The Curse of Dimensionality

Stefano Cavuoti

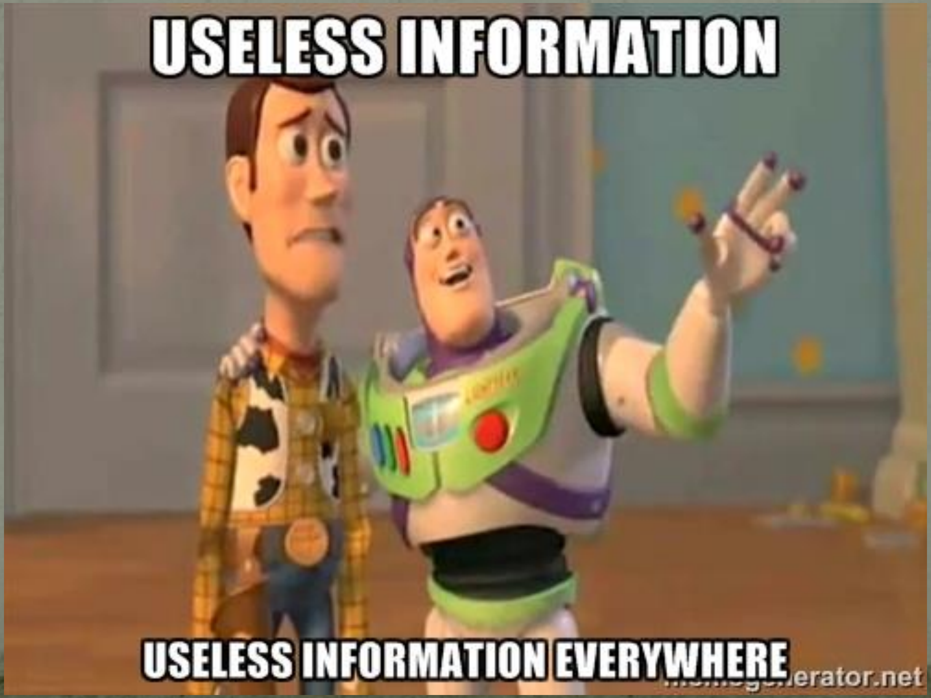
INAF – Capodimonte Astronomical Observatory – Napoli

Redundancy of parameter space

By definition, machine learning models are based on learning and self-adaptive techniques.

A priori, real world data are intrinsically carriers of embedded information, hidden by noise.

In almost all cases the signal-to-noise (S/N) ratio is low and the amount of data prevents the human exploration. We don't know which feature of a pattern is much carrier of good information and how and where the correlation of features gives the best knowledge of the solution.



Examples of BoK – wine classification

If we want to classify the cultivars from which a bottle of wine is made we could analyze a lot of parameters.

- Alcohol
- Malic acid
- Ash (cenere)
- Alcalinity of ash
- Magnesium
- Total phenols
- Flavanoids
- Nonflavanoid phenols
- Proanthocyanins
- Color intensity
- Hue
- OD280/OD315 of diluted wines
- Proline
- …
- Date of production

The date of production is not related to the kind of cultivars and for this task this parameter is useless or could be even harmful!

Curse of dimensionality

The following example is extracted from:
http://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/

Given a set of images, each one depicting either a cat or a dog. We would like to create a classifier able to distinguish dogs from cats automatically.
To do so, we first need to think about a descriptor for each object class that can be expressed by numbers, such that a mathematical algorithm can use those numbers to recognize objects. We could for instance argue that cats and dogs generally differ in color. A possible descriptor discriminating these two classes could then consist of three numbers: the average red, green and blue colors of the images.
A simple linear classifier for instance, could linearly combine these features to decide on the class labels

```
If $0.5*red + 0.3*green + 0.2*blue > 0.6$ :
        return cat;
else:
        return dog;
```

Curse of dimensionality

However, these three color-describing numbers, called features, will obviously not suffice to obtain a perfect classification. Therefore, we could decide to add some features that describe the texture of the image, for instance by calculating the average edge or gradient intensity in both the X and Y direction. We now have 5 features that, in combination, could possibly be used by a classification algorithm to distinguish cats from dogs.

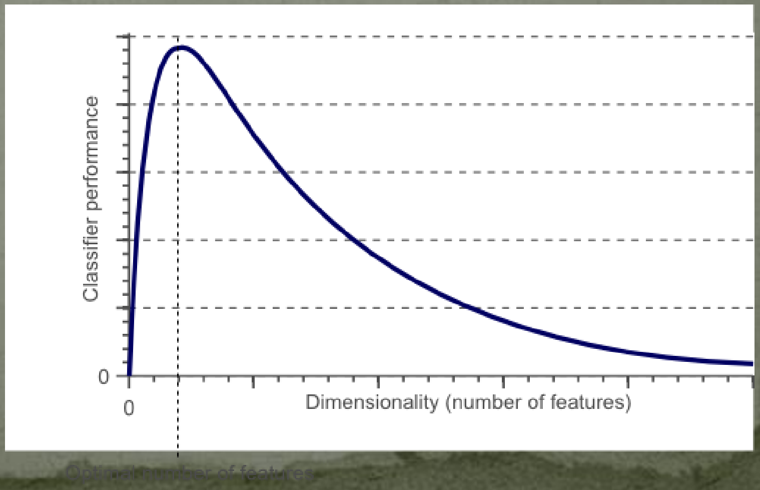
To obtain an even more accurate classification, we could add more features, based on color or texture histograms, statistical moments, etc.

Maybe we can obtain a perfect classification by carefully defining a few hundred of these features?

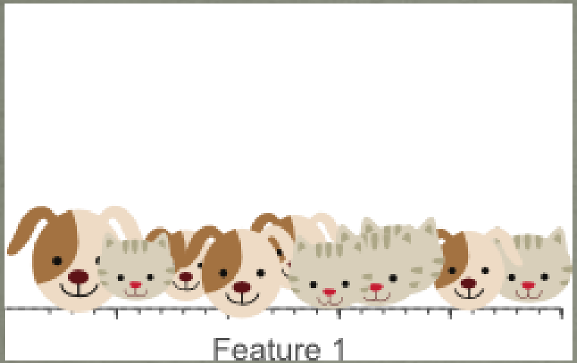
The answer to this question might sound a bit counter-intuitive: *no, we can't!*.

In fact, after a certain point, increasing the dimensionality of the problem, by adding new features, would actually degrade the performance of our classifier.

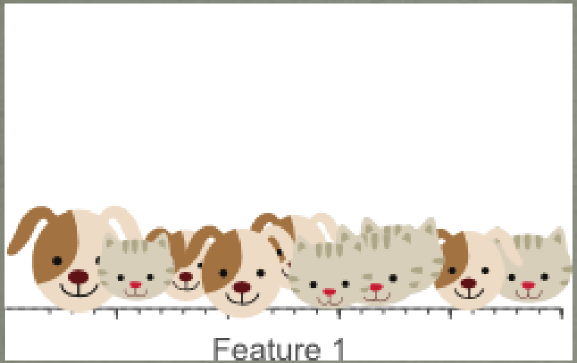
This is illustrated by figure, and is often referred to as 'The Curse of Dimensionality'.

