

Instructor: Sael Lee

CS549 Spring – Computational Biology

Lecture 14: Biomarker Discovery with feature selection methods

Resources: .

- Abeel, T., Helleputte, T., Van de Peer, Y., Dupont, P., & Saeys, Y. (2010). Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics* .26(3), 392–8.
- Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene Selection for Cancer Classification using Support Vector Machines. *Machine Learning*, 46(1-3), 389–422.

Data and text mining

Robust biomarker identification for cancer diagnosis with ensemble feature selection methods

Thomas Abeel^{1,2}, Thibault Helleputte^{3,4}, Yves Van de Peer^{1,2}, Pierre Dupont^{3,4}
and Yvan Saeys^{1,2,*}

¹Department of Plant Systems Biology, VIB, Technologiepark 927, 9052 Gent, ²Department of Molecular Genetics, Ghent University, Ghent, ³Department of Computing Science and Engineering INGI and ⁴Machine Learning Group, Université catholique de Louvain, Louvain, Belgium

Received on May 5, 2009; revised on October 26, 2009; accepted on November 2, 2009

Advance Access publication November 25, 2009

Associate Editor: Thomas Lengauer

Abstract

Motivation:

“Biomarker discovery is an important topic in biomedical applications of computational biology, including applications such as gene and SNP selection from high-dimensional data. Surprisingly, the **stability with respect to sampling variation or robustness of such selection processes** has received attention only recently. ...”

Abstract cont.

Results:

“We show that the robustness of SVMs for biomarker discovery can be substantially increased by using ensemble feature selection techniques, while at the same time improving upon classification performances. The proposed methodology is evaluated on four microarray datasets showing increases of up to almost 30% in robustness of the selected biomarkers, along with an improvement of ~15% in classification performance. The stability improvement with ensemble methods is particularly noticeable for small signature sizes (a few tens of genes), which is most relevant for the design of a diagnosis or prognosis model from a gene signature. ...”

Microarray Datasets

Types of cancer

Table 1. Overview of the datasets used

Name	References	# samples (+/-)	# dim.	SDR
Leukemia	Golub <i>et al.</i> (1999)	72 (47/25)	7,129	0.010
Colon	Alon <i>et al.</i> (1999)	62 (40/22)	2,000	0.031
Lymphoma	Alizadeh <i>et al.</i> (2000)	45 (22/23)	4,026	0.011
Prostate	Singh <i>et al.</i> (2002)	102 (52/50)	6,033	0.017

very low samples/
dimensions ratio.

SDR refers to the ratio between the number of samples and the number of dimensions (or features).

Colon Cancer dataset: is made of samples from 40 tumor and 22 normal colon tissues measuring more than 6500 genes

Leukemia dataset: model to discriminate between acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL) tissues

Microarray Datasets

Types of cancer

Table 1. Overview of the datasets used

Name	References	# samples (+/-)	# dim.	SDR
Leukemia	Golub <i>et al.</i> (1999)	72 (47/25)	7,129	0.010
Colon	Alon <i>et al.</i> (1999)	62 (40/22)	2,000	0.031
Lymphoma	Alizadeh <i>et al.</i> (2000)	45 (22/23)	4,026	0.011
Prostate	Singh <i>et al.</i> (2002)	102 (52/50)	6,033	0.017

very low samples/
dimensions ratio.

SDR refers to the ratio between the number of samples and the number of dimensions (or features).

The lymphoma dataset: comes from a study on diffuse large B-cell Lymphoma in discriminate between two types of lymphoma based on gene expression.

The prostate dataset: was first published in One of the tasks addressed by the authors is to build a model able to discriminate between normal and tumor prostate tissue

Microarray Expression Normalization

The objective of data normalization is to enhance the similarity of genes sharing a common expression pattern throughout the data, but in different ranges of absolute expression values.

IQR-normalization :

The normalized expression value \bar{f}_{ij} is defined as follows.

$$\bar{f}_{ij} = \frac{f_{ij} - m_j}{IQR_j/1.35}$$

where f_{ij} is the original expression value of gene j from sample i , m_j is the median of expression of this gene over all samples and IQR_j stands for the gene-specific interquartile range.

Microarray Expression Normalization

$$\bar{f}_{ij} = \frac{f_{ij} - m_j}{IQR_j / 1.35}$$

The IQR-normalization is more robust to the presence of outliers than a classical Z-score (centering to the mean with unit SD), but the 1.35 scaling factor makes both normalization equivalent whenever the data happens to be normally distributed.

* The normalization parameters for each gene are always estimated from the training samples only and applied subsequently to the validation samples

Stability Evaluation

Stability Concept: adding or deleting a few samples should not drastically modify the top-ranked markers identified by the algorithm.

slight variations of the original dataset, and compare the outcome of the marker selection algorithm across these different variations.

Variations: subsampling the original dataset without replacement containing 90% of the samples of the original dataset.

