



# Machine Learning for Bioinformatics & Systems Biology

## 1. Introduction & density estimation

Marcel Reinders

*Delft University of Technology*

Perry Moerland

*Amsterdam UMC, University of Amsterdam*

Lodewyk Wessels

*Netherlands Cancer Institute*

*Some material courtesy of Robert Duin, David Tax & Dick de Ridder*

# Programme

Day	Lecturer	Subjects
Monday 11/10	Marcel Reinders	Introduction to machine learning Bayesian framework Density estimation Bayesian classification
Tuesday 12/10	Perry Moerland	Parametric classifiers Nonparametric classifiers Discriminant analysis Decision trees & random forests
Wednesday 13/10	Lodewyk Wessels	Feature selection Sparse classifiers Feature extraction Embeddings
Thursday 14/10	Perry Moerland	Hierarchical clustering Agglomerative clustering Model-based clustering Hidden Markov models
Friday 15/10	Marcel Reinders	Artificial neural networks Support vector machines Classifier ensembles Complexity

# Certificates and examination

- To obtain a certificate of successful completion:
  - Analyse a biological dataset (preferably one from your own practice) using the tools provided in the course
  - Write a short report (5-10 pages) on the results
  - Hand this in no later than **November 5, 2021 (3 weeks after end of course)**
- If you have no dataset available, one will be provided
- Grade will be “pass” or “fail”, with at most one resubmission
- If no report or “fail”: certificate of attendance

# BioSB: The Netherlands Bioinformatics and Systems Biology research school

- Yearly conference: 15-16 February 2022, Lunteren
- Courses:

## Upcoming courses

Date	Level	Course	Duration
11-15 Oct 2021	fund.	<a href="#">Machine Learning</a>	5 days
18-22 Oct 2021	other	<a href="#">Single Cell Analysis</a>	5 days
13-17 Dec 2021	fund.	<a href="#">Integrated modeling and optimization</a>	5 days

- YoungCB: Regional Student Group (RSG) Netherlands of the International Society of Computational Biology



# Course

# Modelling .... Learning from examples



# Machine learning

- Wikipedia:
  - "the scientific study of **algorithms** and **statistical models** that computer systems use to perform a specific task without using explicit instructions, relying on **patterns** and **inference** instead ... Machine learning algorithms build a **mathematical model** based on **sample data**, known as "**training data**", in order to make **predictions** or **decisions** without being explicitly programmed to perform the task."
- Christopher M. Bishop:
  - "**Pattern recognition** has its origins in **engineering**, whereas **machine learning** grew out of **computer science**. However, these ... can be viewed as two facets of the **same field**"

# Machine learning (2)

- The construction of **approximate, generalizing (predictive) models** by **learning from examples**, for problems for which *no full physical model is known* (yet)
- Focus in this course will be on **classification** and **statistical machine learning**, not (so much) on *regression, structural/syntactic pattern recognition and reinforcement learning.*

- Related areas
  - Applied statistics
  - Pattern recognition
  - Artificial intelligence
  - Computer vision
  - Data mining

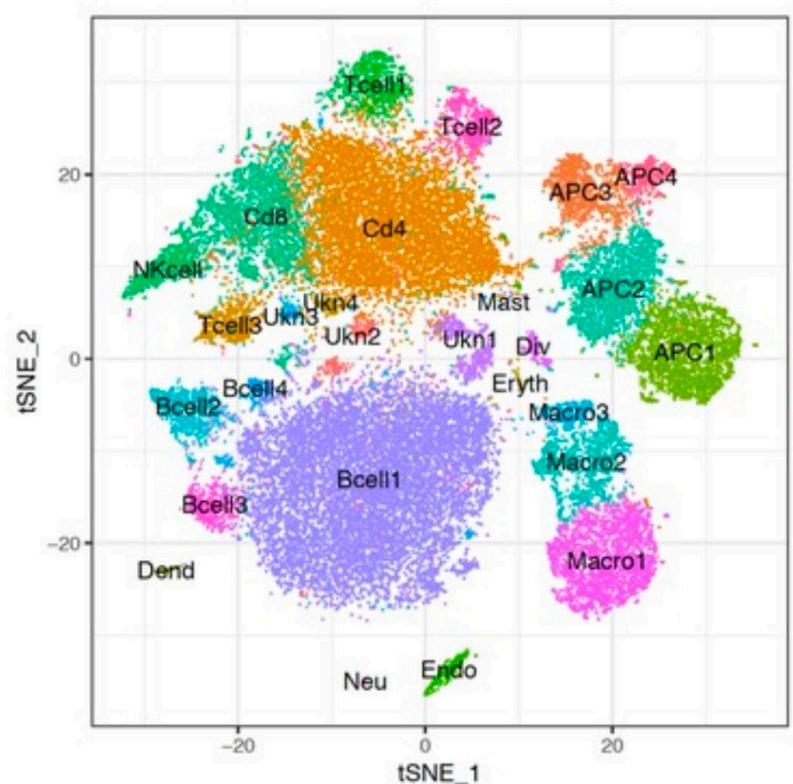


# Machine learning (3)

- Examples:
  - Computer vision: license plate reading, people counting, face detection, smart cameras, ...
  - Signal processing: thermostat, speech/speaker recognition, ...
  - Information retrieval: Google, Amazon, automated translation, ...
  - Biometrics: fingerprint recognition, iris scan, signature verification...
  - Defensive: friend-or-foe recognition, target tracking, ...
  - Medicine: interpreting scans, diagnostic systems, ...

# Machine learning (4)

- Bioinformatics:
  - Gene (function) prediction, SNP prioritization, ...
  - Diagnosis/prognosis, biomarker discovery, ...
  - Network inference: PPI, metabolic networks, ...
  - Cell-type identification, ...
  - Etc.



# Goal

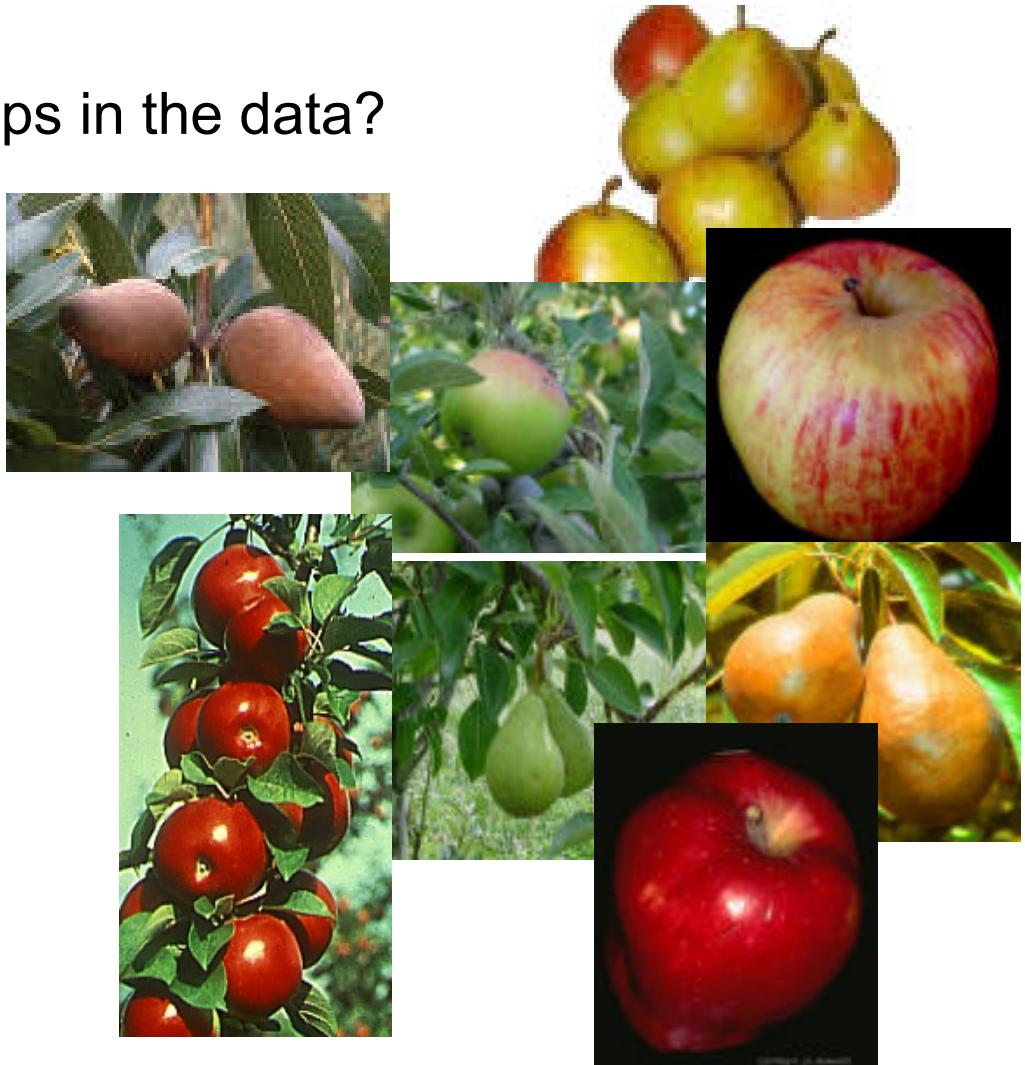
- After having followed this course, the student has a good understanding of **a wide range of machine learning techniques** and is able to **recognize what method is most applicable** to data analysis problems (s)he encounters in bioinformatics and systems biology applications.
- Many problems are in fact machine learning problems!

# Machine learning (5)

- Finding structure in data
  - Outlier/anomaly detection
  - Clustering
  - Dimensionality reduction,  
selecting useful (combinations of) features
  - Regression
  - Classification
  - ...
- All aimed at *generalisation*:  
**making a prediction for data you have not yet seen**

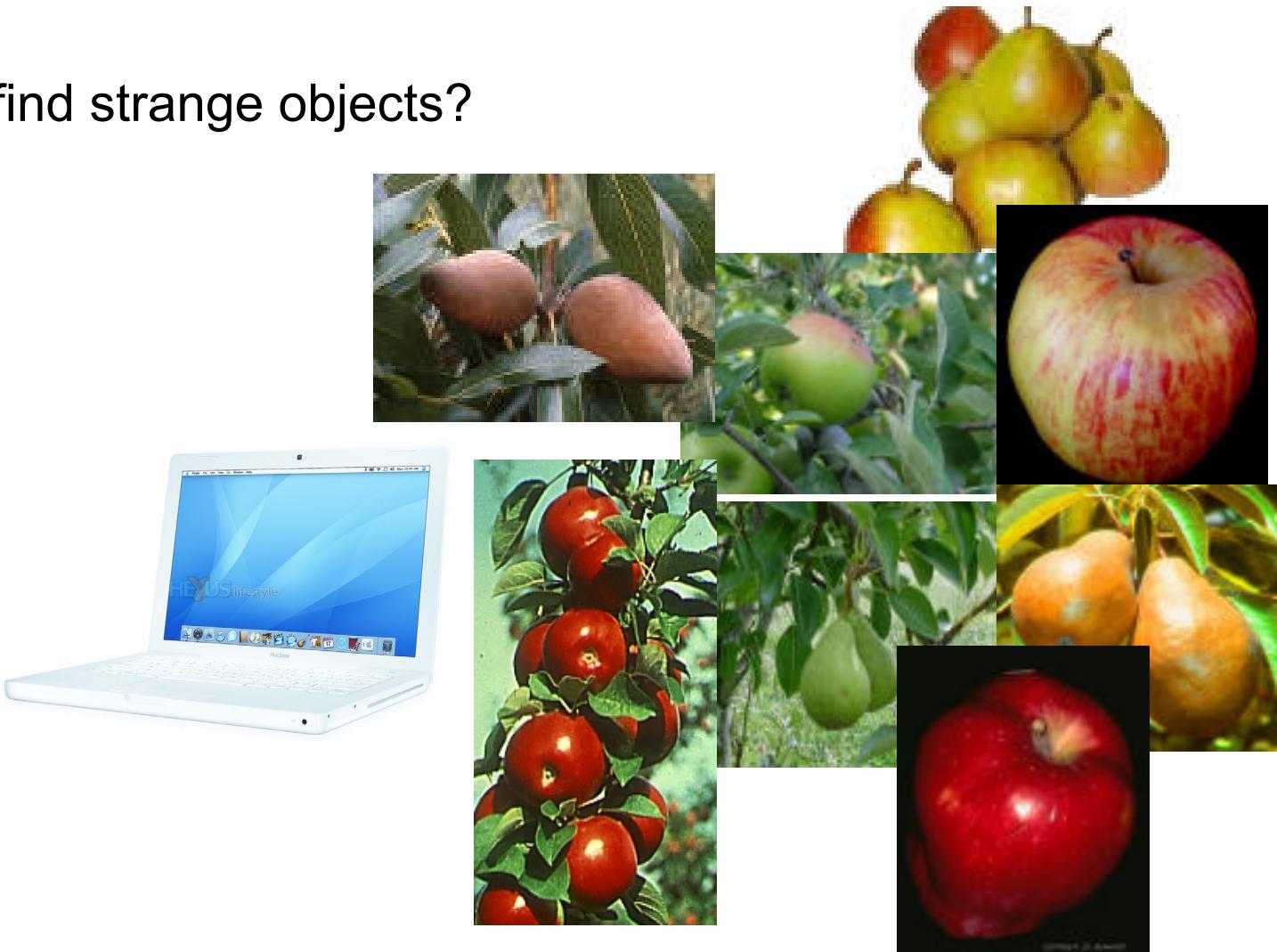
# Clustering

- Can we find natural groups in the data?
- E.g. red vs green fruit



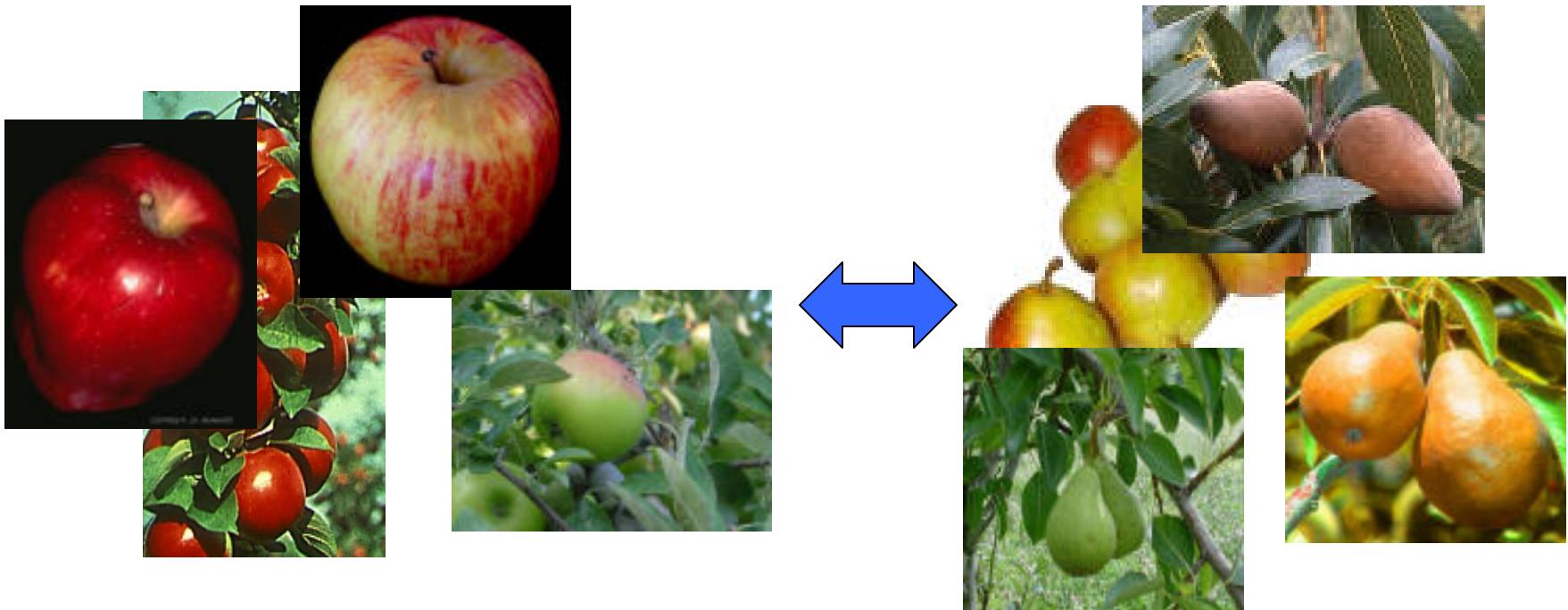
# Outlier detection

- Can we find strange objects?



# Dimensionality reduction

- Can we find predictive measurements?



Red

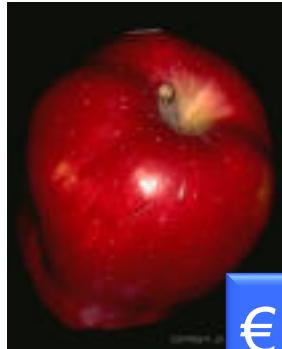
Yellow

# Regression

- Can we predict real-valued outputs?



€ 0.25



€ 0.30



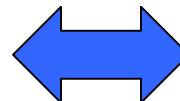
€ 0.40



€ 0.45

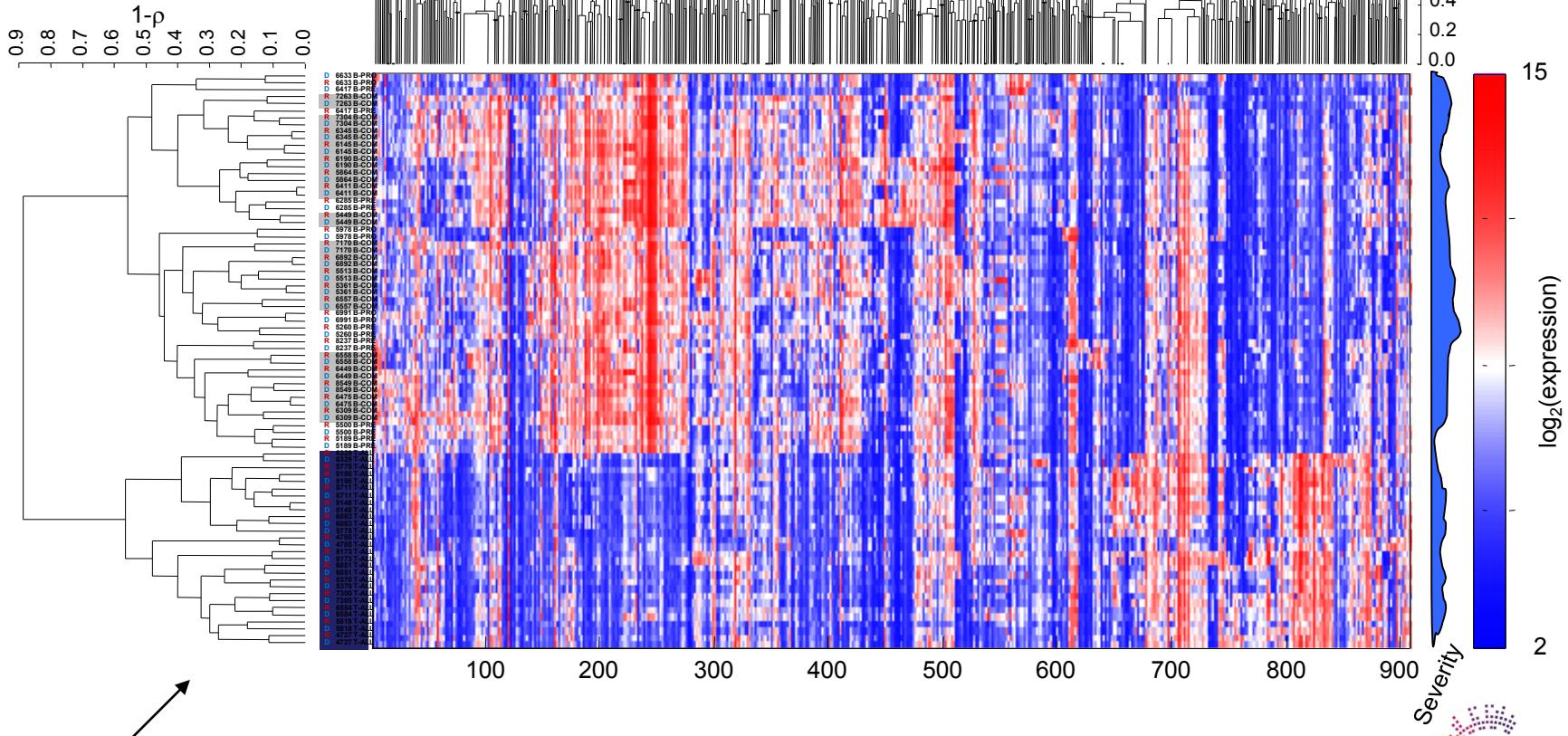
# Classification

- Can we distinguish apples from pears?



# Machine learning in bioinformatics

- Example:  
gene expression  
diagnostics

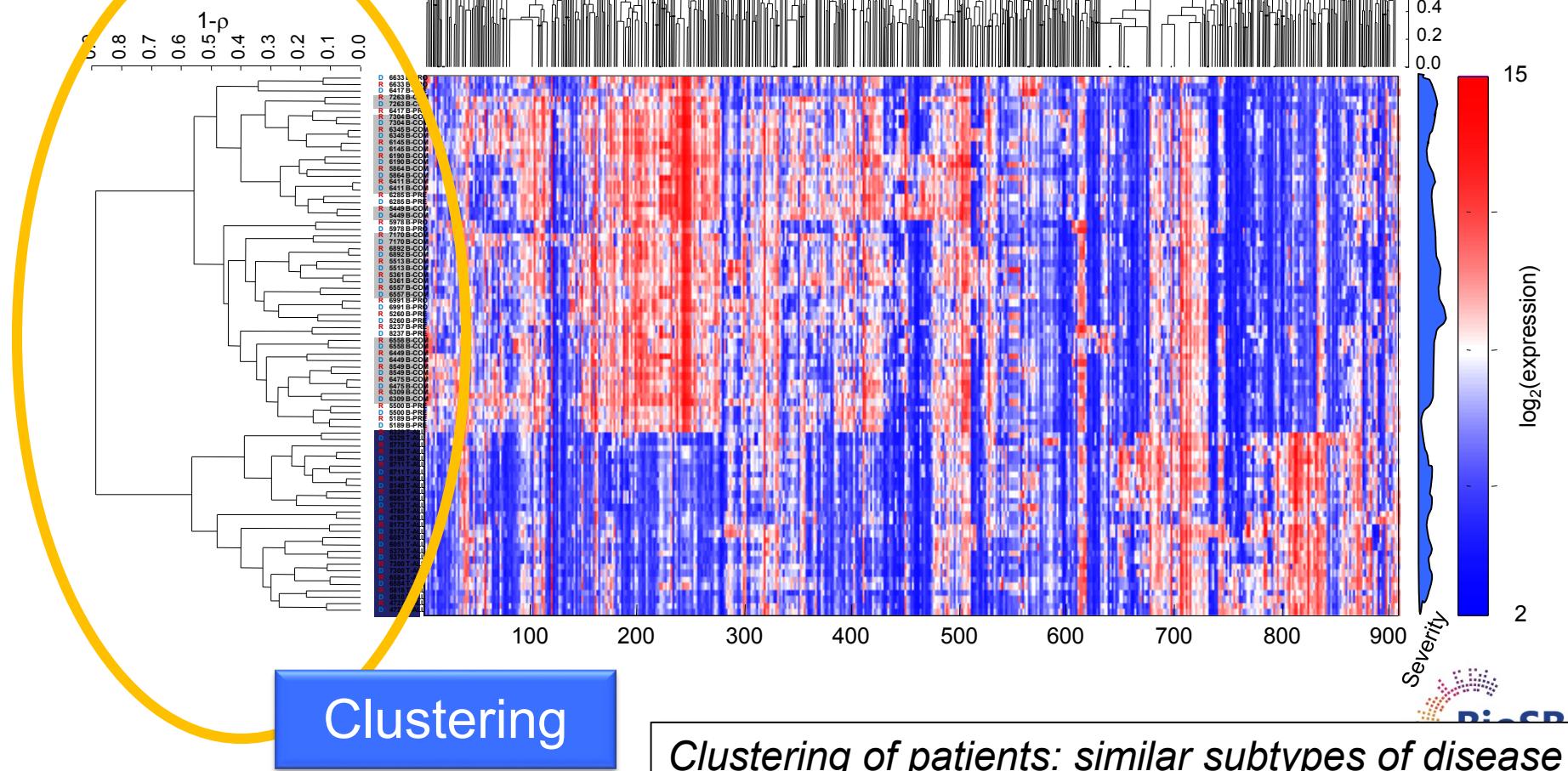


samples

diagnosis/relapse in childhood leukemia

# Machine learning in bioinformatics

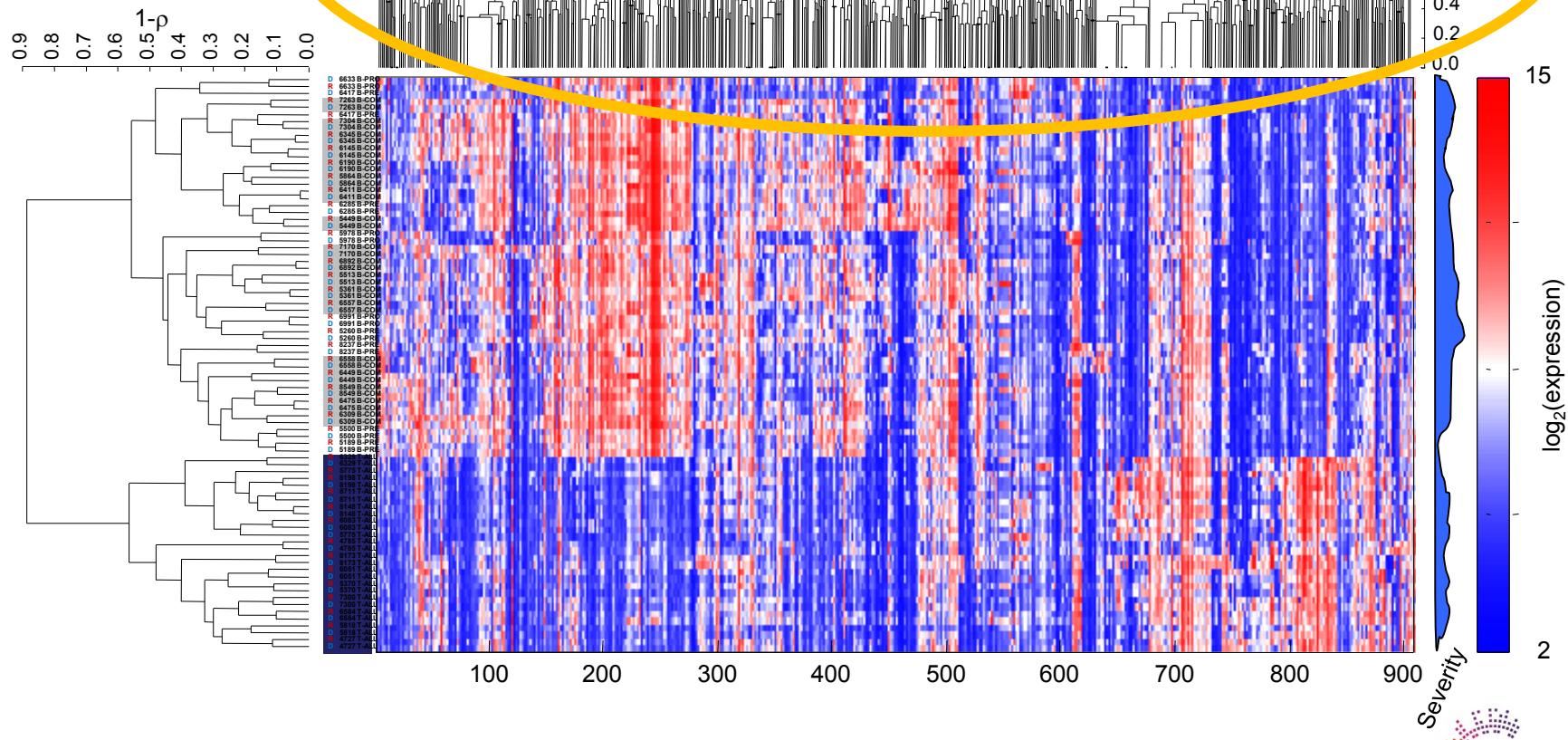
- Example:  
gene expression  
diagnostics



# Machine learning in bioinformatics

## Clustering

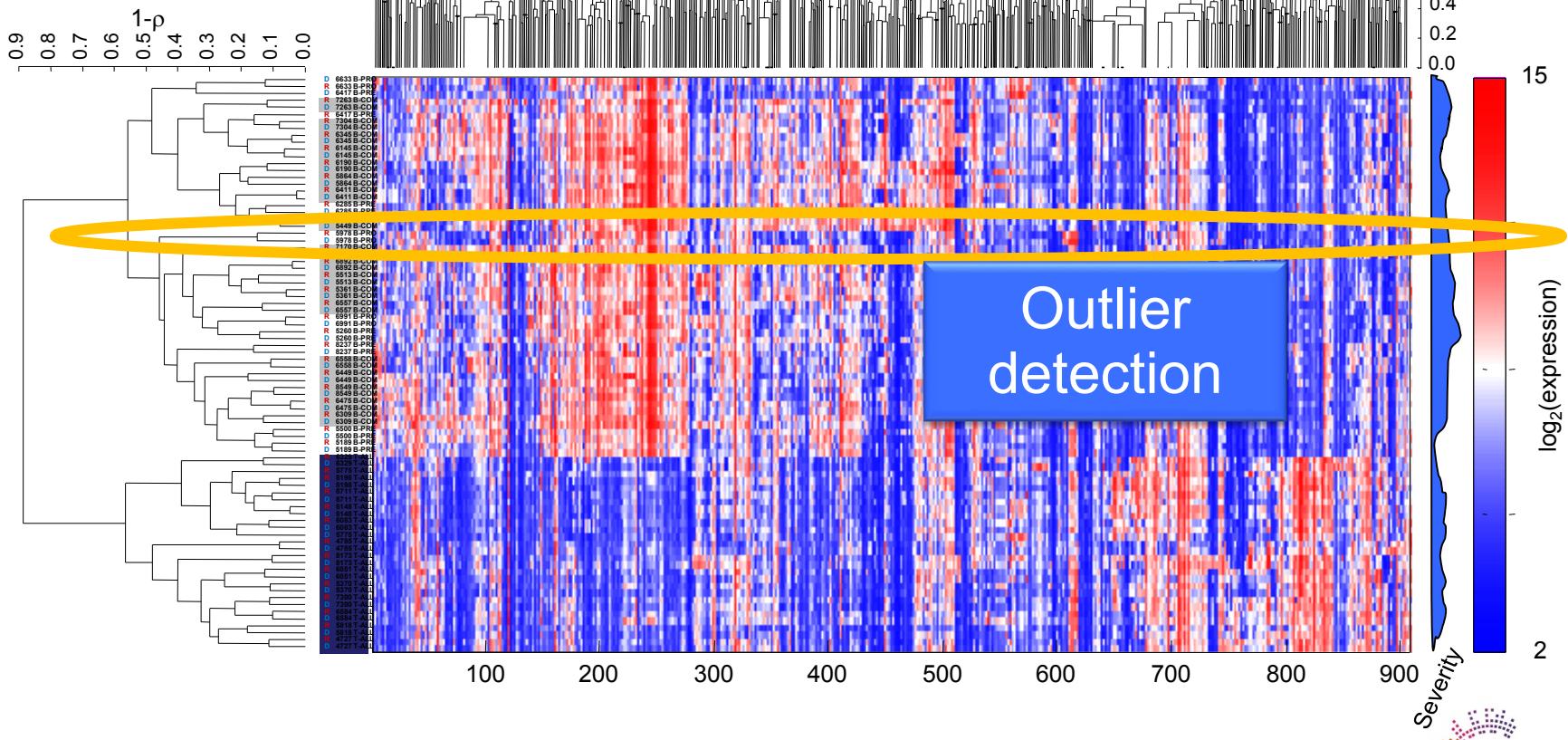
- Example:  
gene expression  
diagnostics



