



EMBnet



**Karolinska
Institutet**

EMBNET COST.CHARME TRAINING SCHOOL "BIG DATA FOR LIFE SCIENCES"

Introduction to Machine Learning

Gioele La Manno

*September 18th
Ultuna campus,
Uppsala Sweden*

My aim with this lecture

- Make you aware of the main classes of Machine learning (ML) algorithms
- Enable you to use them by understanding the general framework
- Recognize that common biological questions about data can be formulated and solved by using one or more of ML algorithms
- Stretch your mind to think multidimensional

Not covered:

- Mathematical formulation of each algorithm I will mention
- Comparative between different approaches
- I will not talk about Bayesian models, HMM or neural networks

Overview

Introduction to machine learning

- High dimensionality
- Unsupervised methods
- Regression
- Classification
- Cross-validation and model selection

What is Machine learning?

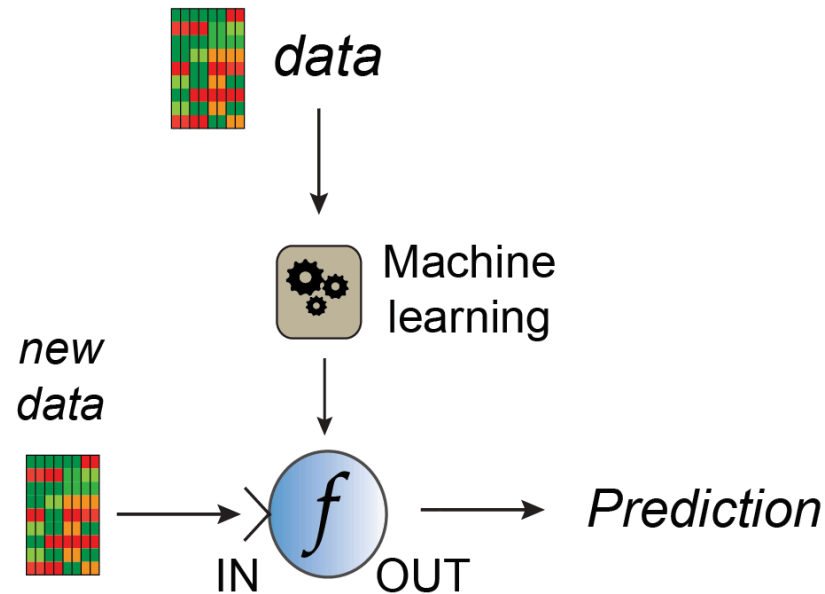
“ “ The field of machine learning is concerned with the question of how to construct computer programs that automatically improve with experience.

Tom Mitchel

In other words:

Machine Learning studies models that can learn to make predictions from data instead of using static instructions

What is Machine learning?



In a programmatic way:

```
trained_model <- model(data, known_quantity)
predicted_quantity <- trained_model(new_data)
```

Supervised or Unsupervised?

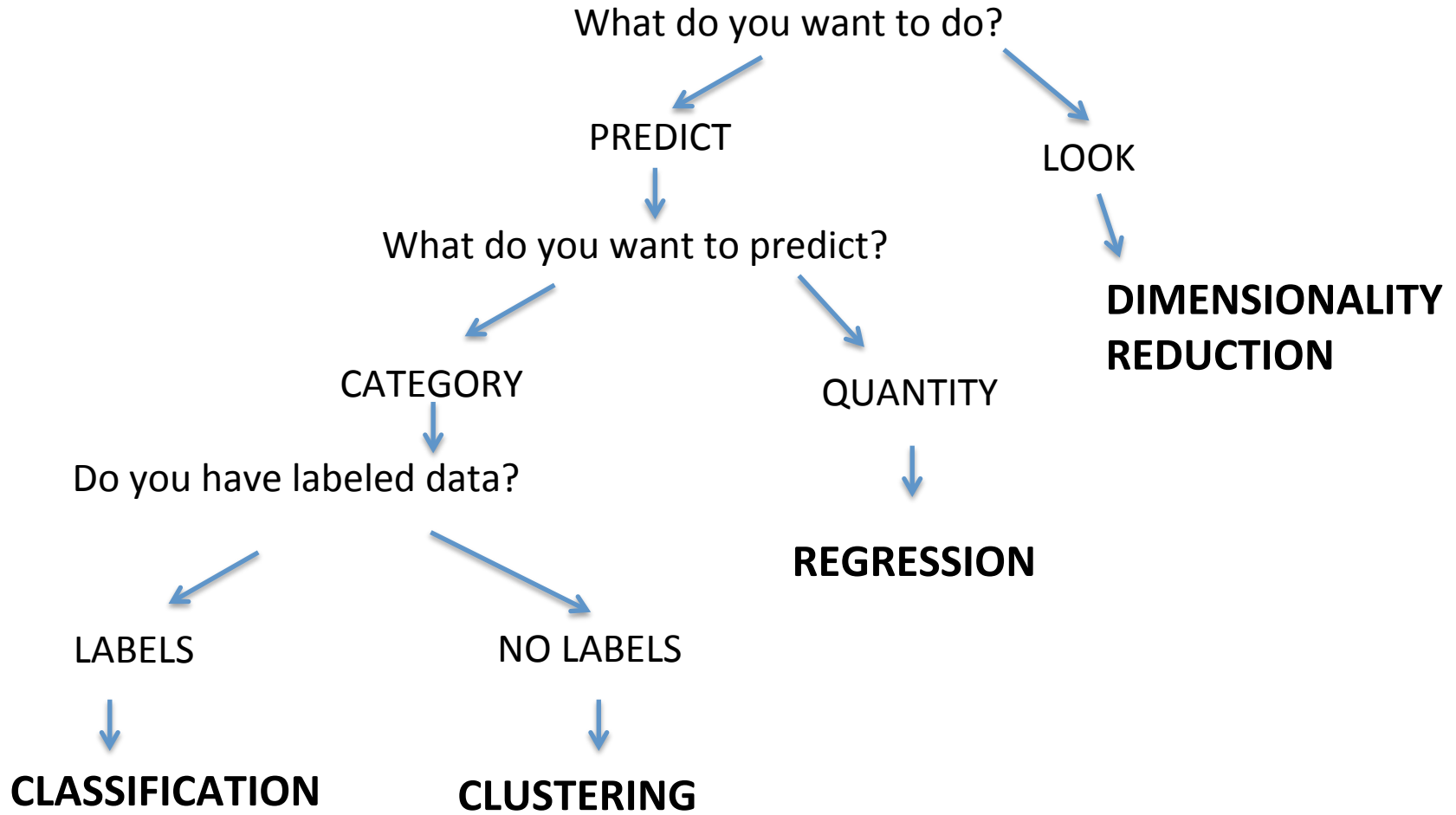
Supervised learning

Training your machine to learn a function
by showing couples of input and corresponding output (target)
→ *Classification and Regression*

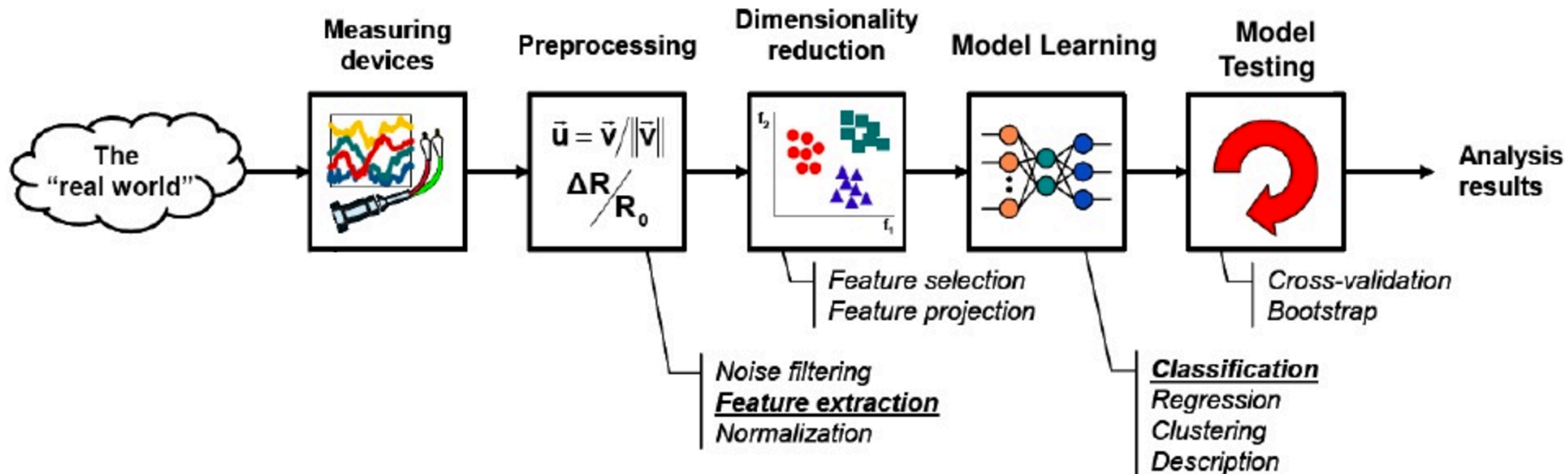
Unsupervised learning

Training your machine to learn structure or relationships
by presenting to it a set of inputs
→ *Clustering and Dimensionality reduction*

Classes of machine learning problems



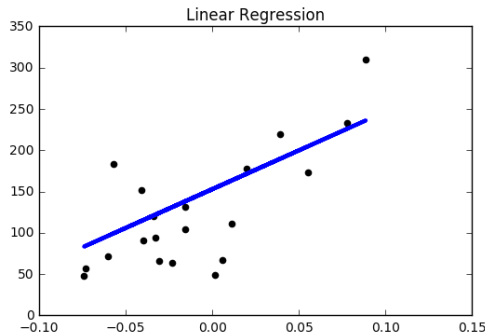
A data analysis pipeline



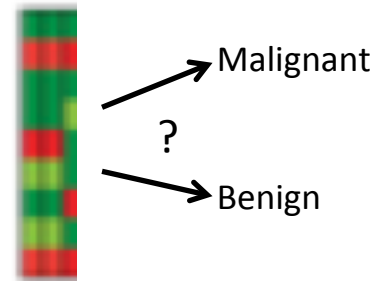
A little bit of demystification

Often machine learning models are a generalization of well known models you use every day.

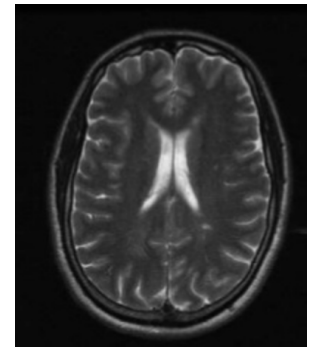
Linear regression



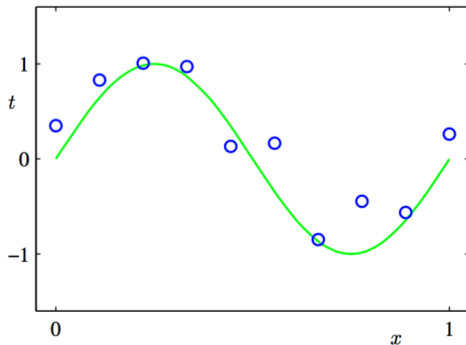
Classification



MRI



Curve fitting



Missing data prediction



Dimensionality of the data

Trivial cases

Example:

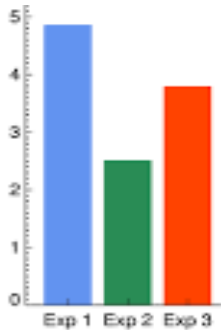
Typical assay

Point:

x_1

1 dimension

Representation of the space:



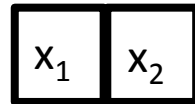
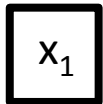
Dimensionality of the data

Trivial cases

Example:

Typical assay

FACS 2 markers

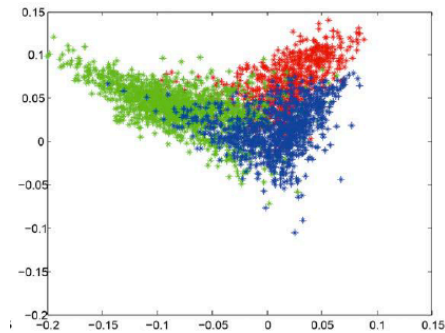
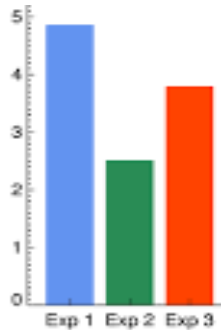


1 dimension

2 dimensions

Point:

Representation of the space:



Dimensionality of the data

Trivial cases

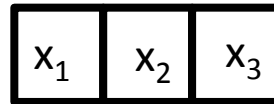
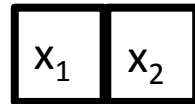
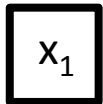
Example:

Typical assay

FACS 2 markers

IHC 3 colors

Point:

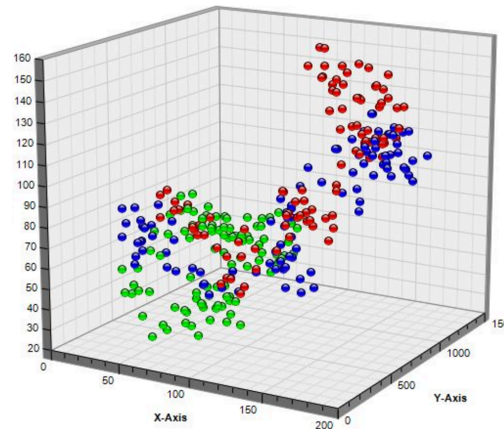
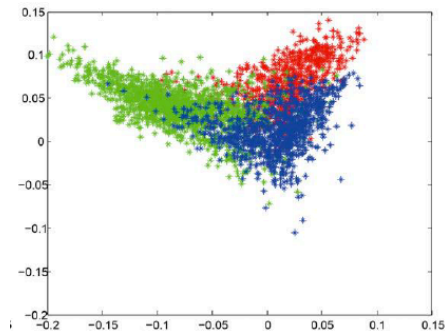
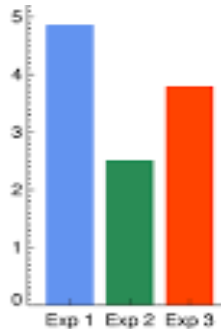


1 dimension

2 dimensions

3 dimensions

Representation of the space:



Dimensionality of the data

Trivial cases

Example:

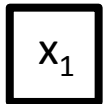
Typical assay

FACS 2 markers

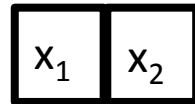
IHC 3 colors

Real time 4 genes

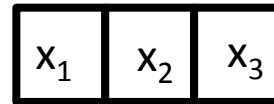
Point:



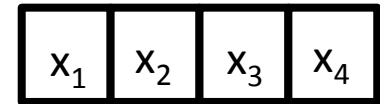
1 dimension



2 dimensions

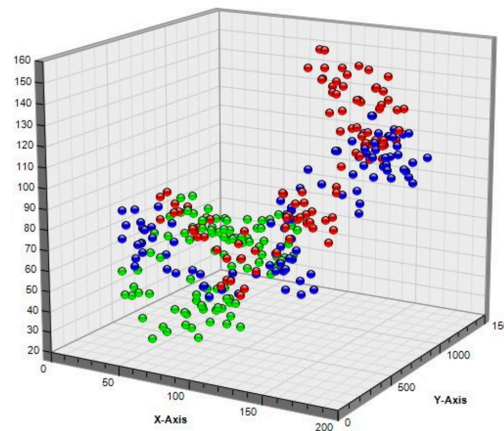
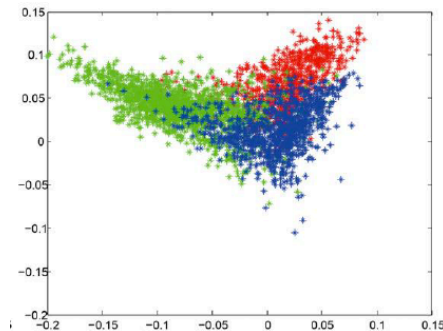
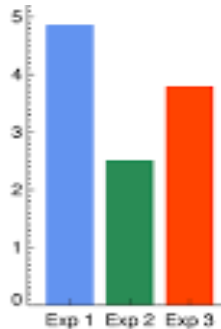


3 dimensions



4 dimensions

Representation of the space:

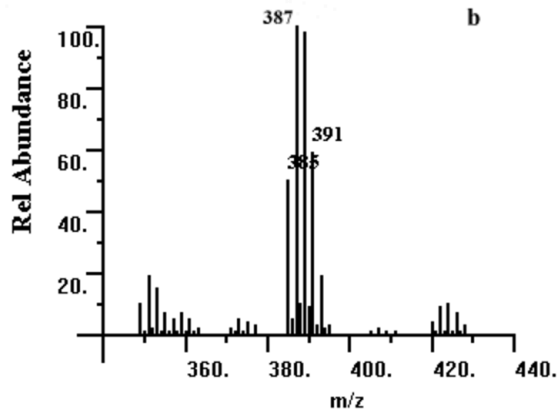


Dimensionality of the data

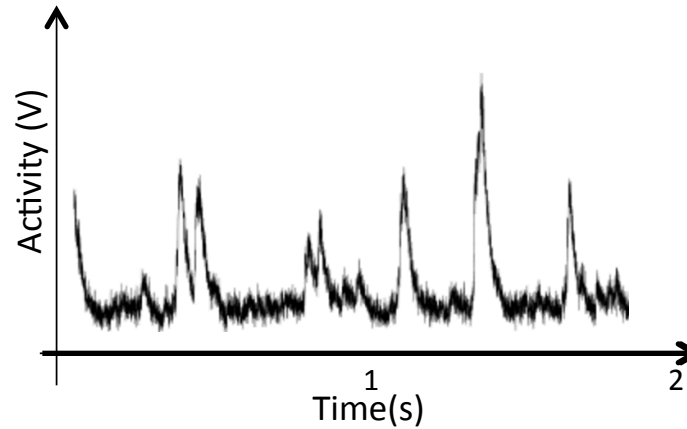
Less Trivial cases

Example:

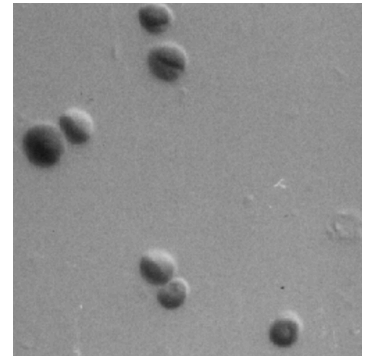
Mass spec trace
Range (340-440m/z)
resolution (2m/z)



Electrophysiological
Recording
2sec @ 200Hz



Micrograph
128x128 pixels

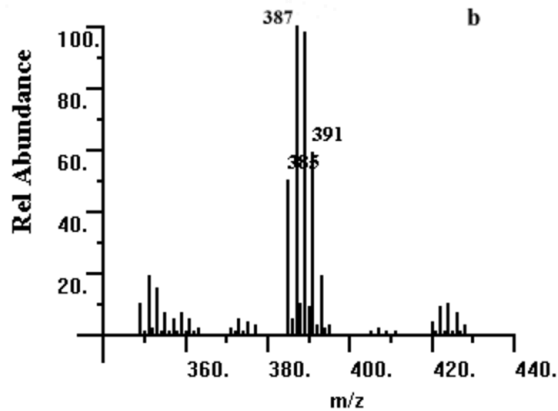


Dimensionality of the data

Less Trivial cases

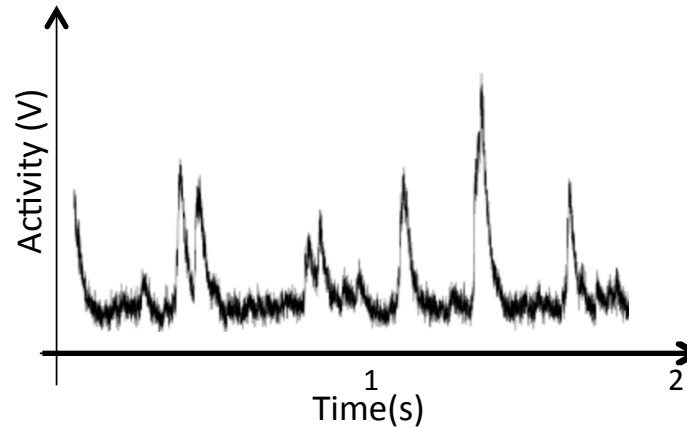
Example:

Mass spec trace
Range (340-440m/z)
resolution (2m/z)



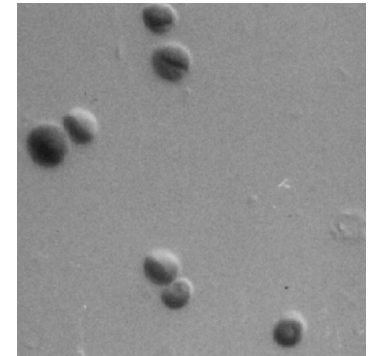
50 dimensions

Electrophysiological
Recording
2sec @ 200Hz



400 dimensions

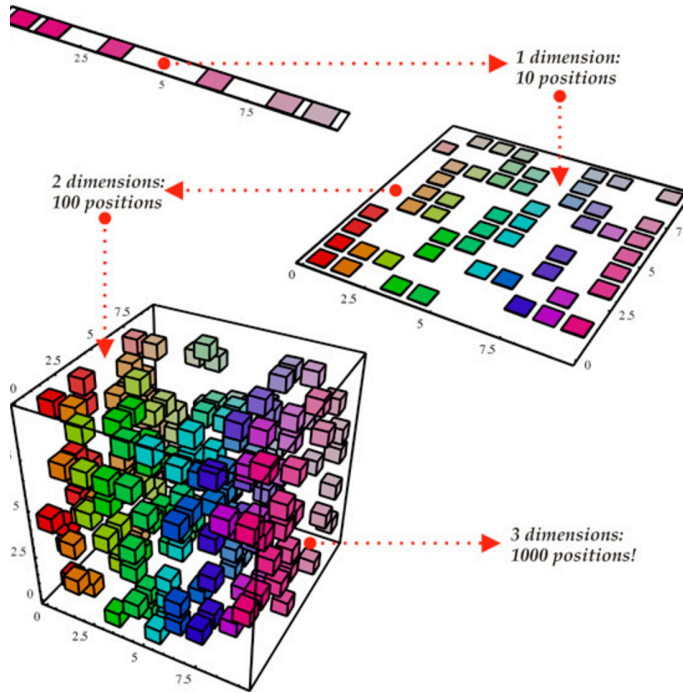
Micrograph
128x128 pixels



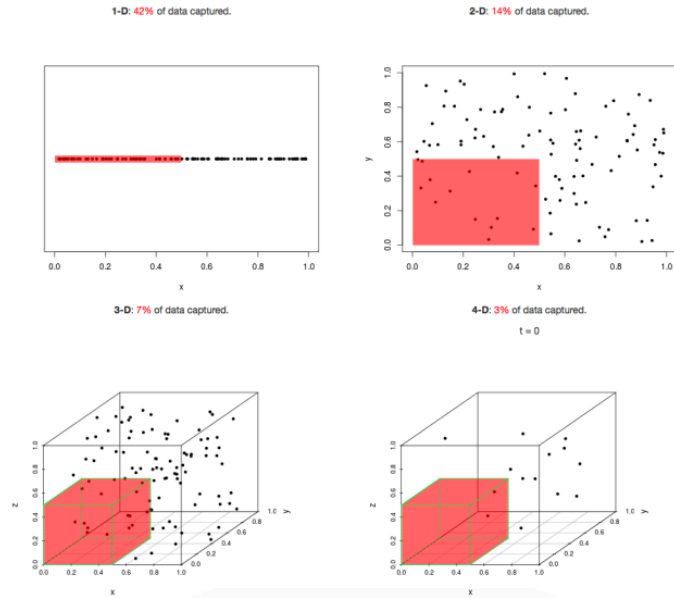
16384 dimensions

The curse of dimensionality

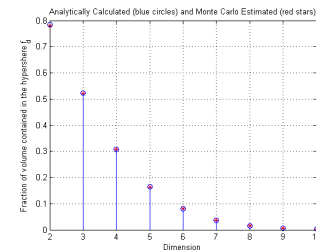
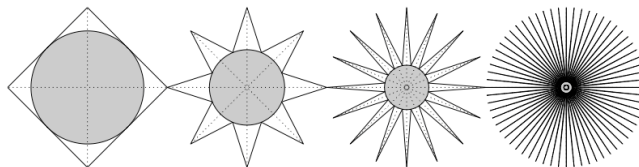
Possible position increase exponentially



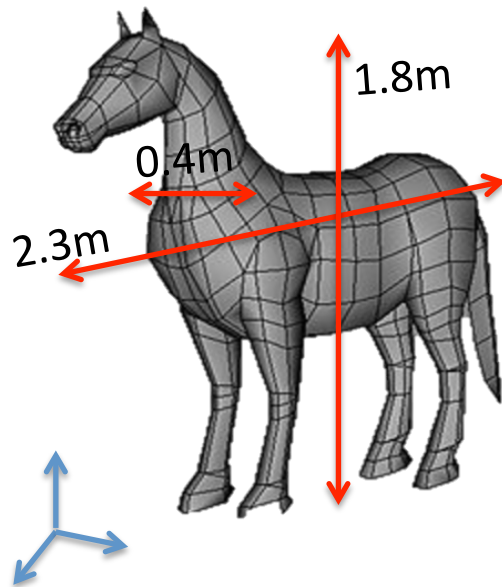
Space is emptier / points are distant



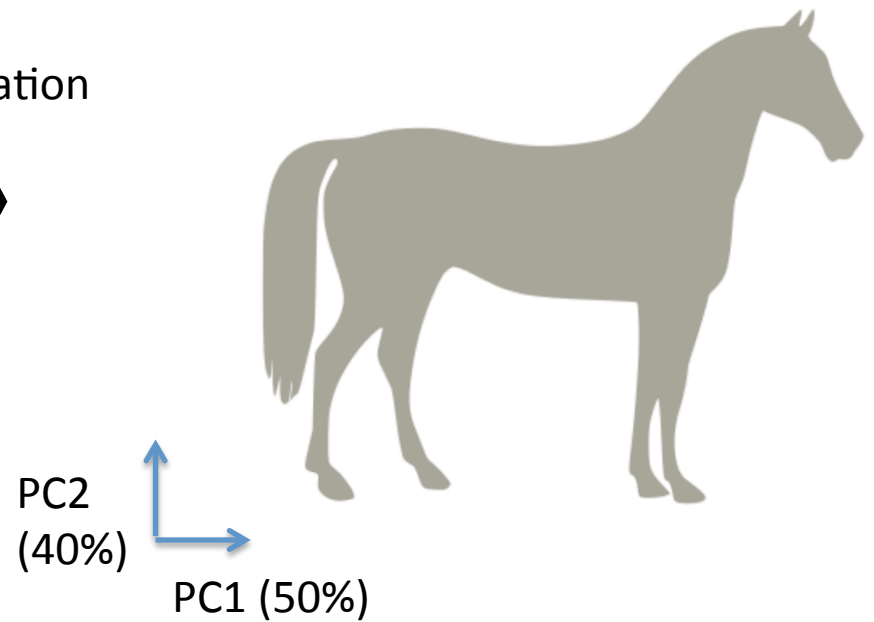
Not only this.... Also weird things start to happen



Principal component analysis: a rotation in multidimensional space

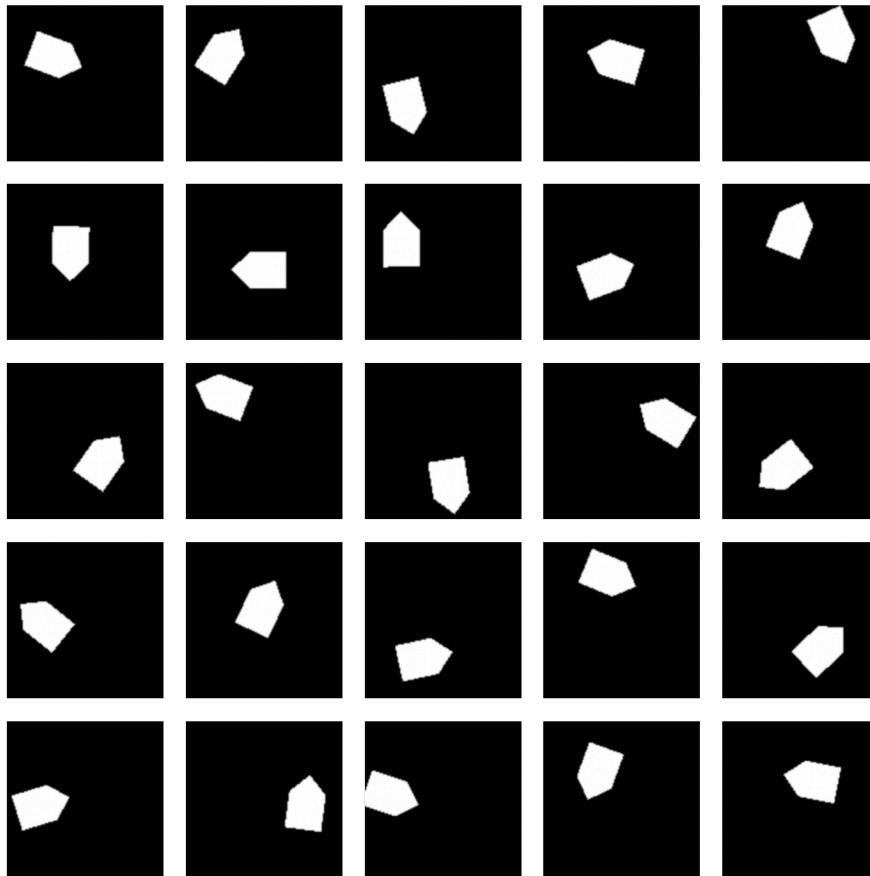


Rotation



Dimensionality reduction

Conveyor belt



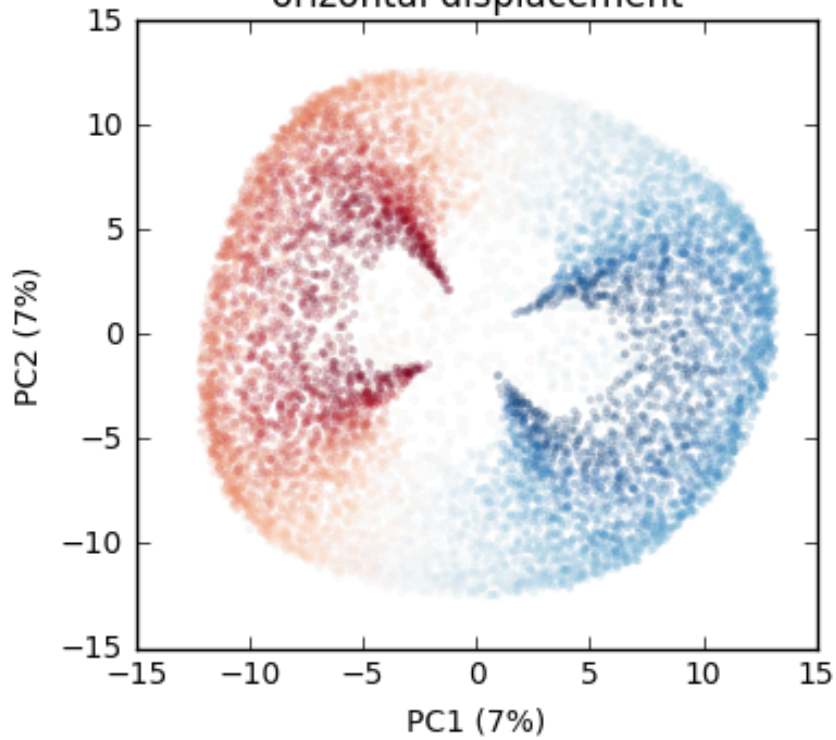
Images
128 * 128 pixels

(16K dimensions)

