UNIVERSITY OF CAMBRIDGE

MPhil in Computational Biology 2016/17

Course: **Functional Genomics**

Lecture 16 (Wednesday 30th November 2016)

Classification

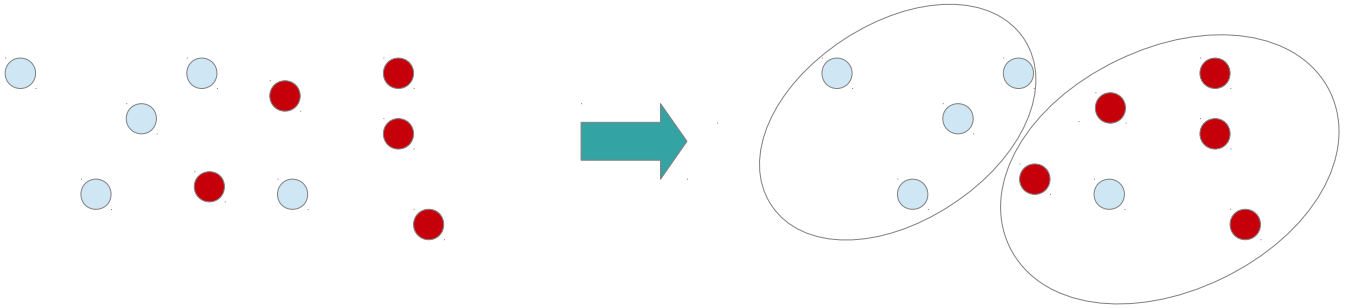
Geoff Macintyre

geoff.macintyre@cruk.cam.ac.uk

Refresher

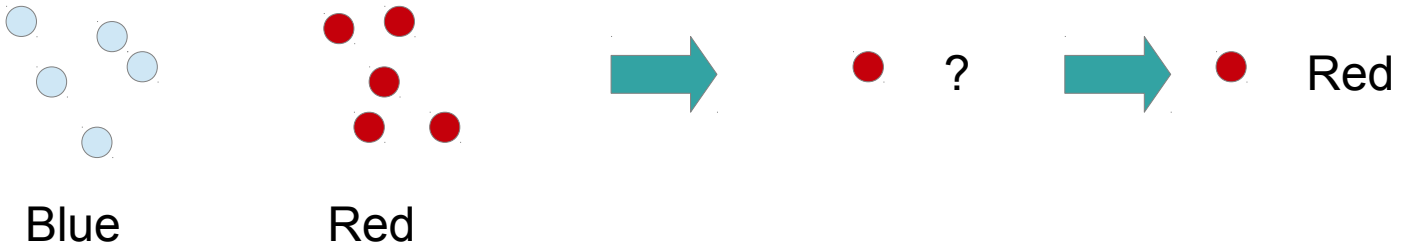
Unsupervised learning (e.g. clustering)

- Algorithms *operate* on **unlabelled** examples

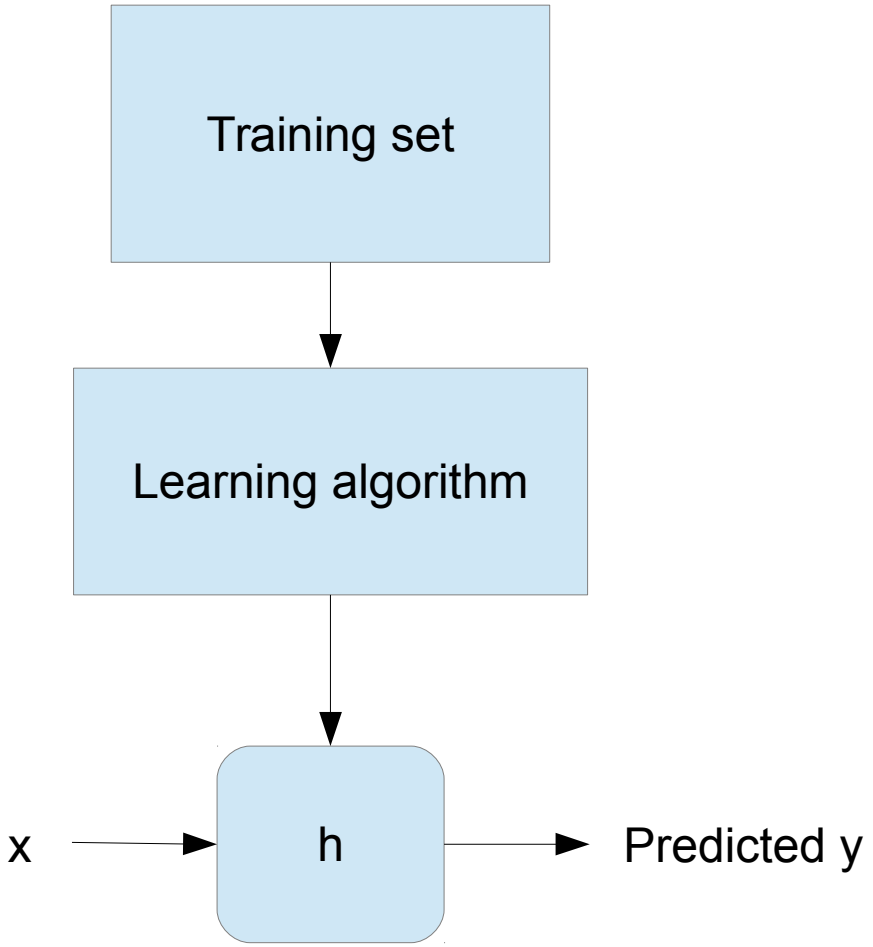


Supervised learning (e.g. classification)

