Data Science 6

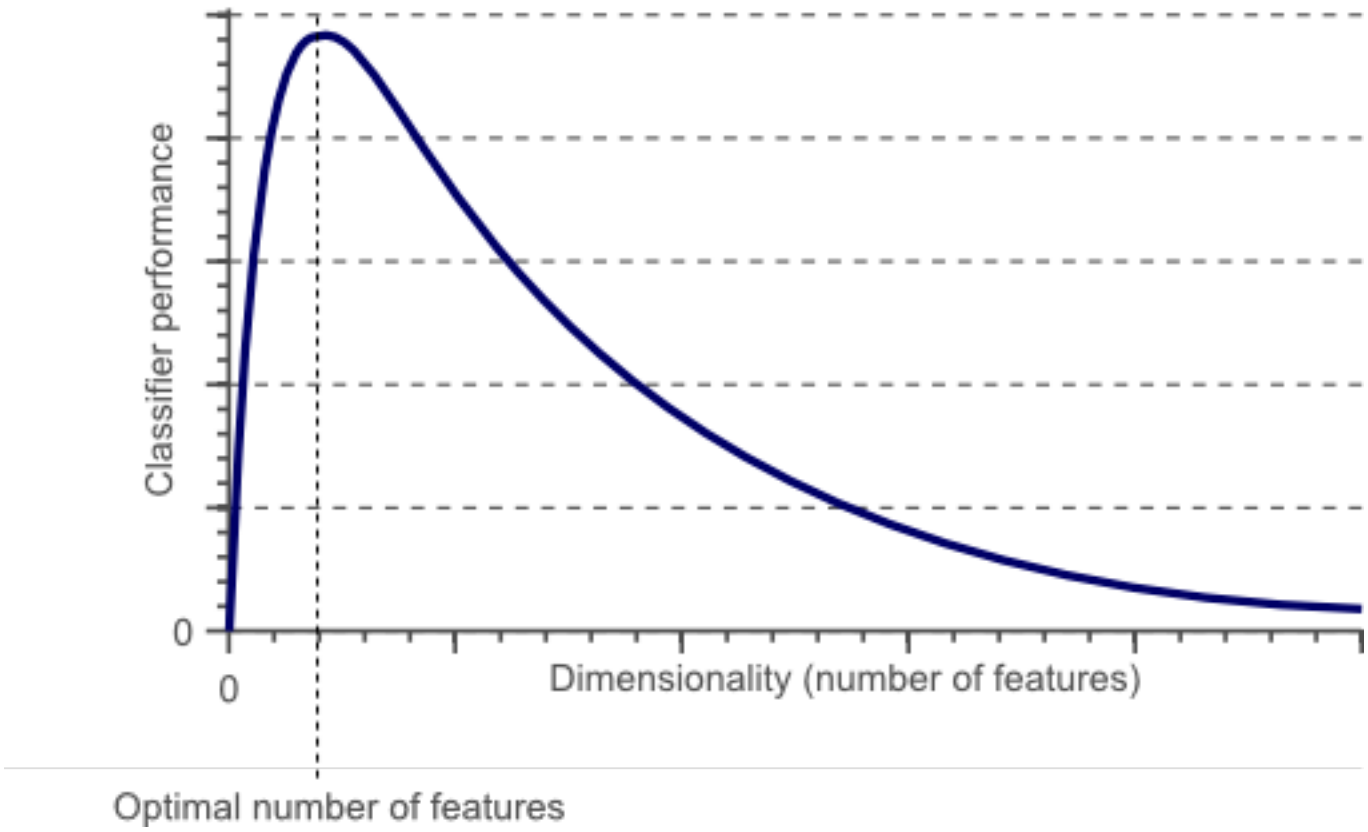
TILBURG UNIVERSITY

Understanding Society

Overview

- the trade off between model complexity and feature sparsity
- the curse of dimensionality
- dimensionality reduction
- PCA
- overfitting-underfitting

The curse of dimensionality



- http://www.datasciencecentral.com/profiles/blogs/about-the-curse-of-dimensionality

Trade-off in machine learning

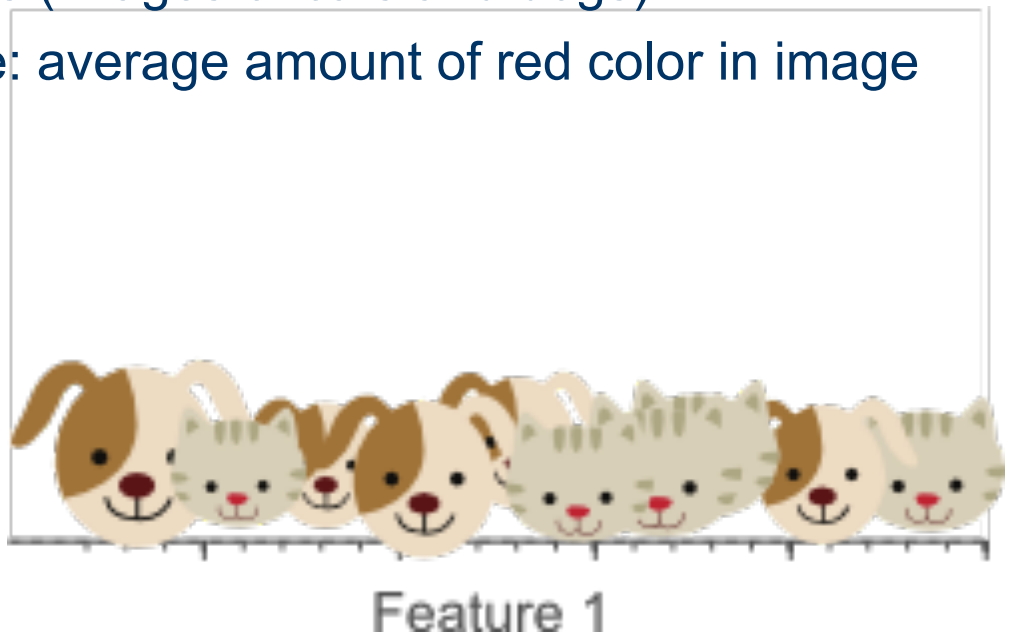
Informative features. We want to increase the number of features to put all the relevant information in the classifier

Curse of dimensionality. We want to decrease the number of features to avoid the curse of dimensionality

- Machine learning algorithms should optimise the trade-off between **informative features** and **curse of dimensionality** by means of dimensionality reduction techniques

Curse of dimensionality and overfitting

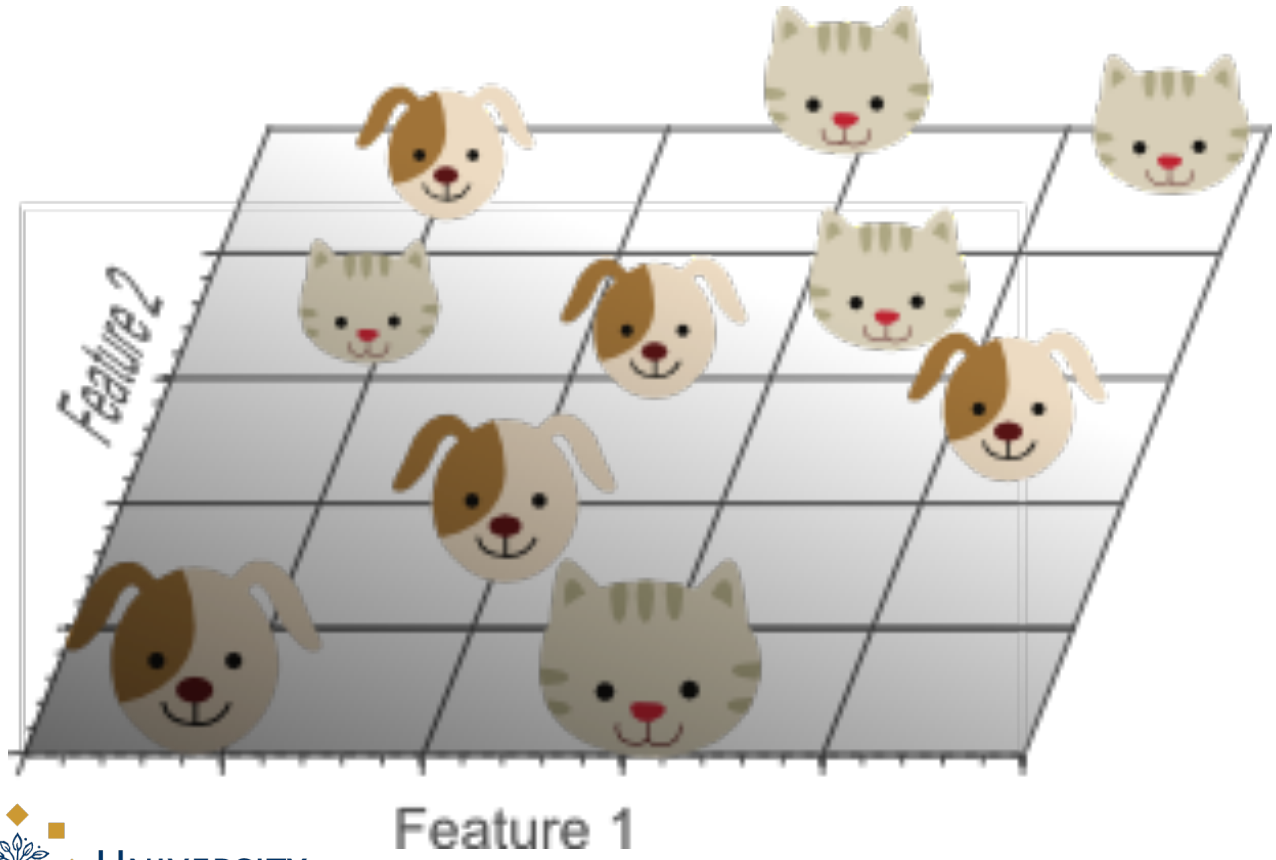
- Classification task: CATS versus DOGS
- 10 instances (images of cats and dogs)
- First feature: average amount of red color in image



