Clustering and Discrimination

- cluster analysis – does not know the # of groups in advance but wishes to establish groups and then analyze group membership.
- Discriminant function analysis – analyzes group membership for known groups (pre-specified).
- They differ according to their aims, which in turn depend on the availability of a pre-existing basis for the grouping.
- Alternative terminology
  - Computer science: **unsupervised** and **supervised learning.**
  - Microarray literature: **class discovery** and **class prediction.**

Eg: Tumor classification

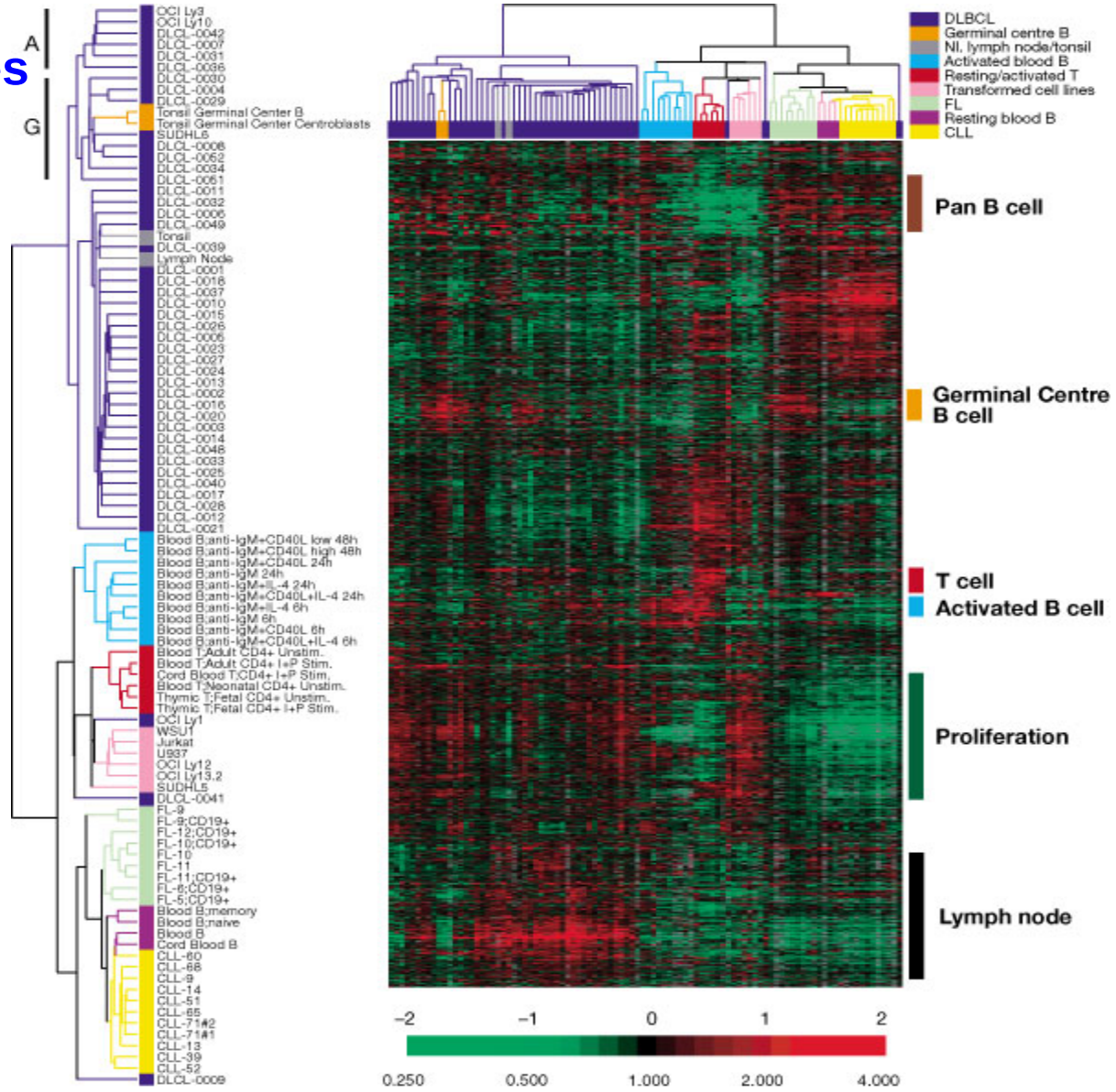
A reliable and precise classification of tumors is essential for successful diagnosis and treatment of cancer.

Current methods for classifying human malignancies rely on a variety of morphological, clinical, and molecular variables.

In spite of recent progress, there are still uncertainties in diagnosis. Also, it is likely that the existing classes are heterogeneous.

DNA microarrays may be used to characterize the molecular variations among tumors by monitoring gene expression on a genomic scale. This may lead to a more reliable classification of tumors.

Clusters on both genes and arrays



Nature February, 2000 Paper by Allzadeh. A *et al Distinct types of diffuse large B-cell lymphoma identified by Gene expression profiling*

Clustering microarray data

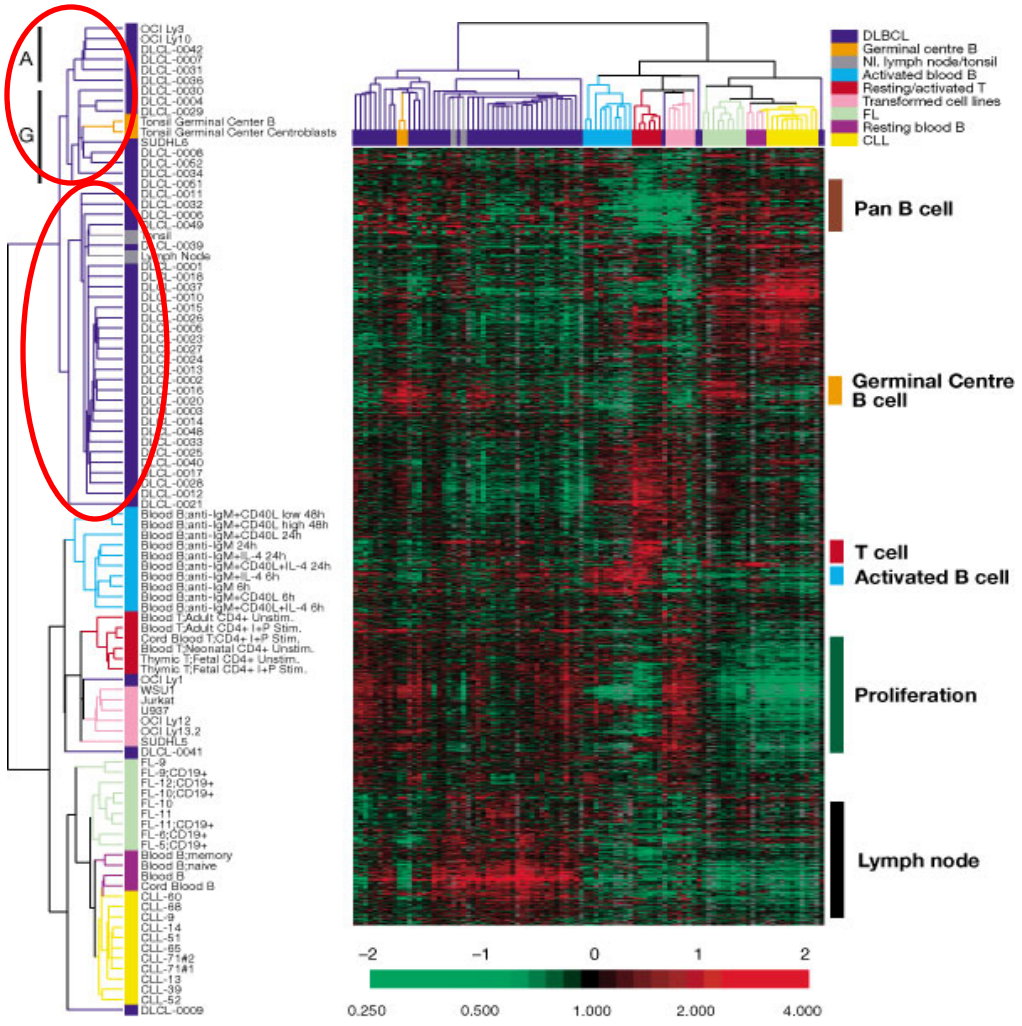
We can cluster genes (rows), mRNA samples (cols), or both at once.

- **readily interpretable figures.**
- **identifying patterns in time or space.**
- **seeking new subclasses of cell samples (tumors, etc).**

Alizadeh et al (2000) *Distinct types of diffuse large B-cell lymphoma*

- Three subtypes of lymphoma (FL, CLL and DLBCL) have different genetic signatures. (81 cases total)

- DLBCL group can be partitioned into two subgroups with significantly different survival. (39 DLBCL cases)



Basic principles of clustering

Aim: **to group observations that are "similar" based on predefined criteria.**

Issues: Which genes / arrays to use?
Which similarity or dissimilarity measure?
Which clustering algorithm?

- It is advisable to **reduce** the number of genes from the full set to some more manageable number, before clustering. The basis for this reduction is usually quite context specific, see LDA later (another popular one is PCA/SVD).

Minkowski metric, with EUC and MAN as special cases:

$$z_k = d_k(x_k, y_k) = |x_k - y_k| \text{ and } F(z_1, \ldots, z_m) = \left(\sum_{k=1}^{m} z_k^{\lambda}\right)^{1/\lambda}.$$

EUC Euclidean metric

$$d_{euc}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2}.$$

MAN Manhattan metric

$$d_{man}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{m} |x_i - y_i|.$$

Two main Classes

Distance

- **Manhattan**
- **Euclidean**
- **Mahalanobis distance**
- **Many more ….**

Correlation

Correlation-based distance measures:

COR Pearson sample correlation distance

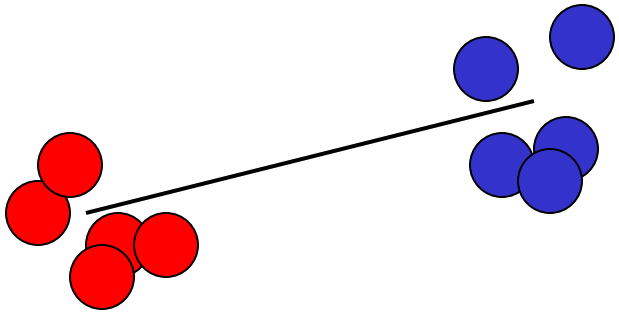
$$d_{cor}(\mathbf{x}, \mathbf{y}) = 1 - r(\mathbf{x}, \mathbf{y}) = 1 - \frac{\sum_{i=1}^{m} (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{m} (x_i - \overline{x})^2 \sum_{i=1}^{m} (y_i - \overline{y})^2}}.$$

EISEN Cosine correlation distance

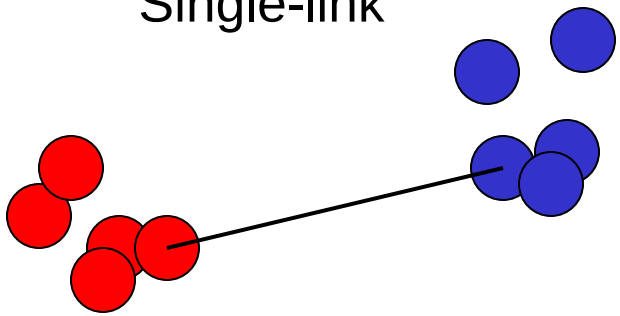
$$d_{eisen}(\mathbf{x}, \mathbf{y}) = 1 - \frac{\mathbf{x}'\mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|} = 1 - \frac{\left|\sum_{i=1}^{m} x_i y_i\right|}{\sqrt{\sum_{i=1}^{m} x_i^2 \sum_{i=1}^{m} y_i^2}}$$