Clustering of genes: similar 'disruptive' processes

# Machine learning in bioinformatics

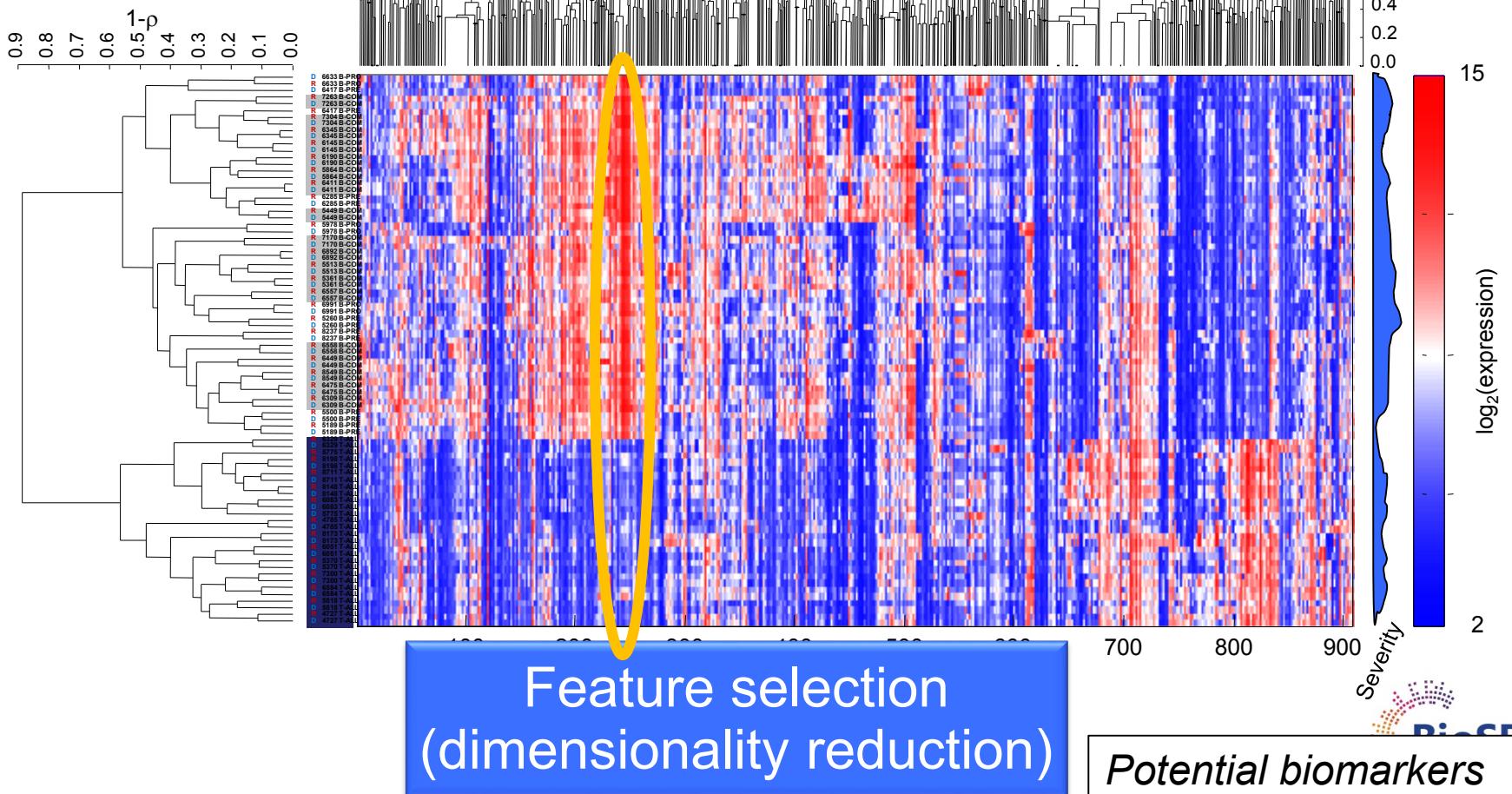
- Example:  
gene expression  
diagnostics



Technical error / rare patient-rare genetic background

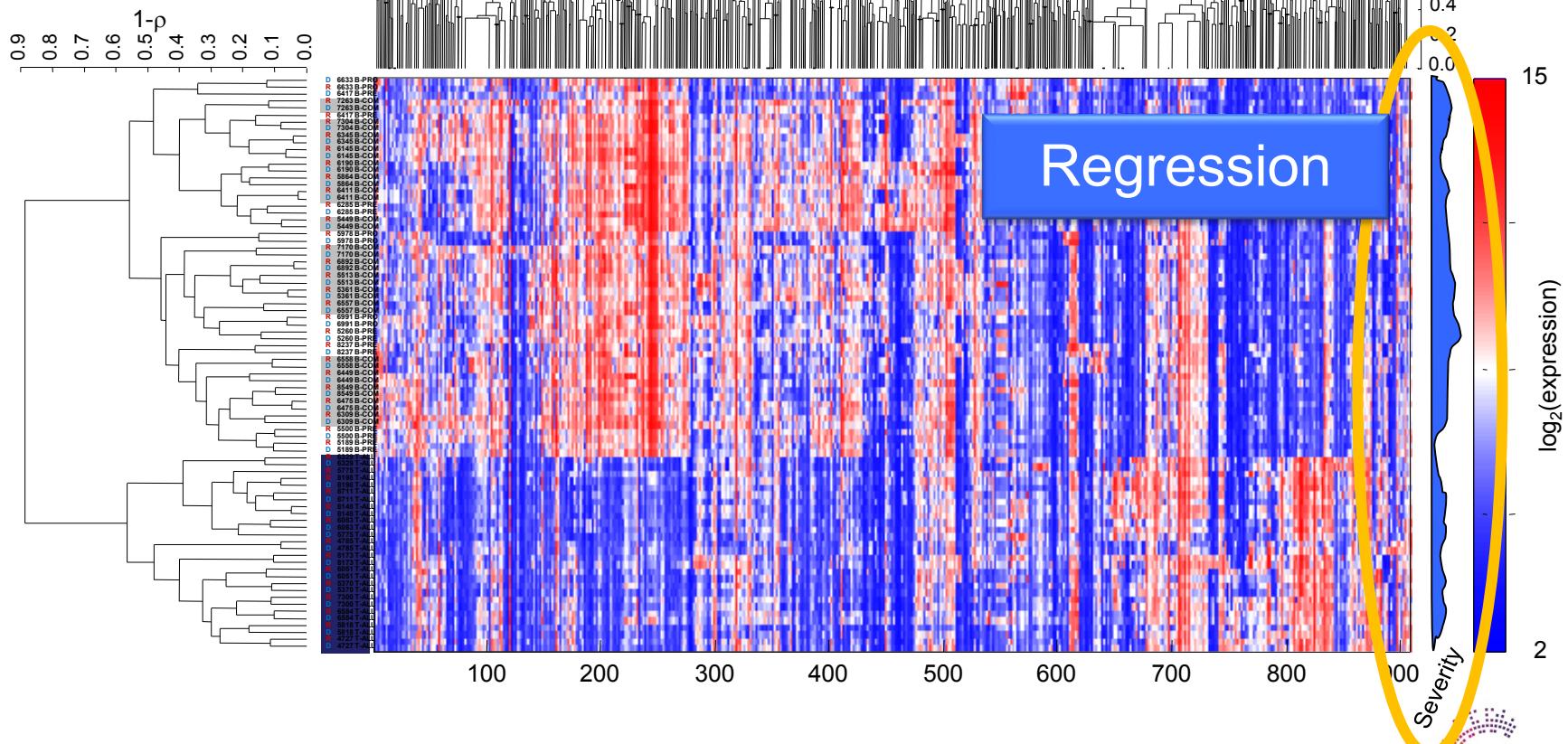
# Machine learning in bioinformatics

- Example:  
gene expression  
diagnostics



# Machine learning in bioinformatics

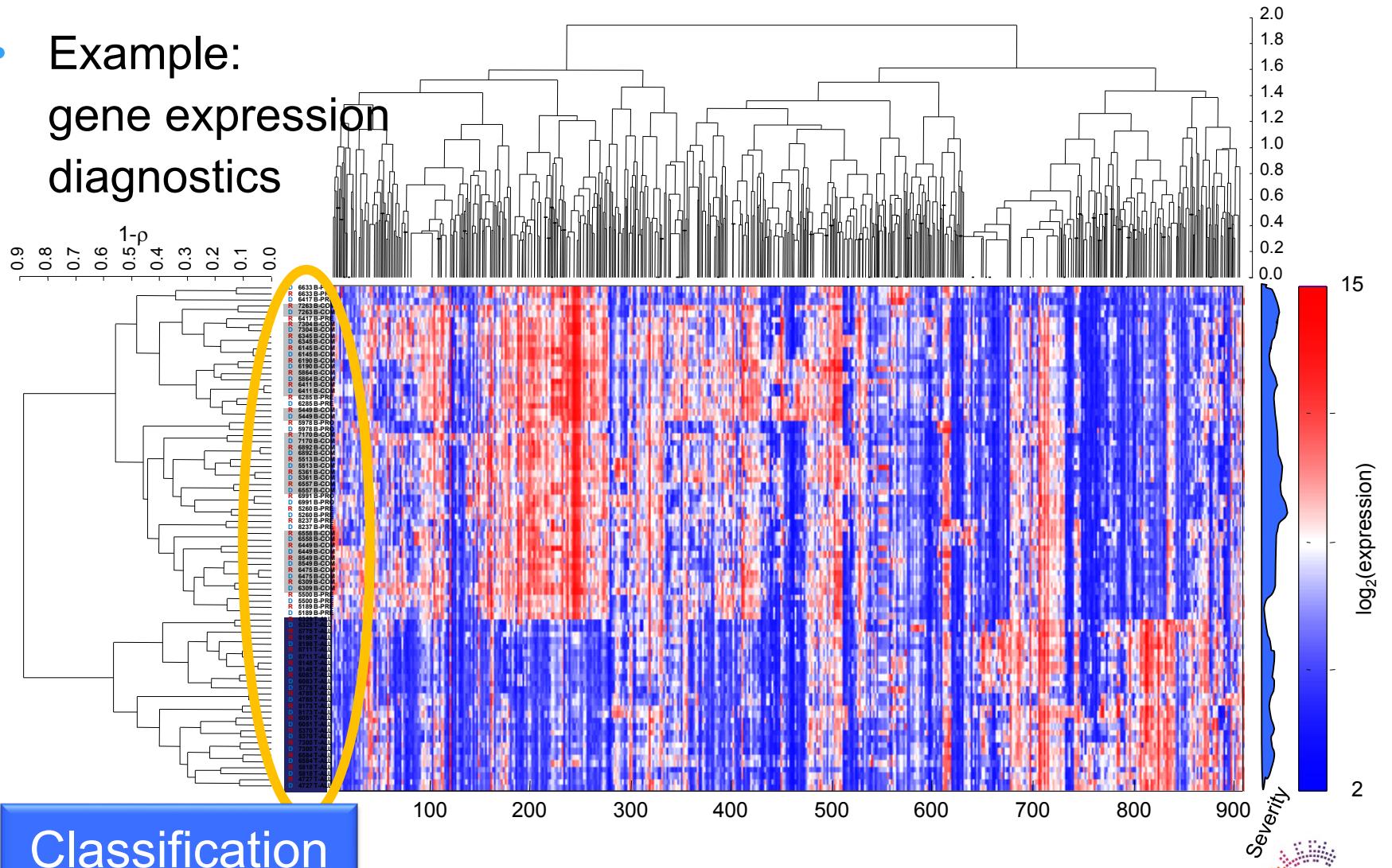
- Example:  
gene expression  
diagnostics



E.g. Predicting survival time

# Machine learning in bioinformatics

- Example:  
gene expression  
diagnostics



Classification

E.g. Predicting metastasis

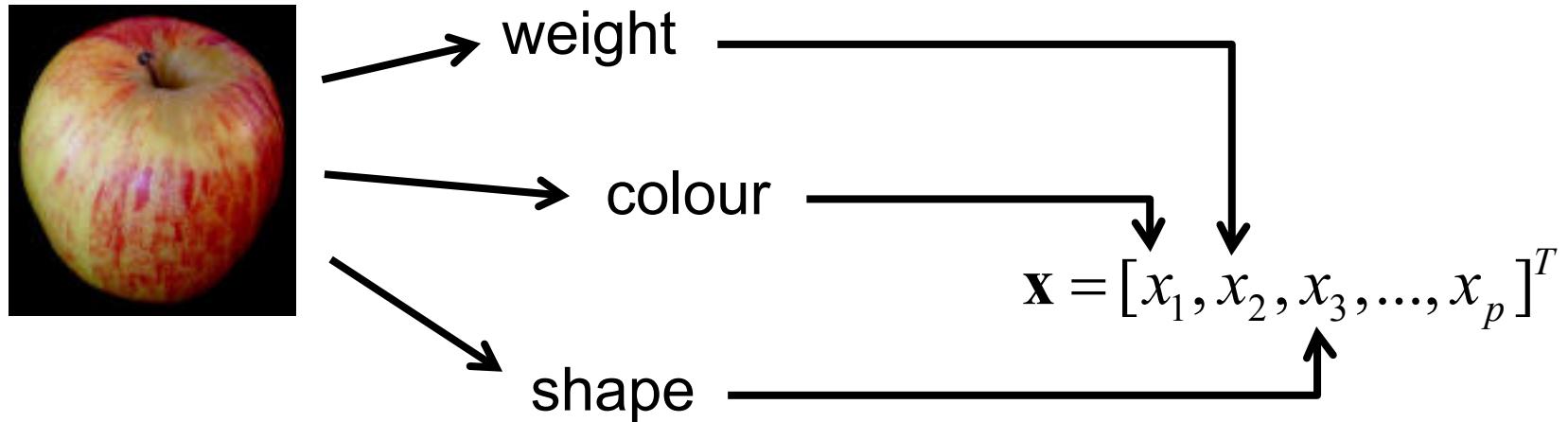
# Machine learning in bioinformatics (2)

- Tools applicable to any type of biological data
- Examples:
  - Protein sequence data:
    - Clustering: finding orthologous groups
    - Classification: prediction of EC number, subcellular localization, ...
    - Regression: predicting secondary structure
  - TF binding data (ChIP):
    - Clustering: finding functional gene groups
    - Classification: predicting gene annotation
    - Regression: finding cis-regulatory modules
  - ...

# Terminology

# Measurements and features

- To automate these tasks, we have to find a mathematical *representation* of objects
- Objects are usually represented by *features*, i.e. sets of useful *measurements* obtained from some sensors



# Measurements and features (2)

- This course assumes measurements as given, i.e. sensor accuracy etc. are not *explicitly* modeled
- However,
  - in general measurements will never be perfect
  - objects within a class will vary intrinsically
- Hence, we need statistics to model all variation

*This is important!*

*If we know everything and there is no noise, you'll need different algorithms/models*

# Datasets

- A *dataset* is a set of measurements on many objects
- For clustering:

Object	Weight	Colour
Apple #1	25	36
Apple #2	20	34
Apple #3	35	40
Pear #1	35	55
Pear #2	37	55
Pear #3	40	57
Pear #4	36	41

# Datasets

- A *dataset* is a set of measurements on many objects
- For regression:

Object	Weight	Colour	Price
Apple #1	25	36	0.21
Apple #2	20	34	0.17
Apple #3	35	40	0.33
Pear #1	35	55	0.41
Pear #2	37	55	0.26
Pear #3	40	57	0.35
Pear #4	36	41	0.29

# Datasets

- A *dataset* is a set of measurements on many objects
- For classification:

Object	Weight	Colour	Label
Apple #1	25	36	A
Apple #2	20	34	A
Apple #3	35	40	A
Pear #1	35	55	P
Pear #2	37	55	P
Pear #3	40	57	P
Pear #4	36	41	P

# Datasets

- A *dataset* is a set of measurements on many objects
- For classification:

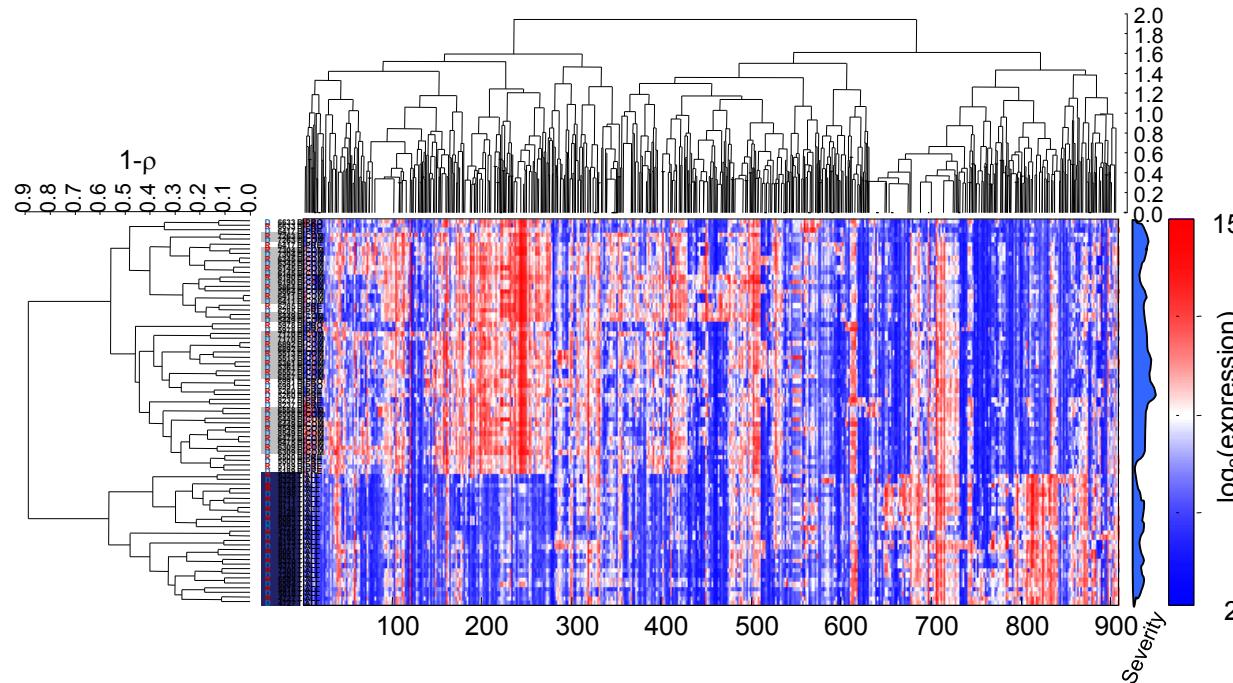
Object	Weight	Colour	Label
Apple #1	25	36	A
Apple #2	20	34	A
Apple #3	35	40	A
Pear #1	35	55	P
Pear #2	37	55	P
Pear #3	40	57	P
Pear #4	36	41	P

**measurement** **feature** **labels**

**object** **dataset**

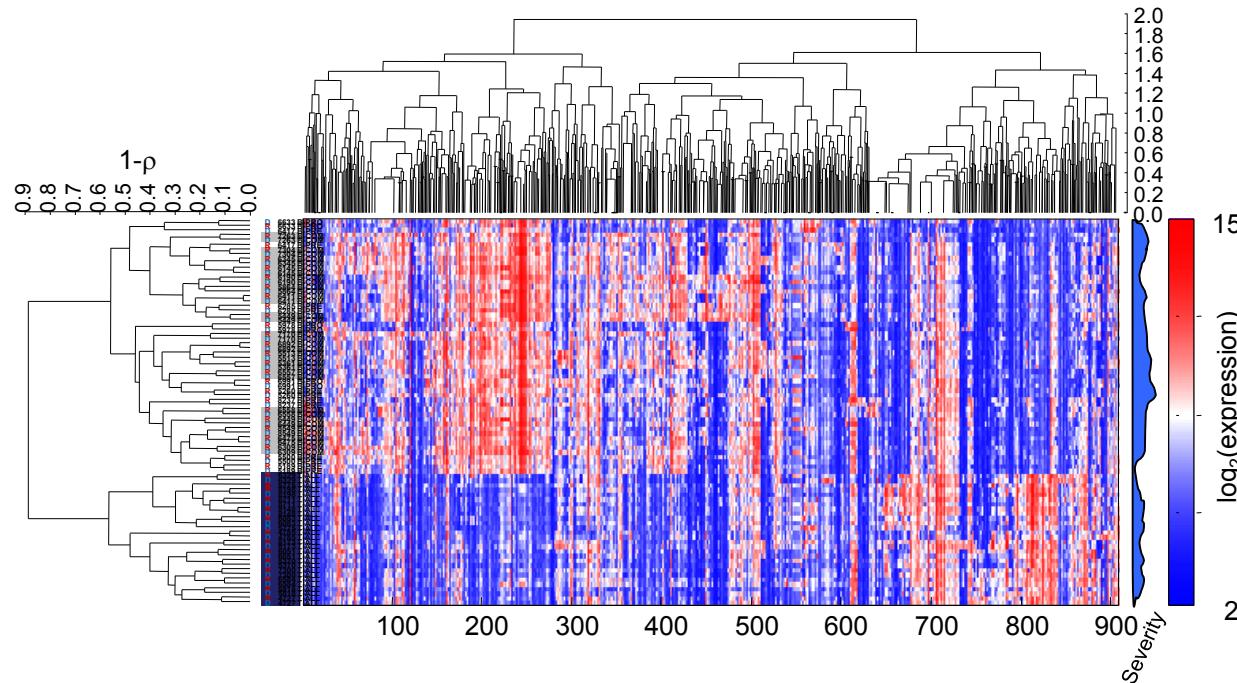
# Datasets (2)

- What objects, labels/targets and features are depends on the problem...
- Gene expression-based diagnostics:
  - object: patient
  - feature: gene expression, copy number, mutational pattern, ....
  - label: relapse; regressor/dependent variable: survival time



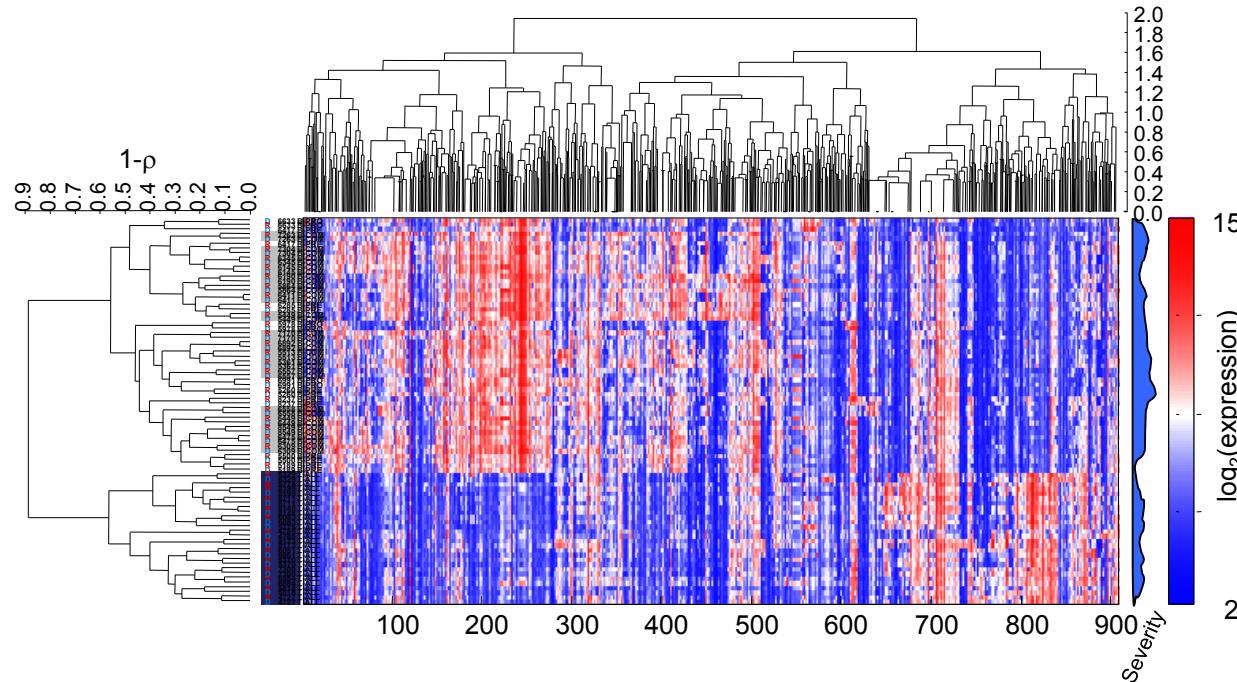
# Datasets (2)

- What objects, labels/targets and features are depends on the problem...
- Protein-protein interactions:
  - object: protein PAIR
  - feature: gene expression correlation, difference in annotation, ...
  - label: complex or not; regressor/dependent variable: binding strength



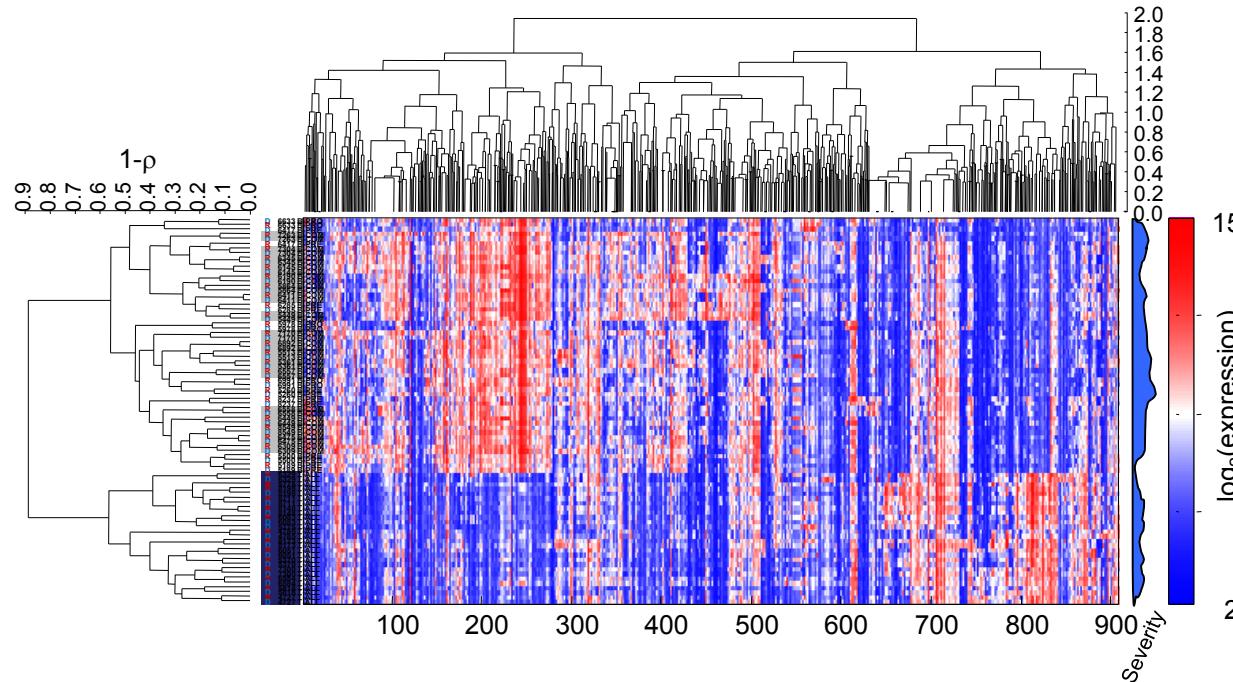
# Datasets (2)

- What objects, labels/targets and features are depends on the problem...
- Gene prediction:
  - object: gene
  - feature: sequence (representation), conservation of sequence, ...
  - label: gene or not; regressor/dependent variable: conservation



# Datasets (2)

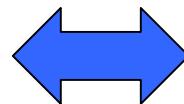
- What objects, labels/targets and features are depends on the problem...
- TFBS detection:
  - object: location on genome
  - feature: ChIP-seq, sequence features, distance to TSS ...
  - label: TFBS or not; regressor/dependent variable: specificity



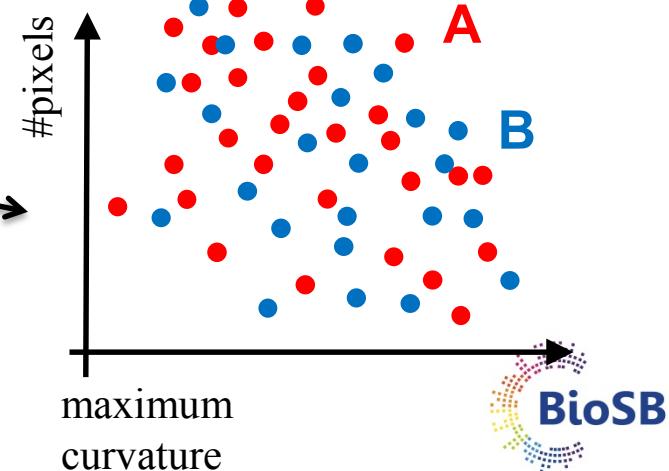
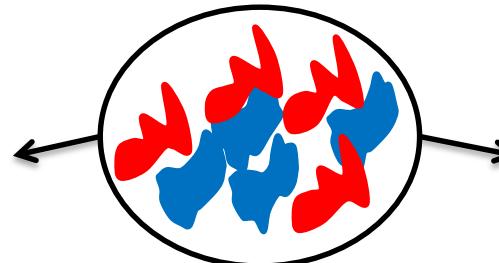
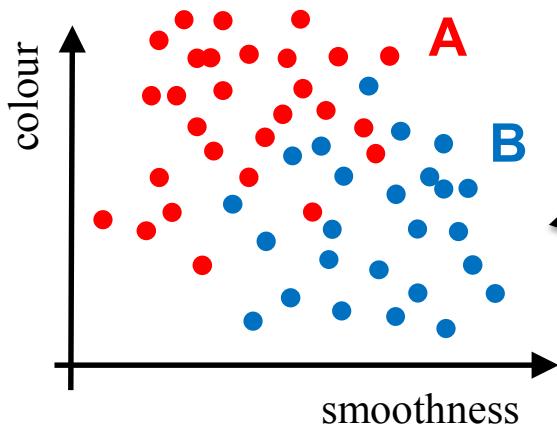
# Measurements and features (3)

- Problems

- simple
- knowledge present
- a few good features
- almost separable classes (classification) or a linear relation (regression)

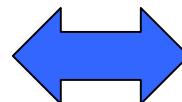


- complex
- lack of knowledge
- many poor features
- overlapping classes (classification) or highly non-linear relation (regression)



# Measurements and features (3)

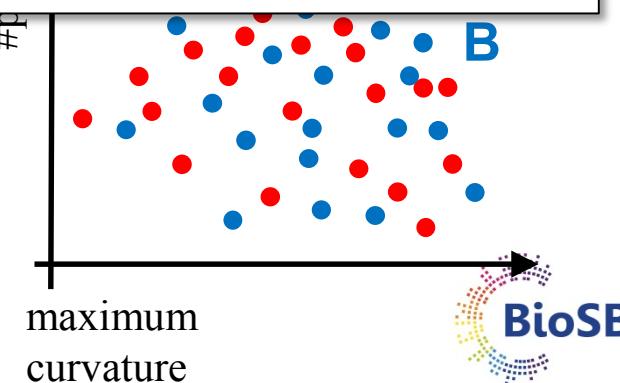
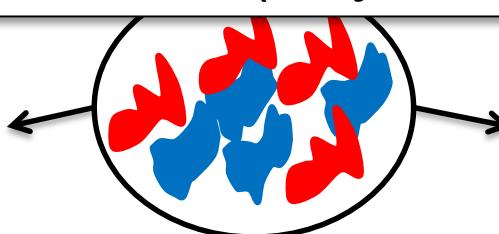
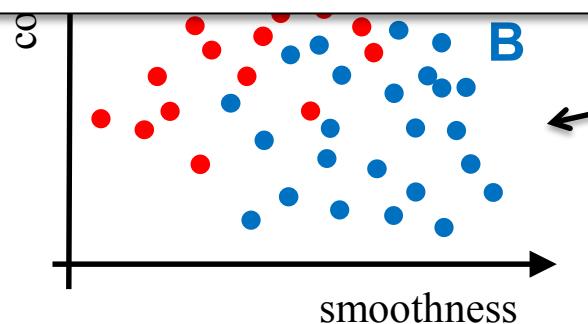
- Problems
  - simple
  - knowledge present
  - a few good features
- complex
  - lack of knowledge
  - many poor features



overlapping classes

Features (object representations) are important!

