# Sample Classification based on High-dimensional Data

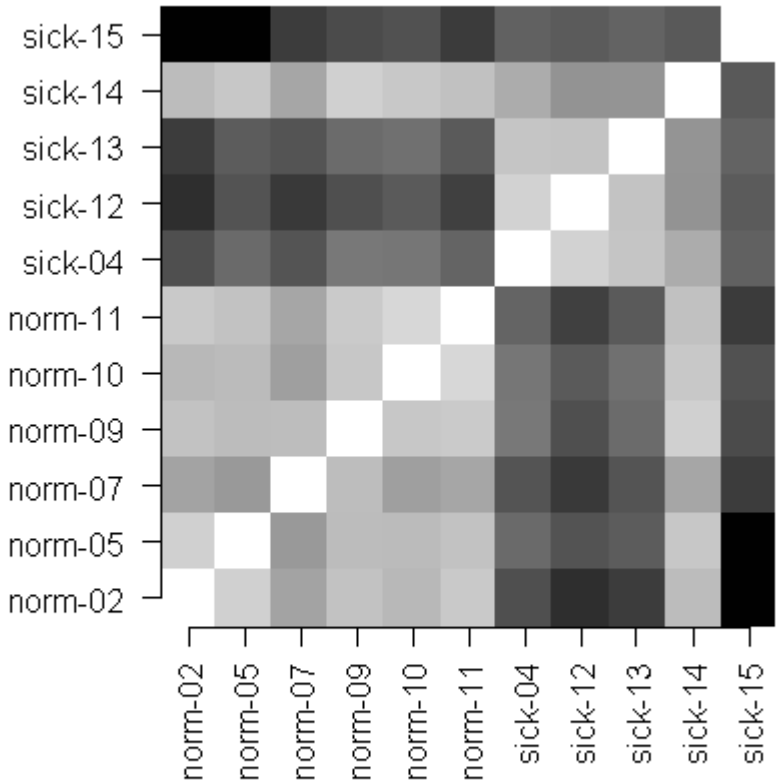
Hubert Rehrauer

# Example: Classifying tumors

- Tumor diagnosis: is a tumor benign or malignant?
- Classification according to physiological characteristics (appearance, size, shape, …) not reliable
- Assumption: Malignancy is defined at the molecular level
- Approach: Make a molecular profile and predict malignancy

- More general idea: Use gene expression profiles to identify disease characteristics and disease subtypes

# Example: Physiological misclassification




Data set from the exercises:
Based on the physiological properties, the sample “sick-14“ was classified as “sick“
On the gene expression level we see that it has the same profile as the “normal” samples.

# Example: Classifying samples for personalized medicine

- One aspect of pharmacogenomics is to find molecular markers that do predict whether a therapy will work or not
- Example:
  - Measure estrogen receptor status for tamoxifen (antihormone) therapy
  - Measure HER2/NEU status for herceptin therapy in breast cancer
- From the outcome of the measurement one can predict the final endpoint (which may be years away)

- In the above two examples the classification task is simple because it is already known which molecular markers do predict the treatment outcome

- If the markers are not known the challenging part is to find which of the 25000 genes are predictive

# Classification

A classifier

- is an algorithm or rule that can predict the class of a measured sample
- is always trained using data from a set of training samples with known class

Classification is also known as

- Machine learning
- Supervised classification

Mathematical formulation of the problem:

- Let x be the vector of feature values for a sample
- xi with i=1,…,p are the measurements for the individual features (for microarrays these are the expression values of the p genes)
- Assume y is the class label of the sample
- We are looking for a function that predicts the class label given the feature vector:

$$\hat{y} = C(\mathbf{x})$$

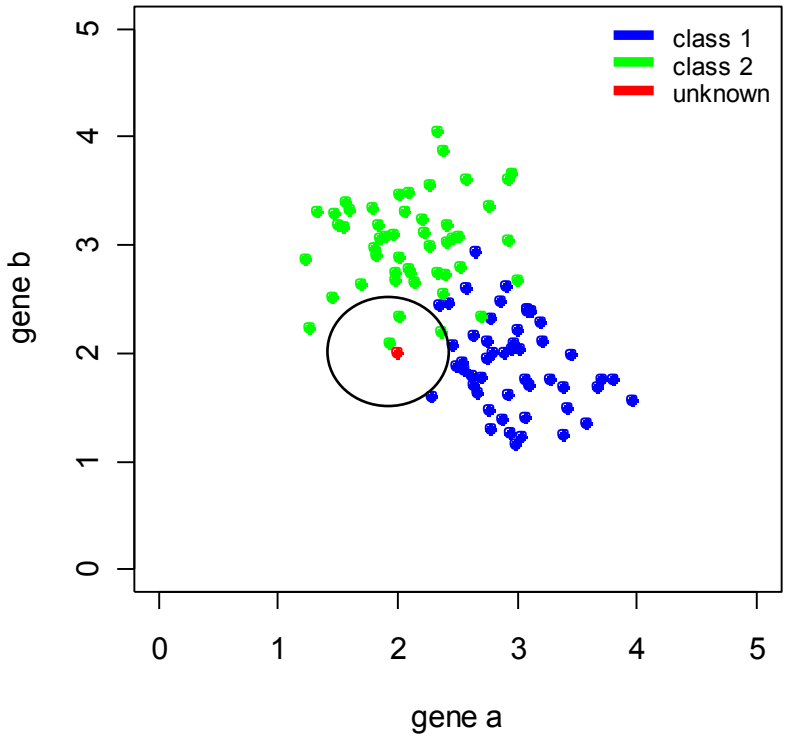
# k-nearest neighbor (KNN) classification