- http://www.datasciencecentral.com/profiles/blogs/about-the-curse-of-dimensionality

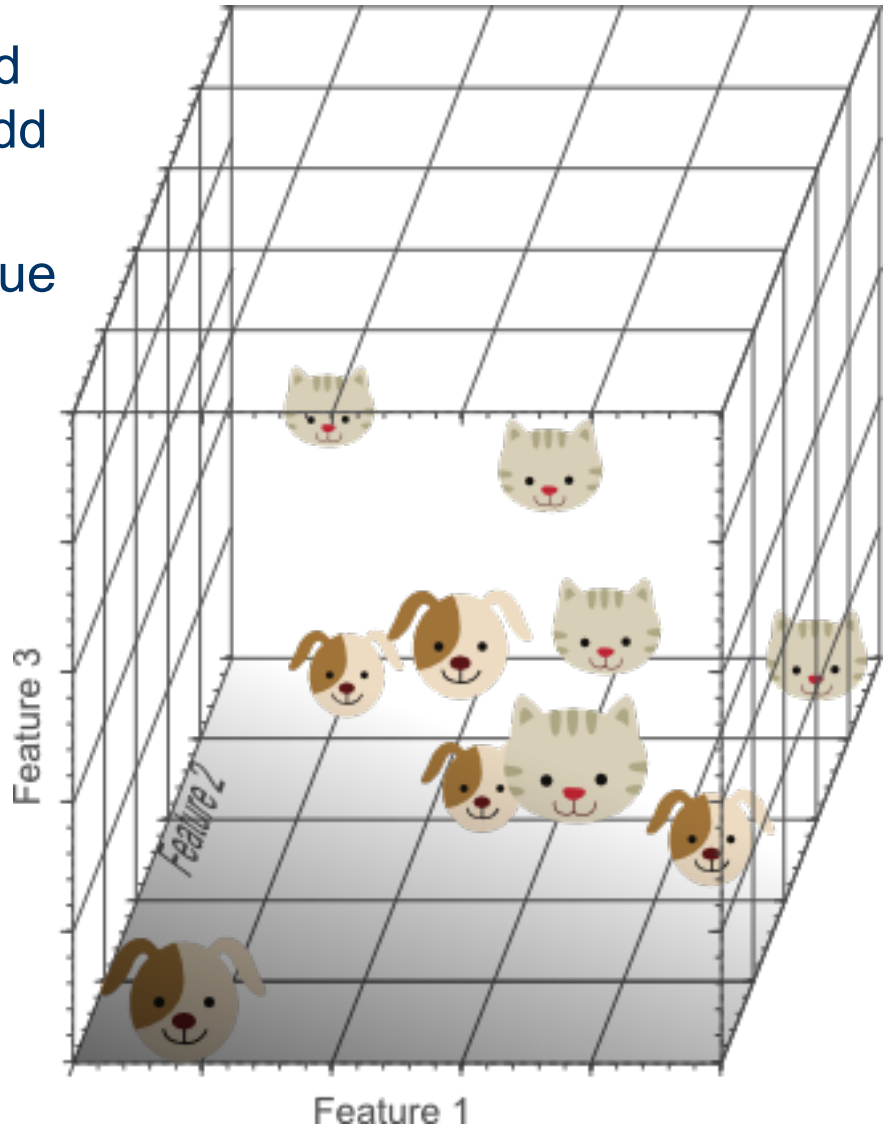
Curse of dimensionality and overfitting

- More information is needed for classification, therefore we add a second feature
- Feature 2: average amount of green color in image

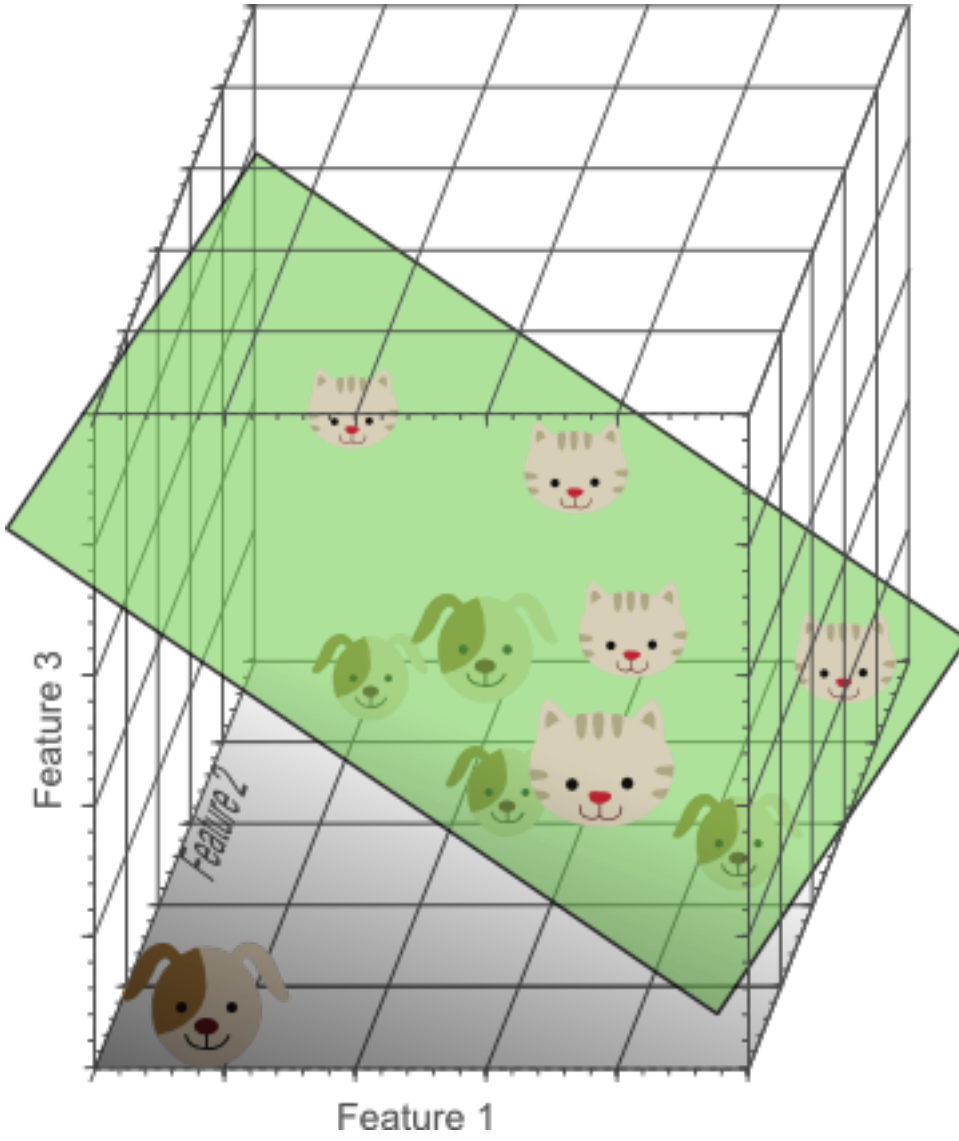


Curse of dimensionality and overfitting

- Even more information is needed for classification, therefore we add a third feature
- Feature 3: average amount of blue color in image



- In three dimensions (= three features), perfect separation of CATS and DOGS is possible with a decision boundary (plane)



Adding features improves classification!…?