Stability Evaluation

Stability analysis process:

- Dataset $X = \{x_1, \dots, x_M\}$ with M instances and N features. Then, k subsamplings of size xM ($0 < x < 1$) are drawn randomly from X , where in our experiments $k=500$ and $x=0.9$.
- Feature selection is performed on each of the k subsamplings, and a marker set—further referred to as a *signature*—of a given size is selected.
- Similarity of the signatures of the k subsamples are evaluated for stability.

Stability Measure

The more similar all signatures are, the higher the stability measure will be.

Stability : Defined as the average over all pairwise similarity comparisons between all signatures on the k subsamplings

$$S_{\text{tot}} = \frac{2 \sum_{i=1}^k \sum_{j=i+1}^k \text{KI}(f_i, f_j)}{k(k-1)}$$

f_i : the signature obtained by the selection method on subsampling i ($1 \leq i \leq k$),

Stability Measure

Kuncheva Index which is a stability index between f_i and f_j

$s = |f_i| = |f_j|$
: signature size

$r = |f_i \cap f_j|$: number of common elements in both signatures

$$KI(f_i, f_j) = \frac{r \cdot N - s^2}{s \cdot (N - s)} = \frac{r - (s^2/N)}{s - (s^2/N)}$$

bias correction term:
selecting common features at random

$-1 < KI(f_i, f_j) \leq 1$ and the greater its value, the larger the number of commonly selected features in both signatures

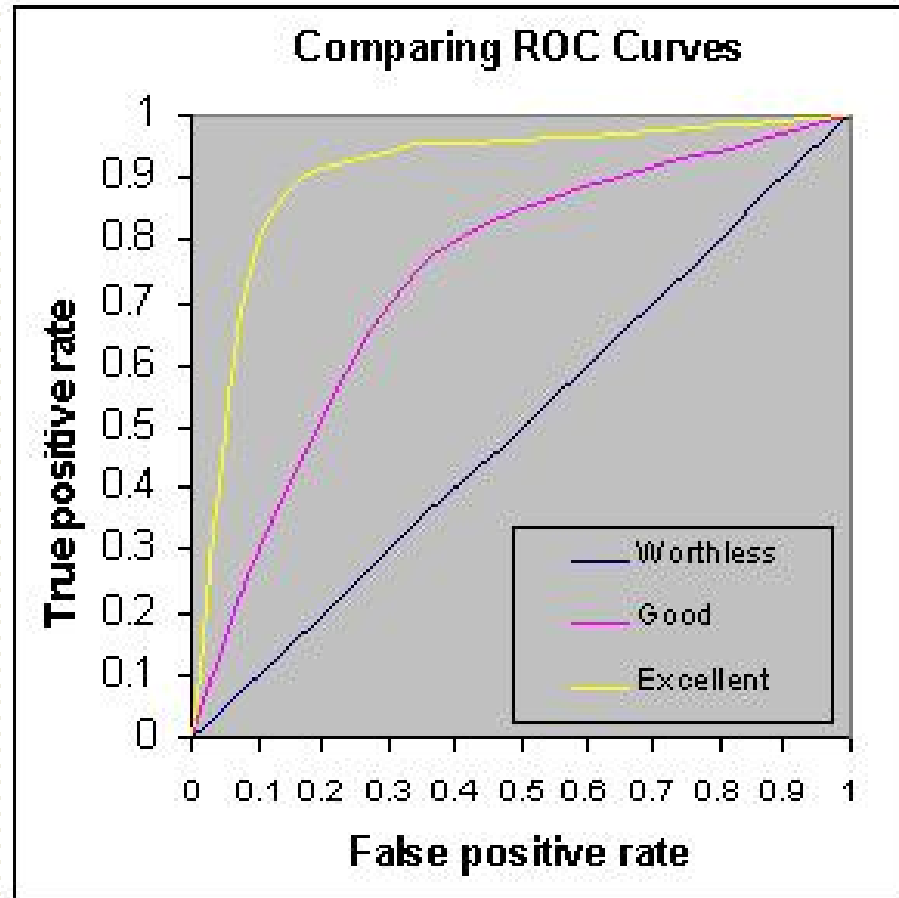
Classification Evaluation

Use the same subsamplings—each containing 90% of the original dataset—as **training sets to select features** and estimate the performance of a classifier. The remaining 10% of the data can be used each time as an independent **validation set** to evaluate classification performance.





Area under the curve of receiver operator curve (AUC ROC):

Receiver Operating Characteristic Methodology: (slides 10-27)
All credits goes to slide by Darlene Goldstein (29 January 2003)

Statistical
measures of the
performance of
a binary
classification test



True disease state vs. Test result

Disease \ Test	Null H. not rejected (Negative test outcome)	Null H. rejected (Positive test outcome)
No disease (D = 0)	 Specificity (TN rate)	 Type I error (FP rate) α
Disease (D = 1)	 Type II error (FN rate) β	 Power $1 - \beta$; Sensitivity (TP rate; recall)