Distance between Clusters

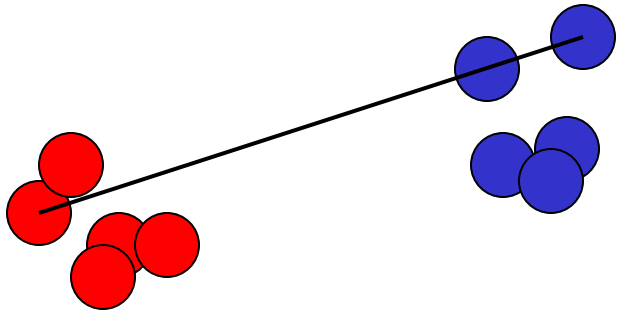
Distance between centroids



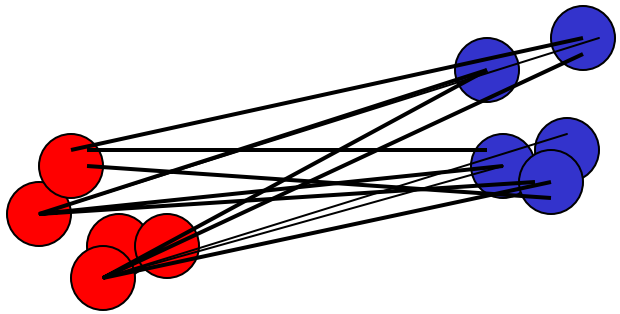
Single-link



Complete-link

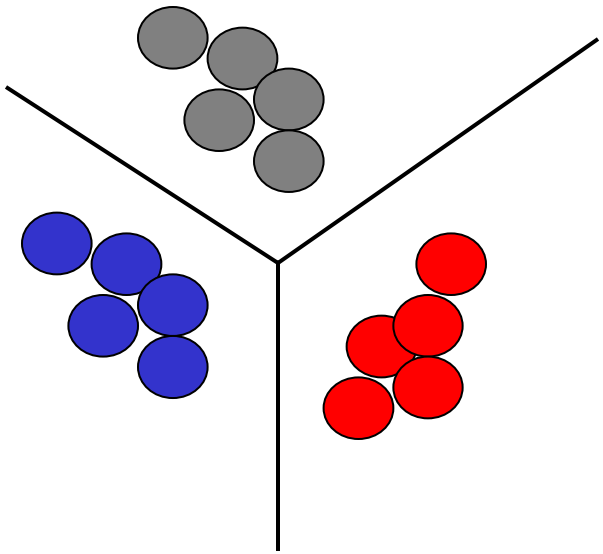


Mean-link (Average linkage/ **UPGMA**)

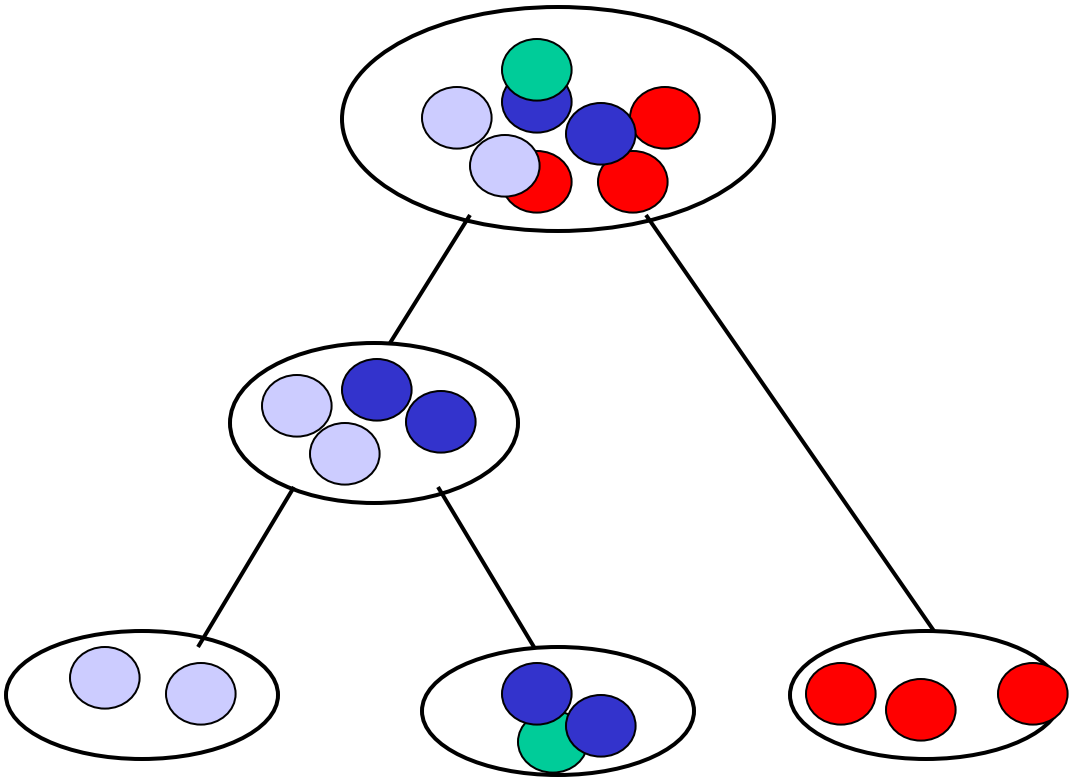


Two basic types of clustering methods

Partitioning



Hierarchical



Partitioning methods

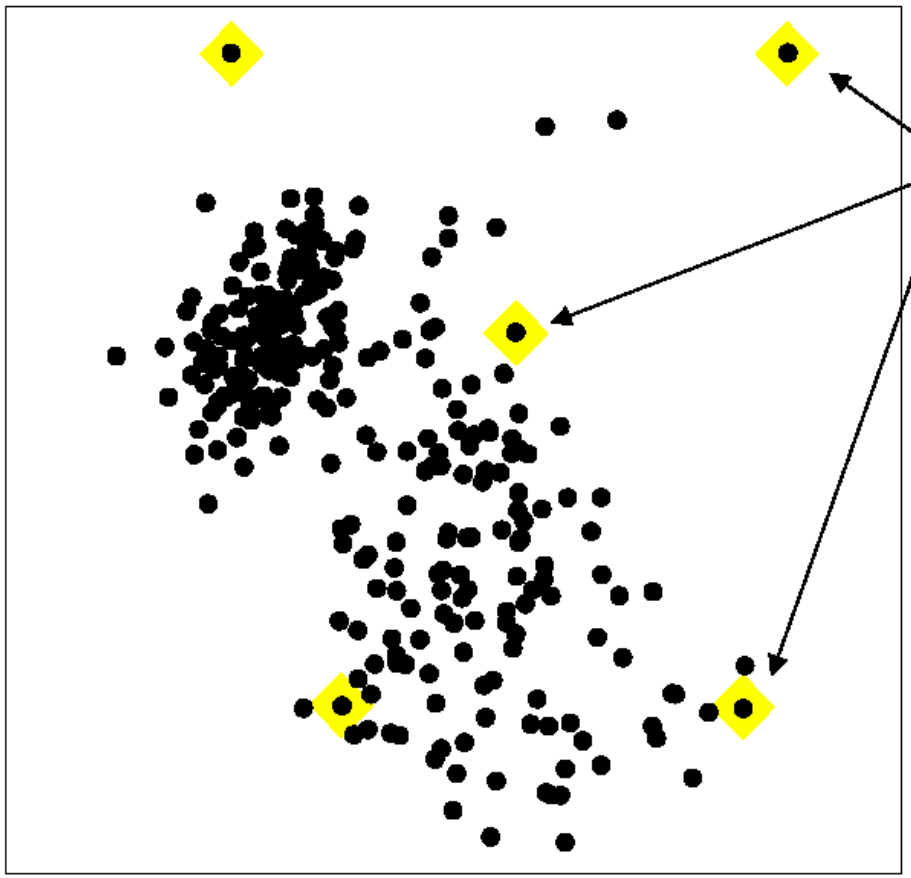
Partition the data into a prespecified number *k* of mutually exclusive and exhaustive groups.

Iteratively reallocate the observations to clusters until some criterion is met, e.g. minimize within cluster sums of squares.

Examples:

- *k*-means, self-organizing maps (SOM), *PAM*, etc.;
- Fuzzy: needs stochastic model, e.g. Gaussian mixtures.

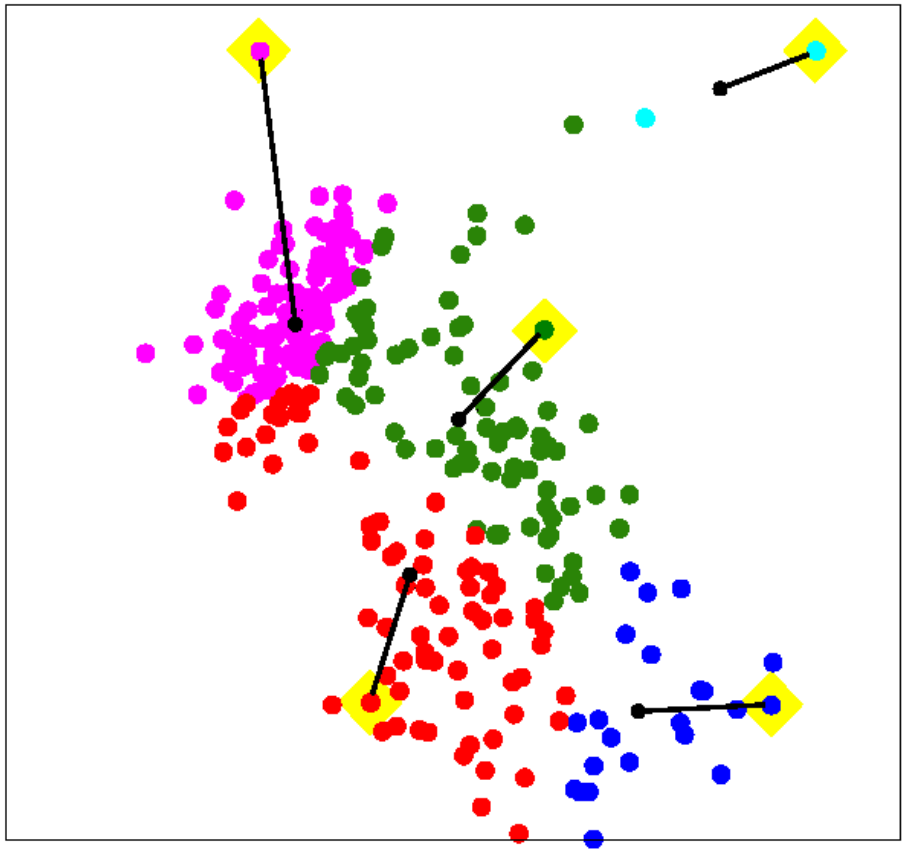
K-Means Clustering



Select k initial seed

K is pre-determined

K-Means Clustering



Step1: For each data point, assign cluster by selecting data points that are closest to the seed