Example:

- Sample classification using the measurements of 2 genes
- In the plot (right) every dot is a sample
- KNN rule with k=3: Given the unknown sample (red dot), find the 3 closest neighbors
- Assign the sample per majority vote of the k neighbors to one of the classes (in our example the predicted class is “green”)

Parameters

- k: number of nearest neighbors
- Distance measure: Euclidean

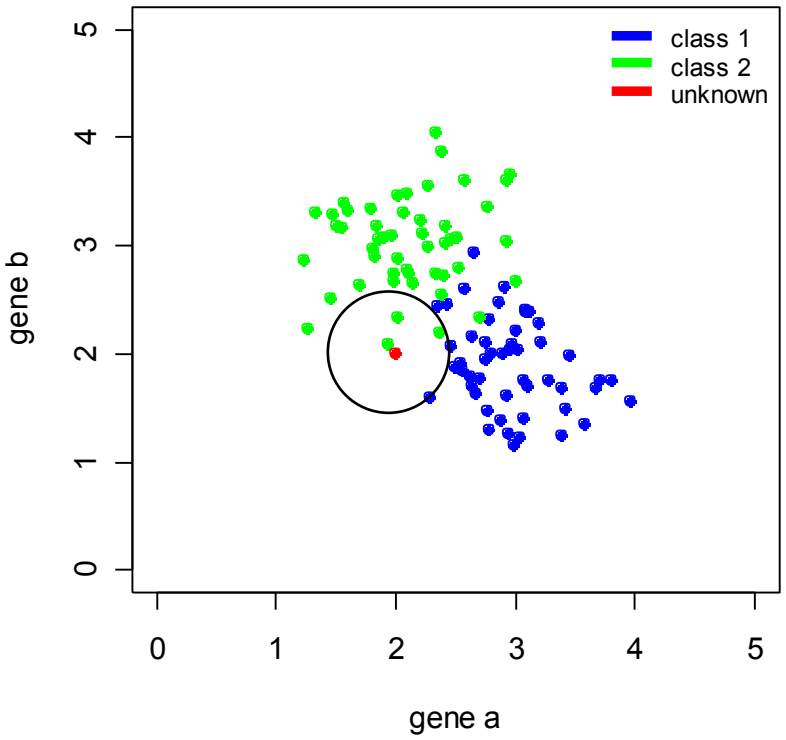


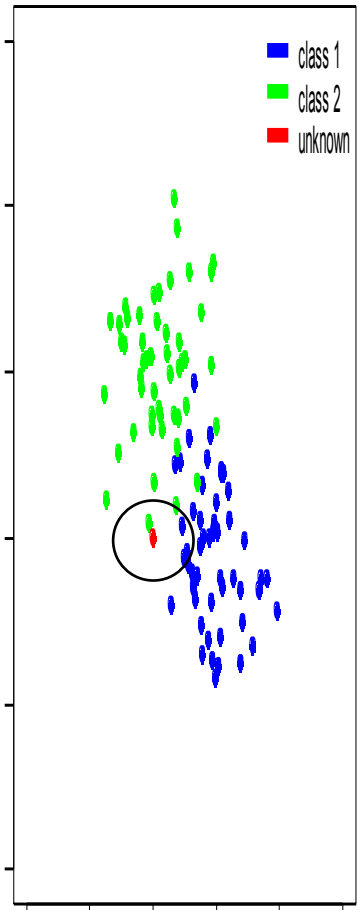

Classification using 20000 genes works accordingly

# KNN properties

- very simple algorithm
- works in general as well as more complex algorithms
- naïve implementation is computationally very intensive, (~N log N where N is the number of data points
- strictly consistent estimator (the error rate is at most twice the Bayes error rate if the number of training sample approaches infinity)
- gets worse if
  - noisy or non-relevant features are present
  - if the features are not scaled proportionally to their importance

# Why scaling matters



# Problems in classification

- Evaluating classifier performance
- Feature selection
- Optimizing classification
  - optimal method
  - optimal parameters

# Classifier performance

- Accuracy: compute the overall rate of correctly assigned samples
- Problematic
  - if performance is different for the different classes
  - if classes have very different sizes

**occurrence**

| Class A | Class B |
|---|---|
| 500 | 50 |

**correct classifications**

| Class A | Class B |
|---|---|
| 400/500 | 0/50 |

Overall accuracy: 400 / 550 = 72.7%, but the error rate for class B is 100%

Balanced accuracy: compute for each class the rate of correctly assigned samples and average the rate across all classes

- (400 / 500 + 0 / 50 )/2 = 80% + 0% = 40%

# Determining the error rate

- Cross-validation
- Subdivide the existing samples where the class is known in
  - Learning set
  - Test set
- Use the samples of the training set and define the classifier
- Classify the samples of the test set and compute the error rate

- Can be used to
- optimize parameters of the classifier
- select features to be used for classification
- select the distance measure
- ….