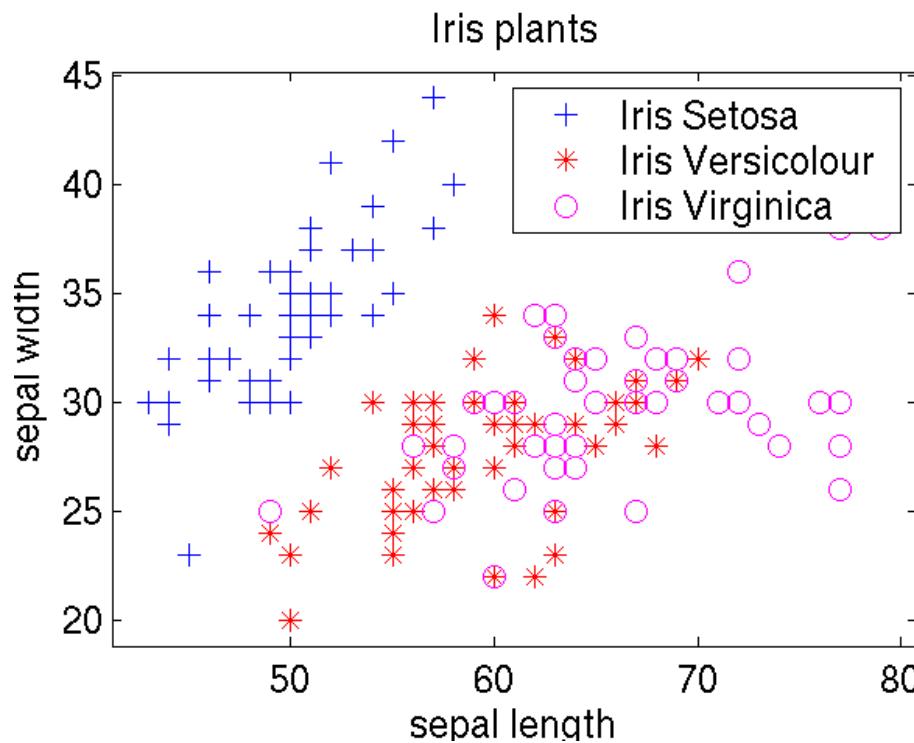
We don't deal too much with which features are measured, although we will touch upon derived features (Day 5: kernels) and learning features (Day 5: neural networks)



# Feature space

- We can interpret objects as vectors in a vector space

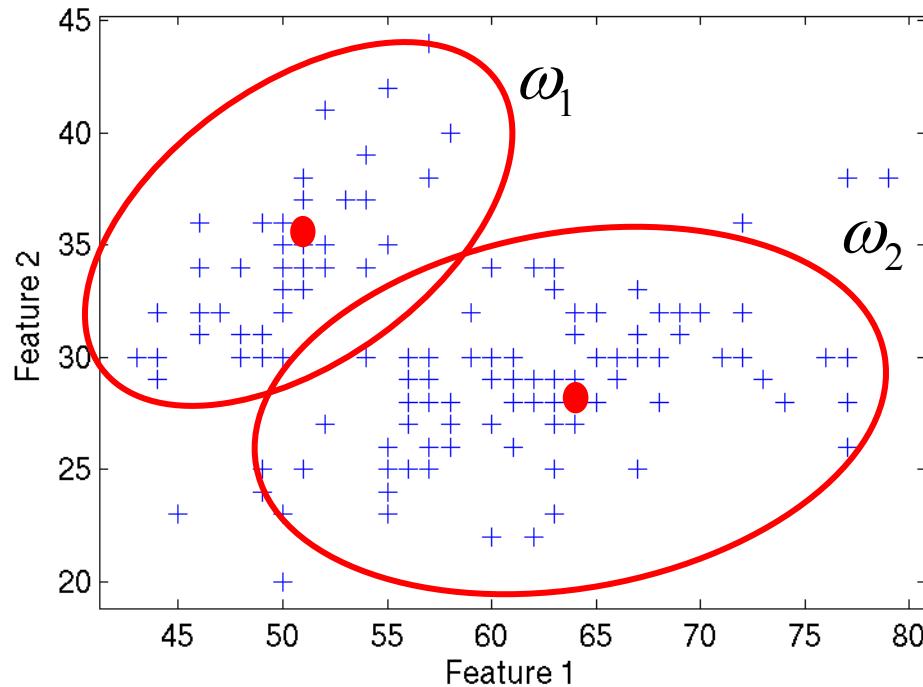
$$\mathbf{x} = [x_1, x_2, x_3, \dots, x_p]^T$$



Iris flower dataset, introduced by **Ronald Fisher (famous statistician)** in 1936 as an example of discriminant analysis

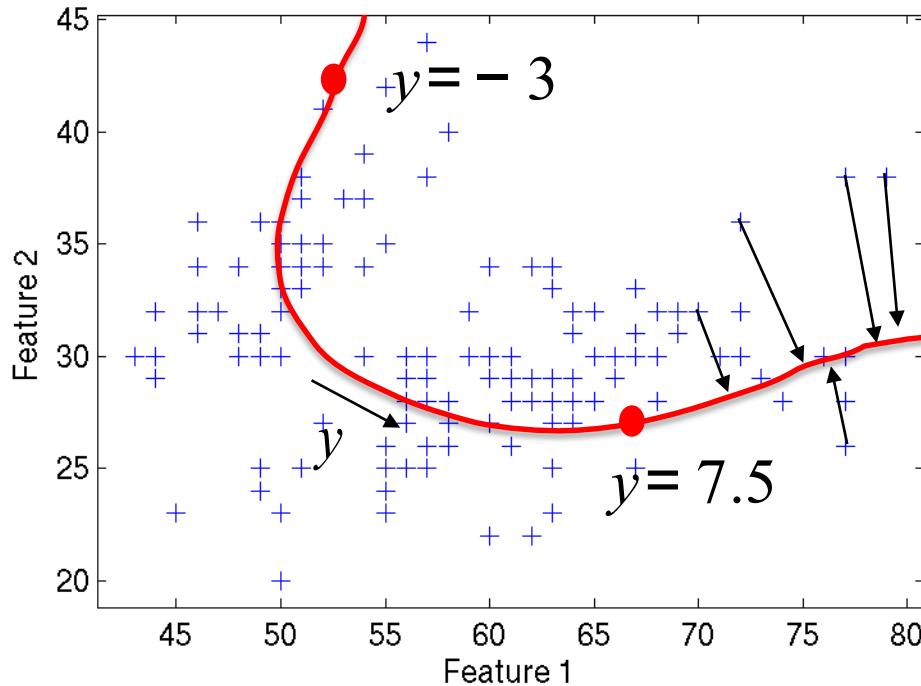
# Clustering

- Given unlabeled data  $x$ ,  
find labels  $\omega$  for natural groups in the data



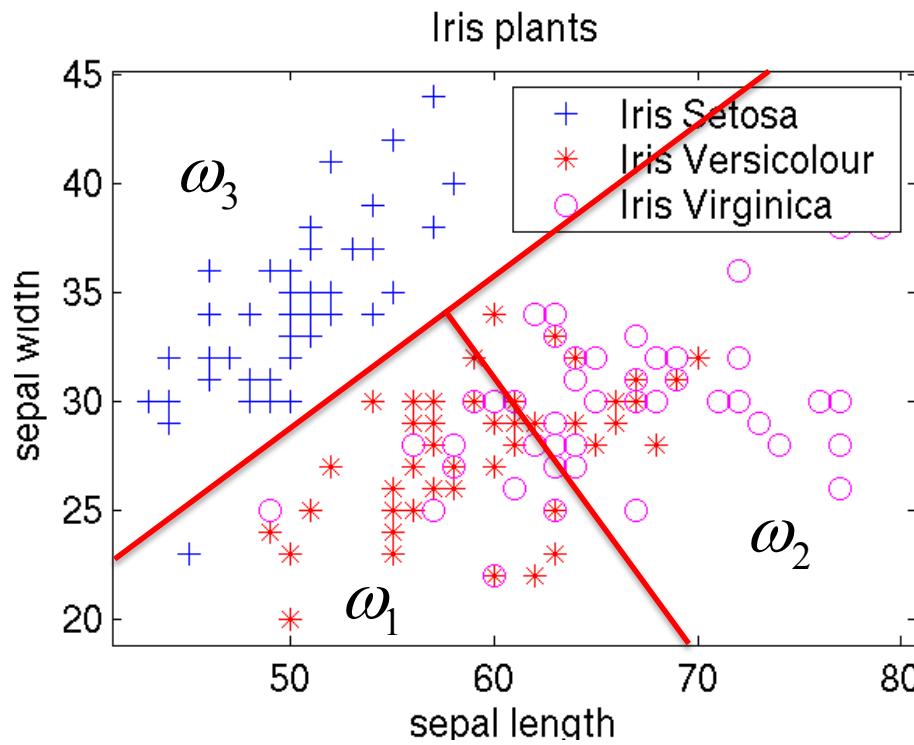
# Dimensionality reduction

- Given unlabeled data  $x$ ,  
map it to a lower dimensional feature vector  $y$



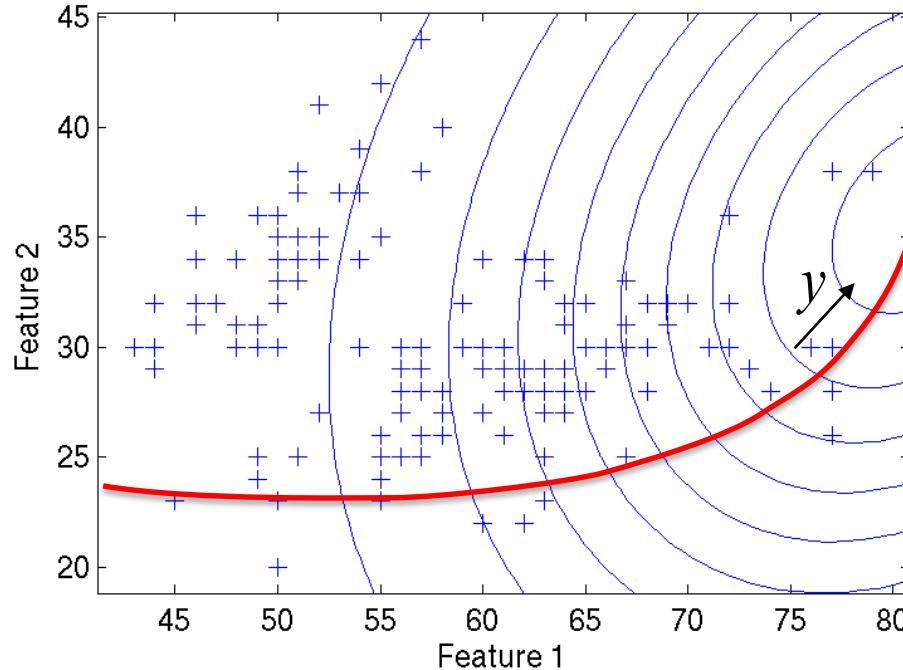
# Classification

- Given labeled data  $x$ ,  
assign each point in feature space to a class  $\omega_i$   
(in effect partitioning the feature space)



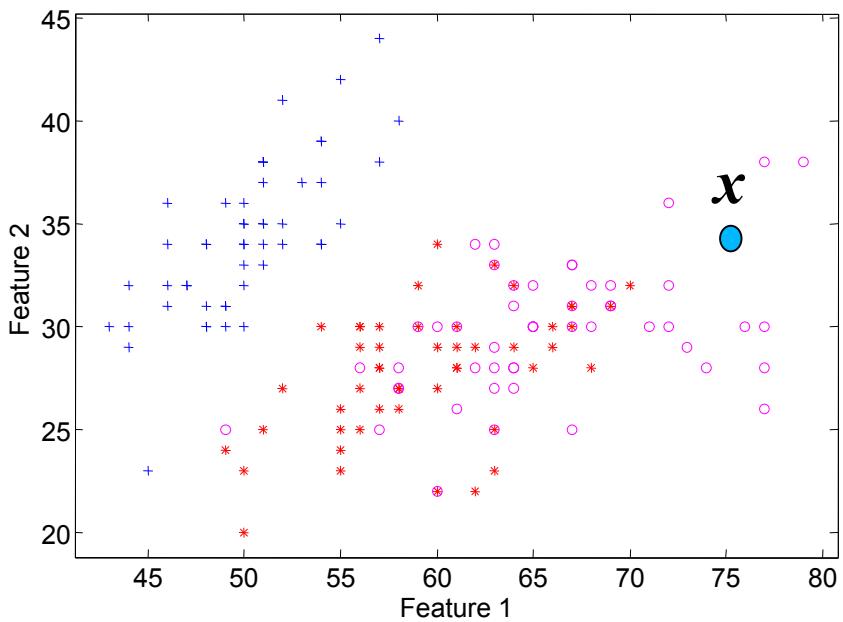
# Regression

- Given labeled data  $x$ ,  
assign each point in feature space a real-valued output  $y$



# General model

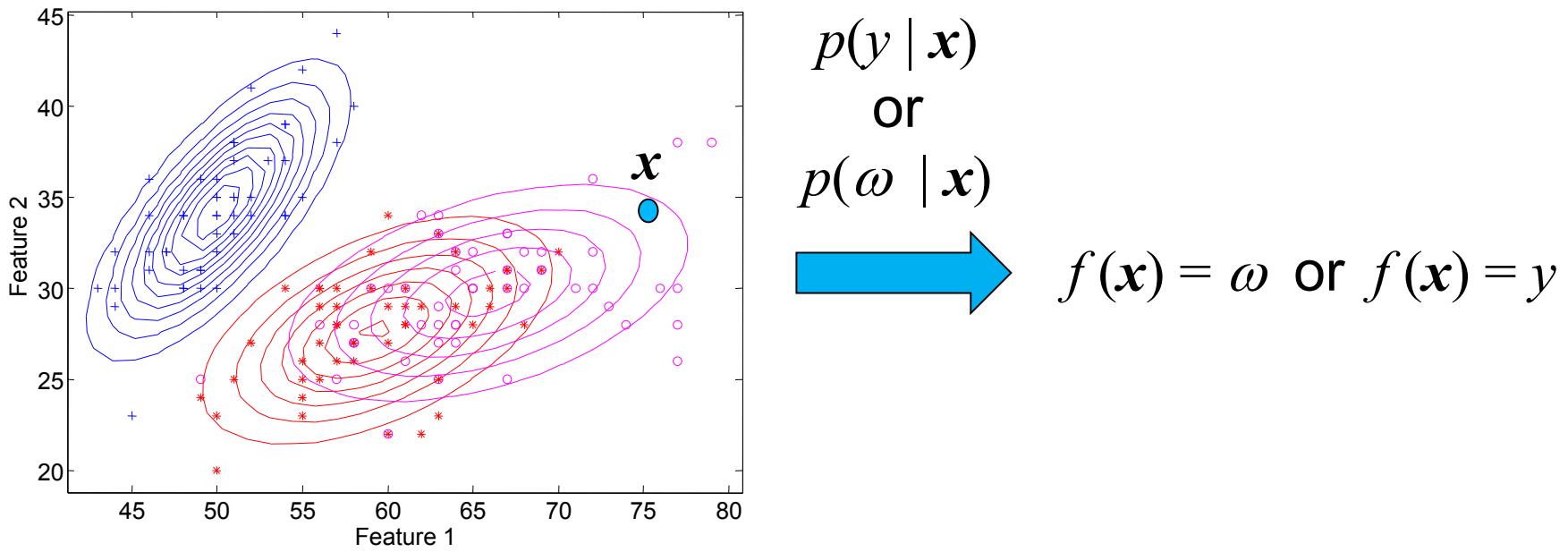
- Construct a model  $f(x)$  that outputs  $\omega$  or  $y$
- This model should be fit to the data



$$f(\mathbf{x}) = \omega \text{ or } f(\mathbf{x}) = y$$

# General model (2)

- Construct a model  $f(\mathbf{x})$  that outputs  $\omega$  or  $y$
- This model should be fit to the data
- Ideally, we know  $p(y | \mathbf{x})$  or  $p(\omega | \mathbf{x})$  over the entire feature space

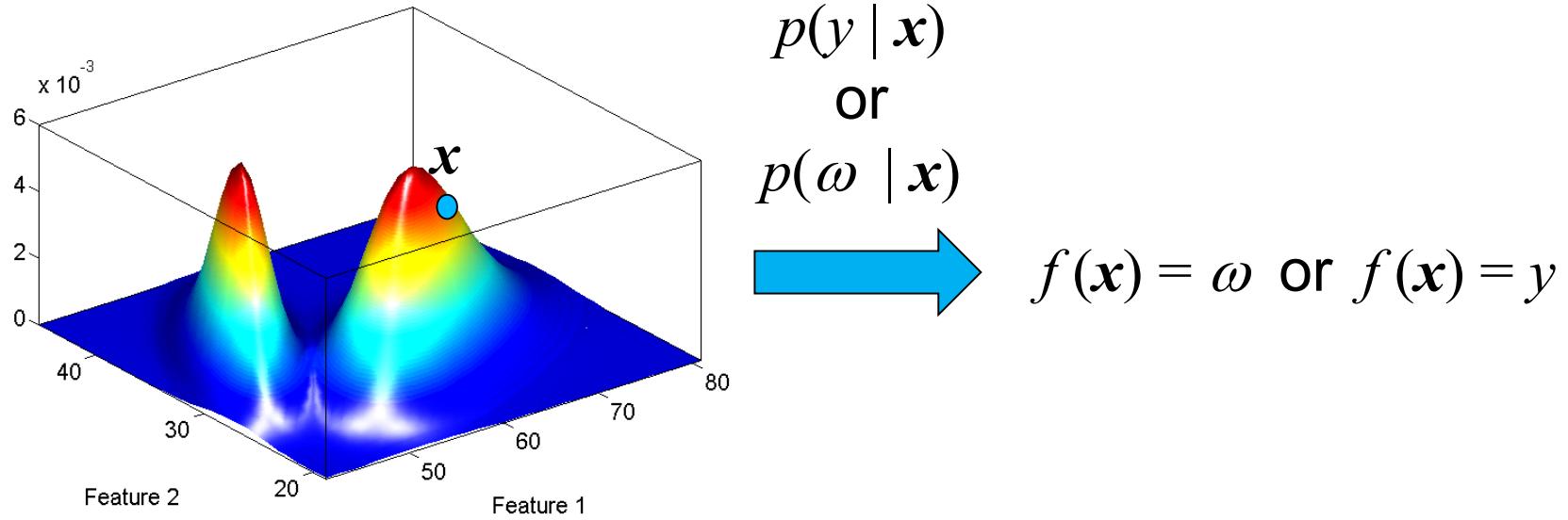


if we know the probability distributions, we can make the most informed decision



# General model (3)

- Construct a model  $f(\mathbf{x})$  that outputs  $\omega$  or  $y$
- This model should be fit to the data
- Ideally, we know  $p(y | \mathbf{x})$  or  $p(\omega | \mathbf{x})$  over the entire feature space



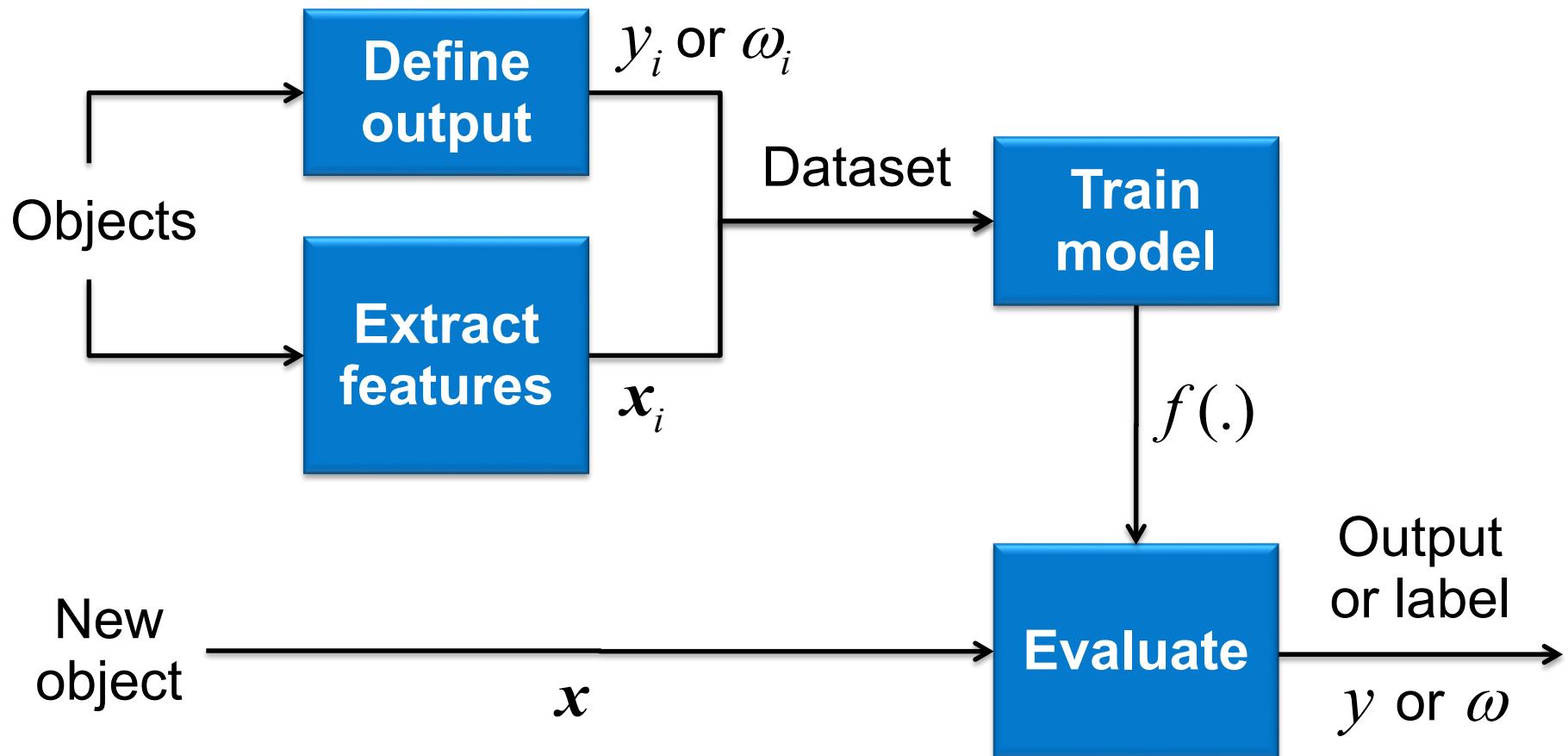
if we know the probability distributions, we can make the most informed decision



# General model (4)

- Clustering: find cluster labels  $\omega$  given object  $x$   
fit model using dataset  $\{x_i\}$   $p(\omega | x)$
- Dimensionality reduction: find mapping  $y$  given object  $x$   
fit model using dataset  $\{x_i\}$   $p(y | x)$
- Classification: find class labels  $\omega$  given object  $x$   
fit model using dataset  $\{x_i, \omega_i\}$   $p(\omega | x)$
- Regression: find target  $y$  given object  $x$   
fit model using dataset  $\{x_i, y_i\}$   $p(y | x)$

# Machine learning pipeline





10min break

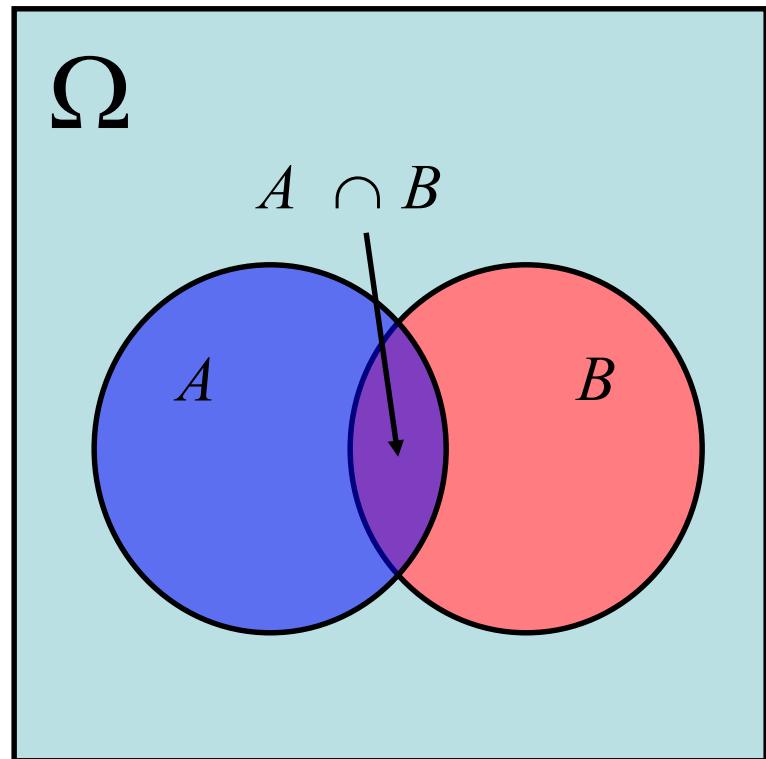
# Statistics

# Required background

- The course is aimed at PhD students with a background in bioinformatics, systems biology, computer science or a related field, and life sciences. A working knowledge of basic statistics and linear algebra is assumed.
- Self-assessment; if you have problems, read the primers
- Now, a brief recap

# Recall: probability

- $\Omega$  : all possible outcomes (sample space)  
e.g. the number of eyes on a dice: 1, 2, 3, 4, 5, 6
- $A \in \Omega$  : event  
e.g. “throwing a 3”
- $P$  : probability measure
  - $0 \leq P(A) \leq 1$
  - $P(\Omega) = 1$
  - $P(A \cup B) = P(A) + P(B) - P(A \cap B)$
  - E.g.  $P(A) = 1/6$



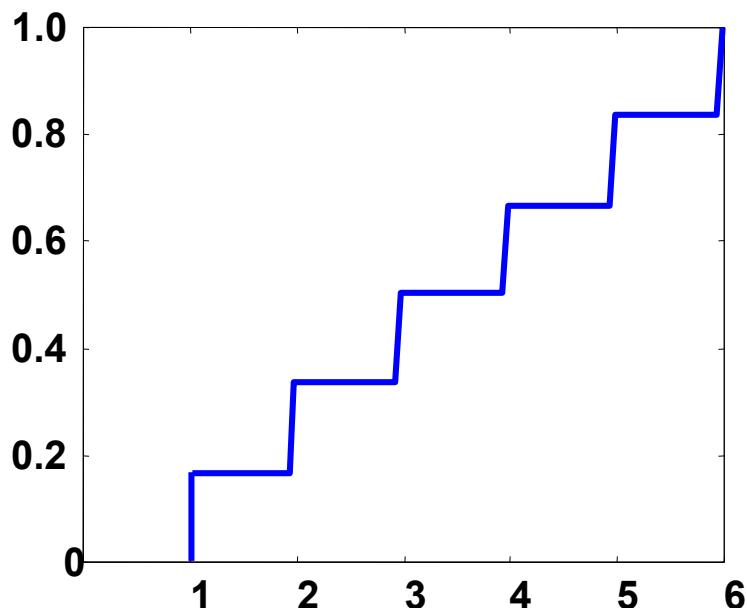
# Recall: probability (3)

- Subjective approach:  
“the probability of  $A$  is a number between 0 and 1 indicating how likely people believe  $A$  to be true”
- Frequentist approach:  
“the probability of  $A$  is a number between 0 and 1 indicating the average ratio of  $A$  being true in a large number of repeated experiments”
- Is really a philosophical debate...  
the “right” approach depends on the problem and the data available

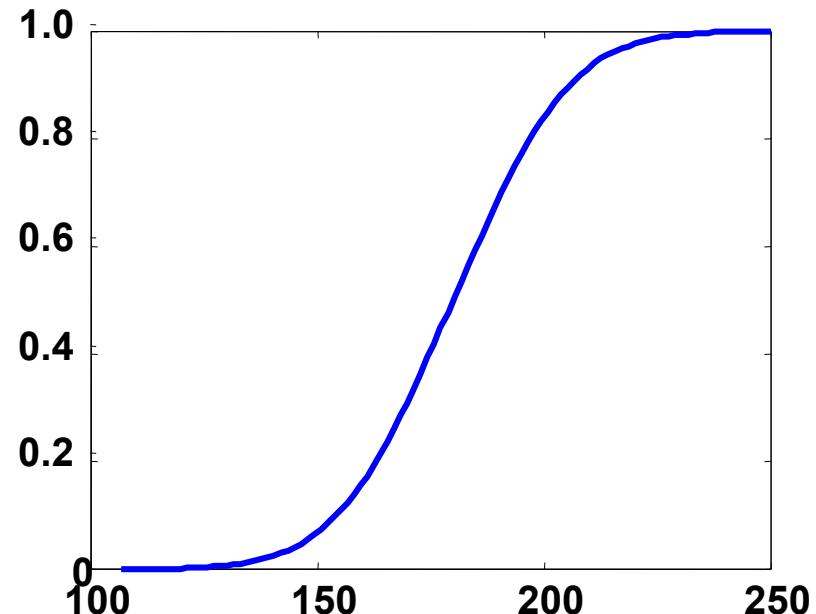
*problems (can) arise in interpretation: what does it mean?*