- This examples suggests that by adding (informative) features, classification is improved.
- This is often the case, but…
- Adding new features increase the volume of feature space exponentially
- For instance: 1 feature has 10 different values

1 feature: 10 possible feature values

2 features: 100 possible feature values

3 features: 1000 possible feature values

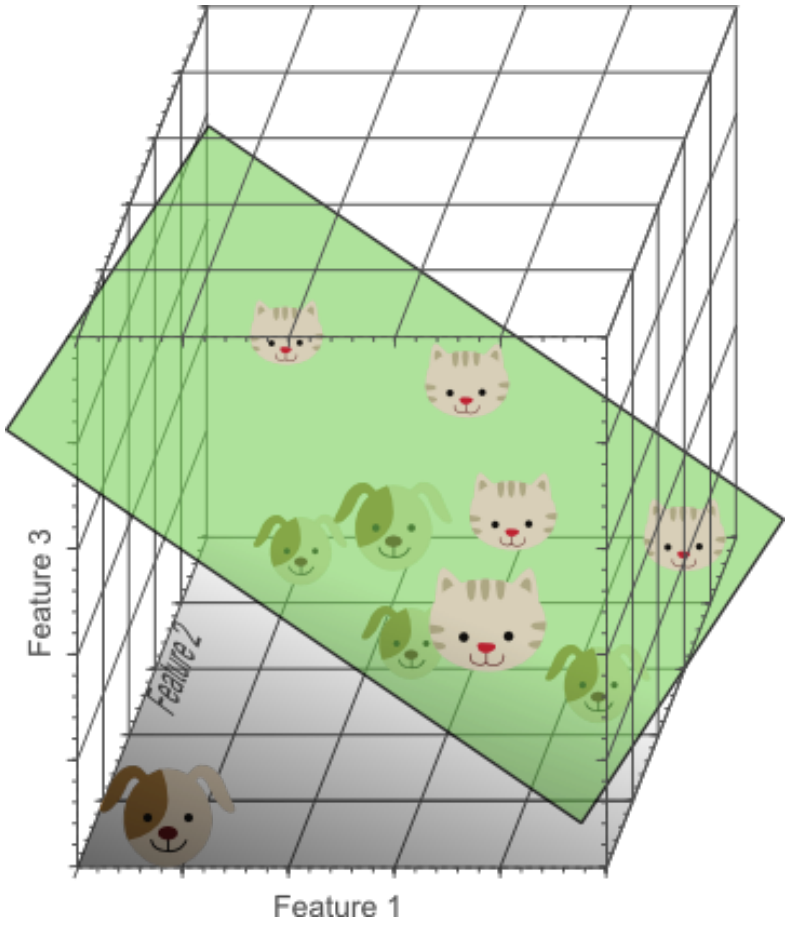
….

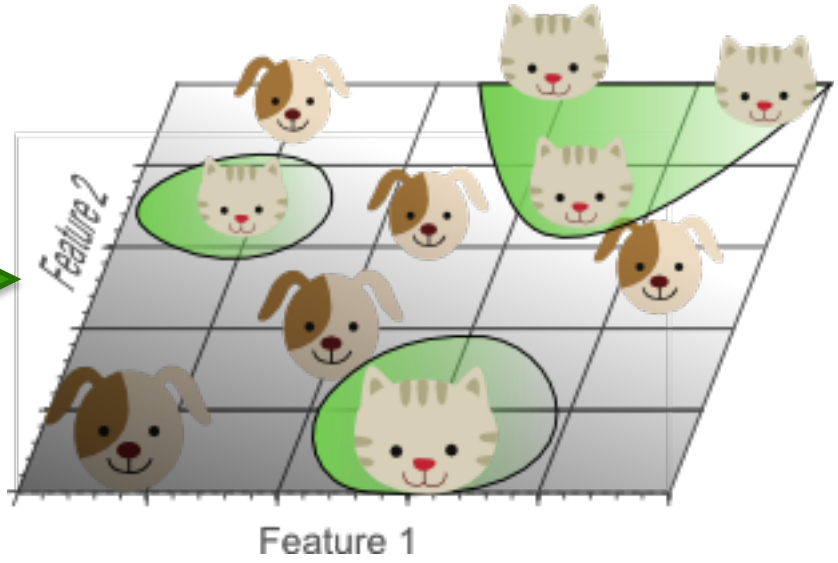
….

Projecting 3D space onto 2D space

linear decision boundary

nonlinear decision boundaries



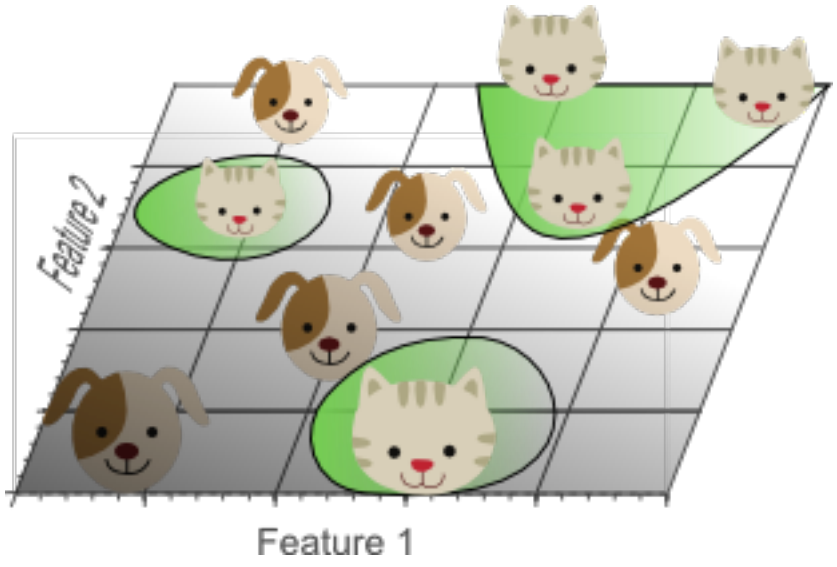


TILBURG UNIVERSITY

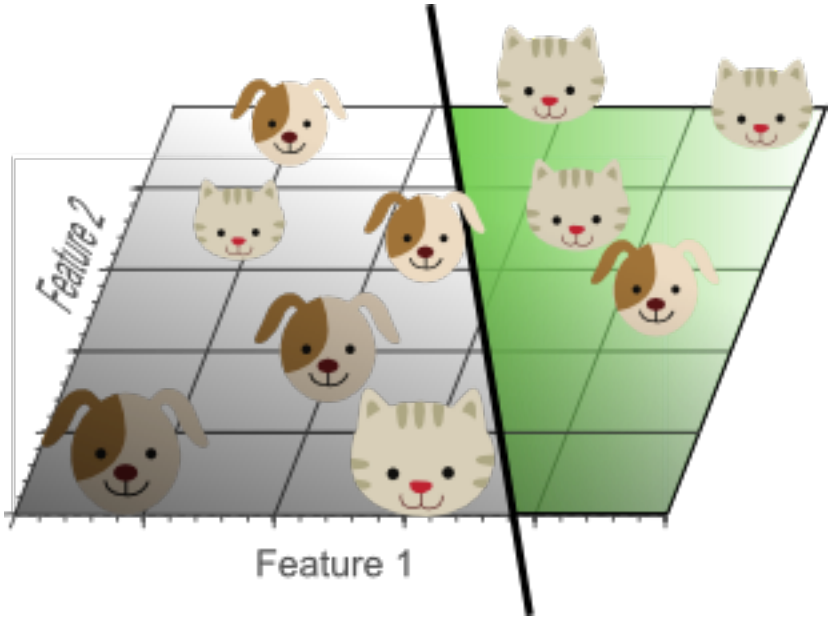
High-dimensional vs low-dimensional feature space

