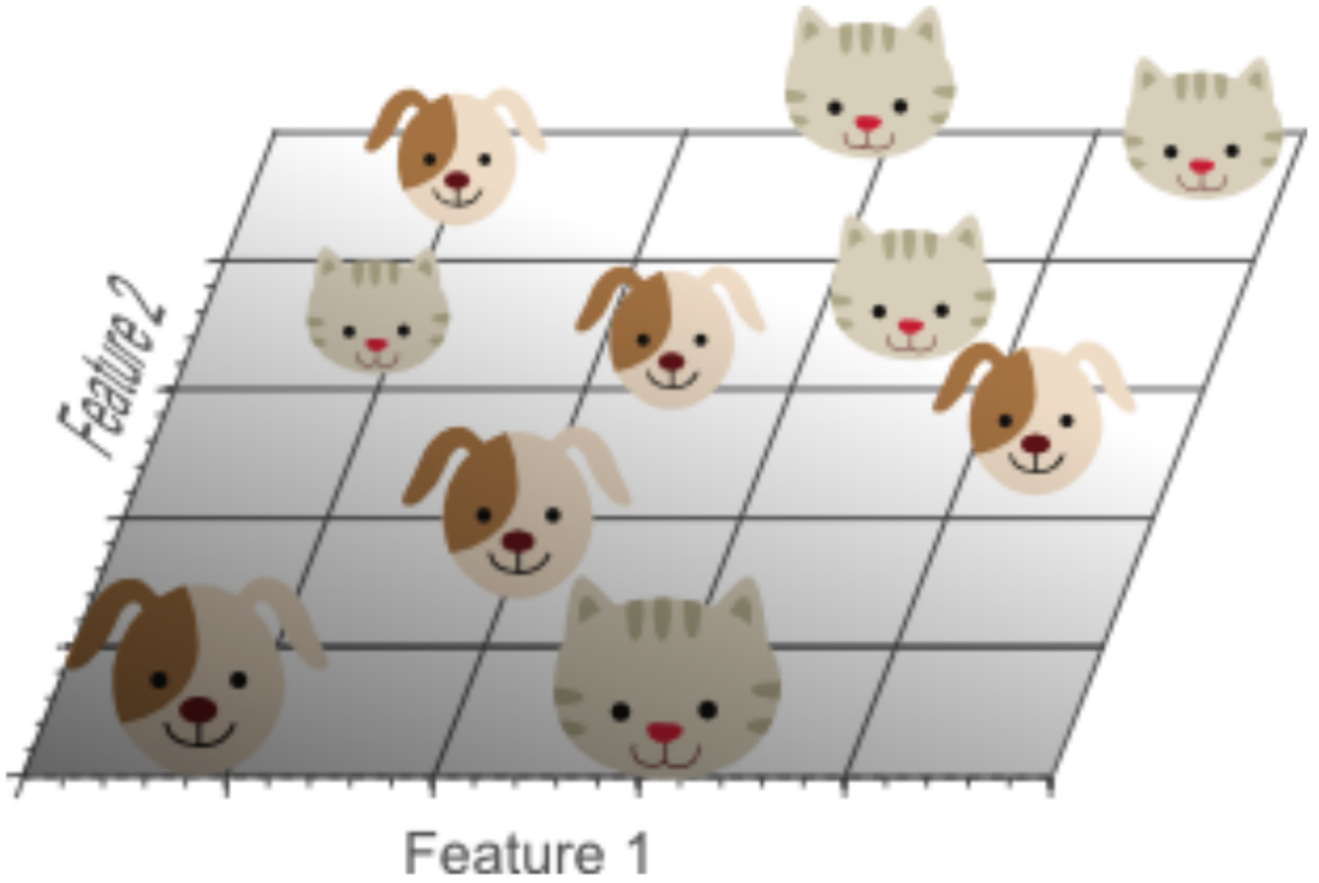


Figure 3. Adding a second feature still does not result in a linearly separable classification problem: No single line can separate all cats from all dogs in this example.

Let’s add cunningness as a third feature

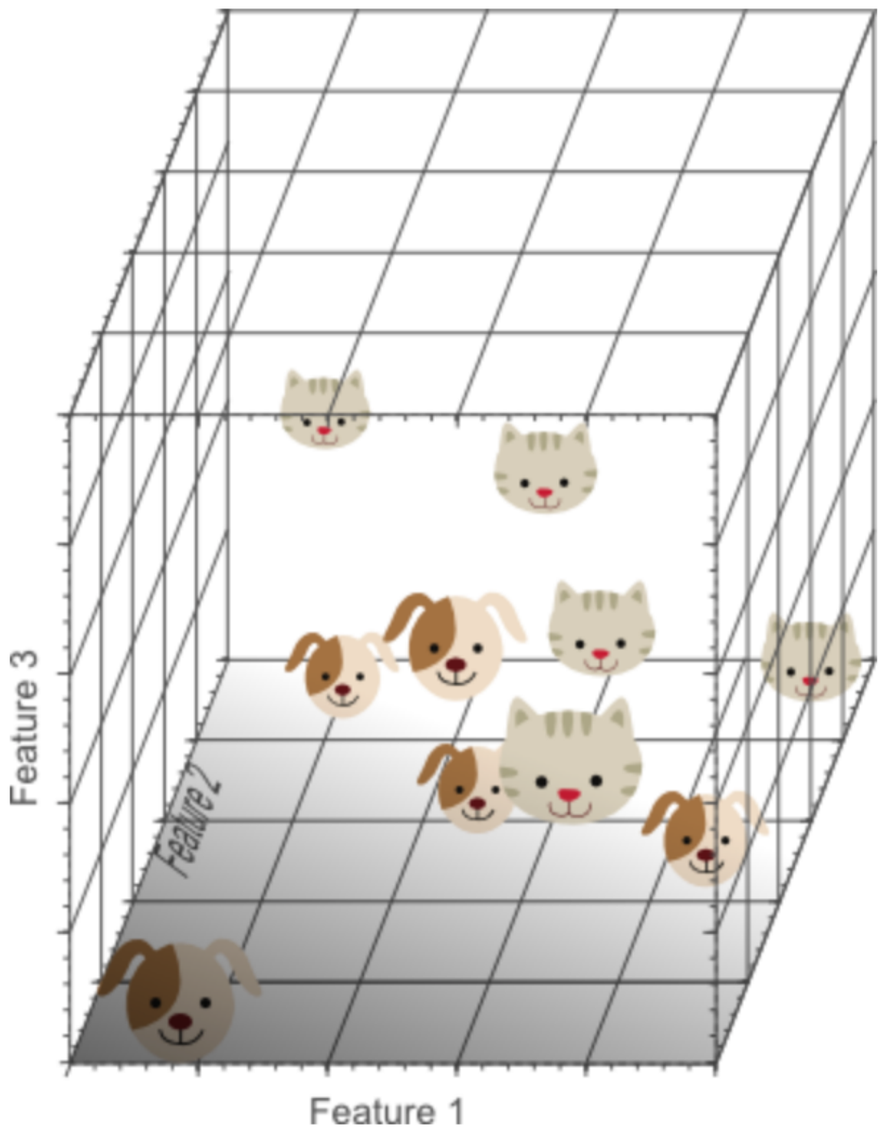


Figure 4. Adding a third feature results in a linearly separable classification problem in our example. A plane exists that perfectly separates dogs from cats.

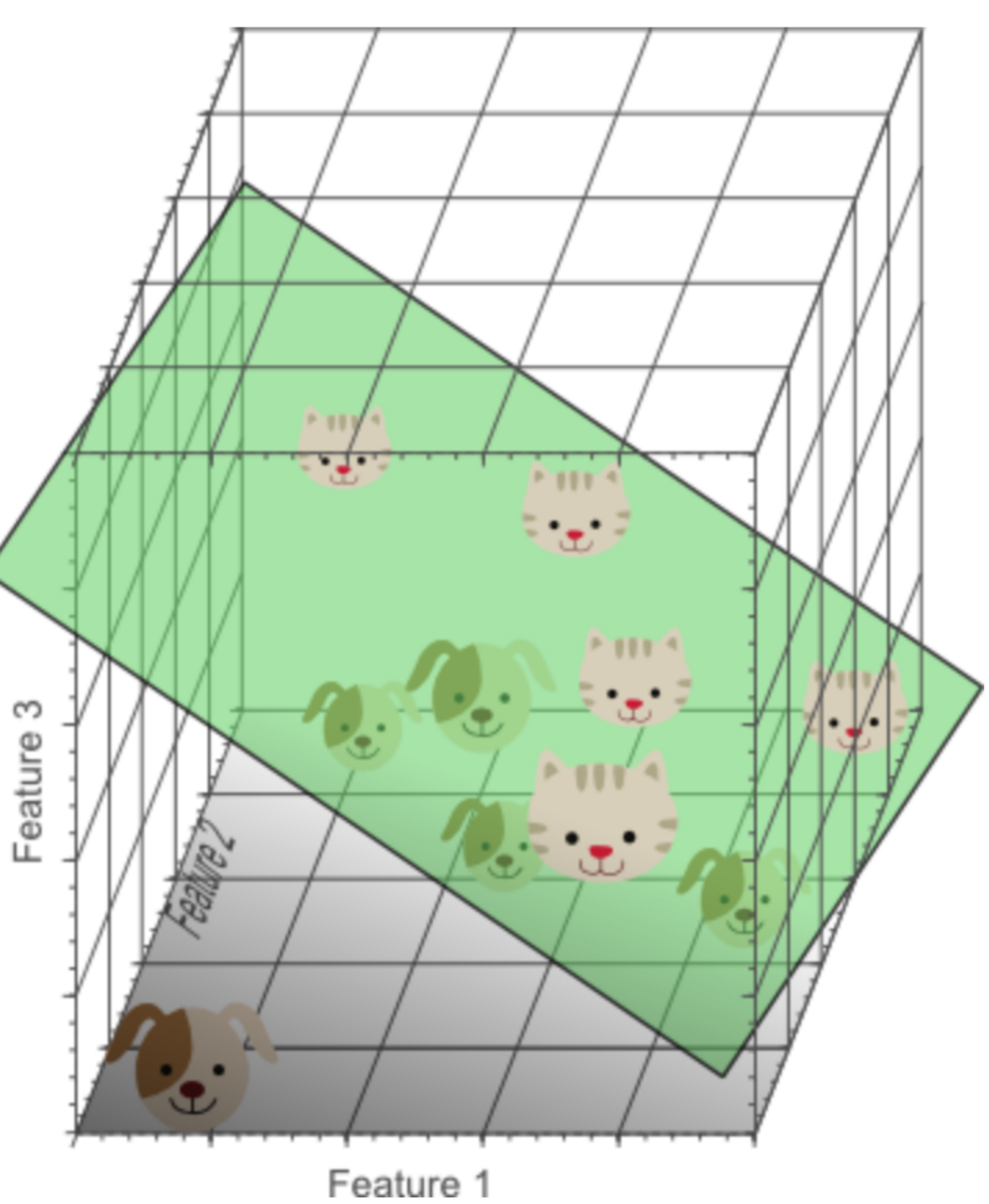
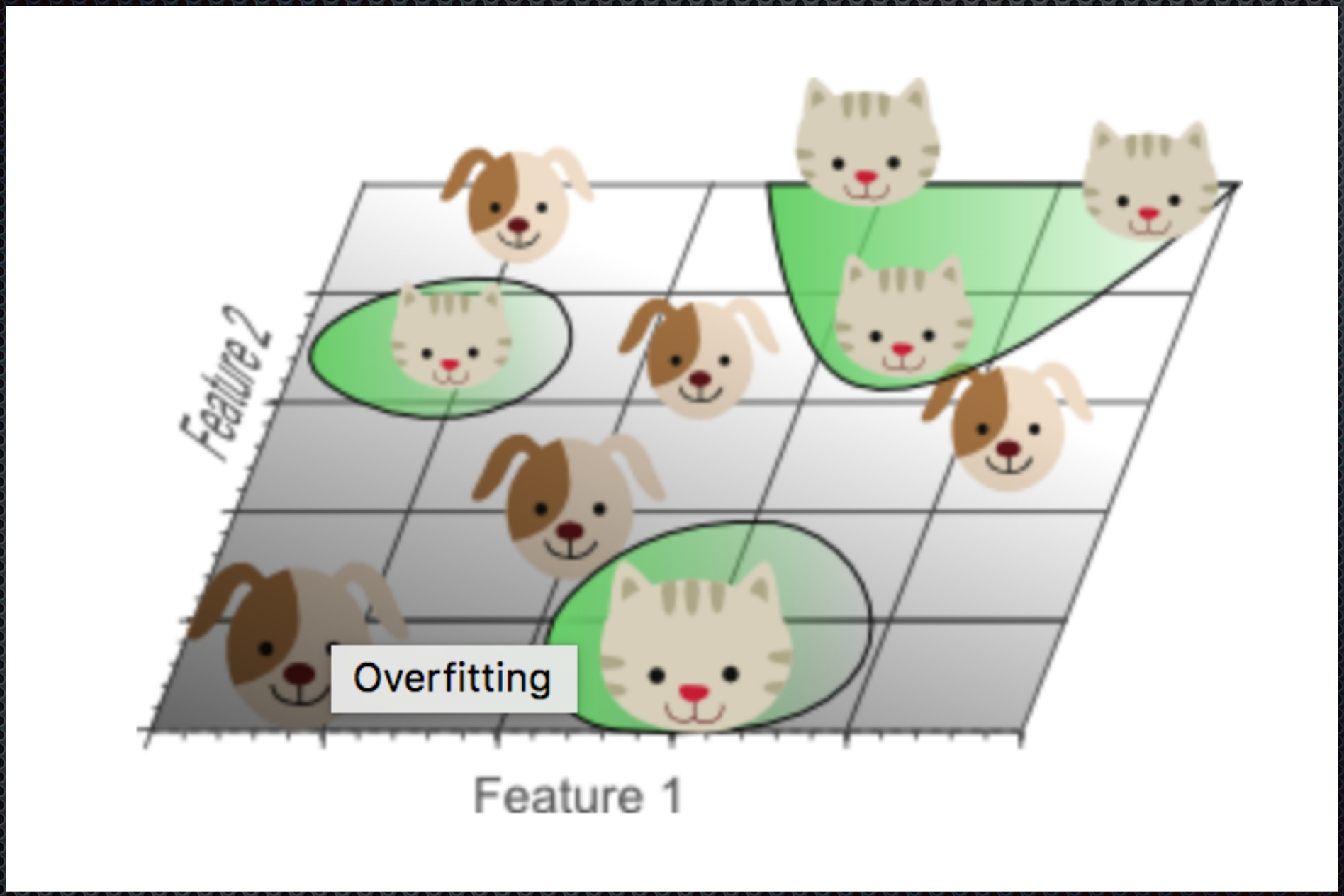
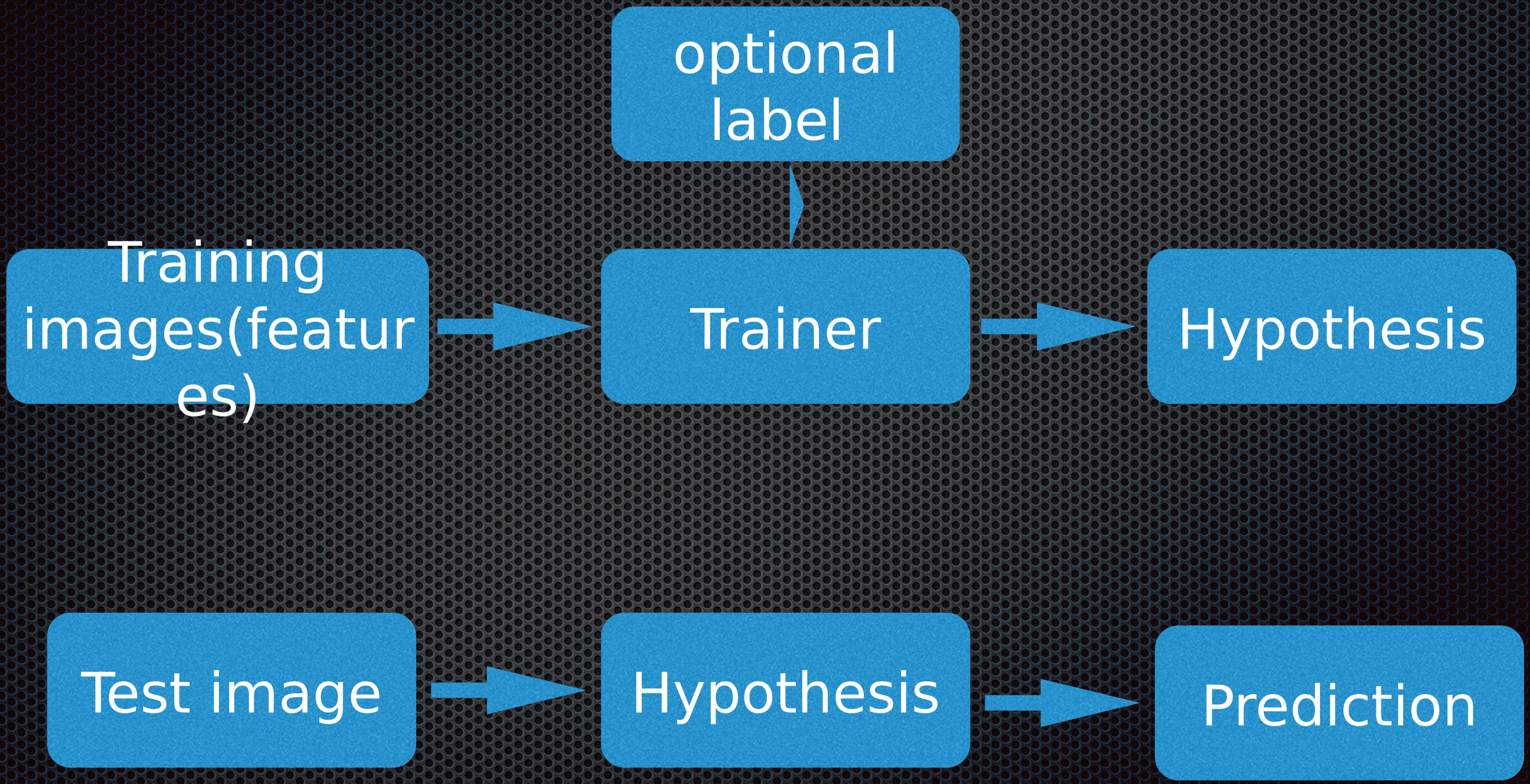