# Recall: CDFs

- Cumulative distribution function
- $P_X(x) = F(x)$  : probability that  $X \leq x$ ,  $\mathbb{R} \rightarrow [0,1]$



e.g. 10,000 dice throws



10,000 body lengths

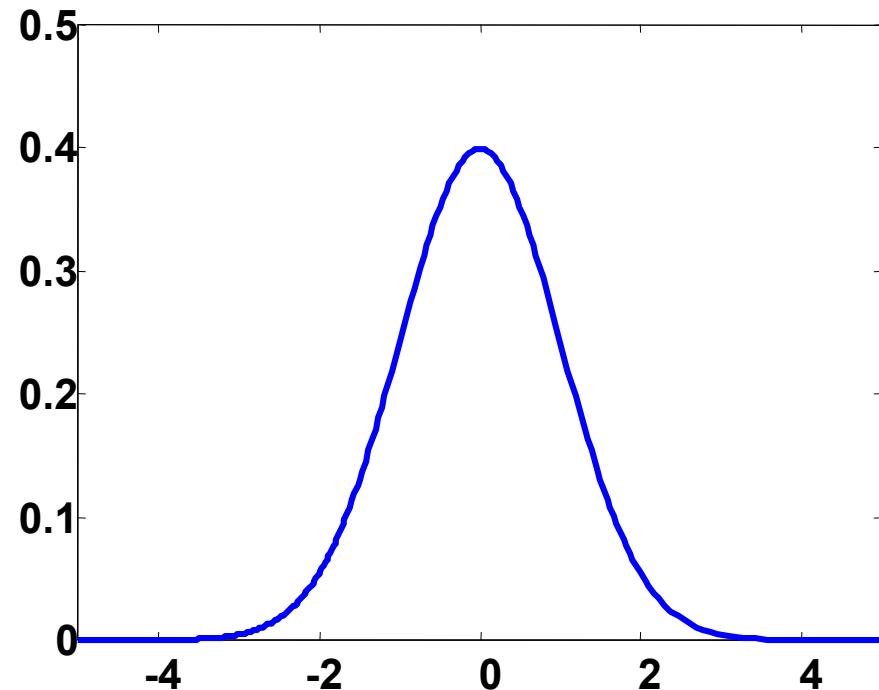
# Recall: PDFs

- $p(x) = \frac{dP(x)}{dx}$  : probability density function

- $p(x) \geq 0$

- $\int_{-\infty}^{\infty} p(x)dx = 1$

- $\int_a^b p(x)dx = P(a \leq x \leq b)$



- **$p(x)$  is not the probability of  $X$  being  $x$  !**

# Recall: expectation

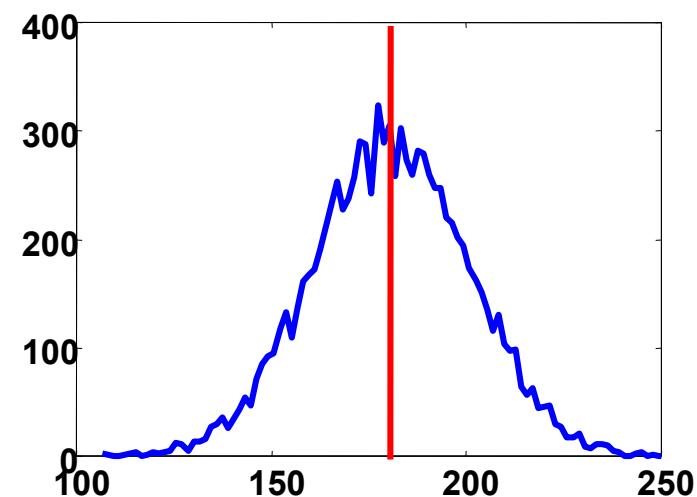
- Expectation: mean of distribution,

$$\mu = \text{E}[X] = \int_{-\infty}^{\infty} x \ p(x) \ dx$$

- Note: expectations are over entire distributions; on data sets  $\{x\}$  we can only estimate the mean,

$$m = \hat{\mu} = \frac{1}{N} \sum_{i=1}^N x_i$$

- $\text{E}[c] = c$
- $\text{E}[aX + bY] = a \text{ E}[X] + b \text{ E}[Y]$



*Important to realize that estimates are always based on a finite dataset!  
m is an estimate(!) of  $\mu$ ; that is why there is a hat!*

# Recall: variance

- Variance: average deviation from expected value,

$$\sigma^2 = \text{var}(X) = \int_{-\infty}^{\infty} (x - \mu)^2 p(x) \, dx$$

or

$$\sigma^2 = E[(X - E(X))^2] = E[X^2] - (E[X])^2$$

- $\sigma$  is called the standard deviation

- $\text{var}(X) \geq 0$
- $\text{var}(c) = 0$
- $\text{var}(aX) = a^2 \text{ var}(X)$

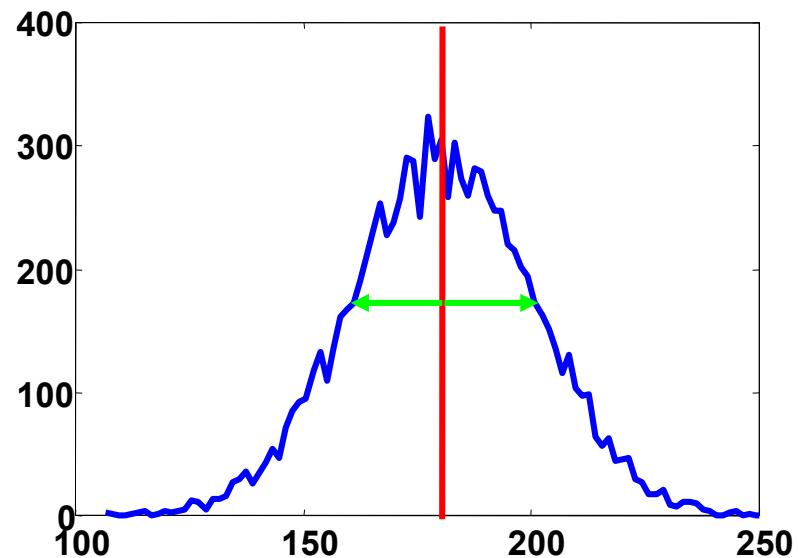
# Recall: variance (2)

- Again, on data sets  $\{x\}$  we can only estimate the variance:

$$s^2 = \hat{\sigma}^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \hat{\mu})^2$$

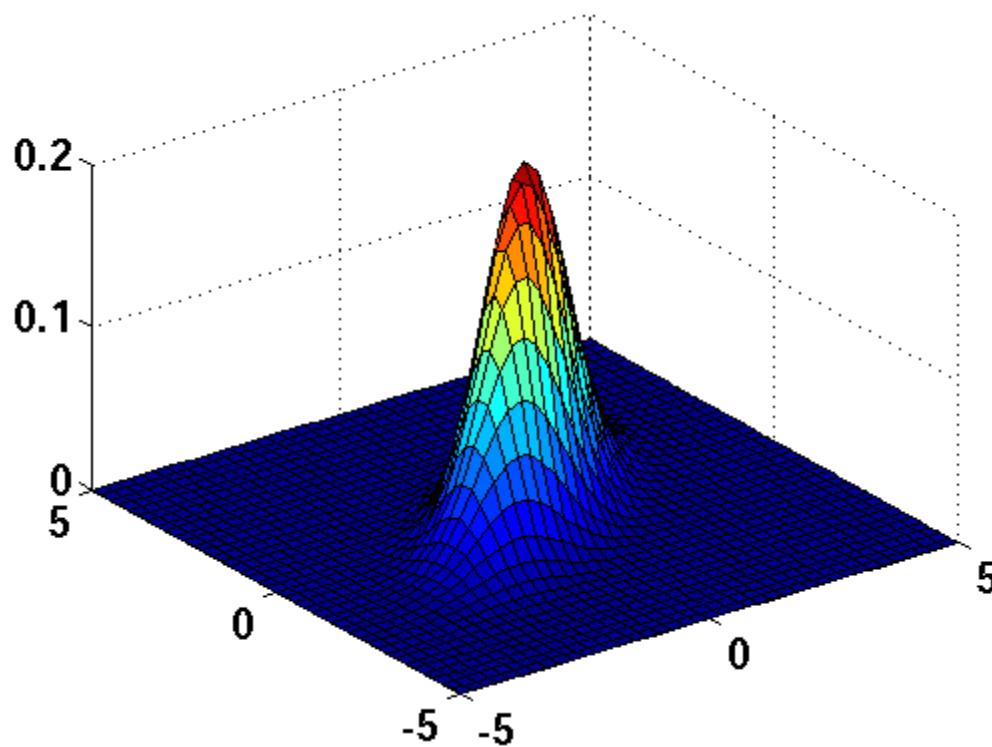
- Usually, this unbiased estimator is used:

$$s^2 = \hat{\sigma}^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \hat{\mu})^2$$



# Recall: joint distributions

- For  $p > 1$  measurements  $x = (x_1, \dots, x_p)$ ,  
joint distributions & densities:



# Recall: covariance

- Covariance: measure of how two random variables vary together,

$$\begin{aligned}\text{cov}(X, Y) &= E[(X - E(X))(Y - E(Y))] \\ &= E[XY] - E[X]E[Y]\end{aligned}$$

- Correlation: normalised covariance,

$$\rho(X, Y) = \frac{\text{cov}(X, Y)}{\sqrt{\text{var}(X)\text{var}(Y)}} \in [-1, 1]$$

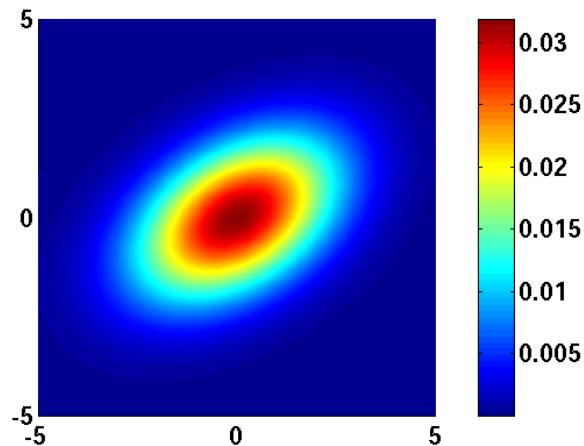
- $\text{cov}(X, Y) = 0$  :  $X$  and  $Y$  are uncorrelated

# Recall: covariance (2)

- For a set of random variables  $X_1 \dots X_p$ , we can calculate a covariance matrix,

$$\Sigma = \begin{bmatrix} \text{cov}(X_1, X_1) & \text{cov}(X_1, X_2) & \dots & \text{cov}(X_1, X_p) \\ \text{cov}(X_2, X_1) & \dots & \dots & \text{cov}(X_2, X_p) \\ \dots & \dots & \dots & \dots \\ \text{cov}(X_p, X_1) & \text{cov}(X_p, X_2) & \dots & \text{cov}(X_p, X_p) \end{bmatrix}$$

e.g.

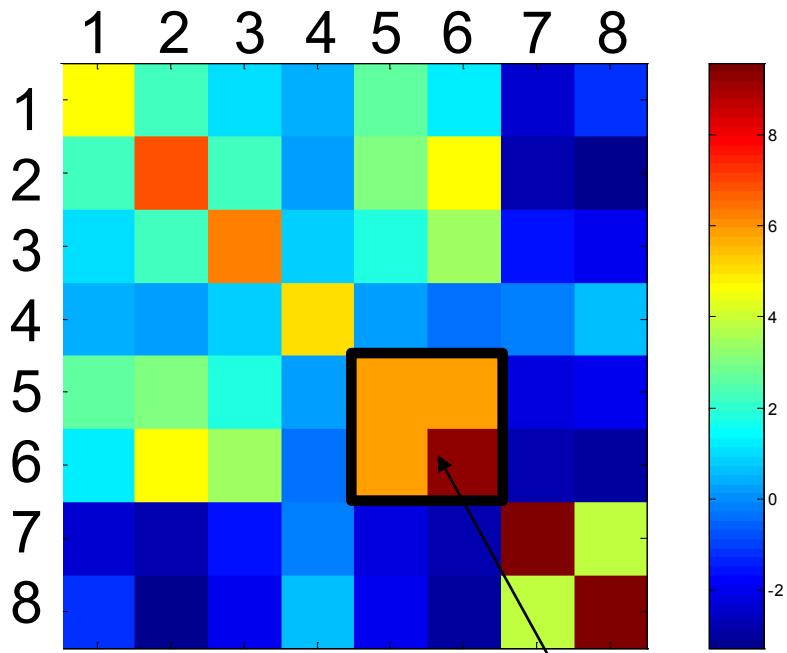


$$\longrightarrow \Sigma = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}$$

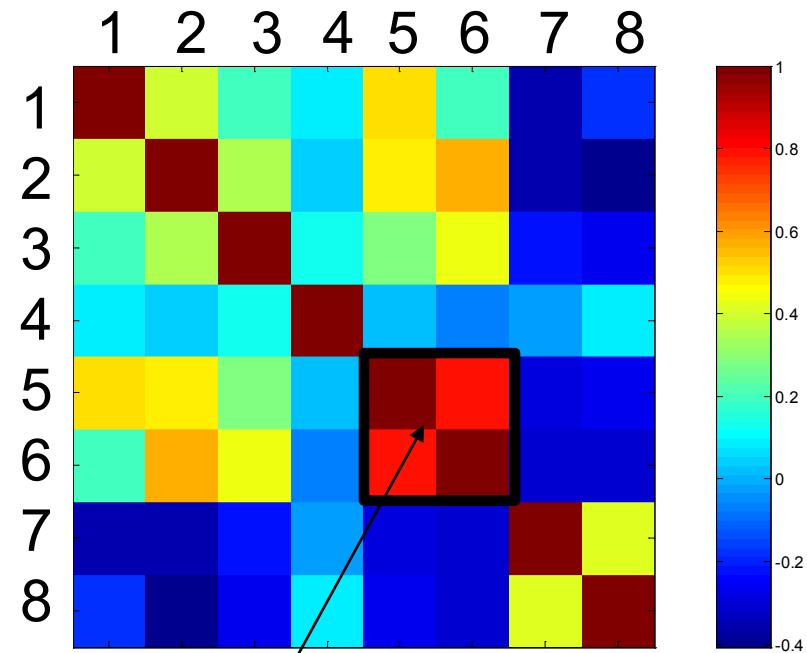
*Pairwise covariance of all features!*

# Recall: covariance (3)

- Example: IMOX data (images of handwritten digits 1:8)



```
imagesc(cov(+a))
```

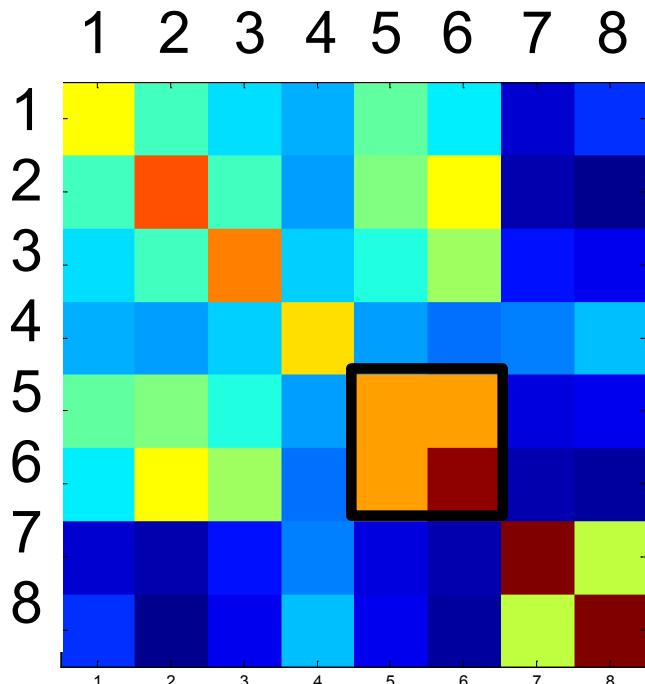


```
imagesc(corrcoef(+a))
```

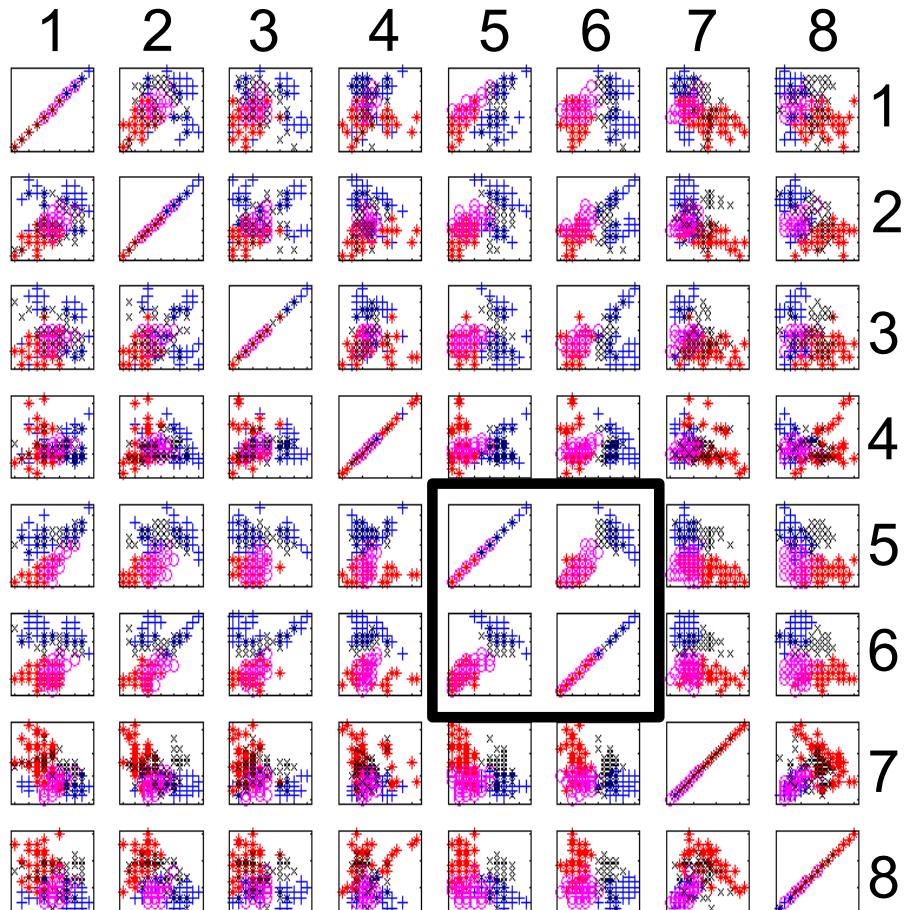
characters 5/6 are alike

# Recall: covariance (4)

- Example: IMOX data



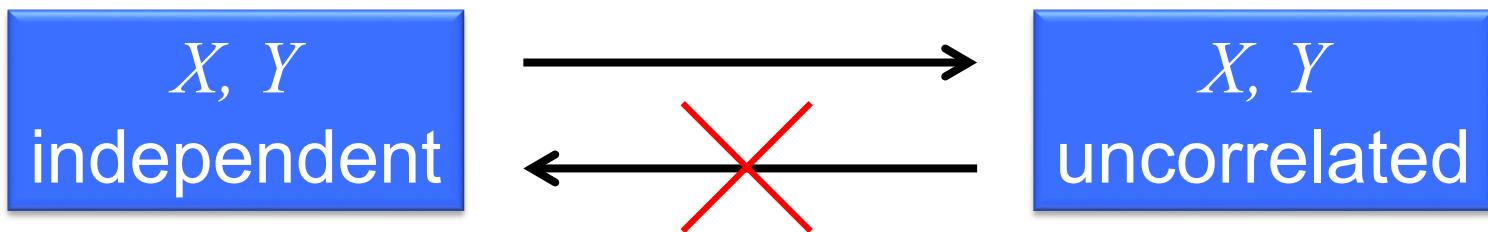
```
imagesc(cov(+a))
```



```
scattered(a,'gridded')
```

# Recall: independence

- Important concept: often needed as assumption!
- Two events  $A$  and  $B$  are independent iff  
$$P(A \cap B) = P(A) P(B)$$
- Two random variables  $X$  and  $Y$  are independent iff  
$$p(x,y) = p(x) p(y)$$



- Uncorrelated: “there’s no *linear* dependence”  
Independent: “there’s no dependence at all”

# Recall: Bayes' theorem

- Conditional probability of  $A$  given  $B$ ,

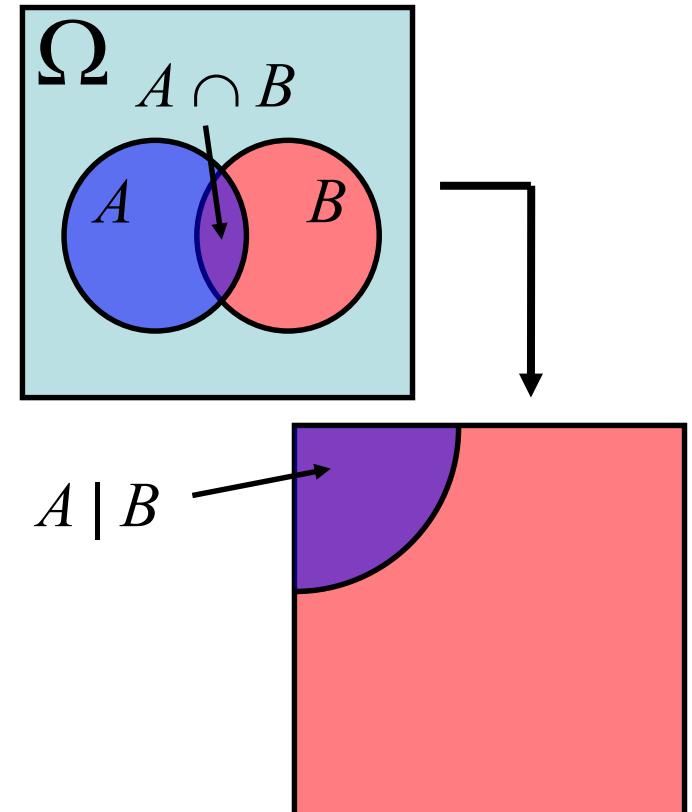
$$P(A | B) = \frac{P(A \cap B)}{P(B)}$$

- As a consequence,

$$\begin{aligned} P(A \cap B) &= P(A | B)P(B) \\ &= P(B | A)P(A) \end{aligned}$$

- Bayes' theorem:

$$P(A | B) = \frac{P(B | A)P(A)}{P(B)}$$



# Bayes' theorem (2)

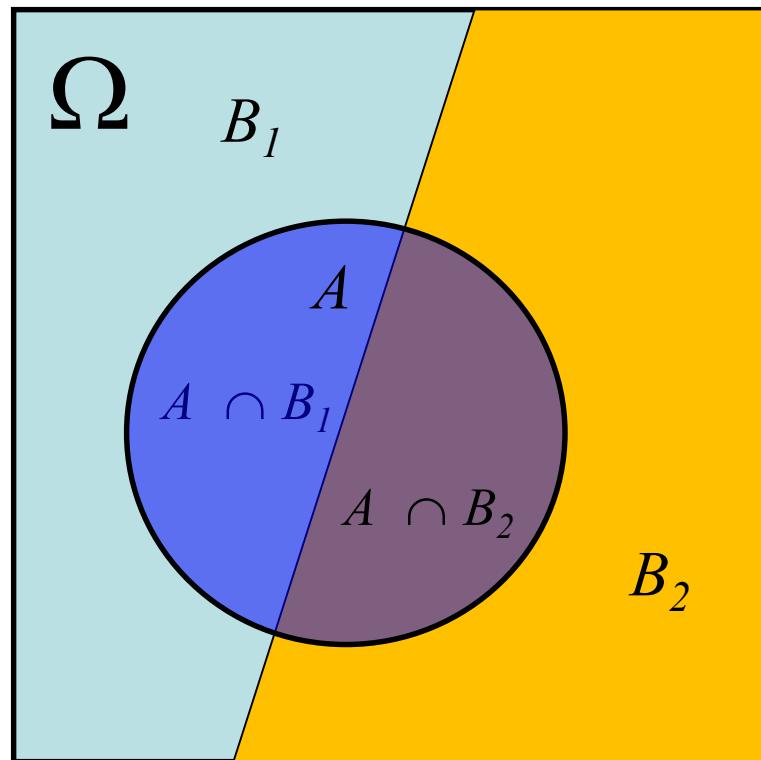
- Bayes' theorem is very useful, but controversial:
  - reverses causality
  - introduces subjective (prior) probabilities

$$P(\text{cause} | \text{effect}) = \frac{P(\text{effect} | \text{cause})P(\text{cause})}{P(\text{effect})}$$

- ... but the cornerstone of pattern recognition and machine learning
  - $P(\text{disease}|\text{temperature}) = \frac{P(\text{temperature}|\text{disease})P(\text{disease})}{P(\text{temperature})}$
  - What is P (disease)? How to measure / know?

# Recall: total probability

- Total probability:
- $P(A) = \sum_{\forall B_i} P(A \cap B_i)$
- $P(A) = \sum_{\forall B_i} P(A|B_i)P(B_i)$



# Bayesian estimation

## Bayes' theorem (3)

- In statistical learning, we want to know  $p(y | x)$  so that we can predict (for example) the most probable output  $y$  for a given input  $x$
- Problem: this is often very hard to model or estimate...
  - Predict gender based on height measurement:  
 $p(\text{gender}|\text{height})?$
  - Predict fruit type based on color measurement:  
 $p(\text{fruit}|\text{color})?$
  - Predict temperature based on thermometer reading:  
 $p(\text{temperature}|\text{thermometer reading})?$

*problem is that you need to measure too much:  
for every **height** you need a number of examples of different genders  
feature = continuous & class label not*

# Bayes' theorem (4)

- Solution: combine probabilities
  - $y$  = cause, outcome, target, label ( $\omega$ ), ...
  - $x$  = effect, measurement, feature, ...

$$p(y | x) = \frac{\underbrace{p(x | y)p(y)}_{\substack{\text{conditional} \\ \text{probability}}}}{\underbrace{p(x)}_{\substack{\text{prior} \\ \text{probability}}}}$$

*posterior probability*      *normalisation*

We update our prior belief (prior) using observations (conditional)

# Bayes' theorem (5)

- Classification example  $p(\omega | x)$  :
  - $\omega \in \{ \text{'man'}, \text{'woman'} \}$  = label
  - $x \in \mathbb{R}^1$  = height measurement(m)
- $p(\omega)$  : prior probability of seeing a ‘man’ or a ‘woman’  
here: ...?
- $p(x|\omega)$  : density of  $x$  (height) when the person is actually  
a ‘man’ or a ‘woman’  
 $p(x)$  : density of height measurement  $x$   
here (total probability):

$$p(x) = \sum_i p(x | \omega_i) p(\omega_i)$$

*Issue: Prior for man/woman? In NL? In Delft? In classroom?*

# Bayes' theorem (6)

- Regression example  $p(y | x)$  :
  - $y \in \mathbb{R}^1$  = outside temperature ( $^{\circ}\text{C}$ )
  - $x \in \mathbb{R}^1$  = thermometer measurement ( $^{\circ}\text{C}$ )
- $p(y)$  : prior probability of having some outside temperature  $y$   
 $p(x|y)$  : density of  $x$  (measured temperature) when outside temperature is actually  $y$
- $p(x)$  : density of a certain thermometer measurement  $x$   
here:

$$p(x) = \int p(x | y)p(y)dy$$

Again: Prior for outside temperature in NL, this month?

Realise that  $p(x|y)$  is still difficult to estimate because now  $y$  is continuous.