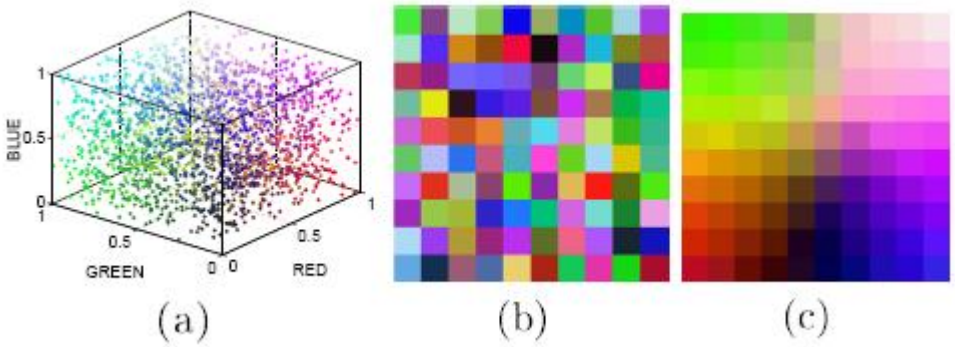
Step2: Calculate cluster mean

Step3: Seed is changed to the mean of the cluster

Step4: Repeat until Seeds don't change

Self Organizing Maps (SOM)

- 2-D square grid (for microarrays) or 1-D of nodes.
- Inputs are n-dimensional vectors (e.g, <255,0,0> for red)
- SOM converges so that similar features (e.g; colors) are grouped together.



a) Input space
b) Initial weights
c) Final weights

The Basic Process

1) Initialize each node's weights.

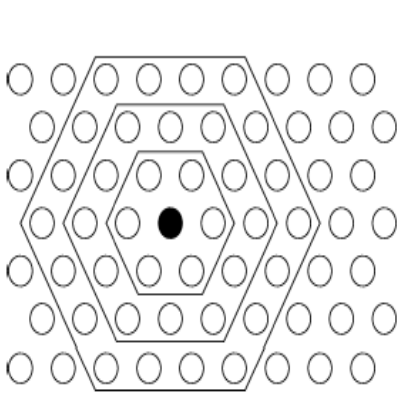
2) Choose a random vector from training data and present it to the SOM.

3) Every node is examined to find the Best Matching Unit (BMU).

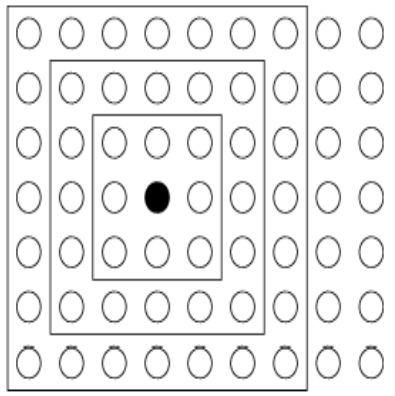
4) The radius of the neighborhood around the BMU is calculated. The size of the neighborhood decreases with each iteration.

5) Each node in the BMU's neighborhood has its weights adjusted to become more like the BMU. Nodes closest to the BMU are altered more than the nodes furthest away in the neighborhood.

6) Repeat from step 2 for enough iterations for convergence.



(a) Hexagonal grid



(b) Rectangular grid

Figure . : In the 2-dimensional case the neurons of the map can be arranged either on a rectangular or hexagonal lattice. Neighborhoods (size 1, 2 and 3) of the unit marked with black dot.

Calculating the Best Matching Unit (BMU)

- Calculating the BMU is done according to the Euclidean distance among the node's weights (W_1, W_2, … , W_n) and the input vector's values (V_1, V_2, … , V_n).

$$Dist = \sqrt{\sum_{i=1}^{n} (V_i - W_i)^2}$$

Determining the BMU Neighborhood

- **Size of the neighborhood: an *exponential decay* function that shrinks on each iteration until eventually the neighborhood is just the BMU itself.**

$$\sigma(t) = \sigma_0(t)\exp(-\frac{t}{\lambda})$$

- **Effect of location within the neighborhood: so that nodes that are closer are influenced more than farther nodes.**

$$\Theta(t) = \exp(-\frac{dist^2}{2\sigma^2(t)})$$

Modifying Weights of Nodes

- The new weight for a node is the old weight, plus a fraction (L) of the difference between the old weight and the input vector… adjusted (theta) based on distance from the BMU.

$$W(t+1) = W(t) + \Theta(t).L(t).(V(t) - W(t))$$

- The learning rate, L, is also an exponential *decay* function.
  - This ensures that the SOM will converge.

$$L(t) = L_0 \exp(-\frac{t}{\lambda})$$

- The lambda represents a time constant, and t is the time step

Hierarchical methods – we already saw this in phylogeny

Hierarchical clustering methods produce a tree or dendrogram.

They avoid specifying how many clusters are appropriate by providing a partition for each *k* obtained from cutting the tree at some level.

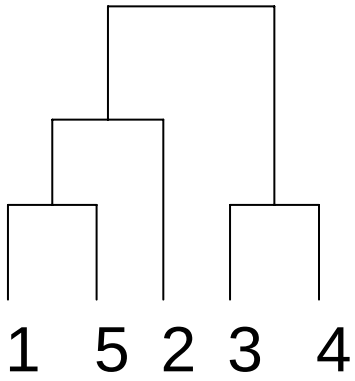
The tree can be built in two distinct ways bottom-up: agglomerative clustering;top-down: divisive clustering.

Agglomerative methods

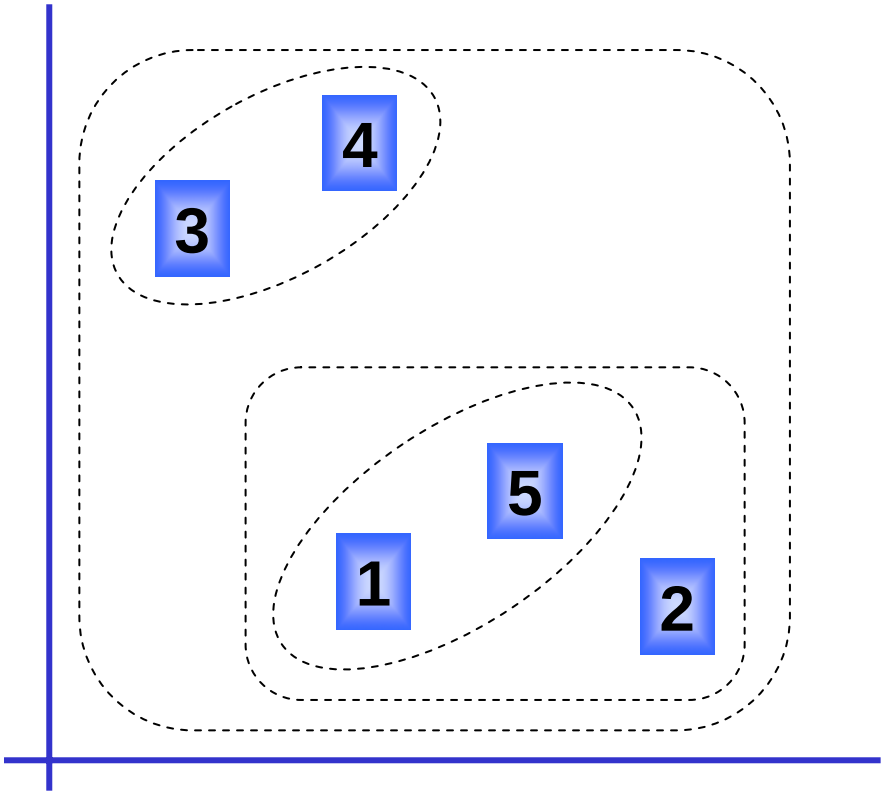
- Start with *n* clusters.

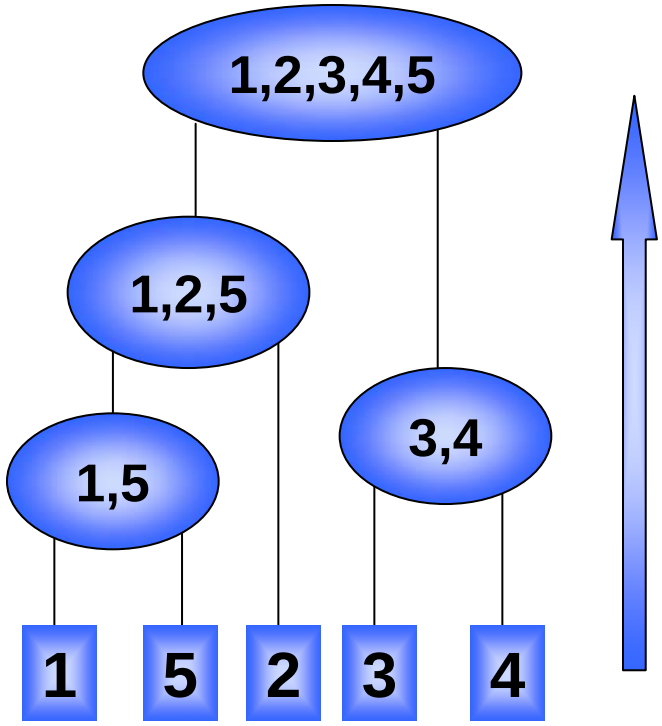
  At each step, merge the two closest clusters using a measure of between-cluster dissimilarity, which reflects the shape of the clusters.
- Between-cluster dissimilarity measures
  - Mean-link: average of pairwise dissimilarities
  - Single-link: minimum of pairwise dissimilarities - Avoid.
  - Complete-link: maximum& of pairwise dissimilarities – pretty good.
  - Distance between centroids

