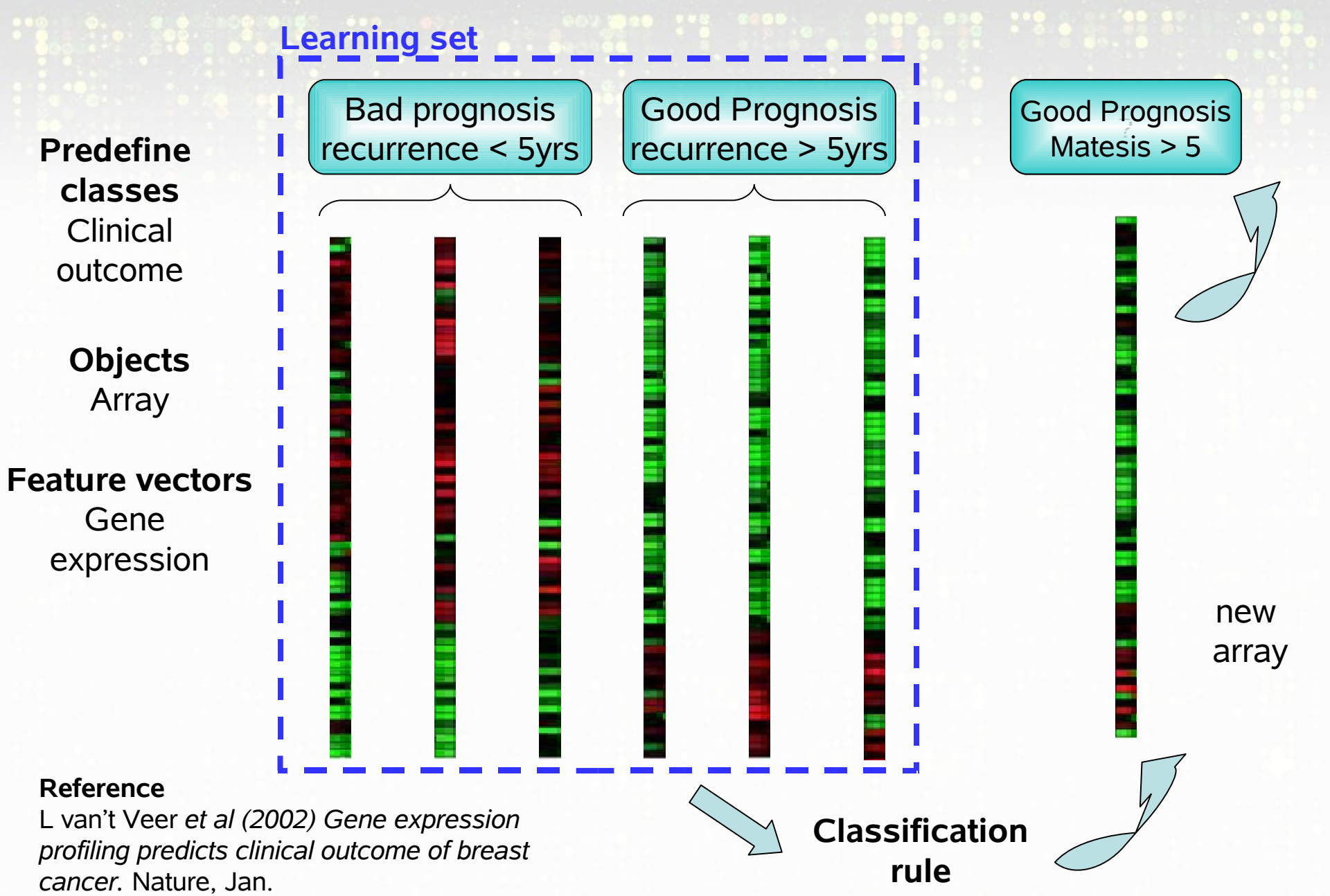
Introduction to Classification Issues in Microarray Data Analysis