- A simple classification model in a high-dimensional space
  e.g., a linear decision boundary (plane) in 3D
- Corresponds to a complex classification model in low-dimensional space
  e.g., non-linear decision boundaries in 2D
- Overfitting is associated with (too) complex models
- Hence, too many features may lead to overfitting too

Complex vs simple model



This model makes no mistakes on the training set, but doesn't generalise well…



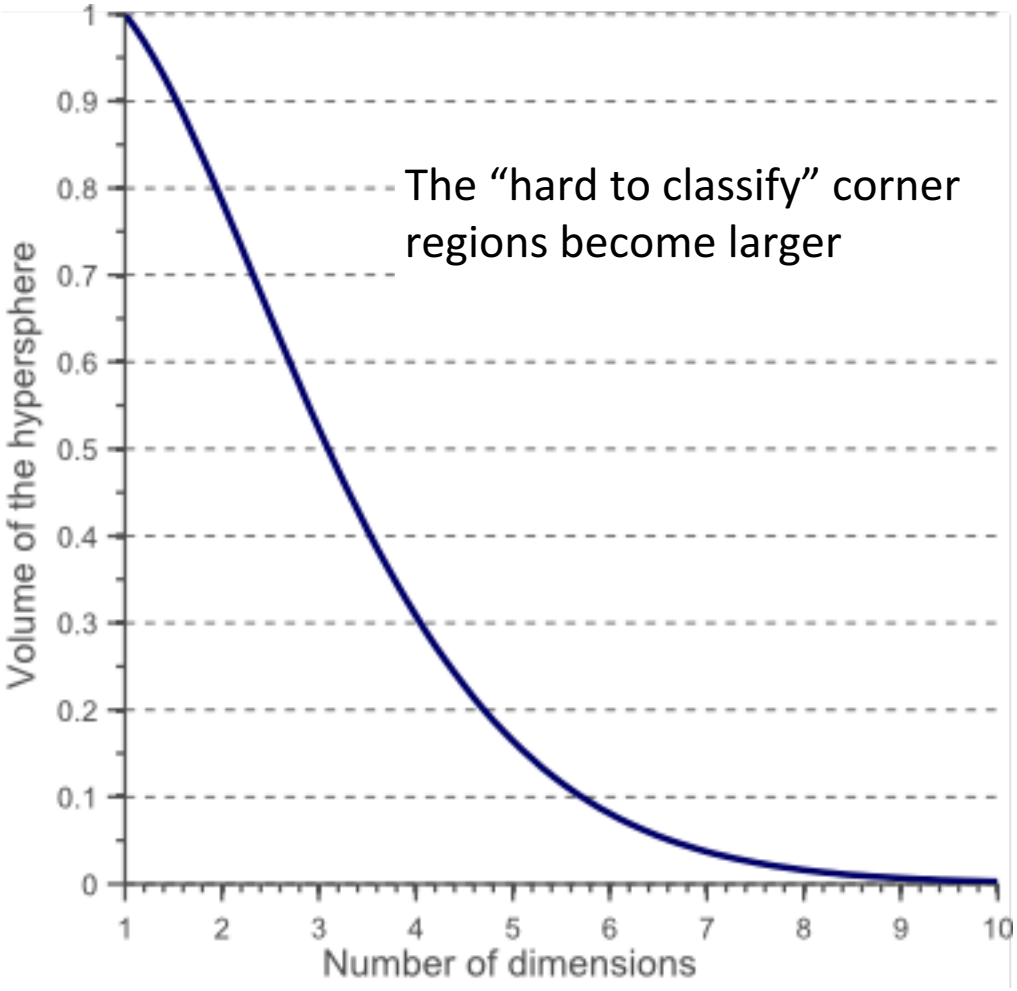
Although this model does make some mistakes on the training set, it generalises much better to unseen instances

High-dimensional feature space

hard to classify

What happens if dimensionality (number of features) increases?

High-dimensional feature space

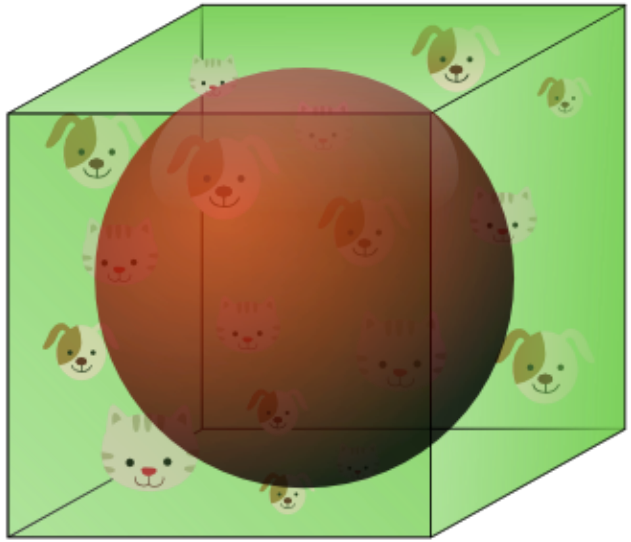


= number of features

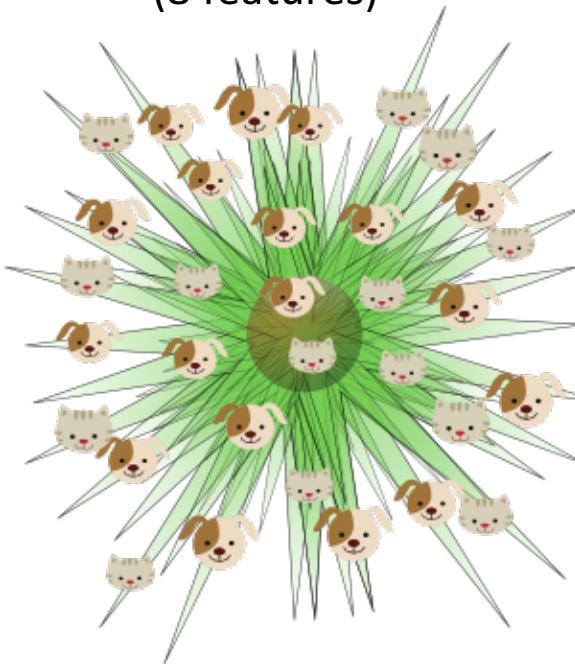
Curse of dimensionality (illustration)

2 dimensional feature space (2 features)

3 dimensional feature space (3 features)



"8 dimensional feature space" (8 features)



98% of the instances are of the "hard to classify" type

How many features?

"Regrettably there is no fixed rule that defines how many features should be used in a classification problem. In fact, this depends on the amount of training data available, the complexity of the decision boundaries, and the type of classifier used."

http://www.datasciencecentral.com/profiles/blogs/about-the-curse-of-dimensionality

Dimensionality Reduction

- Feature Selection

- Feature combination

Principal Component Analysis (PCA)

- PCA operates on the features without labels
  = unsupervised learning
- Each feature is an axis of a N-dimensional coordinate system
- Two features: XY coordinate system
- PCA rotates the XY coordinate system