Illustration of points In two dimensional space



Agglomerative



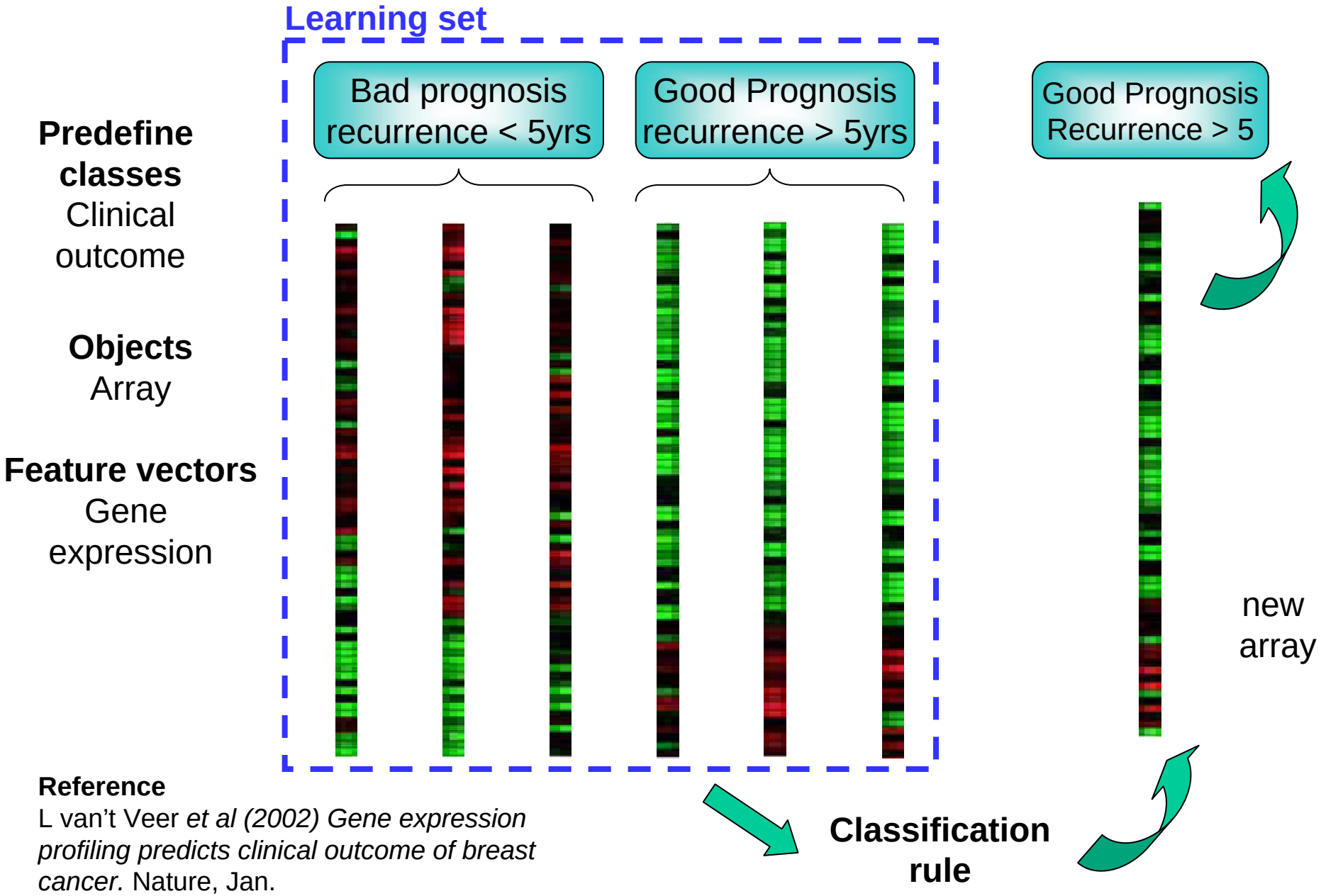


Divisive methods

- Start with only one cluster.
- At each step, split clusters into two parts.
- Split to give greatest distance between two new clusters
- *Advantages.*
    - Obtain the main structure of the data, i.e. focus on upper levels of dendogram.
- *Disadvantages*.
  - Computational difficulties when considering all possible divisions into two groups.

Hybrid Methods

- Mix elements of Partitioning and Hierarchical methods
  - Bagging
    - Dudoit & Fridlyand (2002)
  - HOPACH
    - van der Laan & Pollard (2001)



Reference

L van't Veer *et al (2002) Gene expression profiling predicts clinical outcome of breast cancer.* Nature, Jan.

.

Intro to Decision Theory

- classification as statistical decision theory: must decide which of the classes an object belongs to
- Use the observed feature vector **X** to aid in decision making
- Denote population proportion of objects of class k as $p_k = p(Y = k)$
- Assume objects in class k have feature vectors with class conditional density $p_k(\mathbf{X}) = p(\mathbf{X}|Y = k)$

Intro to Decision Theory – Cont'd

- One criterion for assessing classifier quality is the misclassification rate,

$$p(C(\mathbf{X}) \neq Y)$$

- A loss function L(i,j) quantifies the loss incurred by erroneously classifying a member of class i as class j
- The risk function R(C) for a classifier is the expected (average) loss:

$$R(C) = E[L(Y, C(\mathbf{X}))]$$

Overview of popular classifiers – Bayes and Naïve Bayes

In the unlikely situation that the class conditional densities $p_k(\mathbf{x}) = p(\mathbf{x}|Y = k)$ and class priors π_k are known, let

$$p(k \mid \mathbf{x}) = \frac{\pi_k p_k(\mathbf{x})}{\sum_l \pi_l p_l(\mathbf{x})}$$

Can simplify by Naïve Bayes

denote the **posterior probability** of class k given feature vector $\mathbf{x}$.

The **Bayes rule** predicts the class of an observation $\mathbf{x}$ by that with highest posterior probability

$$C_B(\mathbf{x}) = \text{argmax}_k \; p(k \mid \mathbf{x}).$$

The Bayes rule minimizes the total risk under a symmetric loss function – **Bayes risk**.

$$\text{argmin}_l \left(\sum_{h=1}^{K} L(h,l)\, p(h \mid x)\right) - \textit{BayesRisk}$$

Classifiers- Cont'd
Maximum likelihood discriminant rule

- A **maximum likelihood estimator (MLE)** chooses the parameter value that makes the chance of the observations the highest.

- For known class conditional densities $p_k(\mathbf{X})$, the **maximum likelihood (ML) discriminant rule** predicts the class of an observation **X** by

$$C(\mathbf{X}) = \text{argmax}_k \, p_k(\mathbf{X}) = \text{argmax}_k \, p(\mathbf{X}|Y = k)$$

Compared to $argmax_k \, p(k \mid X)$ *for Bayes*

Gaussian ML discriminant

- **For multivariate Gaussian (normal) class densities X|Y= k ~ $N(\mu_k, \Sigma_k)$, the ML classifier is**

$$C(\mathbf{X}) = \text{argmin}_k \{(\mathbf{X} - \mu_k)\, \Sigma_k^{-1} (\mathbf{X} - \mu_k)' + \log|\, \Sigma_k \,|\}$$

$$\Sigma_k = \frac{1}{n}\sum_{i=1}^{n}(X_i - \overline{X})(X_i - \overline{X})^T \Big|_k$$

- **In practice, population mean vectors μ_k and covariance matrices Σ_k are estimated by corresponding sample quantities**
- **This is a Quadratic discriminant analysis, or QDA**
- **If class densities have same diagonal covariance matrix ie,**

$$\Delta_k = diag(\sigma_{k1}^2, ...\sigma_{kp}^2),\ \text{C(X)=argmin}_k \left\{ \sum_{j=1}^{p} \frac{(x_j - \mu_{kj})^2}{\sigma_{kj}^2} + \log \sigma_{kj}^2 \right\} \rightarrow \text{DQDA}$$

ML discriminant rules - special cases

1. **Linear discriminant analysis, LDA.** When the class densities have the same covariance matrix, $\Sigma_k \equiv \Sigma$, the discriminant rule is based on the square of the Mahalanobis distance and is linear and given by

$$C(x) = \arg\min_k (x-\mu)'\Sigma^{-1}(x-\mu)$$

2. **Diagonal linear discriminant analysis, DLDA.** In this simplest case, when the class densities have the same diagonal covariance matrix $\nabla = \mathrm{diag}(s_1^2, \ldots, s_p^2)$

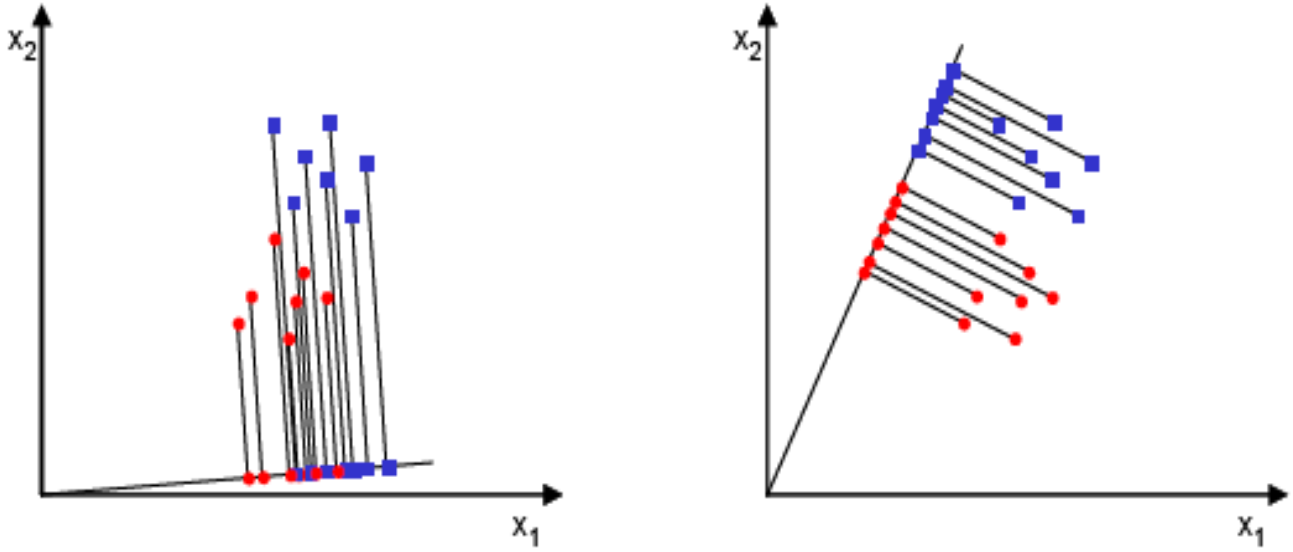
$$C(x) = \arg\min_k \sum_{j=1}^{p} \frac{(x_j - \mu_{kj})^2}{\sigma_j^2}$$

Fisher's linear Discriminant Analysis (FLDA)

- **The objective of LDA is to perform dimensionality reduction while preserving as much of the class discriminatory information as possible**
  - Assume we have a set of D-dimensional samples $\{x^{(1}, x^{(2}, \ldots, x^{(N}\}$, N_1 of which belong to class ω_1, and N_2 to class ω_2. We seek to obtain a scalar $\boldsymbol{y}$ by projecting the samples $\boldsymbol{x}$ onto a line

$$y = w^T x$$

  - Of all the possible lines we would like to select the one that maximizes the separability of the scalars
    - This is illustrated for the two-dimensional case in the following figures