Jane Fridlyand
Jean Yee Hwa Yang

University of California, San Francisco

Elsinore, Denmark
May 17-21, 2004

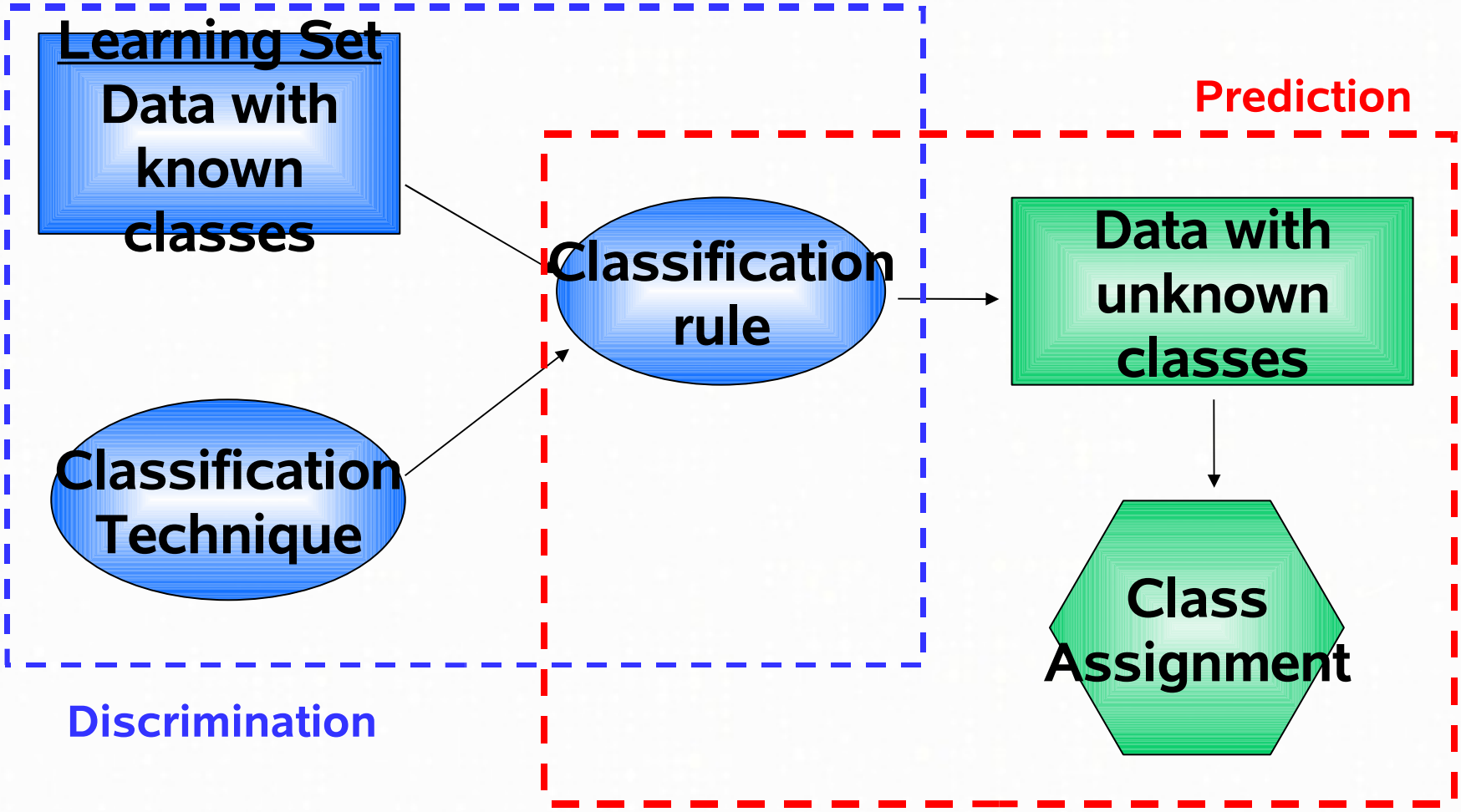


Reference

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

.

