Introduction to Statistical Learning – part 2

Outline

- Figues of merit
- Introduction to basic classifiers
- Complexity control
- Dimensionality Reduction
- Regularization

LOSS FUNCTIONS AND FIGURES OF MERIT

Loss Functions and Empirical Risk

- **Let us consider a binary classification problem**
- **Aim: To estimate a function:**

$$f : \Re^N \rightarrow \{\pm 1\} \qquad (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l) \in \Re^N \times \{\pm 1\}$$

- **The best function f is the one that minimizes the loss function or Expected Risk**

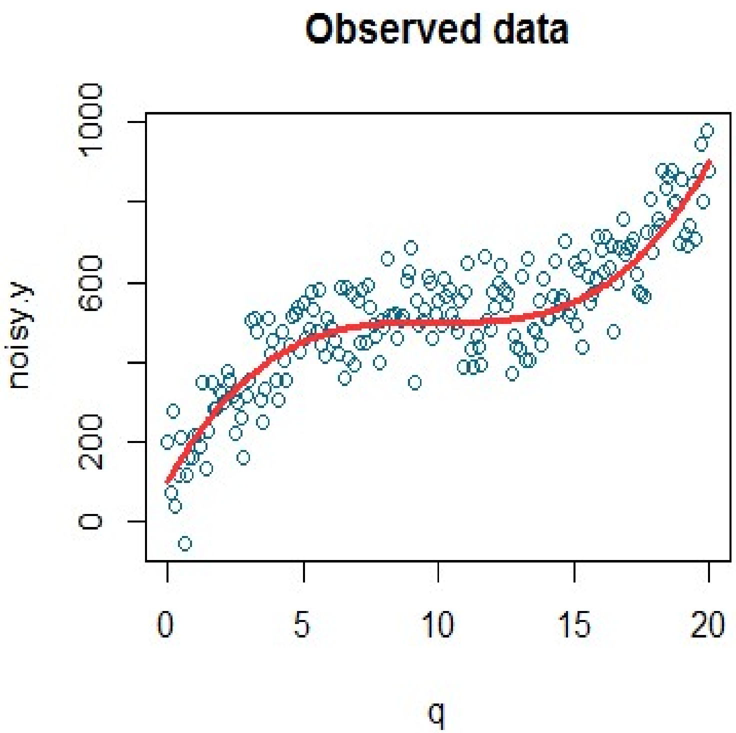
$$R[f] = \int l(f(\mathbf{x}), y) dP(\mathbf{x}, y)$$

- **The expected risk is approximated by the empirical risk:**

$$R_{emp}[f] = \frac{1}{n} \sum_{i=1}^{n} l(f(\mathbf{x}_i), y_i)$$

Loss functions

- The loss function (cost function, objective function) is a measure of how well the predicting model is doing the associated task. Loss functions are minimized in the training set to estimate the parameters of the model.



- In Regression Problems the most well known loss function is the squared loss

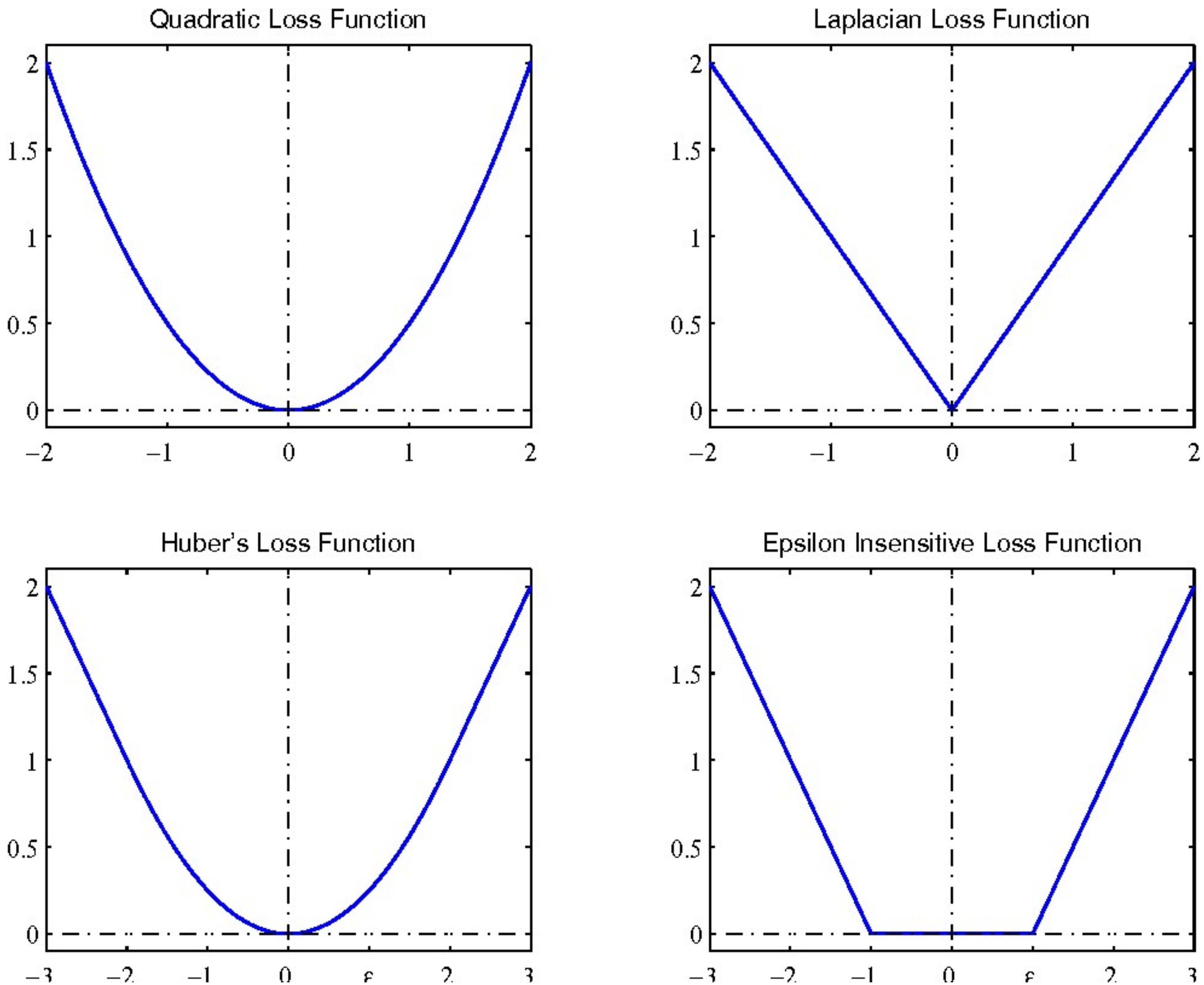
$$l(f(\mathbf{x}), y) = (f(\mathbf{x}) - y)^2$$

- Example: Fitting a third order polynomial to data by least squares

$$\mathbf{f}(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$$

Loss functions

- **Examples of loss functions in regression:**



Loss functions in Classification

- **Let us consider a binary classification problem:**

$$f : \Re^N \to \{\pm 1\} \qquad (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l) \in \Re^N \times \{\pm 1\}$$

Heaviside function

- **Indicator function** $l(f(\mathbf{x}), y) = \Theta(-yf(\mathbf{x}))$
- **Square loss** $l(f(x), y) = (1 - yf(x))^2$
- **Hinge loss** $l(f(x), y) = max(0, (1 - yf(x))$
- **Cross-Entropy** $l(f(x), t) = -t\ln(f(x)) - (1-t)\ln(1-f(x))$

Here f(x) maps to {0,1} $t = (1 + y)/2$

All loss functions give a value of zero when f(x)=y

Binary classifiers are Detectors: Signal Detection Theory

Statisticians: Hypothesis testing	Engineers: Detection theory
Test statistics (T(x) and v-threshold)	Detector
Null hypothesis	Noise hypothesis
Alternative hypothesis	Signal+noise hypothesis
Type I error (decide H_1 when H_0 true)	False Alarm
Type II error (decide H_0 when H_1 true)	False Negative (or Miss)
Level of Significance or Risk α	Probability of False Alarm
Probability of Type II error β	Probability of Miss
Power of test (1-β)	Probability of Detection

Decision	H_o true	H_o false
Accept H_o	1-α	β (unknown) (errot type II)
Reject H_o	α (error type I)	1-β

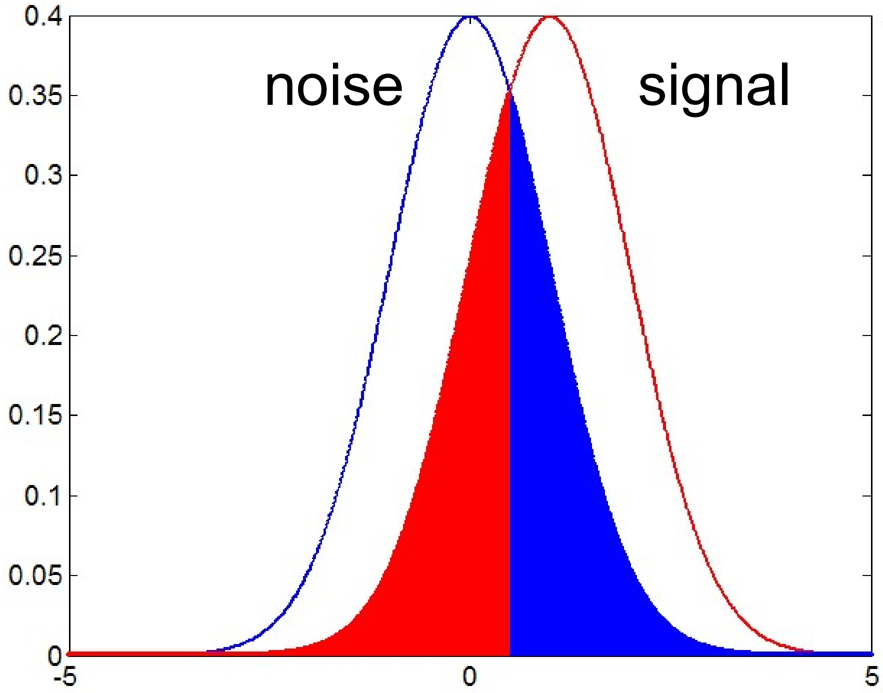
In some cases only the pdf of H_o is known

But….