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

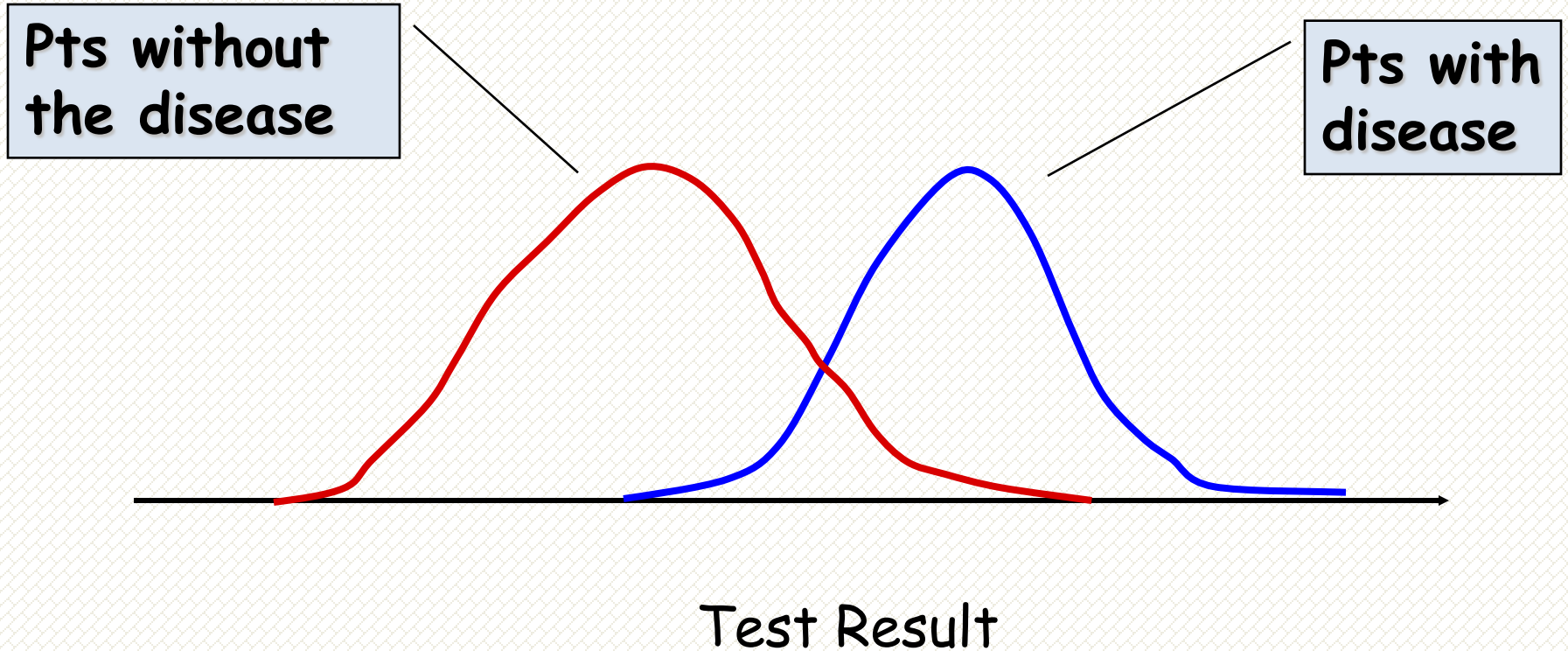
$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN})$$

https://en.wikipedia.org/wiki/Confusion_matrix

Confusion Matrix

		True condition			
		Condition positive	Condition negative		
Total population				Prevalence = $\frac{\sum \text{Condition positive}}{\sum \text{Total population}}$	
Predicted condition	Predicted condition positive	True positive	False positive (Type I error)	Positive predictive value (PPV), Precision = $\frac{\sum \text{True positive}}{\sum \text{Test outcome positive}}$	False discovery rate (FDR) = $\frac{\sum \text{False positive}}{\sum \text{Test outcome positive}}$
	Predicted condition negative	False negative (Type II error)	True negative	False omission rate (FOR) = $\frac{\sum \text{False negative}}{\sum \text{Test outcome negative}}$	Negative predictive value (NPV) = $\frac{\sum \text{True negative}}{\sum \text{Test outcome negative}}$
Accuracy (ACC) = $\frac{\sum \text{True positive} + \sum \text{True negative}}{\sum \text{Total population}}$		True positive rate (TPR), Sensitivity, Recall = $\frac{\sum \text{True positive}}{\sum \text{Condition positive}}$	False positive rate (FPR), Fall-out = $\frac{\sum \text{False positive}}{\sum \text{Condition negative}}$	Positive likelihood ratio (LR+) = $\frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) = $\frac{\text{LR+}}{\text{LR-}}$
		False negative rate (FNR), Miss rate = $\frac{\sum \text{False negative}}{\sum \text{Condition positive}}$	True negative rate (TNR), Specificity (SPC) = $\frac{\sum \text{True negative}}{\sum \text{Condition negative}}$	Negative likelihood ratio (LR-) = $\frac{\text{FNR}}{\text{TNR}}$	