# Bayesian estimation

- Estimate prior,  $p(y)$ , and conditional,  $p(x|y)$
- Use this to obtain posterior,  $p(y|x)$
- Construct a cost function  $\Lambda(y',y)$ :  
the cost of predicting  $y'$  when the true outcome is  $y$ 
  - for classification: cost matrix
  - when all mistakes are equally bad:
    - $\Lambda(y',y) = 0$       when  $y' = y$
    - $\Lambda(y',y) = 1$       otherwise
- Bayes risk of predicting  $y'$  for measurements  $x$  :

$$r(y' | x) = \int \Lambda(y', y) p(y | x) dy$$

*Risk of saying  $y' = \text{integral over all possible situations}$ :*

*Remember total probability:*  $P(r) = \text{SUM\_all\_}y \{ P(r|y)p(y) \}$   
 $= \text{SUM\_all\_}y \{ P(r \text{ and } y) \}$

# Bayesian estimation (2)

- Optimal prediction:

$$\begin{aligned}\hat{y} &= \arg \min_{y'} r(y' | x) \\ &= \arg \min_{y'} \int \Lambda(y', y) p(y | x) dy\end{aligned}$$

- Bayesian estimation: minimize overall risk

$$r^* = E[r(\hat{y} | x)] = \int r(\hat{y} | x) p(x) dx$$

*Best prediction is the one that minimizes the risk  
Best system minimizes expected risk: over all possible x's*

# Bayesian estimation (3)

- Example: diagnostic system

- $\omega = \{ h, d \}$  (healthy, diseased)

		$\omega = h$	$\omega = d$
$\omega' = h$	0	1	
	1	0	

- $\Lambda(\omega', \omega) = \omega' = h$

$$\omega = h \quad \omega = d$$

$$p(\omega = h|x) = 0.95 \\ (\text{by definition})$$

- Say the system predicts  $p(\omega = d|x) = 0.05$ , then

$$r(\omega' = h|x) = \Lambda(h,h) p(\omega = h|x) + \Lambda(h,d) p(\omega = d|x) \\ = p(\omega = d|x) = 0.05$$

$$r(\omega' = d|x) = \Lambda(d,h) p(\omega = h|x) + \Lambda(d,d) p(\omega = d|x) \\ = p(\omega = h|x) = 0.95$$

- Choose minimum risk, thus assign to h (in agreement with what you would expect by  $p(\omega = d|x) = 0.05$ )

# Bayesian estimation (4)

- Example: diagnostic system
  - $\omega = \{ h, d \}$  (healthy, diseased)

	$\omega = h$	$\omega = d$
$\omega' = h$	0	25
$\omega' = d$	1	0

- Say the system predicts  $p(\omega = d|x) = 0.05$ , then

$$\begin{aligned} r(\omega' = h|x) &= \Lambda(h,h) p(\omega=h|x) + \Lambda(h,d) p(\omega=d|x) \\ &= 0 \cdot 0.95 + 25 \cdot 0.05 = 1.25 \end{aligned}$$

$$\begin{aligned} r(\omega' = d|x) &= \Lambda(d,h) p(\omega=h|x) + \Lambda(d,d) p(\omega=d|x) \\ &= 1 \cdot 0.95 + 0 \cdot 0.05 = 0.95 \end{aligned}$$

• *Realize that minimum risk now says to assign to d !*

# Bayesian estimation (5)

- Cost function can have large influence on optimal decision!
- Think about:
  - Fingerprint identification (e.g. in database)
    - Cost of identifying incorrect person
  - Fingerprint verification (e.g. mobile phone)
    - Cost of incorrectly rejecting fingerprint owner
    - Cost of incorrectly allowing imposter
- Cost can often be quantified, e.g. cost of additional human intervention

# Scenarios: priors and risks

- Gene expression-based data classification
  - for artefact detection
  - for generating biological hypotheses
  - for validating biological hypotheses
  - for diagnosis of the common flu
  - for diagnosis of a form of cancer
- Protein-protein interaction prediction
  - for protein complex prediction
  - for discovering signaling pathways
  - for suggesting *in vitro* experiments
  - for suggesting *in vivo* experiments
  - for drug target analysis

# Recapitulation

- *Machine learning* is concerned with the construction of approximate, generalizing models by learning from examples
- The *machine learning pipeline* consists of defining objects and measurements, constructing a predictive function and applying it to unseen data
- *Bayes' theorem* plays a central role in statistical machine learning
- *Bayesian estimation*
  - provides a framework for minimizing cost due to errors
  - combines class-conditional and prior distributions into posterior ones

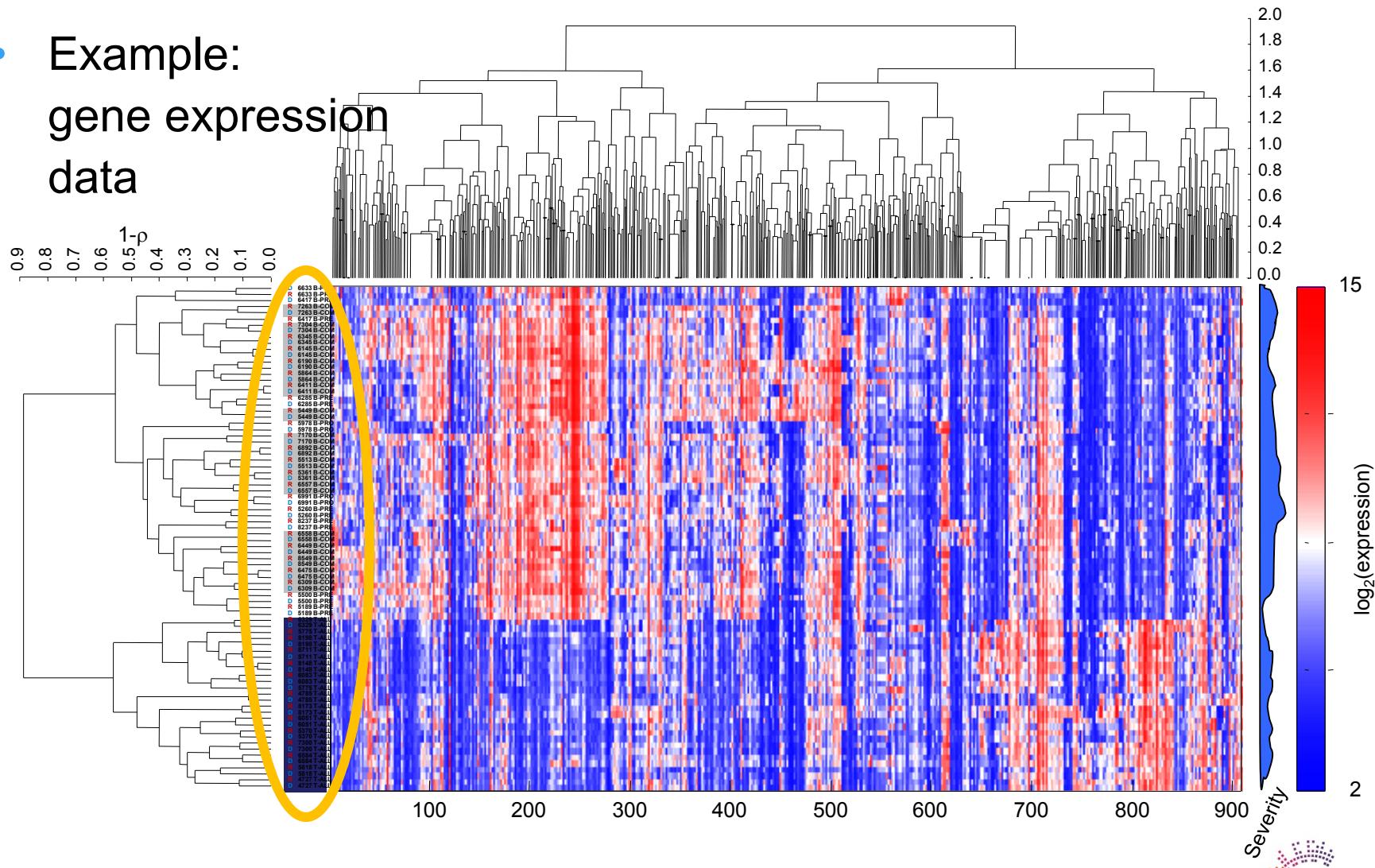


10min break

# Bayesian classification

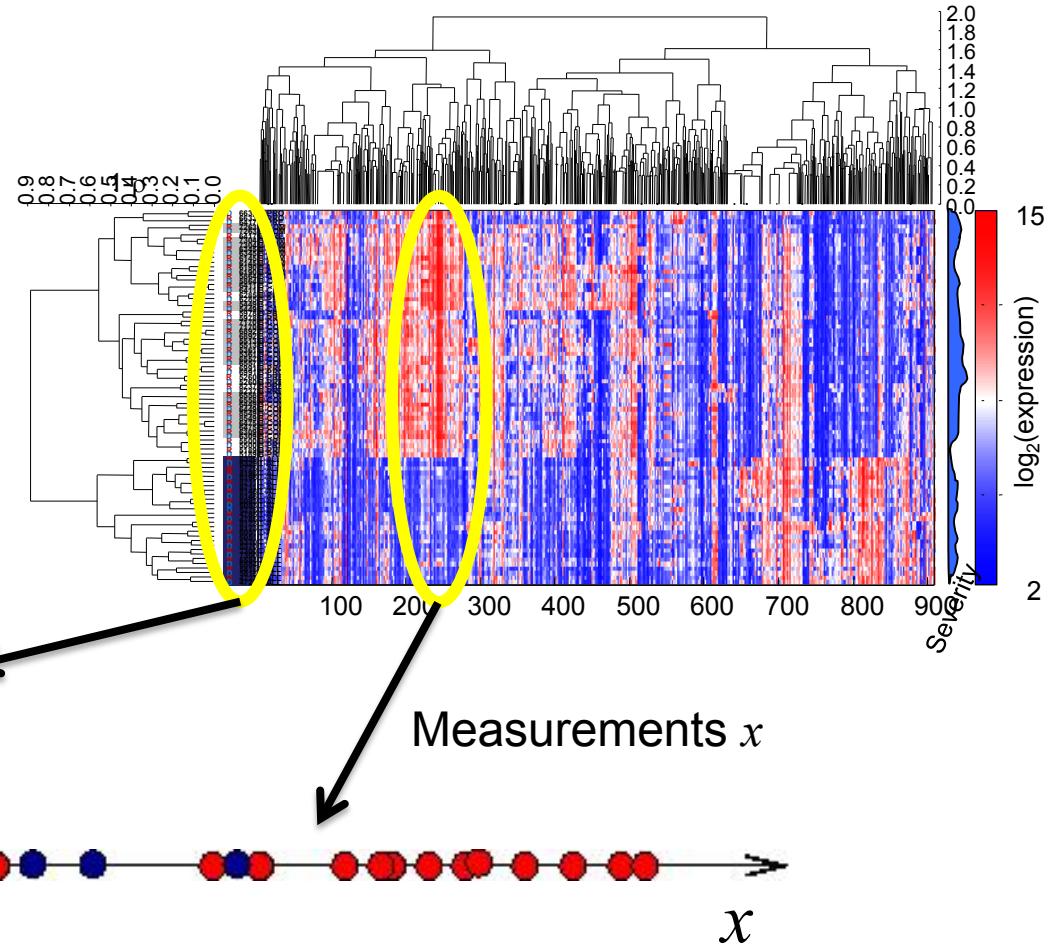
# Classification in bioinformatics

- Example:  
gene expression  
data



- But again: theory applies to any type of data!

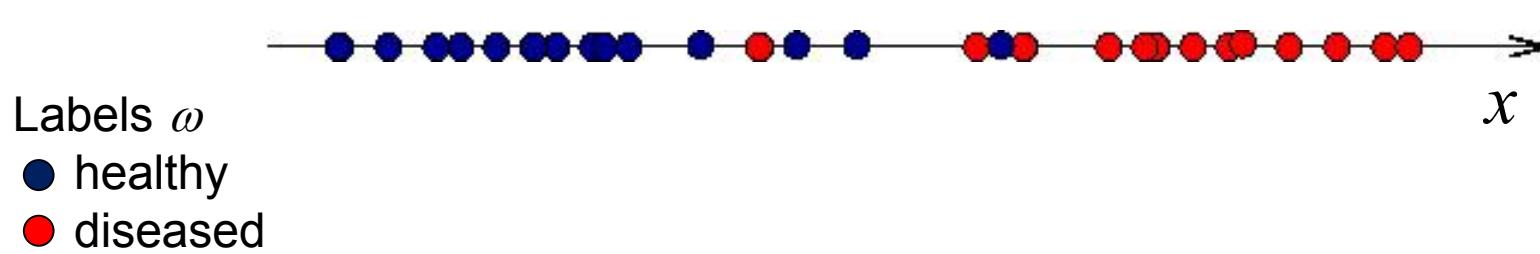
# Classification



As example, consider a single gene expression measurement  $x$

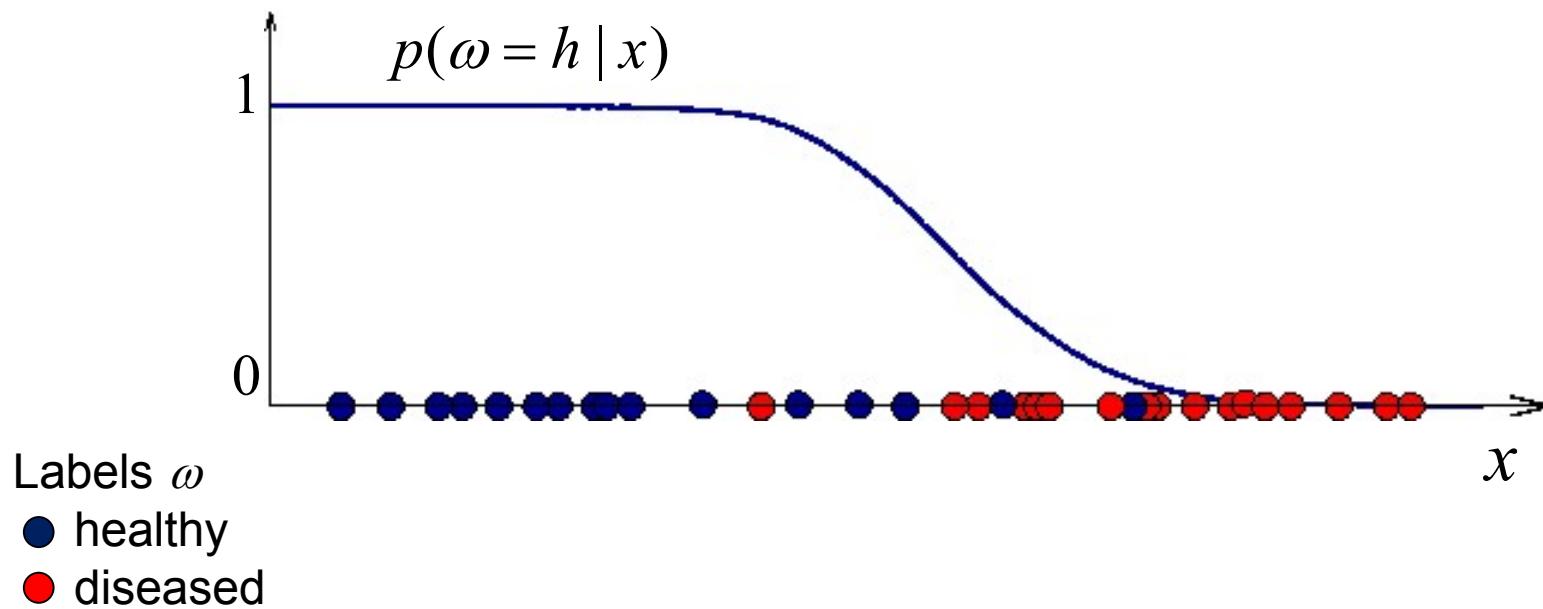
# Posterior probability

- For each object, we have to estimate  $p(\omega|x)$  or  $p(y|x)$



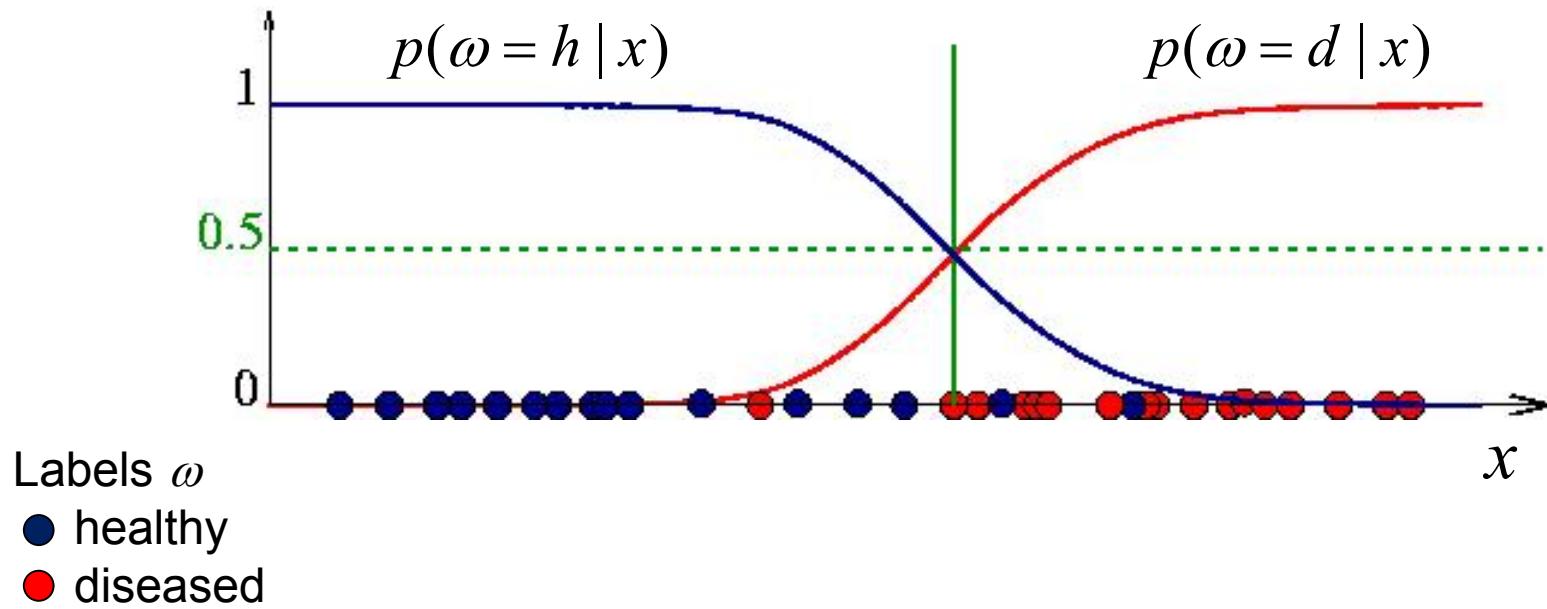
# Posterior probability (2)

- For each object, we have to estimate  $p(\omega|x)$  or  $p(y|x)$



# Posterior probability (2)

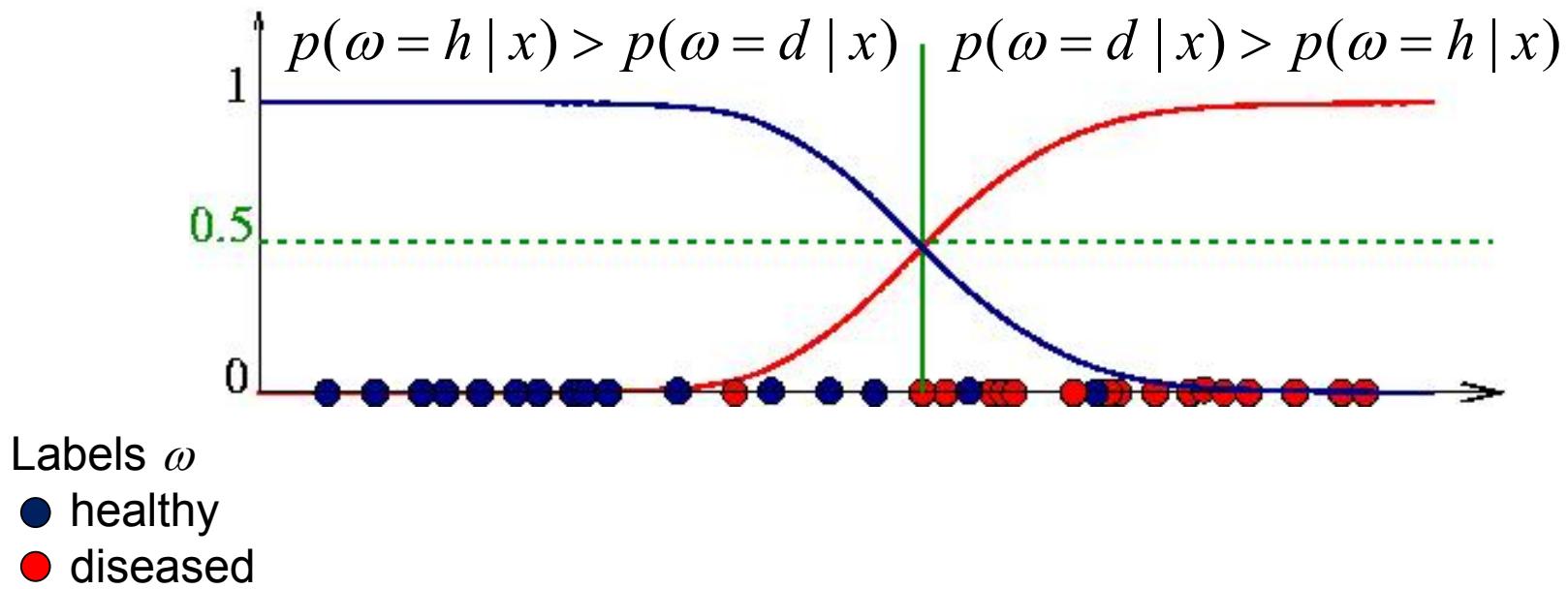
- For each object, we have to estimate  $p(\omega|x)$  or  $p(y|x)$



- Of course:  $\sum_{c=1}^C p(\omega=c | x) = 1$

# Posterior probability (3)

- For each object, we have to estimate  $p(\omega|x)$  or  $p(y|x)$



# A classifier

- There are several ways to describe a classifier:
  - if  $p(\omega = h | x) > p(\omega = d | x)$  then assign to  $h$   
otherwise to  $d$
  - if  $p(\omega = h | x) - p(\omega = d | x) \geq 0$  then assign to  $h$   
otherwise to  $d$
  - if  $\frac{p(\omega = h | x)}{p(\omega = d | x)} \geq 1$  then assign to  $h$   
otherwise to  $d$
  - if  $\ln[p(\omega = h | x)] - \ln[p(\omega = d | x)] \geq 0$  then assign to  $h$   
otherwise to  $d$
- A Bayesian classifier is a *threshold* on the difference between *posterior probabilities*

# Bayes' rule

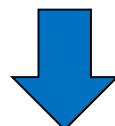
- In many cases, the posterior is hard to estimate
- Often a certain functional form can be assumed for the *class-conditional distributions*
- Use Bayes' theorem to rewrite one into the other:

- posterior distribution: 
$$p(\omega = c | x) = \frac{p(x | \omega = c)p(\omega = c)}{p(x)}$$
- class-conditional distribution:  $p(x | \omega = c)$
- prior distribution:  $p(\omega)$
- data distribution: 
$$p(x) = \sum_{c=1}^C p(x | \omega = c)p(\omega = c)$$

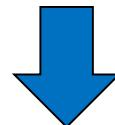
## Bayes' rule (2)

- The decision rule becomes

$$p(\omega = h \mid x) > p(\omega = d \mid x)$$



$$\frac{p(x \mid \omega = h)p(\omega = h)}{p(x)} > \frac{p(x \mid \omega = d)p(\omega = d)}{p(x)}$$



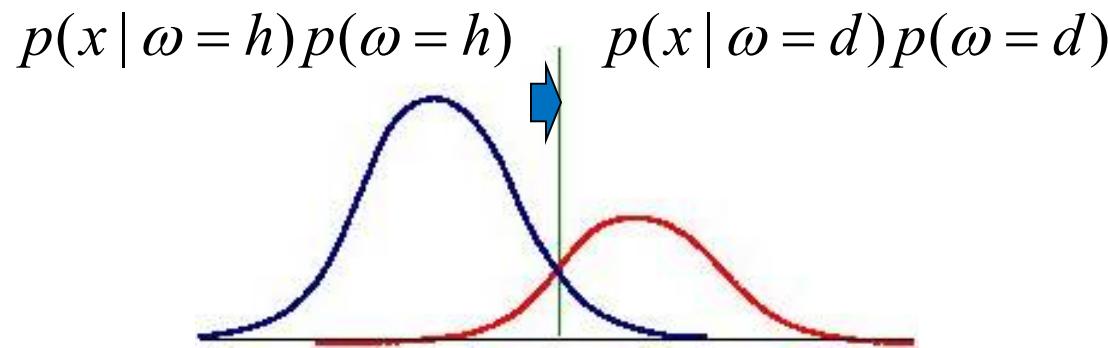
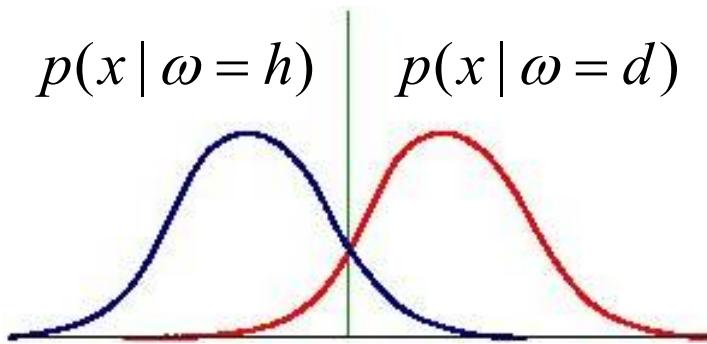
$$p(x \mid \omega = h)p(\omega = h) > p(x \mid \omega = d)p(\omega = d)$$

***Seems trivial, but this is something we can measure!***



## Bayes' rule (3)

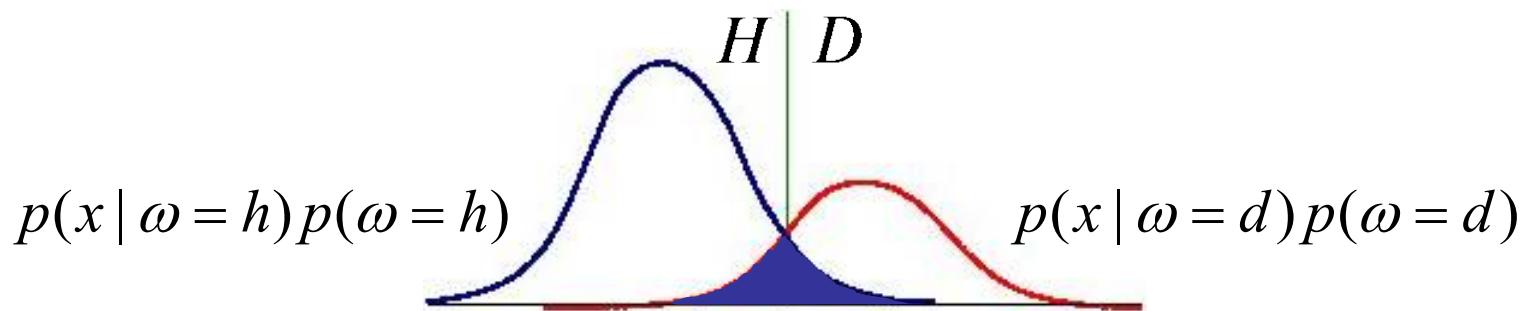
- The effect of the prior:



Prior can shift the decision boundary (as can risk, recall the h/d example)  
If one class is very unlikely, we will not make a large error if we misclassify that class

# Bayes' rule (4)

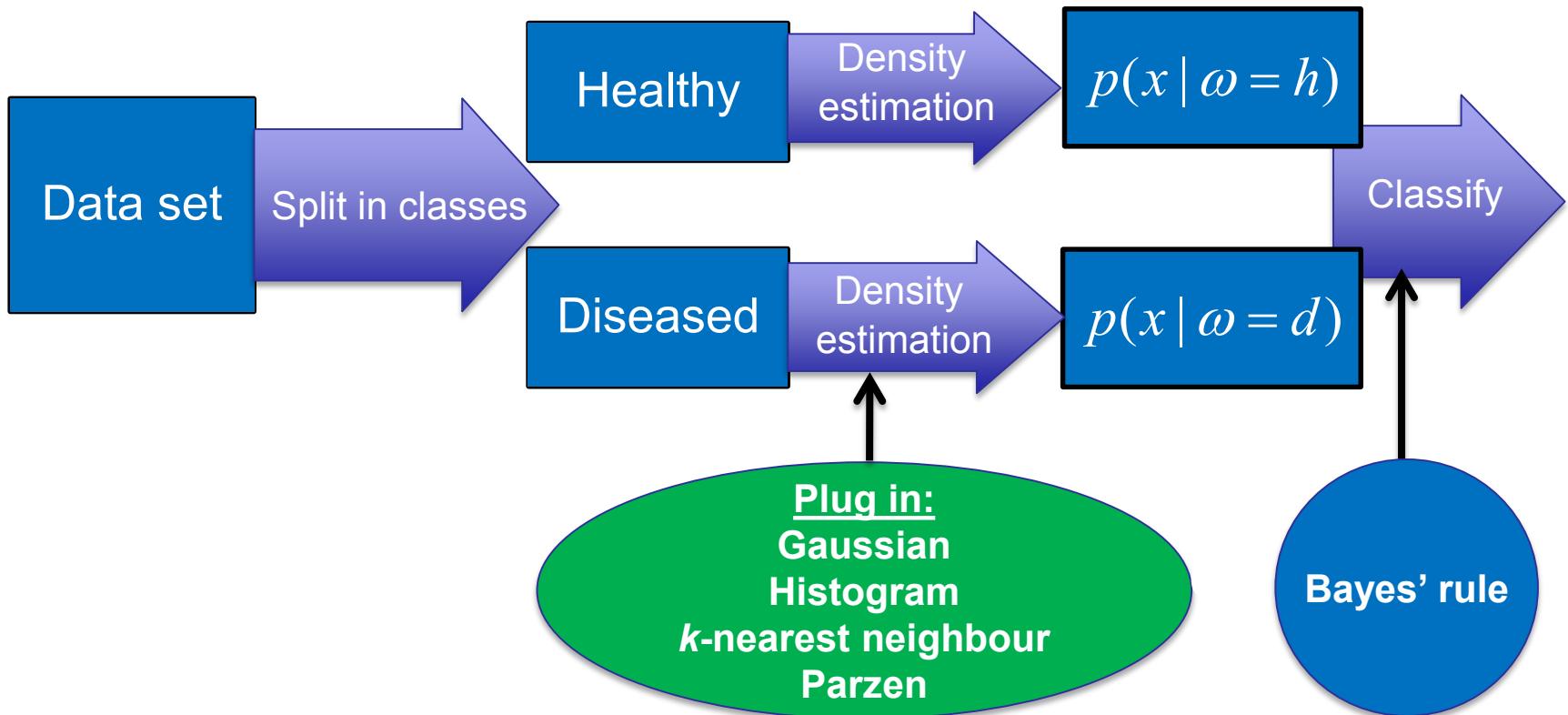
- Bayes' error: ***minimal attainable error***  
(if data follows class-conditional contributions...)