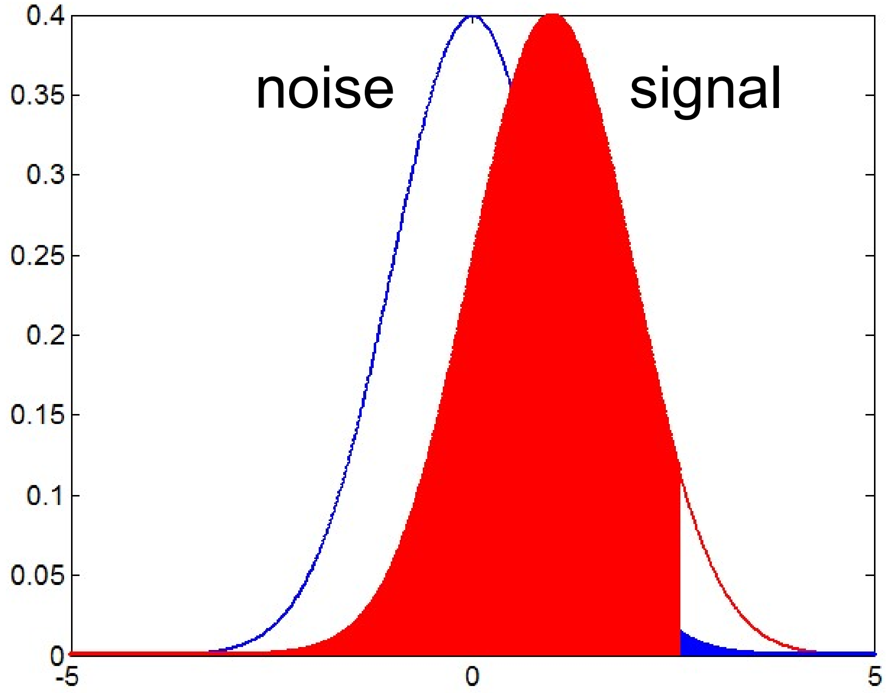
Then β is unknown.

Receiver Operating Characteristics

- **There is a trade-off between False Alarms and False Negatives**



blue: False Alarms
red: Misses



Threshold for a False Alarm rate of 0.01
Most signals are missed

Statistical Decision Theory: Terminology

Real

Decision

	Normal	Alarm
Normal	True-negatives (TN)	False negative (FN)
Alarm	False positive (FP)	True-positives (TP)

Accuracy (Classification Rate)= (TP+TN)/(TP+TN+FP+FN)

Sensitivity (Recall)=TP/(TP+FN) – Probability to correctly classify an Alarm

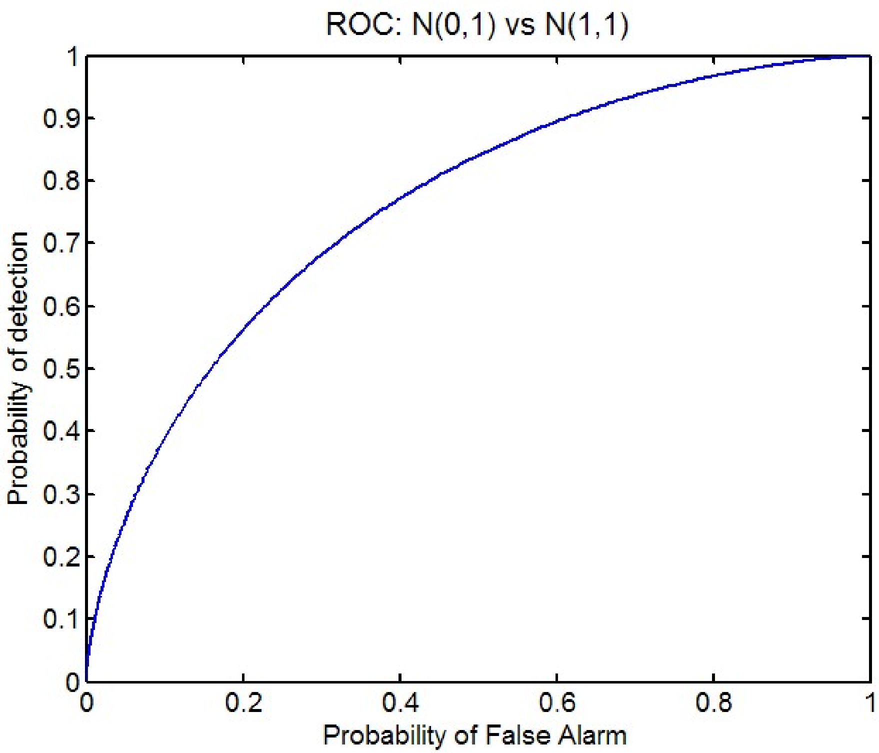
Specificity=TN/(TN+FP) – Probability to correctly classify a Normal state

Precision (Positive Predictive Power)=TP/(TP+FP) – Reliability of Alarm

Negative Predictive Power= TN/(TN+FN) – Reliability of no-alarm

Receiver Operating Characteristics

- **The optimal threshold depend on the relative costs of the false alarms or the false negatives, and on the prior probabilities of both events**
- **But...What the priors or the relative costs are not known?**

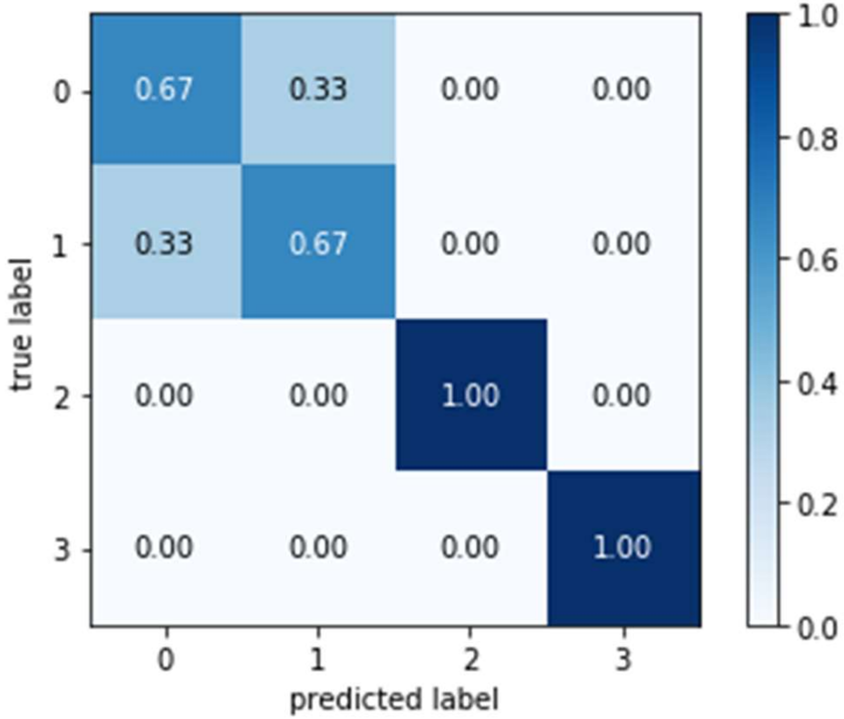


The ROC curve explores the trade-off for all possible values of the threshold

The area under the curve : AUC is a commonly used figure of merit to evaluate classifiers with an analog output.