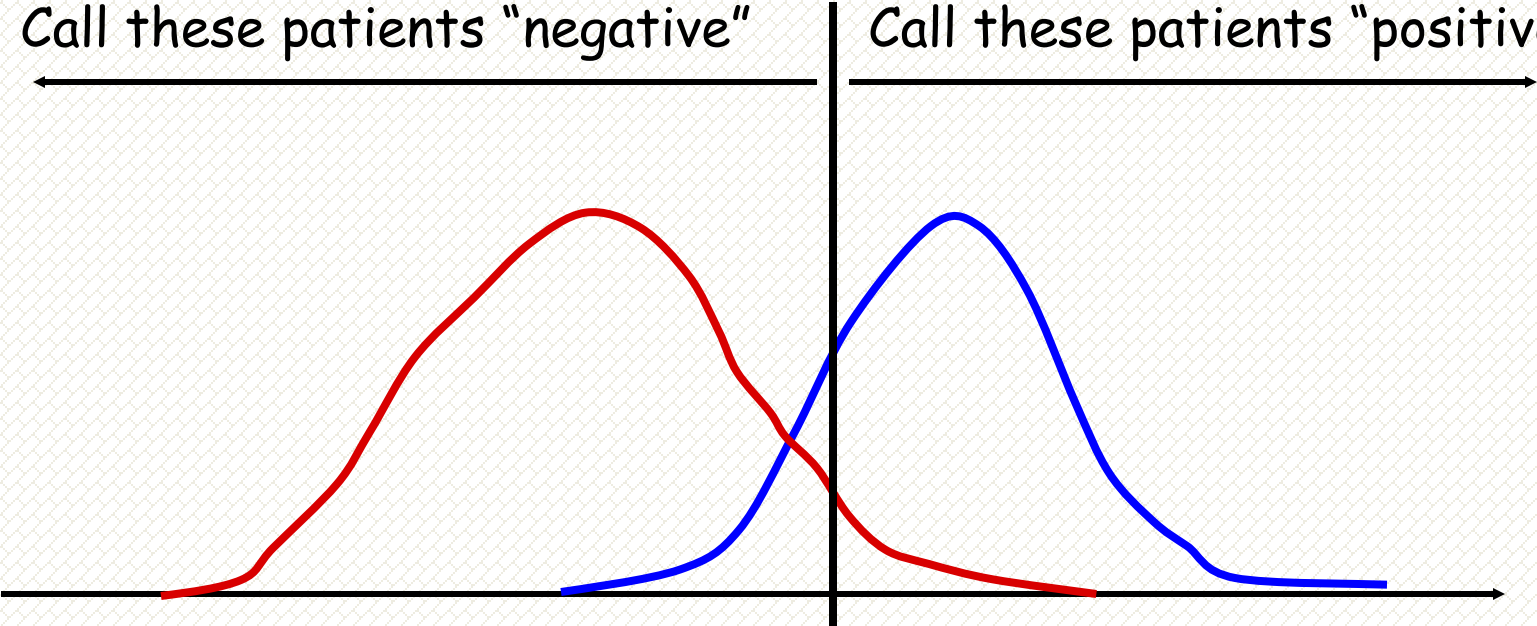
Specific Example



Threshold

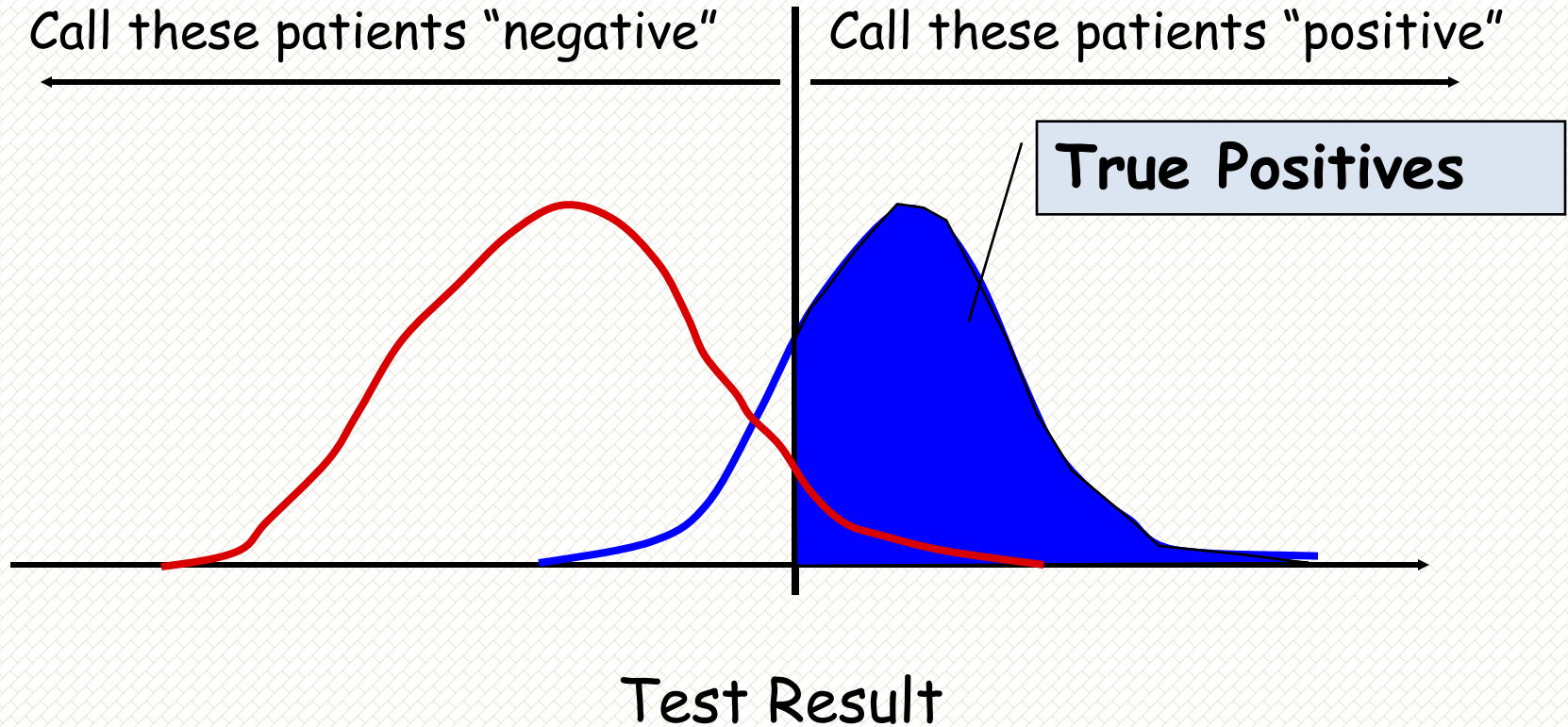
Call these patients "negative"