Discrimination and Allocation



Classification rule
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 $\mathbf{X}$ by

$$C(\mathbf{X}) = \text{argmax}_k \, p_k(\mathbf{X})$$

Gaussian ML discriminant rules

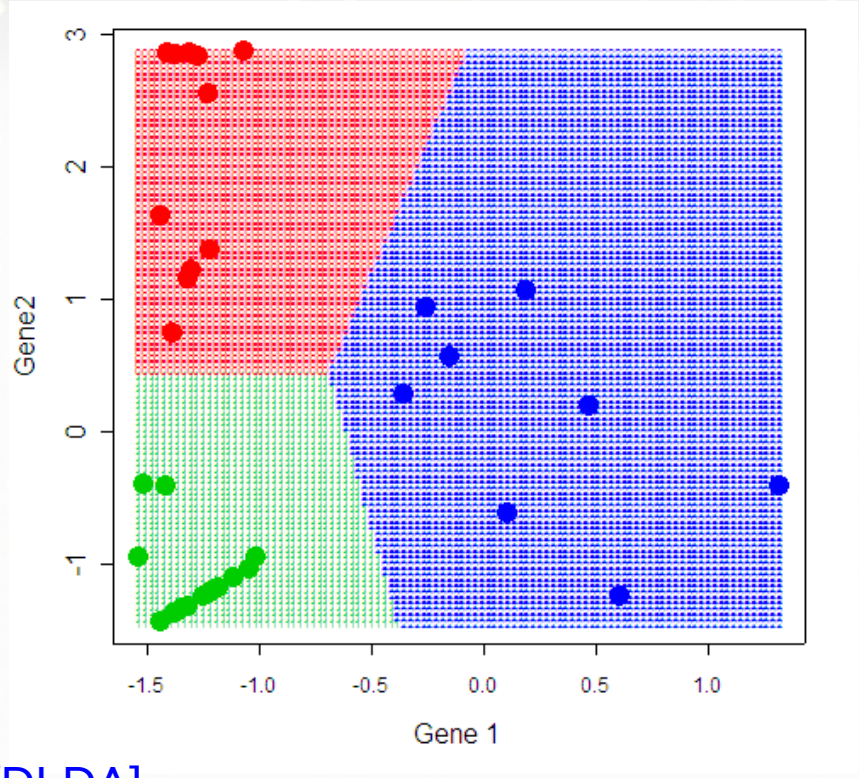
- For multivariate Gaussian (normal) class densities $\mathbf{X}|Y= k \sim N(\mu_k, \Sigma_k)$, the ML classifier is

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

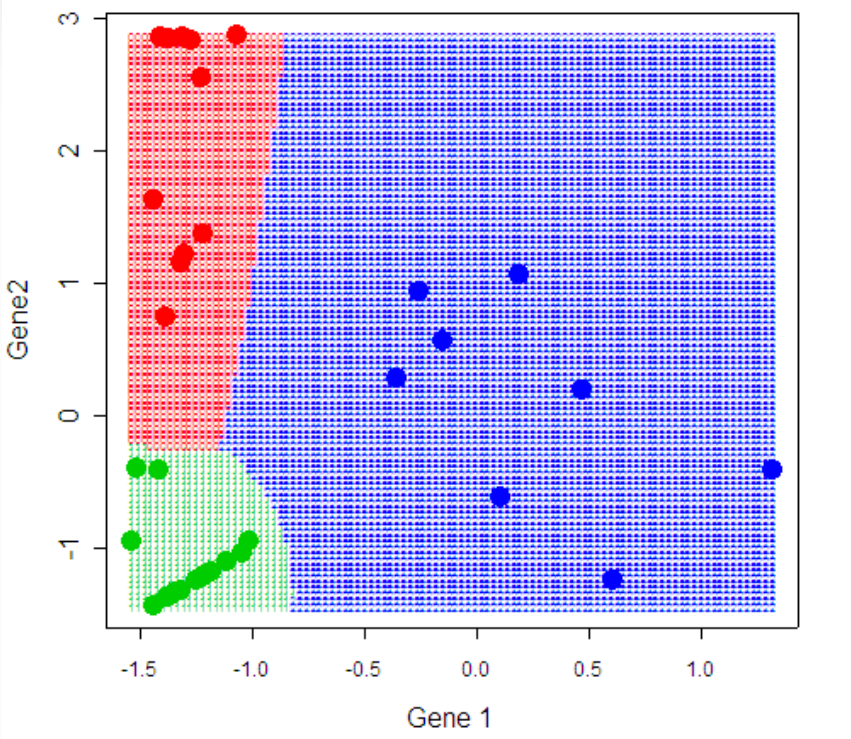
- In general, this is a quadratic rule (Quadratic discriminant analysis, or QDA)