Confusion Matrix

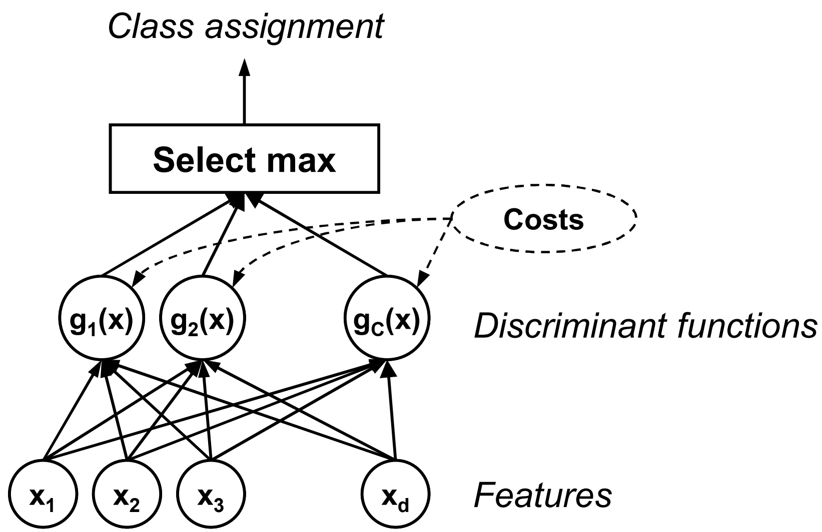
- **Evaluation of classifiers in multi-class problems is mostly based in the so-called confusion matrix.**



INTRODUCTION TO CLASSIFIERS

Discriminant functions

- **A convenient way to represent a pattern classifier is in terms of a family of discriminant functions $g_i(x)$ with a simple MAX gate as the classification rule**



$$\text{Assign } x \text{ to class } \omega_i \text{ if } g_i(x) > g_j(x) \ \forall j \neq i$$

- **How do we choose the discriminant functions $g_i(x)$**
  - Depends on the objective function to minimize
    - Probability of error
    - Bayes Risk

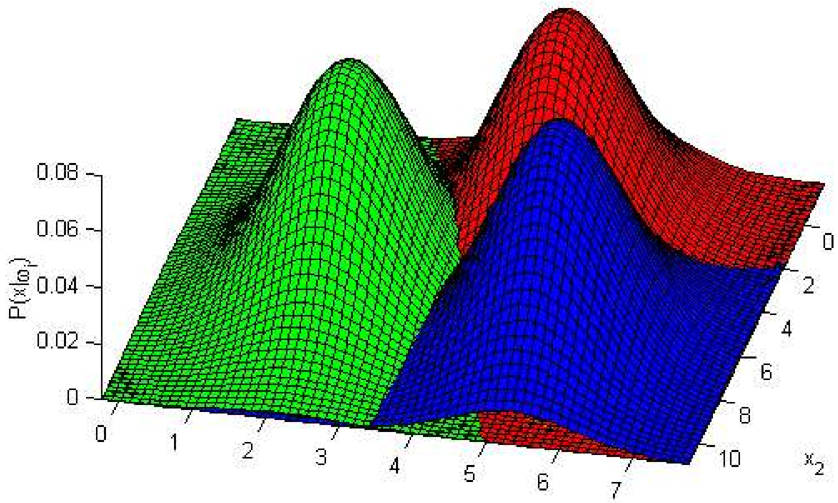
Materials in this lecture adapted from Dr. Ricardo Gutierrez, Texas A&M, USA

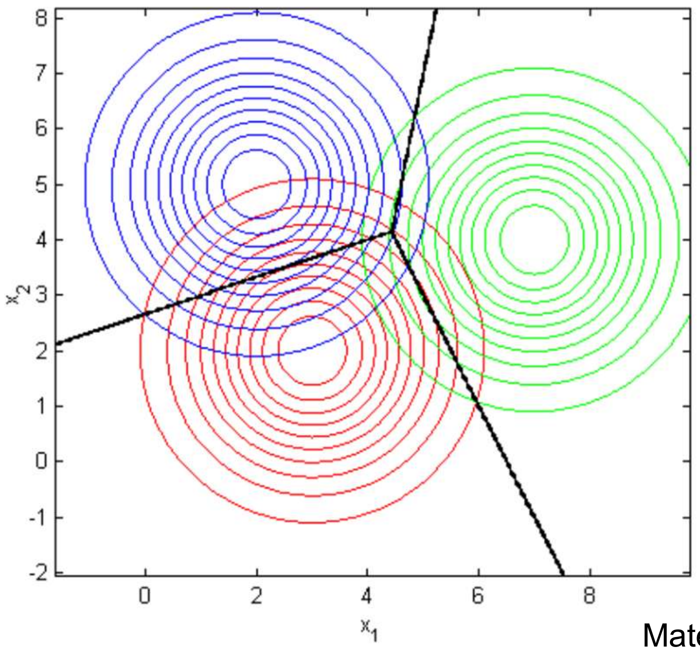
Simple Classifiers: Nearest Centroid Classifier

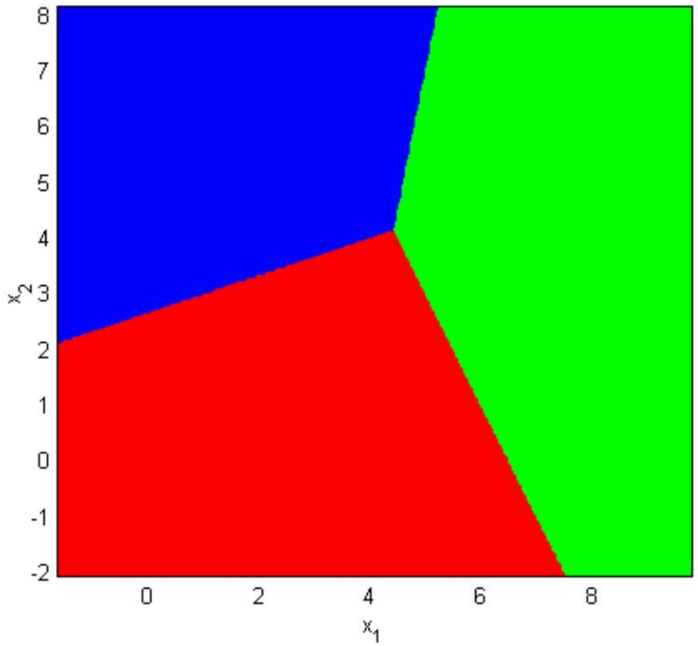
- **In this case, the discriminant becomes**

$$g_i(x) = -(x-\mu_i)^T(x-\mu_i)$$

  - This is known as a **NEAREST CENTROID CLASSIFIER**
  - Notice the linear decision boundaries



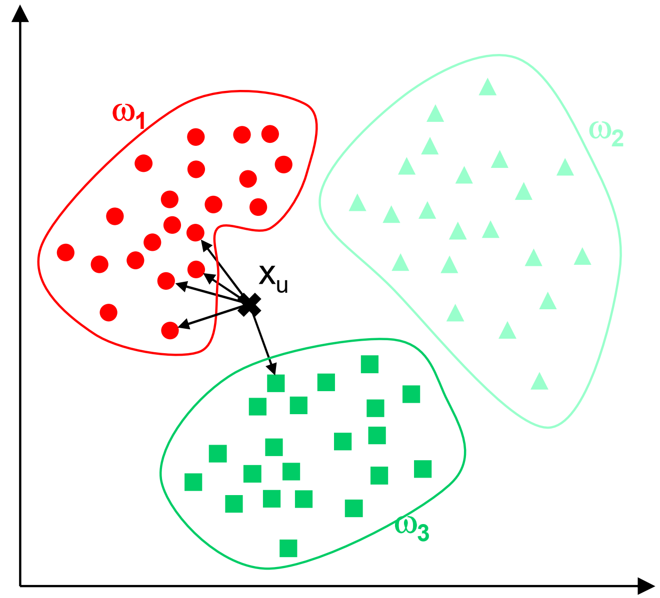




Materials in this lecture adapted from Dr. Ricardo Gutierrez, Texas A&M, USA

K Nearest Neighbor classifier

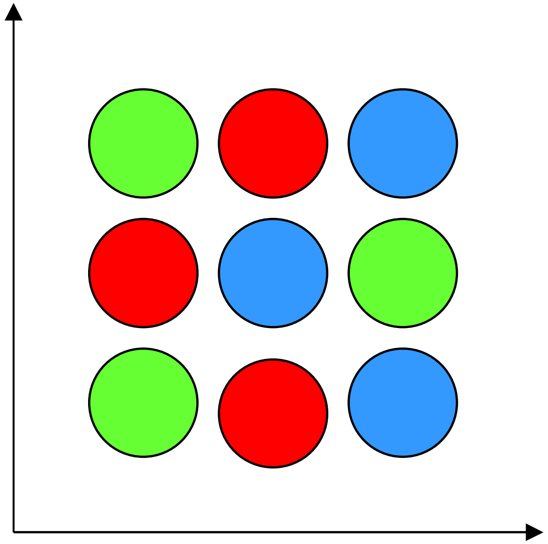
- **The kNN classifier is a very intuitive method**
  - Examples are classified based on their similarity with training data
    - For a given unlabeled example $x_u \in \Re^D$, find the k “closest” labeled examples in the training data set and assign x_u to the class that appears most frequently within the k-subset
- **The kNN only requires**
  - An integer k
  - A set of labeled examples
  - A metric to measure “closeness”