# Cross-validation

- K-fold cross-validation
  - Subdivide the N samples in K groups $T_1$ to $T_K$
  - For k=1, …, K
    - use the k-th group as test set
    - use all other groups as learning set to build the classifier
    - classify the members of the k-th group and determine the error rate
  - repeat with different subdivisions
  - compute the average error rate

- Variants:
  - stratified cross-validation
    - reduces the variance of the error estimate
  - leave-one-out cross-validation:
    - extreme case K=N

# Empirical and true error

- We are looking for a classifier with minimal error

$$E(C) = \sum_{y} \int \text{error}(C(\mathbf{x}), y)\, p(\mathbf{x}, y)\, d\mathbf{x}$$

- However we can only measure the empirical error based on our N samples and use this as estimator of the true error

$$\hat{E}(C) = \frac{1}{N} \sum_{n=1}^{N} \text{error}(C(\mathbf{x}_n), y_n)$$

- The empirical error is a biased estimator if several classifiers are used and one reports the empirical error of the best performing classifier.

# Precision of the empirical error

- Only probabilistic statements can be made
  Chernoff bound for fixed classifier:

$$\Pr\left(\left|\hat{E}(C) - E(C)\right| < \varepsilon\right) = 2e^{-2n\varepsilon^2}$$

Precision of error estimate depends on samples size

In order to have:

$$\Pr\left(\left|\hat{E}(C) - E(C)\right| \geq 0.1\right) = 0.05$$

one needs N=184 samples

$$\Pr\left(\left|\hat{E}(C) - E(C)\right| \geq 0.05\right) = 0.05$$

one needs N=738 samples

Many samples needed to get a precise estimate of the error

# Reported error rates are too optimistic (I)

- Cross-validation is used to find optimal classifier with optimal parameters
- Classifier has parameter a:
- Based on many cross-validation runs the optimal parameter â that minimizes the error is found:

$$\hat{y} = C(\mathbf{x};\mathbf{a})$$

$$\hat{\mathbf{a}} = \arg\min_{\mathbf{a}} Err\left(C(\mathbf{x};\mathbf{a})\right)$$

- The computed error is biased
- Since we chose the classifier with the smallest error, it is too optimistic
- When applying to real-world unknown samples the observed error will be larger

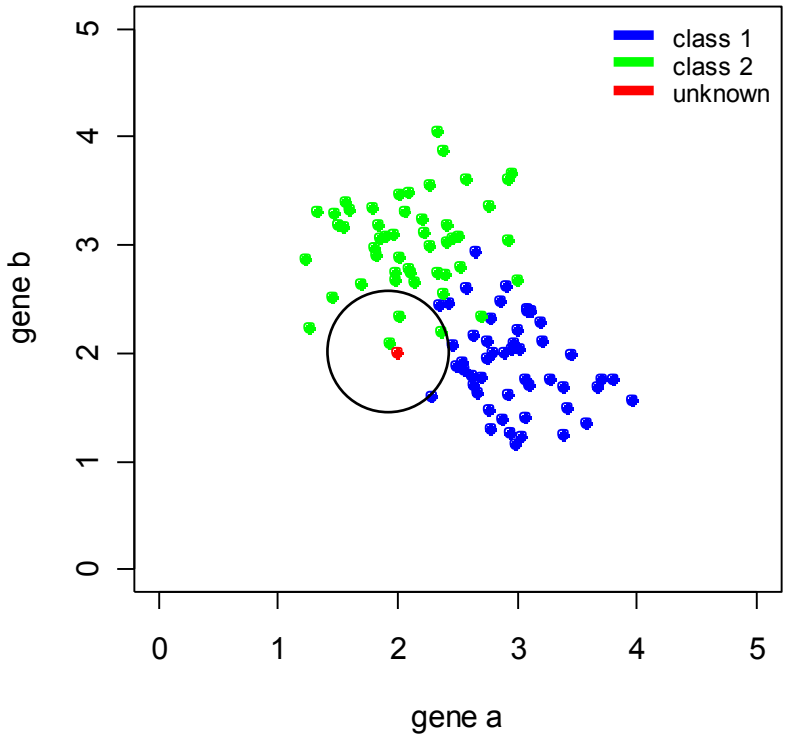
# Reported error rates are too optimistic (II)

- Used set of samples is not representative
- Example: Microarray data
  - microarrays frequently have a batch effect
  - but usually all sample come from one batch (e.g. 6-month study in a single hospital)
- Classifier is optimized to classify samples of this batch correctly
- Error rates when classifying samples in a different study from a different hospital will be higher than predicted