- In practice, population mean vectors μ_k and covariance matrices Σ_k are estimated by corresponding sample quantities

ML discriminant rules - special cases



[DLDA]
Diagonal linear discriminant analysis
class densities have the same diagonal covariance matrix $\nabla = \text{diag}(s_1^2, \ldots, s_p^2)$

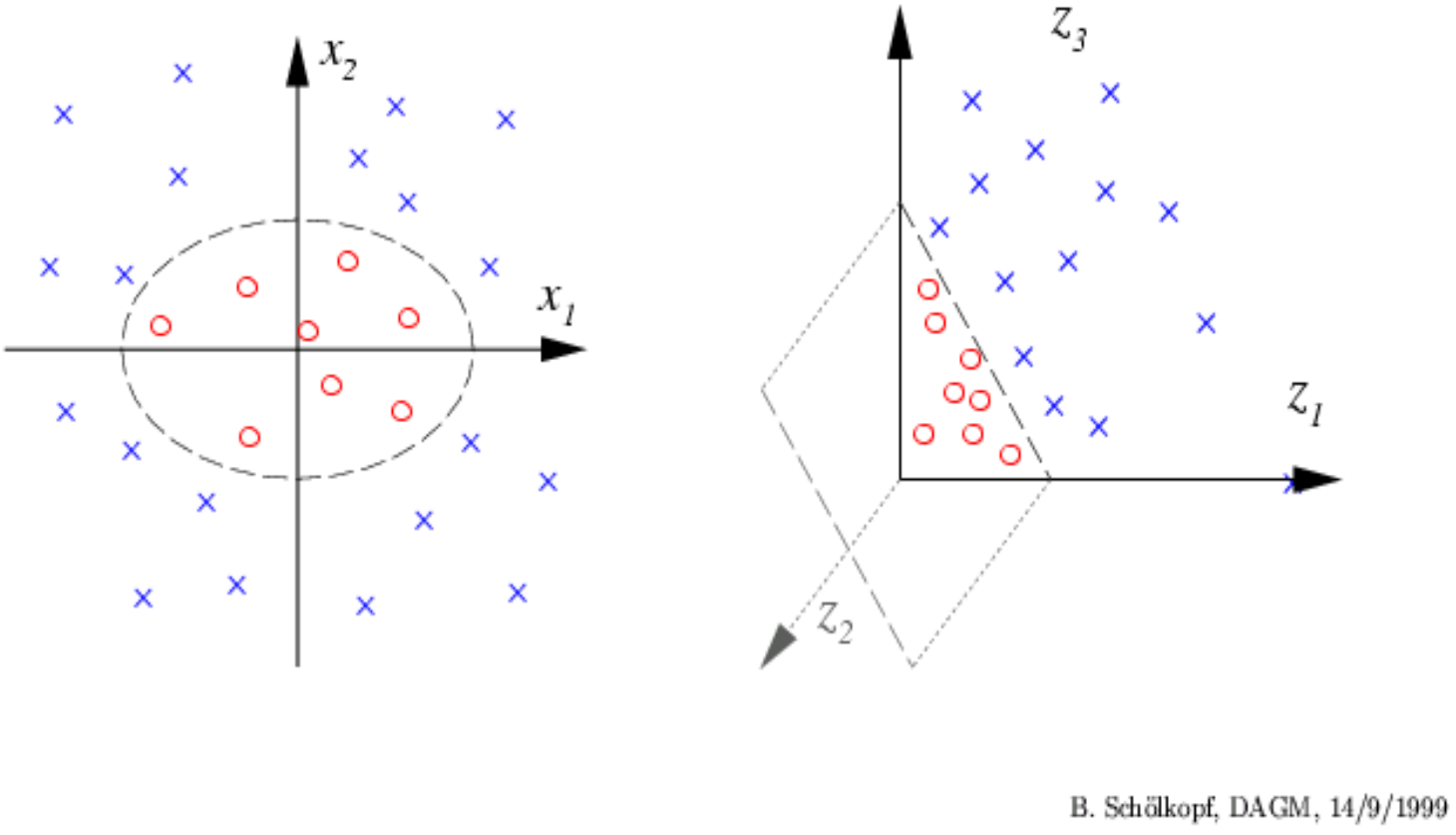


[DQDA]
Diagonal quadratic discriminant analysis)