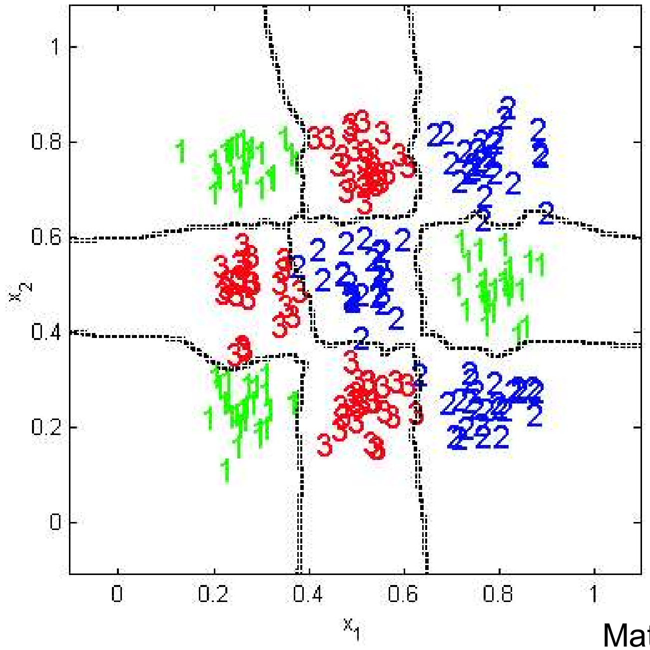


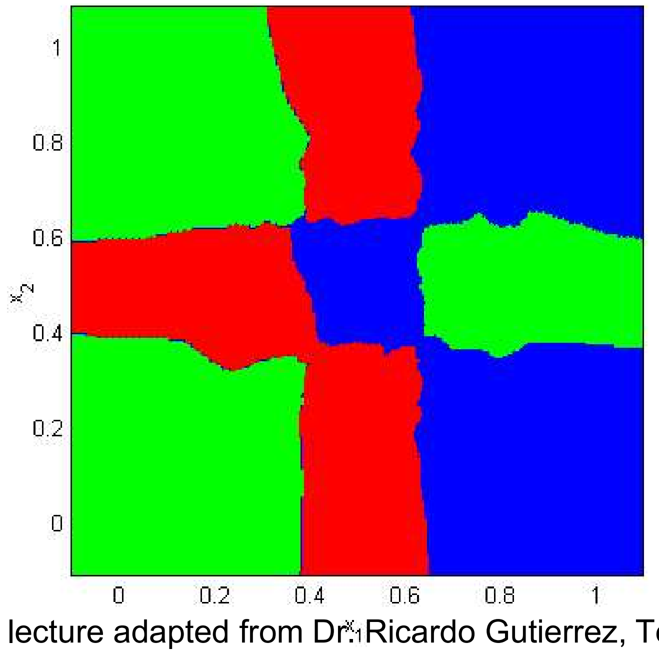
Materials in this lecture adapted from Dr. Ricardo Gutierrez, Texas A&M, USA

kNN in action: example 1

- **We generate data for a 2-dimensional 3-class problem, where the class-conditional densities are multi-modal, and non-linearly separable**
- **We used kNN with**
  - k = five
  - Metric = Euclidean distance





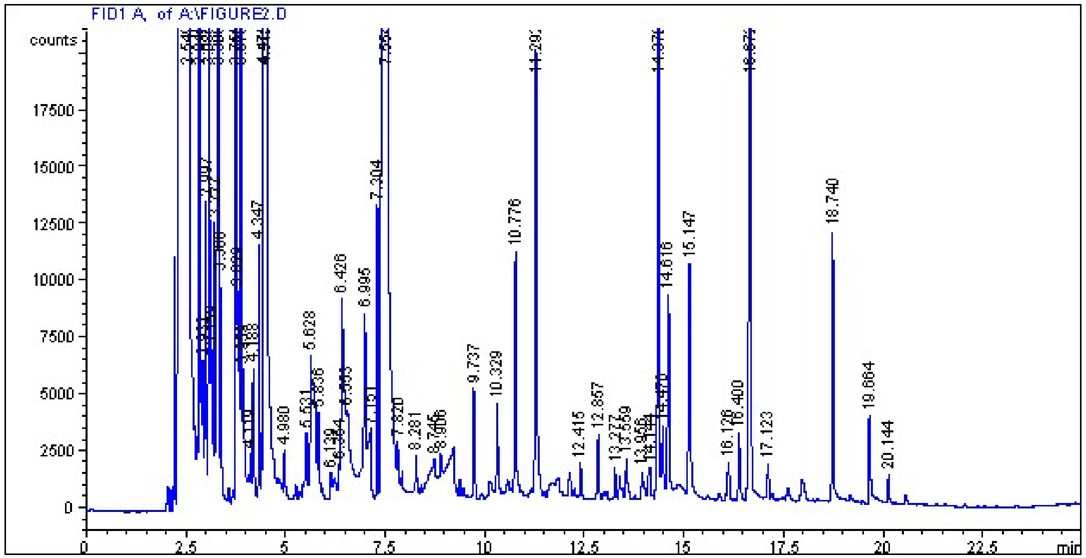


Materials in this lecture adapted from Dr. Ricardo Gutierrez, Texas A&M, USA

COMPLEXITY CONTROL

An example

- **Consider the following scenario:**
  - A GC-MS analysis of urine vapors has to decide if patients has prostatic cancer on the basis of a number of putative biomarkers.
  - The system consist of:
    - Urine collection (clinical setting)
    - Sample storage // inventory
    - Robot for automatic feeding the spectrometer
    - HS-GS-MS analysis (biochemical laboratory)
    - A computer that acquires the data
    - A machine learning suite to analyze the data and give a prediction (our part)

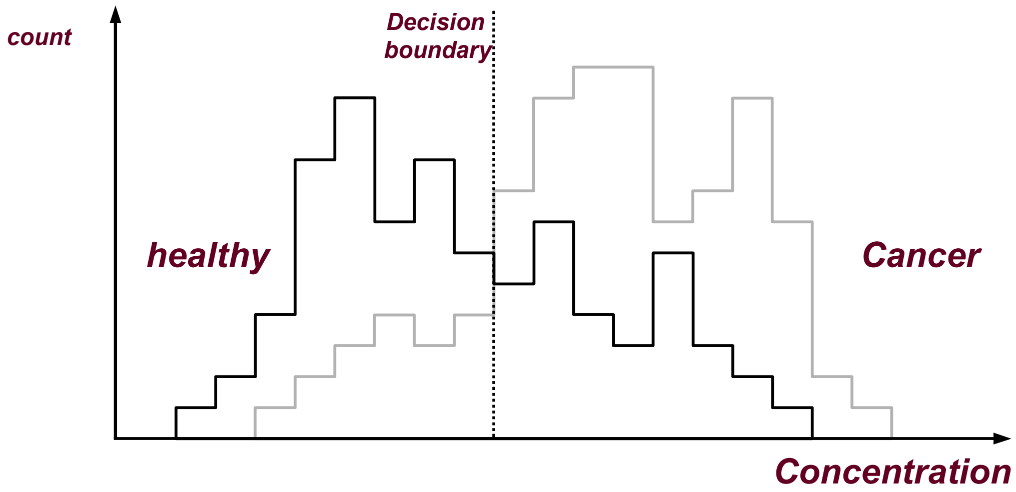


An example

- **Data source:**
  - The GC-MS instrument
- **Preprocessing:**
  - Noise reduction
  - Peak detection
  - Peak alignment & matching
- **Feature extraction**
  - Peak area integration

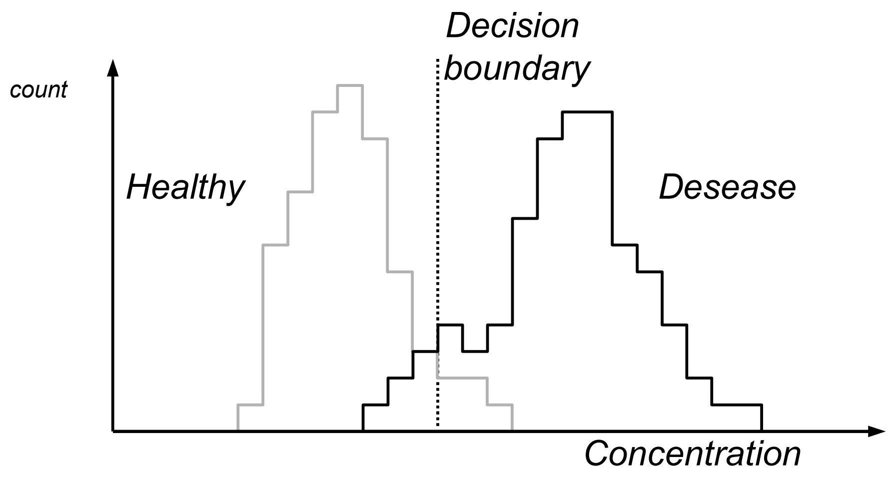
- **Supose literature sais compound A is more present in prostatic cancer urine than in healthy urine**

- **Classification**

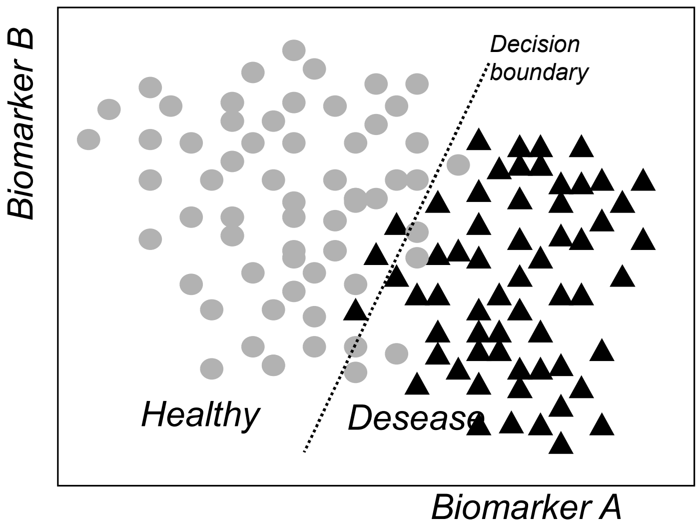


An example

- **Improving predictive ability going multivariate**
  - Committed to achieve a prediction error of 95% we search for other additional biomarkers.
  - We find a good second analyte.