- Algorithms are *trained* on **labelled** examples



Introduction to classification



Some useful notation for classification

- $x_{(i)}$ to denote input, also known as features
- $y_{(i)}$ to denote output, target variable
- A pair $(x_{(i)}, y_{(i)})$ is known as a training example
- A list of *m* training examples

  $\{(x_{(i)}, y_{(i)}); i = 1, \ldots, m\}$ is called a training set.
- ***X*** denote the space of input values, and ***Y*** the space of output values.
- Our learning process is to learn the function

  $h: X \rightarrow Y$, *so that* $h(x)$ is a good predictor of y
- In our examples we will use a binary classifier $\boldsymbol{Y} \rightarrow \{-1, 1\}$

Types of classifiers

Generative (or class-conditional) classifiers

- Learn models for p(x | y_k), use Bayes rule to find decision boundaries
- Examples: naïve Bayes models, Gaussian classifiers

Regression-based classifiers

- Learn a model for p(y_k | x) directly
- Example: logistic regression, neural networks

Discriminative classifiers

- No probabilities
- Learn the decision boundaries directly
- Examples:
  - Linear boundaries: perceptrons, linear SVMs
  - Piecewise linear boundaries: decision trees, nearest-neighbor classifiers
  - Non-linear boundaries: non-linear SVMs

Performance

Measuring performance

		Predicted label	
		Positive	**Negative**
True Label	**Positive**	True positive	False negative
	Negative	False positive	True Negative

- True positive = correctly identified (TP)
- False positive = incorrectly identified (FP) – Type I error
- True negative = correctly rejected (TN)
- False negative = incorrectly rejected (FN) – Type II error

Some common metrics

Accuracy

(TP+TN)/(TP+FP+TN+FN)

Sensitivity / recall / true positive rate

TP/(TP+FN)

False positive rate

1-TN/(TN+FP)

Specificity

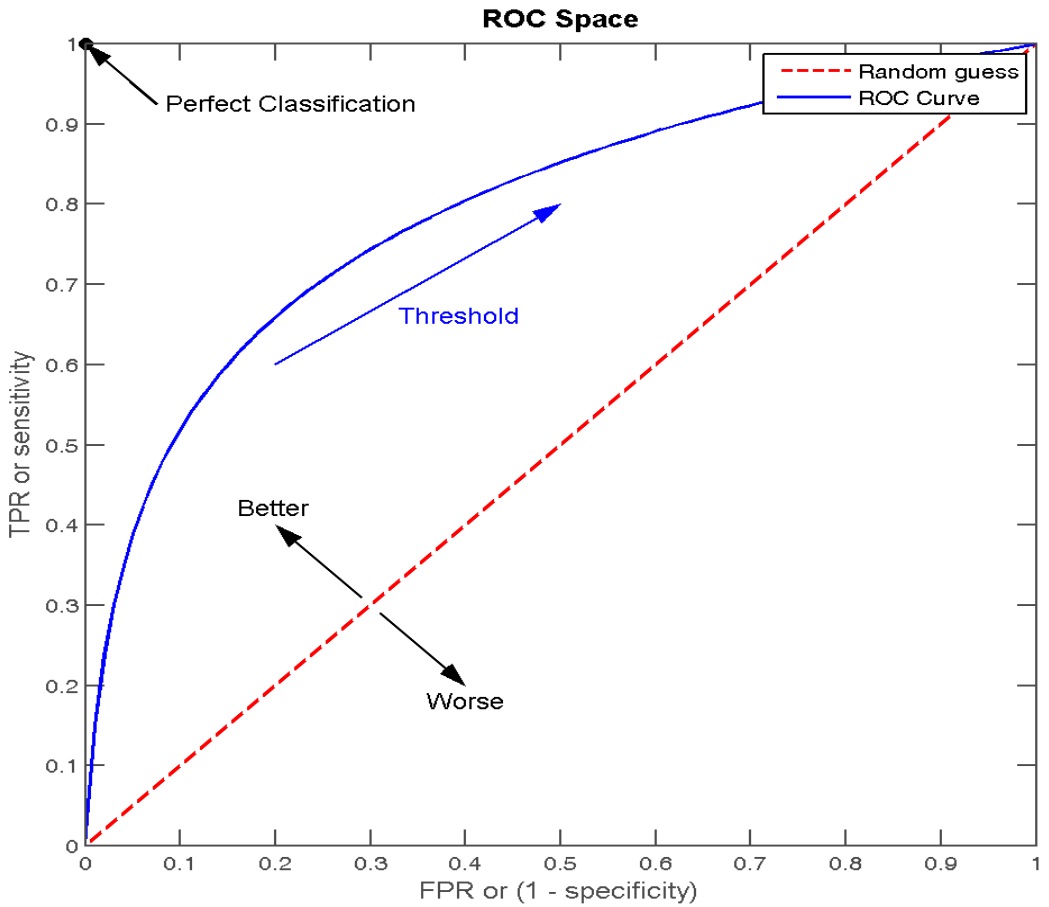
TN/(TN+FP)

Precision / Positive predicted value

TP/(TP+FP)