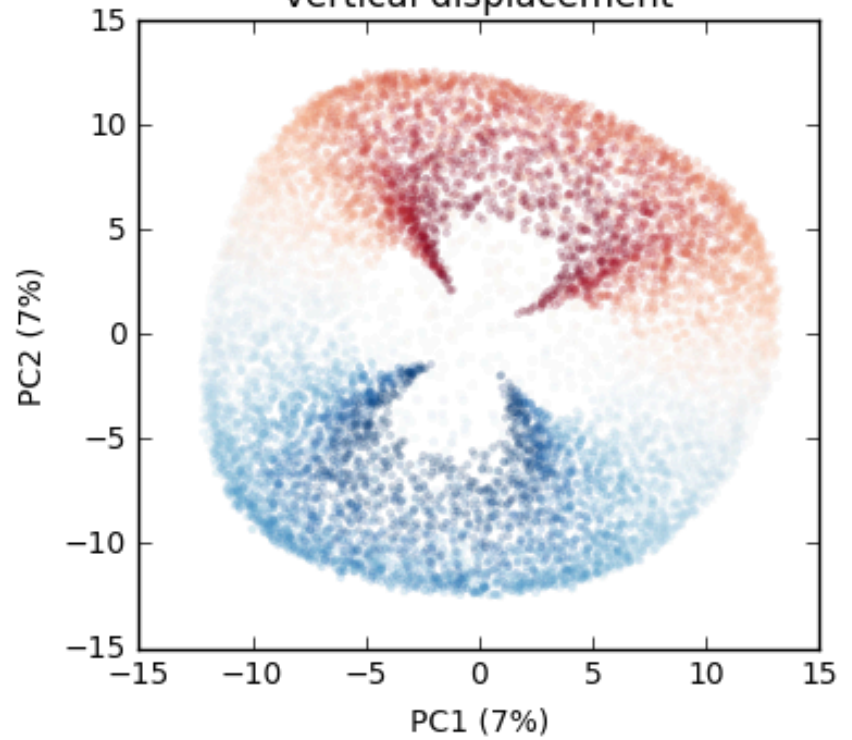
PCA



Colored by
horizontal displacement



Colored by
vertical displacement

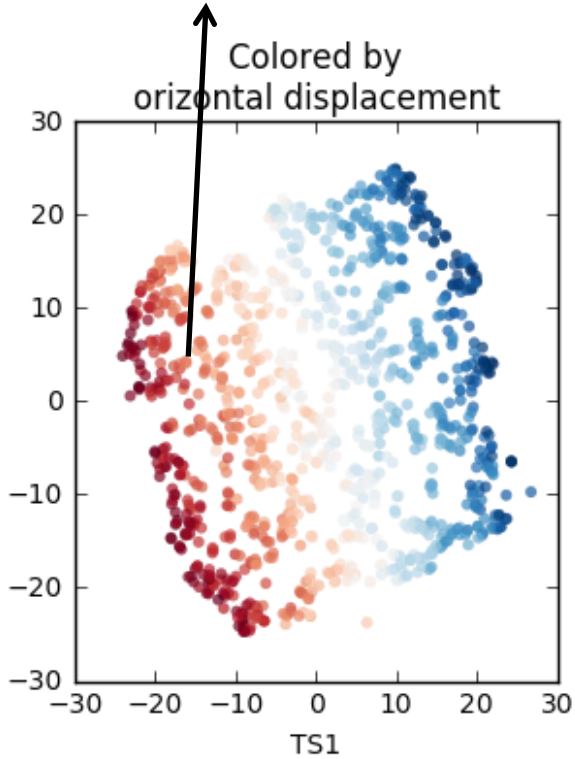


T-SNE

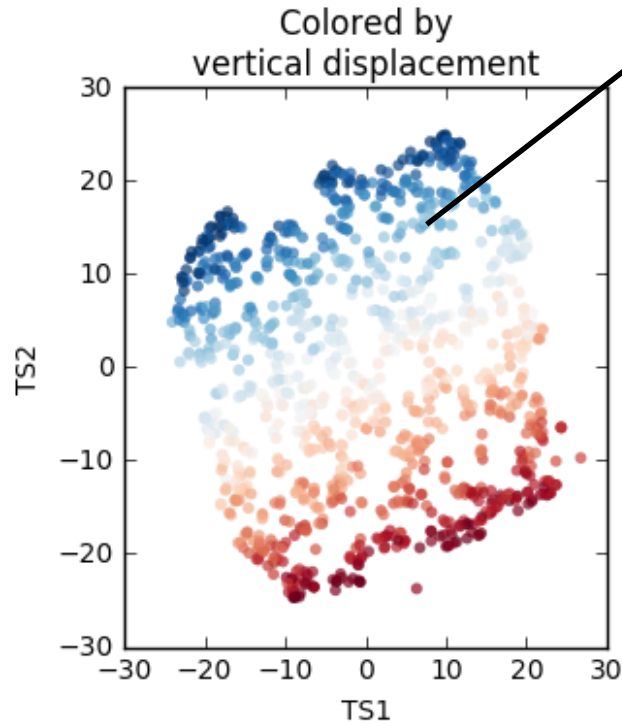
Manifold learning example



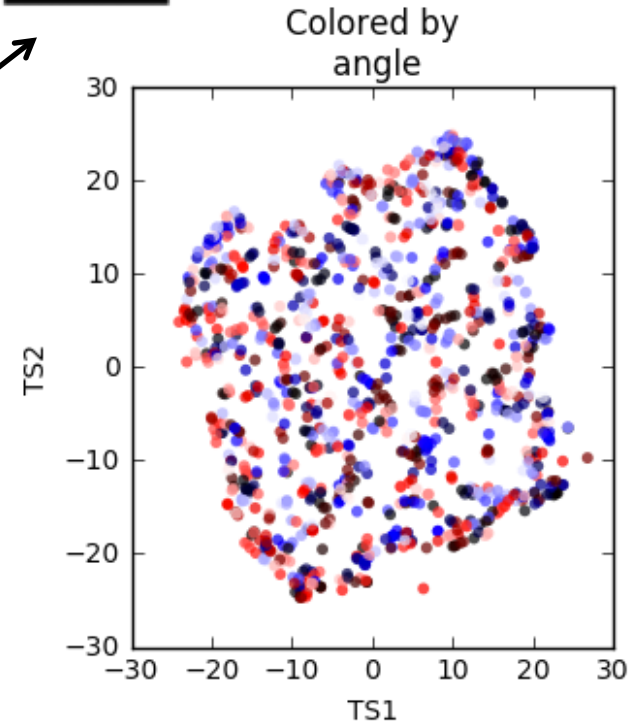
Colored by
horizontal displacement



Colored by
vertical displacement

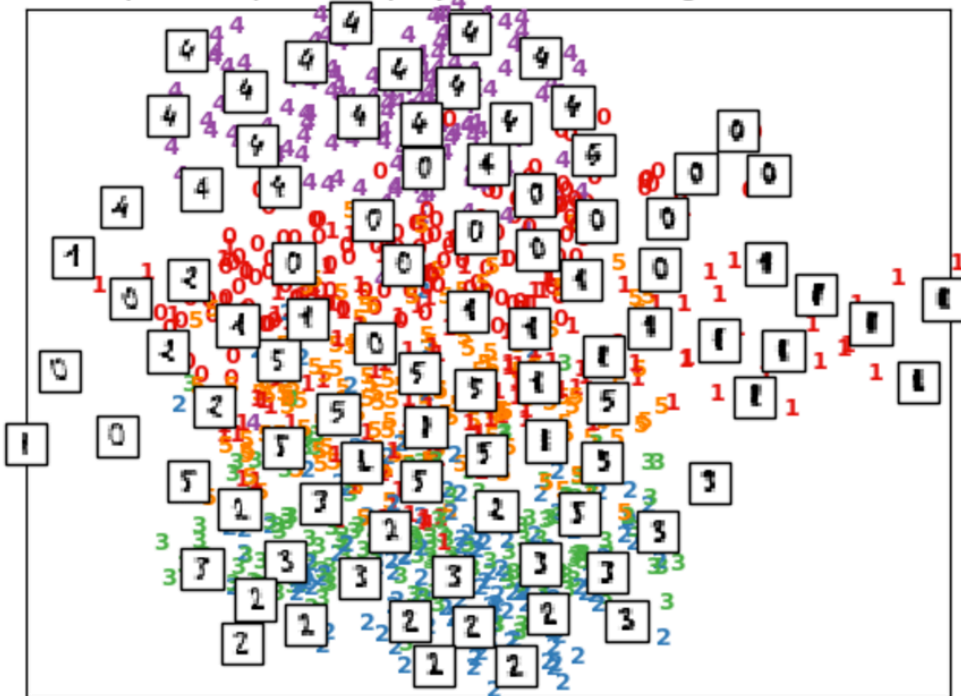


Colored by
angle

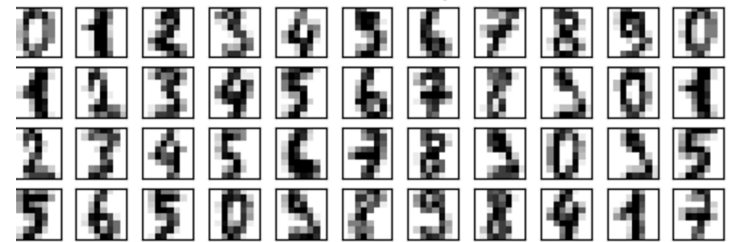


Linear mapping allows returning to the original space

Principal Components projection of the digits (time 0.01s)



Selection from the input data



"New" digits drawn from the kernel density model



Clustering

Definition?

Clustering

Cluster analysis is the task of partitioning the dataset into subsets, so that: the points in each subset are more similar to each other than those from different subsets

We need to define a distance metric:

Euclidean (L2 norm) $\sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \dots + (q_n - p_n)^2} = \sqrt{\sum_{i=1}^n (q_i - p_i)^2}$.

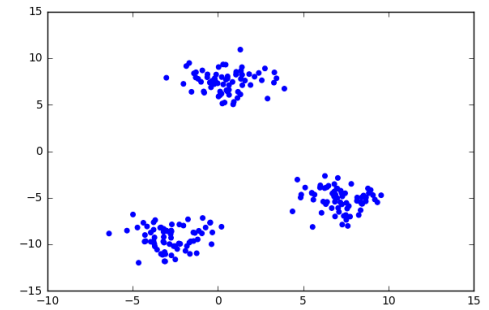
Manhattan (L1 norm) $\sum_{i=1}^n |p_i - q_i|$

Minkowski $(\sum_{i=1}^n |p_i - q_i|^c)^{1/c}$

Jaccard

NNotEQ / NNotZero

Clustering



Clustering algorithms are particularly important when:

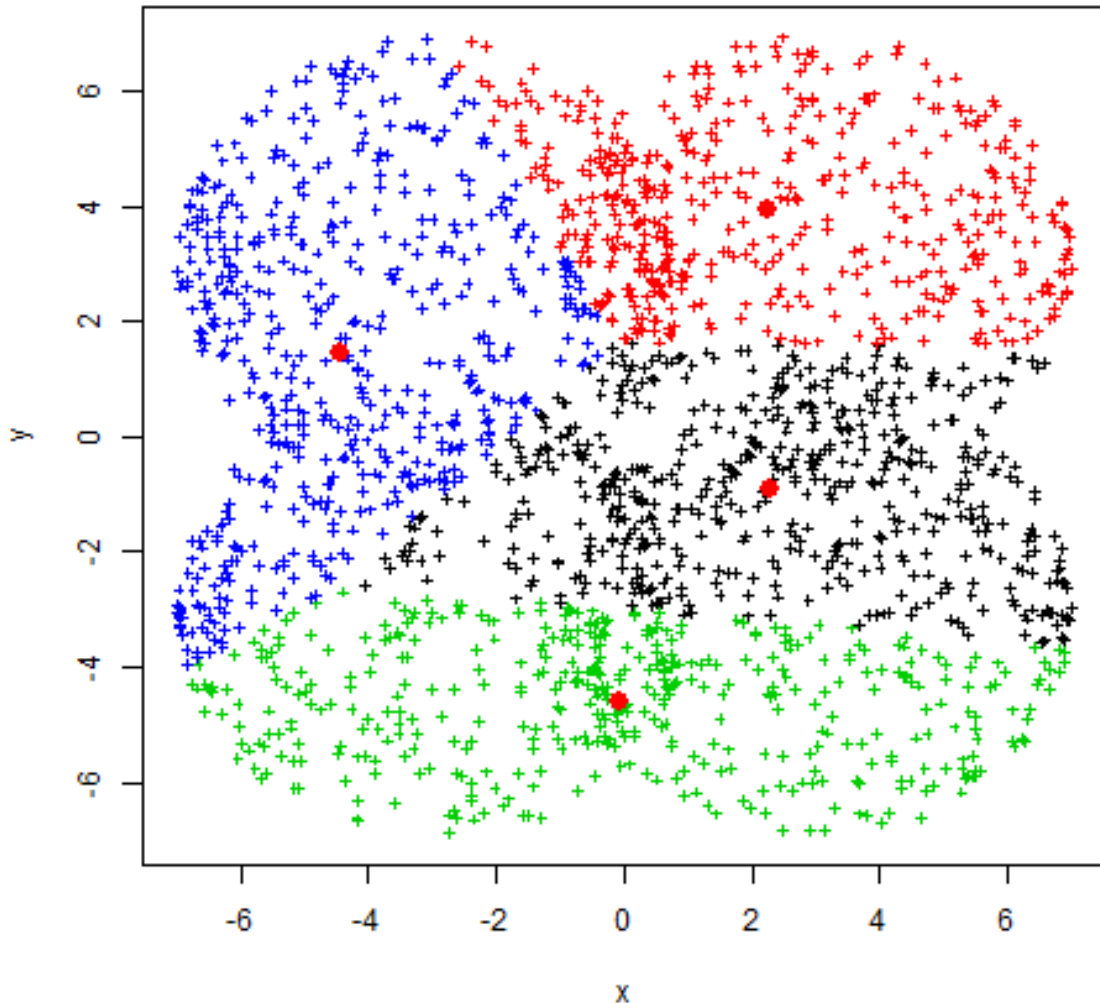
- You are in high dimensions
- Groups are difficult to visualize even in low dimensions
- You are in need of a statistical justification for grouping
- You need to automate things

Rule of thumb if none of the above:

Do it by hand!

K-means

K Means Clustering



Algorithm

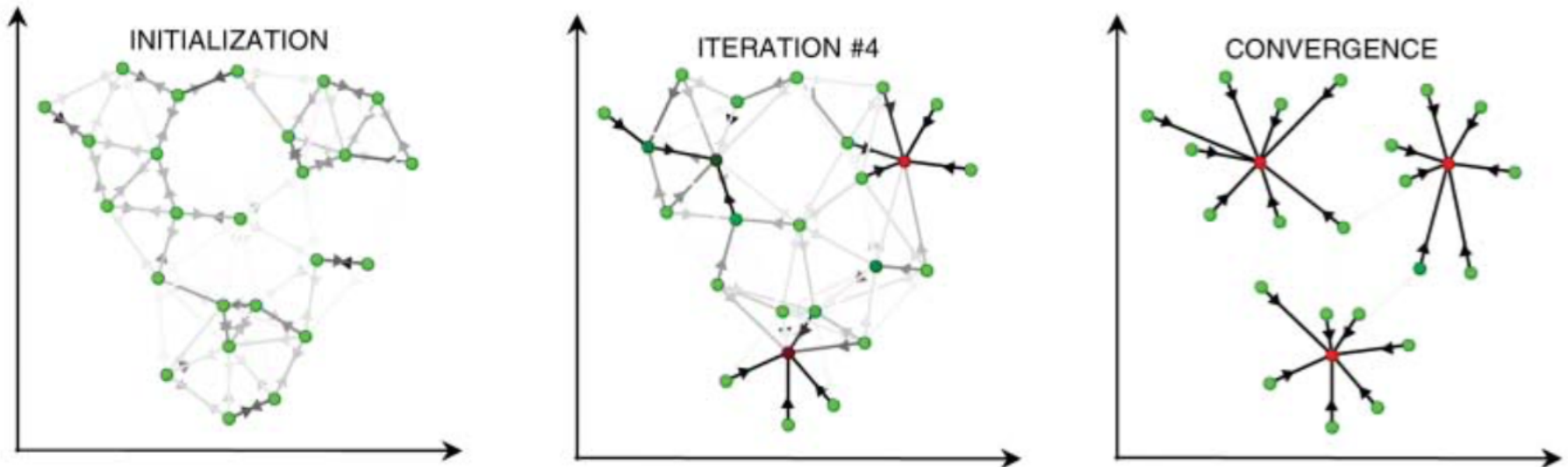
- Choose k centroids randomly.
- Calculate the distance from each point in the dataset to be classified to each centroid.
- Assign each point to the nearest centroid.
- Calculate the centroids of the resulting clusters.
- Repeat until the centroids don't move too much.

Affinity propagation

An algorithm where you do not set the number clusters

Message passing algorithm:

Every point is a candidate to become an “exemplar”
“preference”, “responsibility” and “availability”



Questions about
Dimensionality or clustering?

Regression

Regression is the problem of predicting a target value from an arbitrary input.