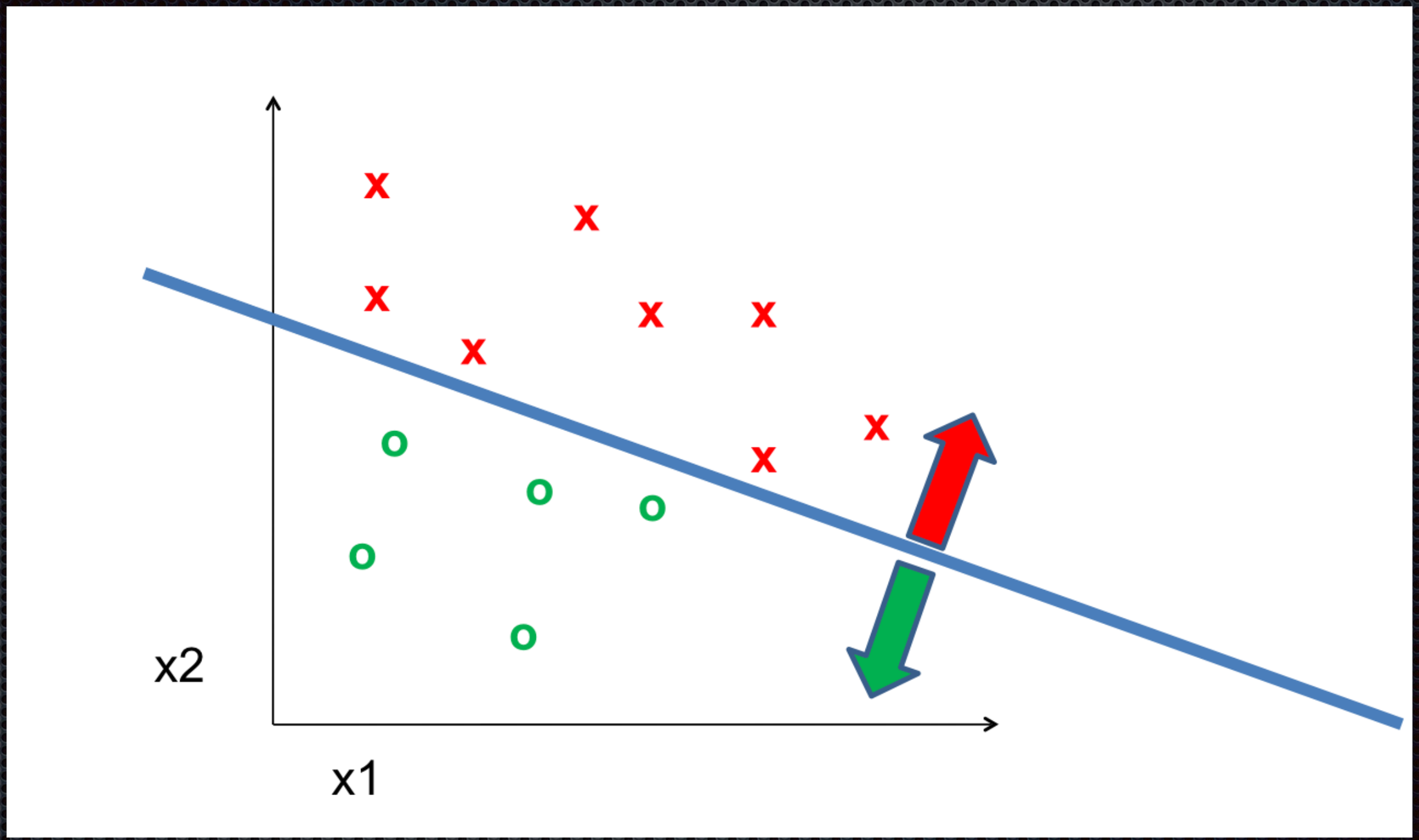
Figure 5. The more features we use, the higher the likelihood that we can successfully separate the classes perfectly.

What's wrong with our hypothesis?



Classifiers




x
x
x
x
x
x
x
x
o
o
o
o
o
x2
x1

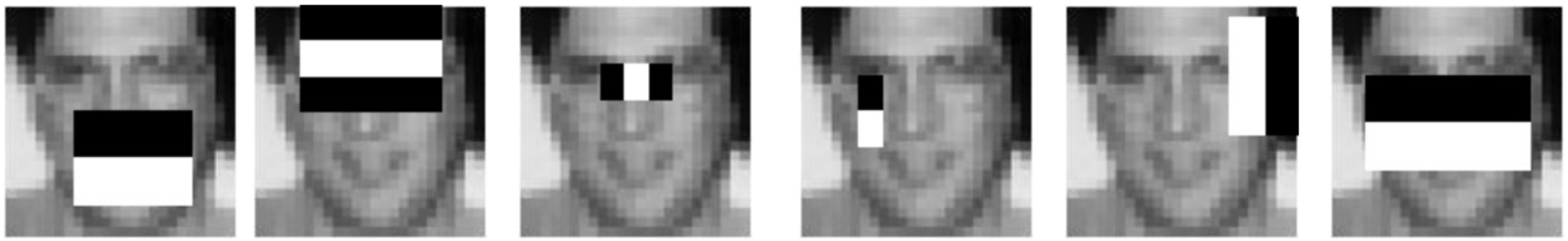
What features to use ?

Faces. How do you recognise them ?

VIOLA JONES - USING HAAR-LIKE FEATURES

Stage 0

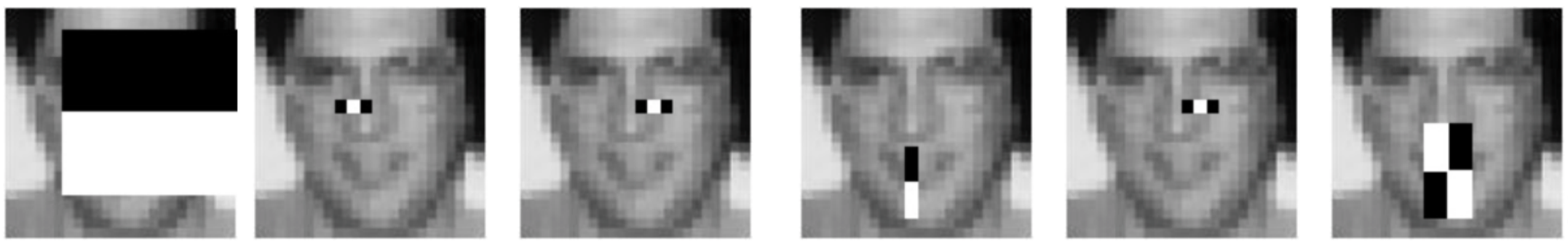
Stage 1



... 10 more

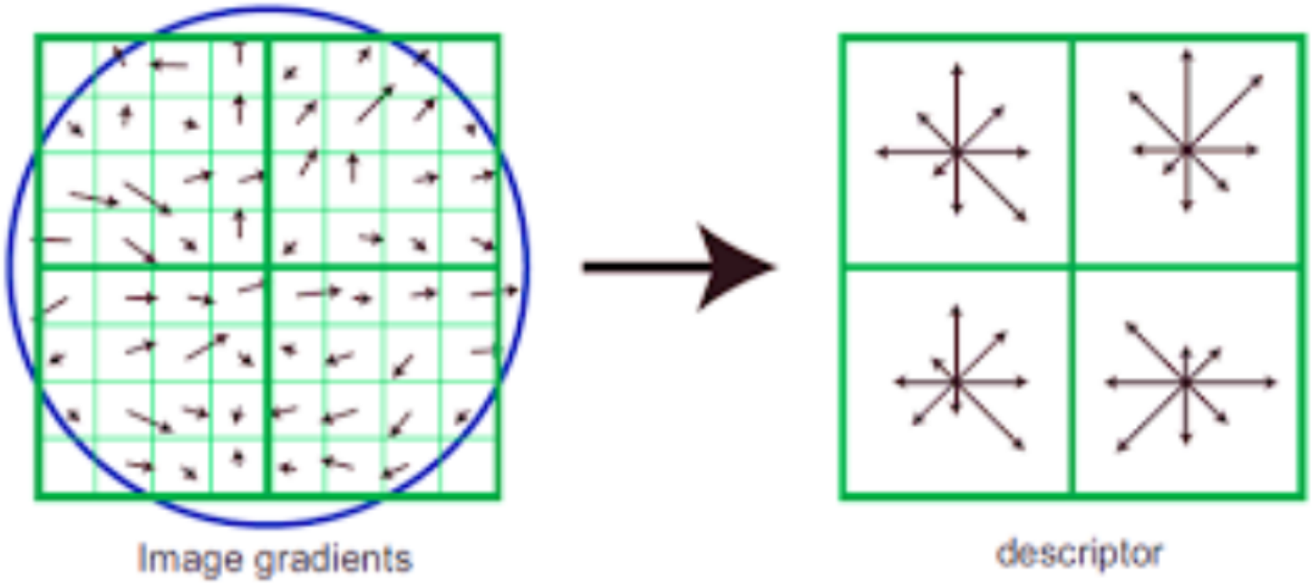
.
.