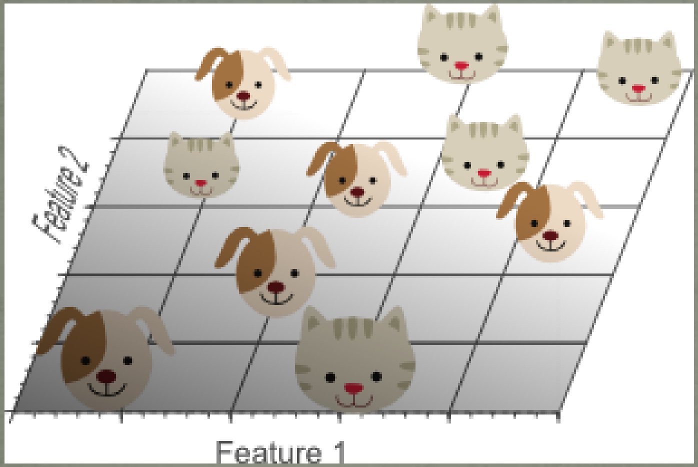


Curse of dimensionality

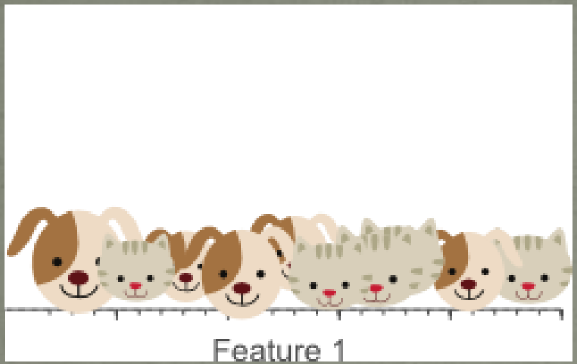


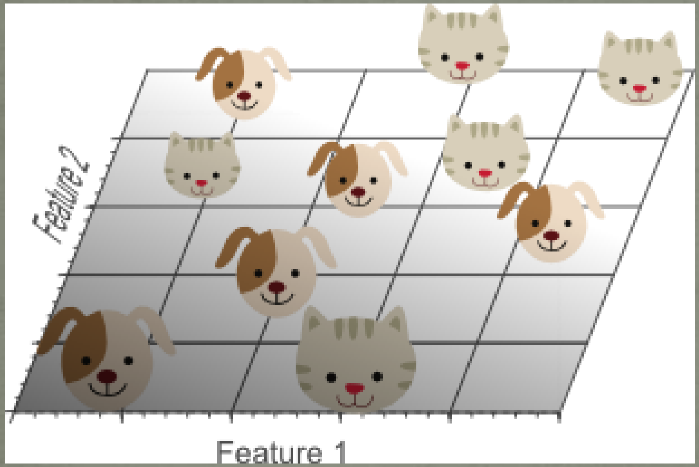
Curse of dimensionality

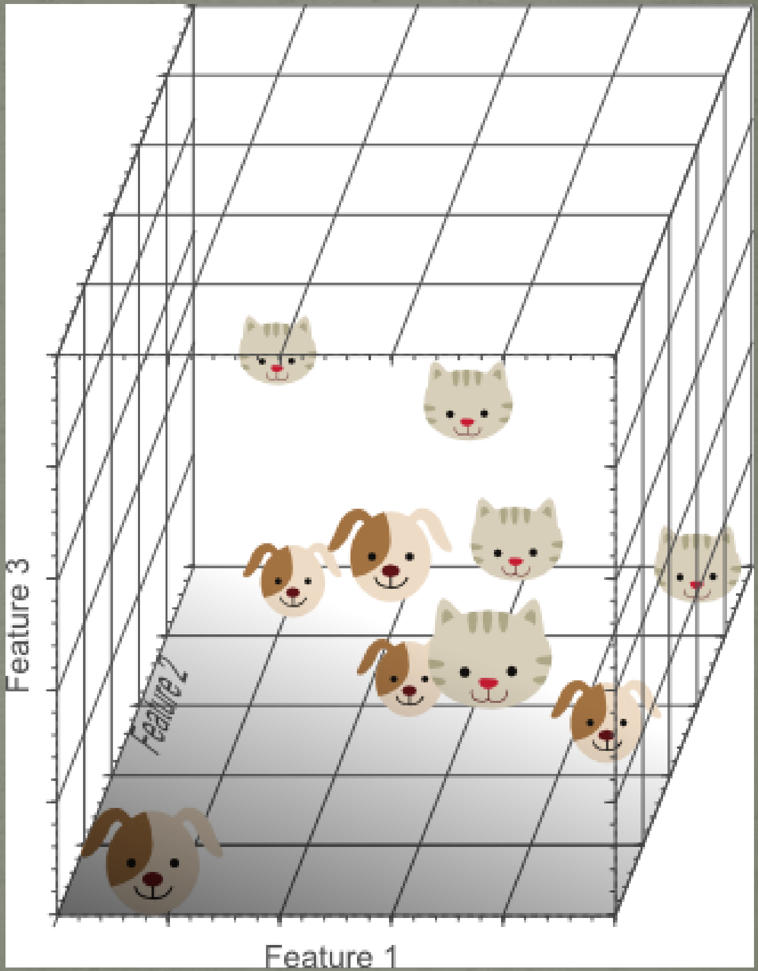




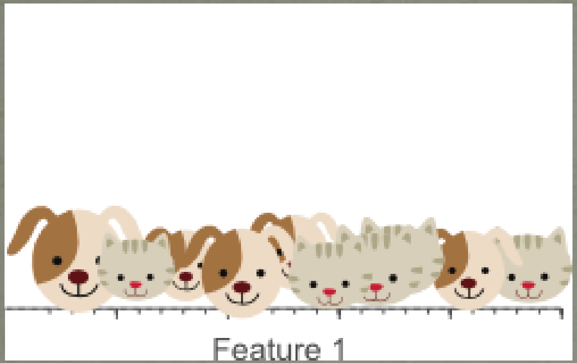
Curse of dimensionality

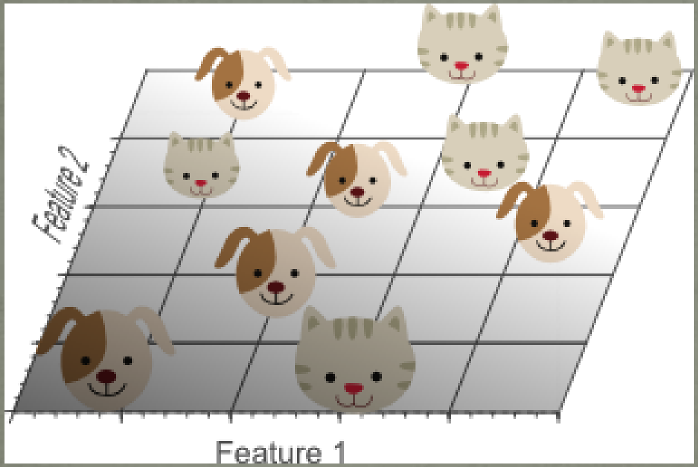


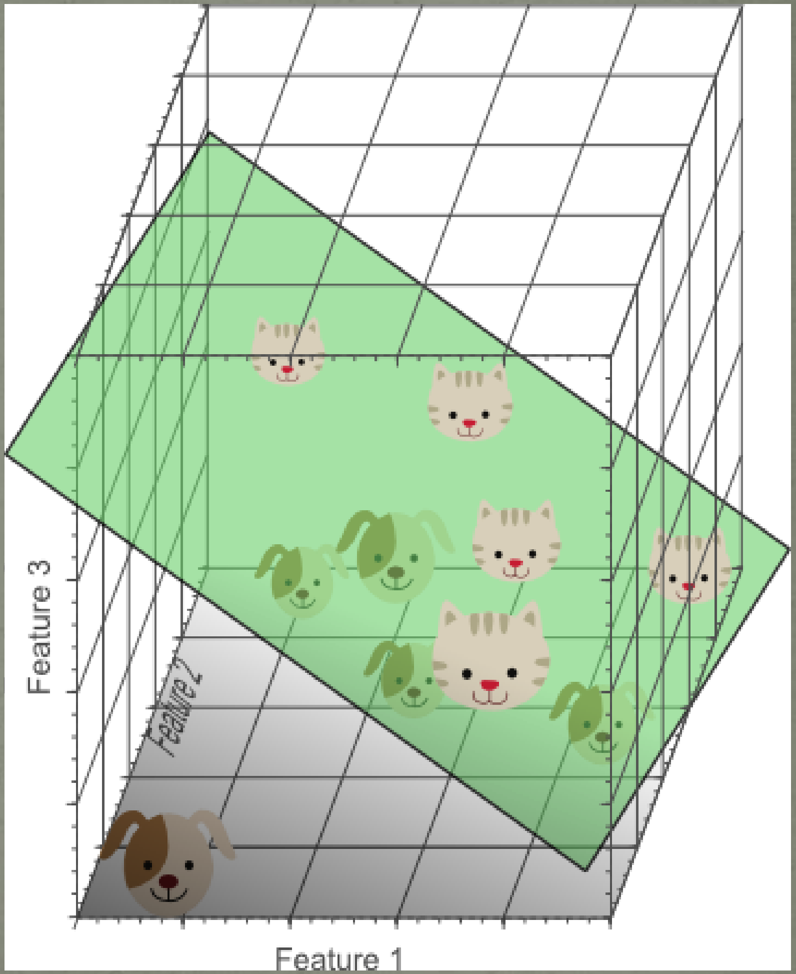




Curse of dimensionality





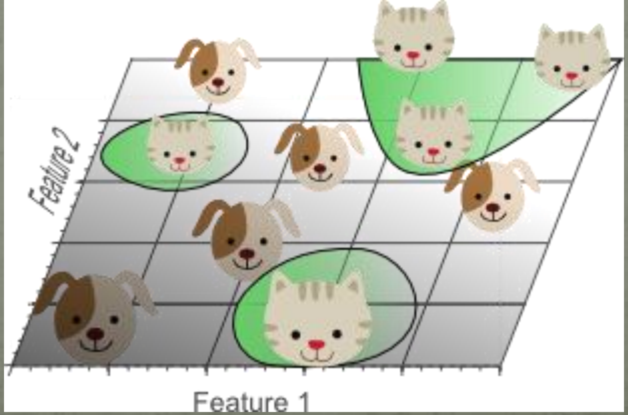


Curse of dimensionality

If we would keep adding features, the dimensionality of the feature space grows, and becomes sparser and sparser.

Due to this sparsity, it becomes much more easy to find a separable hyperplane because the likelihood that a training sample lies on the wrong side of the best hyperplane becomes infinitely small when the number of features becomes infinitely large.

However, if we project the highly dimensional classification result back to a lower dimensional space, we find this:



Curse of dimensionality

Imagine a unit square that represents the 2D feature space.
The average of the feature space is the center of this unit square, and all points within unit distance from this center, are inside a unit circle that inscribes the unit square.
The training samples that do not fall within this unit circle are closer to the corners of the search space than to its center.

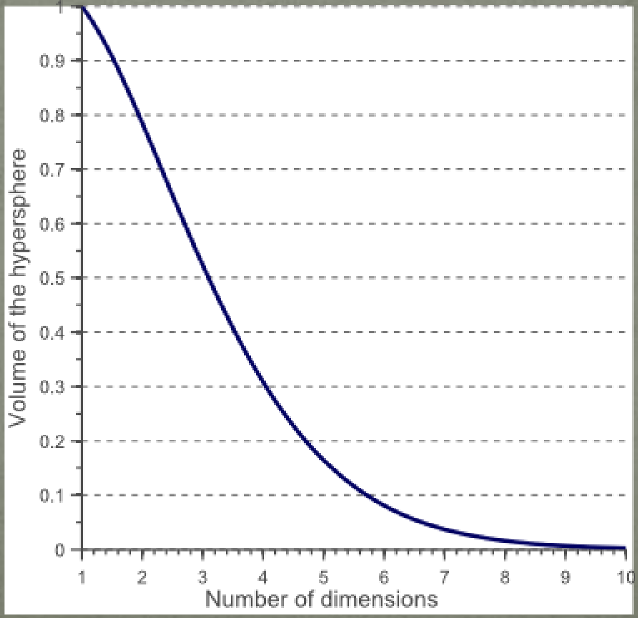
These samples are difficult to classify because their feature values greatly differ (e.g. samples in opposite corners of the unit square). Therefore, classification is easier if most samples fall inside the inscribed unit circle

Curse of dimensionality

The volume of the inscribing hypersphere of dimension d and with radius 0.5 can be calculated as:

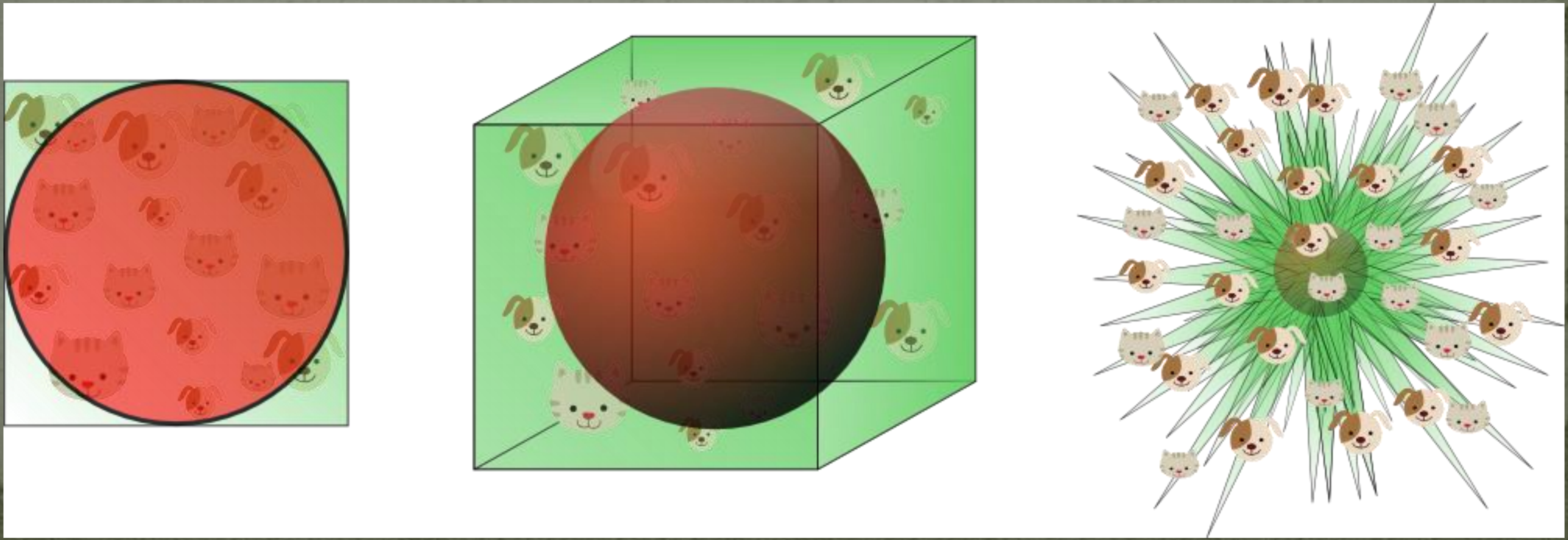
$$V(d) = \frac{\pi^{d/2}}{\Gamma(\frac{d}{2}+1)} 0.5^d.$$