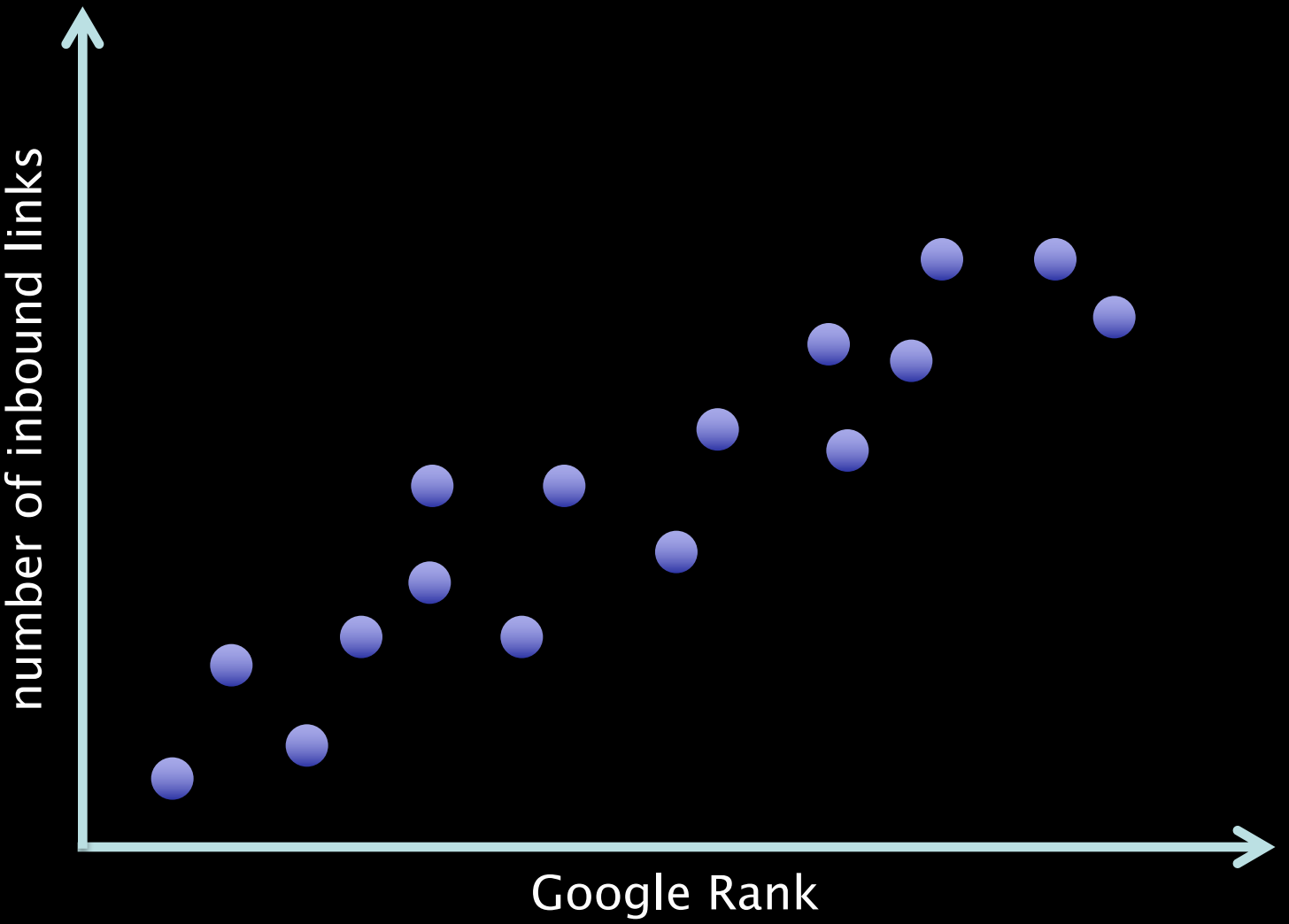
PCA example

- Two features describing a website

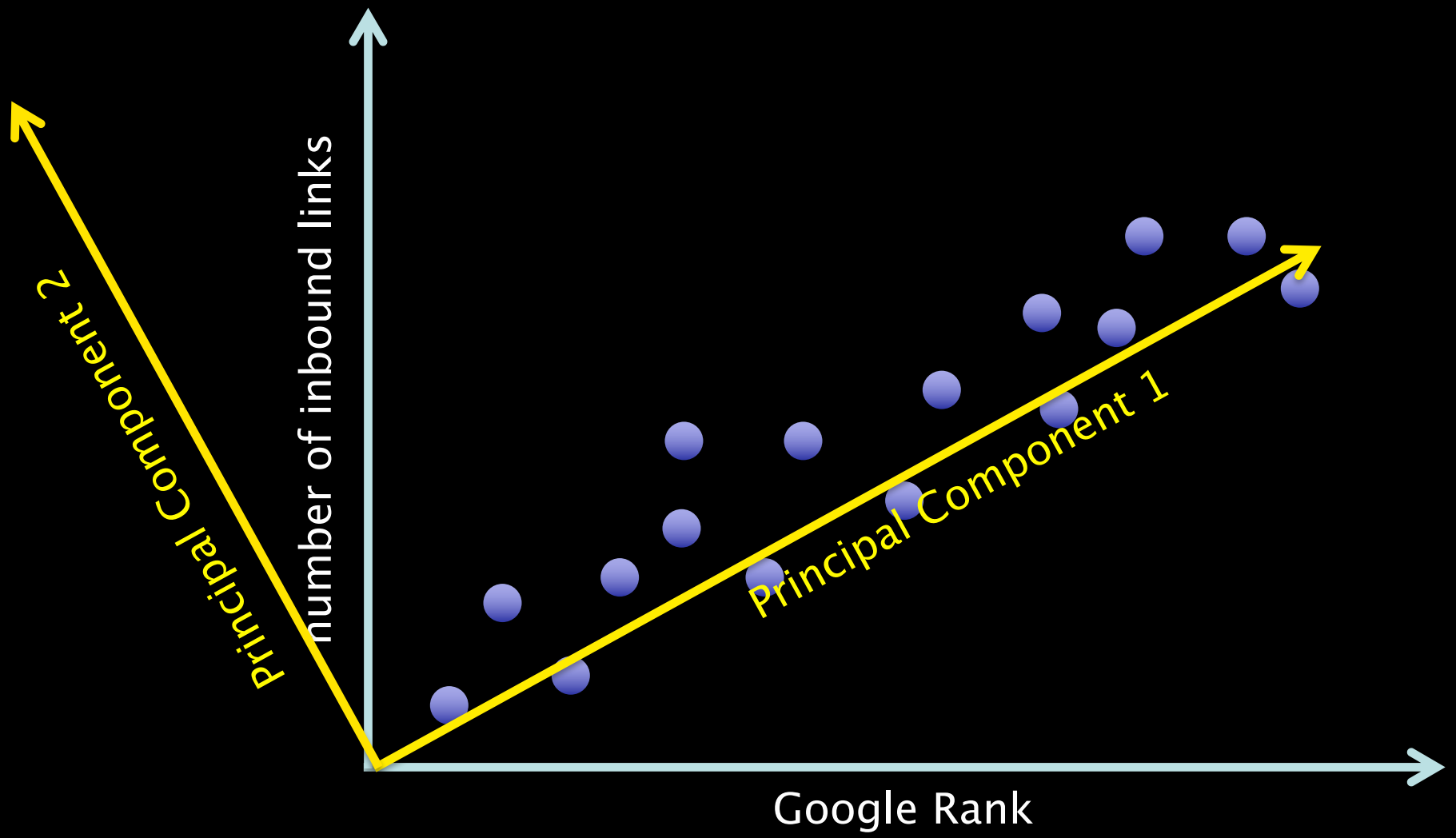
1. Number of inbound links
2. Google Rank