LDA for two classes – Cont'd

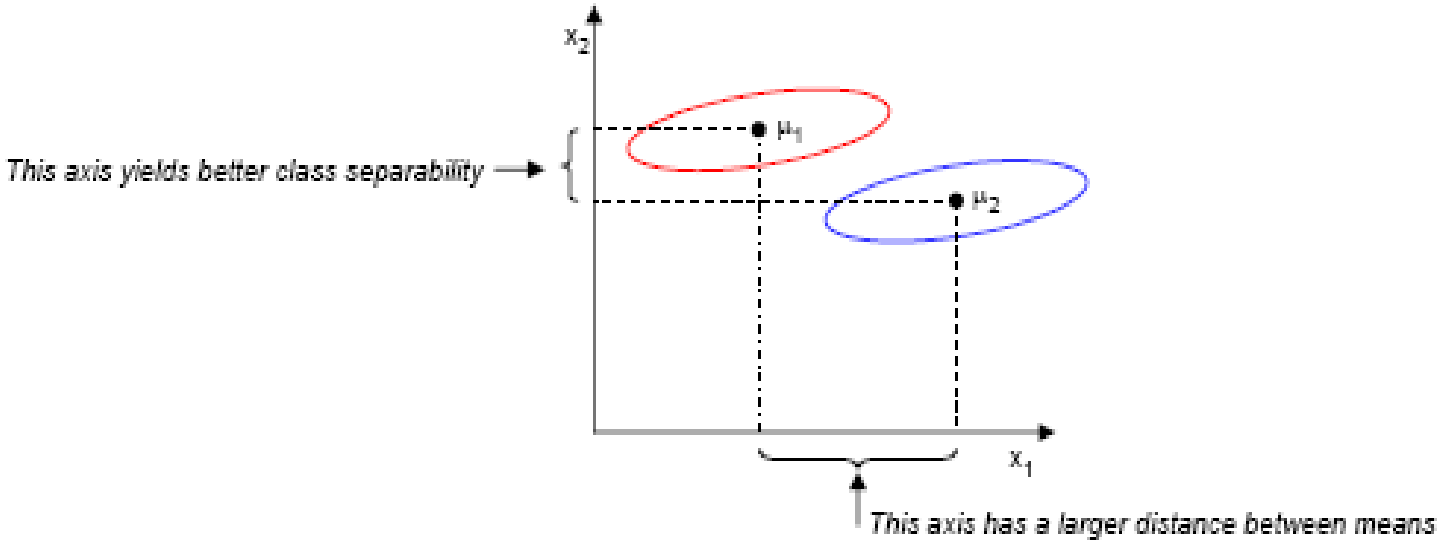
- **In order to find a good projection vector, we need to define a measure of separation between the projections**
  - The mean vector of each class in ***x*** and ***y*** feature space is

$$\mu_i = \frac{1}{N_i}\sum_{x\in\omega_i} x \quad \text{and} \quad \tilde{\mu}_i = \frac{1}{N_i}\sum_{y\in\omega_i} y = \frac{1}{N_i}\sum_{x\in\omega_i} w^T x = w^T \mu_i$$

  - We could then choose the distance between the projected means as our objective function

$$J(w) = \left|\tilde{\mu}_1 - \tilde{\mu}_2\right| = \left|w^T(\mu_1 - \mu_2)\right|$$

  - However, the distance between the projected means is not a very good measure since it does not take into account the standard deviation within the classes



LDA for two classes – Cont'd

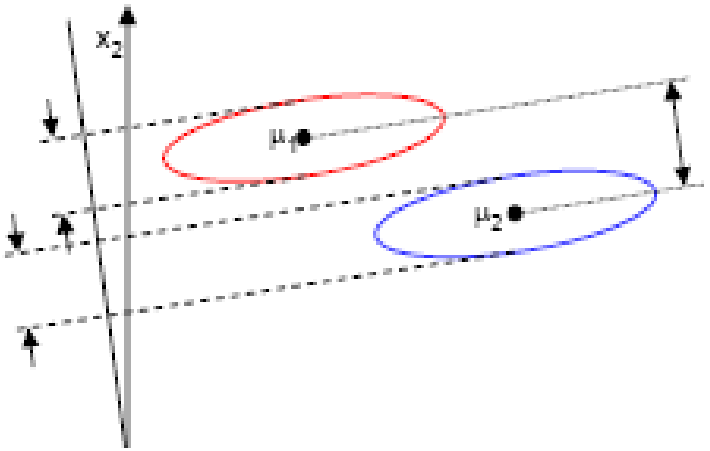
- **The solution proposed by Fisher is to maximize a function that represents the difference between the means, normalized by a measure of the within-class scatter**
  - For each class we define the scatter, an equivalent of the variance, as

$$\tilde{s}_1^2 = \sum_{y \in \omega_1} (y - \tilde{\mu}_1)^2$$

    - where the quantity $\left(\tilde{s}_1^2 + \tilde{s}_2^2\right)$ is called the within-class scatter of the projected examples
  - The Fisher linear discriminant is defined as the linear function $w^T x$ that maximizes the criterion function

$$J(w) = \frac{\left|\tilde{\mu}_1 - \tilde{\mu}_2\right|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

  - Therefore, we will be looking for a projection where examples from the same class are projected very close to each other and, at the same time, the projected means are as farther apart as possible



LDA for two classes – Cont'd

- In order to find the optimum projection w*, we need to express J(w) as an explicit function of w
- We define a measure of the scatter in multivariate feature space **x**, which are scatter matrices

$$S_i = \sum_{x \in \omega_i} (x - \mu_i)(x - \mu_i)^T$$

$$S_1 + S_2 = S_W$$

  - where S_W is called the **within-class scatter matrix**
- The scatter of the projection **y** can then be expressed as a function of the scatter matrix in feature space **x**

$$\tilde{s}_i^2 = \sum_{y \in \omega_i} (y - \tilde{\mu}_i)^2 = \sum_{x \in \omega_i} (w^T x - w^T \mu_i)^2 = \sum_{x \in \omega_i} w^T (x - \mu_i)(x - \mu_i)^T w = w^T S_i w$$

$$\tilde{s}_1^2 + \tilde{s}_2^2 = w^T S_W w$$

- Similarly, the difference between the projected means can be expressed in terms of the means in the original feature space

$$(\tilde{\mu}_1 - \tilde{\mu}_2)^2 = (w^T \mu_1 - w^T \mu_2)^2 = w^T \underbrace{(\mu_1 - \mu_2)(\mu_1 - \mu_2)^T}_{S_B} w = w^T S_B w$$

  - The matrix S_B is called the **between-class scatter**. Note that, since S_B is the outer product of two vectors, its rank is at most one
- We can finally express the Fisher criterion in terms of S_W and S_B as

$$J(w) = \frac{w^T S_B w}{w^T S_W w}$$

LDA for two classes – Cont'd

- To find the maximum of J(w) we derive and equate to zero

$$\frac{d}{dw}[J(w)] = \frac{d}{dw}\left[\frac{w^T S_B w}{w^T S_W w}\right] = 0 \Rightarrow$$

$$\Rightarrow \left[w^T S_W w\right]\frac{d\left[w^T S_B w\right]}{dw} - \left[w^T S_B w\right]\frac{d\left[w^T S_W w\right]}{dw} = 0 \Rightarrow$$

$$\Rightarrow \left[w^T S_W w\right]2S_B w - \left[w^T S_B w\right]2S_W w = 0$$

- Dividing by $w^T S_W w$

$$\frac{\left[w^T S_W w\right]}{\left[w^T S_W w\right]}S_B w - \frac{\left[w^T S_B w\right]}{\left[w^T S_W w\right]}S_W w = 0 \Rightarrow$$

$$\Rightarrow S_B w - J S_W w = 0 \Rightarrow$$

$$\Rightarrow S_W^{-1} S_B w - J w = 0$$

- Solving the generalized eigenvalue problem ($S_W^{-1} S_B w = J w$) yields

$$w^* = \underset{w}{\operatorname{argmax}}\left\{\frac{w^T S_B w}{w^T S_W w}\right\} = S_W^{-1}(\mu_1 - \mu_2)$$

  - This is know as Fisher's Linear Discriminant (1936), although it is not a discriminant but rather a specific choice of direction for the projection of the data down to one dimension

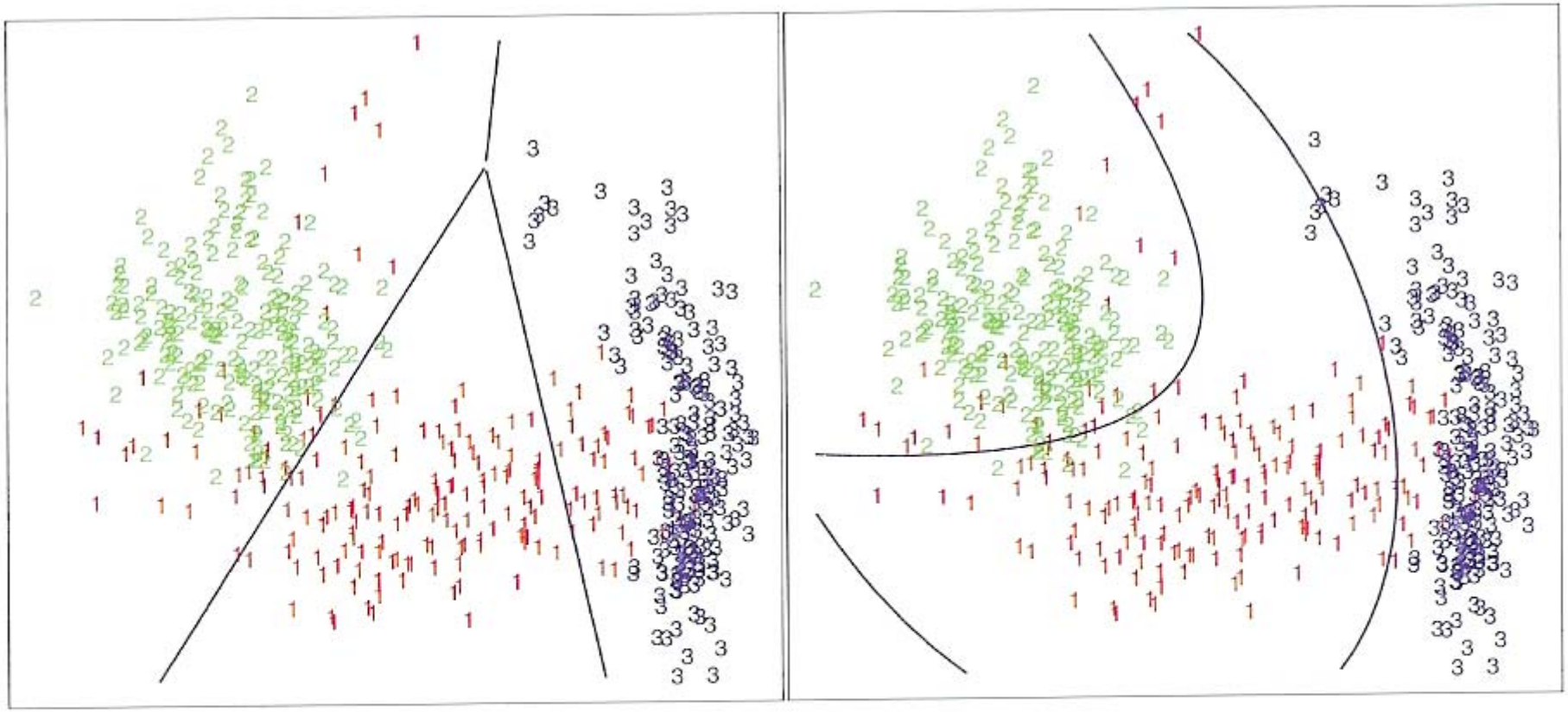
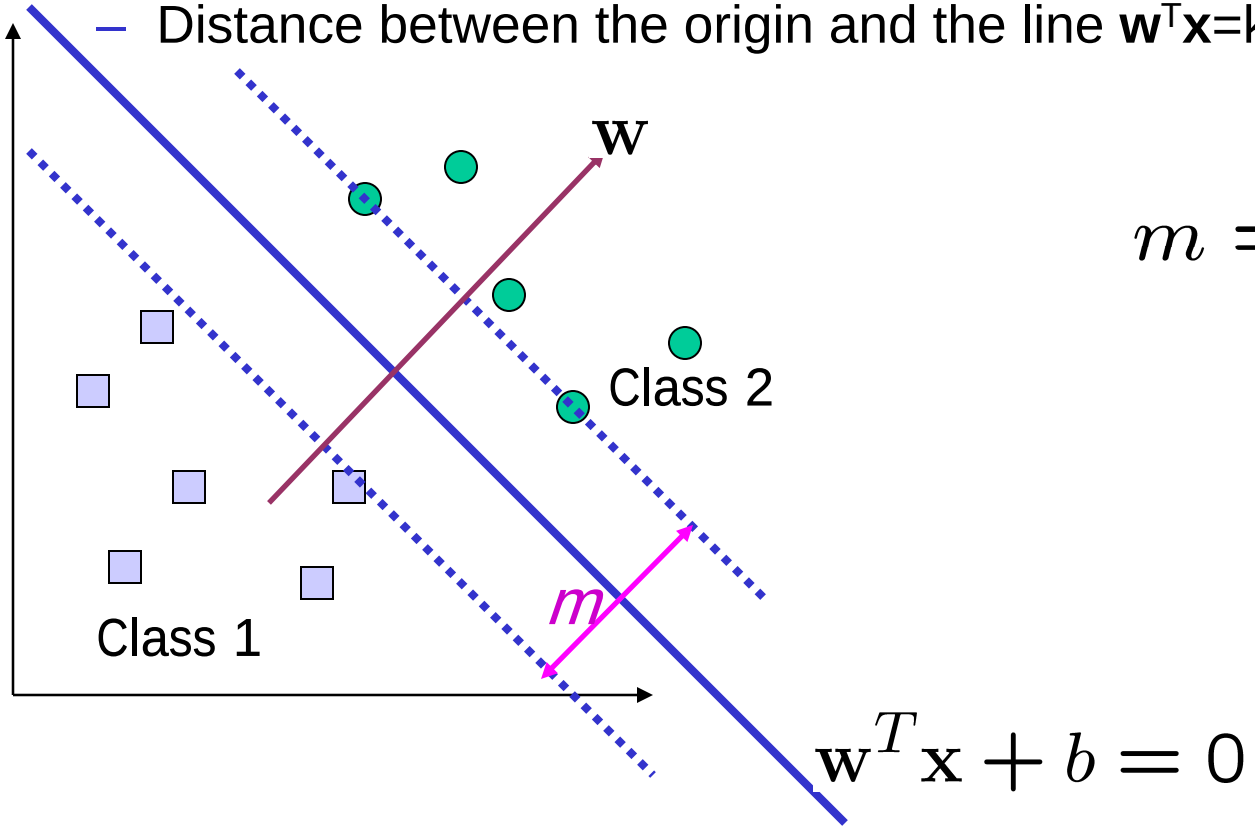