Figure shows how the volume of this hypersphere changes when the dimensionality increases:

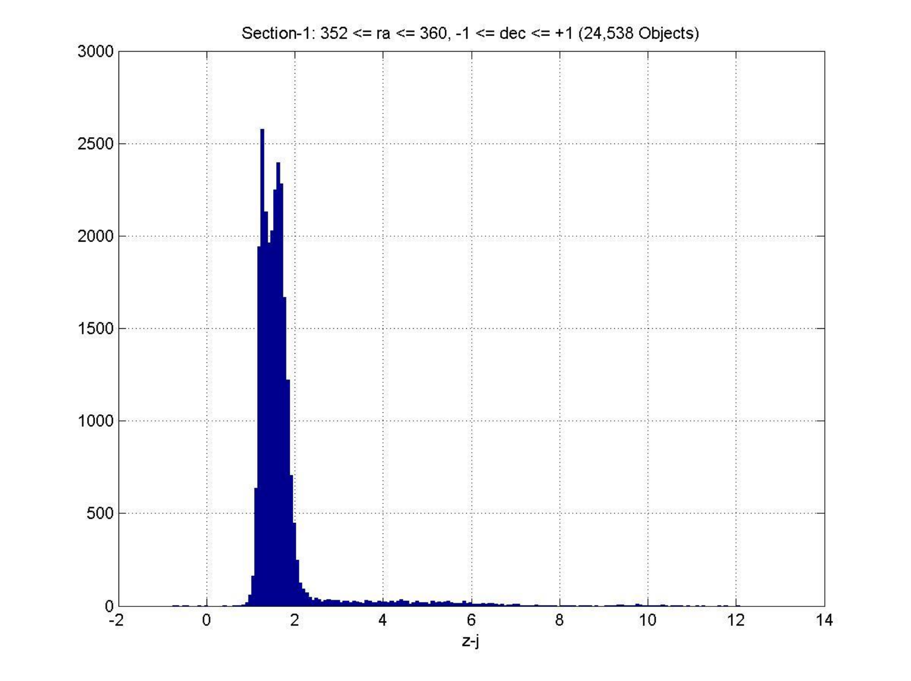


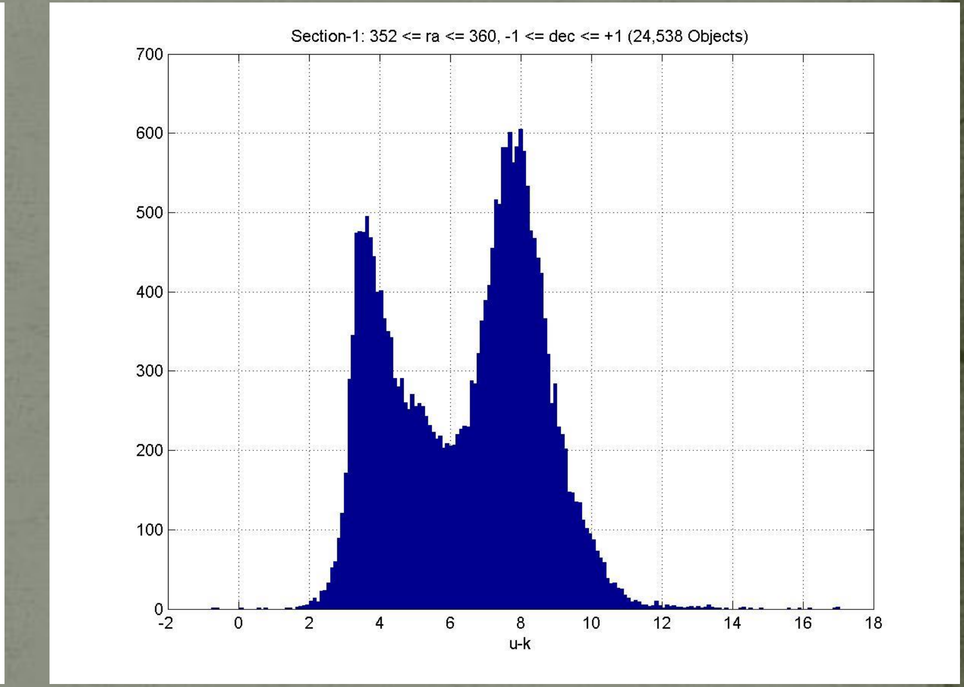
Curse of dimensionality

The volume of the hypersphere tends to zero as the dimensionality tends to infinity, whereas the volume of the surrounding hypercube remains constant. This surprising and rather counter-intuitive observation partially explains the problems associated with the curse of dimensionality in classification: In high dimensional spaces, most of the training data resides in the corners of the hypercube defining the feature space. As mentioned before, instances in the corners of the feature space are much more difficult to classify than instances around the centroid of the hypersphere. This is illustrated by figure 11, which shows a 2D unit square, a 3D unit cube, and a creative visualization of an 8D hypercube which has 2^8 = 256 corners