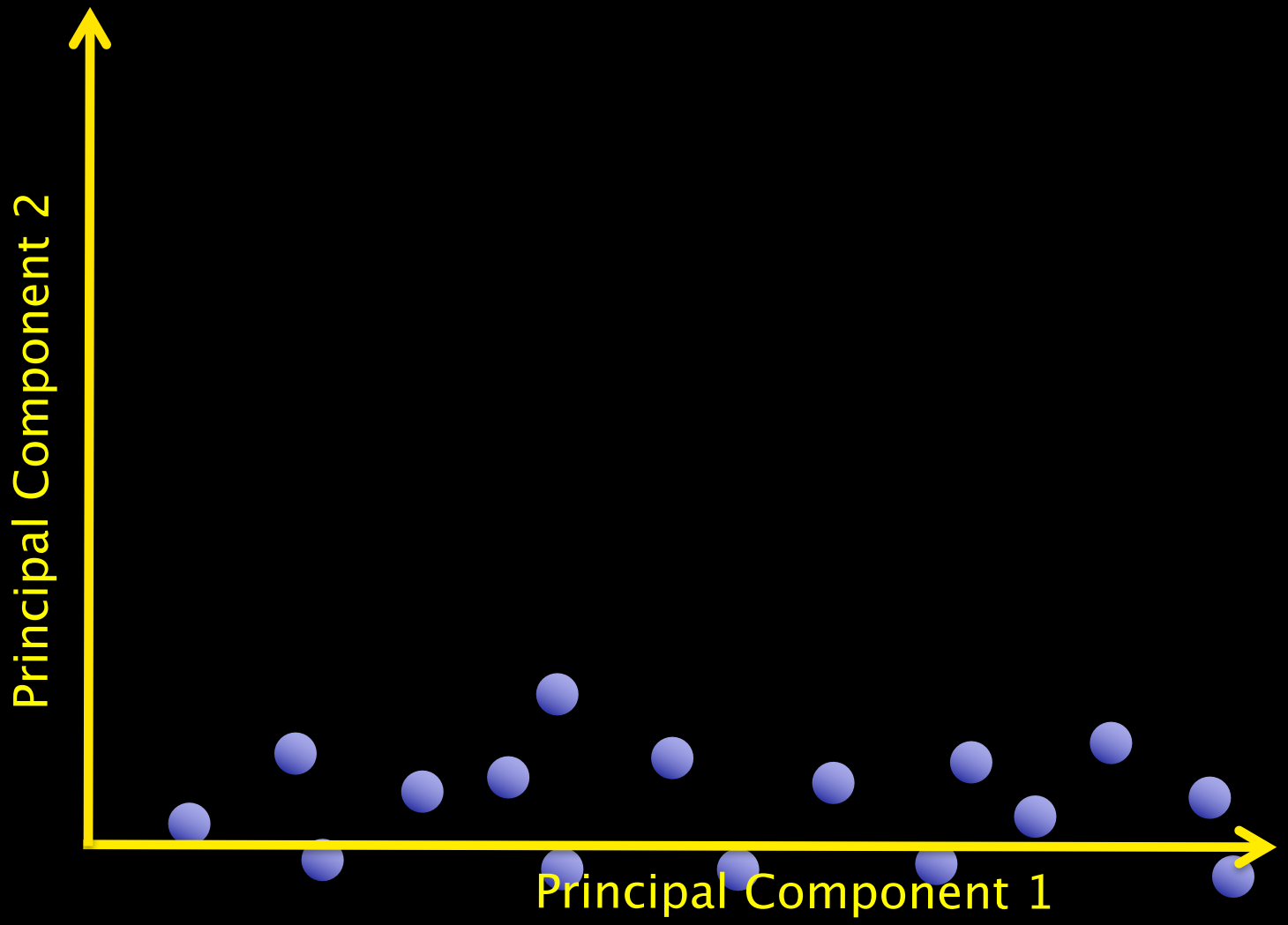
- These features are correlated and therefore redundant

Correlated Features



PCA


number of inbound links
Google Rank
Principal Component 1
Principal Component 2


Principal Component 2
Principal Component 1

PCA

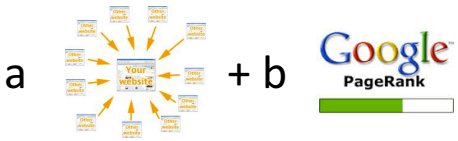
- Takes 2 features as input

1. Number of inbound links

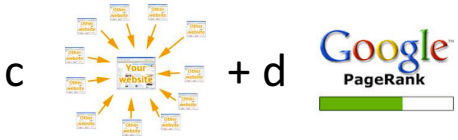
2. Google Rank

- … and gives 2 new features as output

- Principal Component 1



- Principal Component 2

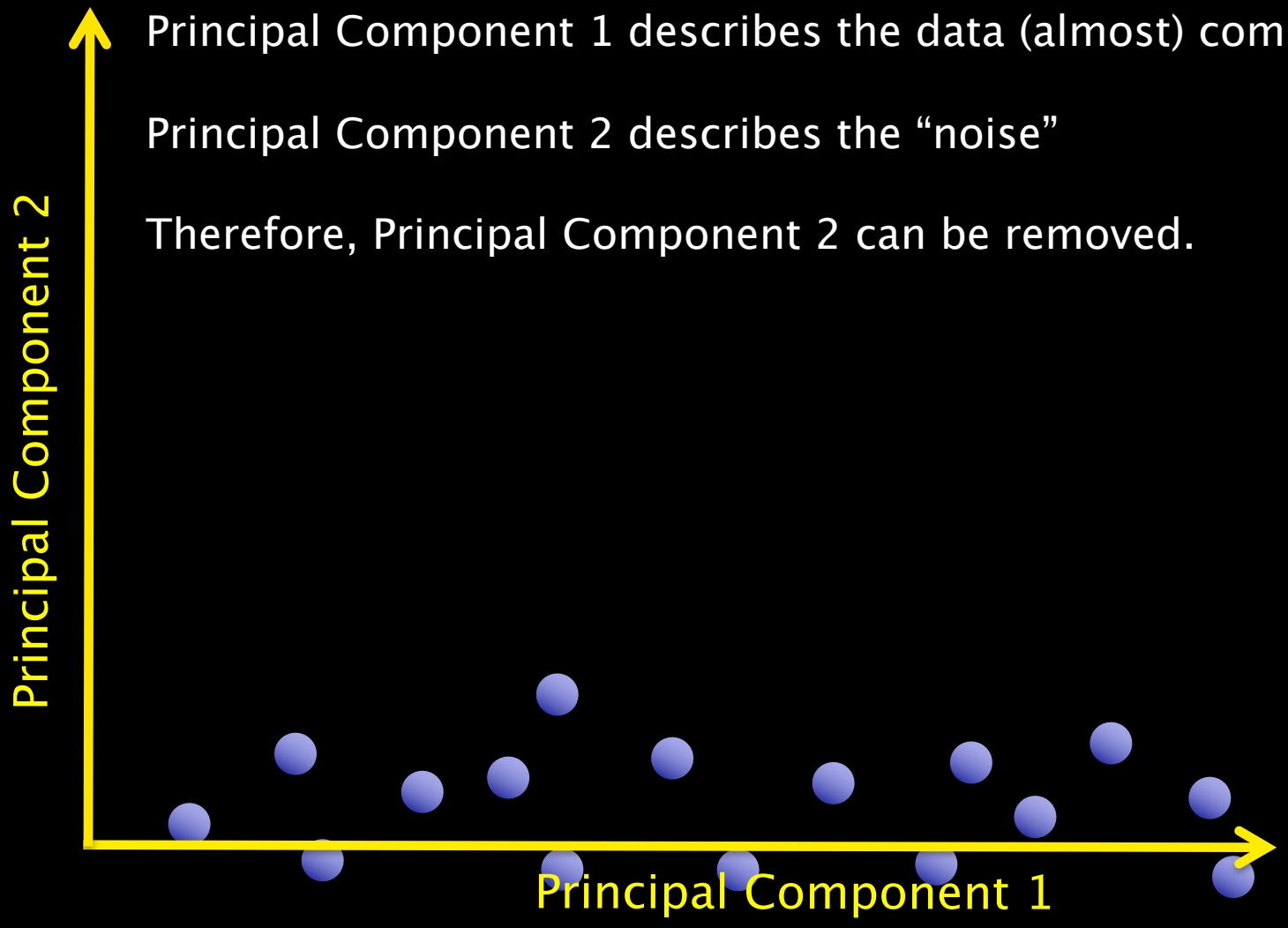


TILBURG UNIVERSITY

Principal Component 1 describes the data (almost) completely

Principal Component 2 describes the “noise”

Therefore, Principal Component 2 can be removed.



PCA (highest component removed)

- Takes 2 features as input

1. Number of inbound links

2. Google Rank

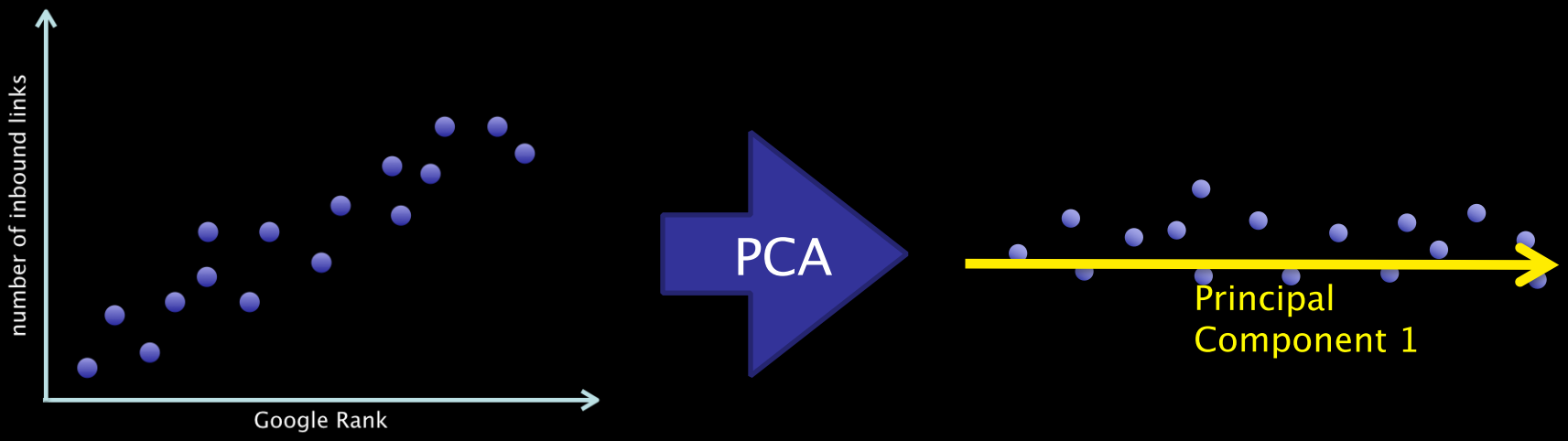
- … and gives **1 feature** as output

- Principal Component 1

a  + b 

Dimensionality Reduction

- PCA performs **dimensionality reduction**
- It reduces 2 dimensions (features) to 1