class densities have different diagonal covariance matrix $\nabla_k = \text{diag}(s_{1k}^2, \ldots, s_{pk}^2)$

Note. Weighted gene voting of Golub et al. (1999) is a minor variant of DLDA for two classes (different variance calculation).

Classification with SVMs

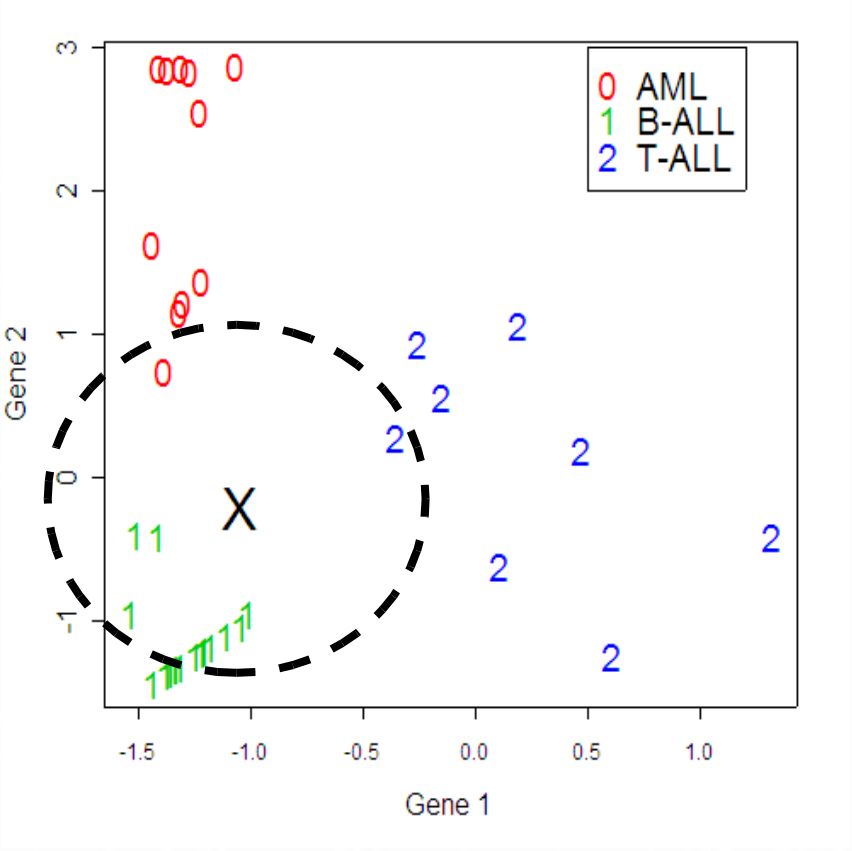
Generalization of the ideas of separating hyperplanes in the original space. Linear boundaries between classes in higher-dimensional space lead to the non-linear boundaries in the original space.