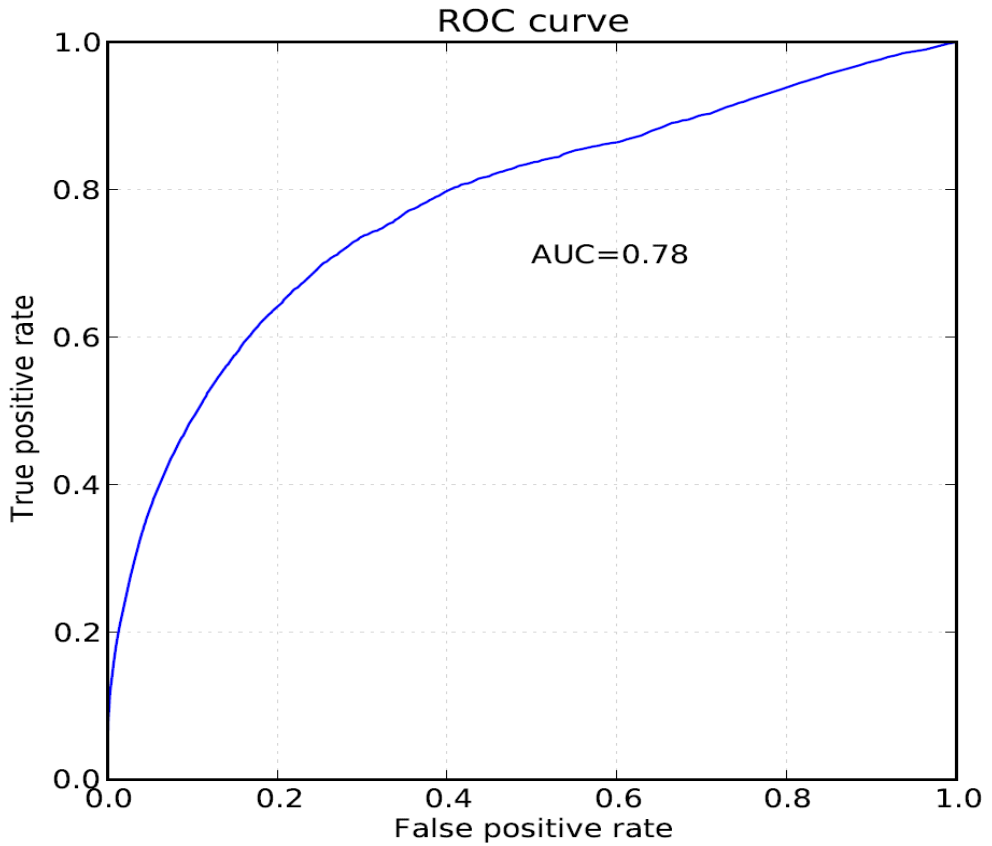
The trade-off between two metrics

- Increasing specificity usually comes at the cost of a decrease in sensitivity
- The Receiver Operating Characteristic Curve shows the trade off between Sensitivity (TPR) and 1-Specificity (FPR)



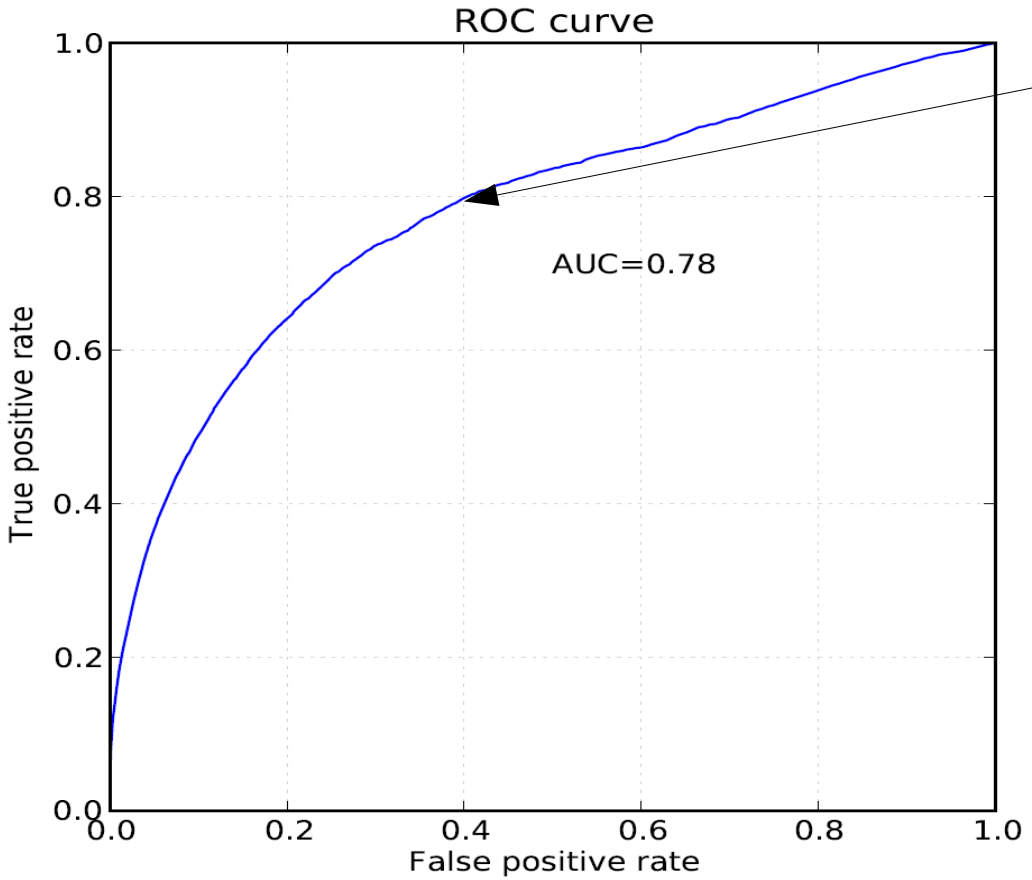
Area under the curve

- What if we don't care about a threshold?
- Or if we want to know the average performance of a classifier?