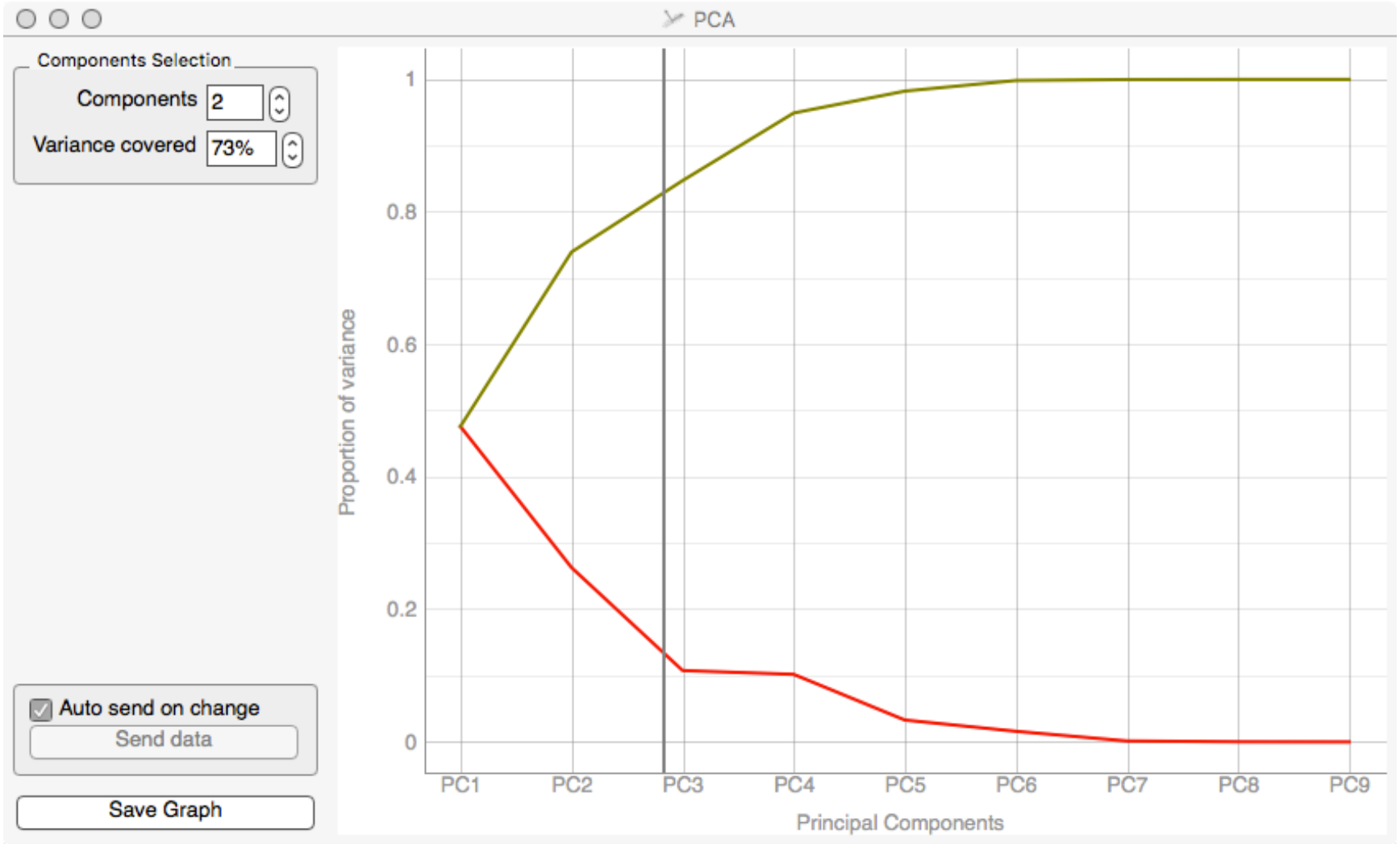


PCA in higher dimensions

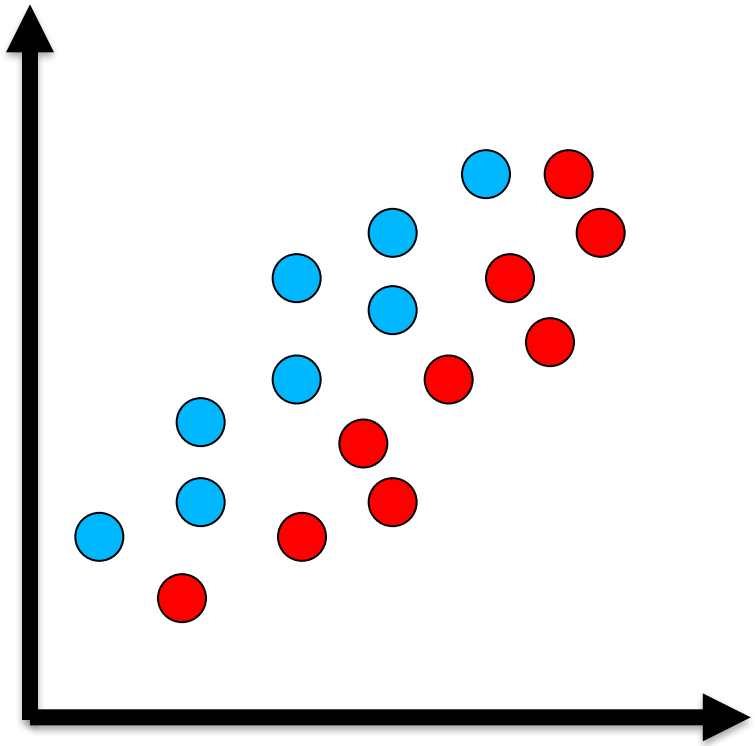
- For datasets with more than 2 features, PCA rotates the coordinate system in such a way that:
  - the projection of the data on the first principal component (new axis) has the largest variance,
  - the projection of the data on the second principal component (new axis) has the one-but-largest variance,
  - and so forth…

- **If** the variation in the data is associated with relevance for classification (or regression), the most relevant features are captured by the first principal components (and the rest captures noise)
- Retaining the first principal components and throwing away the rest effectively reduces the dimensionality