We are looking for a function that returns real values.

$$y = f(x)$$

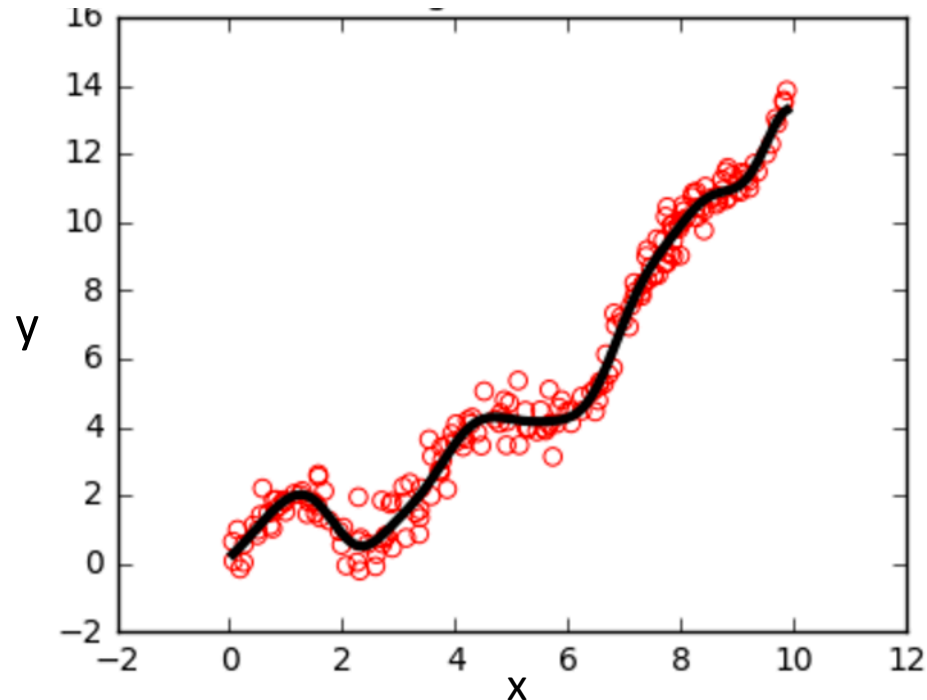


This is a number
(not a category)

Regression and curve fitting

This is basically **fitting** a function.

With the complication that we don't know the function.

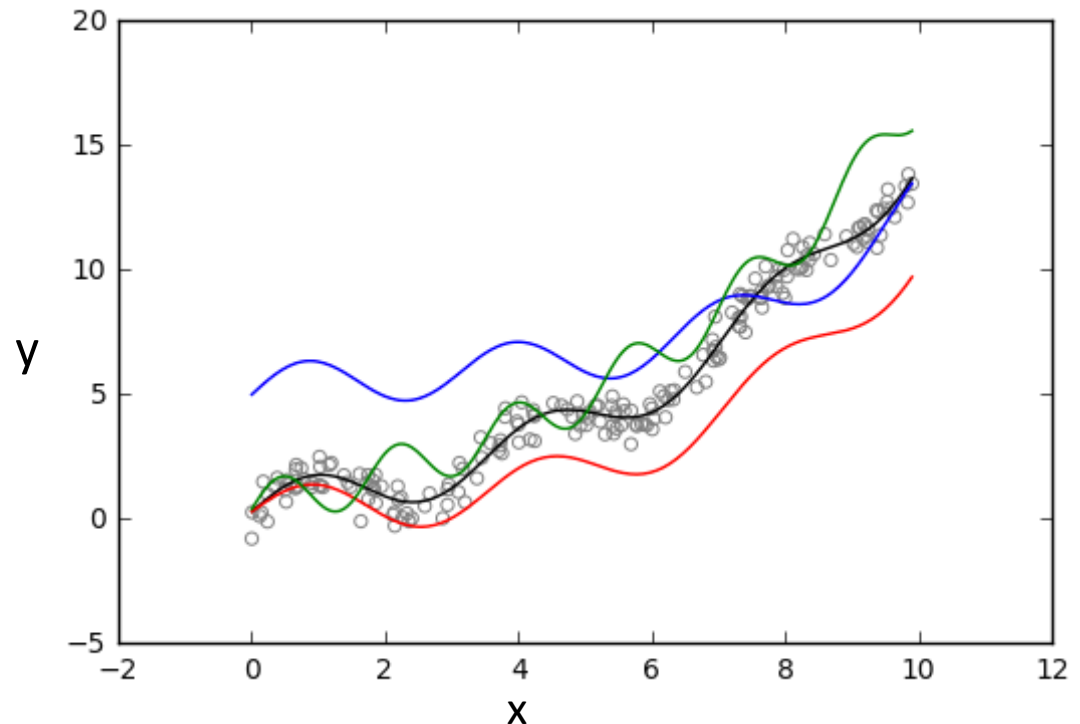


Regression and curve fitting

The situation would be different if we have some physical insight on the kind of function

$$f(x) = ax + bx^3 + \sin(cx) + d \cdot \cos(ex)$$

Reduction of the problem to finding the values of the parameters



Regression and curve fitting

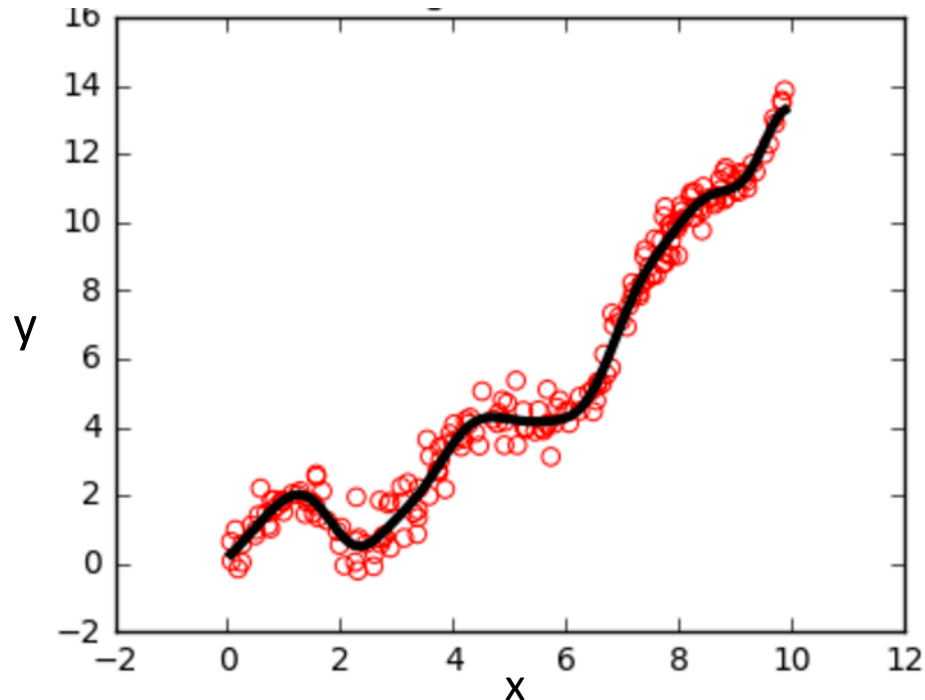
Optimization procedures are guaranteed to find the best values of the parameters

$$f(x) = ax + bx^3 + \sin(cx) + d \cdot \cos(ex)$$

$$\hat{f}(x) = 0.5x + 0.01x^3 + \sin(1.8x) + 0.3 \cos(0.2x)$$

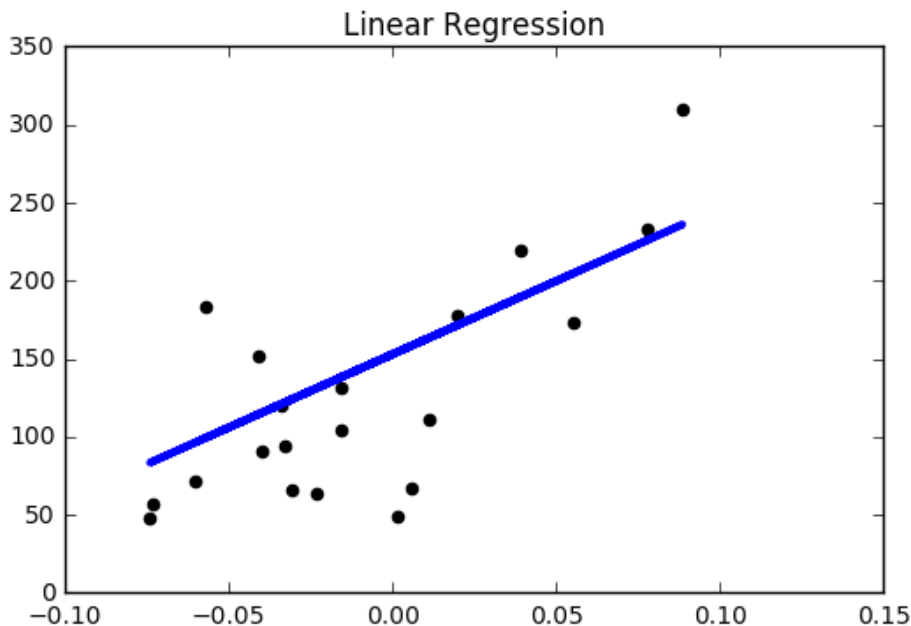
Minimize the sum of squares with respect to a, b, c, d, e

$$\frac{1}{N} \sum_{n=1}^N (\hat{f}(x_n) - y_n)^2$$



Linear regression

$$\mathbf{X} = \underbrace{\begin{bmatrix} -\mathbf{x}_1^\top \\ -\mathbf{x}_2^\top \\ \vdots \\ -\mathbf{x}_N^\top \end{bmatrix}}_{\text{input data matrix}}, \quad \mathbf{y} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}}_{\text{target vector}}.$$