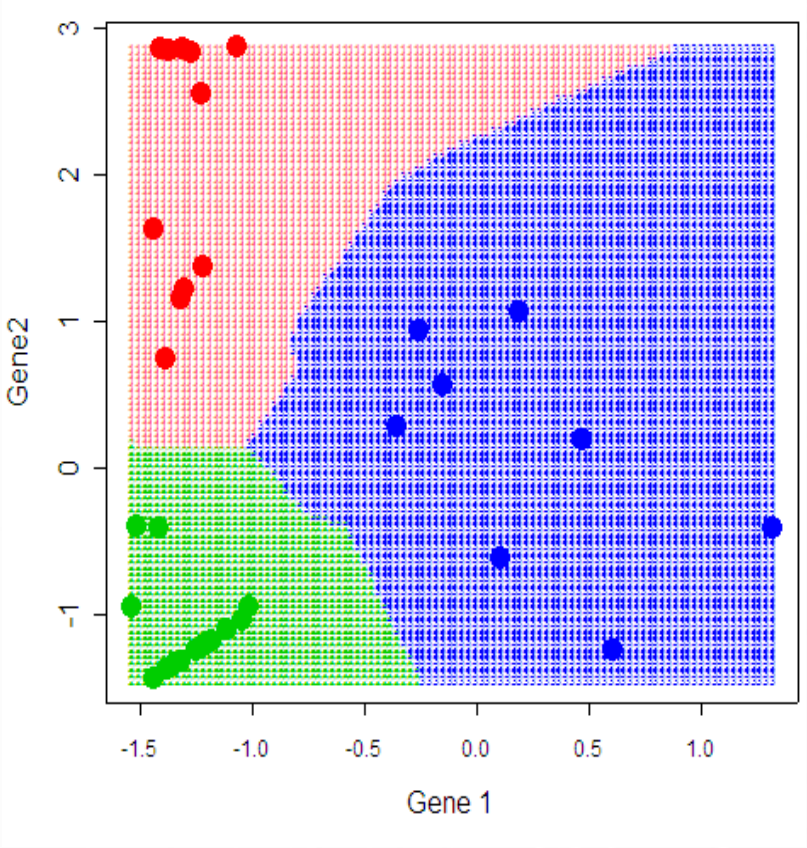


Nearest neighbor classification

- Based on a measure of distance between observations (e.g. Euclidean distance or one minus correlation).
- k-nearest neighbor rule (Fix and Hodges (1951)) classifies an observation **X** as follows:
  - find the k observations in the learning set closest to **X**
  - predict the class of **X** by majority vote, i.e., choose the class that is most common among those k observations.
- The number of neighbors k can be chosen by cross-validation (more on this later).