Call these patients "positive"

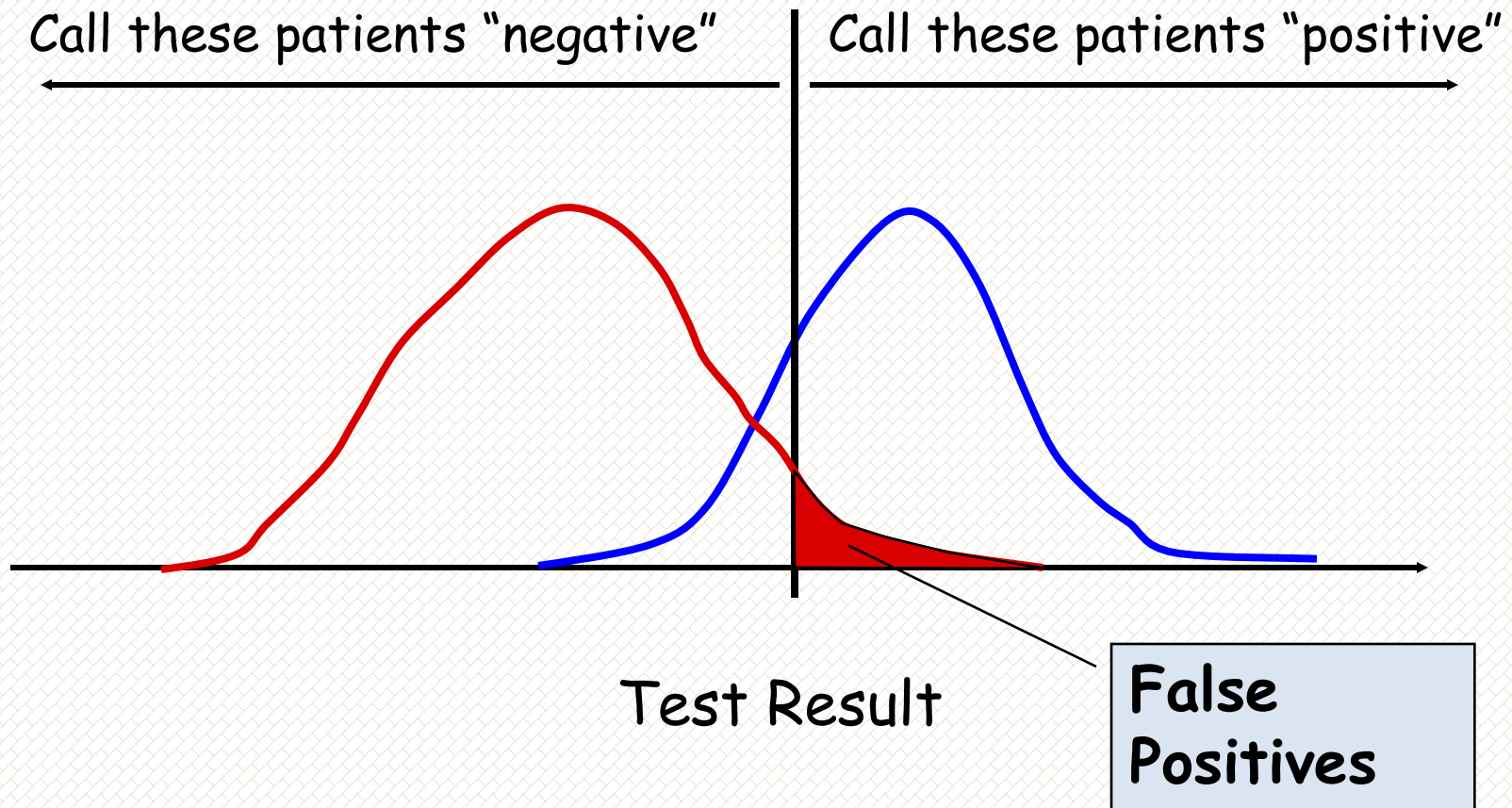