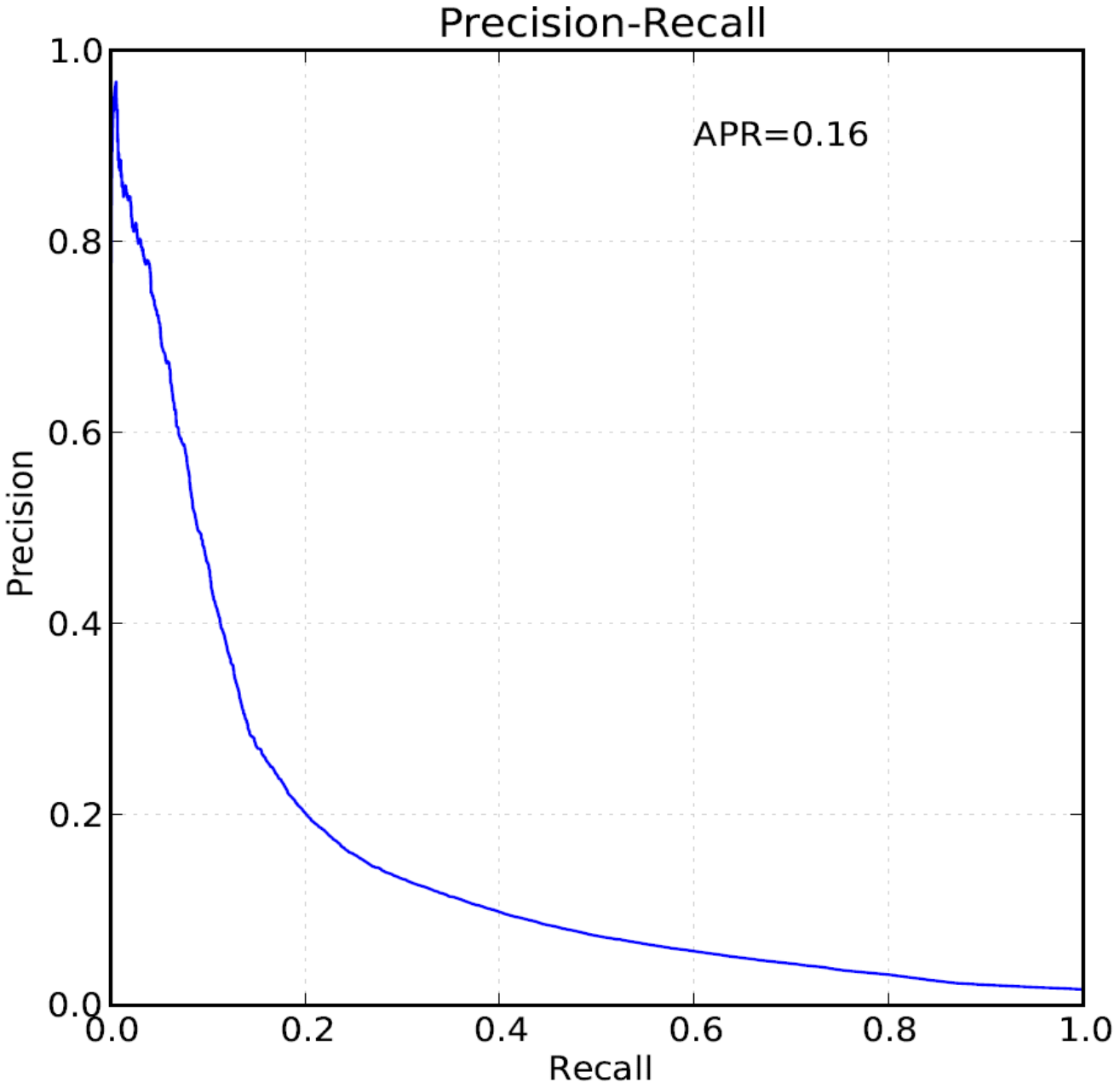
What happens with imbalanced data?

- Few positive examples embedded in a sea of negative examples



80% TPR
40% FPR
BUT....
3% Precision!

Precision-recall curve



Performance summary

- Area under the curve can be used to assess and optimise classifier performance
- ROC curve best used when
  - Class labels are balanced
  - TP, FP, TN and FN are known
- PR curve best used when
  - Class labels are imbalanced
  - TN are not known

Training

Introduction to training (concept learning)

Basic idea:

- Given:
  - a set of training examples with labels
  - a parameterised hypothesis/model
- Use each training example to find values for model parameters which *improves* ability to predict correct class label

Typical approach:

- Divide your data into a training set (80%) and a test set (20%)
- Train the algorithm on the training set and assess performance using the *independent* test set