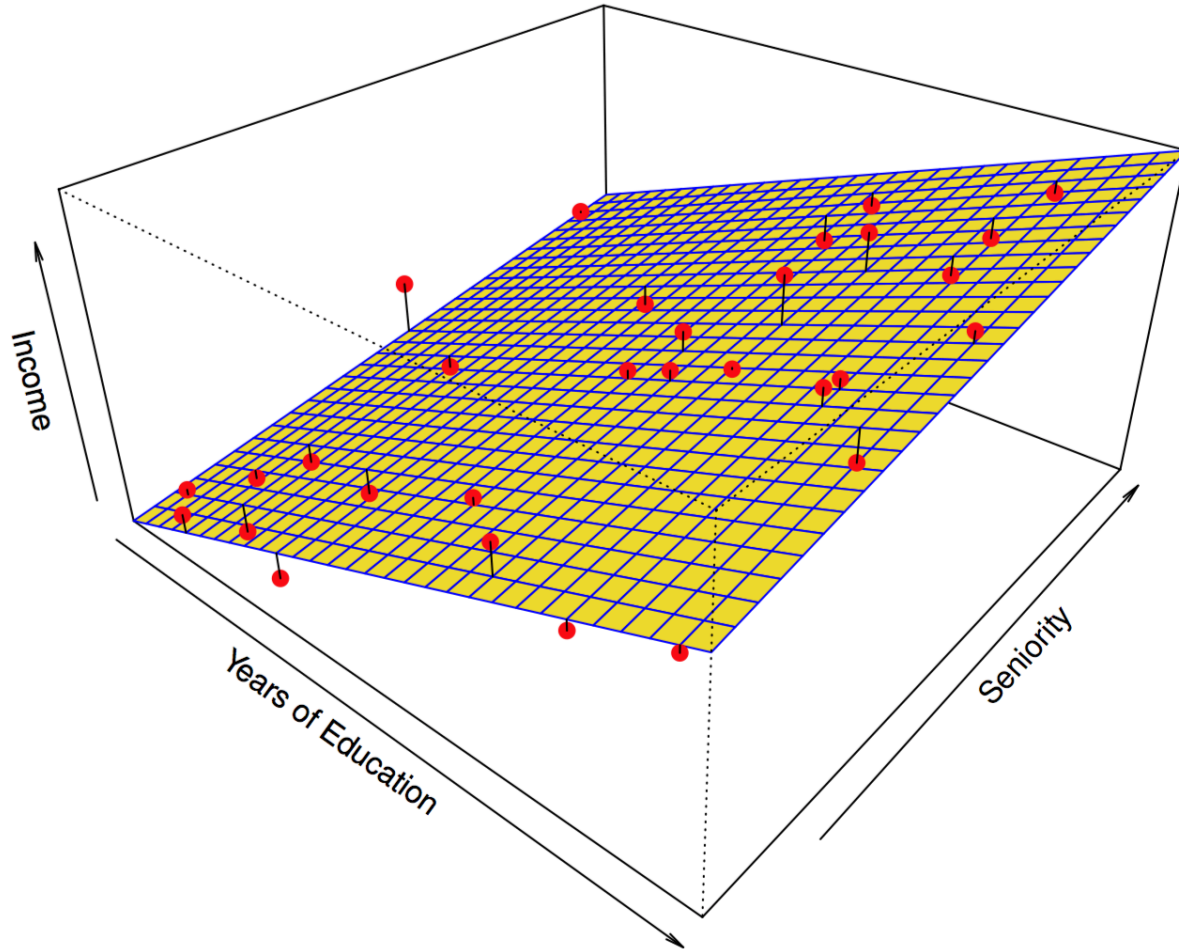


$$y = w_1^*x_1 + w_2^*x_2 + \dots + w_n^*x_n$$

Minimize the RSS

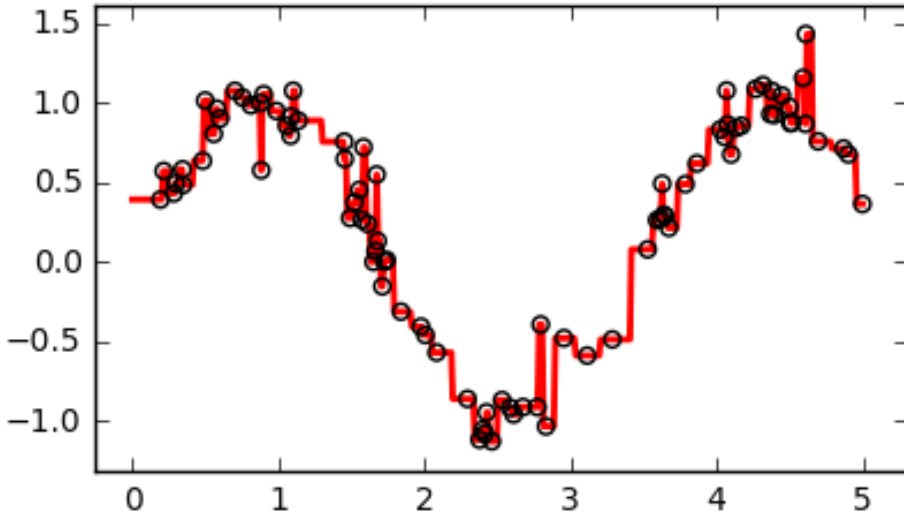
$$\min_w ||Xw - y||_2^2$$

Linear regression

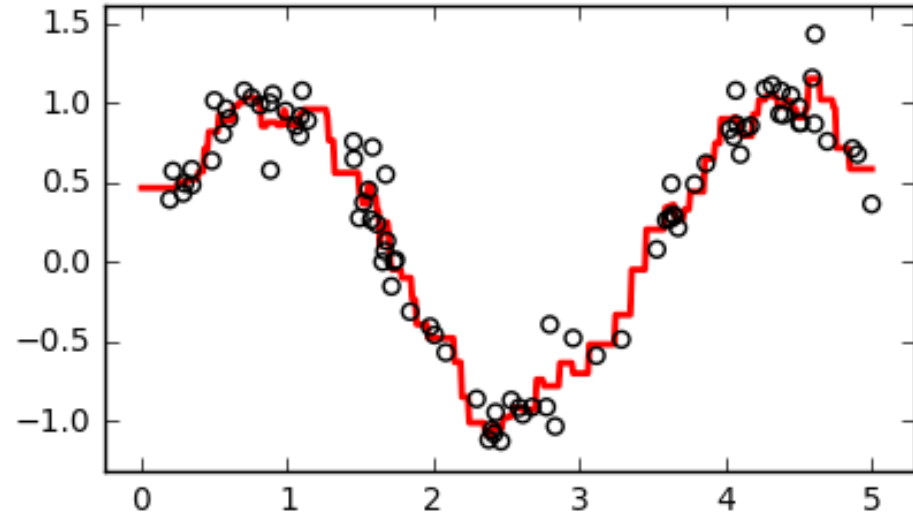


Nearest Neighbours regression

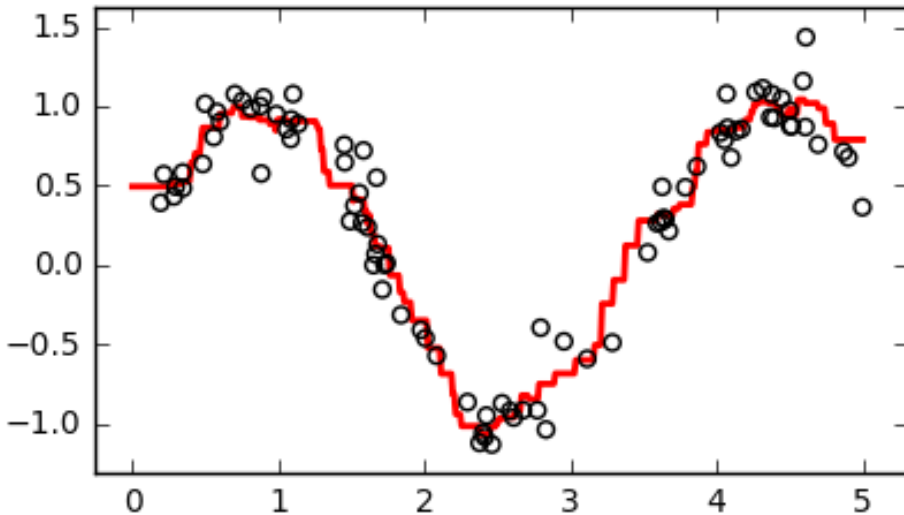
$k = 1$



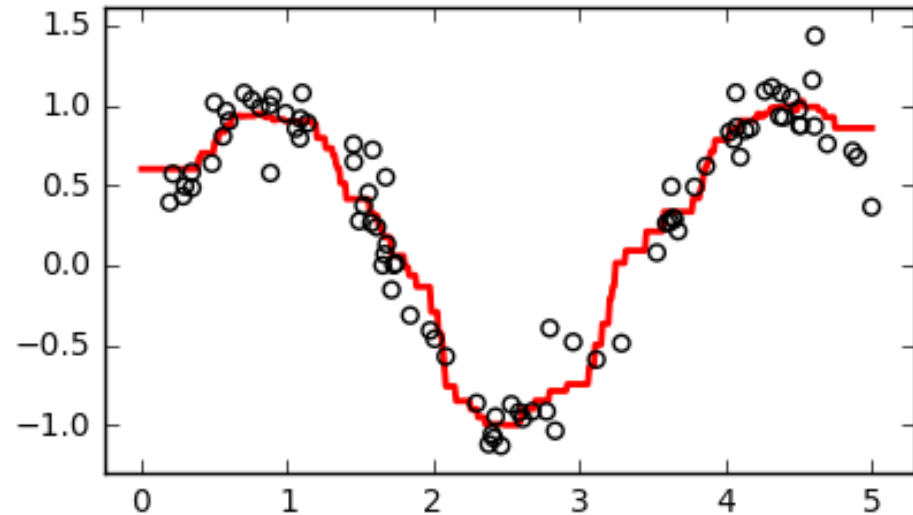
$k = 3$



$k = 5$



$k = 9$



Error of the model

Error that prevent supervised learning algorithms from generalizing beyond their training set:

Bias is error from erroneous assumptions in the learning algorithm.

High bias -> miss the relevant relations

Variance is error from sensitivity to small fluctuations in the training set.

High variance -> modeling the random noise in the training data

Bias-Variance decomposition

$$\mathbf{E}[(y - \hat{f}(x))^2] = \mathbf{Bias}[\hat{f}(x)]^2 + \mathbf{Var}[\hat{f}(x)] + \sigma^2$$

Where:

$$\mathbf{Bias}[\hat{f}(x)] = \mathbf{E}[\hat{f}(x) - f(x)]$$

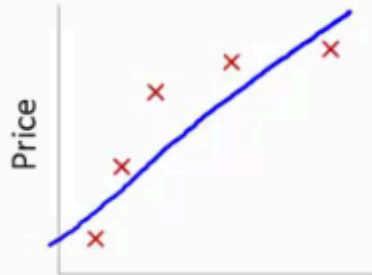
and

$$\mathbf{Var}[\hat{f}(x)] = \mathbf{E}[\hat{f}(x)^2] - \mathbf{E}[\hat{f}(x)]^2$$

The expectation ranges over different choices of the training set:

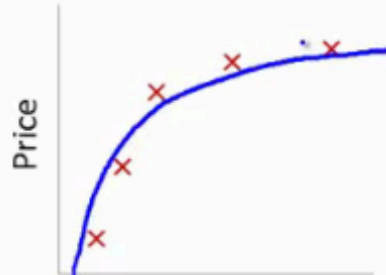
$x_1, x_2, x_3, \dots, x_n$ and $y_1, y_2, y_3, \dots, y_n$

All sampled from the same joint distribution



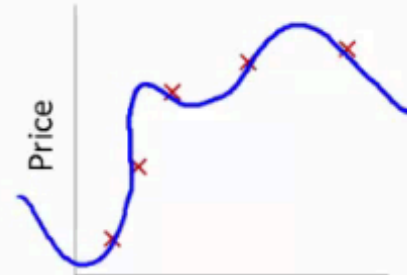
Size
 $\theta_0 + \theta_1x$

High bias
 (underfit)



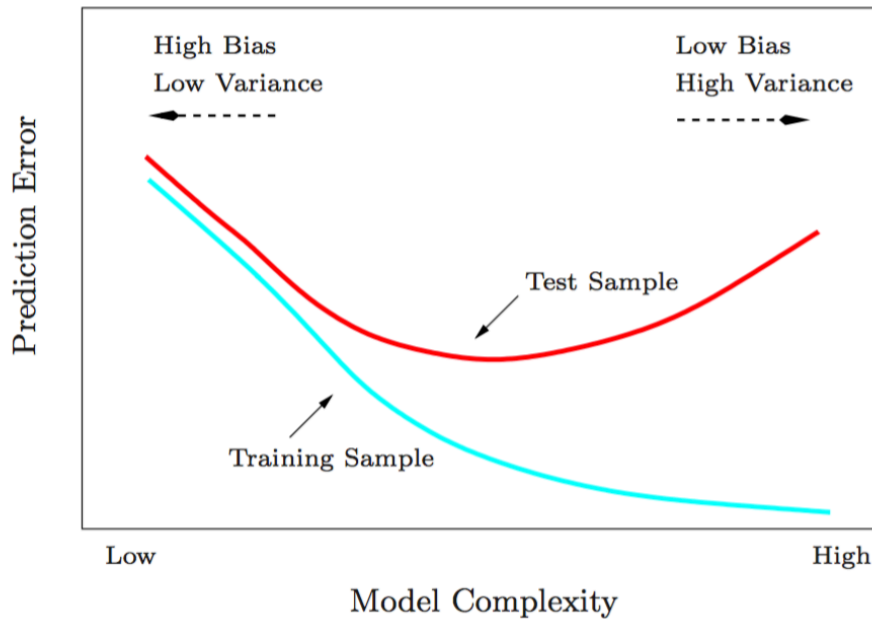
Size
 $\theta_0 + \theta_1x + \theta_2x^2$

"Just right"

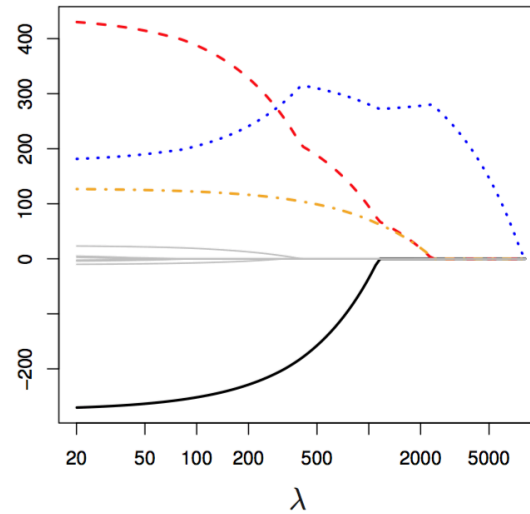
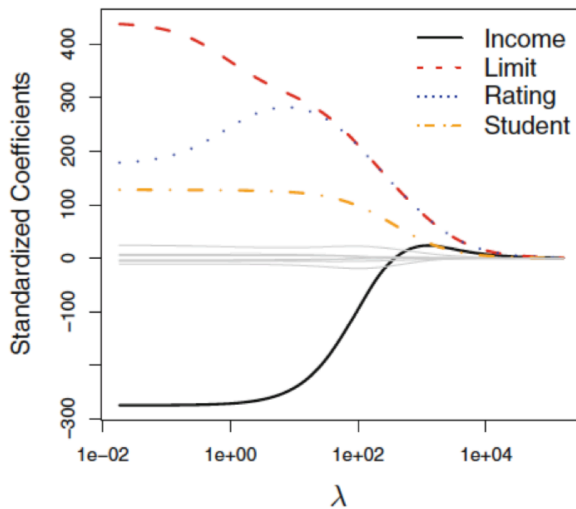


Size
 $\theta_0 + \theta_1x + \theta_2x^2 + \theta_3x^3 + \theta_4x^4$

High variance
 (overfit)



Regularized regression



Ridge regression

$$RSS + \lambda \sum_{i=1}^d w_i^2$$

Lasso regression

$$RSS + \lambda \sum_{i=1}^d |w_i|$$

Not only about predictions but also coefficients

Most often models are not a black box.

- Feature selection
- Insight on the data

$$y = w_1 * X_1 + w_2 * X_2 + w_3 * X_3 + w_4 * X_4 + w_5 * X_5$$

Weighs	2.4	0.4	-1.7	0.1	0.9
--------	-----	-----	------	-----	-----

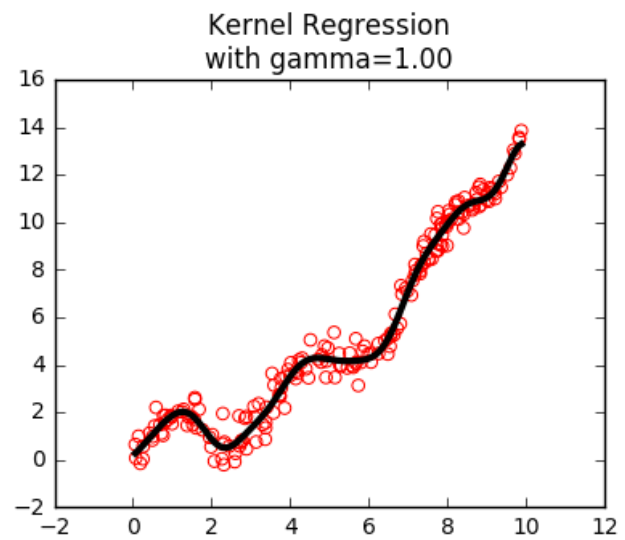
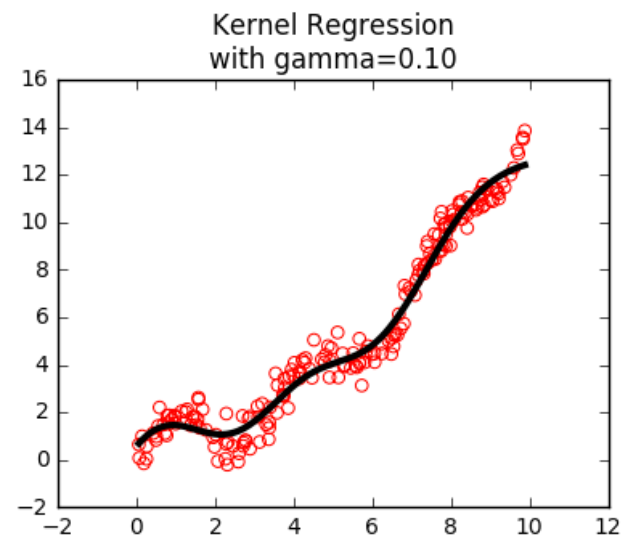
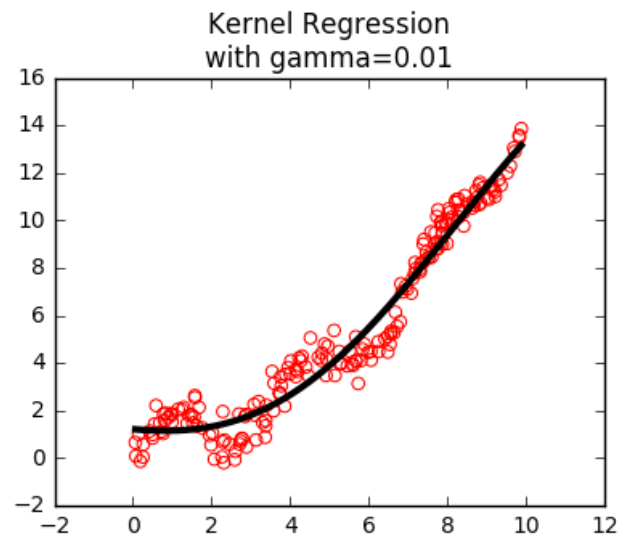
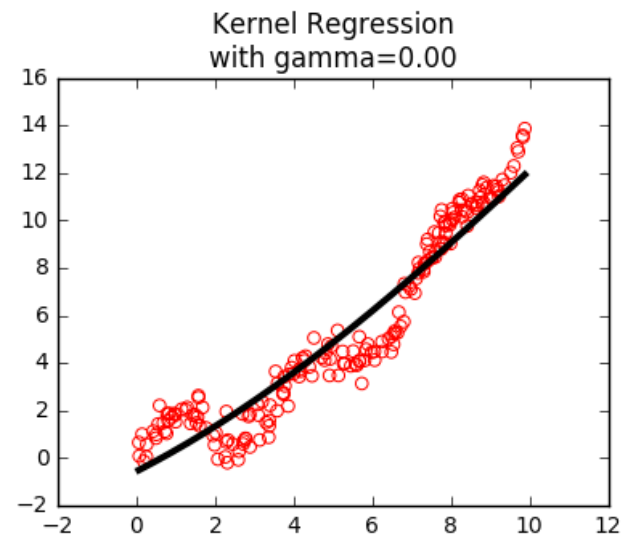
Weighs Lasso	3.2	0	-2.1	0	0
-----------------	-----	---	------	---	---

Multidimensional and multitarget



Kernel based methods For Regression

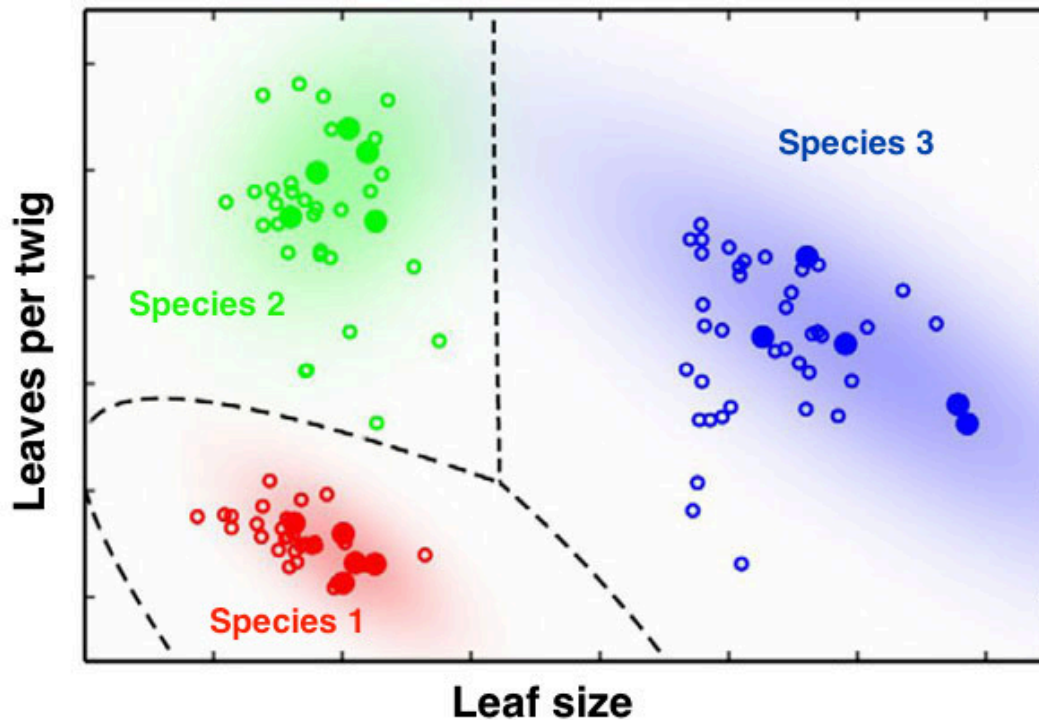
Kernel trick:
Artificially increase the
Dimensions