- $\Lambda(\omega', \omega) = 0$       when  $\omega' = \omega$
- $\Lambda(\omega', \omega) = 1$       otherwise

# Bayes' rule (5)

- In practice:



# Density estimation

# Plug-in Bayes classifier

- Bayes' rule:

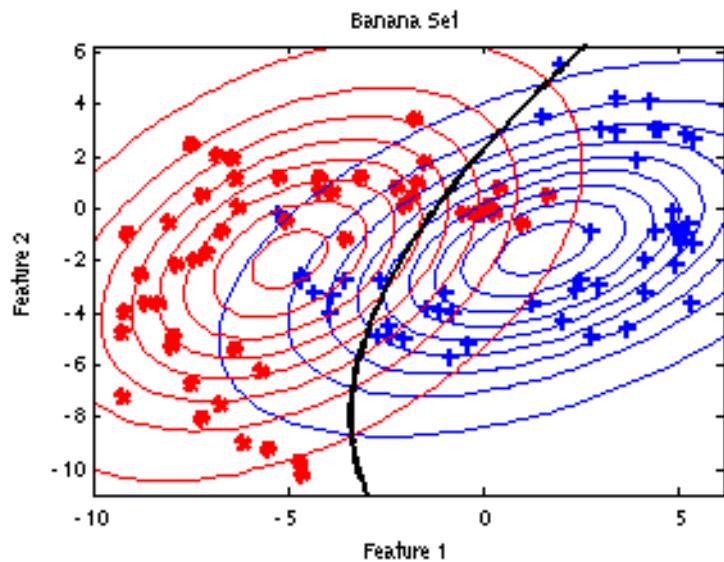
$$c_{opt} = \arg \max_c p(\omega = c | x) = \arg \max_c p(x | \omega = c)p(\omega = c)$$

- Given priors, we only require the class conditional distributions  $p(x|\omega=c)$
- In practice we will always have to *estimate*  $p(x|\omega=c)$  by  $\hat{p}(x | \omega = c)$  and hope that the classifier resulting when we *plug in* this approximation will still perform well
- Density estimation is a very hard problem!
- The resulting classifier will be *sub-optimal* and in general will *not* attain Bayes' error

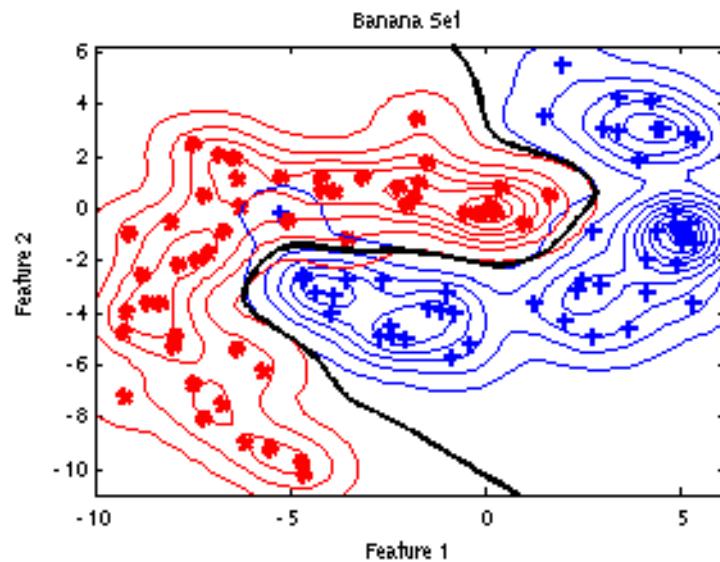
# Plug-in Bayes classifier (2)

- Same problem, two different density estimates  $\hat{p}(x | \omega = c)$

Normal density estimation



Parzen density estimation



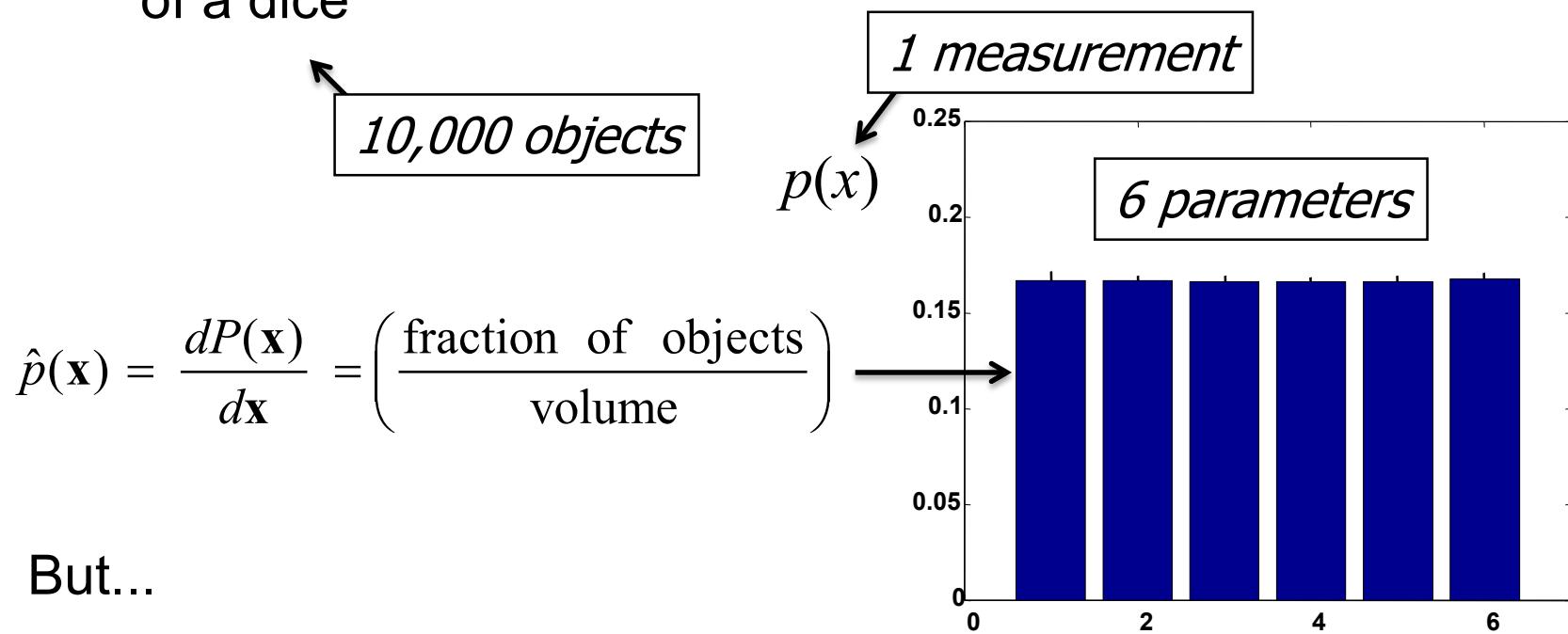
Which one is best (Parzen)

Which one is optimal (none: true dist = normal perpendicular to two half-circles) SB

# Density estimation

- Simplest approach: approximate density by histogram

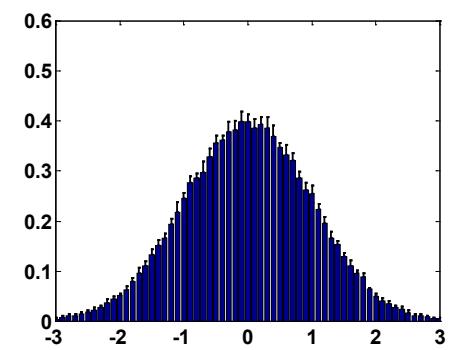
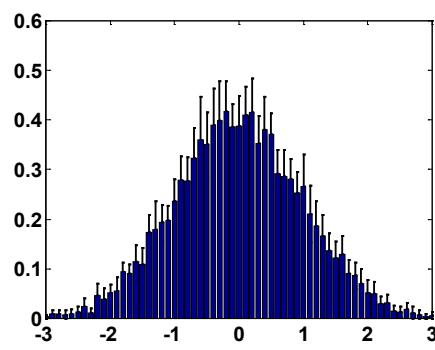
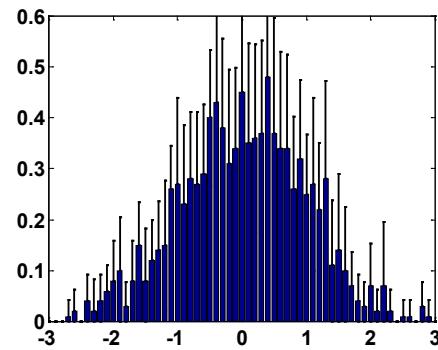
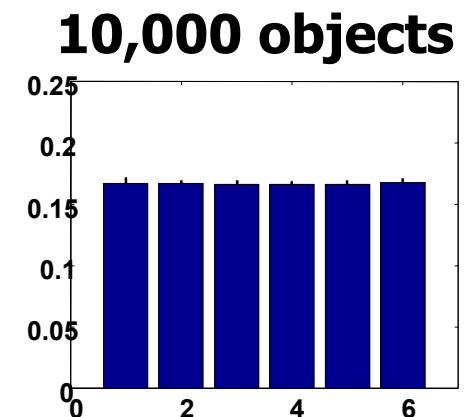
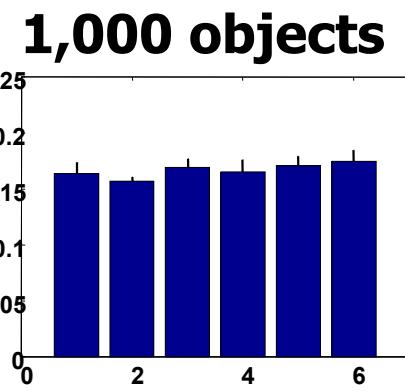
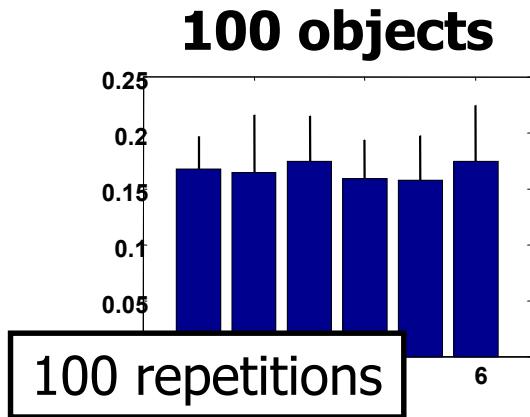
e.g. 10,000 throws  
of a dice



- But...

# Density estimation (2)

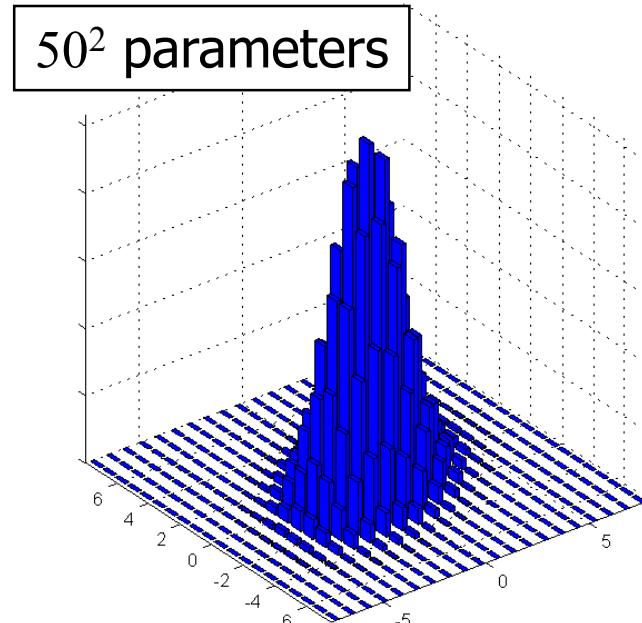
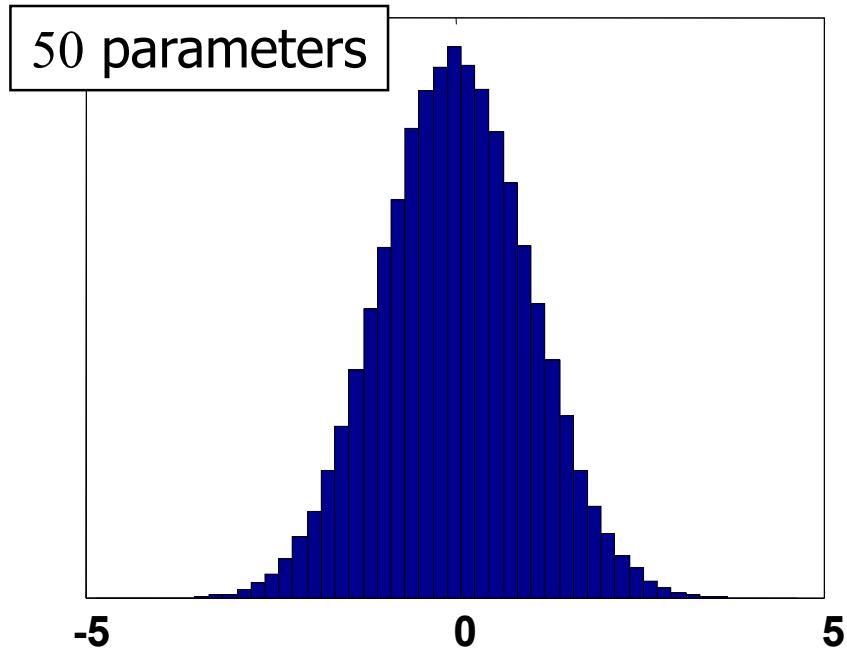
- Problem: accuracy



Gauss: 50 bin  $\rightarrow$  50 parameters to estimate

# Density estimation (3)

- For 1 - dimensional data,  
 $\pm 1000$  points needed
- For  $p$  - dimensional data,  
 $\pm 1000^p$  points needed

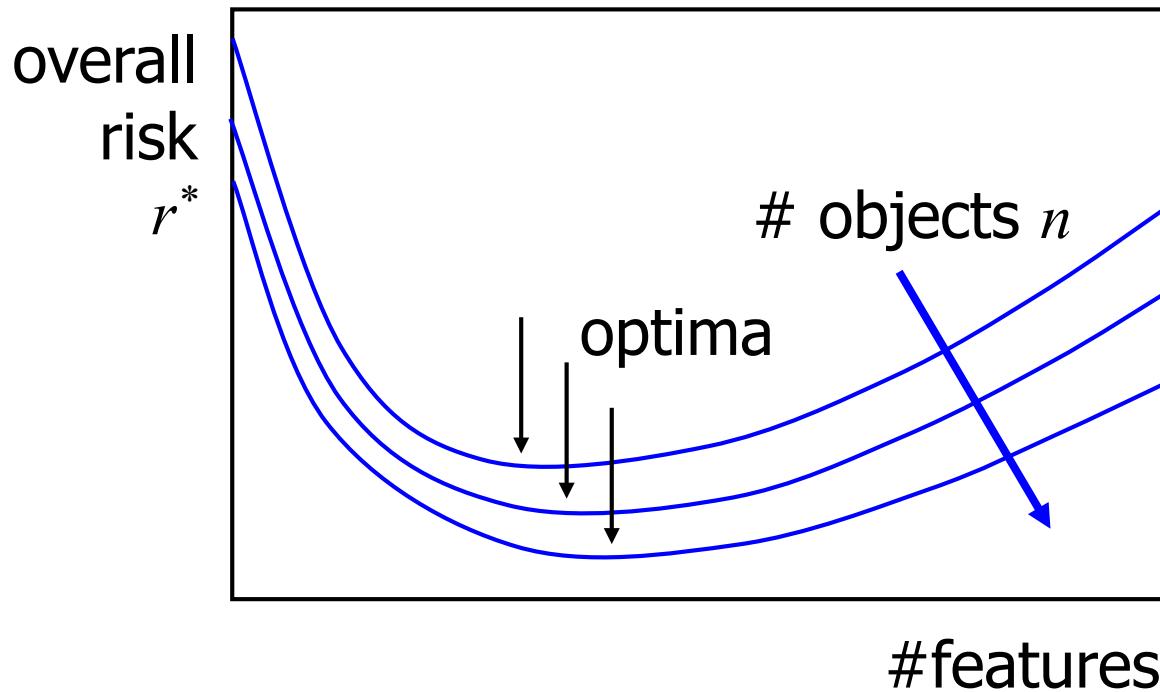


- Unworkable for  $p > 2$  measurements

# Curse of dimensionality

- Intuitively, using more measurements (e.g. width, height, color etc.) should give us more information about the outcome to predict
- But we never know the densities, so we have to estimate them
- The number of parameters (e.g. histogram bins) to estimate increases with the number of measurements
- To estimate these well, you need more objects
- Consequence:  
***there is an optimal number of measurements to use***

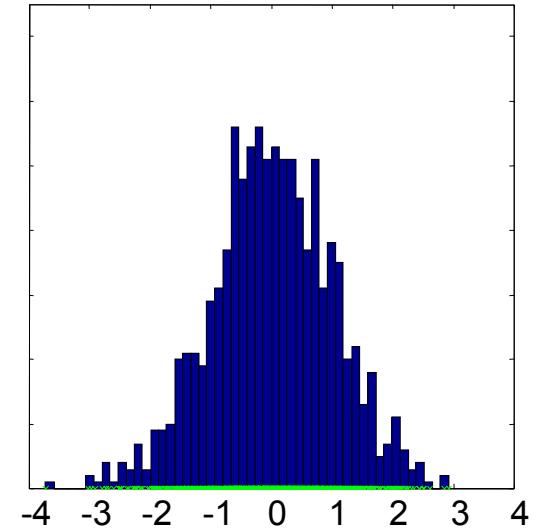
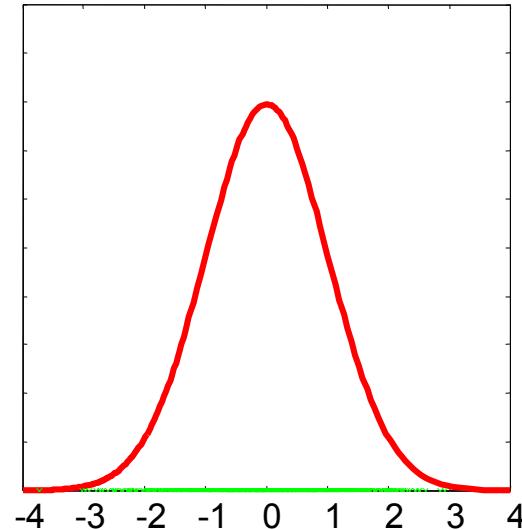
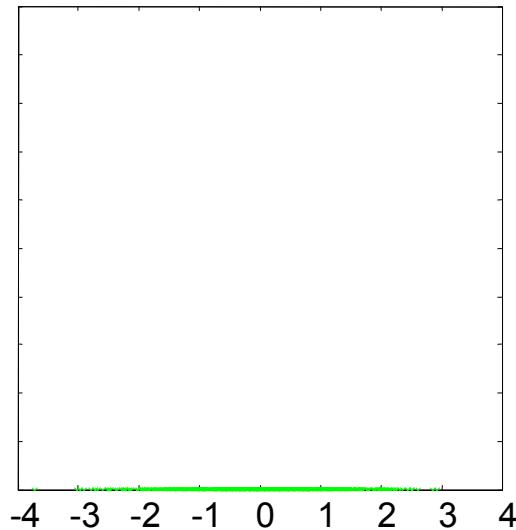
# Curse of dimensionality (2)



So, realize if  $n \rightarrow \text{INF}$  than you can have many features

# Density estimation (4)

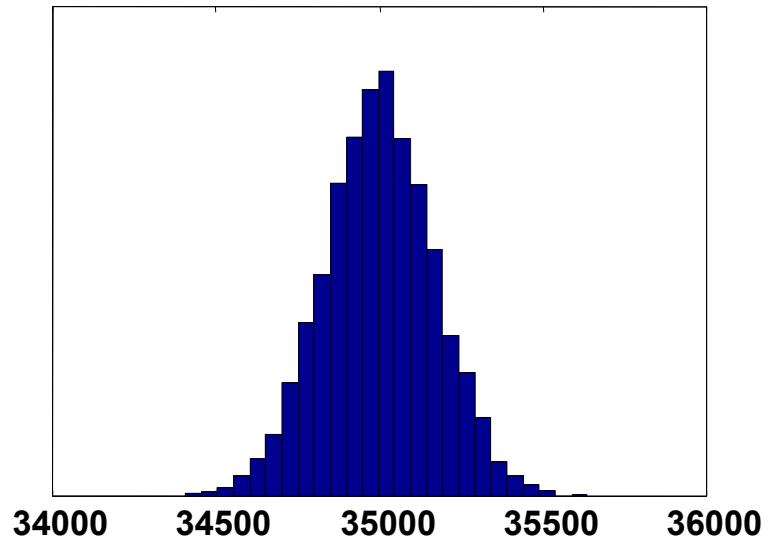
- Two main approaches:
  - *parametric*: assume simple *global* model, e.g. Gaussian, and estimate its parameters
  - *non-parametric*: assume simple *local* model, e.g. uniform, Gaussian, and aggregate



# The Gaussian distribution

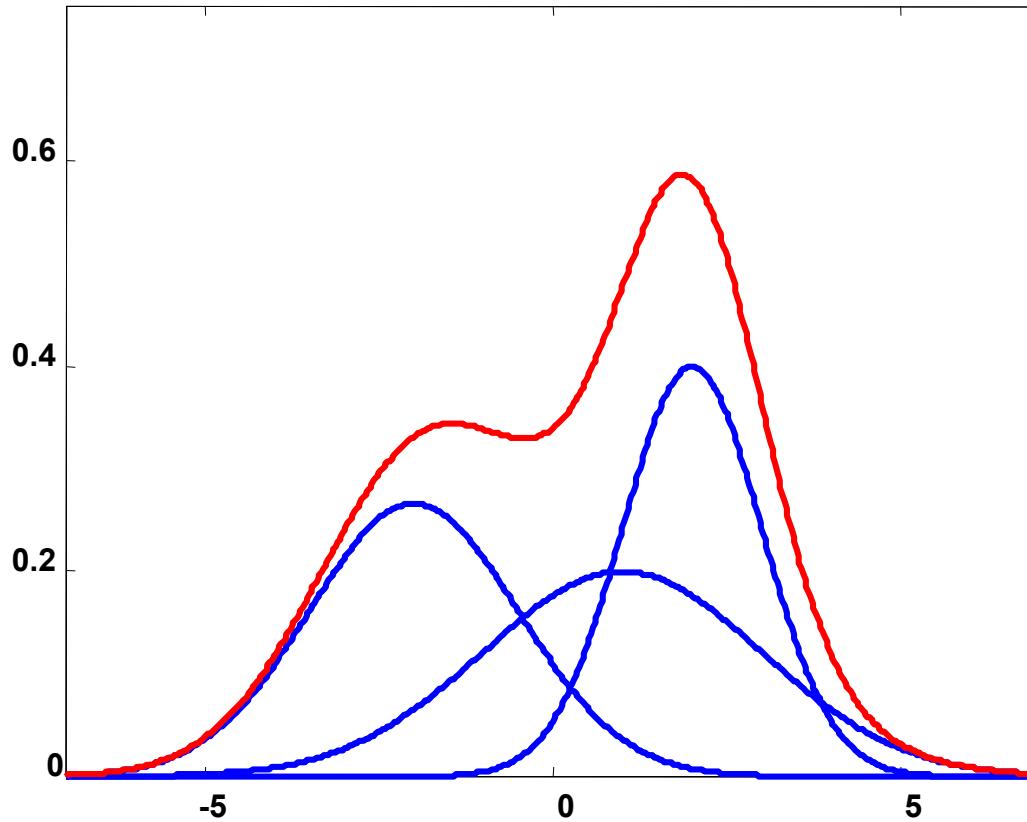
- Why Gaussians?
  - Special distribution: the Central Limit Theorem says that sums of large numbers of i.i.d. (independent, identically distributed) random variables will have a Gaussian distribution
  - Simple, few parameters
  - Often occurs in real life

e.g. sum of eyes of  
10,000 dice throws  
(expectation = 3.5 per throw)



# The Gaussian distribution (2)

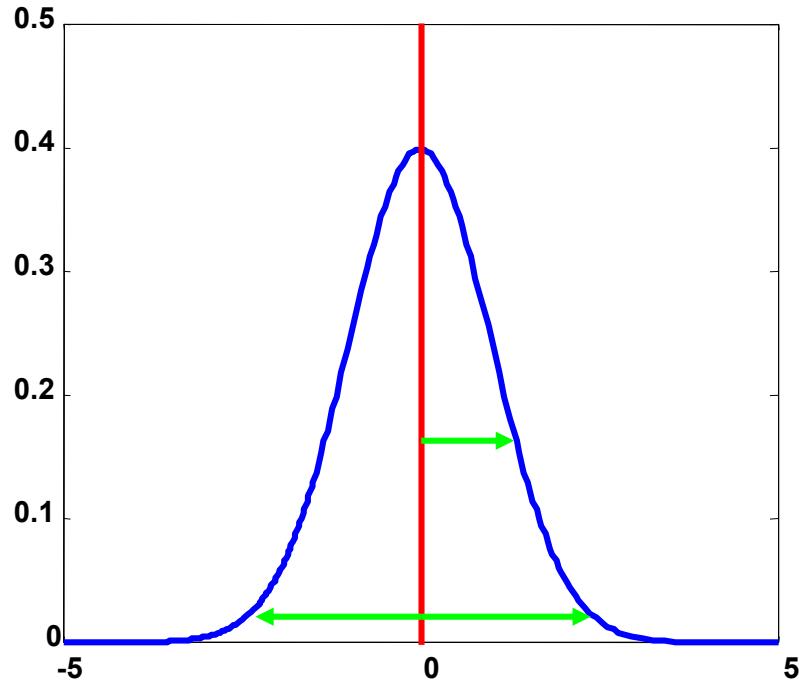
- Not necessarily too restrictive: mixture models (discussed later)



— Gaussian

— Mixture of Gaussians

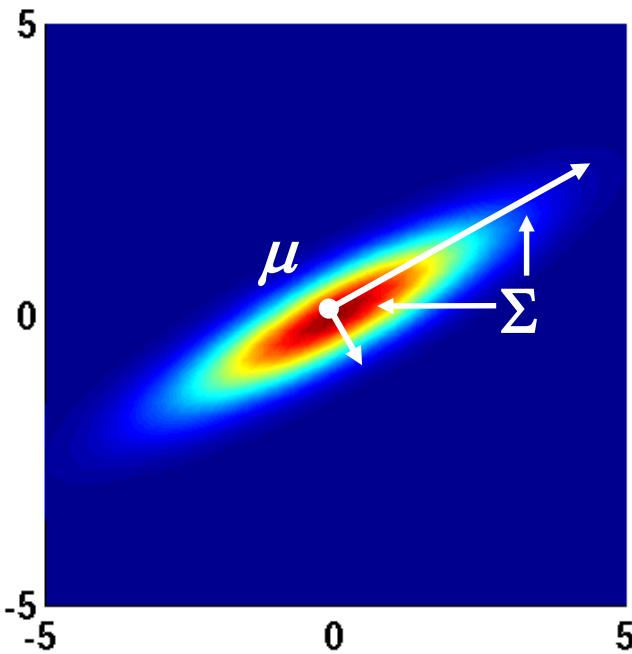
# The Gaussian distribution (3)



- 1-dimensional density:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2} \frac{(x-\mu)^2}{\sigma^2}\right)$$

# Multivariate Gaussian distribution

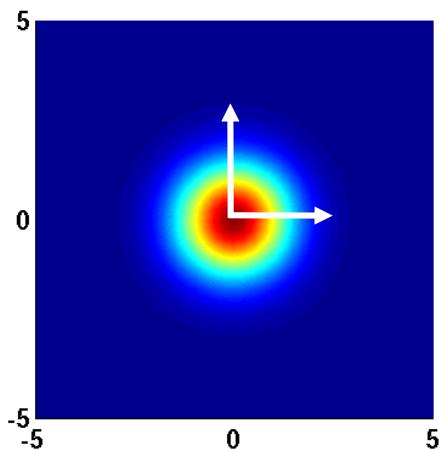


$$\Sigma = \begin{bmatrix} 3 & 1\frac{1}{2} \\ 1\frac{1}{2} & 2 \end{bmatrix}$$

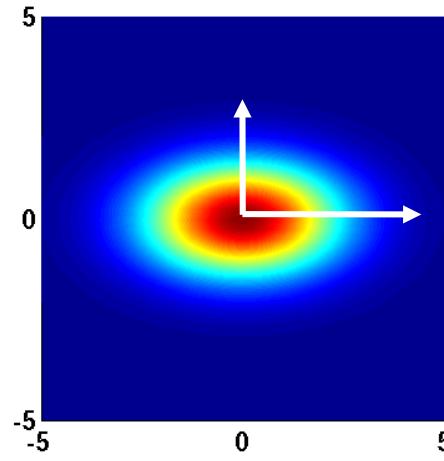
- $p$  - dimensional density:

$$p(\mathbf{x}) = \frac{1}{\sqrt{2\pi^p \det(\Sigma)}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})\right)$$

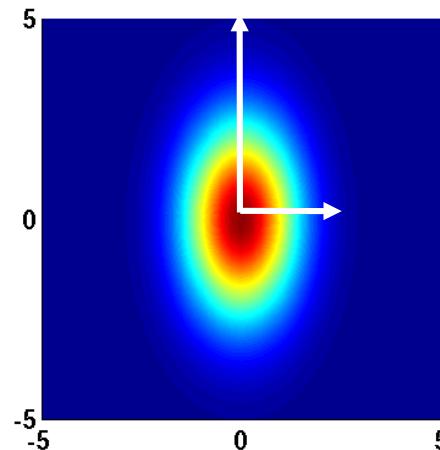
# Multivariate Gaussian distribution (2)



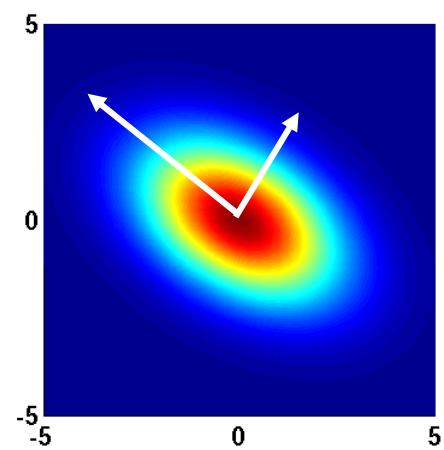
$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



$$\Sigma = \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix}$$



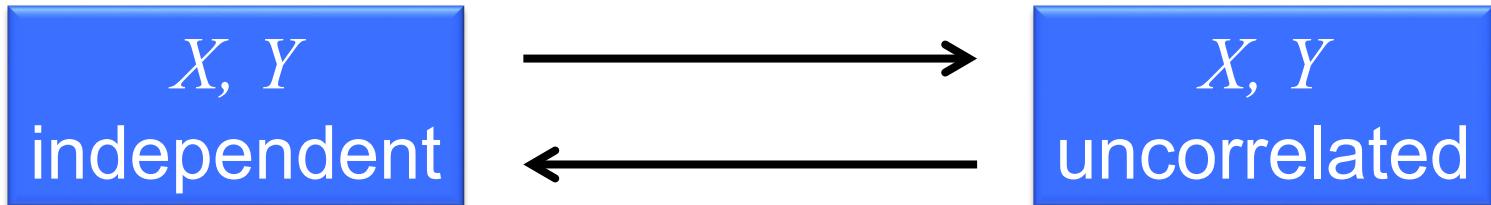
$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix}$$



$$\Sigma = \begin{bmatrix} 3 & -1 \\ -1 & 1 \end{bmatrix}$$

# Special properties

- The Gaussian distribution is a special case:

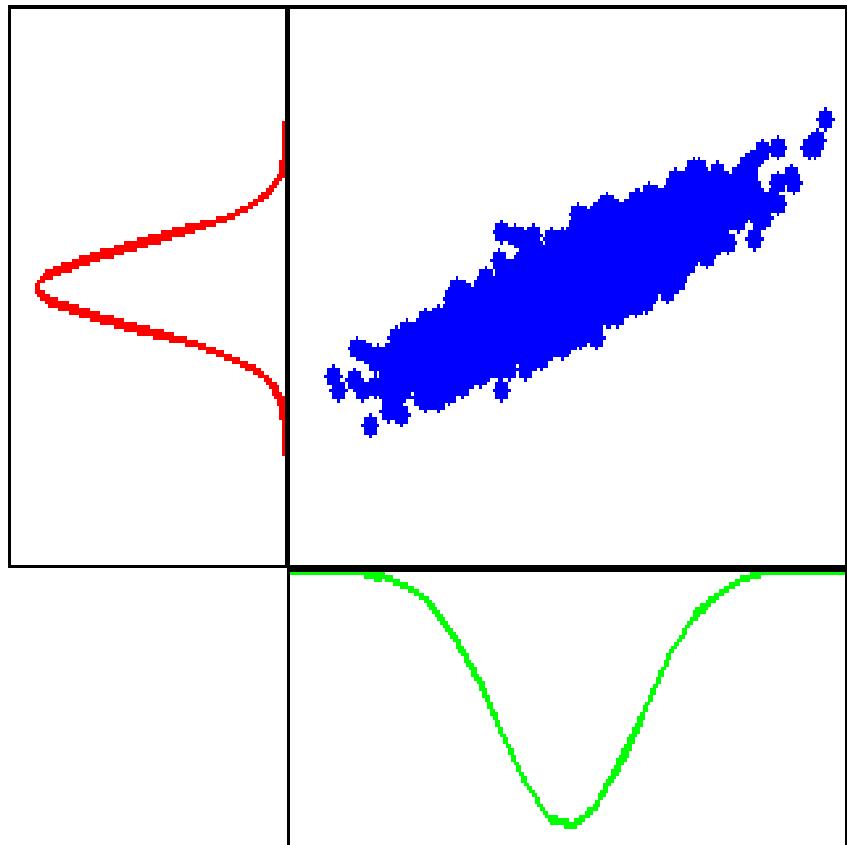


- Proof: if uncorrelated,  $\Sigma$  is diagonal ( $\sigma_1 \dots \sigma_p$ )

$$\begin{aligned} p(\mathbf{x}) &= \frac{1}{\sqrt{2\pi^p \det(\Sigma)}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right) \\ &= \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left(-\frac{1}{2}(x_1 - \mu_1)^\top \sigma_1^{-2}(x_1 - \mu_1)\right) \times \frac{1}{\sqrt{2\pi\sigma_2^2}} \exp\left(-\frac{1}{2}(x_2 - \mu_2)^\top \sigma_2^{-2}(x_2 - \mu_2)\right) \\ &\quad \times \dots \times \frac{1}{\sqrt{2\pi\sigma_p^2}} \exp\left(-\frac{1}{2}(x_p - \mu_p)^\top \sigma_p^{-2}(x_p - \mu_p)\right) = p(x_1)p(x_2)\dots p(x_p) \end{aligned}$$