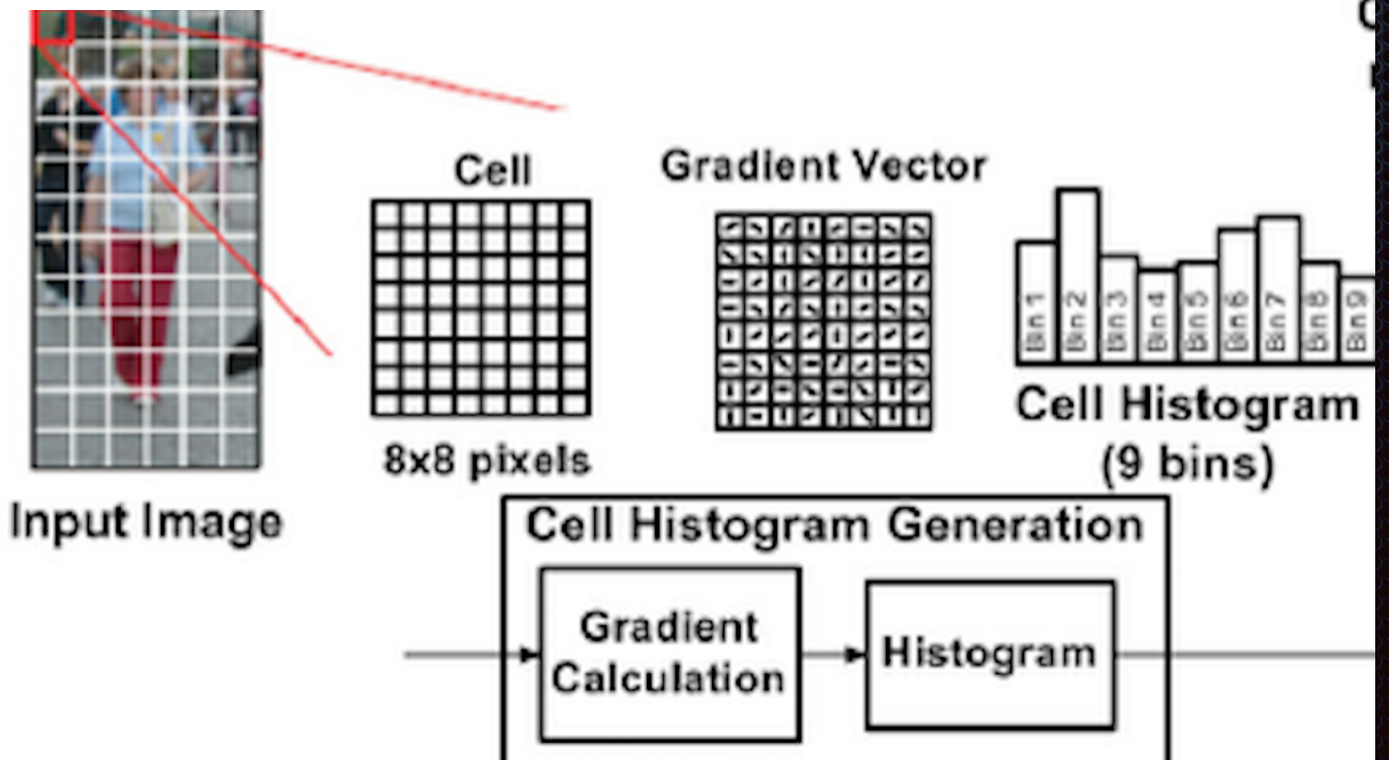
Stage 21



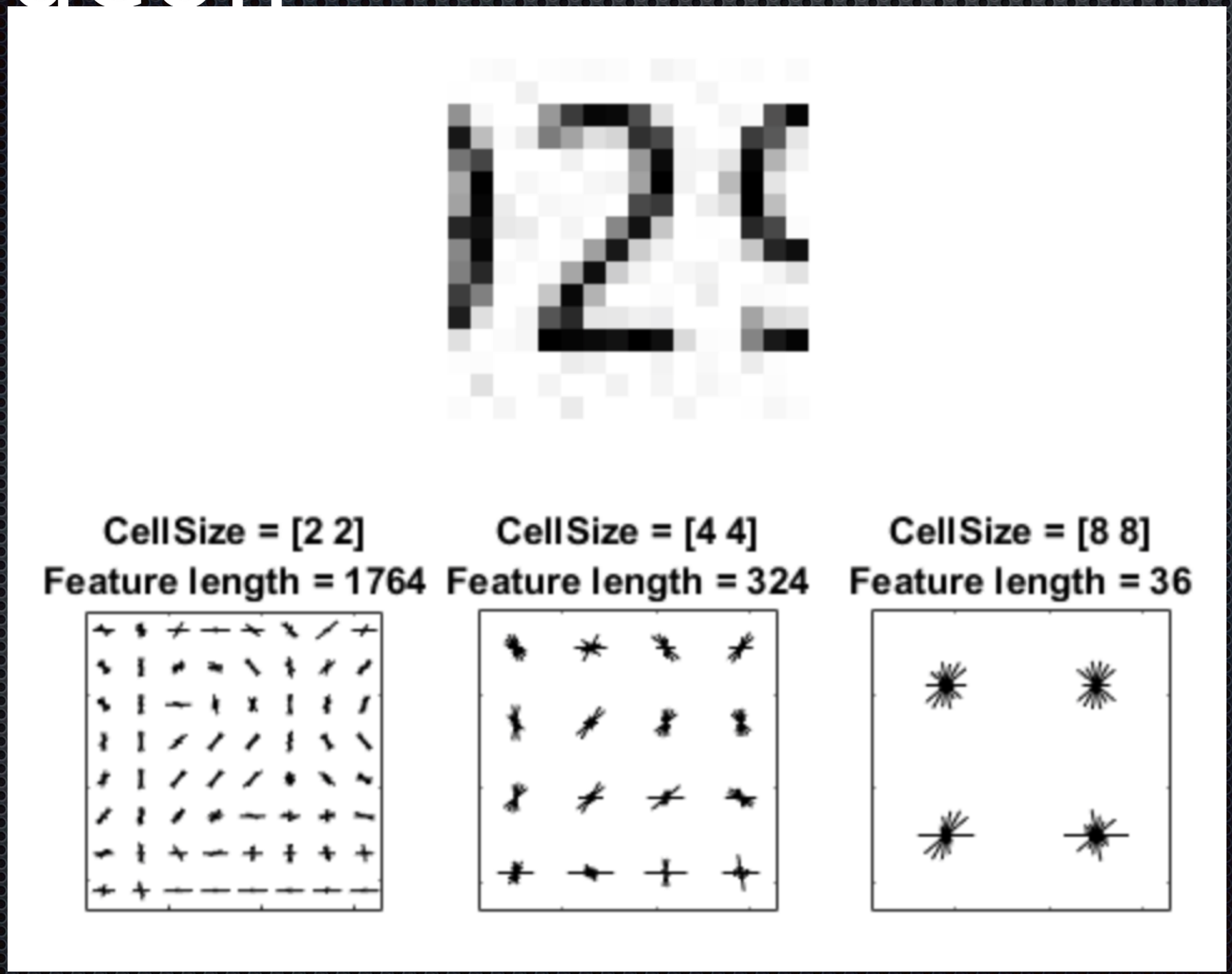
... 206 more

Histogram of Oriented Gradients (HOG)





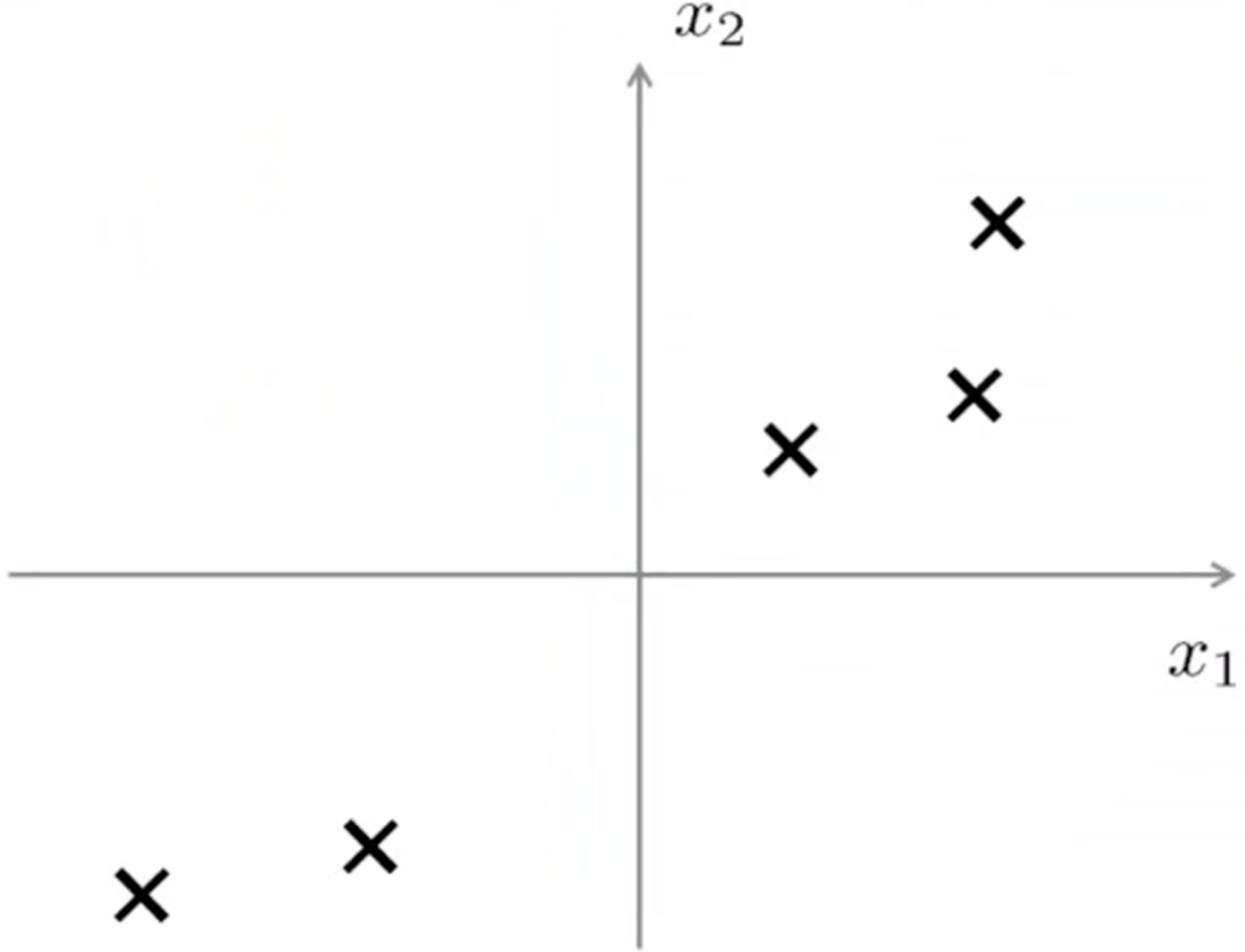
Space vs. Accuracy tradeoff



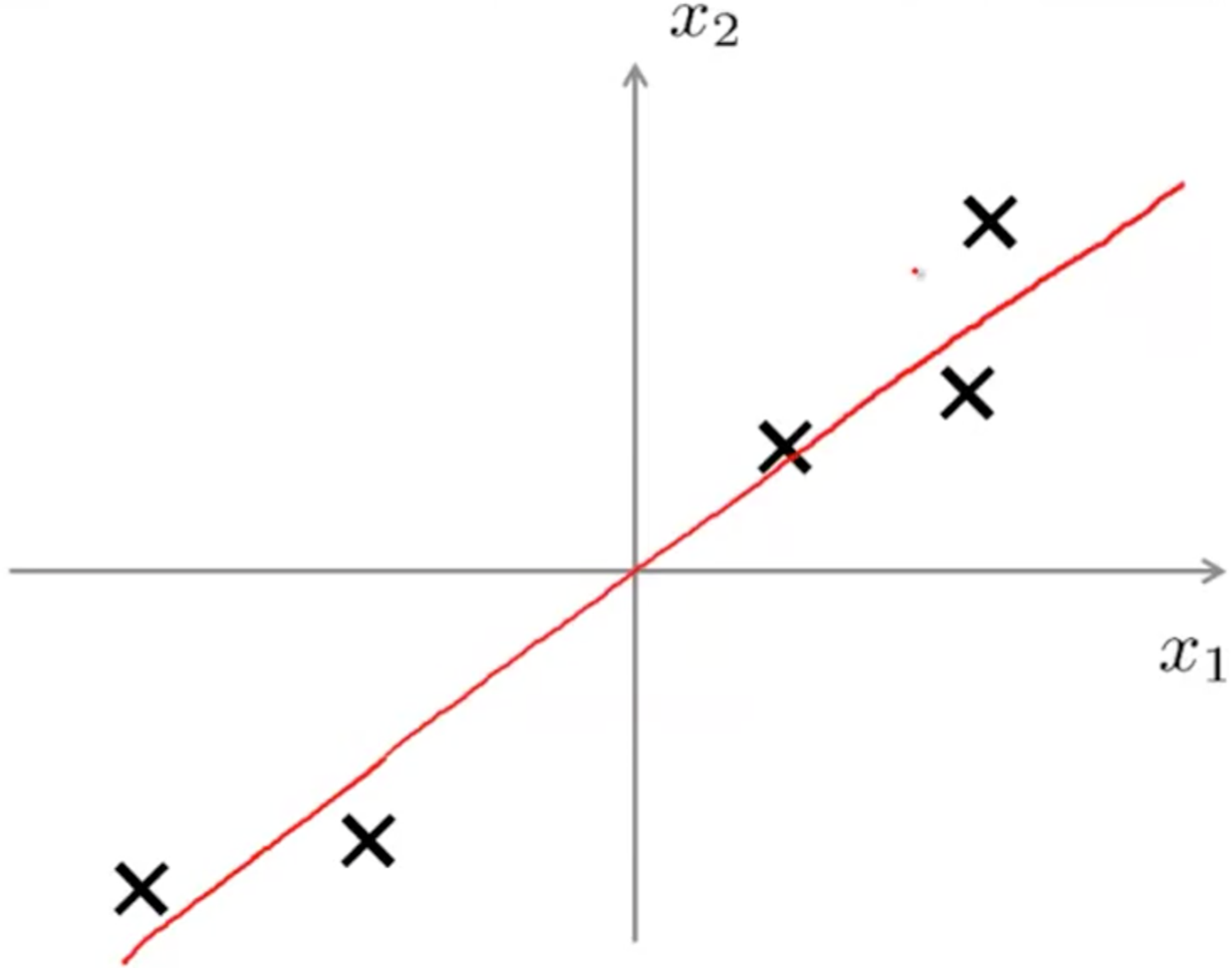
Principal Component analysis

- To handle such large feature dimensions, we reduce the feature space to make computations feasible.