FIGURE 4.1. *The left plot shows some data from three classes, with linear decision boundaries found by linear discriminant analysis. The right plot shows quadratic decision boundaries. These were obtained by finding linear boundaries in the five-dimensional space* $X_1, X_2, X_{12}, X_1^2, X_2^2$. *Linear inequalities in this space are quadratic inequalities in the original space.*

SVM: Large-margin Decision Boundary

- The decision boundary should be as far away from the data of both classes as possible
  - We should maximize the margin, *m*
  - Distance between the origin and the line $\mathbf{w}^T\mathbf{x}=k$ is $k/||\mathbf{w}||$



$$m = \frac{2}{||\mathbf{w}||}$$

Finding the Decision Boundary

- Let $\{x_1, ..., x_n\}$ be our data set and let $y_i \in \{1,-1\}$ be the class label of x_i

$$y_i(W^T x_i + b) \geq 1$$

- The decision boundary should classify all points correctly $\Rightarrow$

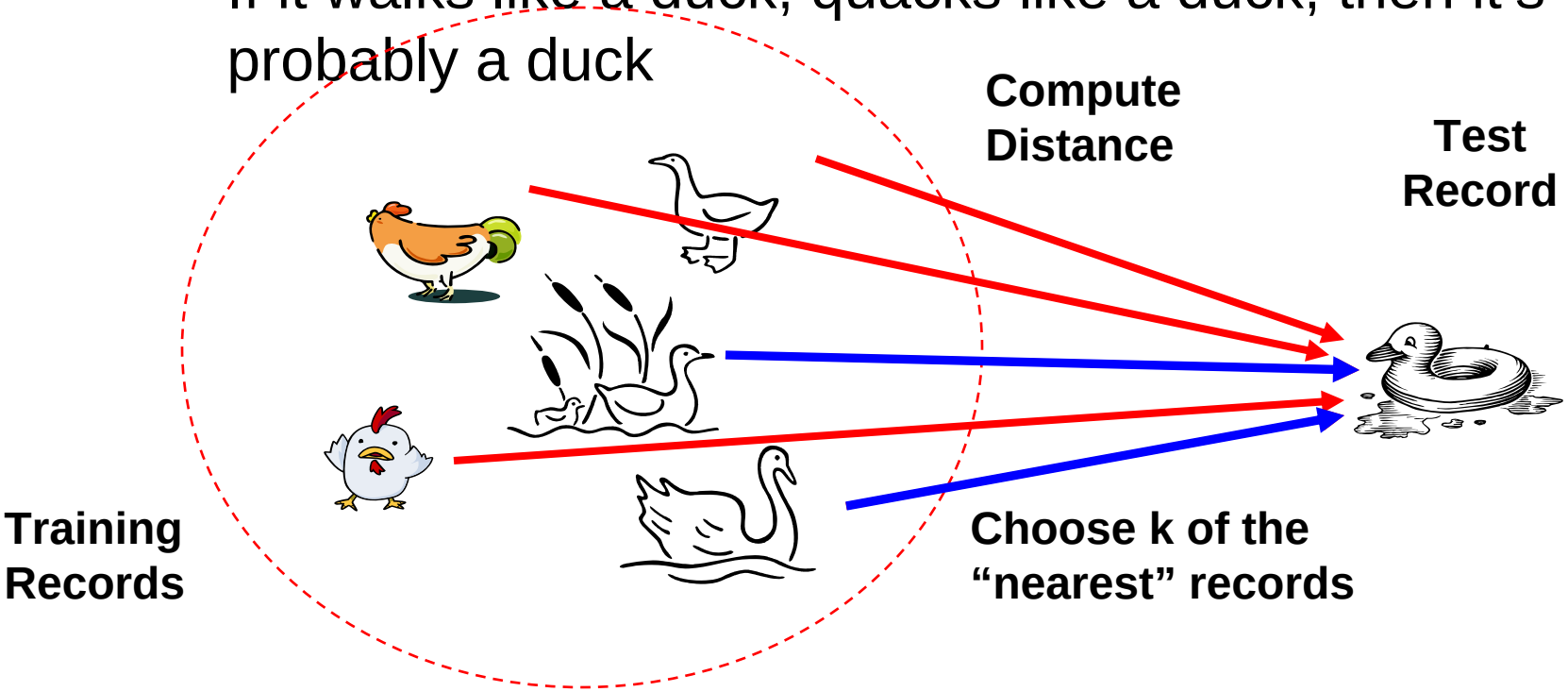
$$Minimize \ \|W\|^2 \ \text{subject to} \ y_i(W^T x_i + b) \geq 1$$

- This is a constrained optimization problem. Can be solved using Lagrangian

- Popular program SVMLight
  http://svmlight.joachims.org/

Nearest Neighbor Classifiers

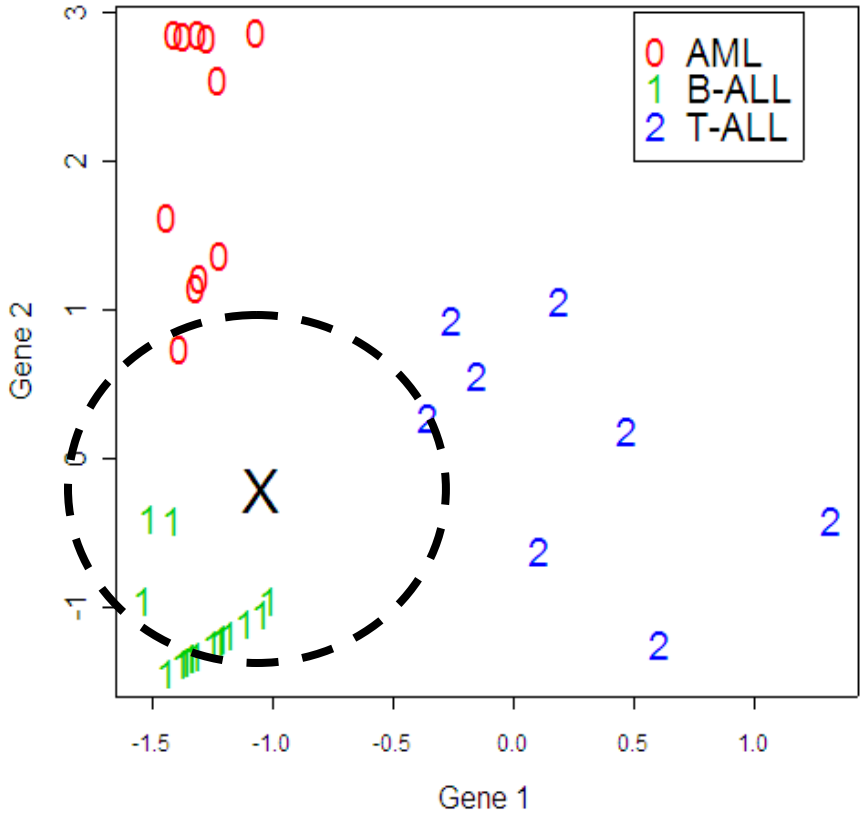
- Basic idea:
  - If it walks like a duck, quacks like a duck, then it's probably a duck



Nearest neighbor classification (k-NN)

- **Requires three things**
  - **The set of stored records**
  - **Distance metric to compute distance between records**
  - **The value of *k*, the number of nearest neighbors to retrieve**

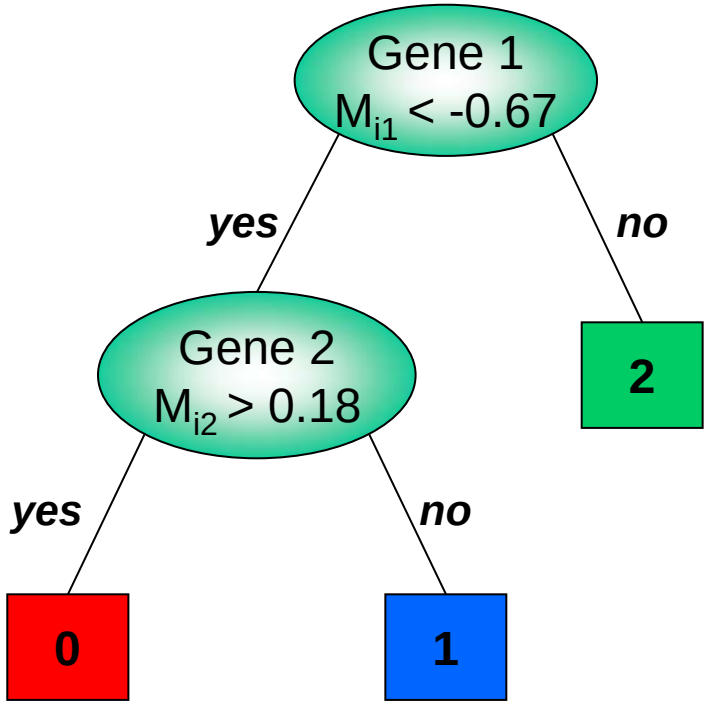
- **To classify an unknown record:**
  - **Compute distance to other training records**
  - **Identify *k* nearest neighbors**
  - **Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)**