Test Result

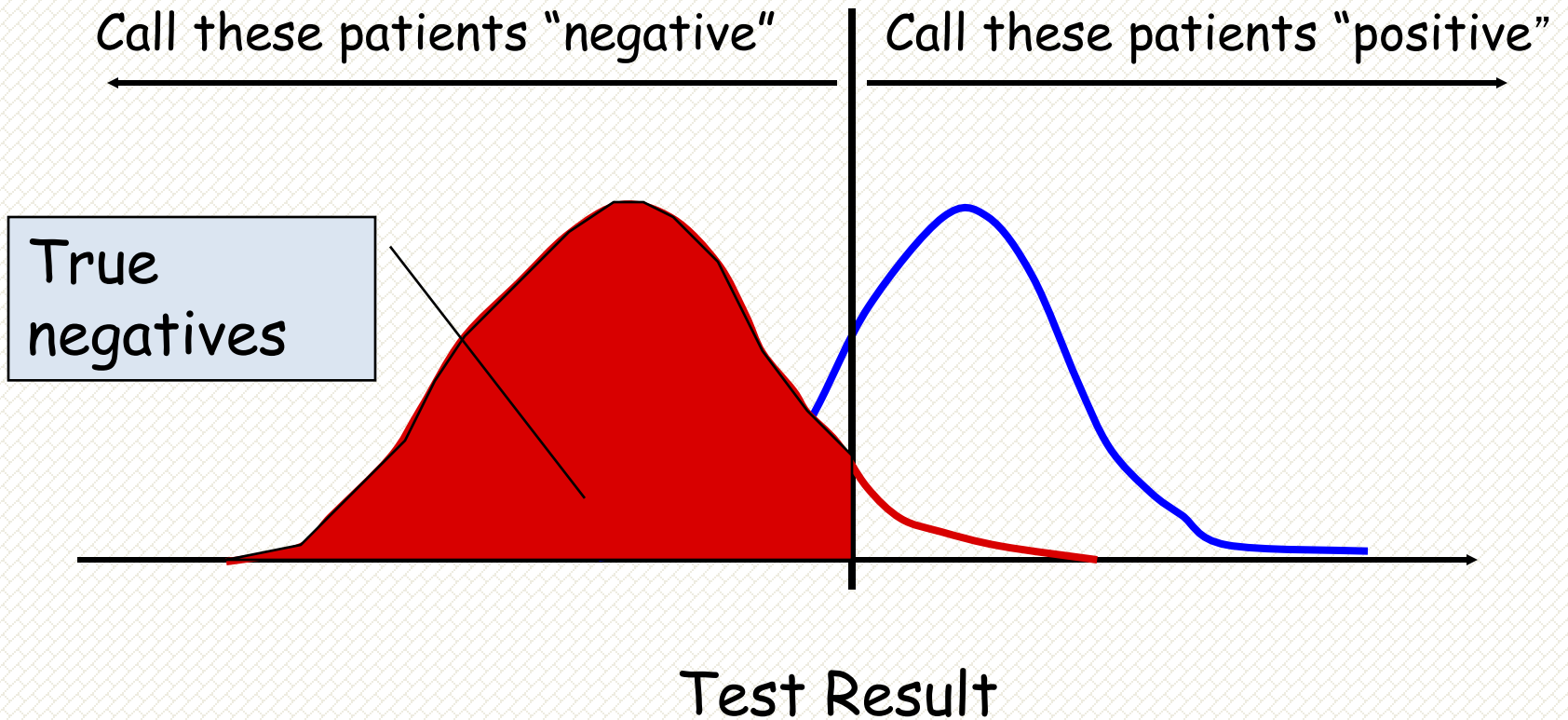
Some definitions ...