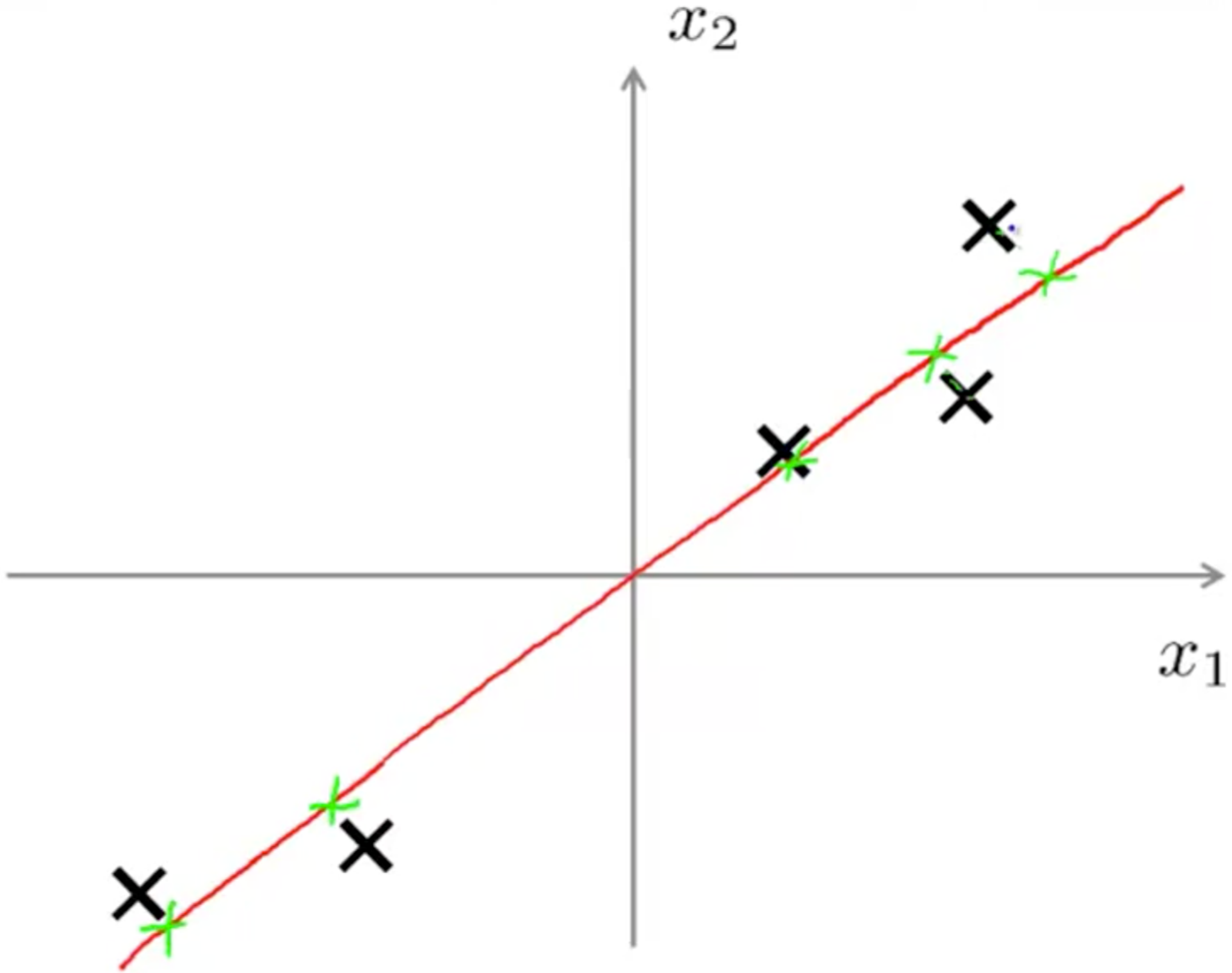
Principal Component Analysis (PCA) problem formulation



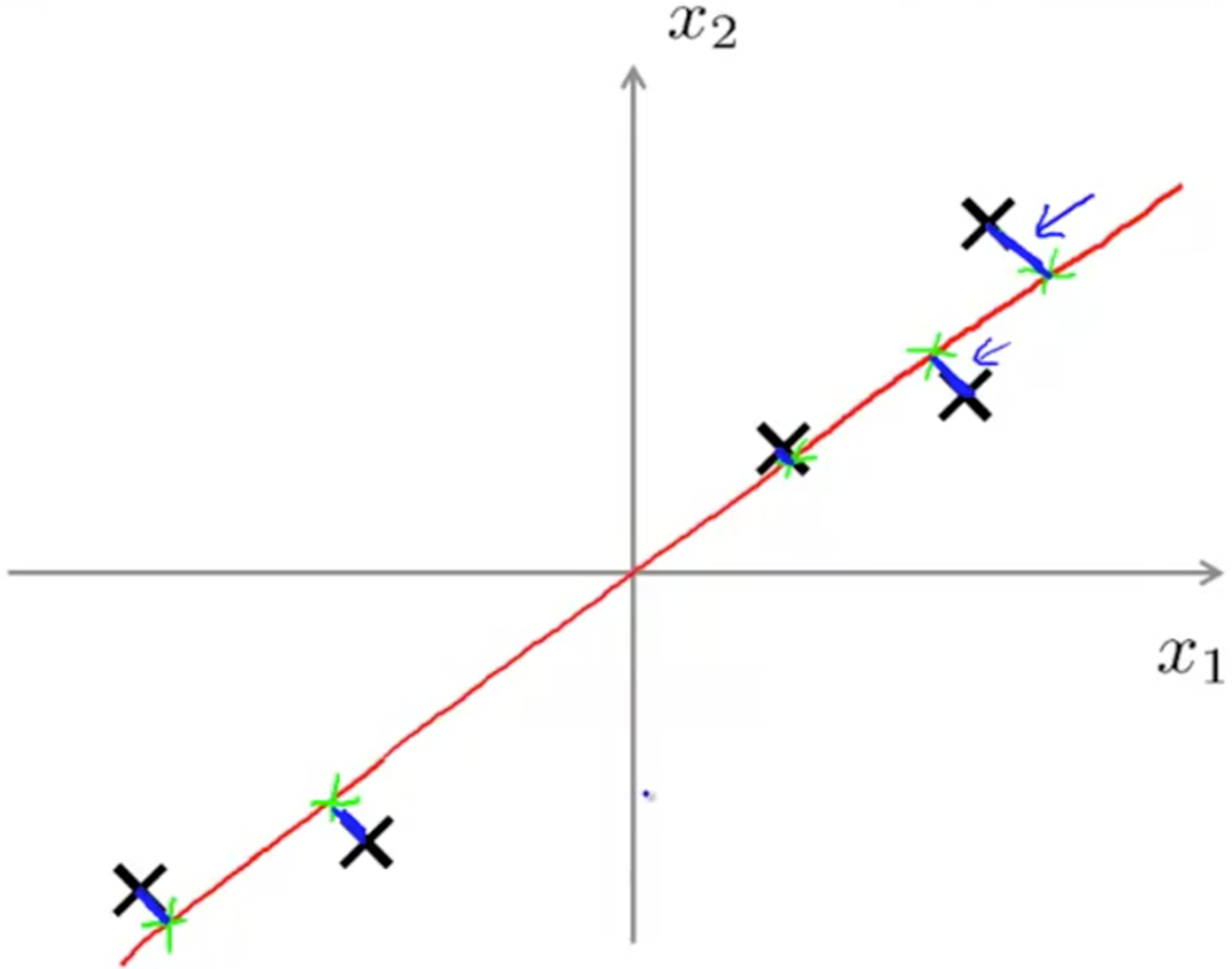
Principal Component Analysis (PCA) problem formulation



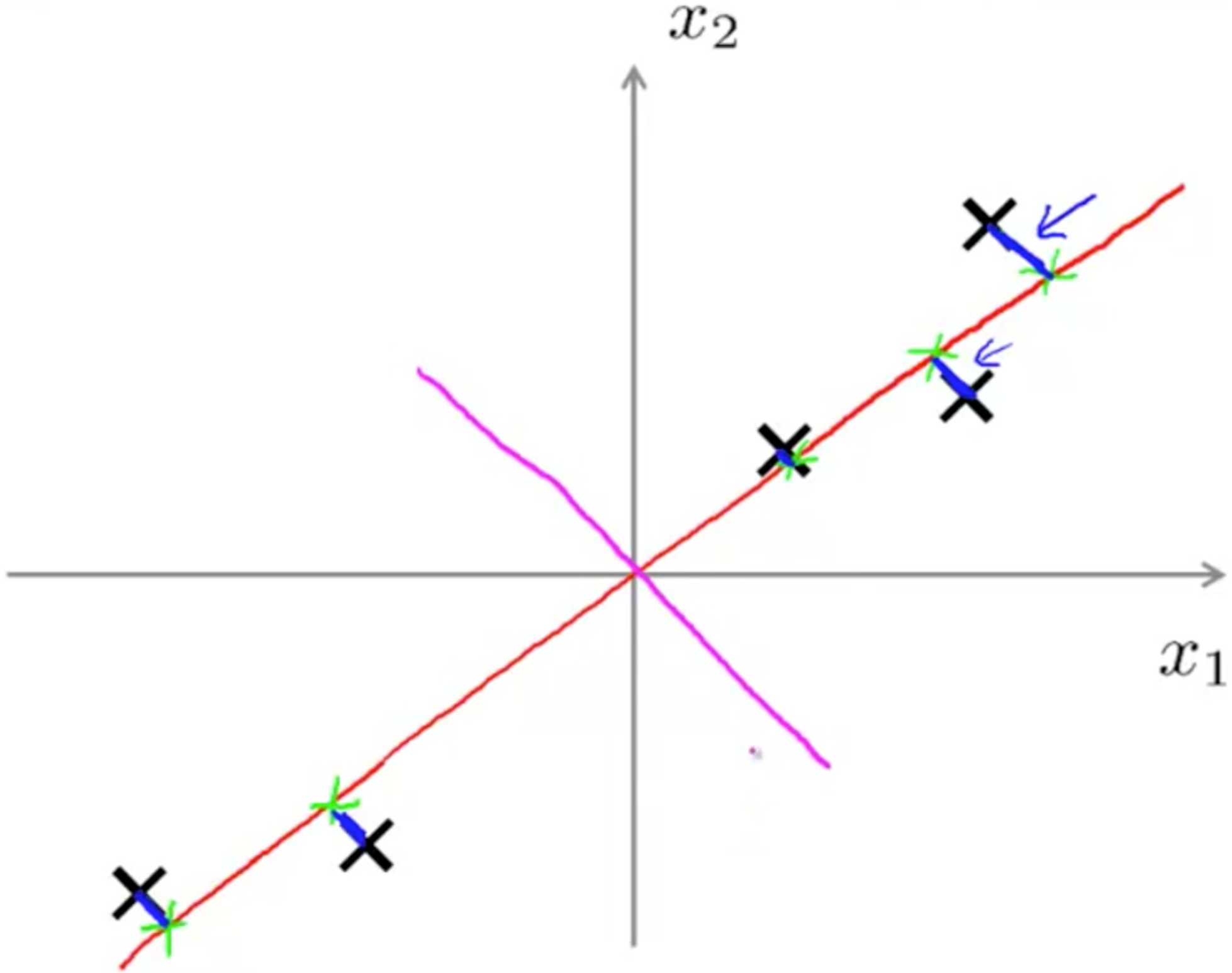
Principal Component Analysis (PCA) problem formulation