Include:
Kernel Ridge
Support Vector Regression
Gaussian Processes

Classification

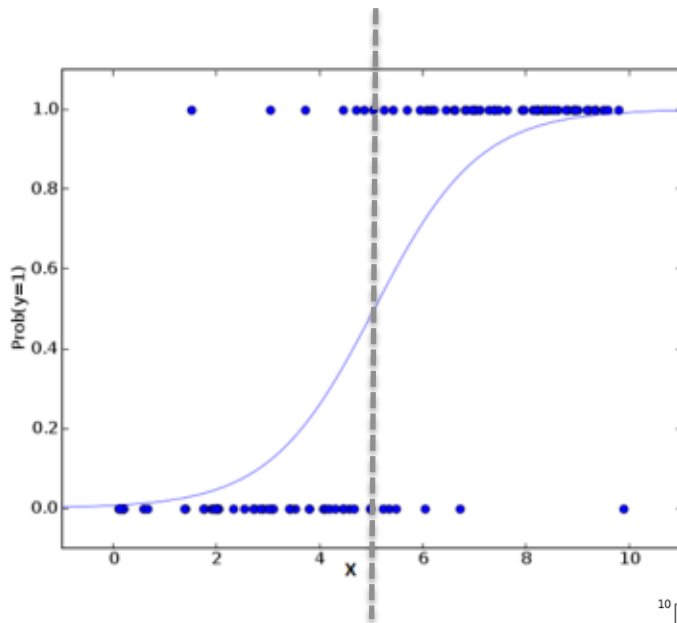
Identifying to which of a set of categories a new observation belongs on the basis of observations whose category membership is known (training set)



Logistic Regression

Logistic regression, despite its name, is a linear model for classification rather than regression

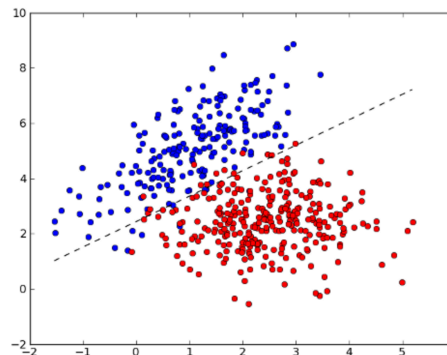
Logistic regression in 1 dimension



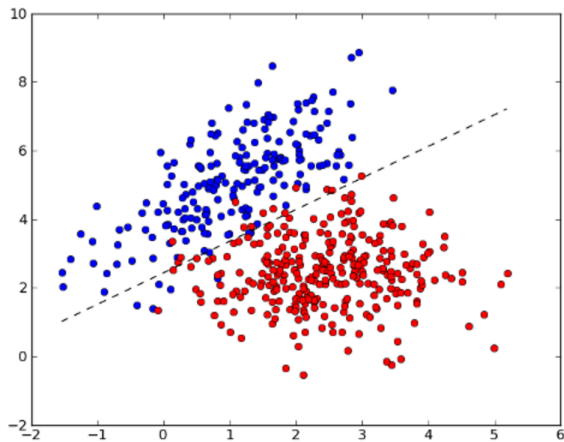
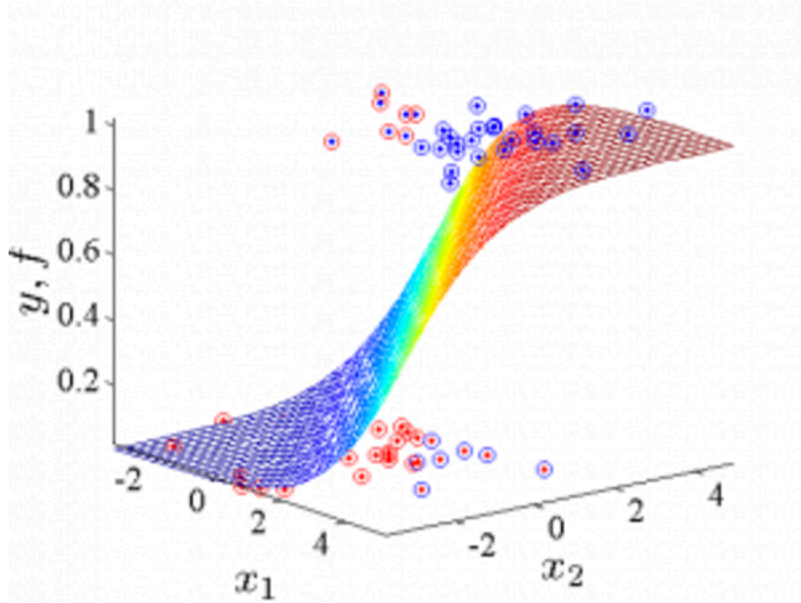
$$\min_{w,c} \frac{1}{2} w^T w + C \sum_{i=1}^n \log(\exp(-y_i(X_i^T w + c)) + 1).$$

$$a = w_1 * x_1 + w_2 * x_2 + \dots + w_n * x_n$$

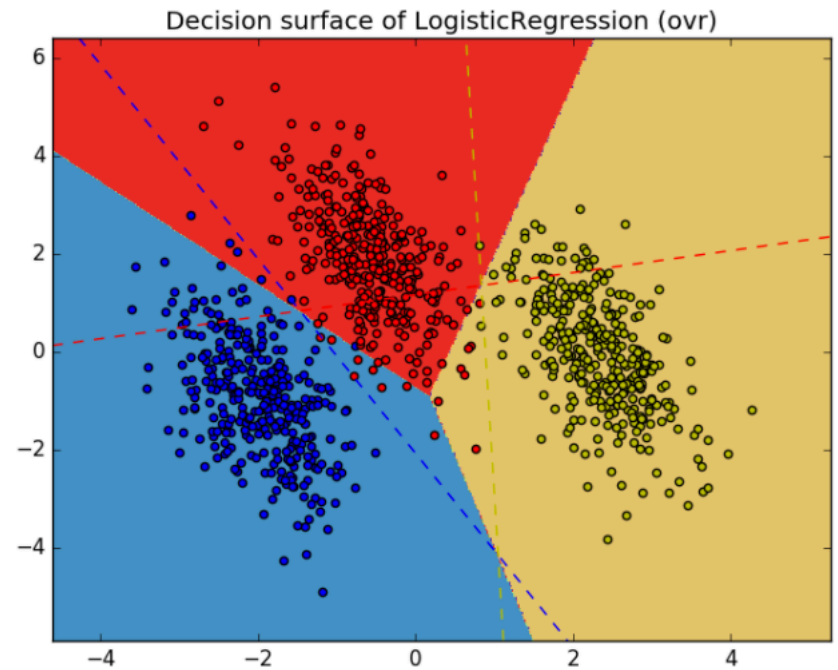
$$y = \text{Sigmoid}(a)$$



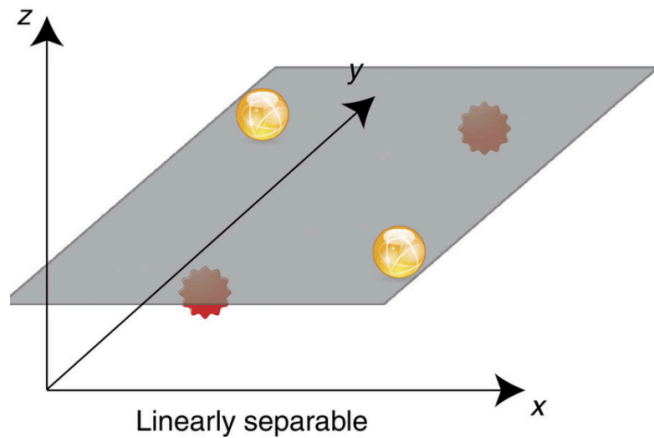
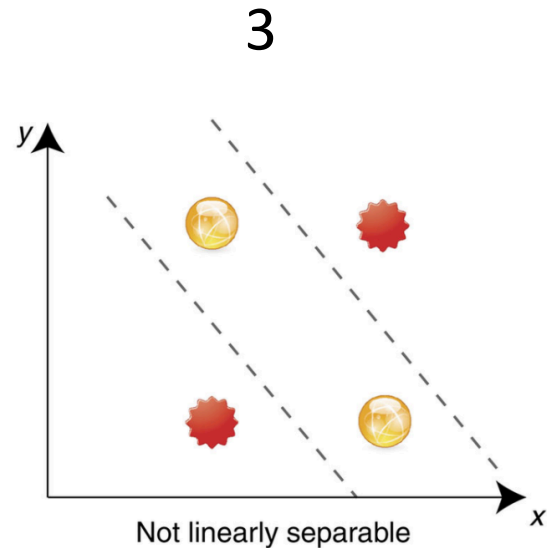
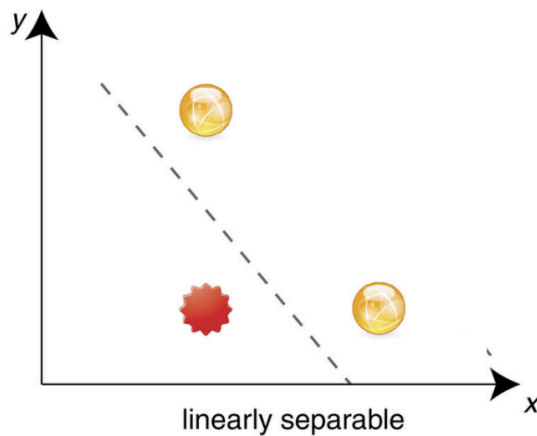
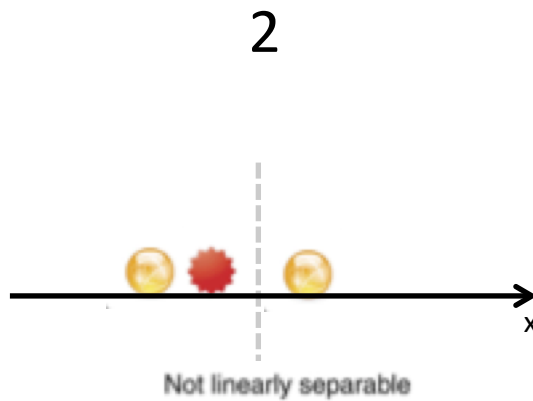
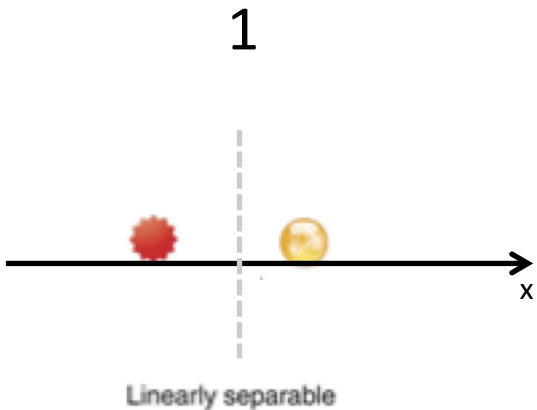
Logistic Regression



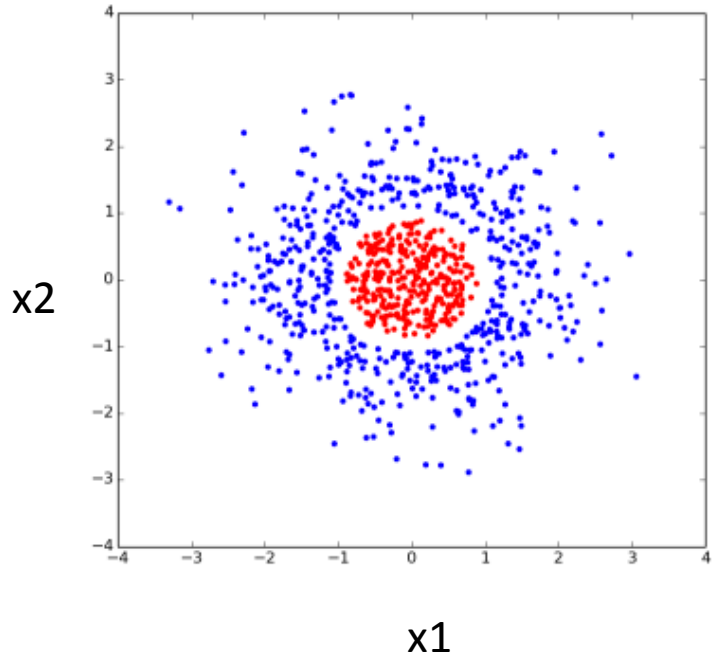
Generalization to Multiple classes



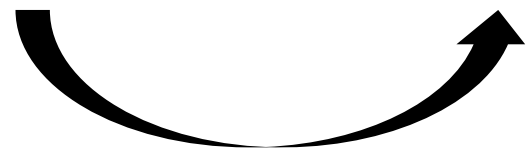
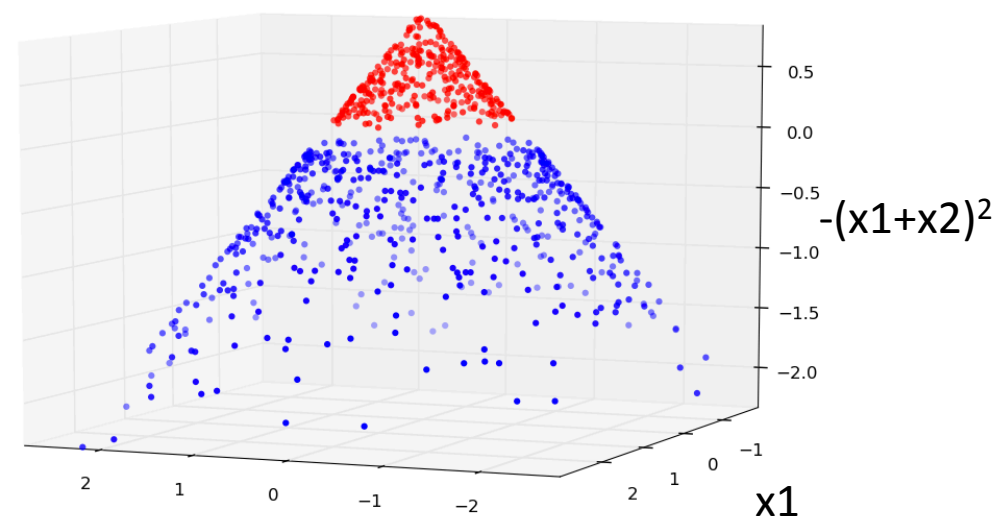
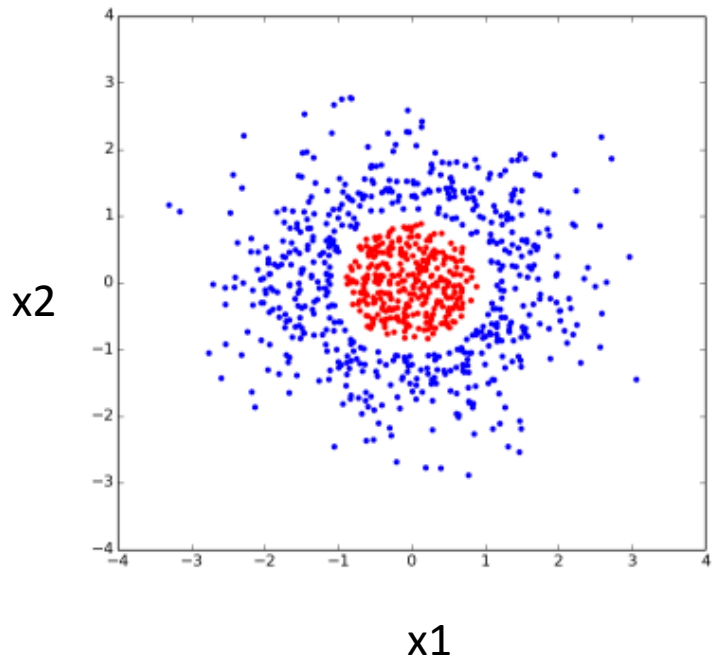
Trivially
N data points are
linearly separable
in **N-1** dimensions



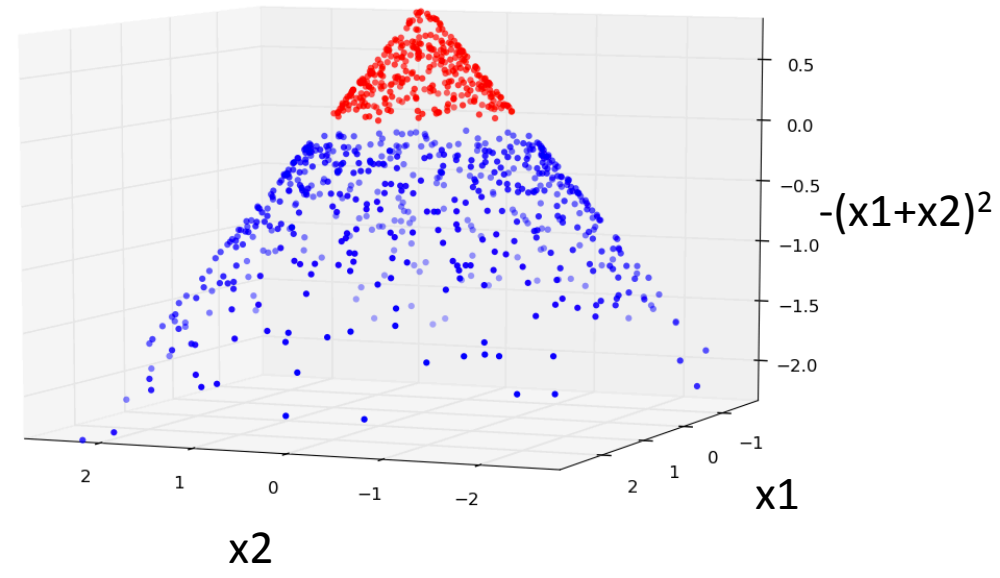
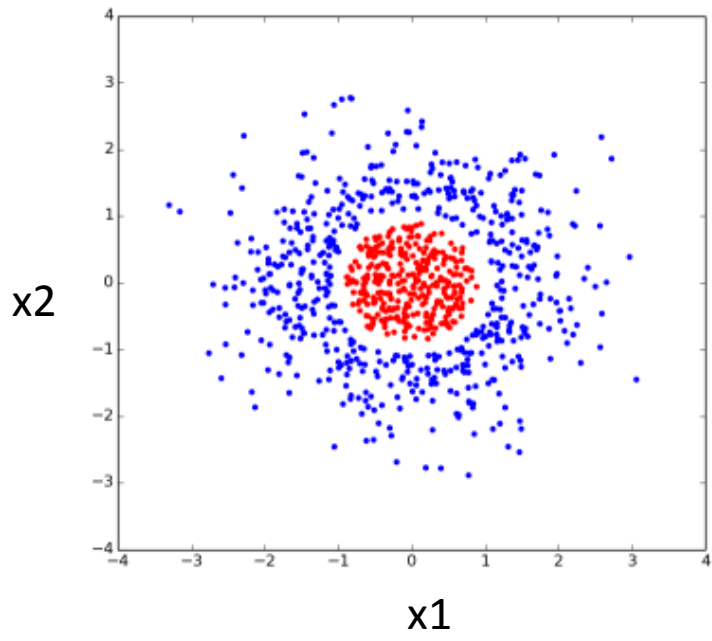
Let's bring everything in high dimensions!
Is it a stupid idea?