# Feature selection

Example:

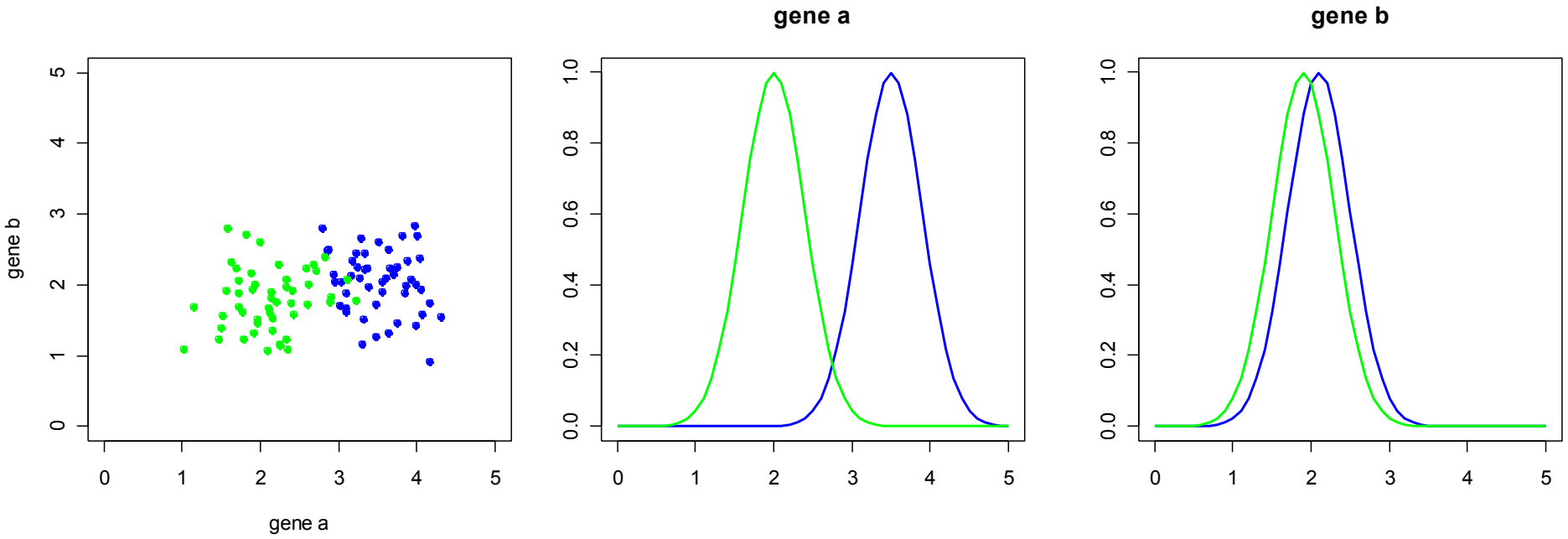
- Classification with 2 features (2 genes)
- 2-dimensional feature space
- Microarray measures 20 000 features
- More genes
  → more information on the samples
  → better classification?
  Not really




- Additional features improve the classification if and only if they provide additional information on class membership

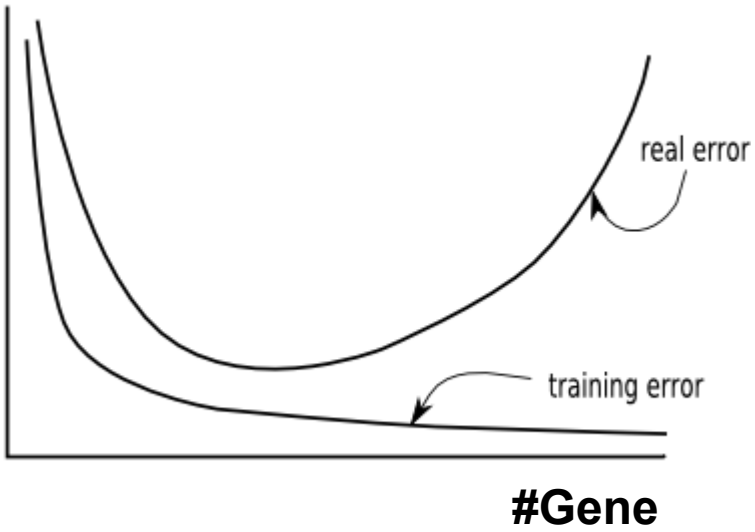
# Feature selection

## Example:




**informative** **non-informative**

# Feature selection




The training error decreases with increasing number of genes but the real error (when classifying new samples) does increase
→ Overfitting

- Why is it not better to have more genes?
- Curse of dimensionality (Richard Bellman)

- Further disadvantages of many genes:
- Learning is slower
- Model is not interpretable