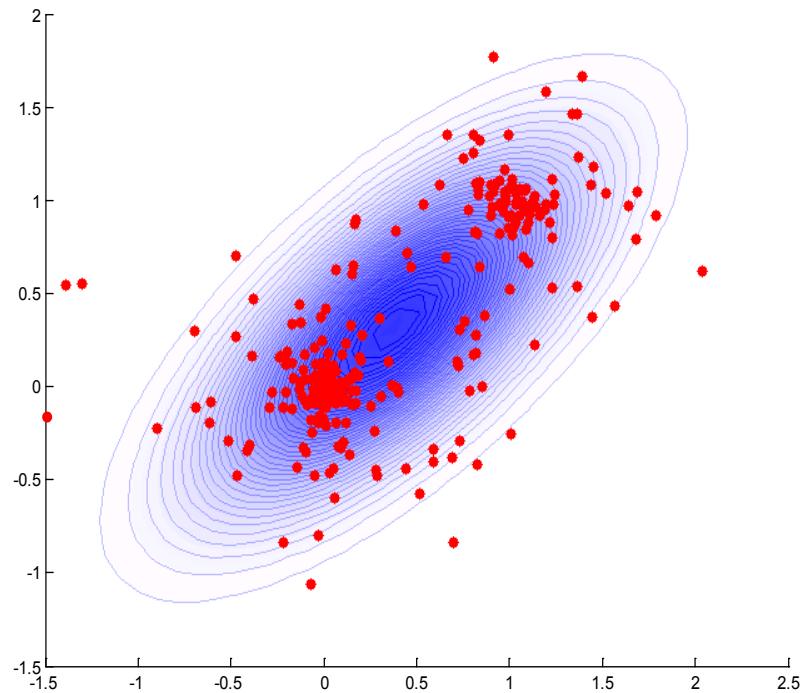
# Special properties (2)

- Any projection of a high-dimensional Gaussian is itself again Gaussian



# Parametric estimation

- Assume model, e.g. Gaussian and estimate mean  $\mu$  and covariance  $\Sigma$  from data
- Sounds simple, but for  $p$  - dimensional data set:
  - $\mu$  : vector with  $p$  elements
  - $\Sigma$  : matrix with  $0.5 p(p+1)$  elements
- Number of parameters increases quadratically with  $p$  : need a *lot* of data for high-dimensional problems

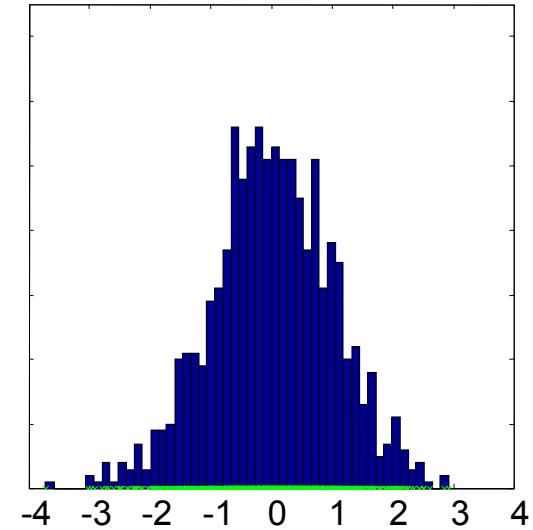
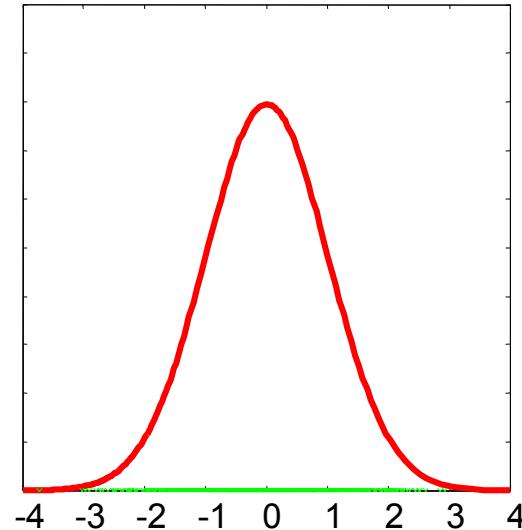
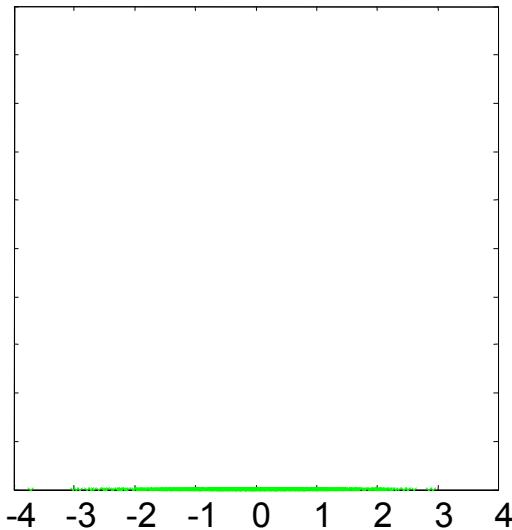




10min break

# Density estimation (4)

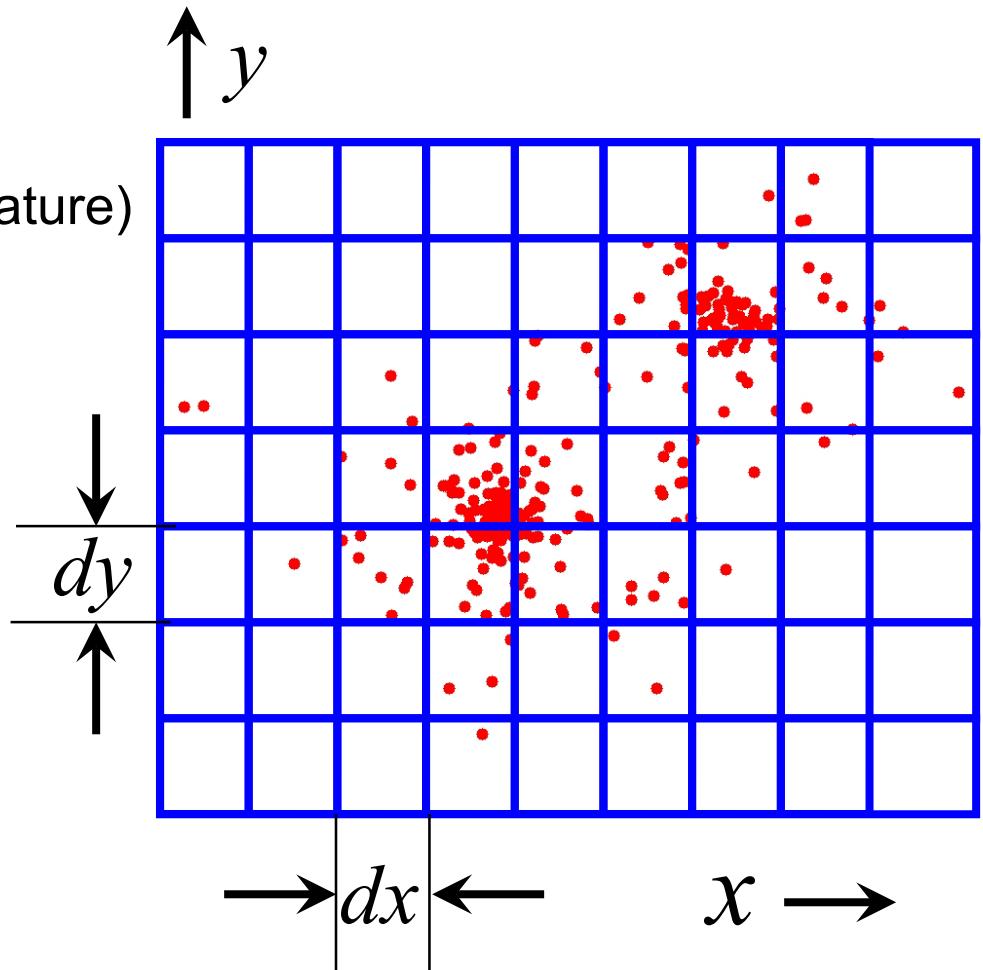
- Two main approaches:
  - *parametric*: assume simple *global* model,  
e.g. Gaussian, and estimate its parameters
  - *non-parametric*: assume simple *local* model,  
e.g. uniform, Gaussian, and aggregate



# Histogramming

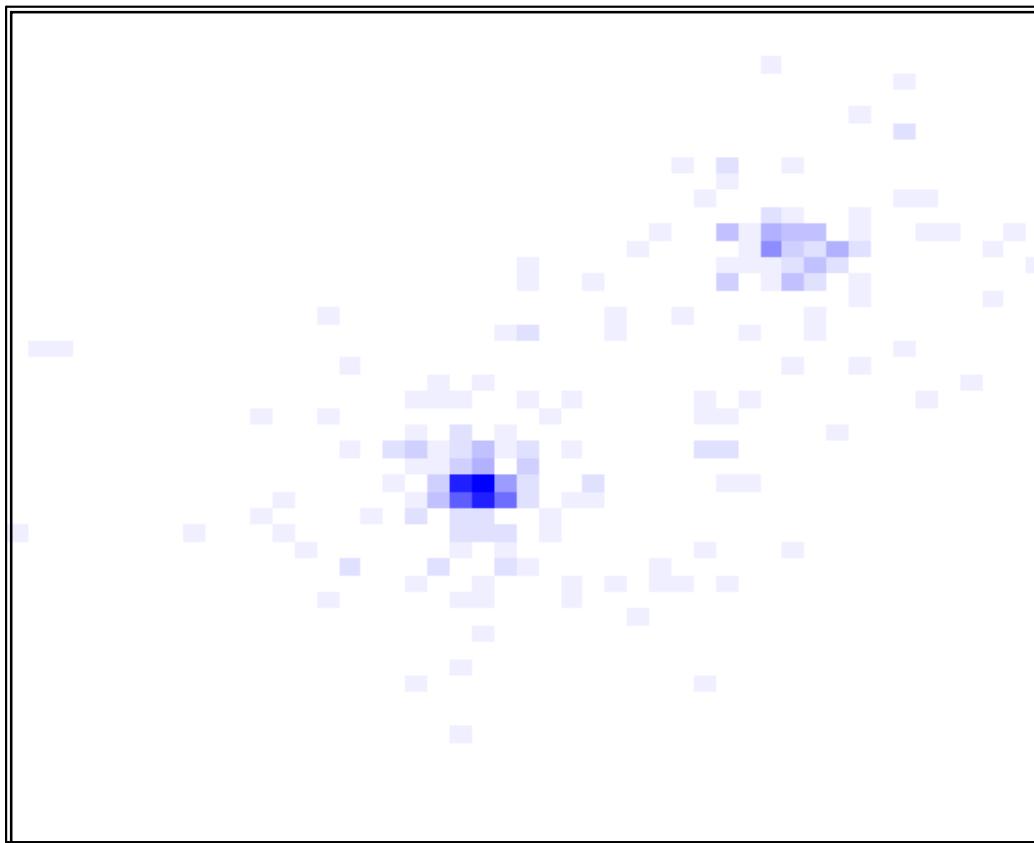
- Histogram method:
  - Divide feature space into  $N^p$  bins ( $N$  bins per feature)
  - Count number of objects in each bin
  - Normalize:

$$\hat{p}(\mathbf{x}) = \frac{n_i}{\sum_{i=1}^{N^p} n_i dx dy}$$



# Histogramming (2)

- For example, using  $N=50$  bins per dimension

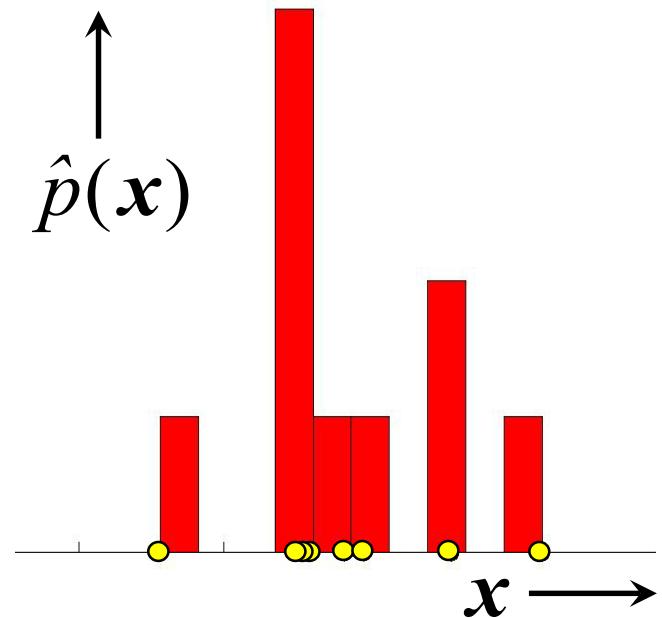


# Histogramming (3)

- Histogram density estimate:

$$\hat{p}(x | dx) = \left( \frac{\text{fraction of objects}}{\text{volume}} \right)$$

- Fix cell size ( $dx$ )
- Count #objects per cell

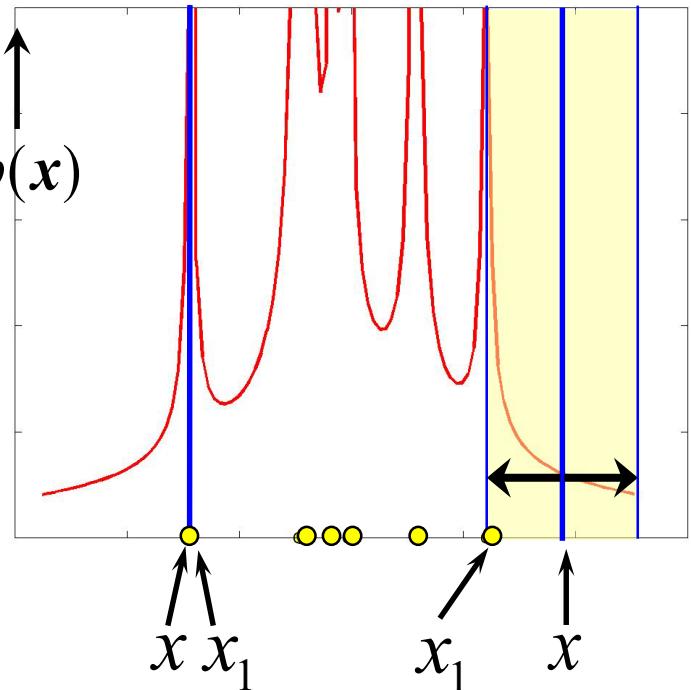


# k-nearest neighbor density estimation

- k-nearest neighbor estimate:

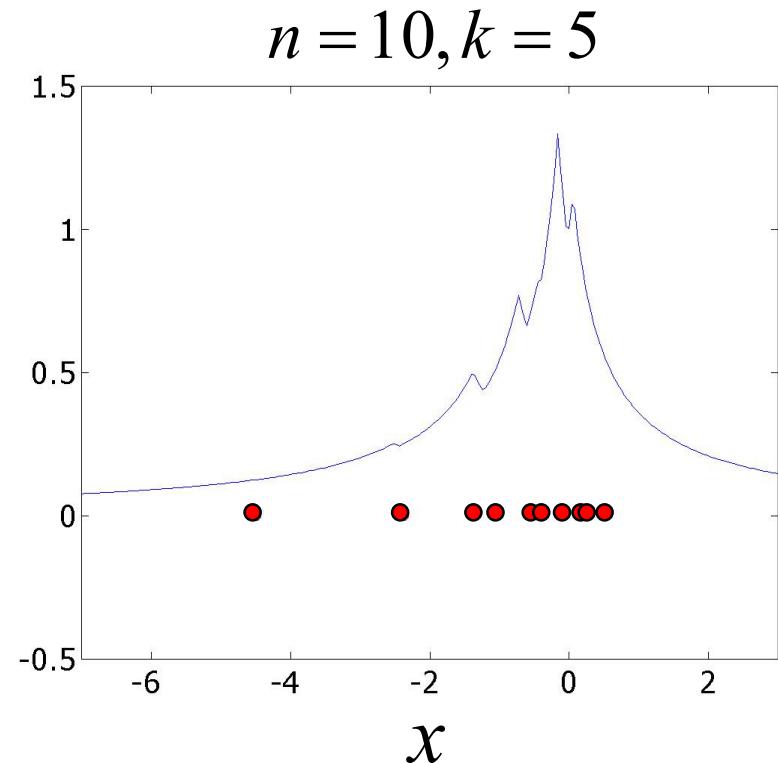
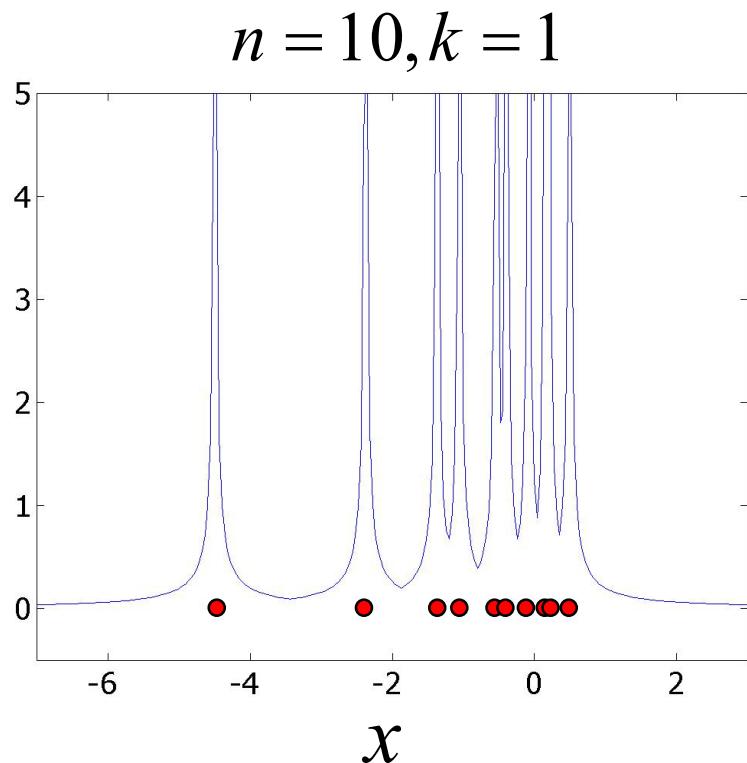
$$\hat{p}(x | k) = \left( \frac{\text{fraction of objects}}{\text{volume}} \right) \hat{p}(x)$$
$$= \frac{k}{n\Delta x_k} = \frac{k}{n \|x - x_k\|}$$

- Fix #objects per cell ( $k$ )
- Determine cell size (volume)



# **k-nearest neighbor density estimation (2)**

- The density estimate for  $k = 1$  contains singularities:



# Parzen density estimation

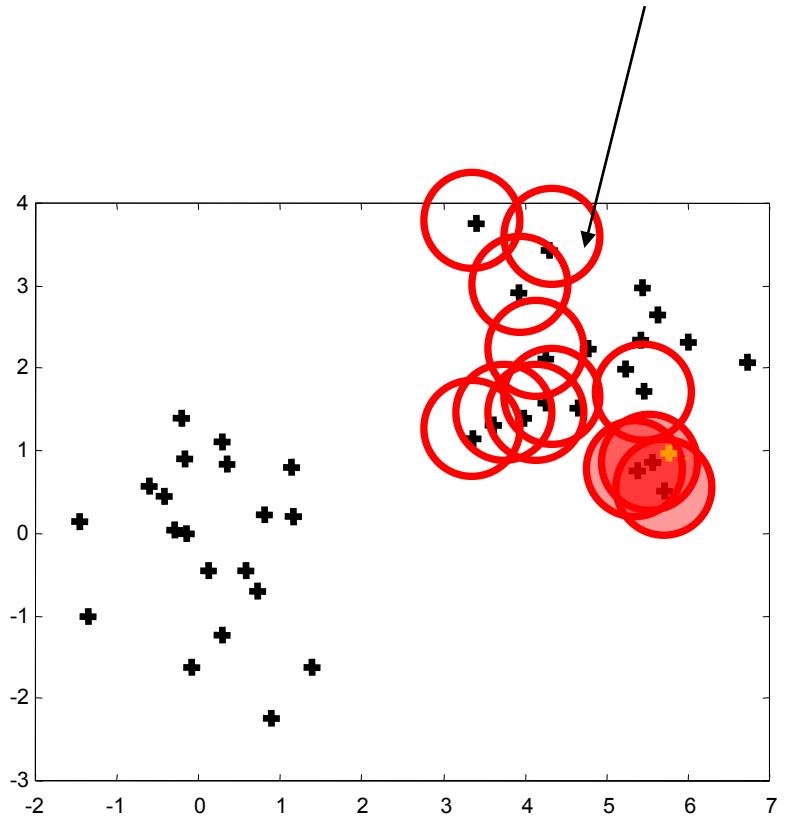
- Procedure:
  - Fix volume of cell
  - Vary positions of cells
  - Add contributions of cells
- Define cell shape (kernel),  
e.g. uniform

$$K(r, h) = \begin{cases} 0 & \text{if } |r| > h \\ \frac{1}{V} & \text{if } |r| \leq h \end{cases}$$

(with  $V$  the volume of the kernel)

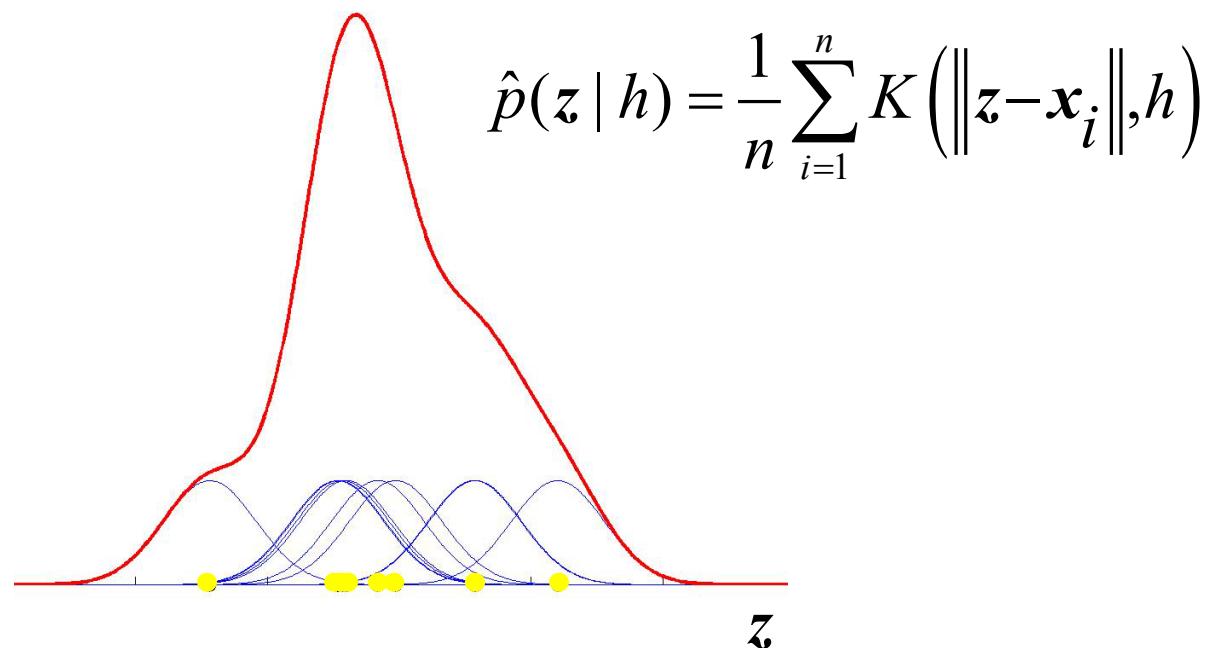
or Gaussian

- For test object  $z$ , sum all cells:  $\hat{p}(z | h) = \frac{1}{n} \sum_{i=1}^n K(\|z - x_i\|, h)$



# Parzen density estimation (2)

- With Gaussian kernel:  $K(r,h) = \frac{1}{2\pi^{1/2}h} \exp\left(-\frac{1}{2}\frac{r^2}{h^2}\right)$

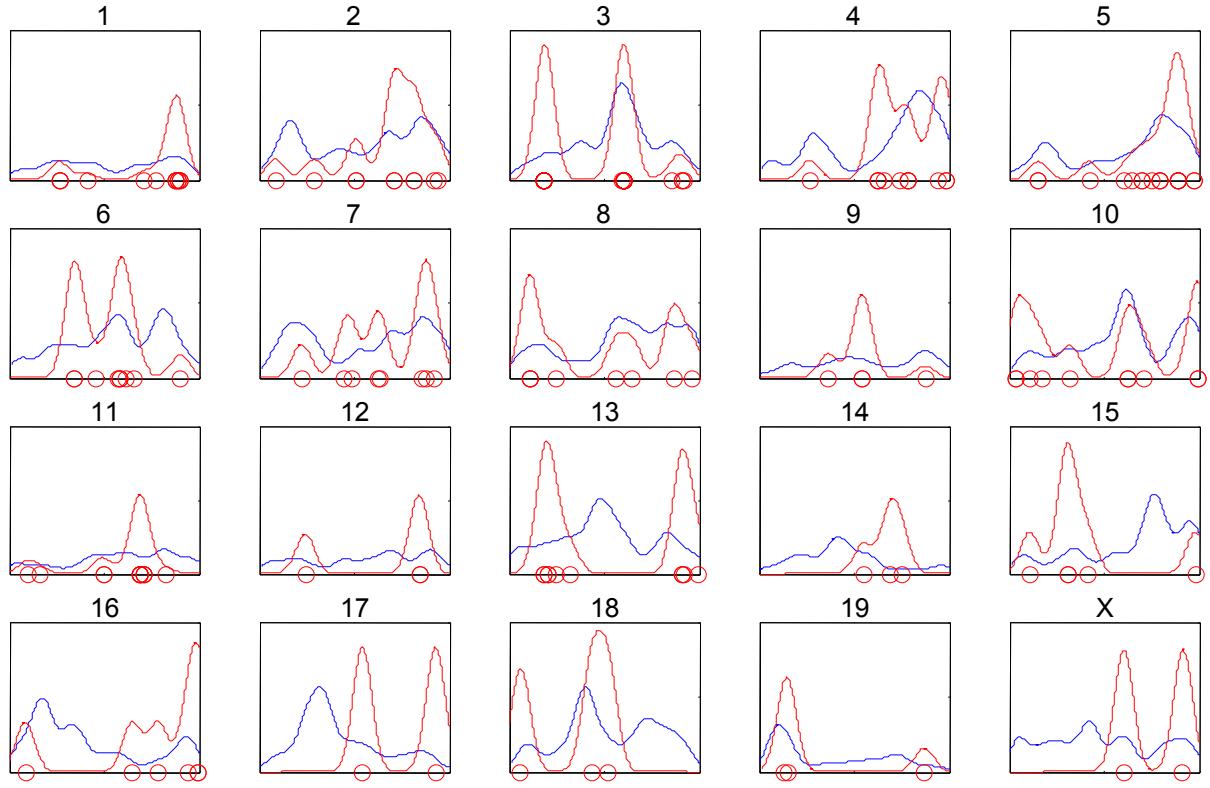


# Parzen density estimation (3)

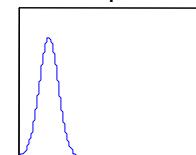
- Example: viral insertions in each chromosome

Density of highly expressed genes

Density of viral insertions



- Feature: position along chromosome



# Parzen density estimation (4)

- Maximum likelihood (ML) estimate: choose kernel width  $h$  such that the probability of the observed data is maximal
  - PDF of observing a point  $z$ :

$$\hat{p}(z | h) = \frac{1}{n} \sum_{i=1}^n K\left(\|z - x_i\|, h\right)$$

- PDF of observing dataset  $x_1, \dots, x_n$  (likelihood):

$$\hat{p}(X|h) = \prod_{i=1}^n \hat{p}(x_i | h)$$

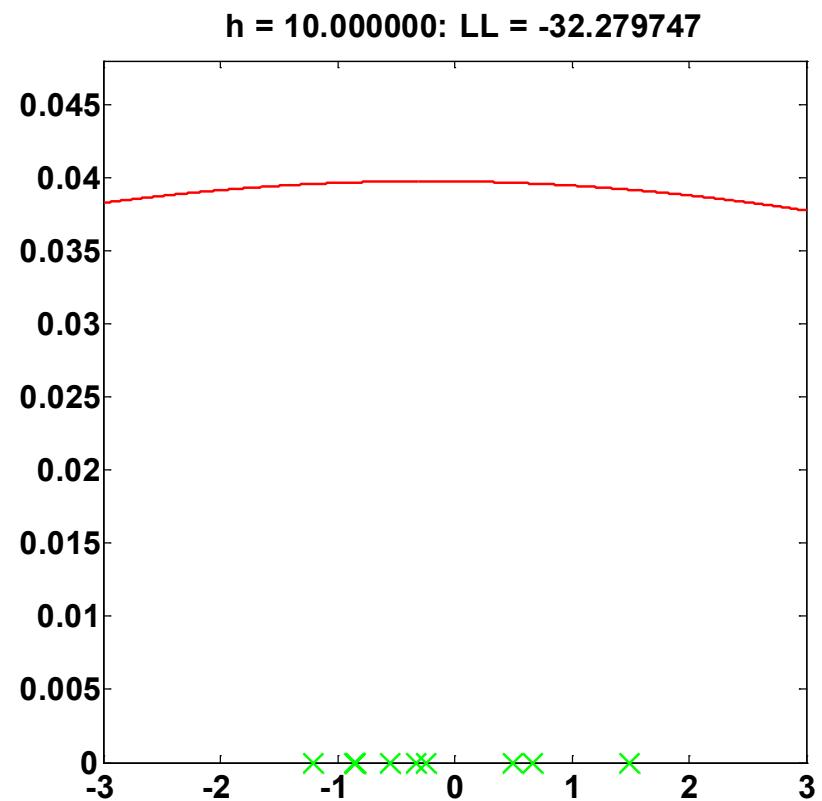
(this assumes independence! )