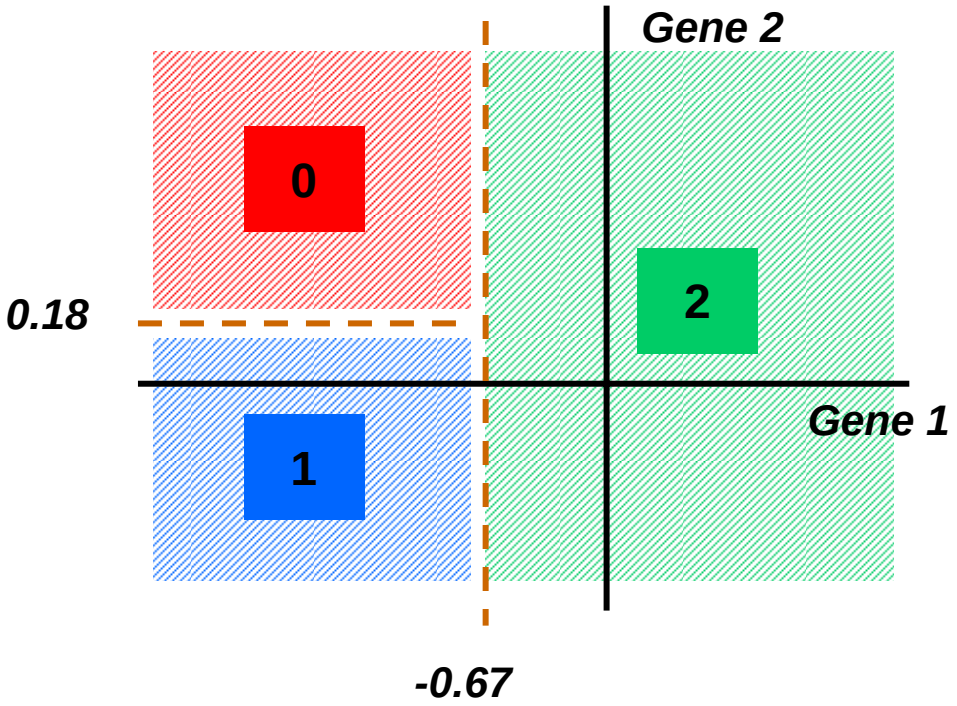


Classification and regression trees (CART)

- Partition the feature space into a set of rectangles, then fit a simple model in each one
- Binary tree structured classifiers are constructed by repeated splits of subsets (nodes) of the measurement space $\underline{X}$ into two descendant subsets (starting with $\underline{X}$ itself)
- Each terminal subset is assigned a class label; the resulting partition of $\underline{X}$ corresponds to the classifier

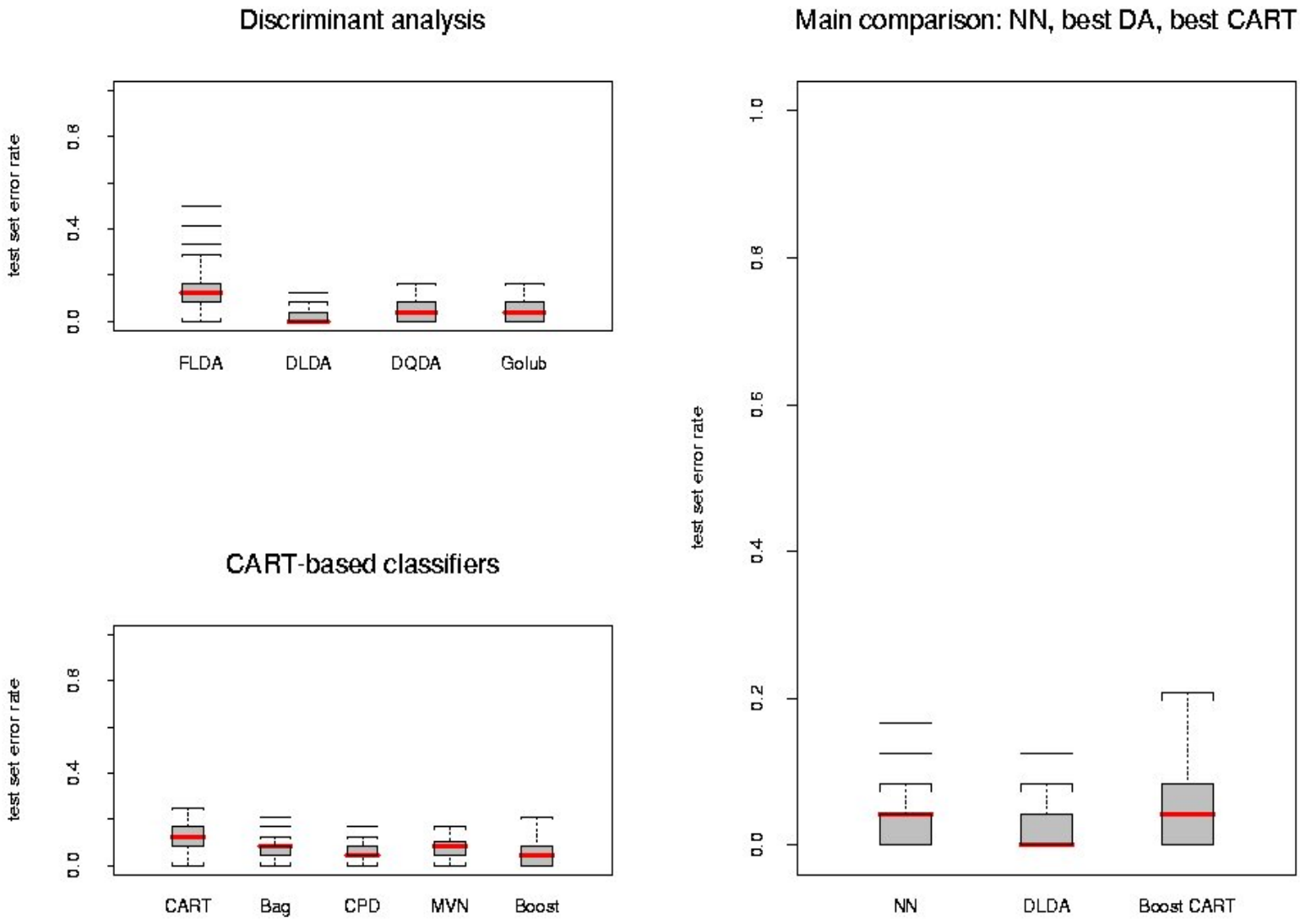
Classification trees



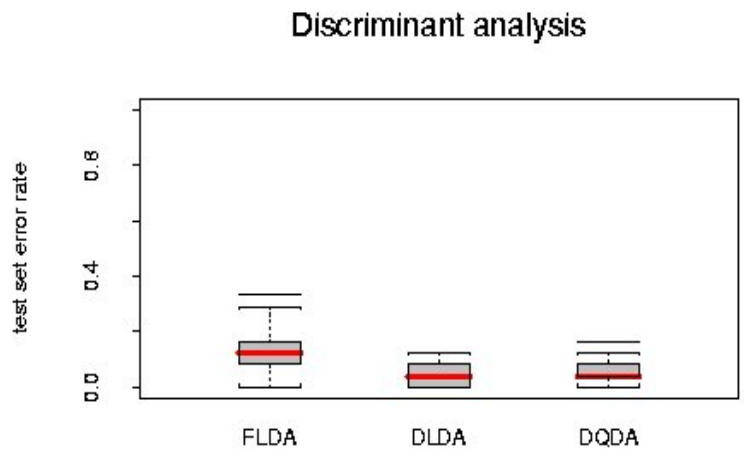


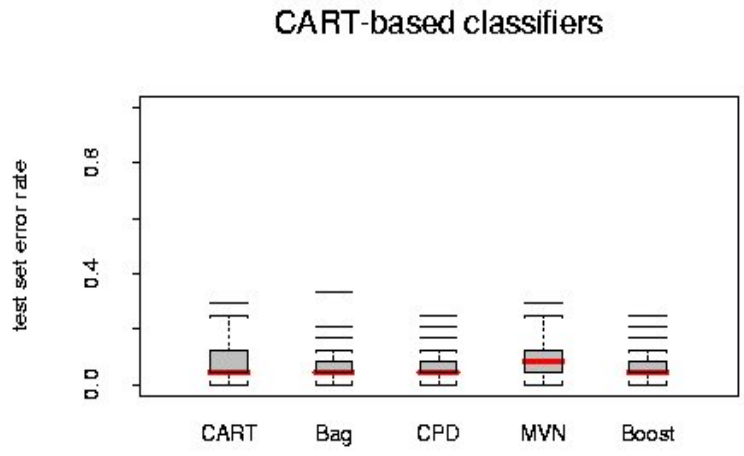
Comparison study datasets

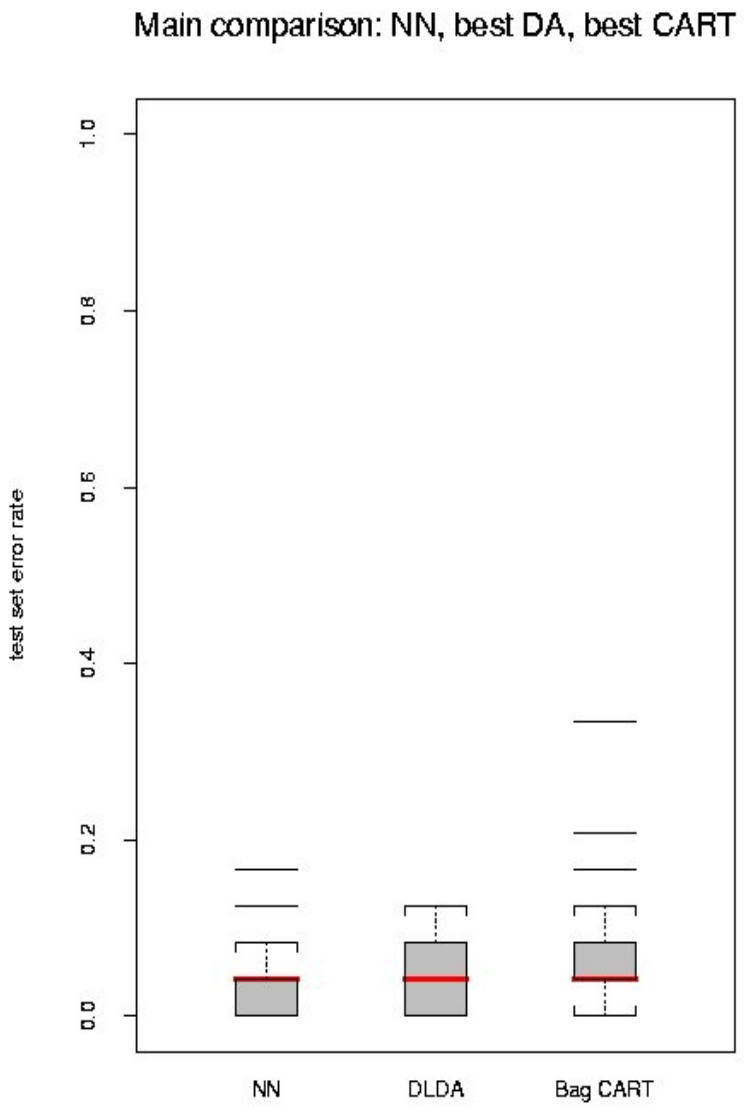
- Leukemia – Golub et al. (1999)
  - n = 72 samples, G = 3,571 genes
  - 3 classes (B-cell ALL, T-cell ALL, AML)
- Lymphoma – Alizadeh et al. (2000)
  - n = 81 samples, G = 4,682 genes
  - 3 classes (B-CLL, FL, DLBCL)
- NCI 60 – Ross et al. (2000)
  - N = 64 samples, p = 5,244 genes
  - 8 classes