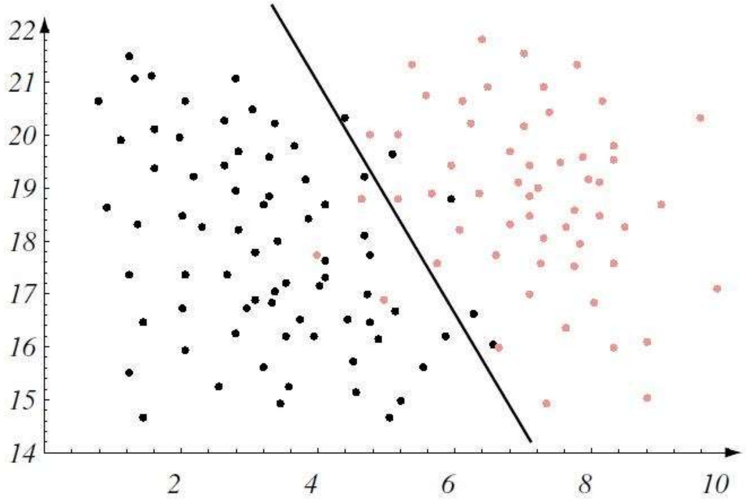


  - Both can be combined
  - And a separating hyperplane may be found
  - Recognition improves to 95.7%



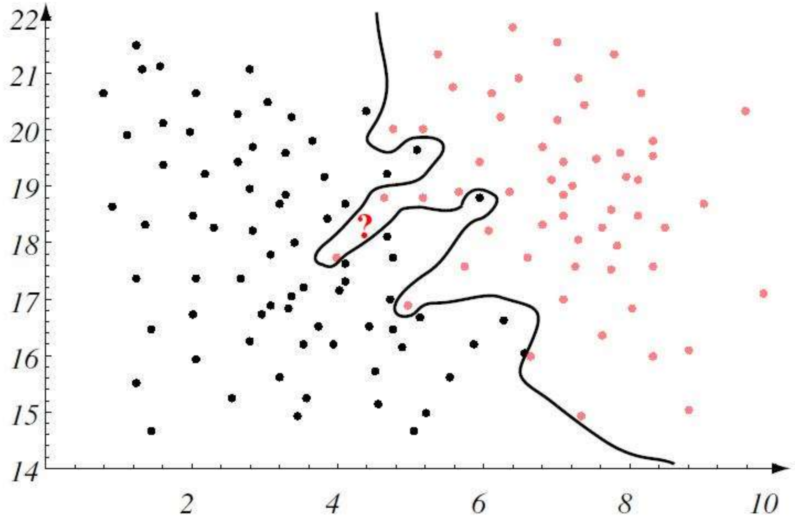
Complexity Control (Algorithmic selection)

- Too simple model:
- Low complexity



- Large training errors
- Large test errors

- Too complex model:
- High complexity

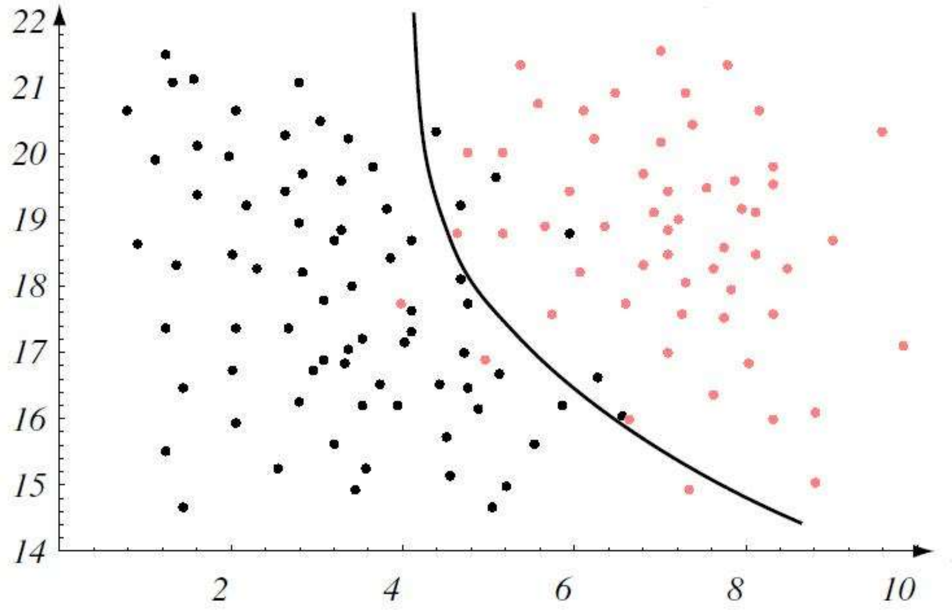


- Zero training errors
- Large test errors
- Poor generalization

Duda, Hart, Stork, 2001

Motivation: Complexity Control

- **Some optimal classifier exist**

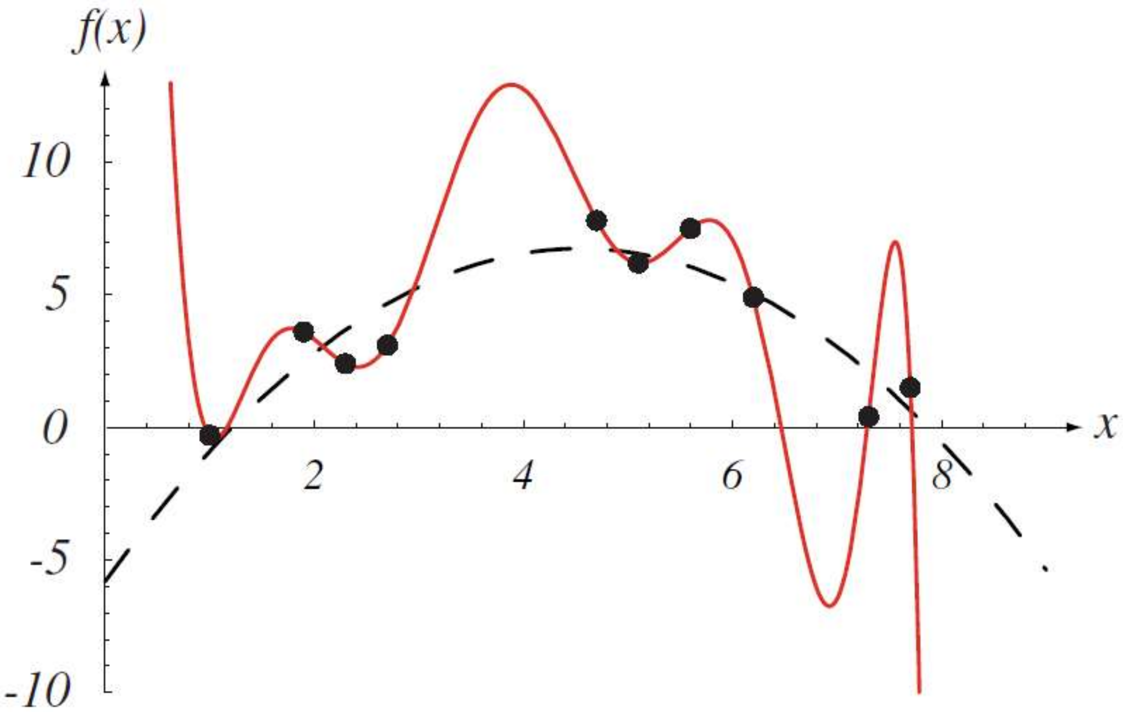


- **The complexity of the model has to be controlled for good performance**

Duda, Hart, Stork, 2001

Motivation: Complexity Control

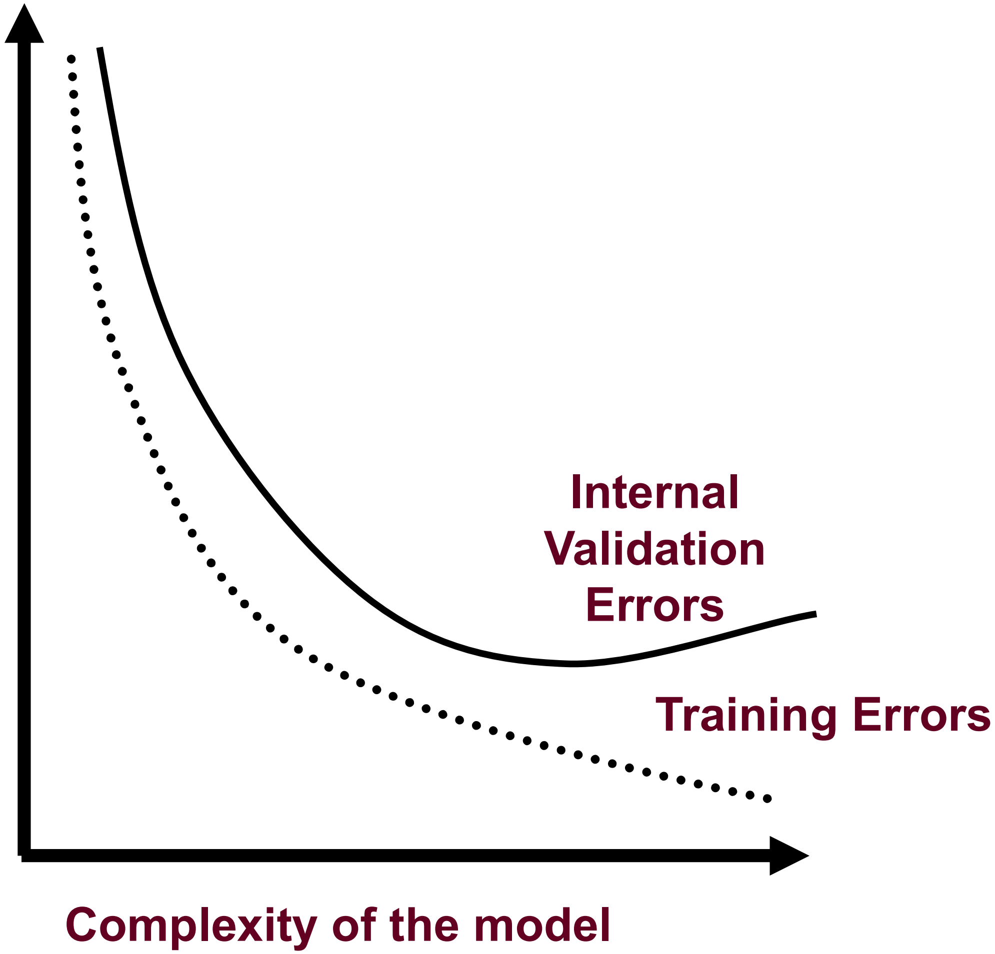
- **Regression example: polynomial fitting**
- **Question: What is the best polynomial order to fit the data?**
  - A 10th order polynomial predicts perfectly the training data but fits also the noise, producing large errors for the prediction of new samples.