- **Maximize log-likelihood w.r.t.  $h$**  (*convenient to avoid multiplication*):

$$LL = \log(g(x_1, \dots, x_n)) = \sum_{i=1}^n \log(\hat{p}(x_i | h))$$

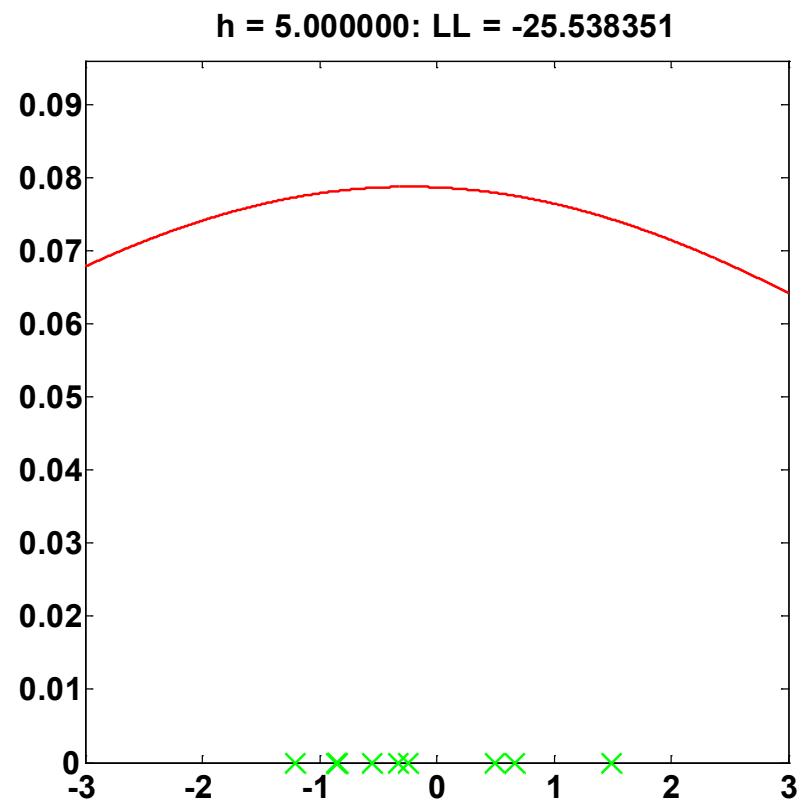
# Parzen density estimation (5)

- Maximum likelihood on training set:



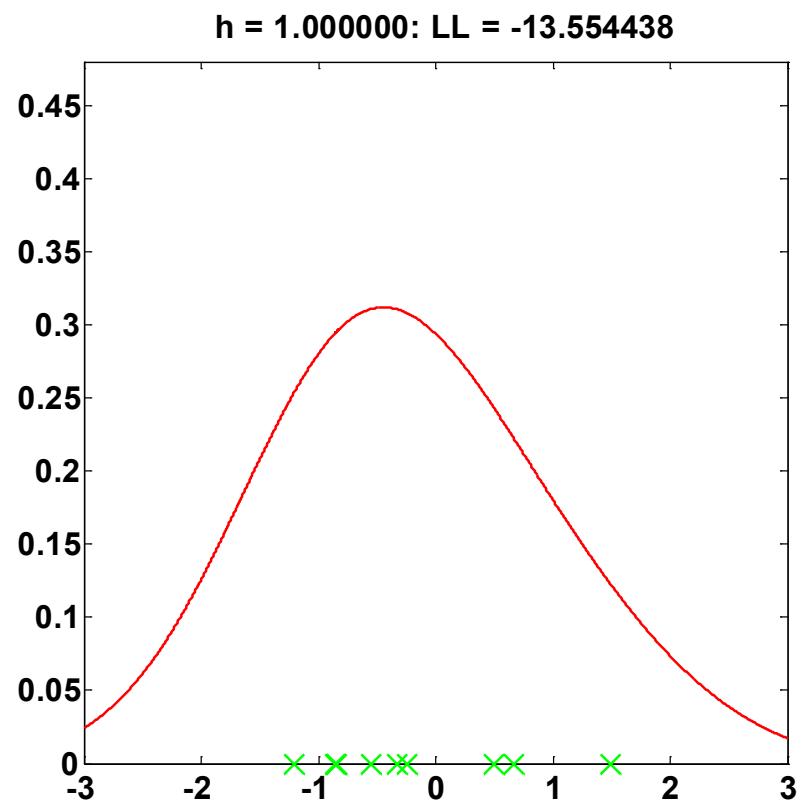
# Parzen density estimation (5)

- Maximum likelihood on training set:



# Parzen density estimation (5)

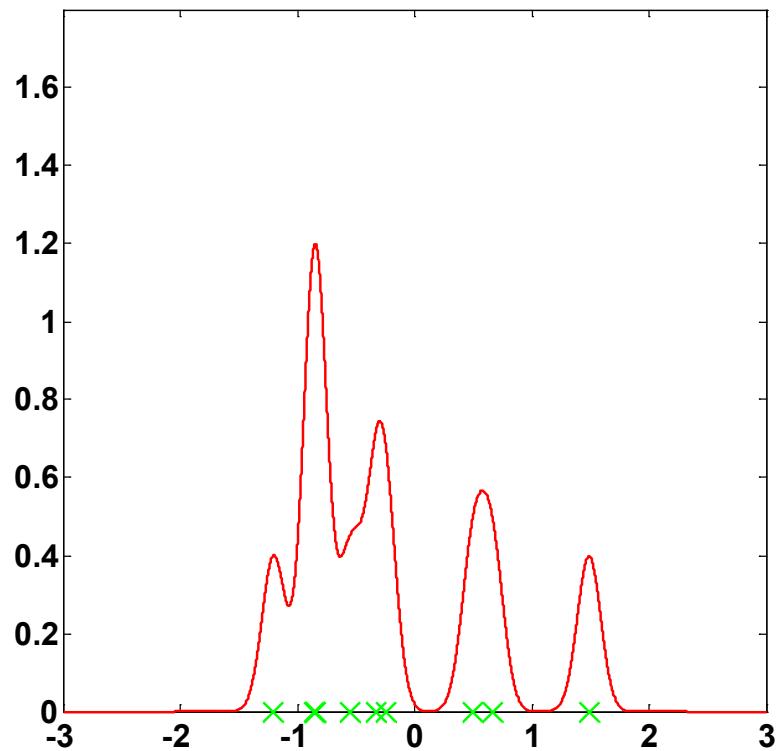
- Maximum likelihood on training set:



# Parzen density estimation (5)

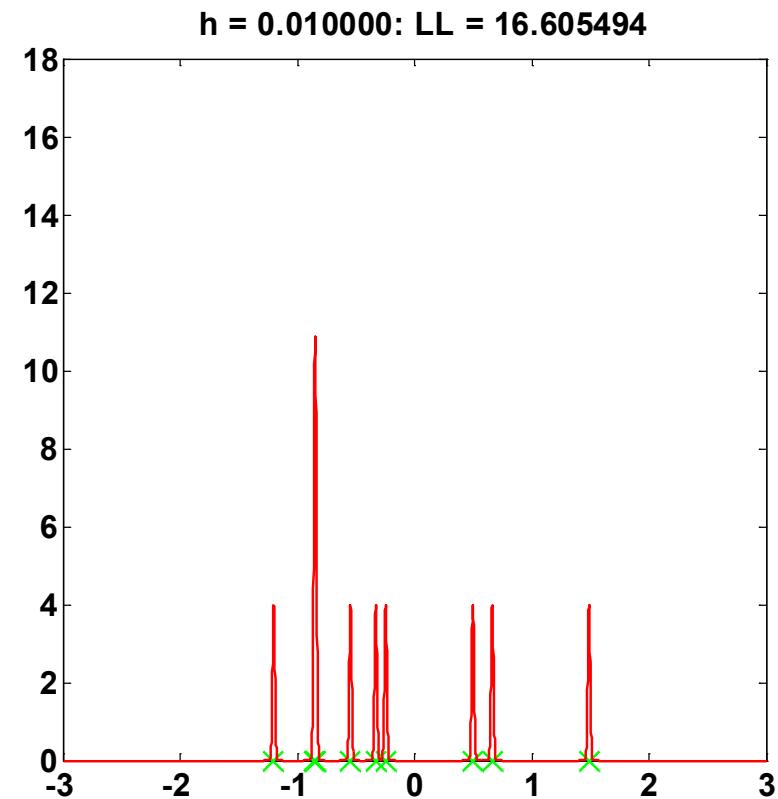
- Maximum likelihood on training set:

$h = 0.100000$ : LL = -4.170235



# Parzen density estimation (5)

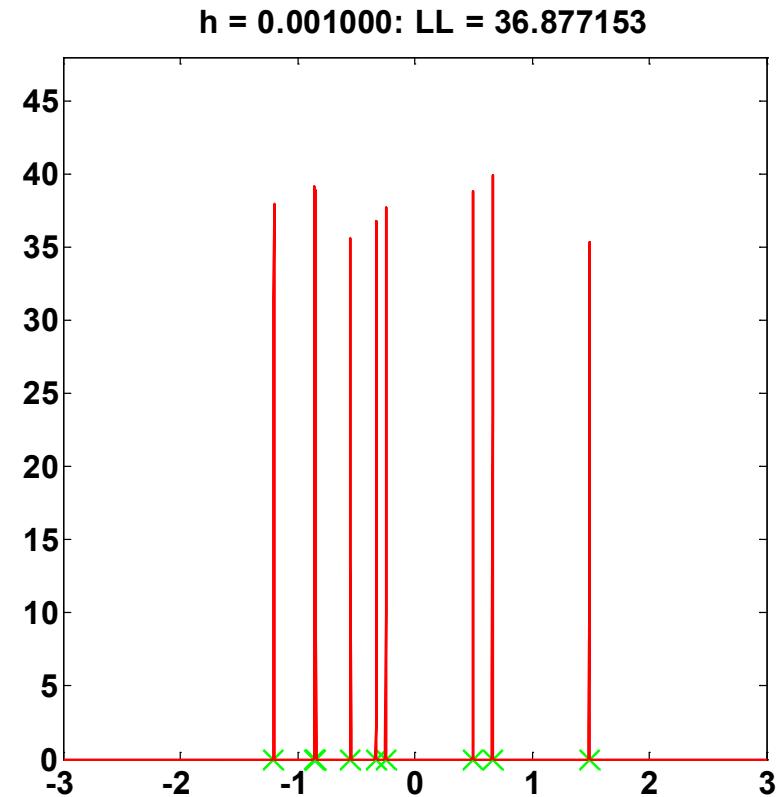
- Maximum likelihood on training set:



# Parzen density estimation (5)

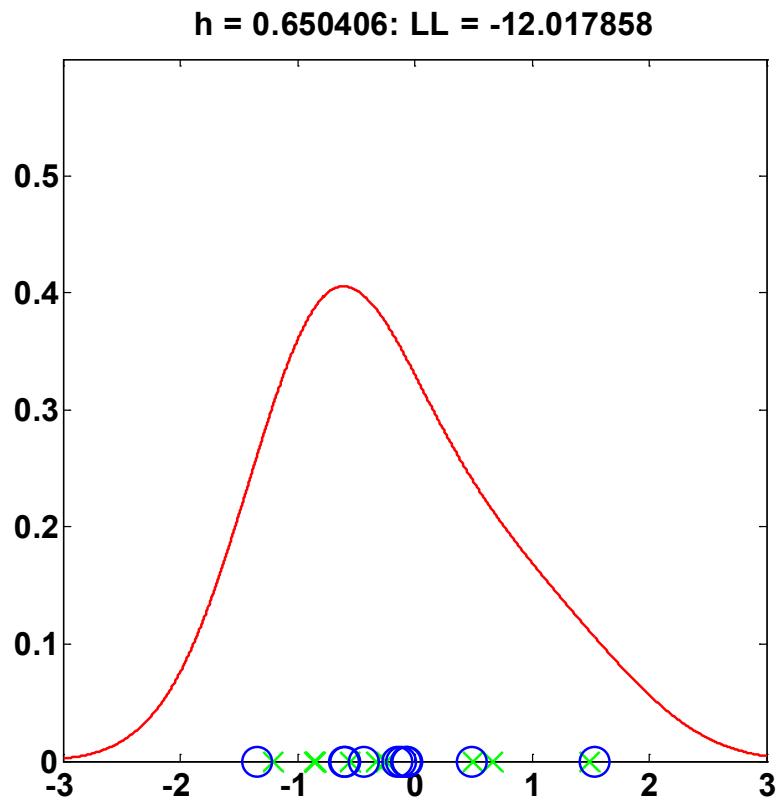
- Maximum likelihood on training set:

- $h \rightarrow 0: LL \rightarrow \infty$
- **Extreme example of *overtraining* : fitting data too much**



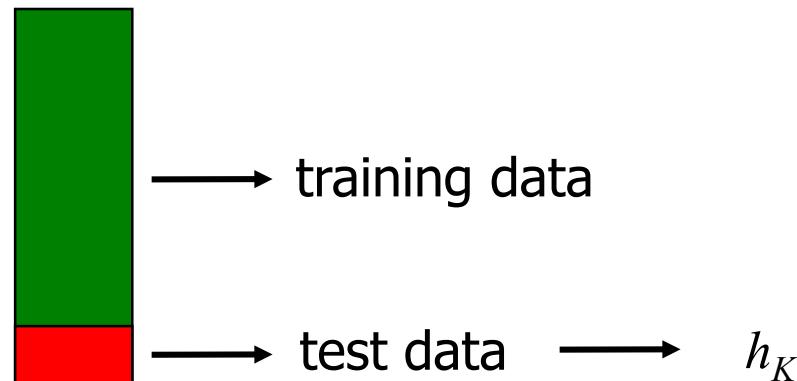
# Cross-validation

- Solution:
  - Split data into *training set* and *validation set*
  - Optimise  $h$  w.r.t. likelihood of validation set, given Parzen model trained on training set
- Problems:
  - Uses a lot of valuable data
  - Sensitive to split of data



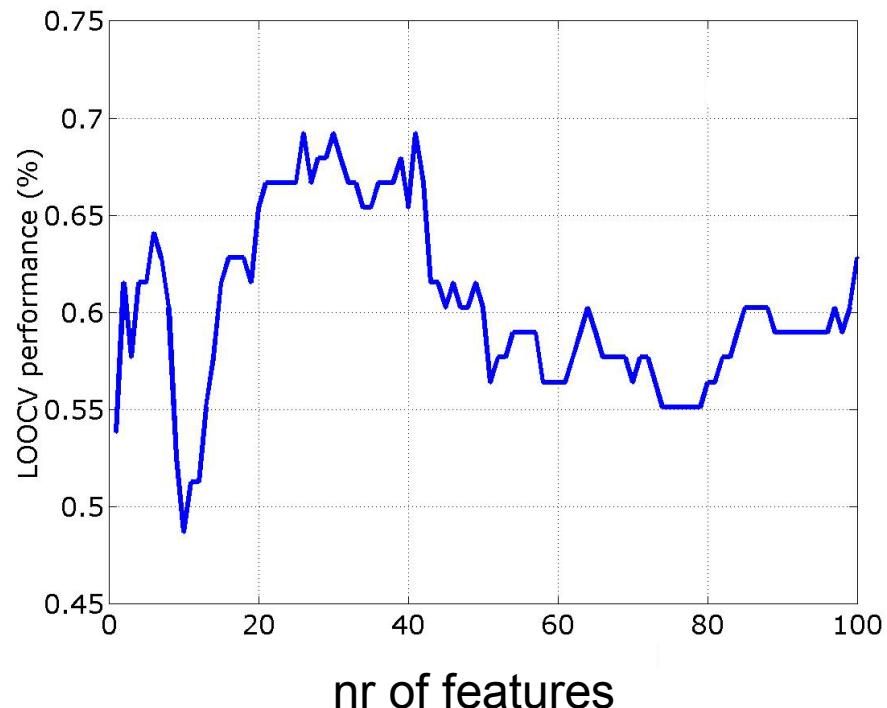
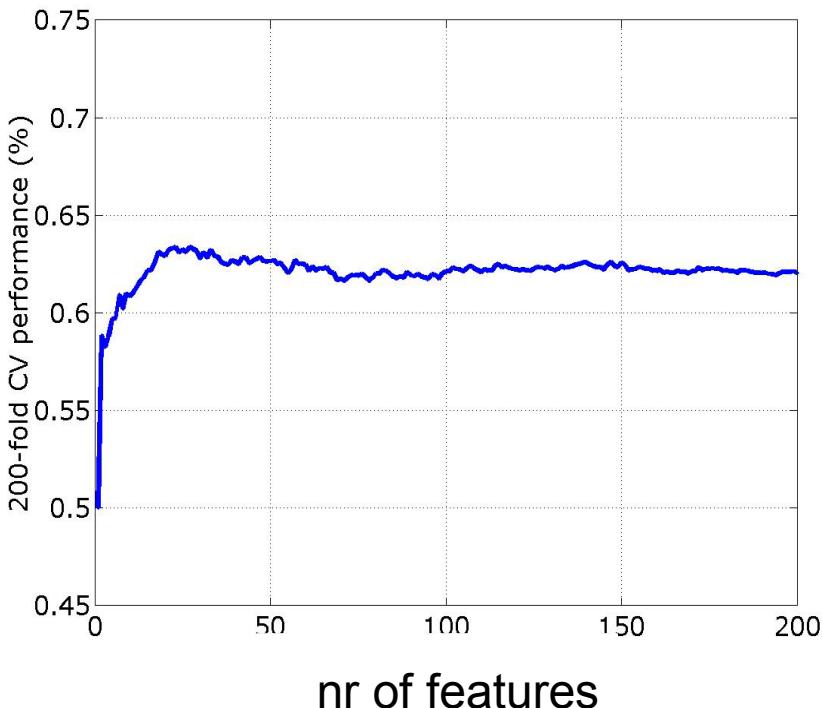
# Cross-validation (2)

- Better solution:  $K$ -fold crossvalidation
  - Split data into  $K$  parts ( $K = n$ : leave-one-out)
  - Repeat  $K$  times:
    - Find  $h$  using  $(K - 1)$  parts for training and 1 part for testing
  - Use average of  $h$ 's as kernel width



# Cross-validation (3)

- (Prefer)  $K$ -fold cross-validation over leave-one-out
  - Smoother (less variance) and more biased (conservative)



# Bootstrap

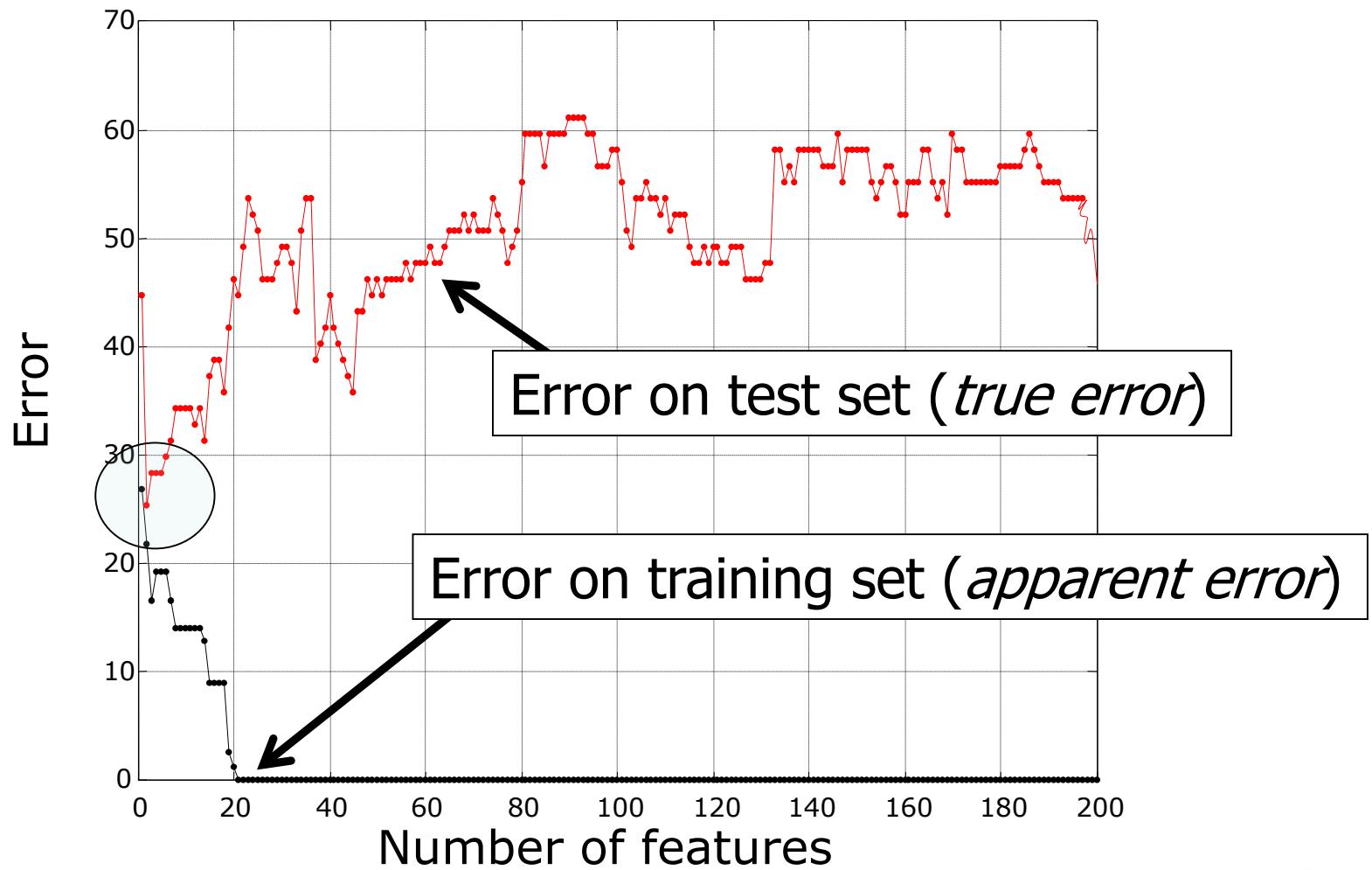
- Alternative to cross-validation:
  - Repeat  $K$  times:
    - Draw  $n$  objects from the dataset, **with replacement** (some objects will be selected more than once)
    - Test using objects that were not selected
- Cross-validation and bootstrap estimates are *biased*
  - They are conservative (i.e. too pessimistic) because they do not use all data available

*You want to get an estimate when you fit on complete/all data.  
CV/Bootstrap are thus biased wrt fitting on complete data!*

# Training, test and validation sets

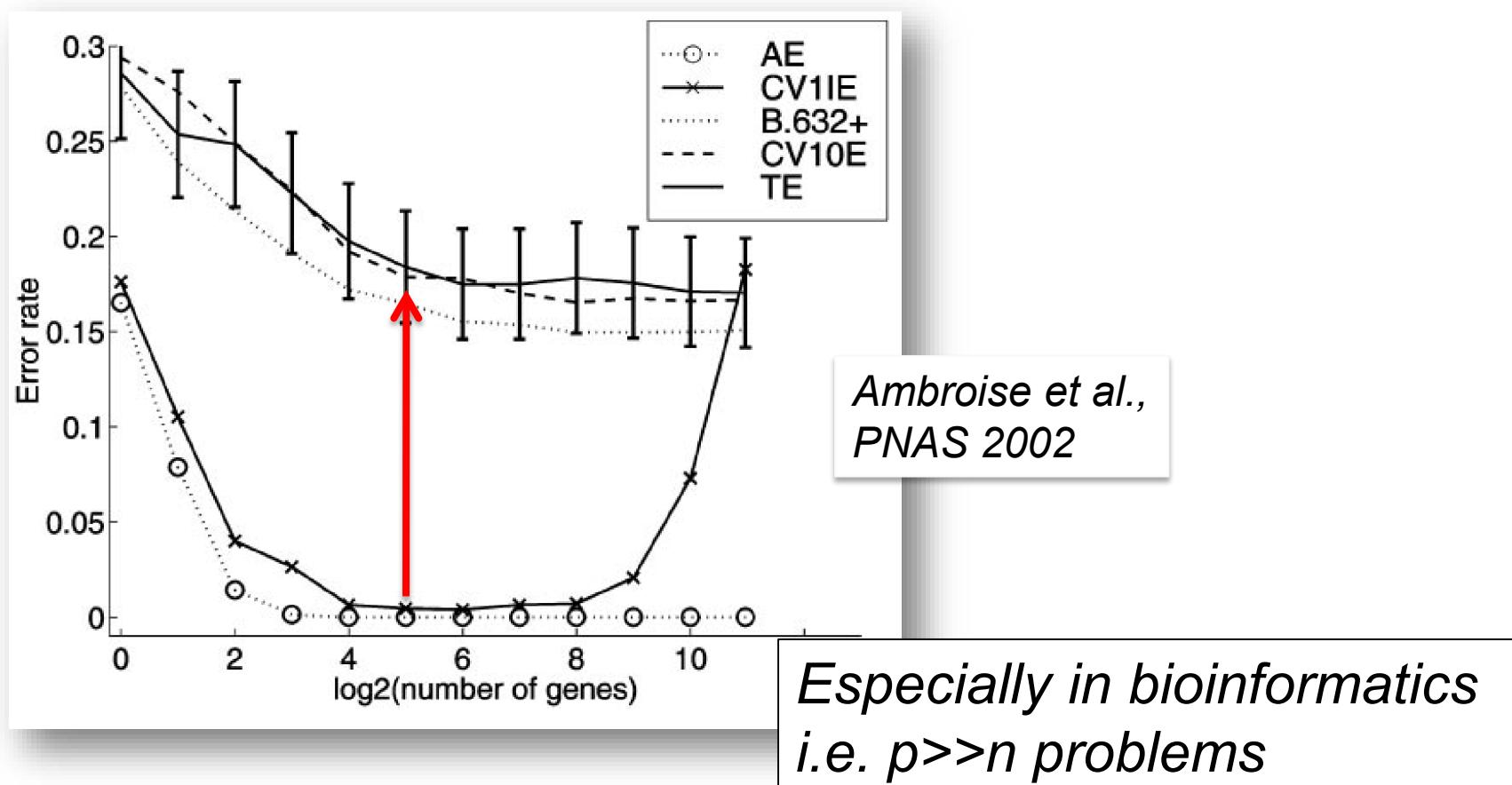
- Terminology:
  - A *training set* is used to estimate parameters
  - An optional *validation set* is used to optimize parameter settings, e.g. by calculating classifier error on this set
  - **A *test set* is only used to judge performance of the entire classifier (only used once!)**
- Error estimates:
  - On training set: *apparent error*
  - On test set: *true error*

# Training, test and validation sets (2)



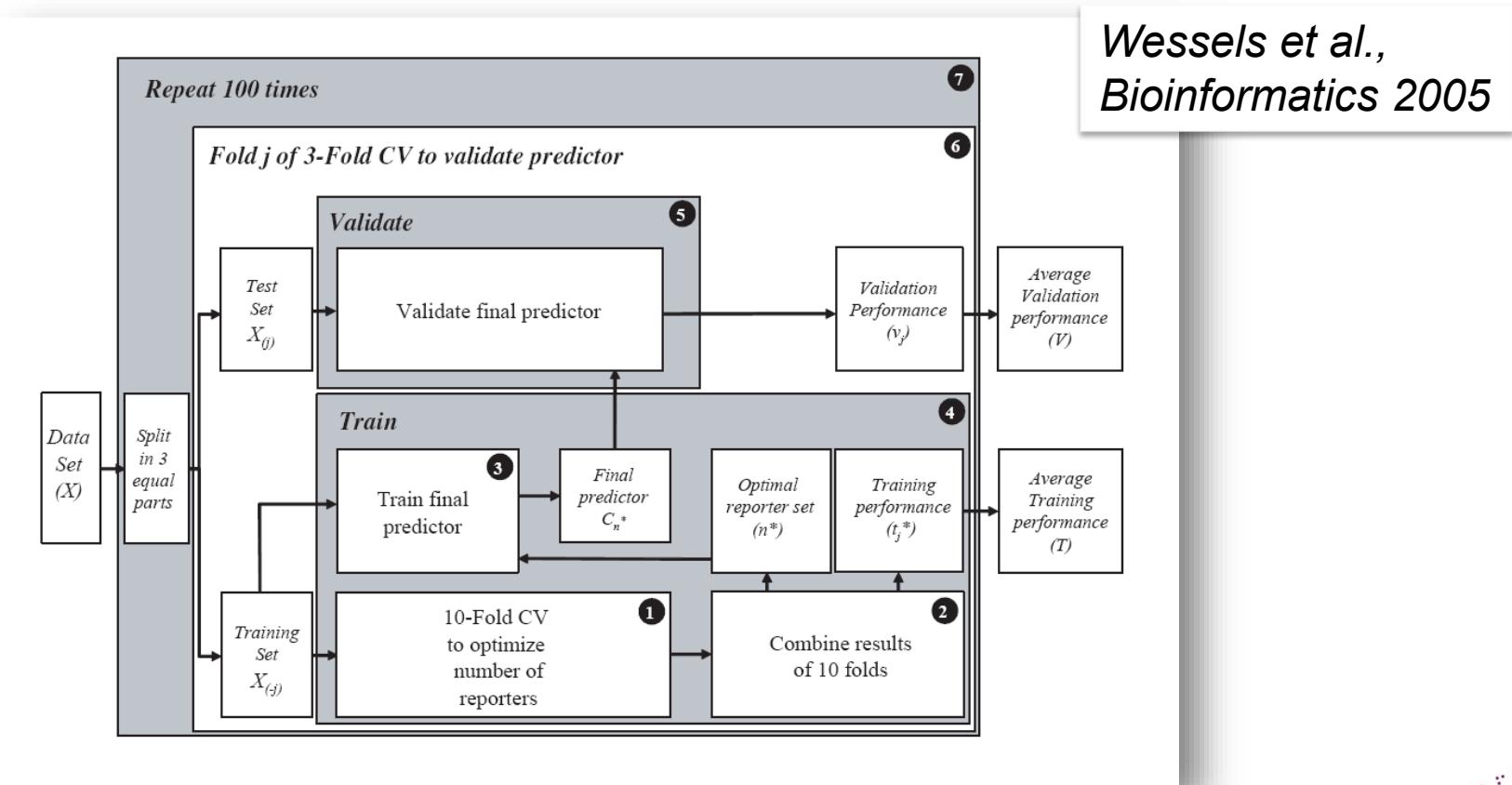
# Training, test and validation sets (3)

- The test set should *never* be used to set any parameters! This leads to biased estimates of performance -- in practice we may do much worse than we predict



# Training, test and validation sets (4)

- Can lead to complicated schemes for estimating parameters, e.g. double/nested cross-validation loops



# Recapitulation

- *Bayesian estimation*
  - provides a framework for minimizing cost due to errors
  - combines class-conditional and prior distributions into posterior ones
- We never *know* these distributions, so we have to *estimate* them; this is problematic due to the *curse of dimensionality*
- Possible approaches:
  - *Parametric*: e.g. Gaussian
  - *Nonparametric*: histogramming,  $k$ -nearest neighbor density estimation, Parzen density estimation

# Recapitulation (2)

- *Maximum likelihood estimation* is a method for estimating parameters of density functions
- To optimize parameters, the error should be calculated on a *validation set*
- A completely independent *test set* should only be used to judge performance of the final classifier
- *Cross-validation* and *bootstrapping* can help to estimate performance when little data is available