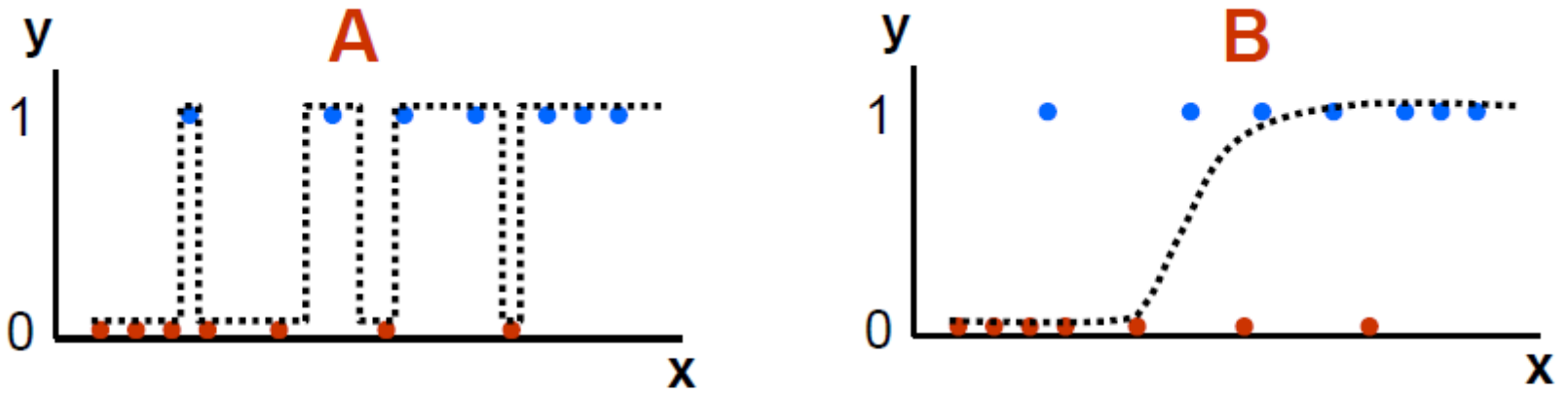
DANGER - Biased training and testing

- Biased training is where all the data available is used for training and testing
- May result in over-fitting.....

Over-fitting

- Over-fitting is where an algorithm is trained too well on a specific dataset and does not generalise
- Can result from
  - Biased training
  - Small training sets
  - Over parameterised models

Over-fitting



Model A gives a smaller classification error on the training set.

Model B is obviously better for predicting future values.

Slide from Ilya Narsky

A good training strategy – cross-validation

Divide your data into three sets

- Training (50%)
- Validation (25%)
- Test (25%)

Validation and training set may be divided multiple times and training performed each time. This is called cross-validation training.

Cross-validation training improves the ability of your model to generalise to new data.

Bayesian inference

Why bayesian?

- Can incorporate prior knowledge
- Models hypotheses in terms of probabilities
- When to use
  - Moderate or large training set available
  - Attributes that describe instances are conditionally independent given classification
- Successful applications:
  - Diagnosis
  - Classifying text documents

Bayesian approaches

- Focuses on P(H|D), probability of a hypothesis given the data
- (Frequentist approach focuses on P(D|H))
- This means that the data is fixed
- and that that they hypotheses are random
- (The hypotheses may be true or false with some probability)
- Can't calculate P(H|D) directly BUT we can using Bayes Theorem

Bayes theorem

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

- $P(h)$ = prior probability of hypothesis h
- $P(D)$ = prior probability of training data D
- $P(h|D)$ = probability of h given D
- $P(D|h)$ = probability of D given h

Bayesian learning

- We want to find the best hypothesis *h,* given the observed training data *D*
- We want the *maximium a posteriori* hypothesis
- We can use Bayes theorem to find this:

$$
\begin{aligned}
h_{MAP} &= \arg\max_{h \in H} P(h|D) \\
&= \arg\max_{h \in H} \frac{P(D|h)P(h)}{P(D)} \\
&= \arg\max_{h \in H} P(D|h)P(h)
\end{aligned}
$$

- *P(D)* is removed as it is a constant independent of *h*

An example

- Does a patient have cancer or not?
  - A patient takes a lab test and the result comes back positive. The test returns a correct positive result in 98% of the cases in which the disease is actually present, and a correct negative result in 97% of the cases in which the disease is not present. Furthermore, 0.008 of the entire population have this cancer
- Should we conclude the patient has cancer?

An example (cont)

A patient takes a lab test and the result comes back positive. The test returns a correct positive result in 98% of the cases in which the disease is actually present, and a correct negative result in 97% of the cases in which the disease is not present. Furthermore, 0.008 of the entire population have this cancer

P(cancer)	=	*P(!cancer)*	=
P(+ | cancer)	=	*P(- | cancer)*	=
P(+ | !cancer)	=	*P(- | !cancer)*	=

An example (cont)

A patient takes a lab test and the result comes back positive. The test returns a correct positive result in 98% of the cases in which the disease is actually present, and a correct negative result in 97% of the cases in which the disease is not present. Furthermore, 0.008 of the entire population have this cancer

P(cancer)	*= 0.008*	*P(!cancer)*	=
P(+ | cancer)	=	*P(- | cancer)*	=
P(+ | !cancer)	=	*P(- | !cancer)*	=

An example (cont)

A patient takes a lab test and the result comes back positive. The test returns a correct positive result in 98% of the cases in which the disease is actually present, and a correct negative result in 97% of the cases in which the disease is not present. Furthermore, 0.008 of the entire population have this cancer

P(cancer)	*= 0.008*	*P(!cancer)*	*= 0.992*
P(+ | cancer)	=	*P(- | cancer)*	=
P(+ | !cancer)	=	*P(- | !cancer)*	=

An example (cont)

A patient takes a lab test and the result comes back positive. The test returns a correct positive result in 98% of the cases in which the disease is actually present, and a correct negative result in 97% of the cases in which the disease is not present. Furthermore, 0.008 of the entire population have this cancer

P(cancer)	*= 0.008*	*P(!cancer)*	*= 0.992*
P(+ | cancer)	*= 0.98*	*P(- | cancer)*	=
P(+ | !cancer)	=	*P(- | !cancer)*	=

An example (cont)

A patient takes a lab test and the result comes back positive. The test returns a correct positive result in 98% of the cases in which the disease is actually present, and a correct negative result in 97% of the cases in which the disease is not present. Furthermore, 0.008 of the entire population have this cancer

P(cancer)	*= 0.008*	*P(!cancer)*	*= 0.992*
P(+ | cancer)	*= 0.98*	*P(- | cancer)*	*= 0.02*
P(+ | !cancer)	=	*P(- | !cancer)*	=

An example (cont)

A patient takes a lab test and the result comes back positive. The test returns a correct positive result in 98% of the cases in which the disease is actually present, and a correct negative result in 97% of the cases in which the disease is not present. Furthermore, 0.008 of the entire population have this cancer

P(cancer)	*= 0.008*	*P(!cancer)*	*= 0.992*
P(+ | cancer)	*= 0.98*	*P(- | cancer)*	*= 0.02*
P(+ | !cancer)	*= 0.03*	*P(- | !cancer)*	*= 0.97*

Should we conclude the patient has cancer?