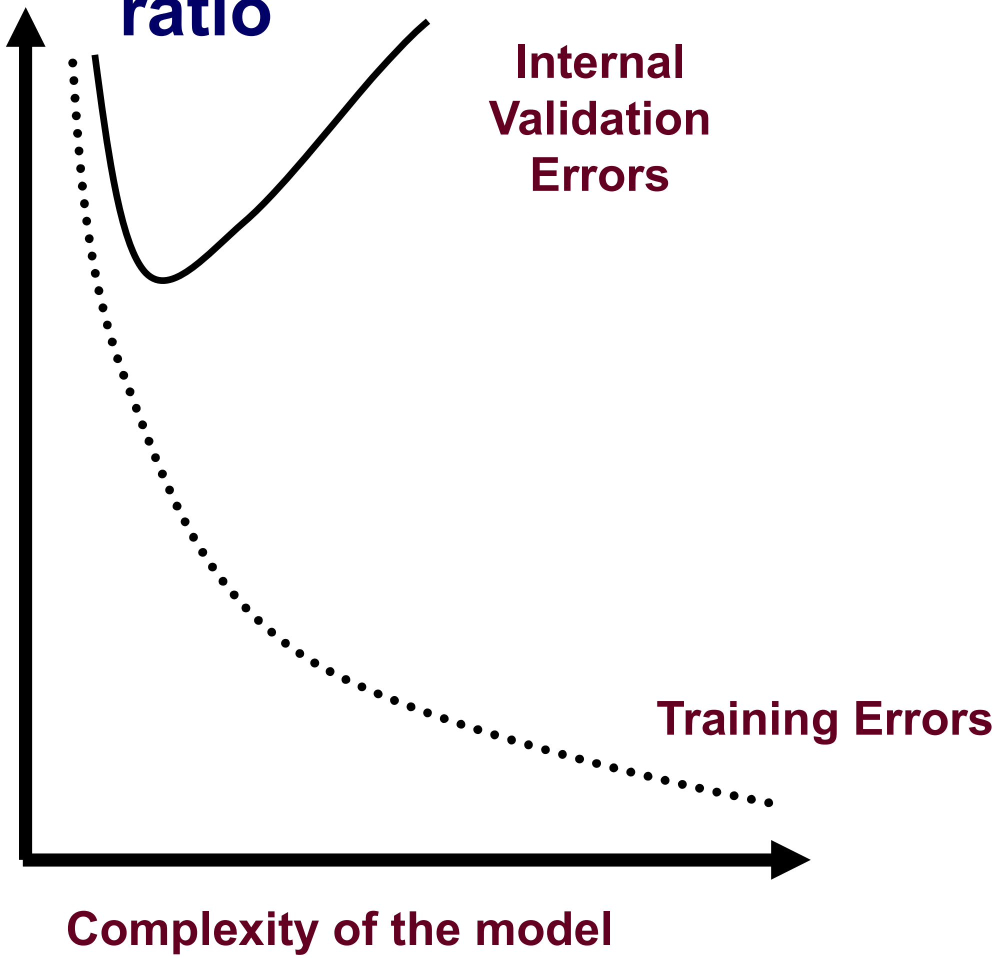
Duda, Hart, Stork, Pattern Classification, 2001

Motivation: Complexity Control

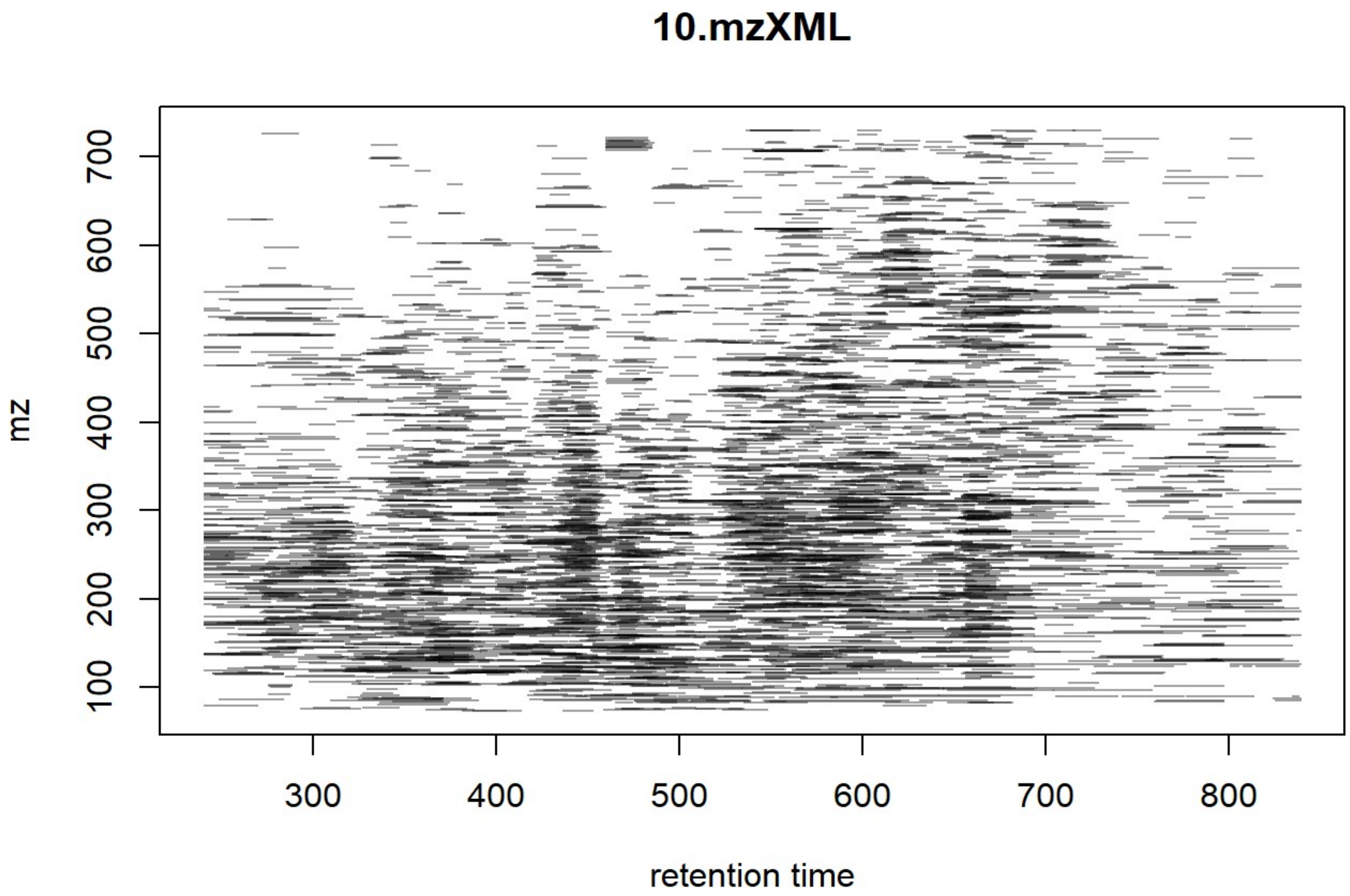
- Large Samples/Dim ratio



- Small Samples/Dim ratio



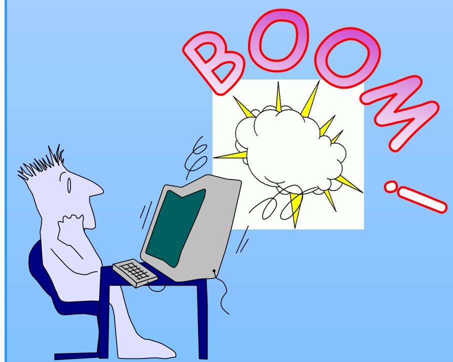
DIMENSIONALITY REDUCTION



The Curse of Dimensionality (Bellman, 1961)

Curse of Dimensionality:

The performance of learning algorithms is clearly sub-optimal when there is a small number of examples / dimensionality ratio



Example of LC-MS data

Fragments/peaks may be found using XCMS, MzMine, Py-MS and others

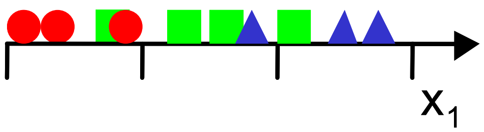
Table 1. Summary of working examples obtained from LC-MS untargeted metab experiments. Further experimental details and methods can be obtained from refe (KO=Knock-Out; WT=Wild-Type).