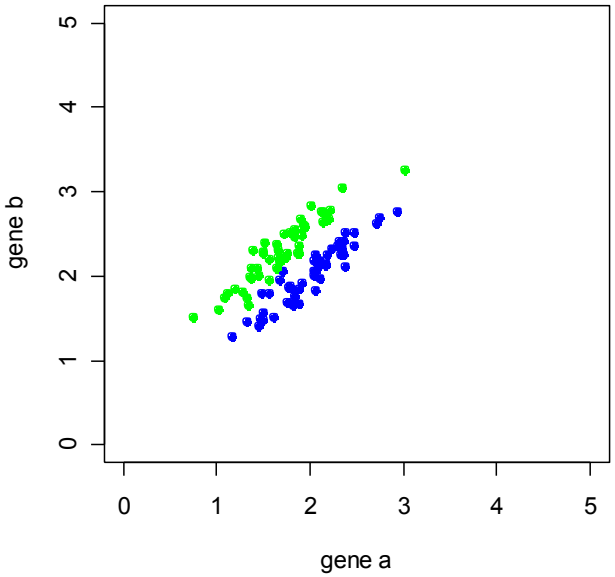
# Feature selection: Methods

- Discriminant analysis:
- later …

- Filtering based on information content of genes
- run t-test, ANOVA, … and choose the m best genes
- Problems
  - filtering methods work on individual genes
  - significance does not imply predictive power
  - synthetically informative genes are discarded
  - correlated informative genes are kept
- Alternative Methods
- Nearest Shrunken Centroids (Tibshirani etal. 2002)
- Gene shaving (Hastie etal. 2000)
- Elimination of correlated genes (Yeung and Bumgarner 2003)

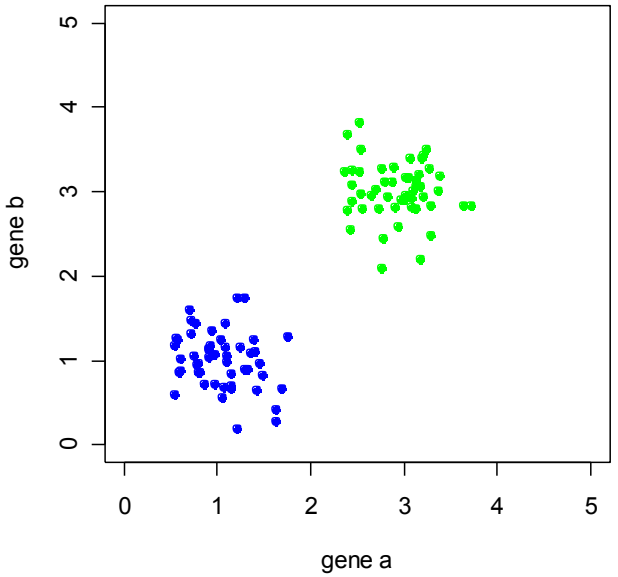
# Example: Gene pairs

## synthetically informative




Each gene alone cannot discriminate between the two classes
Both genes together allow class discrimination

## correlated informative




Each gene alone does already discriminate the classes
Using both genes is no advantage

# Distance measure

- Choosing an appropriate distance measure is key to successful classification
- Distance between two samples may be:
  - correlation of gene expression vectors
  - Euclidean distance of gene expression vectors

- Expression values have to be transformed, e.g. log-transformation, such that
  - high expressed genes (signal 20 000 – 40 000)
  - low expressed genes (signal 200 – 400)
  - contribute with similar weights to the classification

# Classification algorithms

- K-NN: introduced earlier
- Discriminant analysis
  Model probability densities within the classes
  - Linear discriminant analysis
  - Quadratic discriminant analysis
  - Fisher's discriminant analysis
  - Bayesian discriminant analysis

- Support Vector Machines:
  Model class boundaries
  - linear kernel functions
  - quadratic kernel functions
- Top-scoring Pairs:
  Find pairs of genes whose ordering is inversed in two classes

# Discriminant analysis

- Starting point:
  The p-dimensional feature vectors x are multivariate normally distributed given the class k:

$$\Pr[\mathbf{x} \mid k] = \frac{1}{\sqrt{(2\pi)^p |\boldsymbol{\Sigma}_k|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)}$$