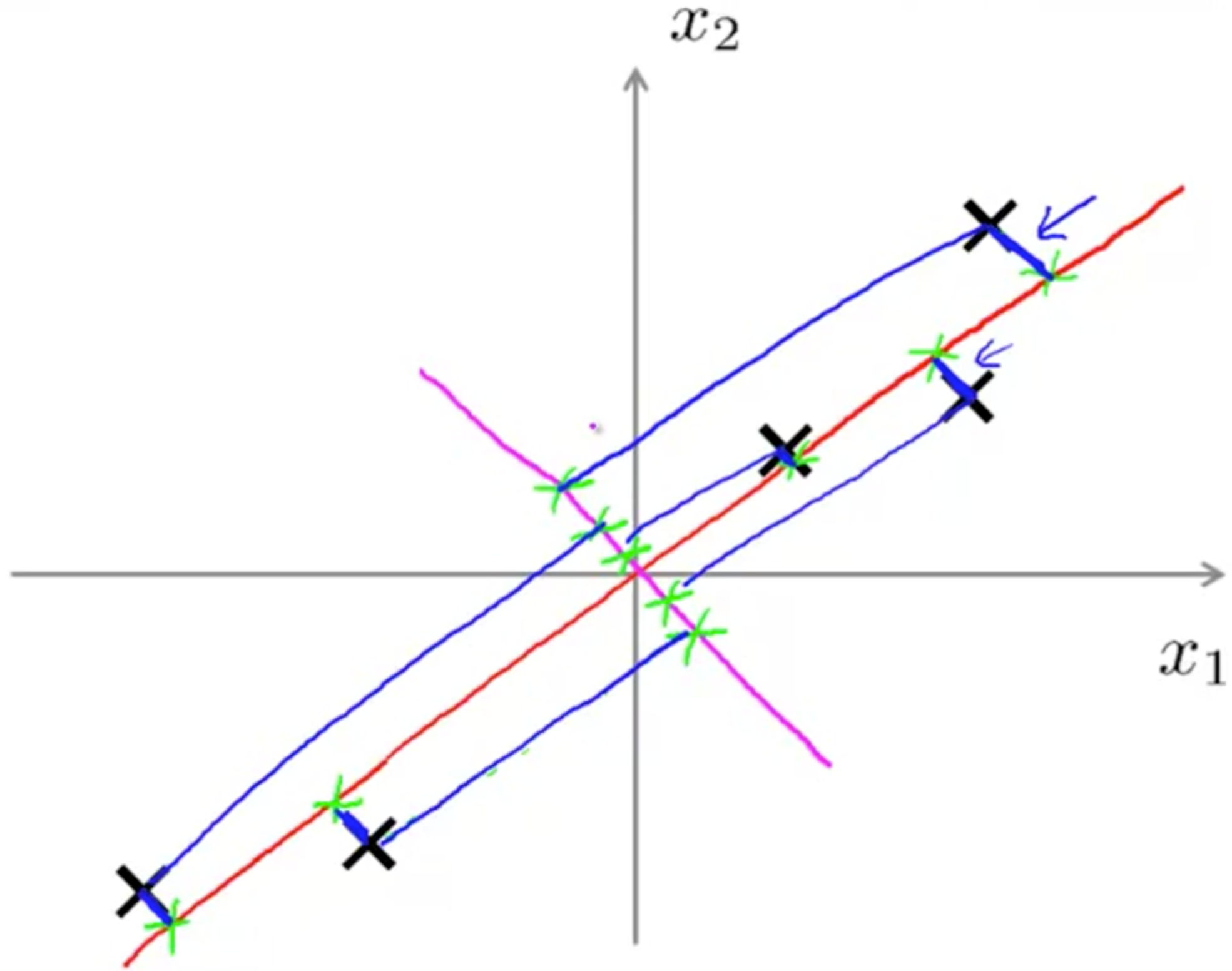
Principal Component Analysis (PCA) problem formulation



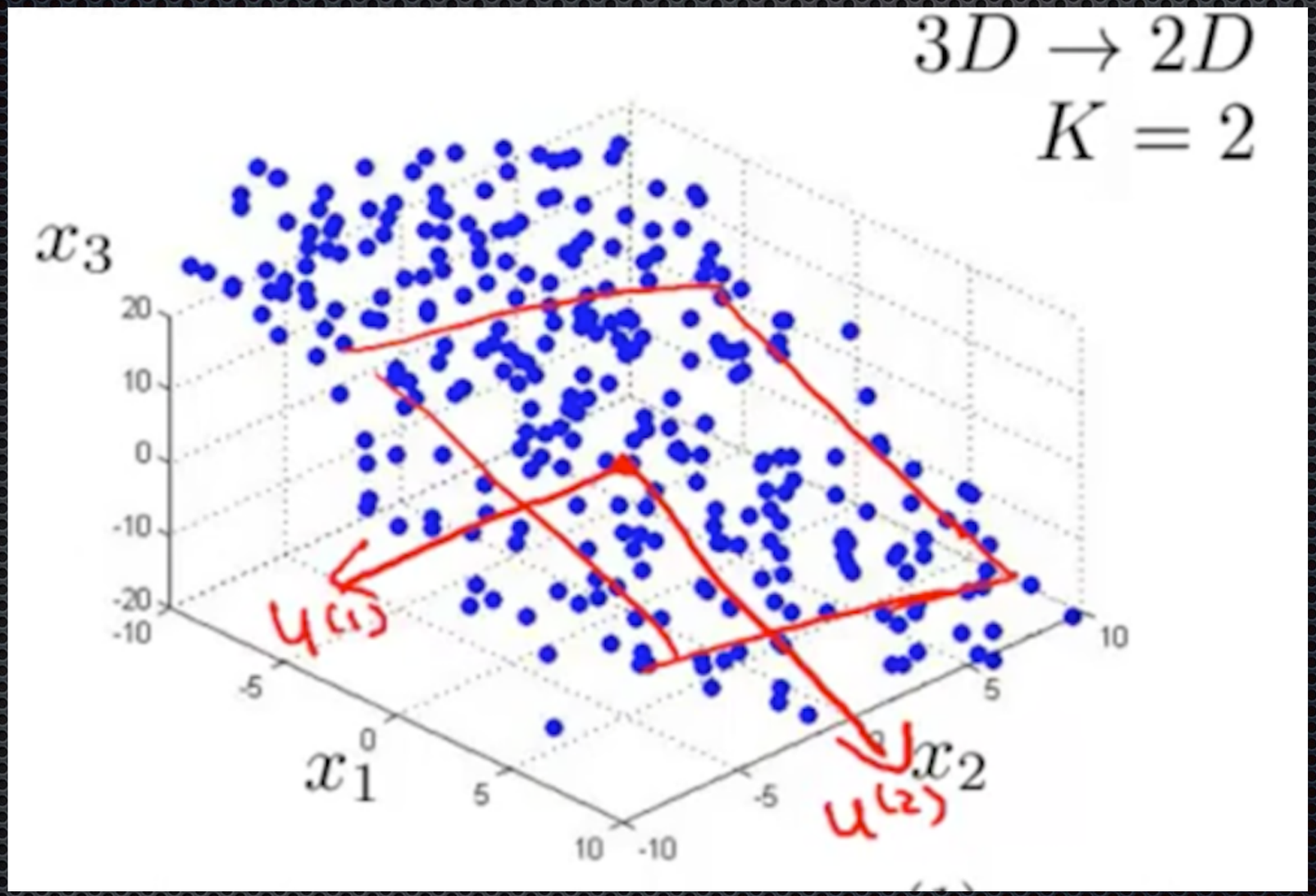
Principal Component Analysis (PCA) problem formulation



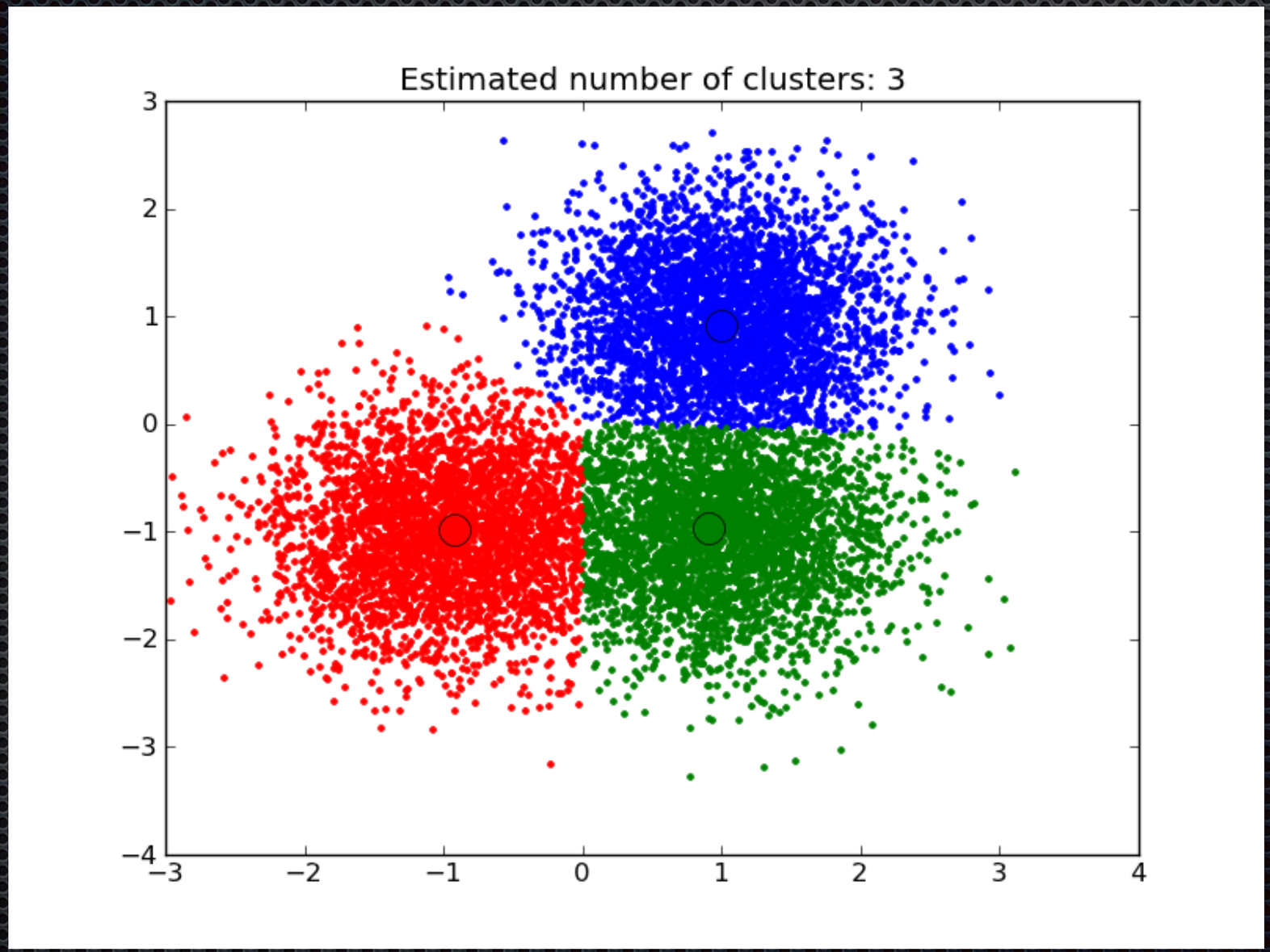
Principal Component Analysis (PCA) problem formulation



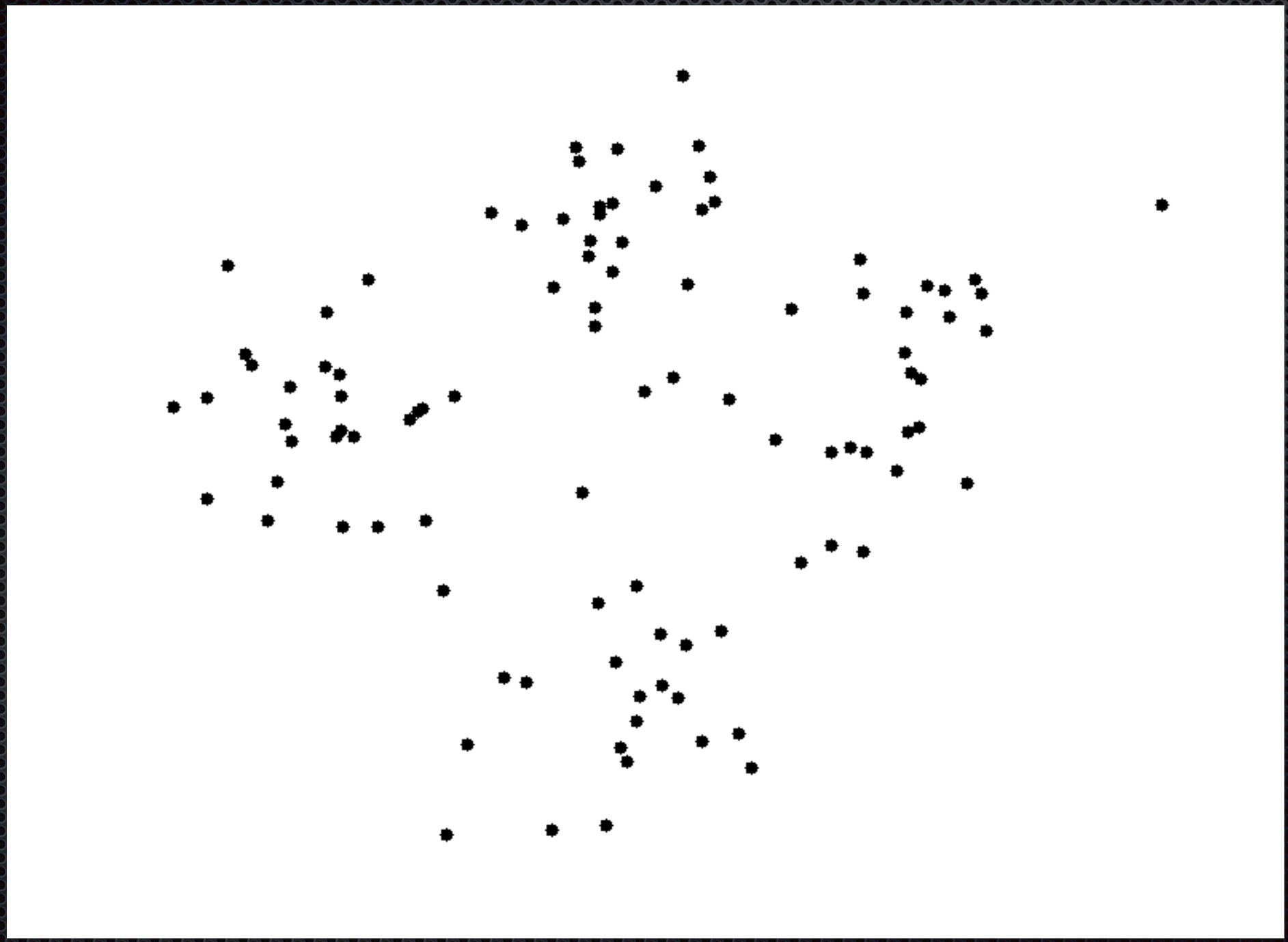
We can generalise any transformation from n initial features to k final features