Examples of BoK – stars and galaxies



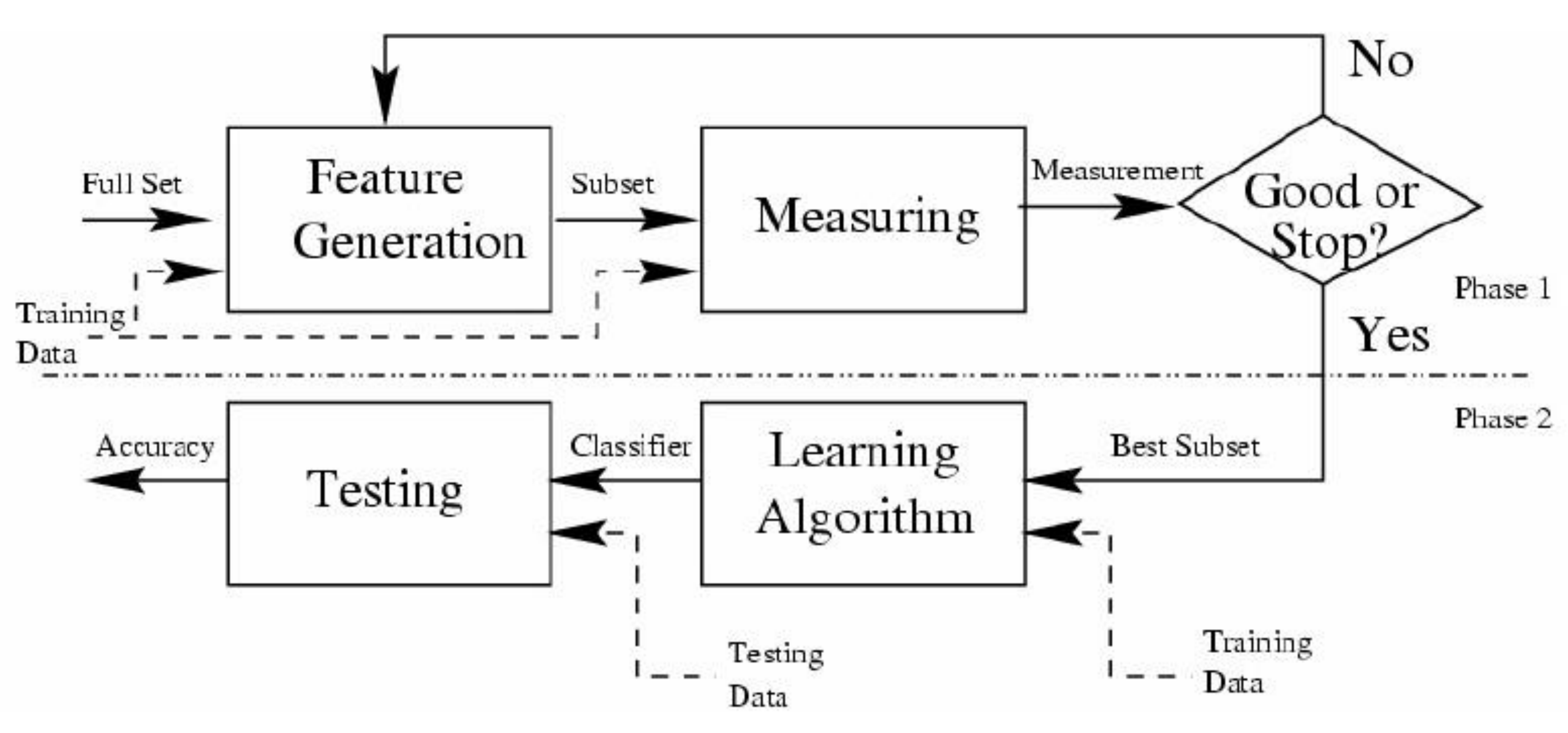


are not separated in this parameter

are separated in this parameter

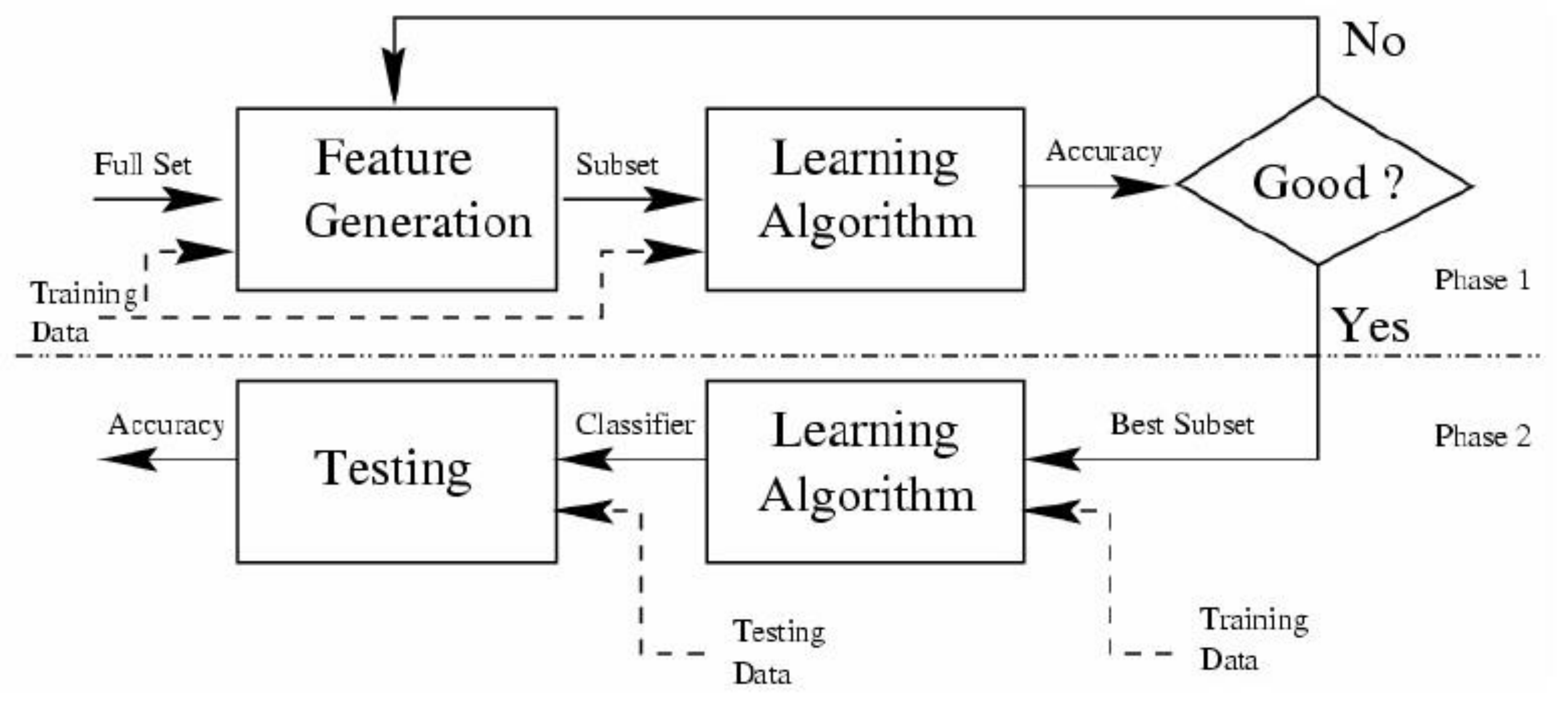
Feature extraction: Filter model

- Separating feature selection from classifier learning
- Relying on general characteristics of data (*information, distance*, *dependence*, *consistency*)
- No bias toward any learning algorithm, fast



Feature extraction: Wrapper model

- Relying on a predetermined classification algorithm
- Using predictive accuracy as goodness measure
- High accuracy, computationally expensive