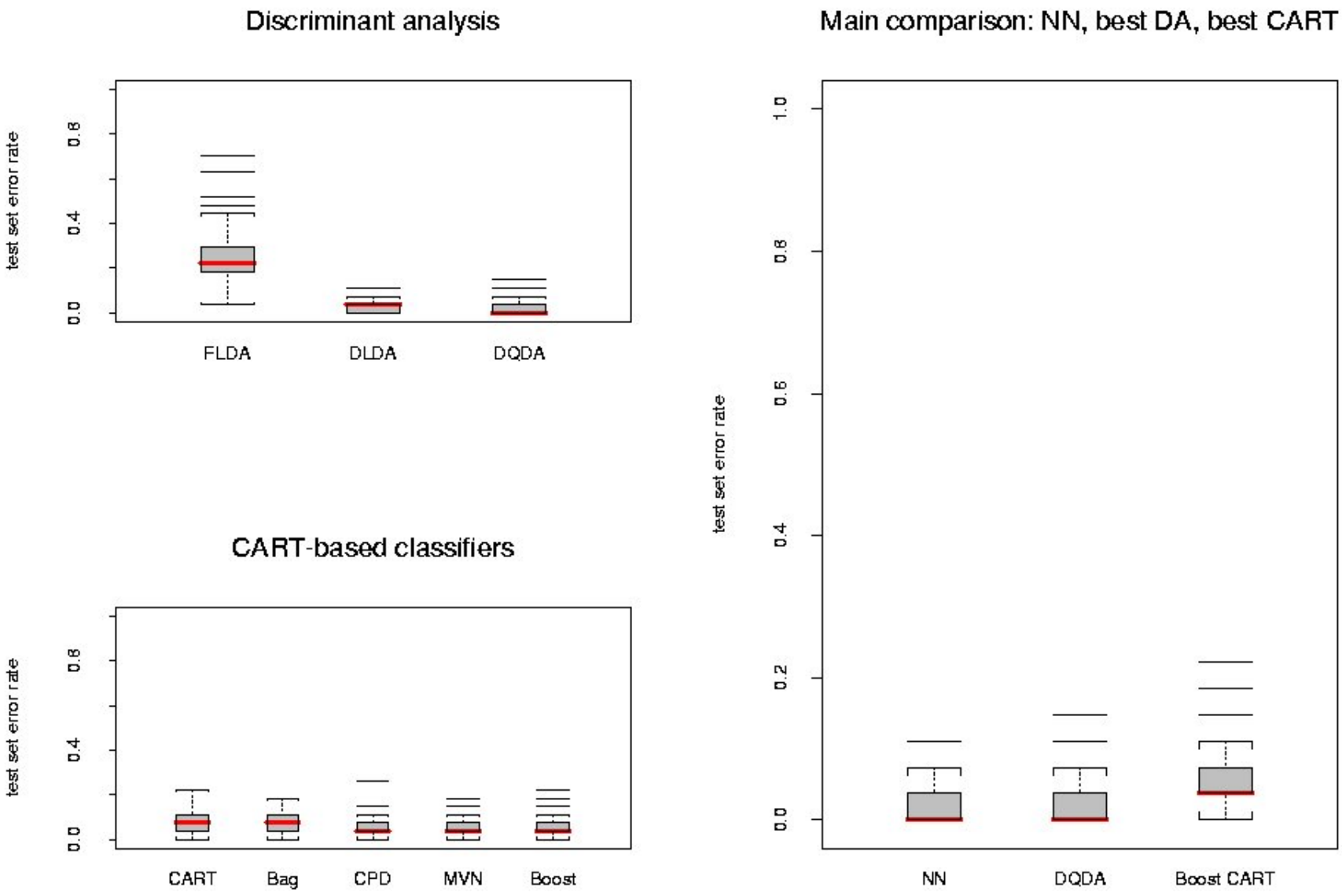
Leukemia data, 2 classes: Test set error rates;150 LS/TS runs



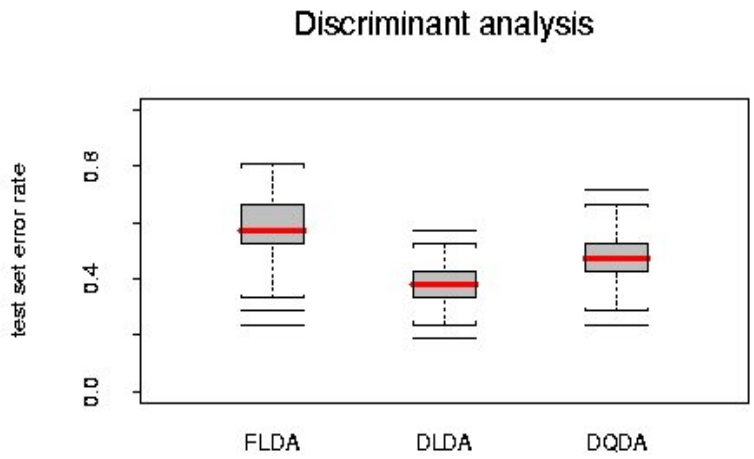


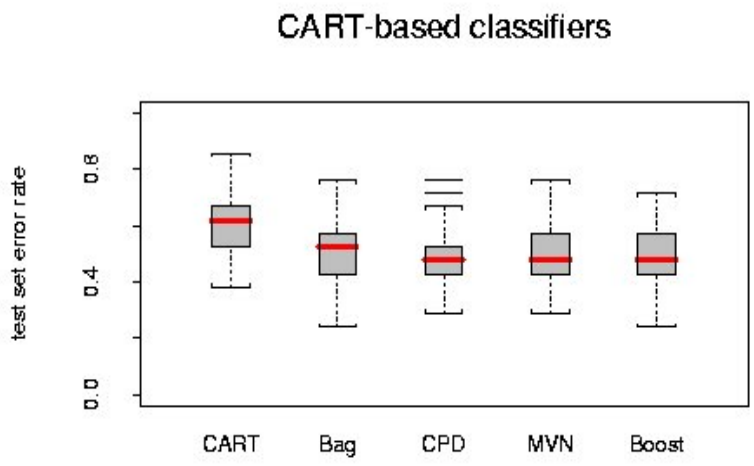


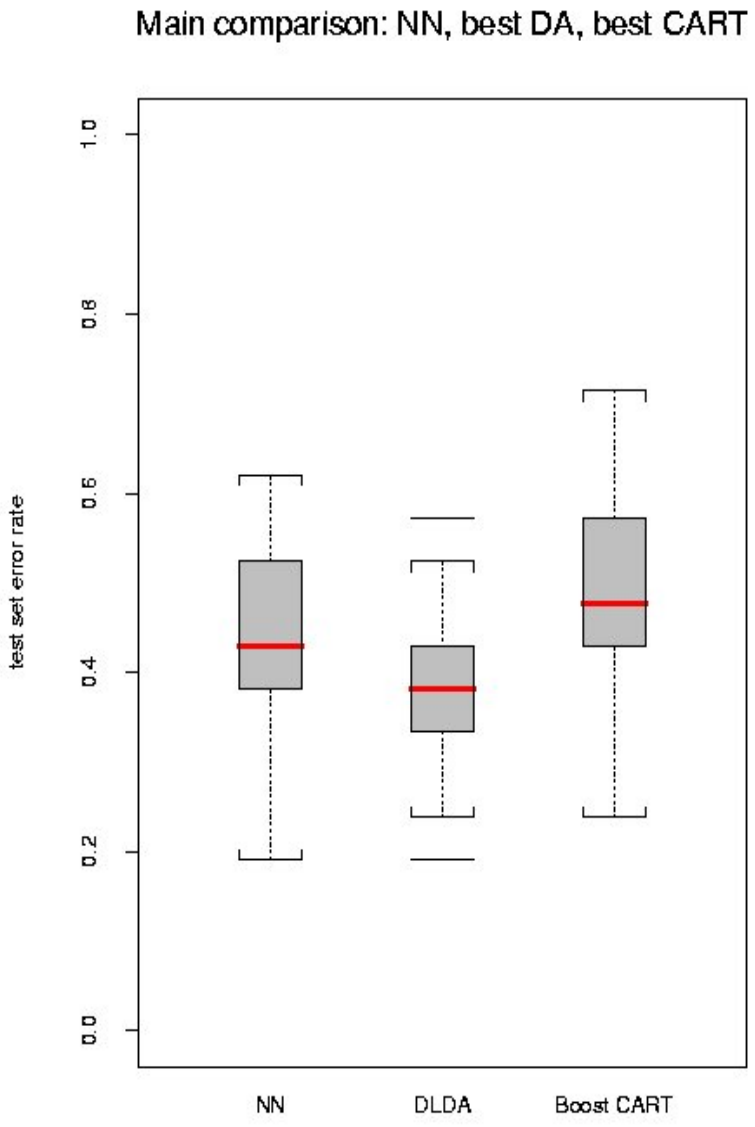
Leukemia data, 3 classes: Test set error rates;150 LS/TS runs



Lymphoma data, 3 classes: Test set error rates; N=150 LS/TS runs







NCI 60 data :Test set error rates;150 LS/TS runs

Results

- In the main comparison, NN and DLDA had the smallest error rates, FLDA had the highest
- For the lymphoma and leukemia datasets, increasing the number of genes to G=200 didn't greatly affect the performance of the various classifiers; there was an improvement for the NCI 60 dataset.
- More careful selection of a small number of genes (10) improved the performance of FLDA dramatically.

Softwares

NAME	COMPANY	FEATURE
dChip	Harvard	Model based analysis, clustering, and loads of other uses
ImaGene	Biodiscovery	quantification of gene expression value, constant-factor normalization
GeneSight	Biodiscovery	background adjustment, clustering(hierarchical, SOM)
GeneSpring	Silicon Genetics	normalization, clustering(hierarchical, SOM), fold-change
Spotfire	Spotfire	PCA, clustering, fold-change
Resolver	Rosetta	clustering, PCA, fold-change, plots
LifeArray	Incyte	clustering, PCA, fold-change
Expressionist	GeneData	clust, PCA, fold-change
GeneExpress	Gene Logic	clustering, PCA, fold-change
IPLab MicroArray Suit	Scanalytics	clustering, image, fold-changes
J-Express	U Bergen	clustering, PCA
UCI/NCGR	UCI/NCGR	t-test for fold-change
TreeView	Stanford	clustering, SOM, image analysis (similarity, Cluster,Lawrence Berkeley Lab, Eisen Lab)
EPCLUST	EBI	clustering
SOM	Whitehead Inst.	SOM