CLUSTERING



K-Means clustering



K-Means Algorithm

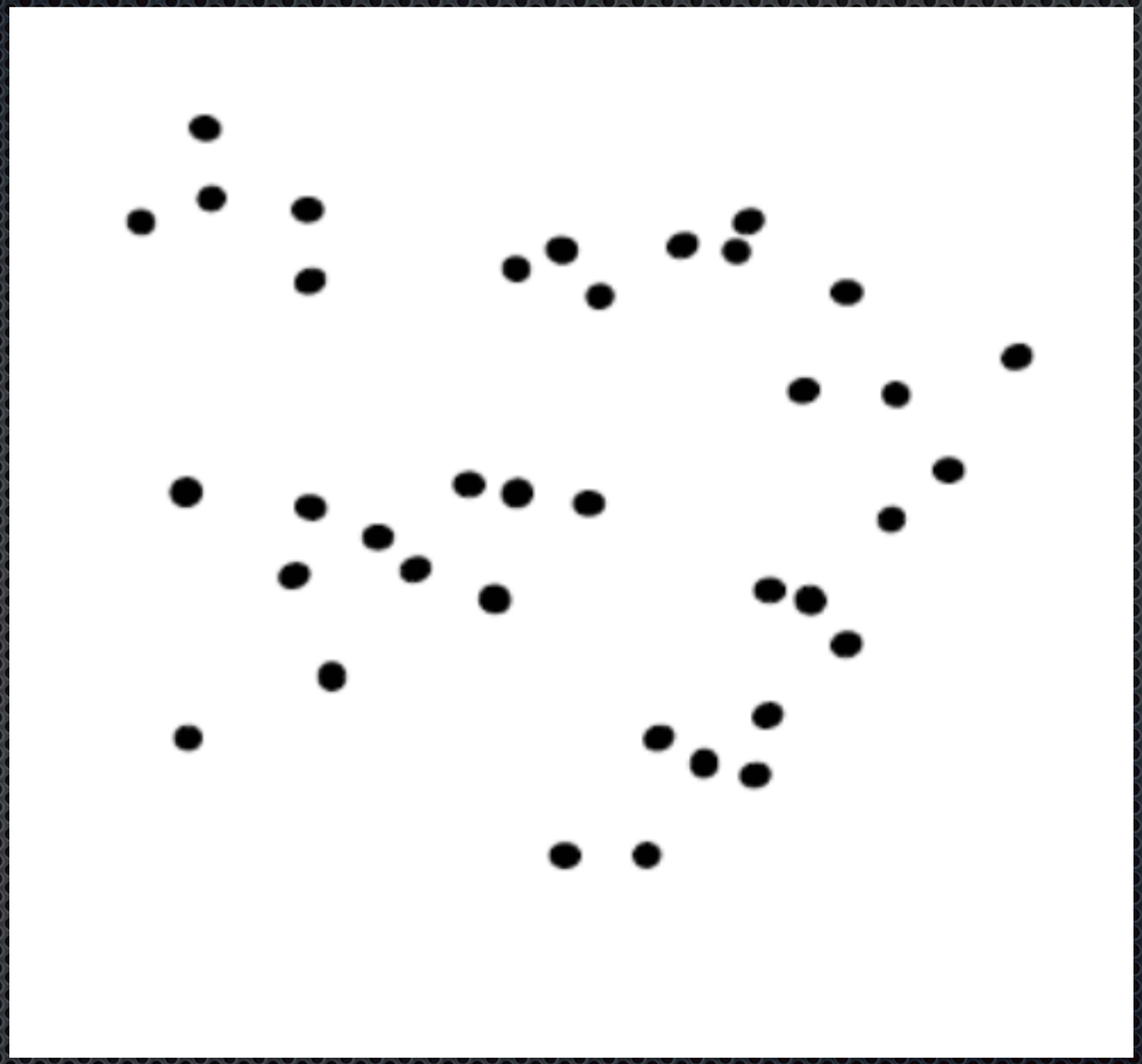
1. Initialize cluster centres.
2. Assign each point to the closest center.
3. Update cluster centers as the mean of the points
4. Repeat 2-3 until no points are re-assigned (t=t+1)

K-means converges to a local minimum

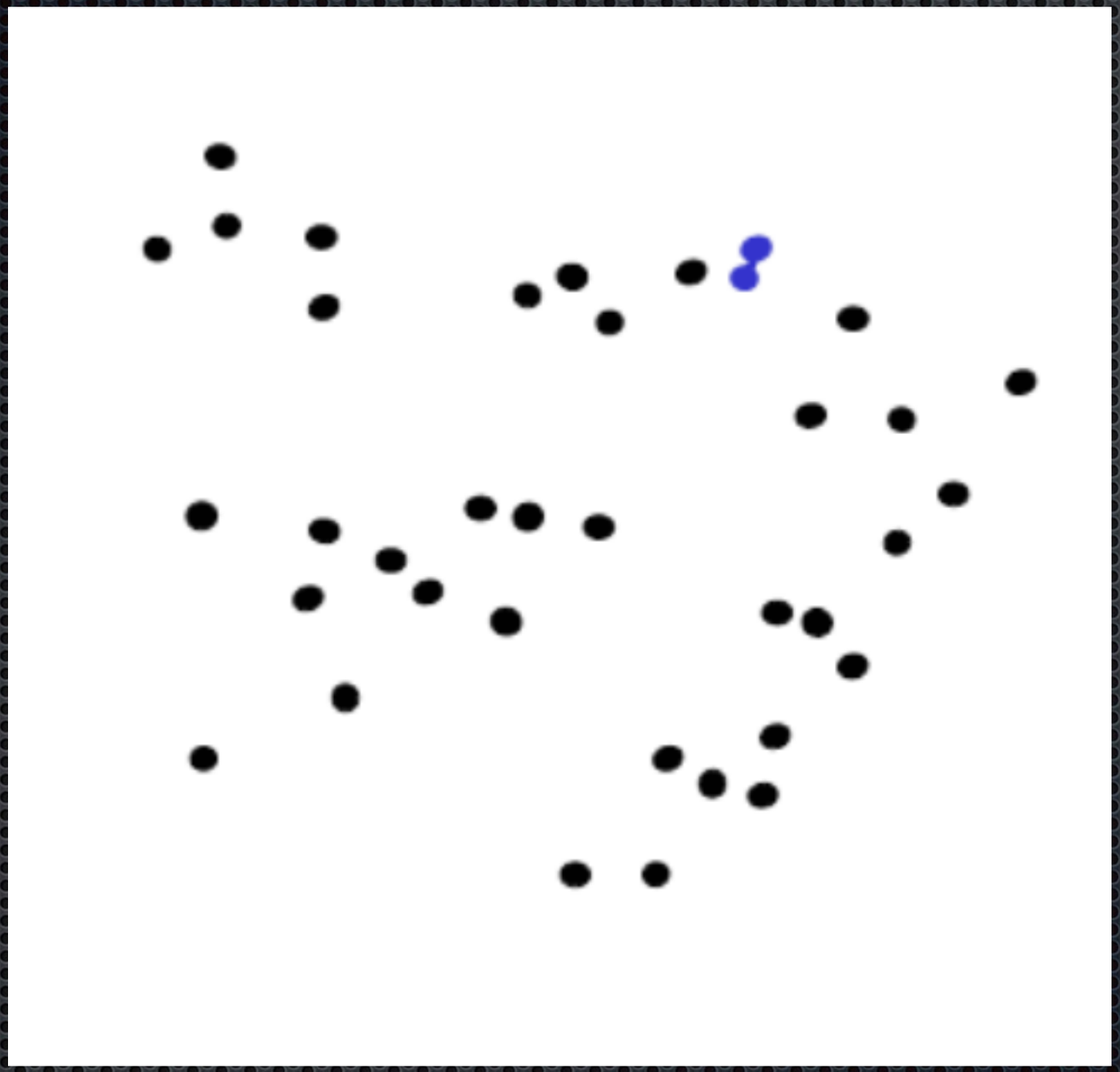
- Discussion

Agglomerative clustering

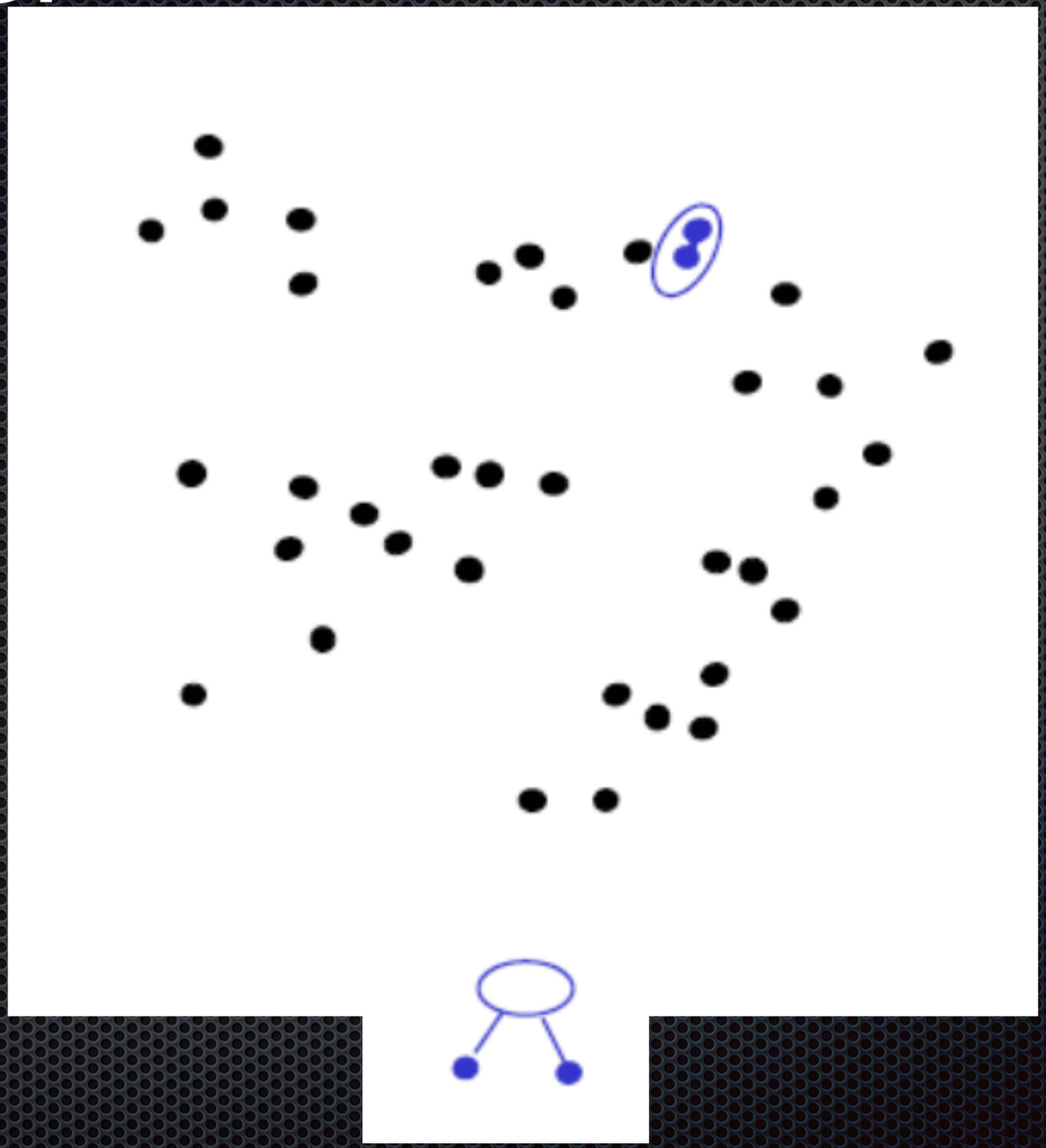
- Start with each point as its own cluster