Pruning: Wrapper model + statistics

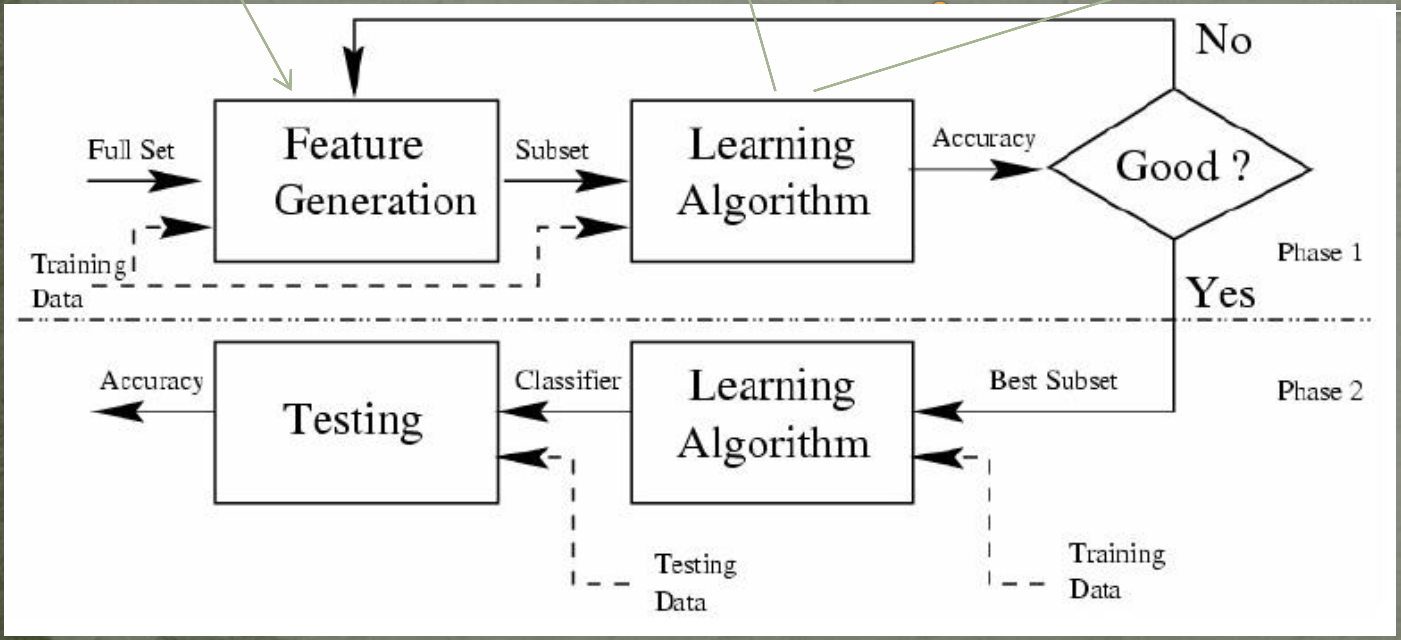
Pruning of data consists of an heuristic evaluation of quality performance of a machine learning technique, based on the Wrapper model of feature extraction, mixed with statistical indicators. It is basically used to optimize the parameter space in classification and regression problems.