	Biofluid/Tissue	Sample groups	# samples /group	# XCMS variables	System
Example #1	Retina	KO	11	4581	LC/ESI-QTOF
		WT	11		
Example #2	Retina	Hypoxia	12	8146	LC/ESI-QTOF
		Normoxia	13		
Example #3	Serum	Untreated	12	9877	LC/ESI-TOF
		Treated	12		
Example #4	Neuronal cell cultures	KO	15	8221	LC/ESI-QTOF
		WT	11		

M. Vinaixa,Metabolites, 2012

Dimensionality reduction

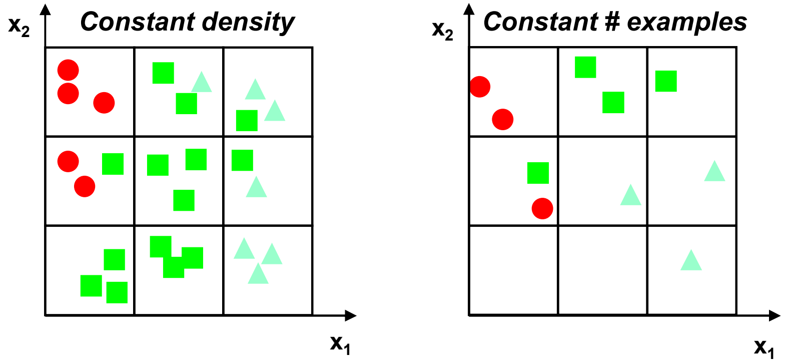
- **The “curse of dimensionality” [Bellman, 1961]**
  - Refers to the problems associated with multivariate data analysis as the dimensionality increases
- **Consider a 3-class pattern recognition problem**
  - A simple (Maximum Likelihood) procedure would be to
    - Divide the feature space into uniform bins
    - Compute the ratio of examples for each class at each bin and,
    - For a new example, find its bin and choose the predominant class in that bin
  - We decide to start with one feature and divide the real line into 3 bins



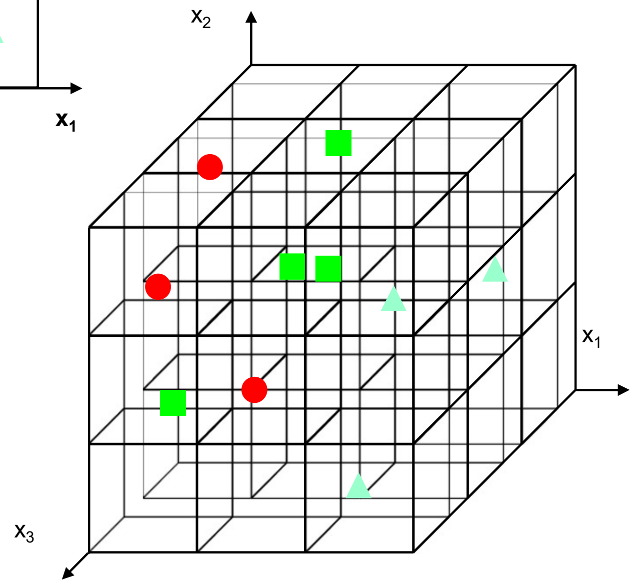
    - Notice that there exists a lot of overlap between classes ⇒ to improve discrimination, we decide to incorporate a second feature

Dimensionality reduction

- **Moving to two dimensions increases the number of bins from 3 to 3^2=9**
  - QUESTION: Which should we maintain constant?
    - The density of examples per bin? This increases the number of examples from 9 to 27
    - The total number of examples? This results in a 2D scatter plot that is very sparse

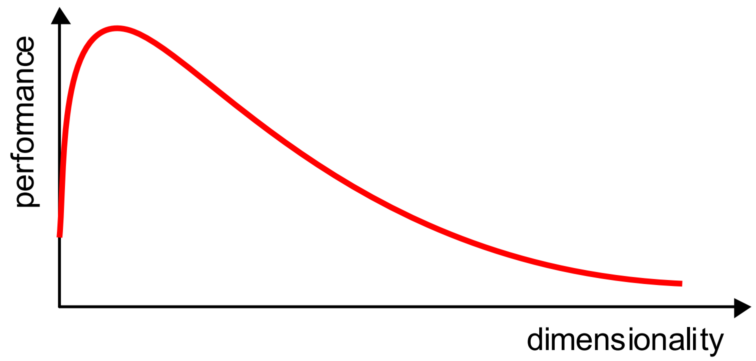


- **Moving to three features …**
  - The number of bins grows to 3^3=27
  - To maintain the initial density of examples, the number of required examples grows to 81
  - For the same number of examples, the 3D scatter plot is almost empty



Dimensionality reduction

- **Of course, our approach to divide the sample space into equally spaced bins was quite inefficient**
  - There are other approaches that are much less susceptible to the curse of dimensionality, **but the problem still exists**
- **How do we beat the curse of dimensionality?**
  - By reducing the dimensionality
  - By using regularized classifiers
- **In practice, the curse of dimensionality means that**
  - For a given sample size, there is a maximum number of features above which the performance of our classifier will degrade rather than improve
    - In most cases, the information that was lost by discarding some features is compensated by a more accurate mapping in lower-dimensional space



Dimensionality reduction

- **Two approaches to perform dim. reduction $\Re^N \rightarrow \Re^M$ (M<N)**
  - **Feature selection**: choosing a subset of all the features

$$[x_1\ x_2 \ldots x_N] \xrightarrow{\text{feature selection}} [x_{i_1}\ x_{i_2} \ldots x_{i_M}]$$

  - **Feature extraction**: creating new features by combining existing ones

$$[x_1\ x_2 \ldots x_N] \xrightarrow{\text{feature extraction}} [y_1\ y_2 \ldots y_M] = f\left([x_{i_1}\ x_{i_2} \ldots x_{i_M}]\right)$$

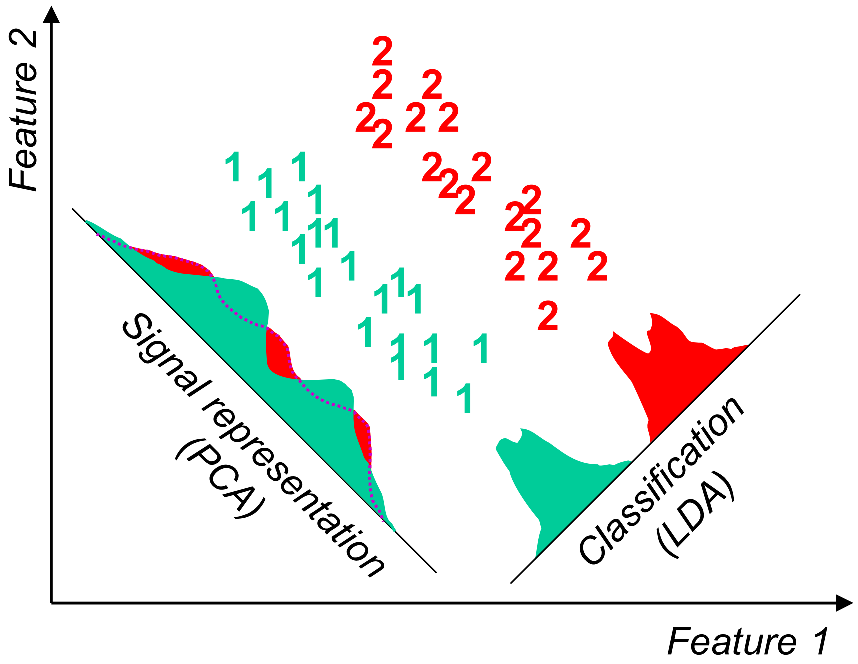
    - In either case, the goal is to find a low-dimensional representation of the data that preserves (most of) the information or structure in the data

- **Linear feature extraction (feature projection)**
  - The "optimal" mapping y=f(x) is, in general, a non-linear function whose form is problem-dependent
    - Hence, feature extraction is commonly limited to linear projections **y=Wx**