$P(cancer) = 0.008$ $P(!cancer) = 0.992$

$P(+ \mid cancer) = 0.98$ $P(- \mid cancer) = 0.02$

$P(+ \mid !cancer) = 0.03$ $P(- \mid !cancer) = 0.97$

Hypothesis 1: Patient has cancer

$h_1(+) = P(+ \mid cancer)P(cancer) = 0.98x0.008 = 0.0078$

Hypothesis 2: Patient doesn't have cancer

$h_2(+) = P(+ \mid !cancer)P(!cancer) = 0.03x0.992 = 0.0298$

$h_{MAP} = !cancer$

Bayesian classification

The previous example dealt with “what is the most probable *hypothesis* given the training data”

We are usually interested in “what is the most probable *classification* of a new instance given the training data”

Why not just apply MAP?

Answer: $h_{MAP}(x)$ is not the most probable classification!

Consider:

- Three possible hypotheses:

  $P(h_1|D) = 0.4$; $P(h_2|D) = 0.3$; $P(h_3|D) = 0.3$

- Given new instance x,

  $h_1(x) = +$; $h_2(x) = -$; $h_3(x) = -$

- What's most probable classification of x?

Bayes Optimal Classifier

In general, the most probable classification of a new instance is obtained by combining predictions of all hypotheses, weighted by their posterior probabilities

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i) P(h_i|D)$$

the optimal classification of a new instance is the value v_j for which $P(v_j|D)$ is maximum (theorem of total probability).

Example:

$$P(h_1|D) = .4, \;\; P(-|h_1) = 0, \;\; P(+|h_1) = 1$$
$$P(h_2|D) = .3, \;\; P(-|h_2) = 1, \;\; P(+|h_2) = 0$$
$$P(h_3|D) = .3, \;\; P(-|h_3) = 1, \;\; P(+|h_3) = 0$$

therefore

$$\sum_{h_i \in H} P(+|h_i)P(h_i|D) \;=\; .4$$
$$\sum_{h_i \in H} P(-|h_i)P(h_i|D) \;=\; .6$$

and

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D) \;=\; -$$

Bayes optimal classifier

No other classification methods using the same hypothesis space and same prior knowledge can outperform this approach.

But.....

It is VERY costly as it computes the posterior probability for EVERY hypothesis.

→ Bad if your hypothesis space is large

Naive Bayes Classifier

Assume target function $f : X \to V$, where each instance x described by attributes $\langle a_1, a_2 \ldots a_n \rangle$. Most probable value of $f(x)$ is:

$$
\begin{aligned}
v_{MAP} &= \underset{v_j \in V}{\operatorname{argmax}} P(v_j | a_1, a_2 \ldots a_n) \\
v_{MAP} &= \underset{v_j \in V}{\operatorname{argmax}} \frac{P(a_1, a_2 \ldots a_n | v_j) P(v_j)}{P(a_1, a_2 \ldots a_n)} \\
&= \underset{v_j \in V}{\operatorname{argmax}} P(a_1, a_2 \ldots a_n | v_j) P(v_j)
\end{aligned}
$$

Naive Bayes assumption:

$$P(a_1, a_2 \ldots a_n | v_j) = \prod_i P(a_i | v_j)$$

which gives

Naive Bayes classifier: $v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_i P(a_i | v_j)$

Naive Bayes Algorithm

Naive_Bayes_Learn($examples$)

For each target value v_j

$\hat{P}(v_j) \leftarrow$ estimate $P(v_j)$

For each attribute value a_i of each attribute a

$\hat{P}(a_i|v_j) \leftarrow$ estimate $P(a_i|v_j)$

Classify_New_Instance(x)

$$v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} \hat{P}(v_j) \prod_{a_i \in x} \hat{P}(a_i|v_j)$$

Example

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Consider a new instance

$$\langle Outlk = sun, Temp = cool, Humid = high, Wind = strong$$

Want to compute:

$$v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_i P(a_i|v_j)$$

$$P(y)\ P(sun|y)\ P(cool|y)\ P(high|y)\ P(strong|y) = .005$$

$$P(n)\ P(sun|n)\ P(cool|n)\ P(high|n)\ P(strong|n) = .021$$

$$\rightarrow v_{NB} = n$$

naive Bayes subtleties

- Conditional independence assumption is often violated.....but works well anyway!
- Problem: what if none of the training instances with target value v_j have attribute value a_j?

$$\hat{P}(a_i|v_j) = 0, \text{ and...}$$

$$\hat{P}(v_j) \prod_i \hat{P}(a_i|v_j) = 0$$

Typical solution is Bayesian estimate for $\hat{P}(a_i|v_j)$

$$\hat{P}(a_i|v_j) \leftarrow \frac{n_c + mp}{n + m}$$

$$\hat{P}(a_i|v_j) = 0, \text{ and...}$$

$$\hat{P}(v_j) \prod_i \hat{P}(a_i|v_j) = 0$$

Typical solution is Bayesian estimate for $\hat{P}(a_i|v_j)$

$$\hat{P}(a_i|v_j) \leftarrow \frac{n_c + mp}{n + m}$$

where

- n is number of training examples for which $v = v_j$,
- n_c number of examples for which $v = v_j$ and $a = a_i$
- p is prior estimate for $\hat{P}(a_i|v_j)$
- m is weight given to prior (i.e. number of "virtual" examples)

Biomarker discovery

Biomarker discovery

- Concerned with finding the minimal set of measurable biological features that allow prediction of:
  - Diagnosis
  - Prognosis
  - Response to drug
  - …...
- Examples:
  - Gene expression
  - Protein expression
  - Metabolite levels
  - Lipid levels
  - …....