Classification

- Confusion matrix
- Completeness
- Purity
- Contamination

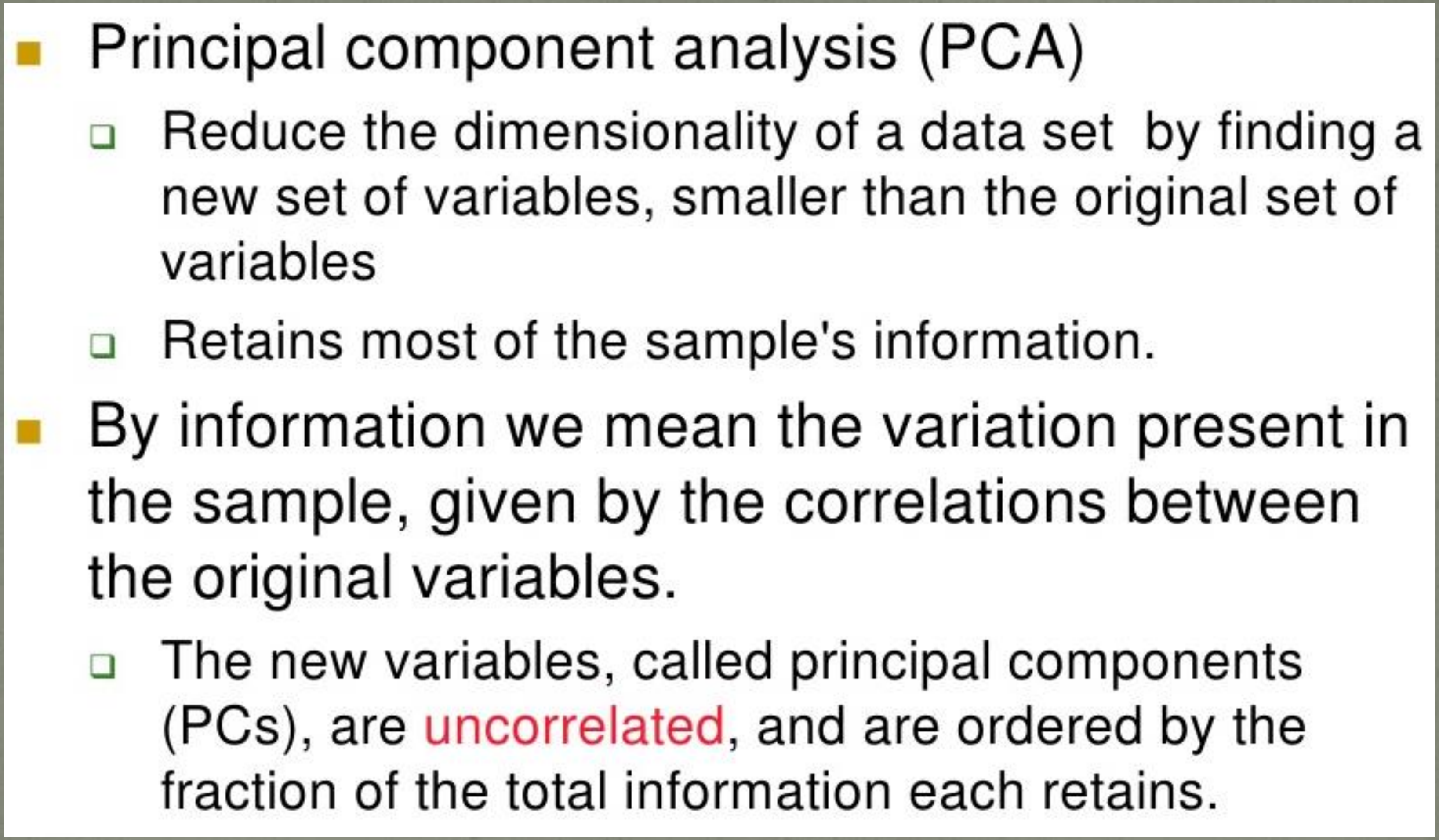
All permutations of data features

Regression



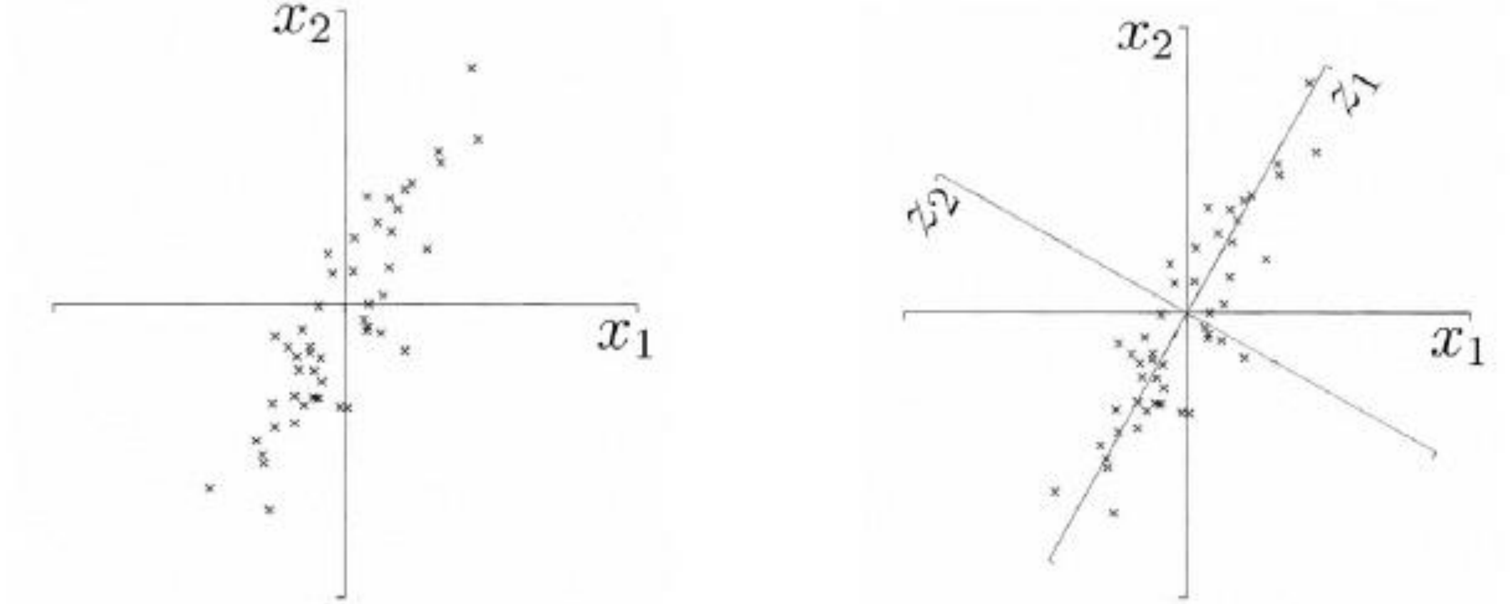
- Bias
- Standard Deviation
- MAD
- RMS
- Outlier percentages

Unsupervised Feature Extraction: PCA



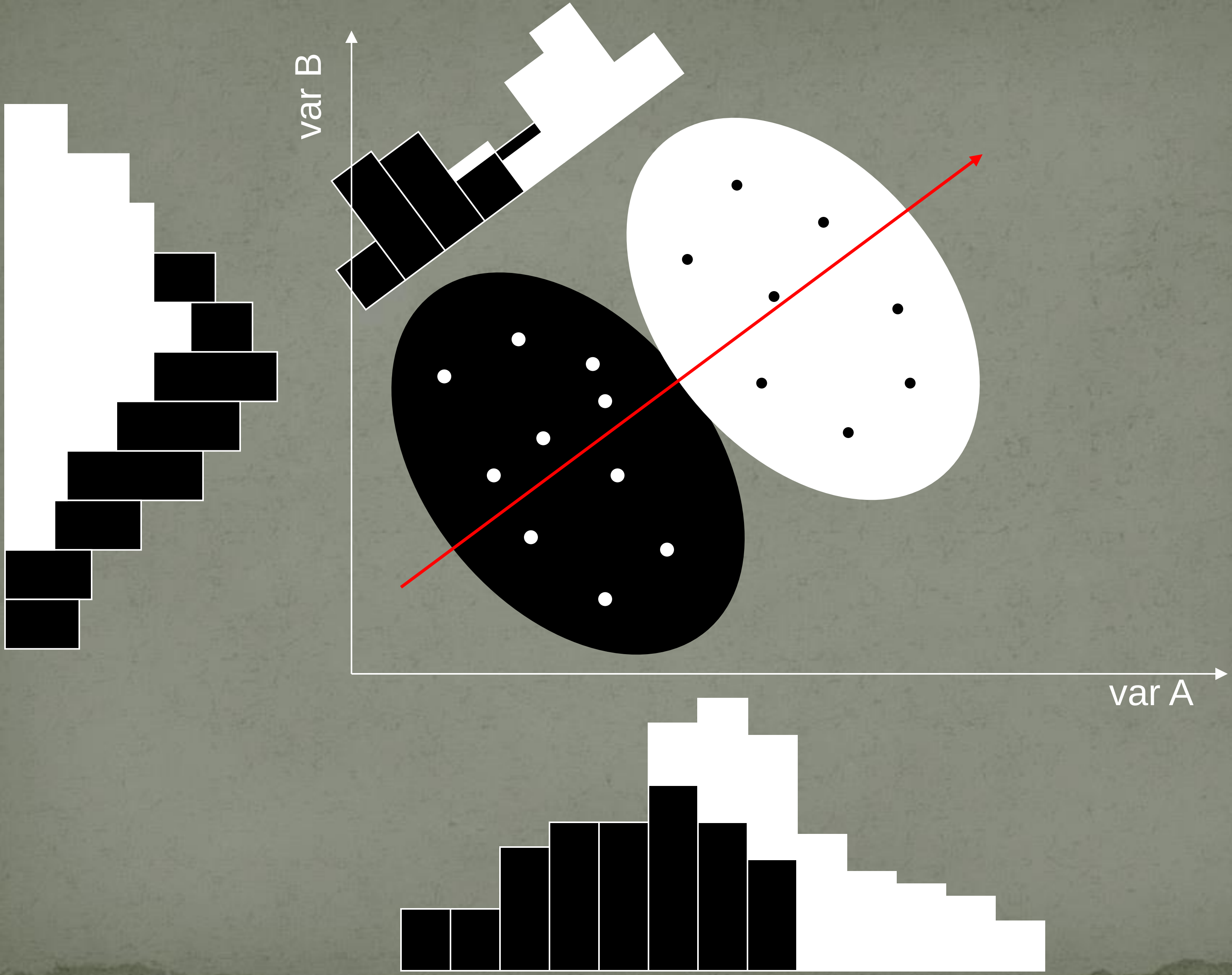
- Principal component analysis (PCA)
  - Reduce the dimensionality of a data set by finding a new set of variables, smaller than the original set of variables
  - Retains most of the sample's information.
- By information we mean the variation present in the sample, given by the correlations between the original variables.
  - The new variables, called principal components (PCs), are uncorrelated, and are ordered by the fraction of the total information each retains.