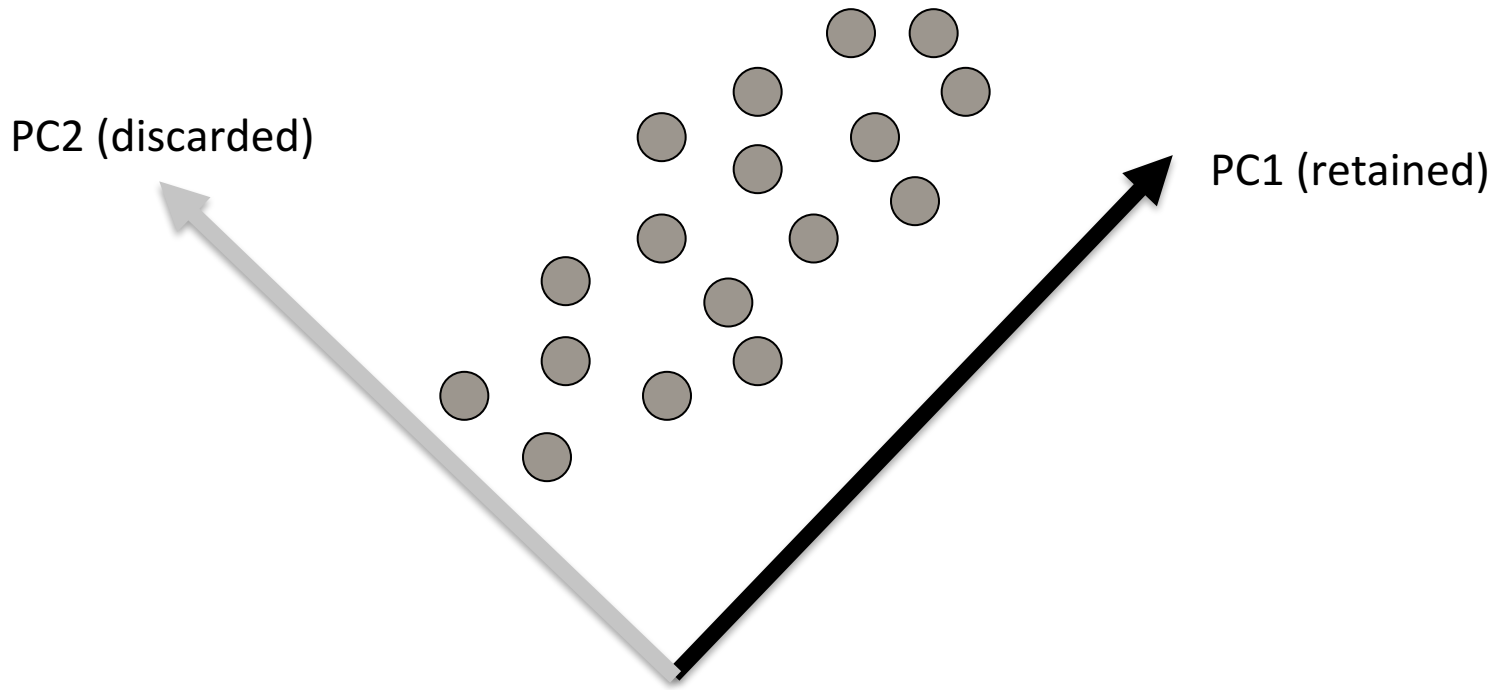
Scree Plot


PCA
Components Selection
Components 2
Variance covered 73%
Auto send on change
Send data
Save Graph
Proportion of variance
1
0.8
0.6
0.4
0.2
0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
PC8
PC9
Principal Components

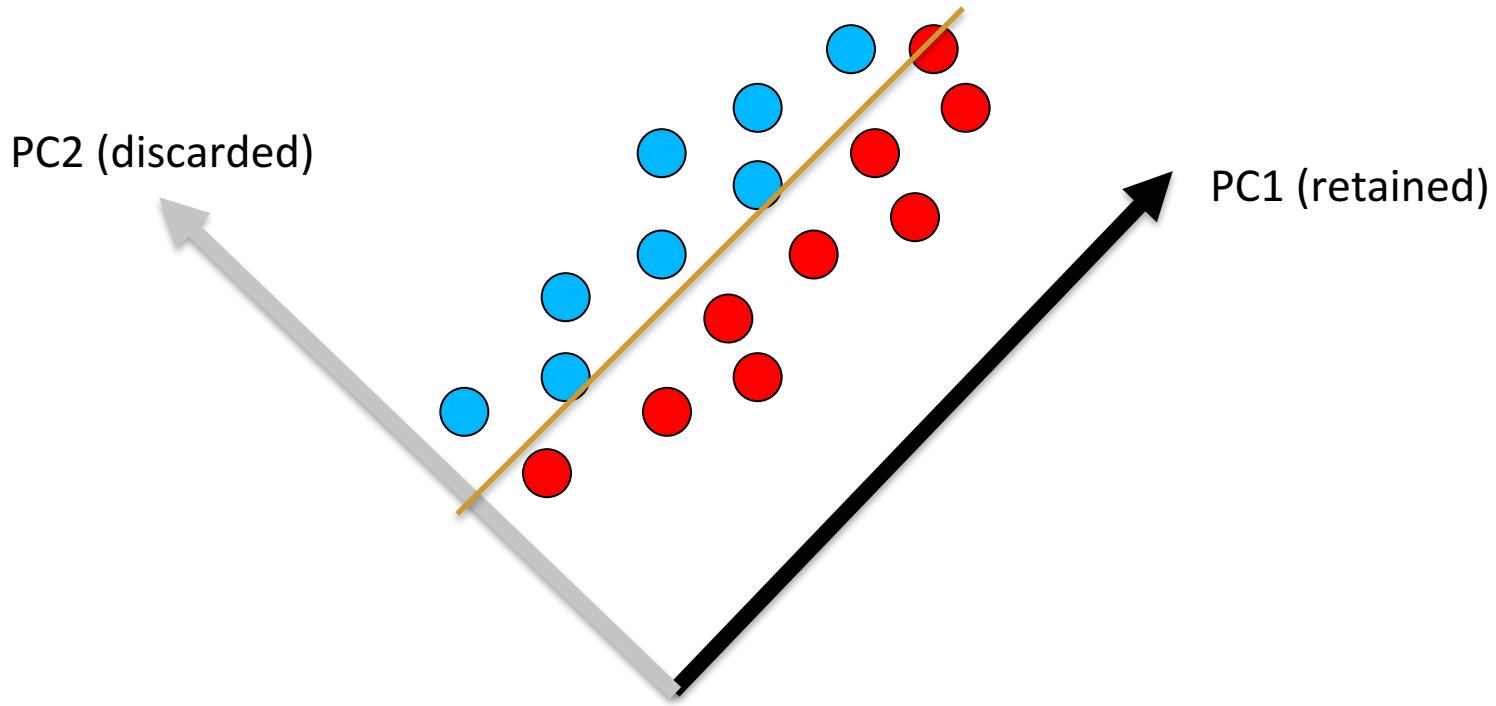
PCA is color blind! (unsupervised)



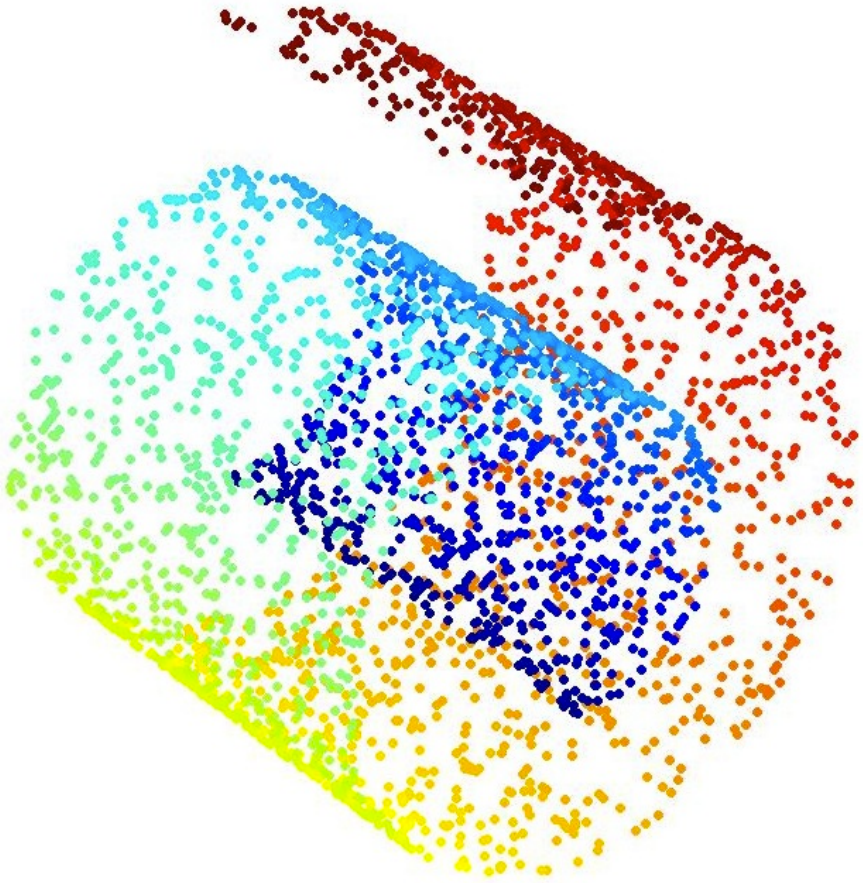
Unlabelled points



PC2 separates, PC1 does not!



Swiss Roll



Unfolded Swiss Roll (2D)