Case study: identification of biomarkers which predict clinical outcome in breast cancer

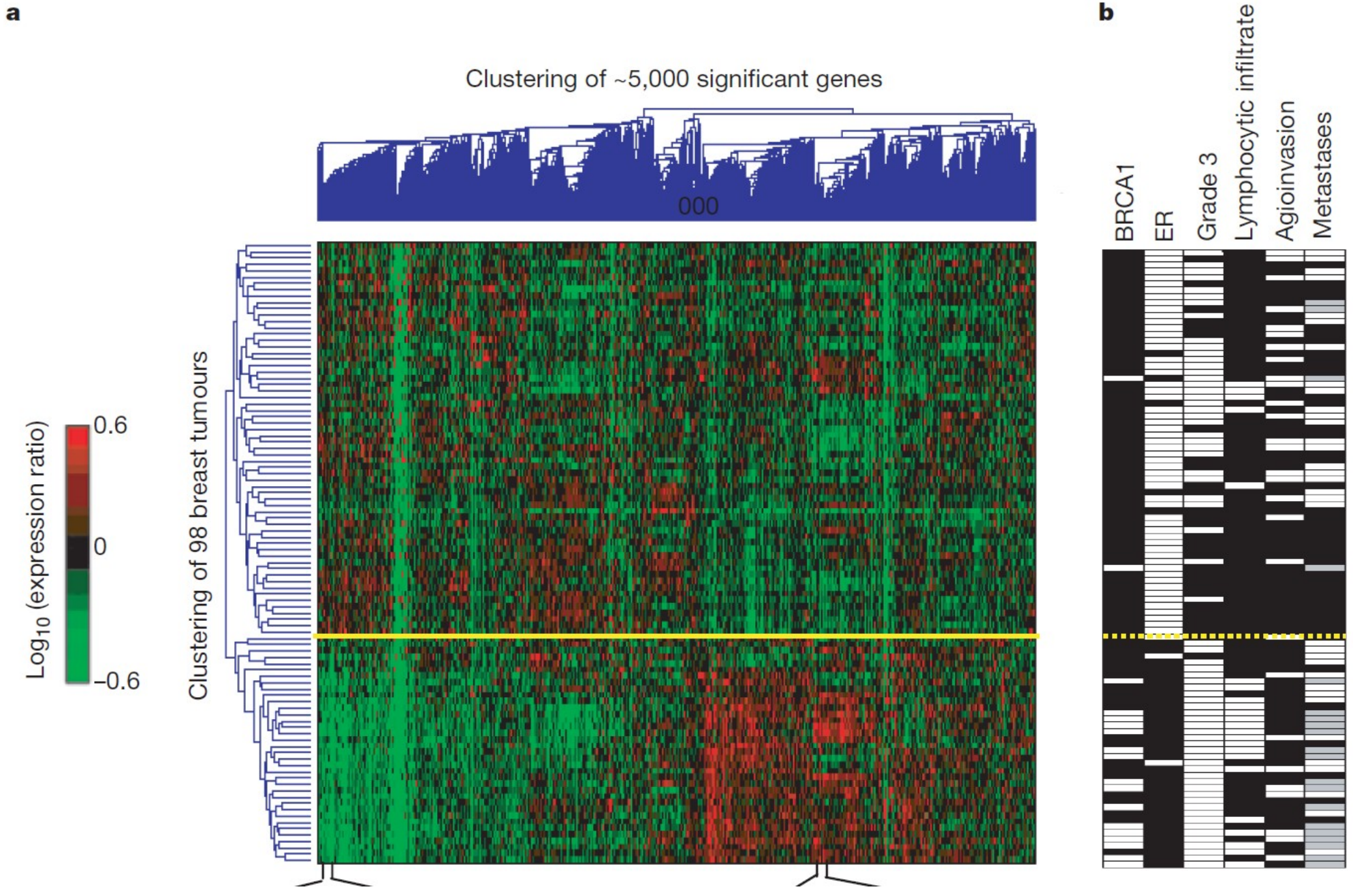
Gene expression profiling predicts clinical outcome of breast cancer

Laura J. van 't Veer*†, Hongyue Dai†‡, Marc J. van de Vijver*†, Yudong D. He‡, Augustinus A. M. Hart*, Mao Mao‡, Hans L. Peterse*, Karin van der Kooy*, Matthew J. Marton‡, Anke T. Witteveen*, George J. Schreiber‡, Ron M. Kerkhoven*, Chris Roberts‡, Peter S. Linsley‡, René Bernards* & Stephen H. Friend‡

Study overview

- Performed gene expression profiling on 98 primary breast tumours from young individuals
  - 34 who got metastases < 5 years
  - 44 who were disease free after 5 years
  - 18 BRCA1 carriers
  - 2 BRCA2 carriers
- Recorded their progression time to metastasis
- Short progression time = poor prognosis
- Performed unsupervised clustering and showed a potential to separate samples based on good and poor prognosis
- Used supervised classification to identify a 70 gene signature which accurately predicts poor prognosis