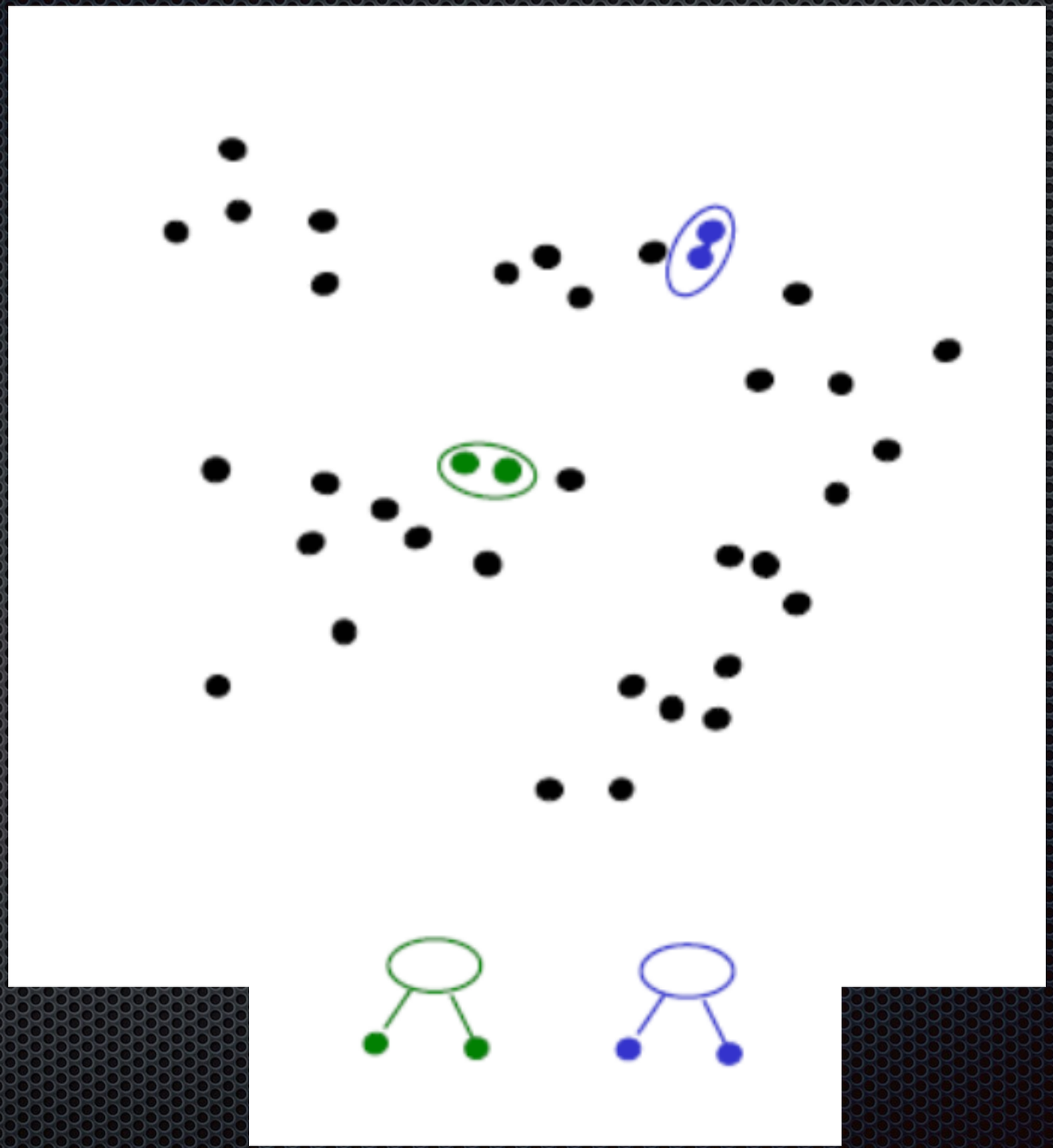
Find the most similar pair of clusters



Merge them into a parent cluster



Repeat



Terminate when you have the required number of clusters

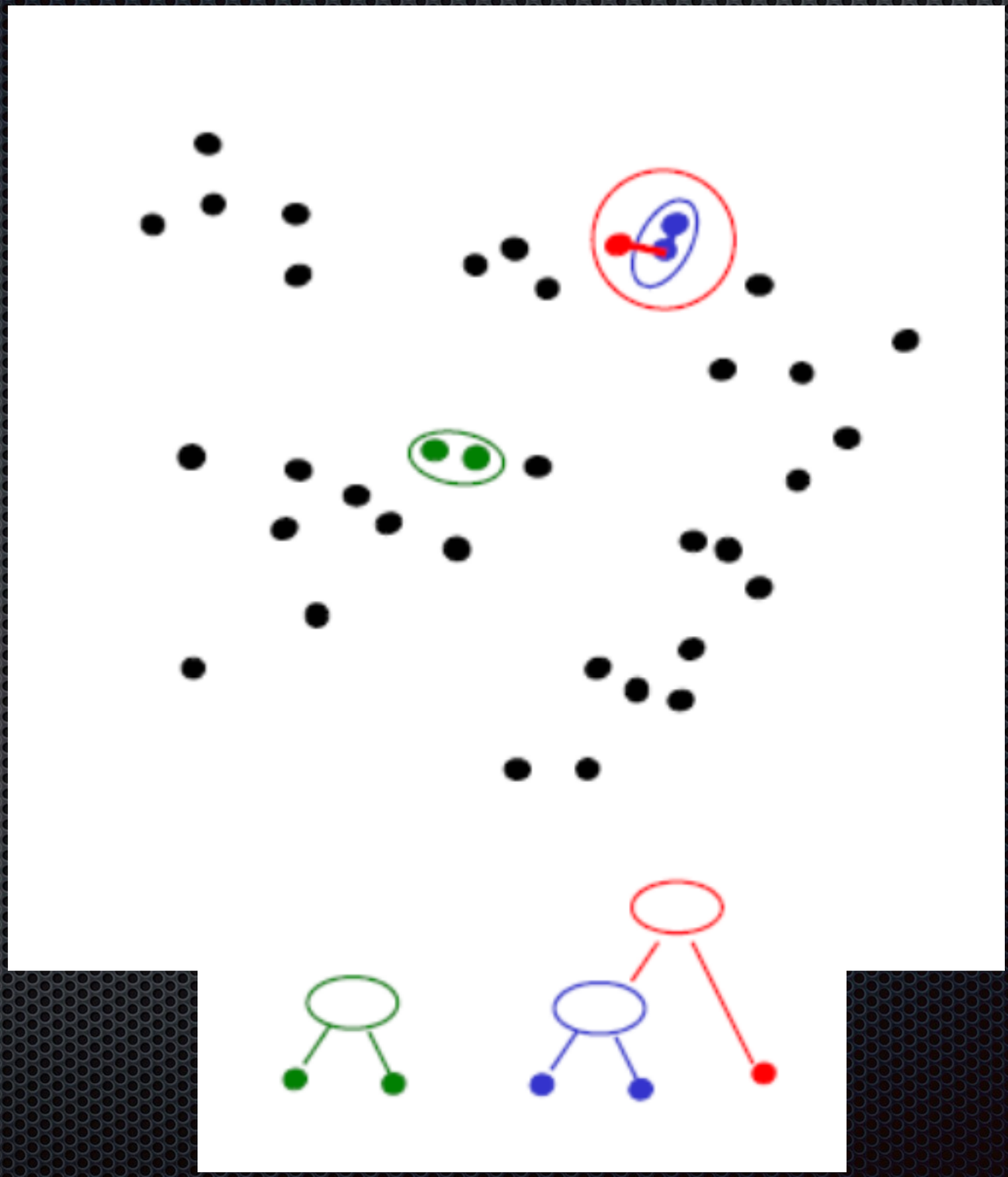
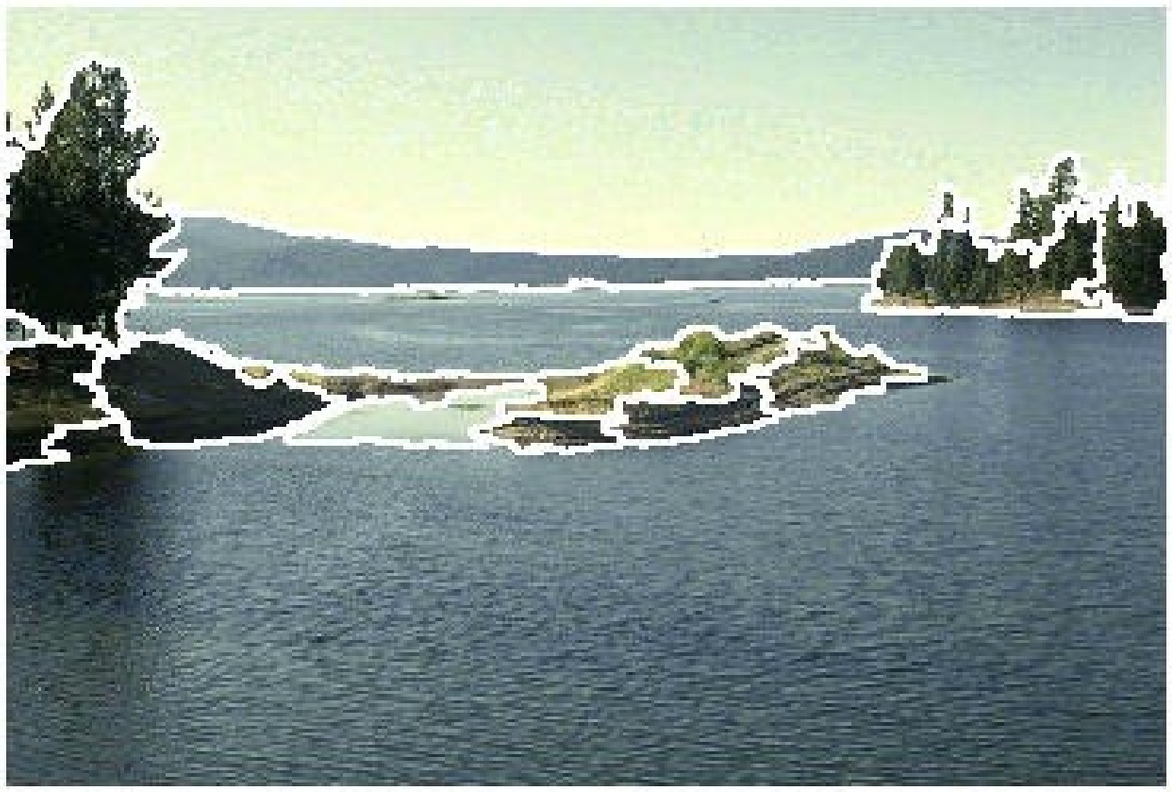
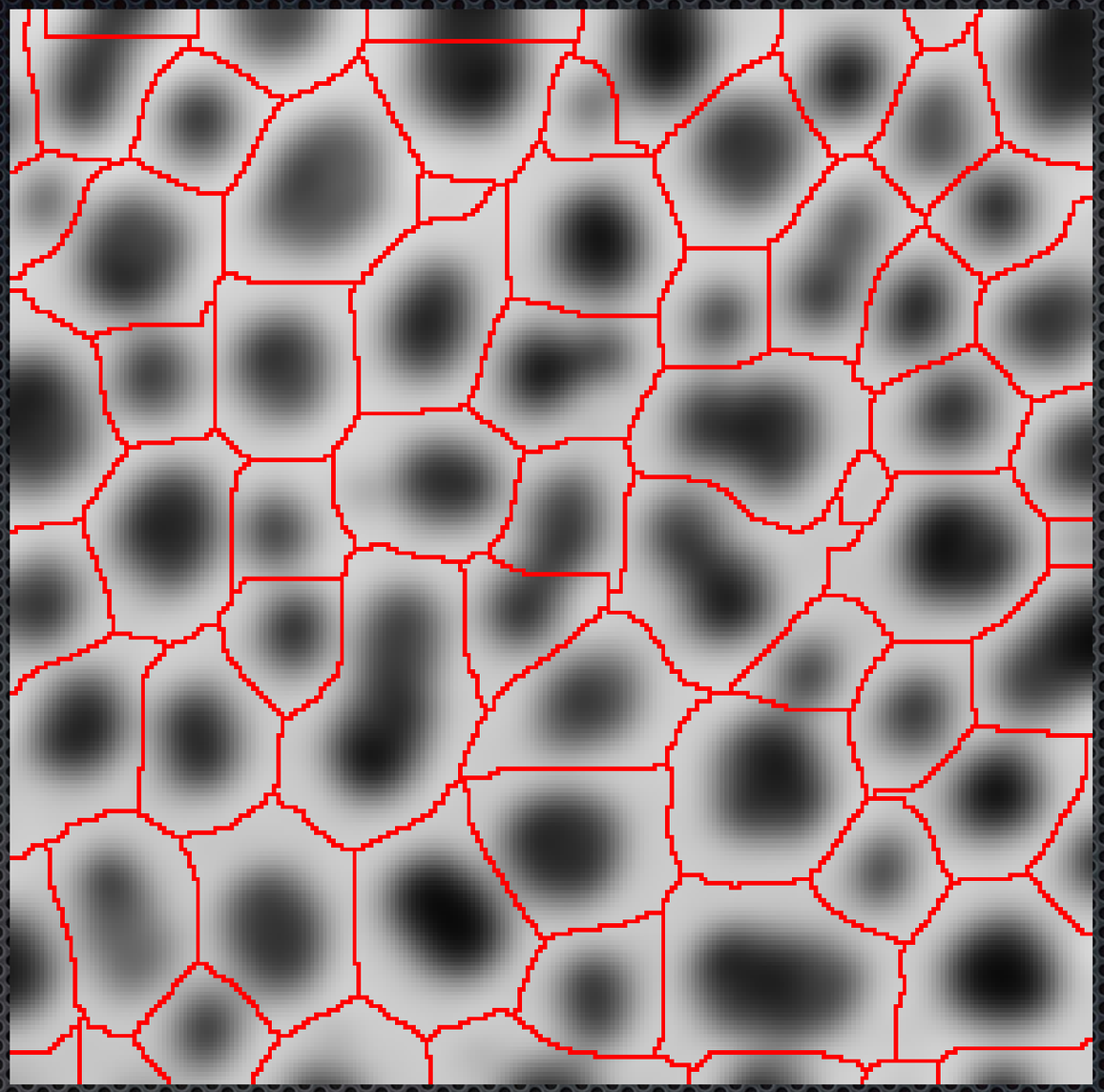
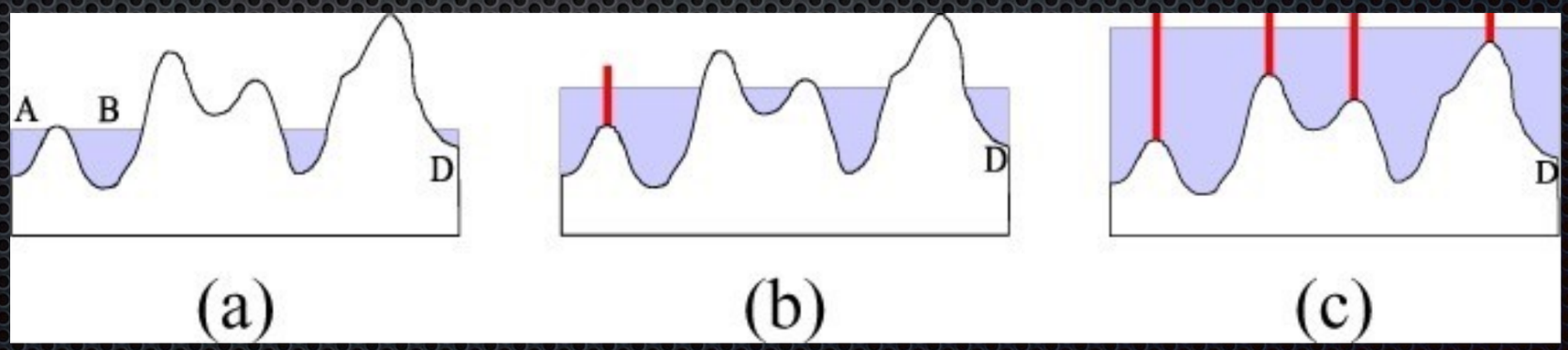