Nearest neighbor rule

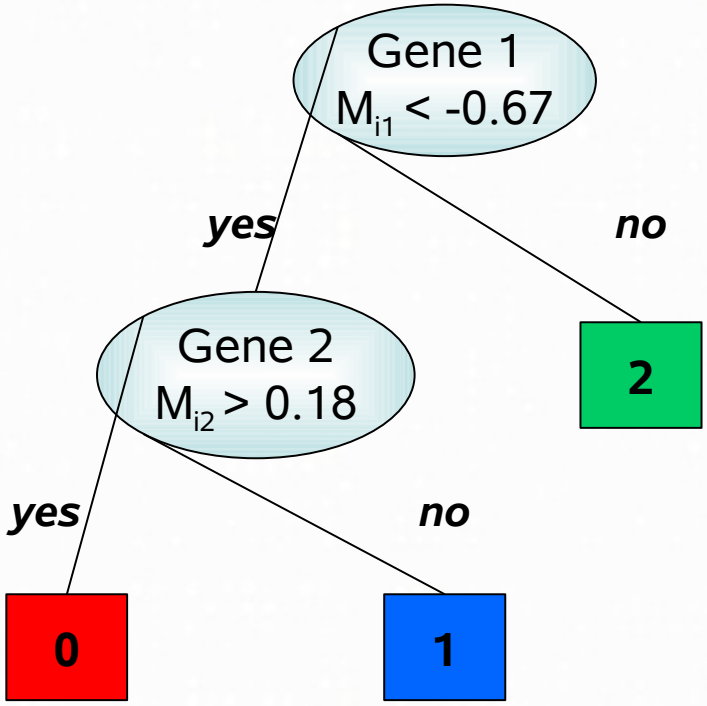


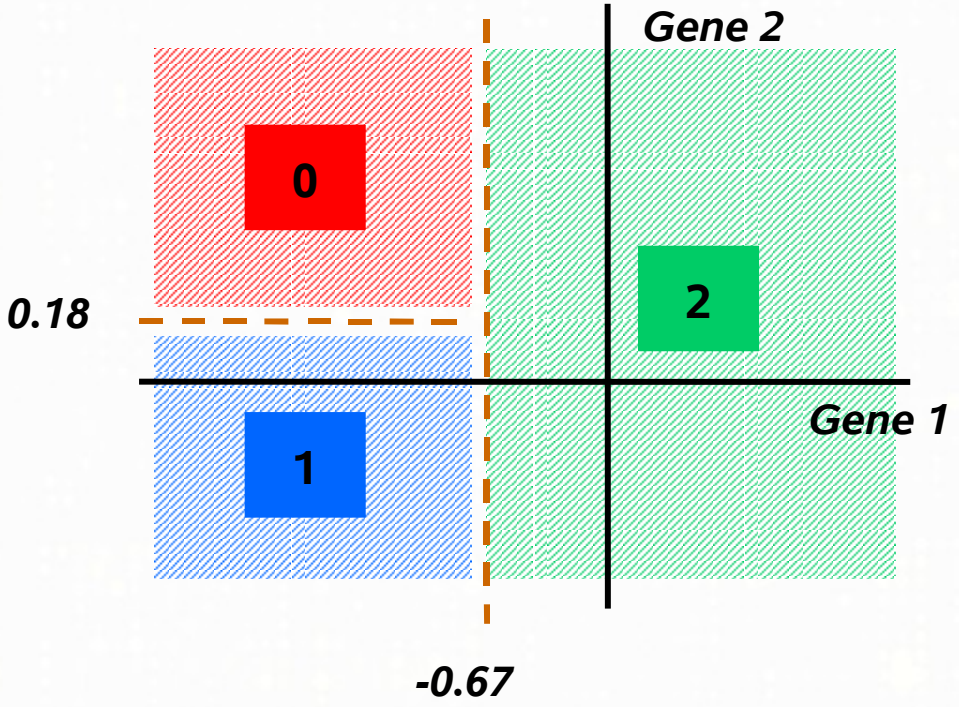


Classification tree

- 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 tree





Three aspects of tree construction

- Split selection rule:
  - Example, at each node, choose split maximizing decrease in impurity (e.g. Gini index, entropy, misclassification error).
- Split-stopping:
  - Example, grow large tree, prune to obtain a sequence of subtrees, then use cross-validation to identify the subtree with lowest misclassification rate.
- Class assignment:
  - Example, for each terminal node, choose the class minimizing the resubstitution estimate of misclassification probability, given that a case falls into this node.

Supplementary slide

Other classifiers include…

- Neural networks
- Projection pursuit
- Bayesian belief networks
- …

Why select features

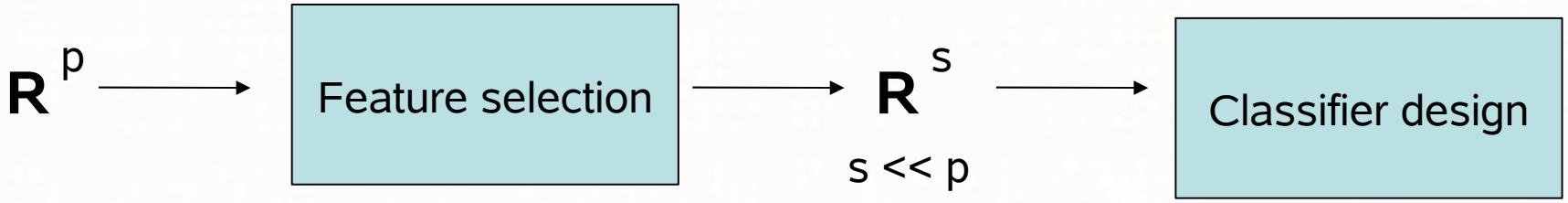
- Lead to better classification performance by removing variables that are noise with respect to the outcome
- May provide useful insights into etiology of a disease
- Can eventually lead to the diagnostic tests (e.g., “breast cancer chip”)

Approaches to feature selection

- Methods fall into three basic category
  - Filter methods
  - Wrapper methods
  - Embedded methods

- The simplest and most frequently used methods are the filter methods.