Geometric view of PCs



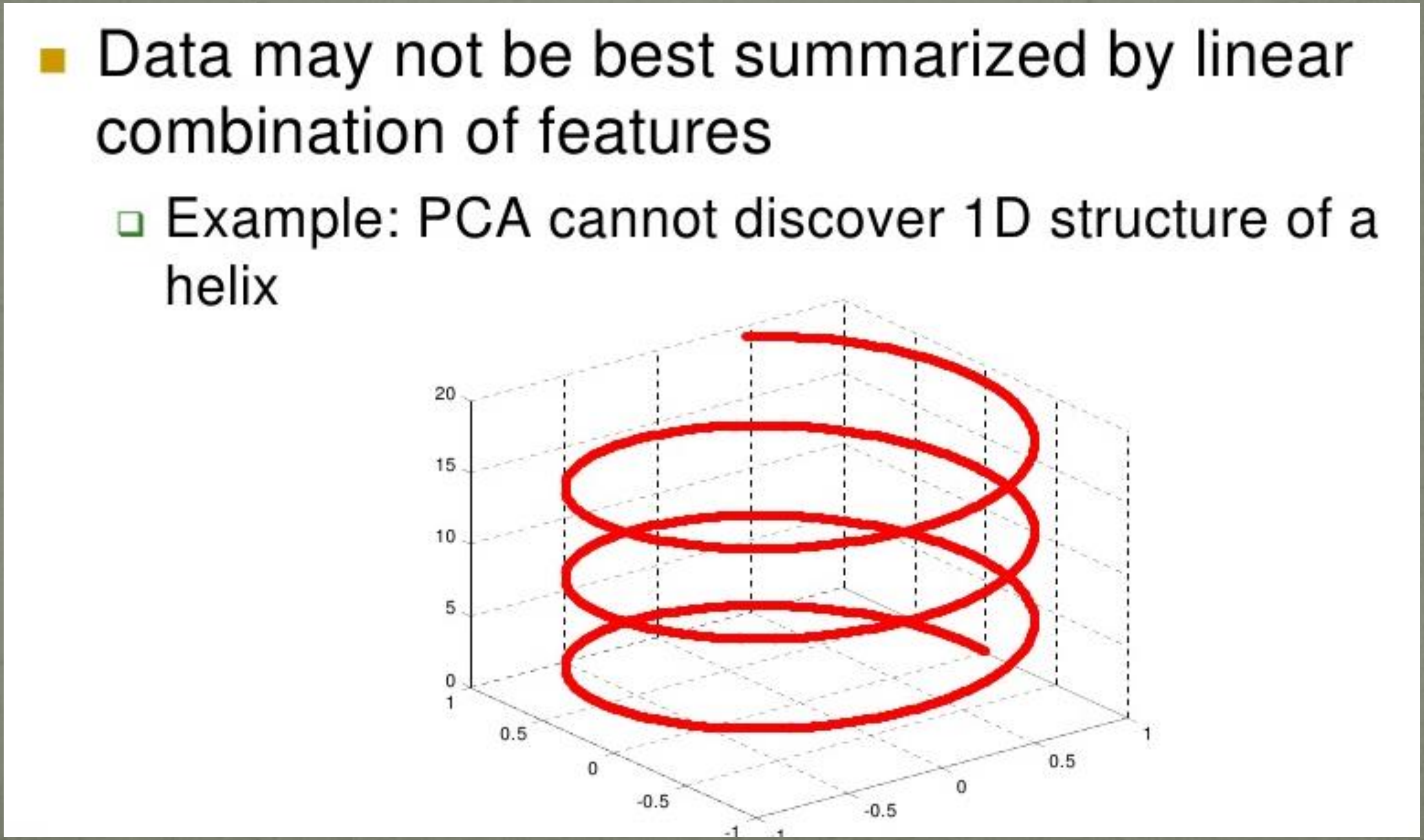
- the 1st PC z_1 is a minimum distance fit to a line in X space
- the 2nd PC z_2 is a minimum distance fit to a line in the plane perpendicular to the 1st PC

PCs are a series of linear least squares fits to a sample, each orthogonal to all the previous.


var B
var A

Limits of PCA

PCA finds linear principal components, based on the Euclidean distance. It is well suited for clustering problems

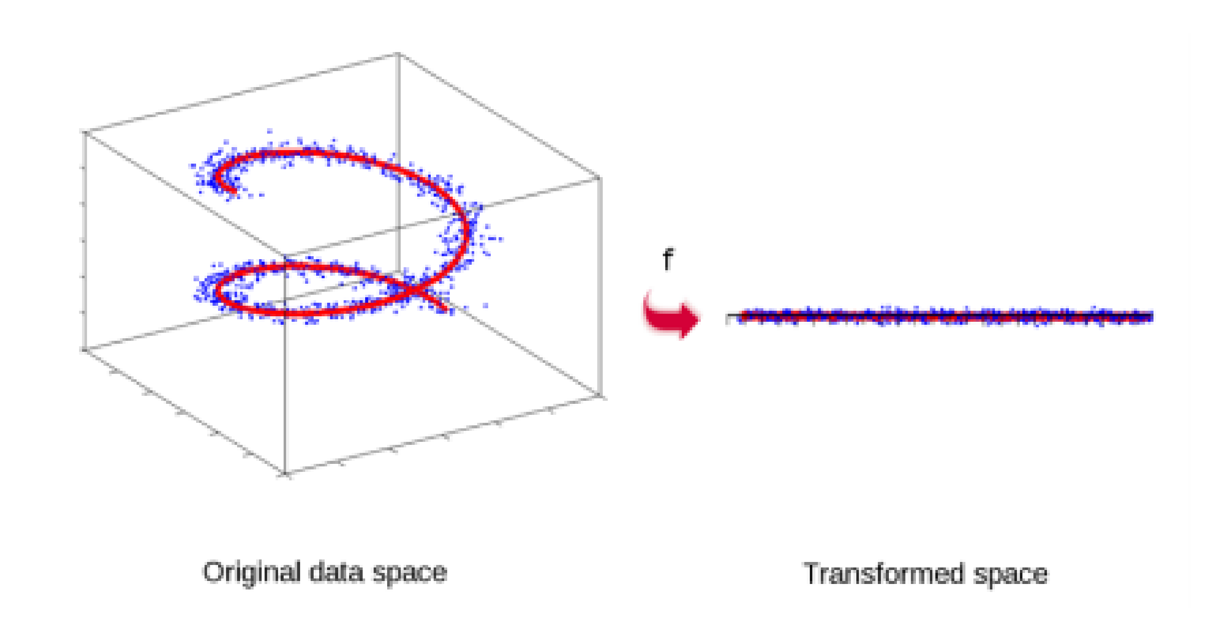


- Data may not be best summarized by linear combination of features
  - Example: PCA cannot discover 1D structure of a helix

Limits of PCA

In order to overcome such kind of limits several methods have been developed based on different principles:

- Combination of PCA (segments)
- Principal curves,
- Principal surfaces
- Principal manifolds



Recap

Feature Selection:

- May Improve performance of classification algorithm
- Classification algorithm may not scale up to the size of the full feature set either in sample or time
- Allows us to better understand the domain
- Cheaper to collect a reduced set of predictors
- Safer to collect a reduced set of predictors

N-n-n-no time, no time, no time!
Hello, good bye,
I am late, I am late....

Just time for a few questions!

Big Bang
Radiation era
~300,000 years:
"Dark ages" begin
~400 million years: Stars
and nascent galaxies form
~1 billion years: Dark ages end
Galaxies evolve
~9.2 billion years: Sun, Earth, and solar system have formed
~13.7 billion years: Present
8 9 10 11 12 1 2 3 4 5 6 7