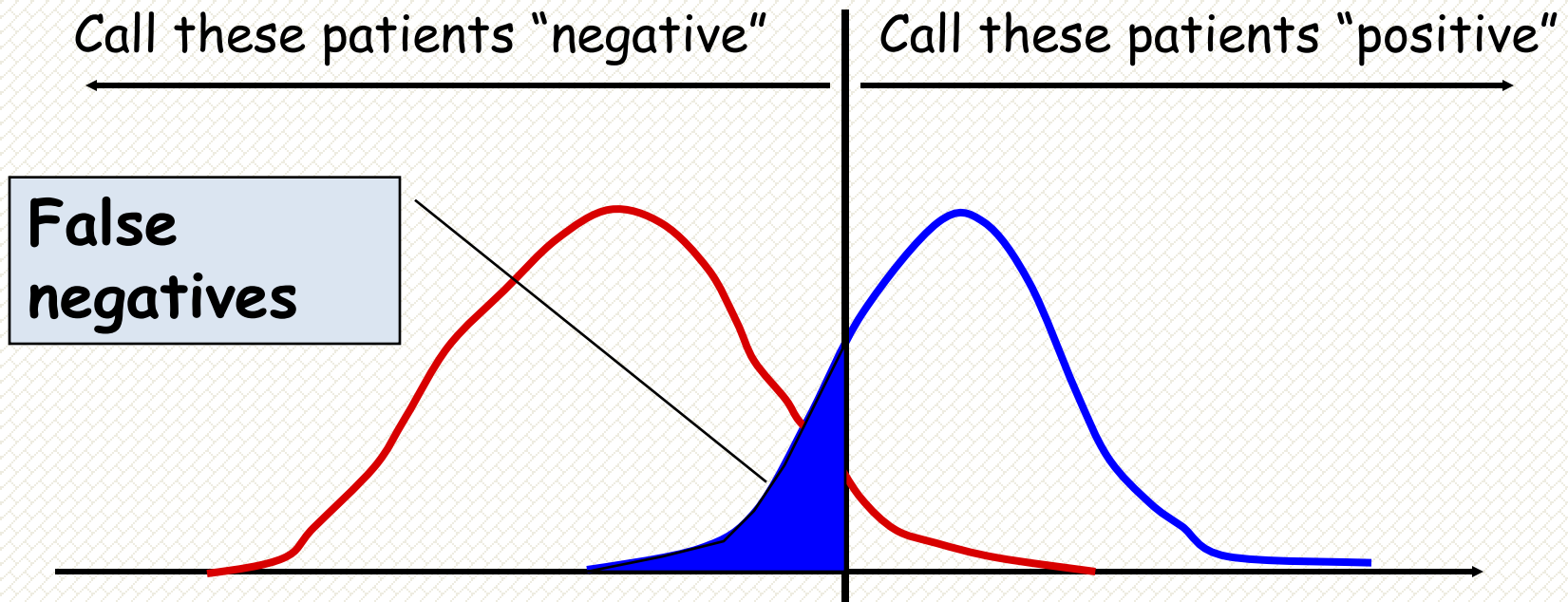
without the disease
with the disease



without the disease
with the disease

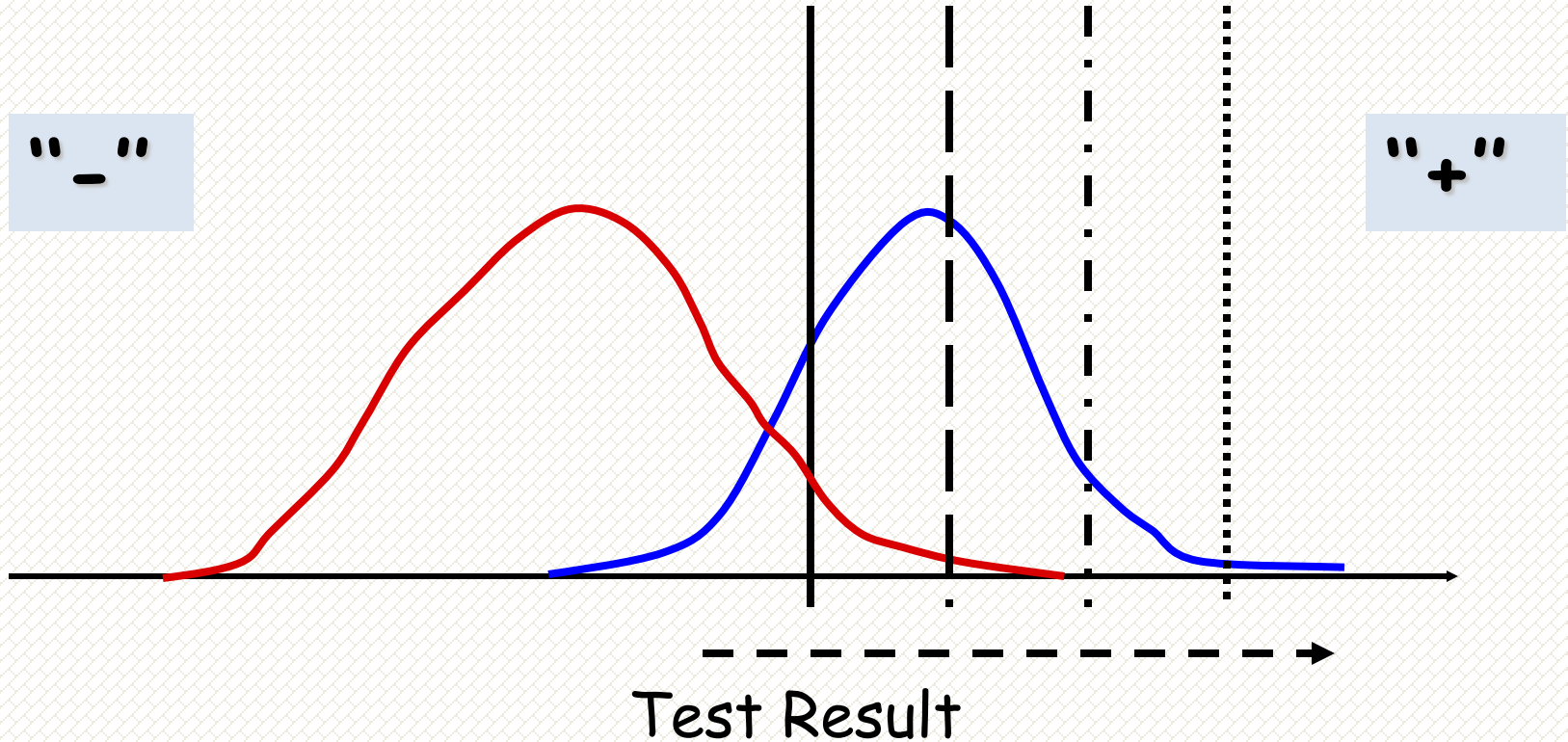