Filter methods



- Features are scored independently and the top s are used by the classifier

- Score: correlation, mutual information, t-statistic, F-statistic, p-value, tree importance statistic etc

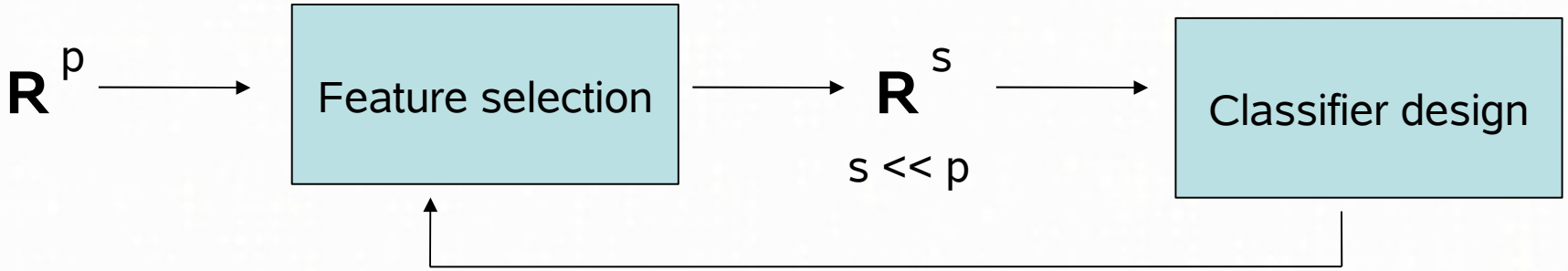
Easy to interpret. Can provide some insight into the disease markers.

Problems with filter method

- Redundancy in selected features: features are considered independently and not measured on the basis of whether they contribute new information
- Interactions among features generally can not be explicitly incorporated (some filter methods are smarter than others)
- Classifier has no say in what features should be used: some scores may be more appropriates in conjuction with some classifiers than others.

Supplementary slide

Wrapper methods



•Iterative approach: many feature subsets are scored based on classification performance and best is used.

•Selection of subsets: forward selection, backward selection, Forward-backward selection, tree harvesting etc

Problems with wrapper methods

- Computationally expensive: for each feature subset to be considered, a classifier must be built and evaluated
- No exhaustive search is possible (2^p subsets to consider) : generally greedy algorithms only.
- Easy to overfit.

Supplementary slide

Embedded methods

- Attempt to jointly or simultaneously train both a classifier and a feature subset
- Often optimize an objective function that jointly rewards accuracy of classification and penalizes use of more features.
- Intuitively appealing

Some examples: tree-building algorithms, shrinkage methods (LDA, kNN)

Performance assessment

- Any classification rule needs to be evaluated for its performance on the future samples. It is almost never the case in microarray studies that a large independent population-based collection of samples is available at the time of initial classifier-building phase.

- One needs to estimate future performance based on what is available: often the same set that is used to build the classifier.

- Assessing performance of the classifier based on
  - Cross-validation.
  - Test set
  - Independent testing on future dataset

Performance assessment (I)

- Resubstitution estimation: error rate on the learning set.
  - Problem: downward bias

- Test set estimation:

  1) divide learning set into two sub-sets, L and T; Build the classifier on L and compute the error rate on T.

  2) Build the classifier on the training set (L) and compute the error rate on an independent test set (T).

  - L and T must be independent and identically distributed (i.i.d).
  - Problem: reduced effective sample size

Supplementary slide

Performance assessment (II)

- V-fold cross-validation (CV) estimation: Cases in learning set randomly divided into V subsets of (nearly) equal size. Build classifiers by leaving one set out; compute test set error rates on the left out set and averaged.
  - Bias-variance tradeoff: smaller V can give larger bias but smaller variance
  - Computationally intensive.

- Leave-one-out cross validation (LOOCV).

  (Special case for V=n). Works well for stable classifiers (k-NN, LDA, SVM)

Supplementary slide

Performance assessment (III)

- Common practice to do feature selection using the learning , then CV only for model building and classification.

- However, usually features are unknown and the intended inference includes feature selection. Then, CV estimates as above tend to be downward biased.

- Features (variables) should be selected only from the learning set used to build the model (and not the entire set)