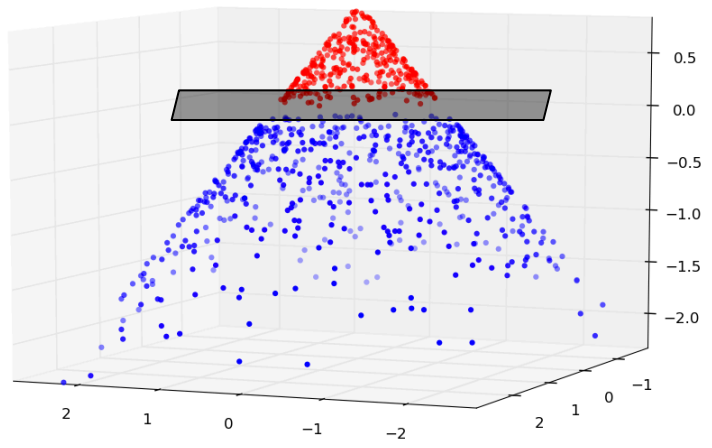
What do I do in this case?



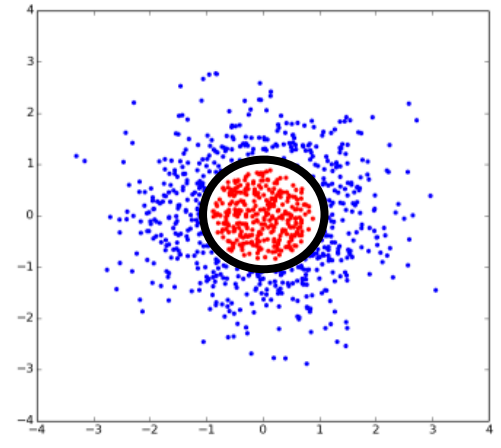
Bring to higher dimensions
Using nonlinear function of
The original dimensions



Learn dividing plane
in high dimensions



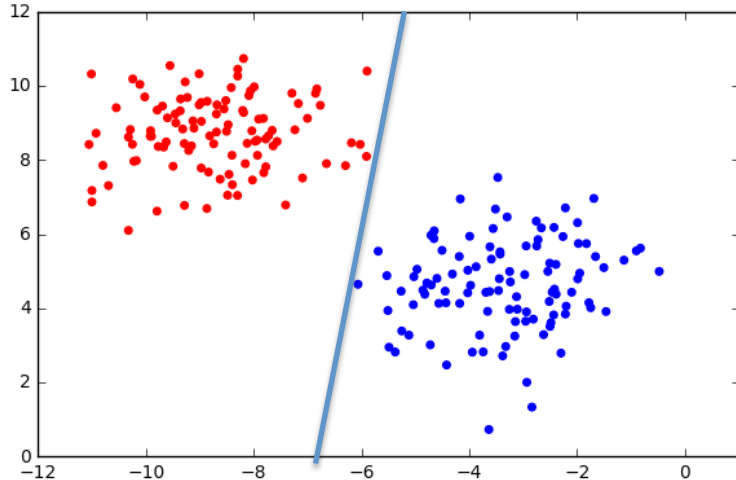
Corresponds to
this non linear
boundary



Support vector machines (classification)

In high dimensions everything is far from each other and therefore is easier that things
Become linearly separable.

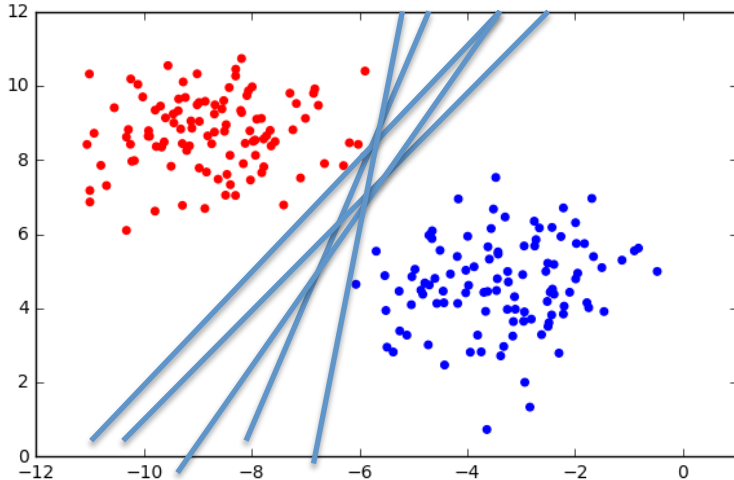
Different possible boundaries:



Support vector machines (classification)

In high dimensions everything is far from each other and therefore is easier that things
Become linearly separable.

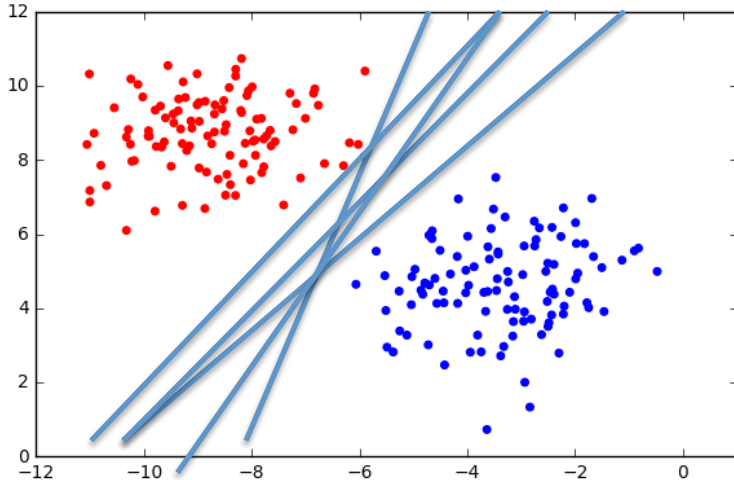
Different possible boundaries:



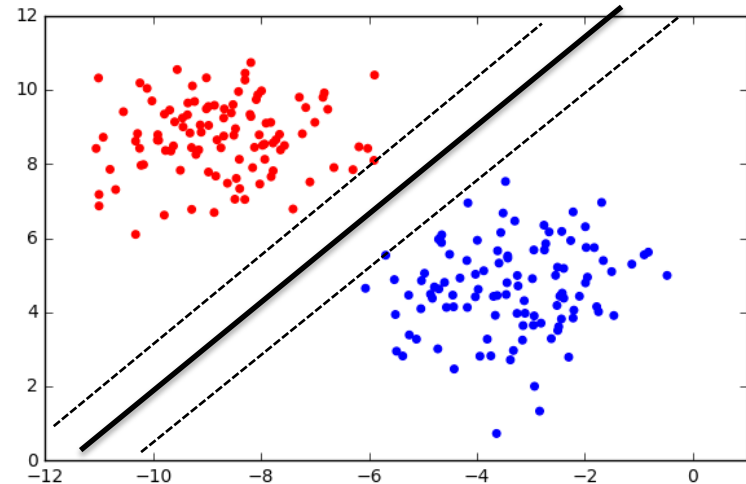
Support vector machines (classification)

In high dimensions everything is far from each other and therefore is easier that things
Become linearly separable.

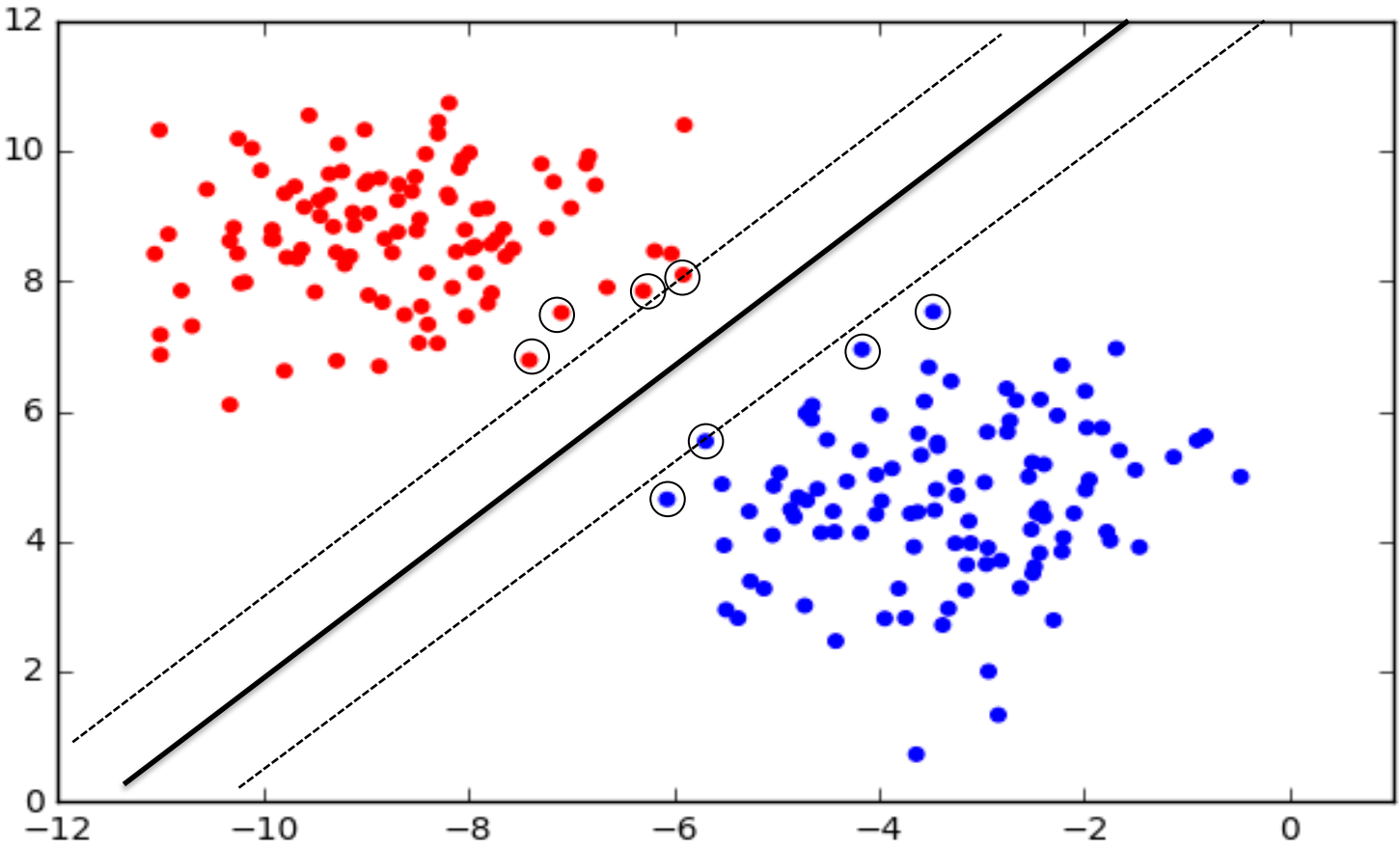
Different possible boundaries:



Max margin boundary



The points close to the margin are the one that dictate its position: “support vectors”