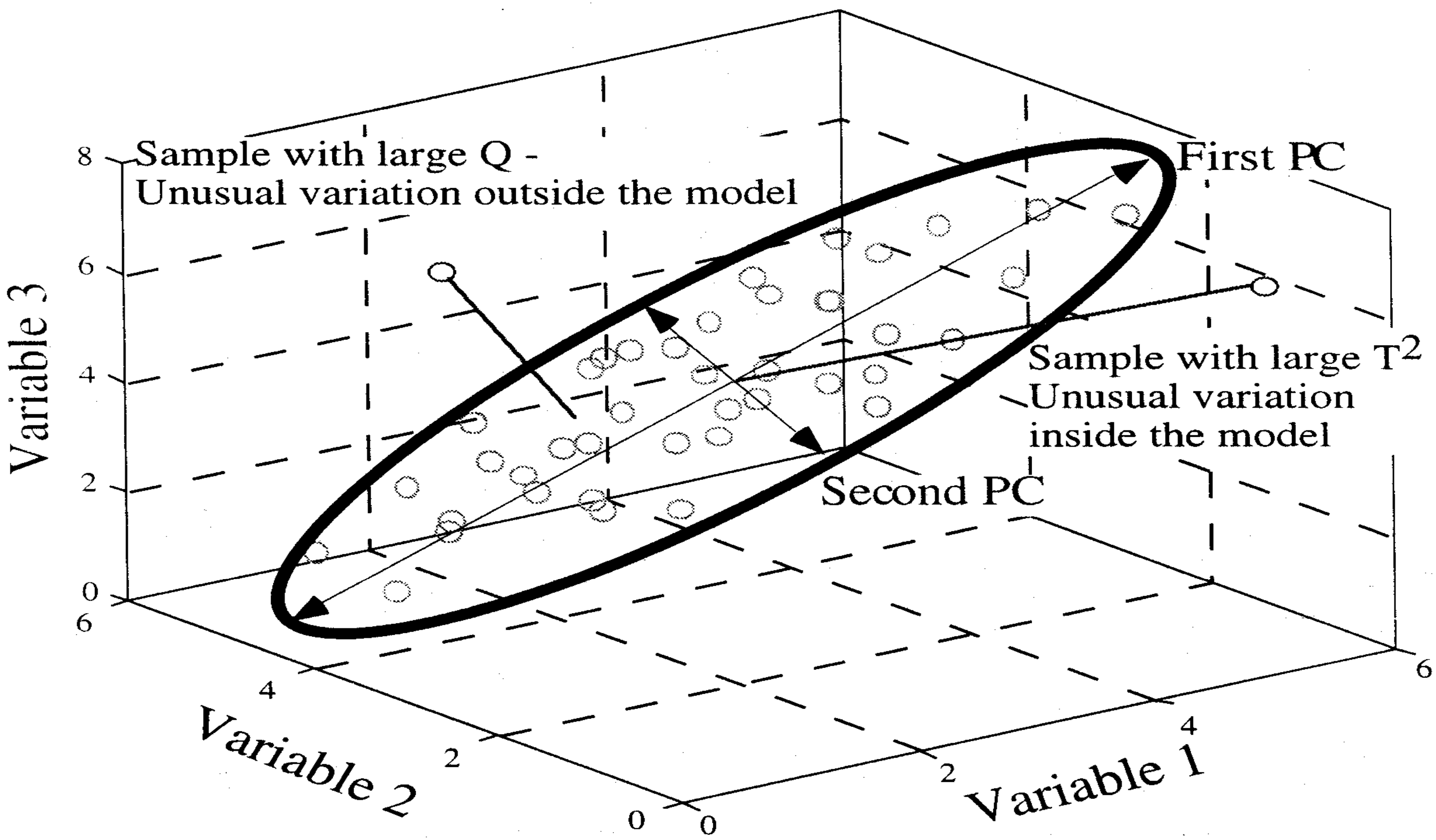
$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \\ x_N \end{bmatrix} \xrightarrow{\text{linear feature extraction}} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1N} \\ w_{21} & w_{22} & \cdots & w_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ w_{M1} & w_{M2} & & w_{MN} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \\ x_N \end{bmatrix}$$

Signal representation versus classification

- **Two criteria can be used to find the "optimal" feature extraction mapping y=f(x)**
  - **Signal representation**: The goal of feature extraction is to represent the samples accurately in a lower-dimensional space
  - **Classification**: The goal of feature extraction is to enhance the class-discriminatory information in the lower-dimensional space
- **Within the realm of linear feature extraction, two techniques are commonly used**
  - Principal Components (PCA)
    - **Unsupervised**
  - Fisher's Linear Discriminant (LDA)
    - **Supervised**



PCA Geometry



INTRO TO REGULARIZATION

Introduction to regularization

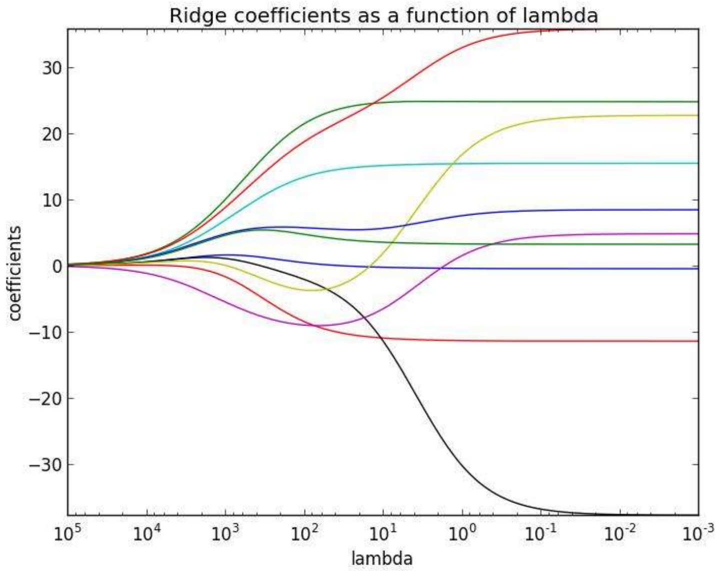
- **Some loss functions lead to less complex models. Usually, a penalty term increases the loss function for complex models.**
- **Examples in linear regression:**

$$y_k = f(X_k) + \varepsilon_k = \beta_0 + \sum_{j=1}^{p} x_{k,j}\beta_j + \varepsilon_k$$

- Ridge Regression compared to Ordinary Least Squares

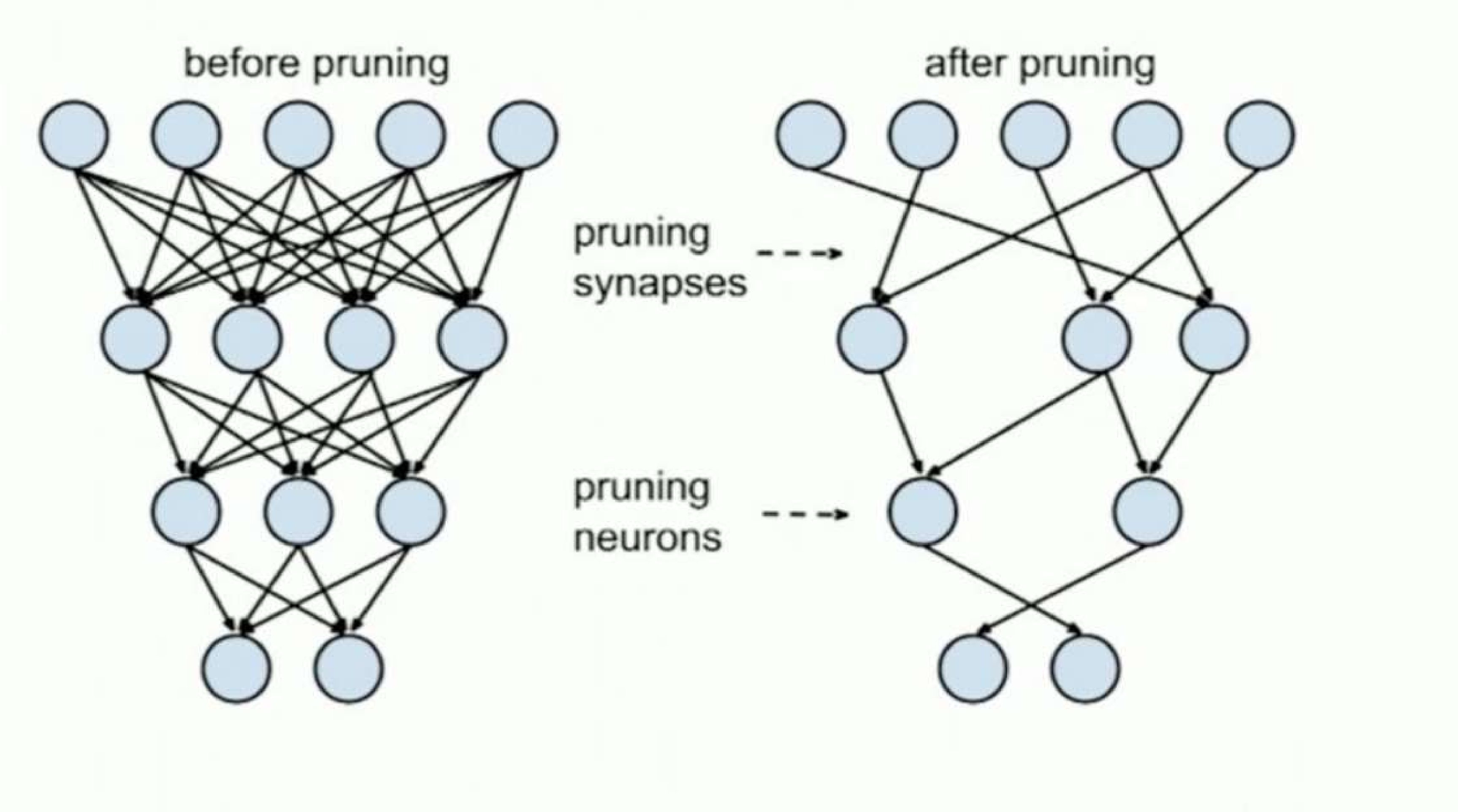
$$\hat{\boldsymbol{\beta}} = \boldsymbol{arg}\min_{\boldsymbol{\beta}} \left\{ \sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} \left(\boldsymbol{y_i} - \boldsymbol{\beta_0} - \sum_{\boldsymbol{j=1}}^{\boldsymbol{p}} \boldsymbol{x_{ij}\beta_j} \right) \right\}$$

$$\hat{\beta}^{ridge} = \arg\min_{\beta} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}$$



Introduction to regularization

- In Neural Neworks, models can be made simpler by pruning the network.
- Several methodologies have been proposed to remove neurons or connections.



Summary

- **The advances in sensing, instrumentation, imaging, wearables is producing a DATA AVALANCHE.**
- **In many settings, data interpretation becomes a bottleneck.**
- **The presence of Machine Learning in Health is incresing extremely fast.**
- **Caution words:**
  - In Machine Learning methodological errors leading to overoptimistic results are common.
  - Only precise methodology development, avoiding the use of algorithms as black-boxes and rigorous validation can prevent those errors.