# Linear Discriminant Analysis

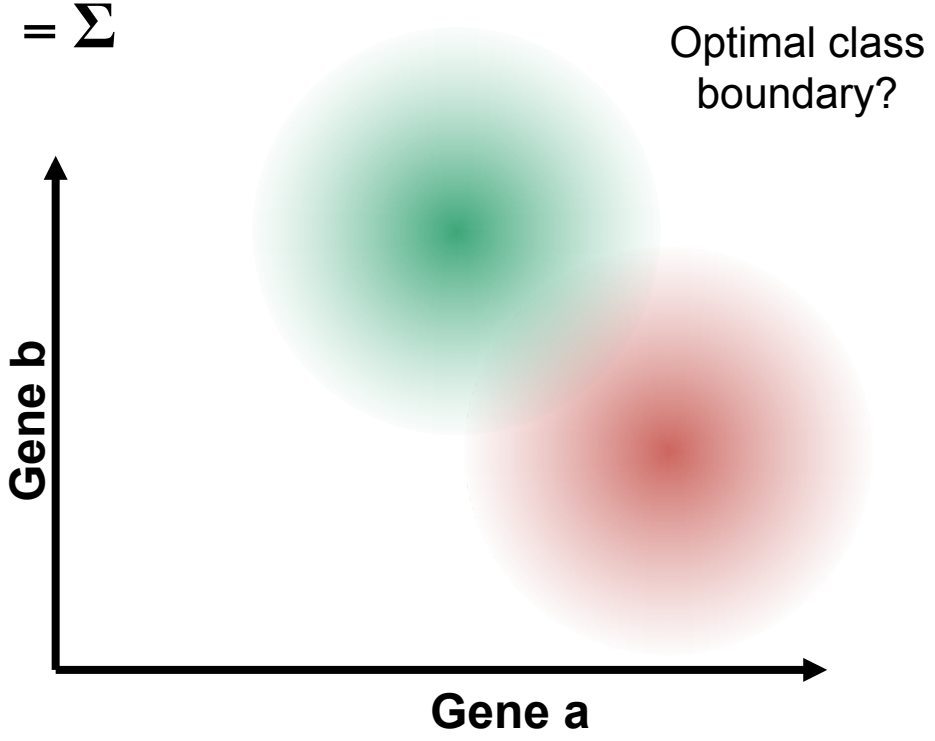
- Assume each class has the same covariance matrix:

$$\Sigma_k = \Sigma$$

- Example
  - 2 classes: k=1,2
  - 2 genes: a, b

- Classification rule:

$$\hat{k} = \arg\max_k \Pr[\mathbf{x} \mid k]$$

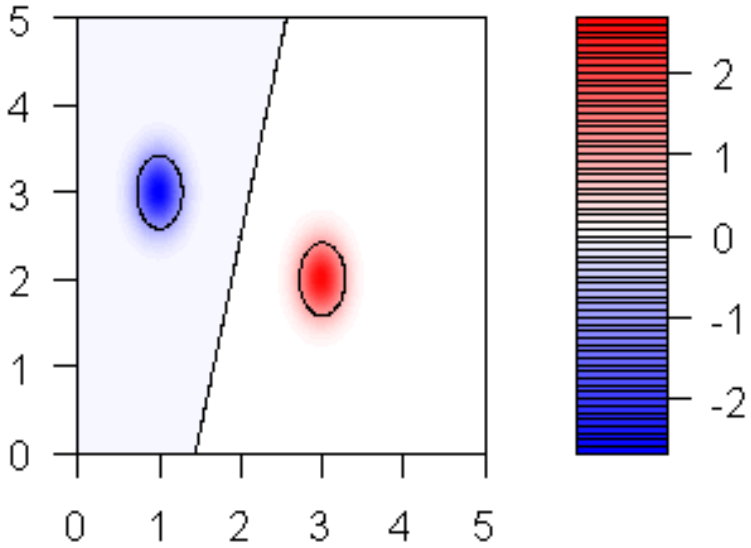


# Linear Discriminant Analysis

- Visualize:

$$\Pr[\mathbf{x} \mid k = 1] - \Pr[\mathbf{x} \mid k = 2]$$

- Class boundary where the probabilitie straight line
- In higher dimensions (more genes) th
- In order to classify an unknown samp side of the boundary it is located



# Linear Discriminant Analysis

- Compute the likelihood ratio:

$$\log \frac{\Pr[\mathbf{x} \mid k=1]}{\Pr[\mathbf{x} \mid k=2]} = \frac{1}{2}(\mathbf{x}-\mu_2)^T \Sigma^{-1}(\mathbf{x}-\mu_2) - \frac{1}{2}(\mathbf{x}-\mu_1)^T \Sigma^{-1}(\mathbf{x}-\mu_1)$$

$$= \mathbf{x}^T \Sigma^{-1}(\mu_1 - \mu_2) + C$$

$$= \mathbf{x}^T \mathbf{w} + C$$

$$\mathbf{w} = \Sigma^{-1}(\mu_1 - \mu_2)$$

Classification only depends on a scalar product $\mathbf{x}^T \mathbf{w}$

Linear in **x**

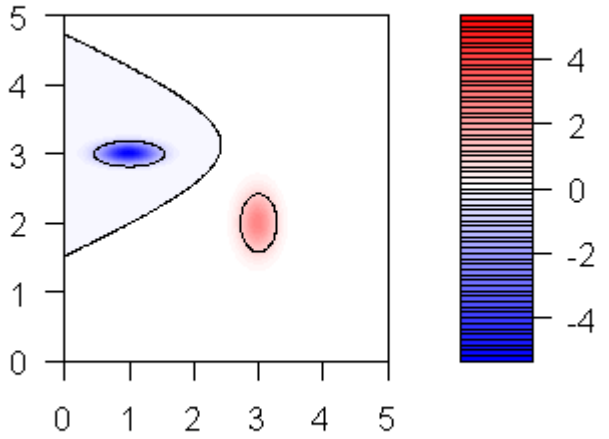
# Quadratic Discriminant Analysis

- Assume different covariance matrices

$$\Sigma_1^{-1} \neq \Sigma_2^{-1}$$

$$\log \frac{\Pr[\mathbf{x}\,|\,1]}{\Pr[\mathbf{x}\,|\,2]} = \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_2)^T \boldsymbol{\Sigma}_1^{-1}(\mathbf{x}-\boldsymbol{\mu}_2) - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}_2^{-1}(\mathbf{x}-\boldsymbol{\mu}_1)$$