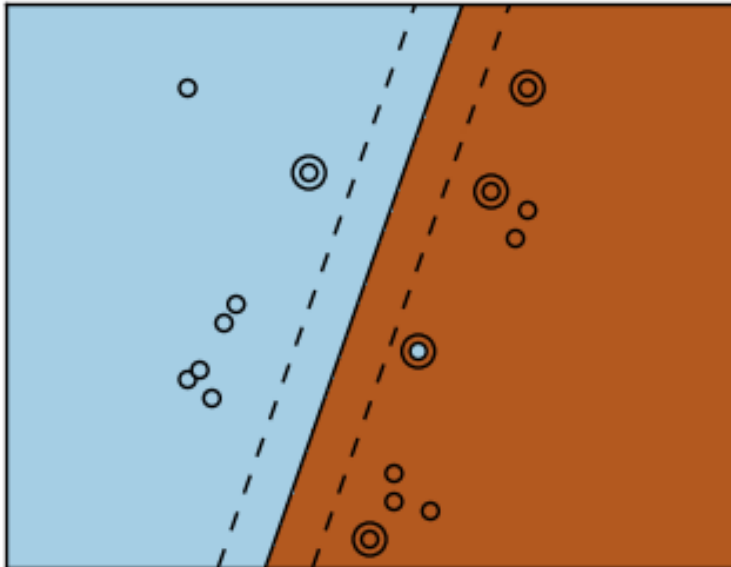


Kernel Trick

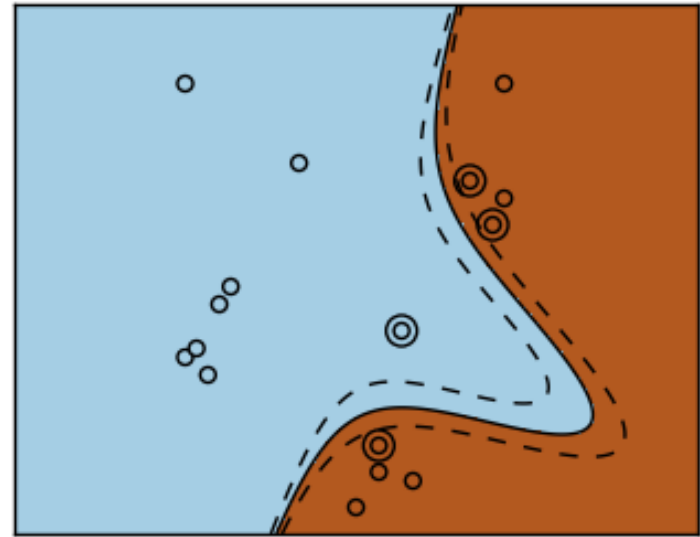
$$\min_{\vec{w}, b} \|\vec{w}\|$$

$$t_i(\vec{w}^T \cdot \phi(\vec{x}_i) + b) \geq 1$$

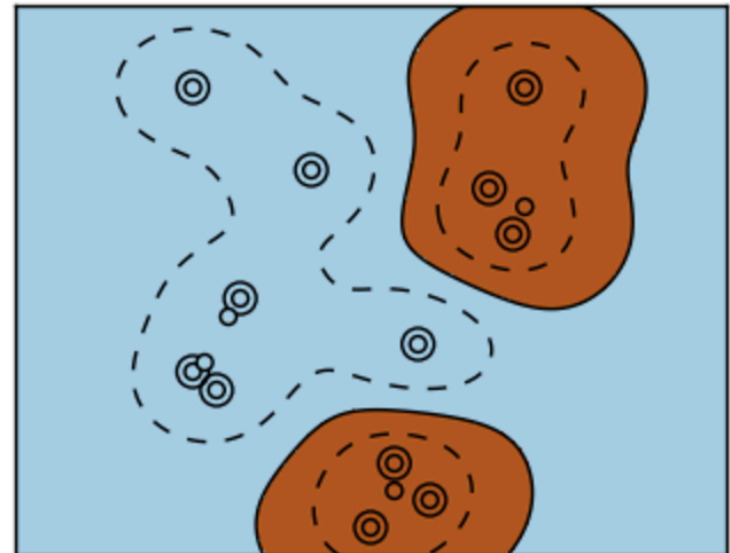
Linear kernel



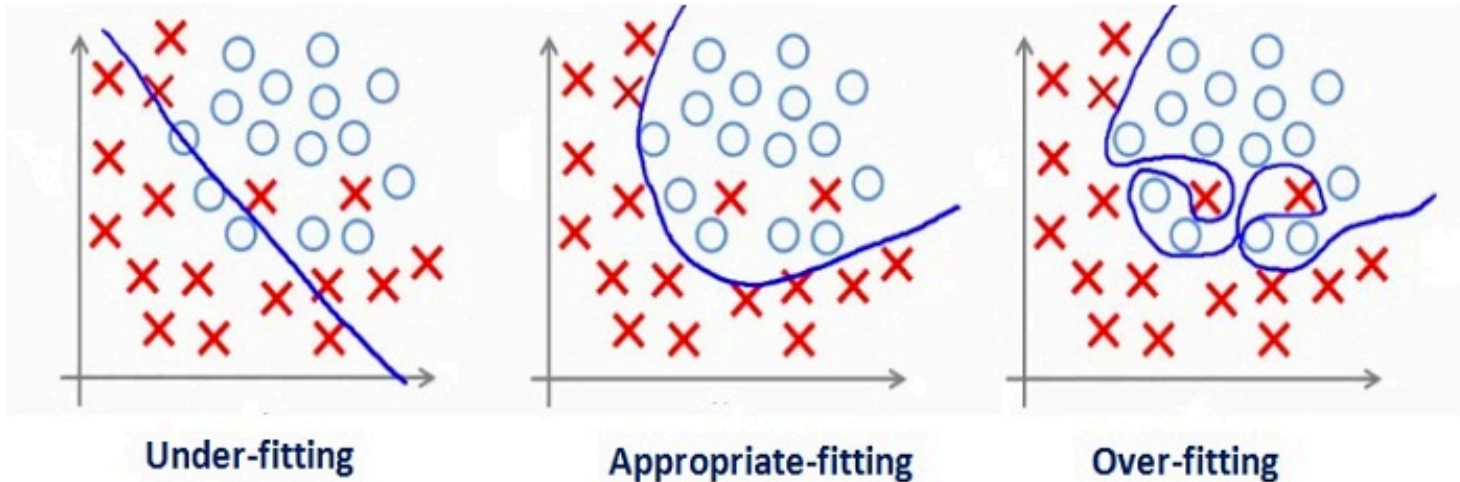
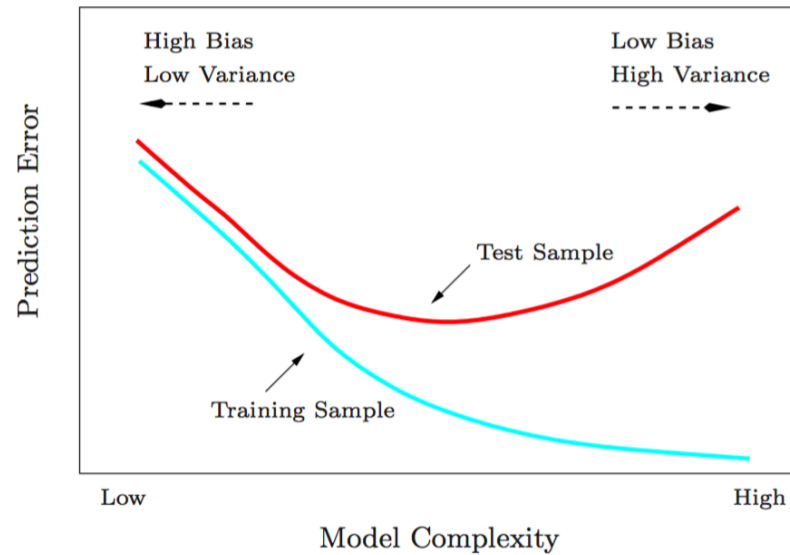
Polynomial kernel



Gaussian kernel

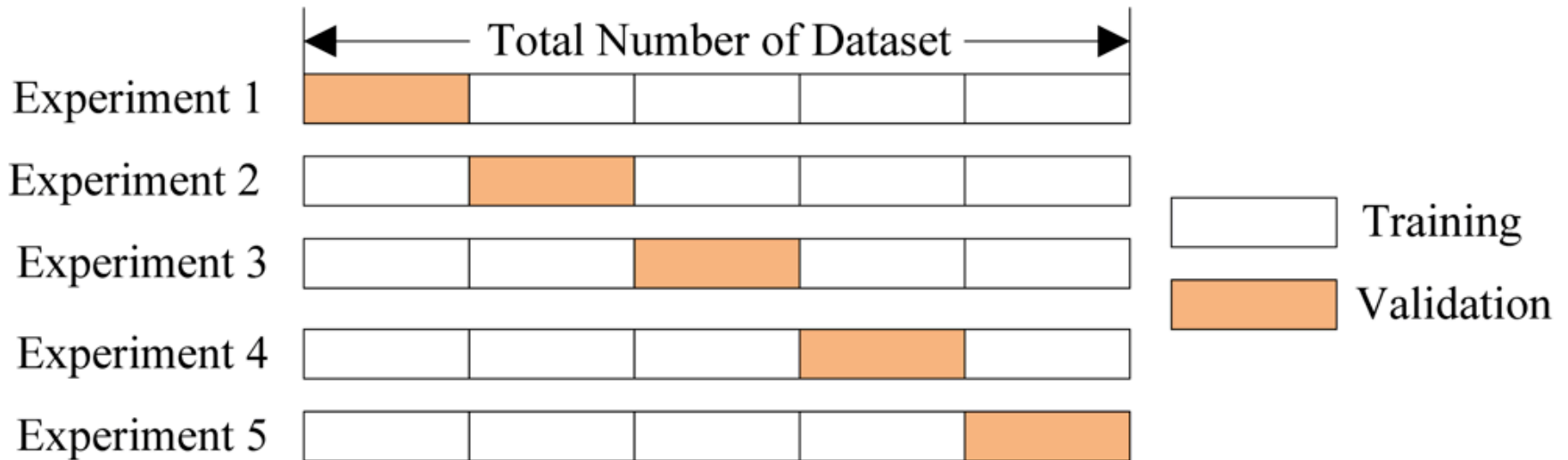


Underfitting and Overfitting



Cross-validation

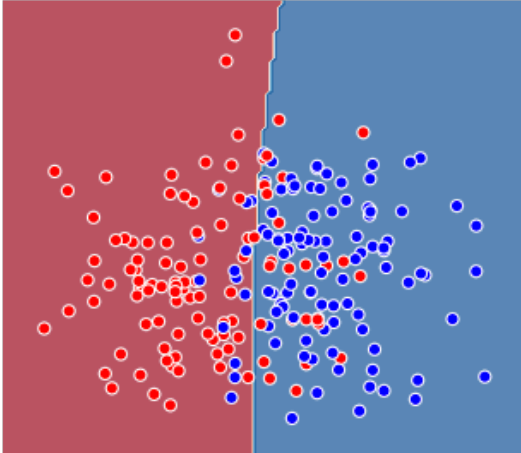
CV is a model validation technique for assessing how the results will generalize to an independent data set.



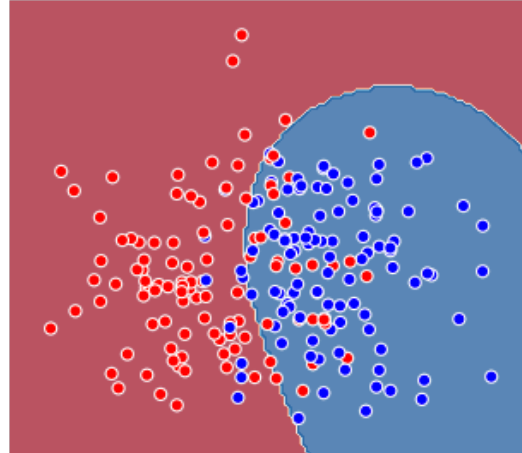
Cross-validation

It is both a way to evaluate and to select a model

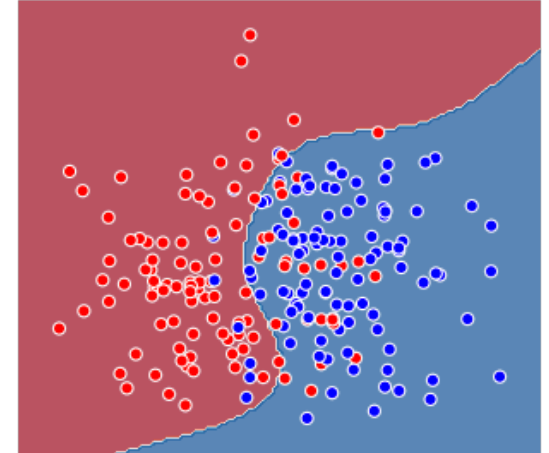
regularization=200.00
train-set score=0.810
CV score=0.771



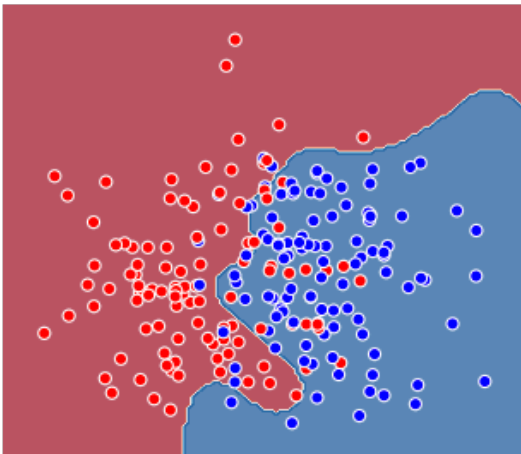
regularization=20.00
train-set score=0.824
CV score=0.795



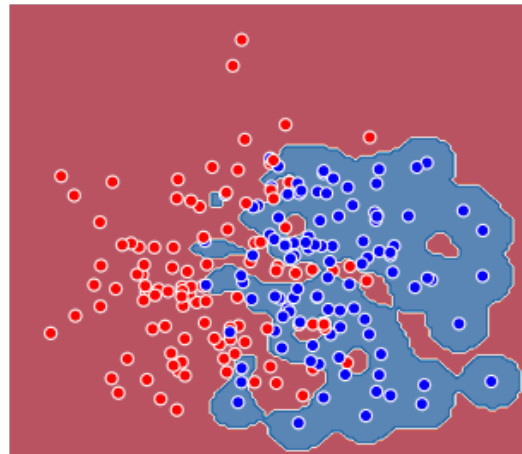
regularization=2.50
train-set score=0.838
CV score=0.809



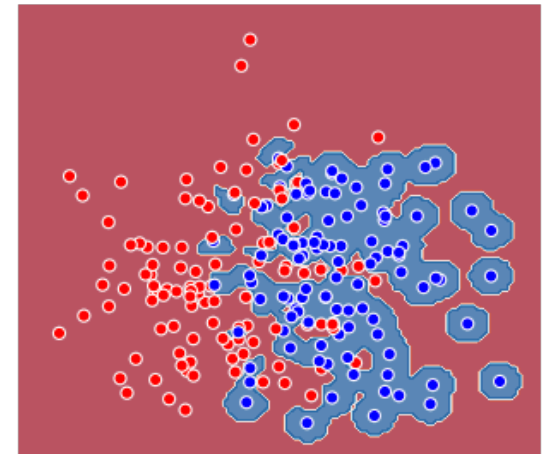
regularization=0.50
train-set score=0.838
CV score=0.781



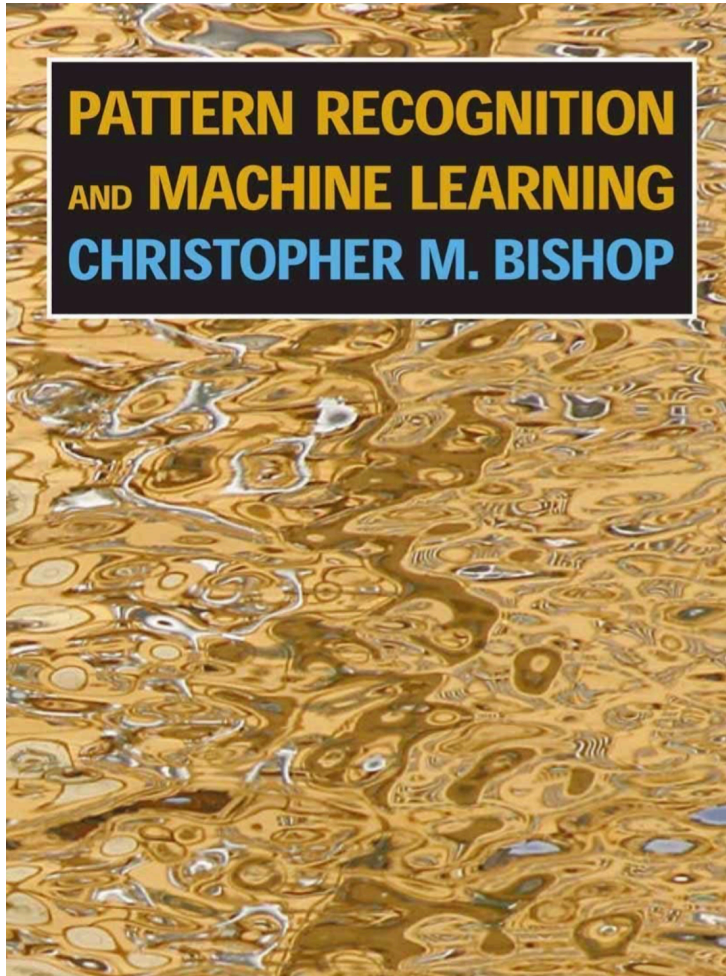
regularization=0.05
train-set score=0.957
CV score=0.699



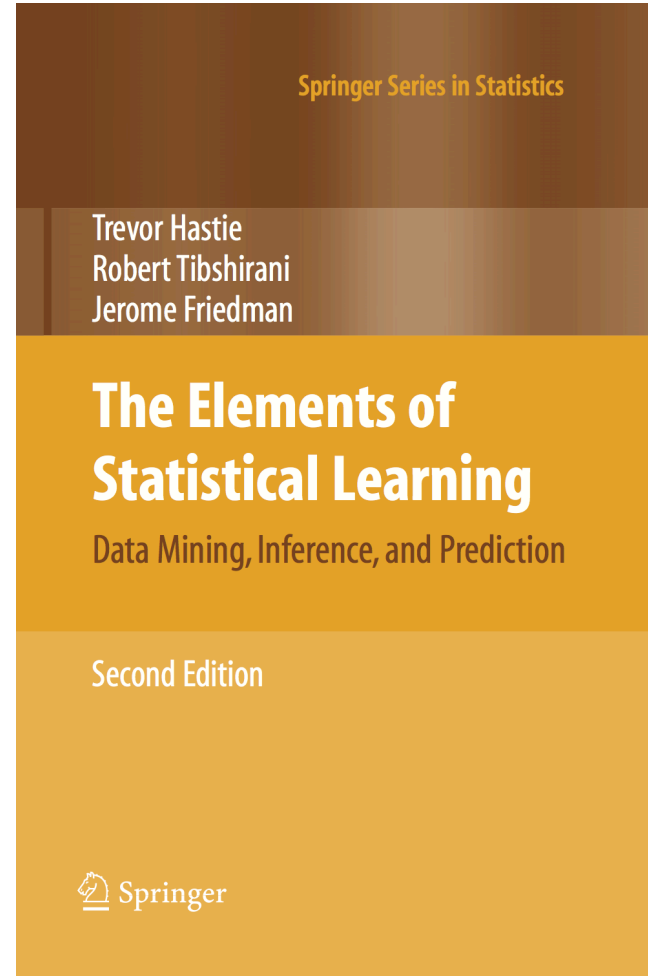
regularization=0.01
train-set score=0.995
CV score=0.647



Learning more about learning



Code implementations available @
<https://github.com/PRML/PRMLT>



Made available for FREE by the authors @:
<https://web.stanford.edu/~hastie/ElemStatLearn/>