Image segmentation

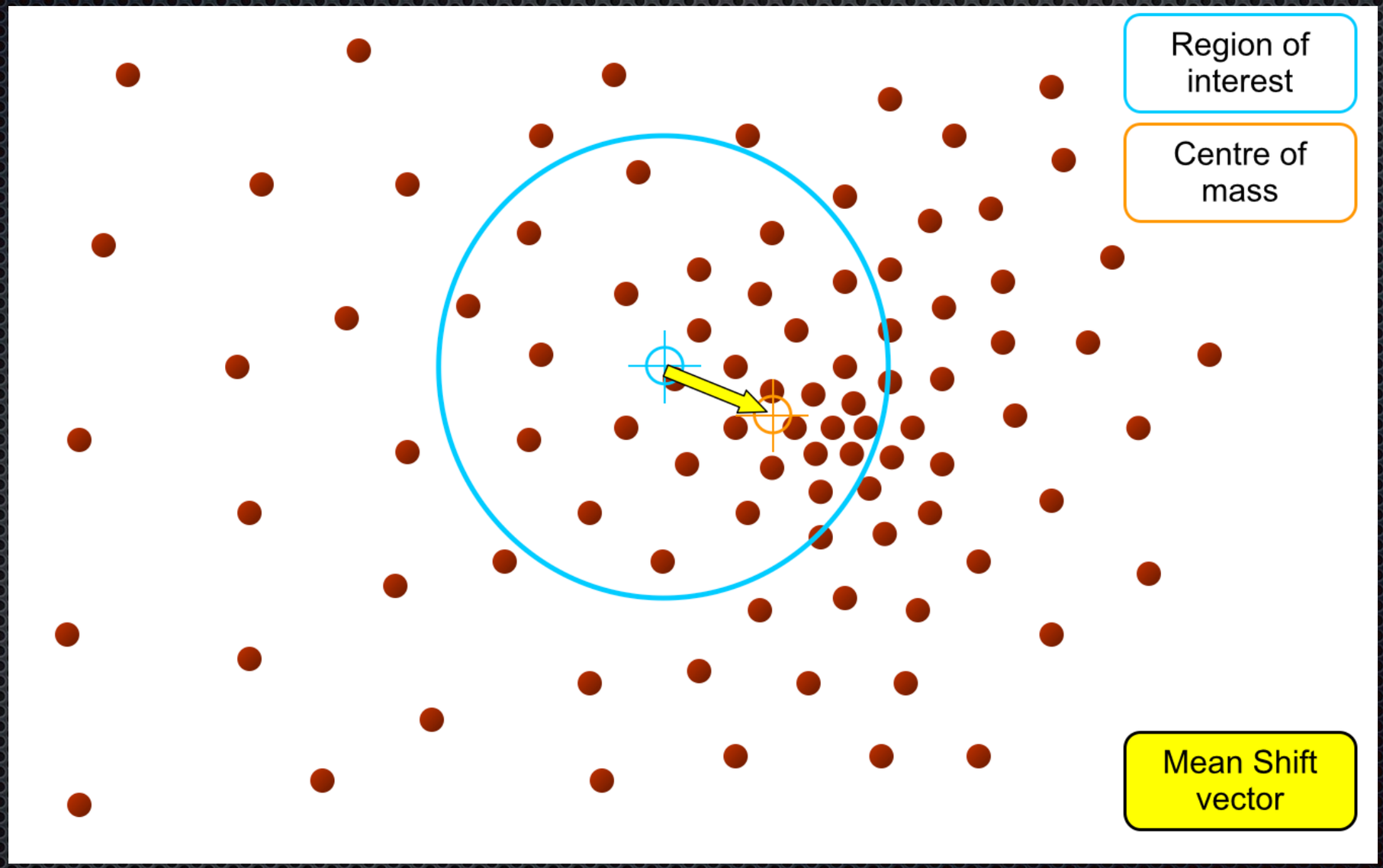
Segmented "landscape 1"	Segmented "landscape 2"
	

Watershed Algorithm

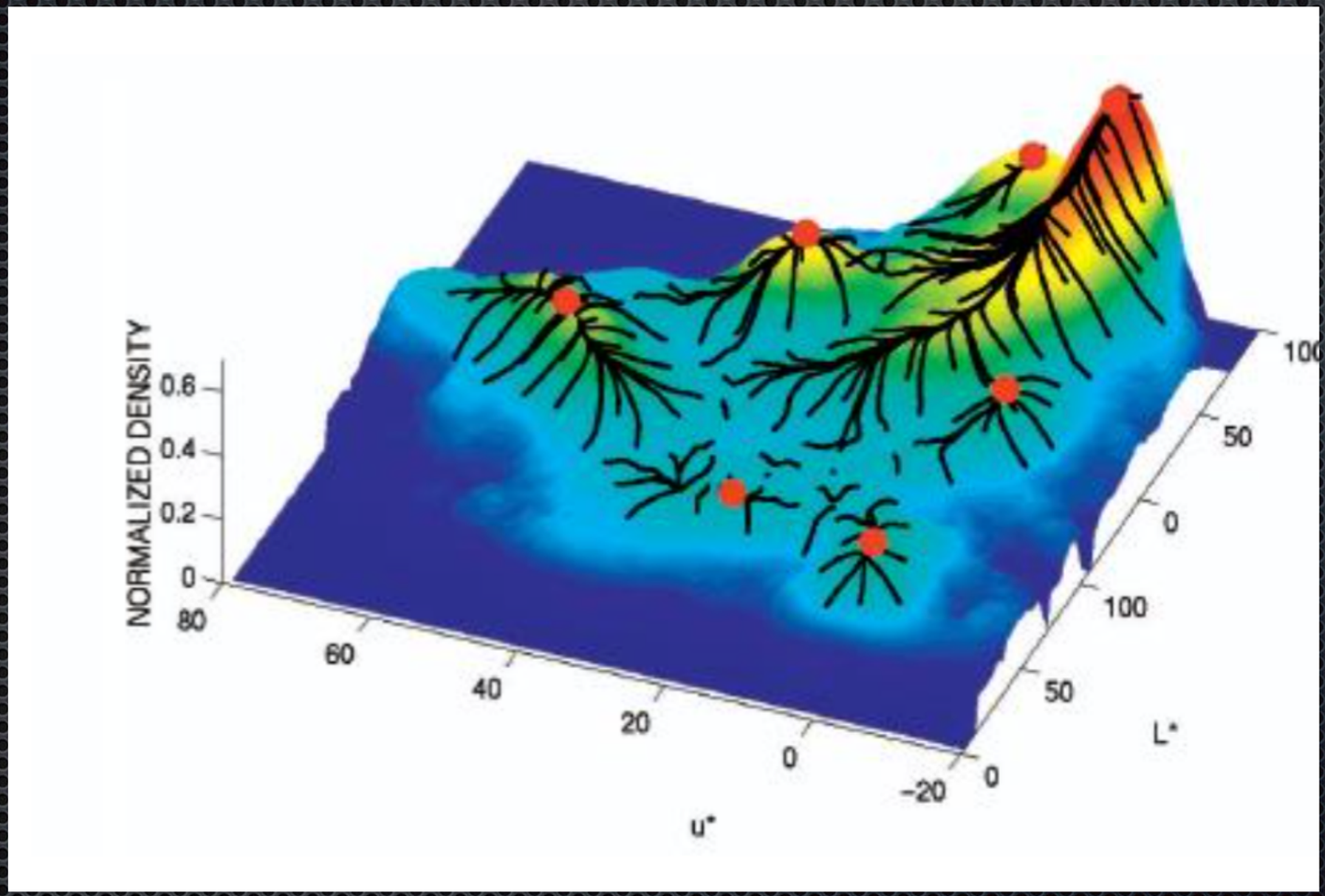




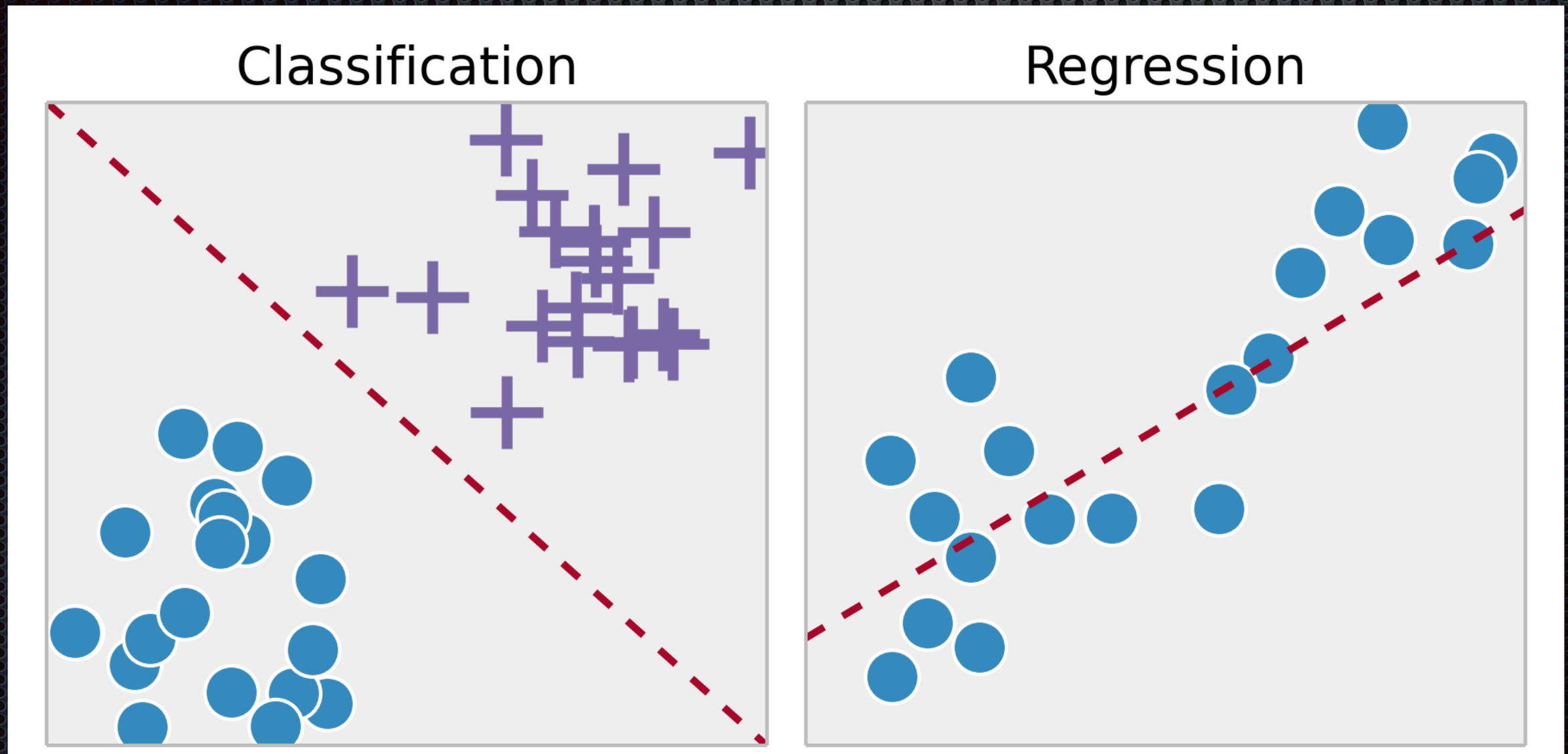
How to find an extremum ?
Mean shift - intuition



A ridges perspective



Regression intro, discussion



- Octave introduction

Everything unexplored is a purpose.

Thank you