Unsupervised clustering

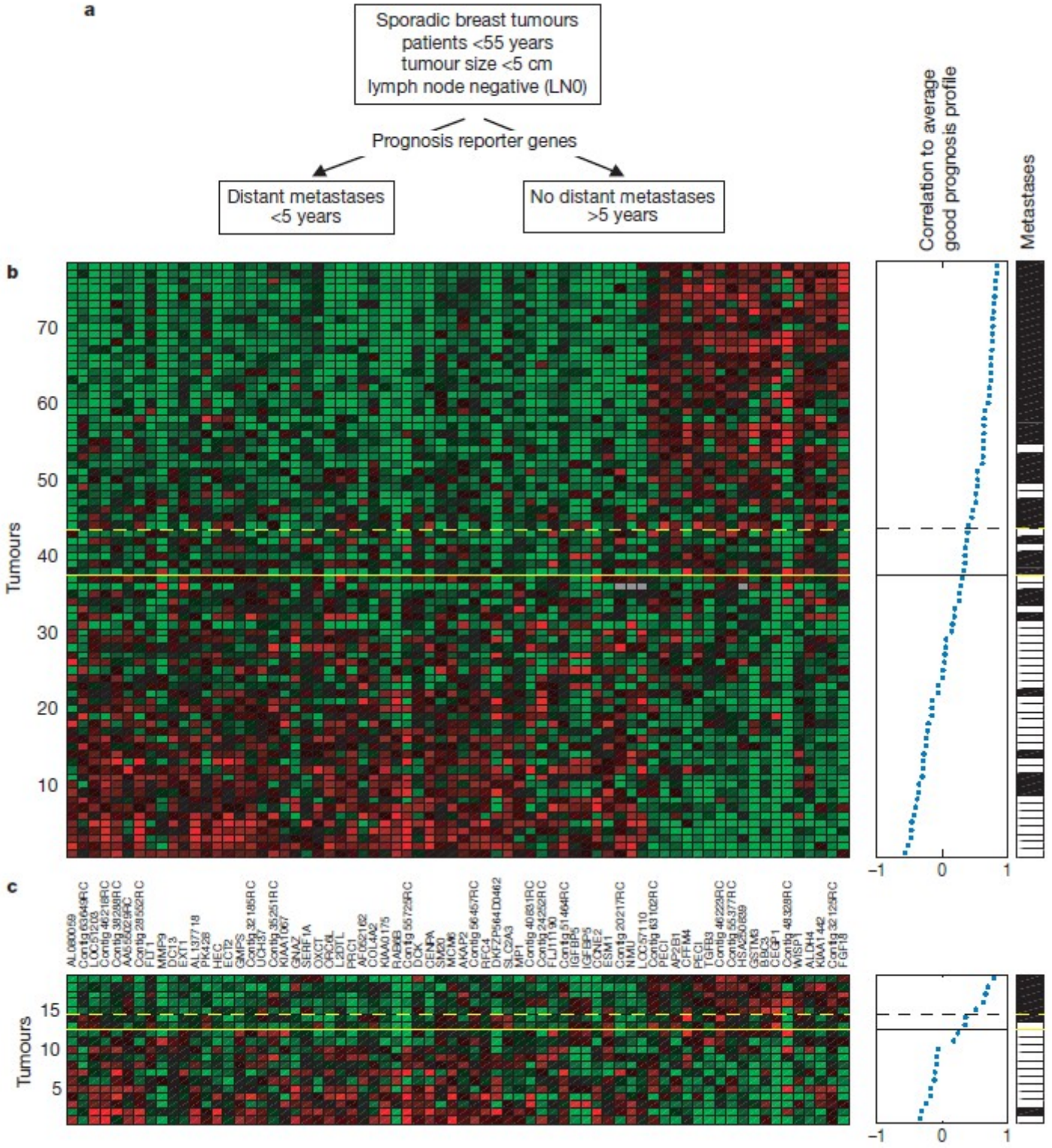


Supervised classification

- Used a correlation co-efficient based method to select a subset of 231 genes where expression correlated with outcome

  → putative biomarkers
- Used a leave-one-out cross validation procedure to determine error of 'classifier'
- Repeated procedure adding 5 genes at a time until error increased

  → 70 genes remained (biomarkers)
- Overall 83% accuracy


a
Sporadic breast tumours
patients <55 years
tumour size <5 cm
lymph node negative (LN0)
Prognosis reporter genes
Distant metastases
<5 years
No distant metastases
>5 years
b
Tumours
70
60
50
40
30
20
10
Correlation to average
good prognosis profile
Metastases
−1
0
1
c
AL080059
Contig63649RC
LOC51203
Contig46218RC
Contig38288RC
AA555029RC
Contig29552RC
FLT1
MMP9
DC13
EXT1
AL137718
PK428
HEC
ECT2
GMPS
Contig32185RC
UCH37
Contig35251RC
KIAA1067
GNAZ
SERF1A
OXCT
ORC6L
L2DTL
PRC1
AF052162
COL4A2
KIAA0175
RAB6B
Contig55725RC
DCK
CENPA
SM20
MCM6
AKAP2
Contig56457RC
RFC4
DKFZp564D0462
SLC2A3
MP1
Contig40831RC
Contig24252RC
FLJ11190
Contig51464RC
IGFBP5
IGFBP5
CDNE2
ESM1
Contig20217RC
NMU
LOC57110
Contig63102RC
PECI
AP2B1
CFFM4
PECI
TGFB3
Contig46223RC
Contig55377RC
HSA250839
GSTM3
BBC3
CEGP1
Contig48328RC
WISP1
ALDH4
KIAA1442
Contig32125RC
FGF18
Tumours
15
10
5
−1
0
1

References

- Tom M. Mitchell. *Machine Learning.* WCB/McGraw-HIll
- Gene expression profiling predicts clinical outcome of breast cancer. van 't Veer, Laura J.; Dai, Hongyue; van de Vijver, Marc J.; He, Yudong D.; Hart, Augustinus A. M. et al. Nature (2002)
- Supervised learning in R: https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R