- No simplification possible
- Class boundary is quadratic



# Fisher's Discriminant Analysis

- Similar to linear discriminant analysis, but different approach
- Consider linear combination of features (scalar product) : $z = \mathbf{x}^T \mathbf{w}$
- Average in class k: $E[z \mid k] = \boldsymbol{\mu}_k^T \mathbf{w}$
- Variance in class k: $\mathrm{Var}[z \mid k] = \mathbf{w}^T \boldsymbol{\Sigma}_k \mathbf{w}$
- Empirical definition of separability of two classes

$$S = \frac{\sigma^2_{\text{between}}}{\sigma^2_{\text{within}}} = \frac{(\mathbf{w}^T \boldsymbol{\mu}_2 - \mathbf{w}^T \boldsymbol{\mu}_1)^2}{\mathbf{w}^T \boldsymbol{\Sigma}_1 \mathbf{w} + \mathbf{w}^T \boldsymbol{\Sigma}_2 \mathbf{w}}$$

- Is maximized by

$$\mathbf{w} = (\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)^{-1} (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$$

# Fisher's Discriminant Analysis

- Is equivalent to linear discriminant analysis, if covariance matrices are identical
- Separabiltiy
  - is similar to F-statistic of the ANOVA
  - can be interpreted as signal-to-noise ratio for the class labels
- can be directly extended to more than two classes

- Finds a linear combination of features that provide good separability
  → Can be used for feature selection

# Bayesian Discriminant Analysis

- As an extension if the classes have different sizes (different prior probabilities)
- Maximization of the a posteriori probability:

$$\hat{k} = \arg\max_{k'} \Pr[k' | \mathbf{x}]$$

$$= \arg\max_{k'} \frac{\Pr[\mathbf{x} | k']\pi(k')}{\sum_g \Pr[\mathbf{x} | g]\pi(g)}$$

Additional consideration of the costs of misclassifications:

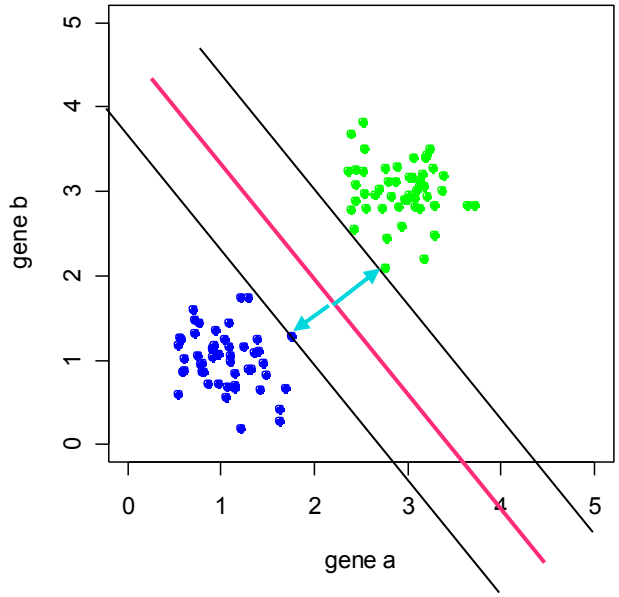
$$\hat{k} = \arg\min_{k'} \sum_g c_{k'g} \Pr[g | \mathbf{x}]$$

$c_{k'g}$ cost if a member of class g is wrongly assigned to class k'

# Support Vector Machines

- Consider the boundary of two classes
- Find the optimal boundary (in the p-dimensional feature space: Hyperplane)
- A plane is optimal if it maximizes the margins

- The boundary is only defined by the points at the border line (support vectors)



# Support Vector Machines

- Classification function:
  classes: yi=±1

$$y_i = \mathrm{sgn}(\mathbf{w}\mathbf{x}_i + b)$$

- For linear separable data:
  Optimization with boundary conditions
  using the Lagrange function:

$$\frac{1}{2}\| w \|^2 - \sum_i \alpha_i \left( y_i (\mathbf{w}\mathbf{x}_i + b) - 1 \right)$$

  Lagrange multiplier $\alpha_i$

- Classification function given by:

$$y = \mathrm{sgn}\left( \sum_i \alpha_i y_i \, \mathbf{x}_i \mathbf{x} + b \right)$$

# Support Vector Machines

Extension:

- Not linearly separable samples
  Introduction of a slack variable that allows misclassifications at additional costs

- Kernel-Trick:
  Map the data in to a space with higher dimensionality with the hope that the data are linearly separable in that space

# SVM: Kernel Trick

Example:
Function h maps from the 2- to the 3-dim. space:

$$h(\mathbf{x}) = h\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{pmatrix}$$

Classification requires the computation of the scalar product. But in order to compute the scalar product in the new space, the data do not need to be transformed.

Scalar products can be computed using a kernel function:

$$h(\mathbf{x})h(\mathbf{x}') = (\mathbf{x}\mathbf{x}')^2 = k(\mathbf{x}, \mathbf{x}')$$

Computing the scalar product is efficient

SVM classification only needs scalar products

$$y = \operatorname{sgn}\left(\sum_i \alpha_i y_i \, k(\mathbf{x}_i, \mathbf{x}) + b\right)$$

# SVM: Typical Kernel-Functions

- linear kernel

$$k(\mathbf{x}, \mathbf{x}') = \mathbf{x}\mathbf{x}'$$

- polynomial kernel

$$k(\mathbf{x}, \mathbf{x}') = (\mathbf{x}\mathbf{x}')^d$$

- Radial-Basis-Function kernel

$$k(\mathbf{x}, \mathbf{x}') = \exp(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2})$$

# Top Scoring Pairs (TSP)

Find pairs of genes whose ordering is inversed in two classes

Properties

- Invariance to quantitation and normalization
- Easy to compute
- Simple decisions rules: “If gene i has higher expression than gene j then it is cancer otherwise it is normal”.
- Decision rule has also an interpretation: Gene i is related to cancer!

## Example

| | Cancer | Normal |
|---|---|---|
| $X_1 < X_2$ | 11 | 2 |
| $X_1 \geq X_2$ | 3 | 15 |

| | Cancer | Normal |
|---|---|---|
| $X_3 < X_4$ | 7 | 10 |
| $X_3 \geq X_4$ | 7 | 7 |

*Conclusion*: We prefer pair $(1, 2)$ to pair $(3, 4)$

Adapted from D. Geman

# TSP: Score Definition

$$\Delta_{ij} = \left| P(X_i < X_j | Y = 1) - P(X_i < X_j | Y = 2) \right|$$

$$\approx \left| \frac{N_{ij}^{(1)}}{n_1} - \frac{N_{ij}^{(2)}}{n_2} \right|$$

where

$$N_{ij}^{(k)} = \left| \{ 1 \le m \le n : Y_m = k,\ X_{im} < X_{jm} \} \right|,\ k = 1, 2$$

Adapted from D. Geman

# TSP: References

- References:
  - Algorithm:
    "Classifying gene expression profiles from pairwise mRNA comparisons," Stat. Appl. in Genetics and Molecular Biology, 3, 2004.
  - Application:
    "Simple decision rules for classifying human cancers from gene expression profiles," Bioinformatics, 21, 3896-3904, 2005
    "Robust prostate cancer marker genes discovered from direct integration of inter-study microarray data," 21, 3905-3911, Bioinformatics, 2005.

# More classification methods

- LASSO
- Classification and Regression Trees (CART)
- Bagging: Aggregation of Classifiers by majority vote
- Boosting: Linear Combinations of Classifier outcomes
- Example
  - Random Forests -- Aggregation of Decision Trees (Breiman, 1996, 1998)
-
- Not considered:
- What if sample represents a new unknown class?
- What if sample belongs to more than one class?

# Classification in R

- Existing R packages
  - MASS: Linear und quadratic discriminant analysis
  - sma: Diagonal LDA
  - class: k-nearest neighbor
  - tspair: top-scoring pairs classifier
  - rpart: classification and regression trees (CART)
  - ipred: bagging
  - e1071: SVM
  - LogitBoost: boosting
  - MLInterfaces: defines consistent interfaces to many of the above classifiers
  - ….
- See also:
  - http://cran.r-project.org/web/views/MachineLearning.html

# Summary

- Molecular profiling data provides an insight into intracellular mechanisms that can be used for classification
- Classification with many features and a few samples is problematic
- If classifier function is optimized this leads to over-optimistic error rates
- In many cases the simple classifiers, TSP and KNN, provide good performance

# References


- Duda, R. O. and Hart, P. E.
  **Pattern Classification and Scene Analysis.**
  Wiley, 1973
- Richard O. Duda, Peter E. Hart & David G. Stork
  **Pattern Classification.**
  Wiley & Sons, 2001
- T. Hastie, R. Tibshirani, and J. Friedman
  **The elements of statistical learning.**
  Springer Series in Statistics,
  Springer-Verlag, New York, 2001
- S. Dudoit, J. Fridlyand, and T. P. Speed (2002).
  **Comparison of discrimination methods for the classification of tumors using gene expression data.**
  Journal of the American Statistical Association, Vol. 97, No. 457, p. 77-87
- L. Breiman, J.H. Friedman, R.A. Olshen, et al.
  **Classification and Regression Trees.**
  Wadsworth, Belmont, CA, 1984