without the disease
with the disease



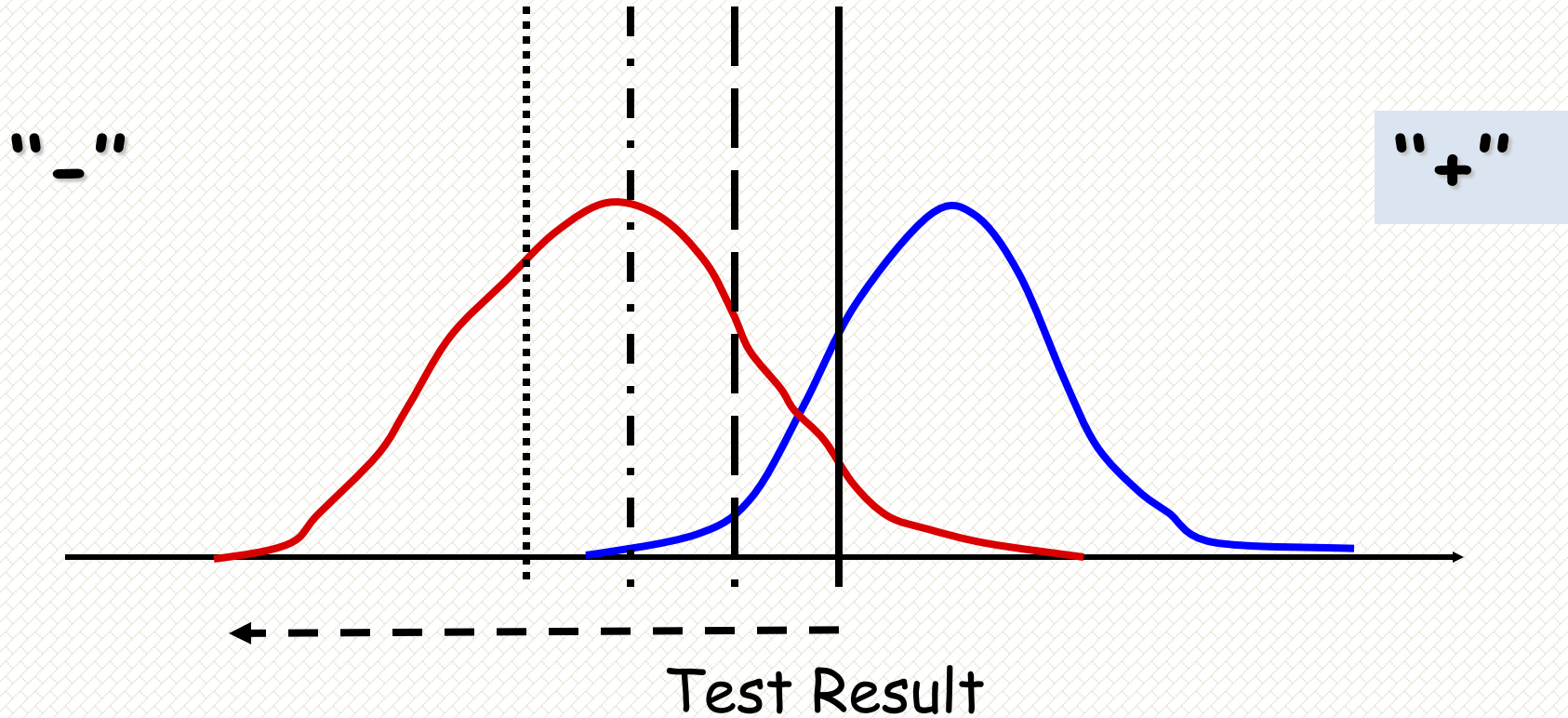
without the disease
with the disease

Moving the Threshold: right



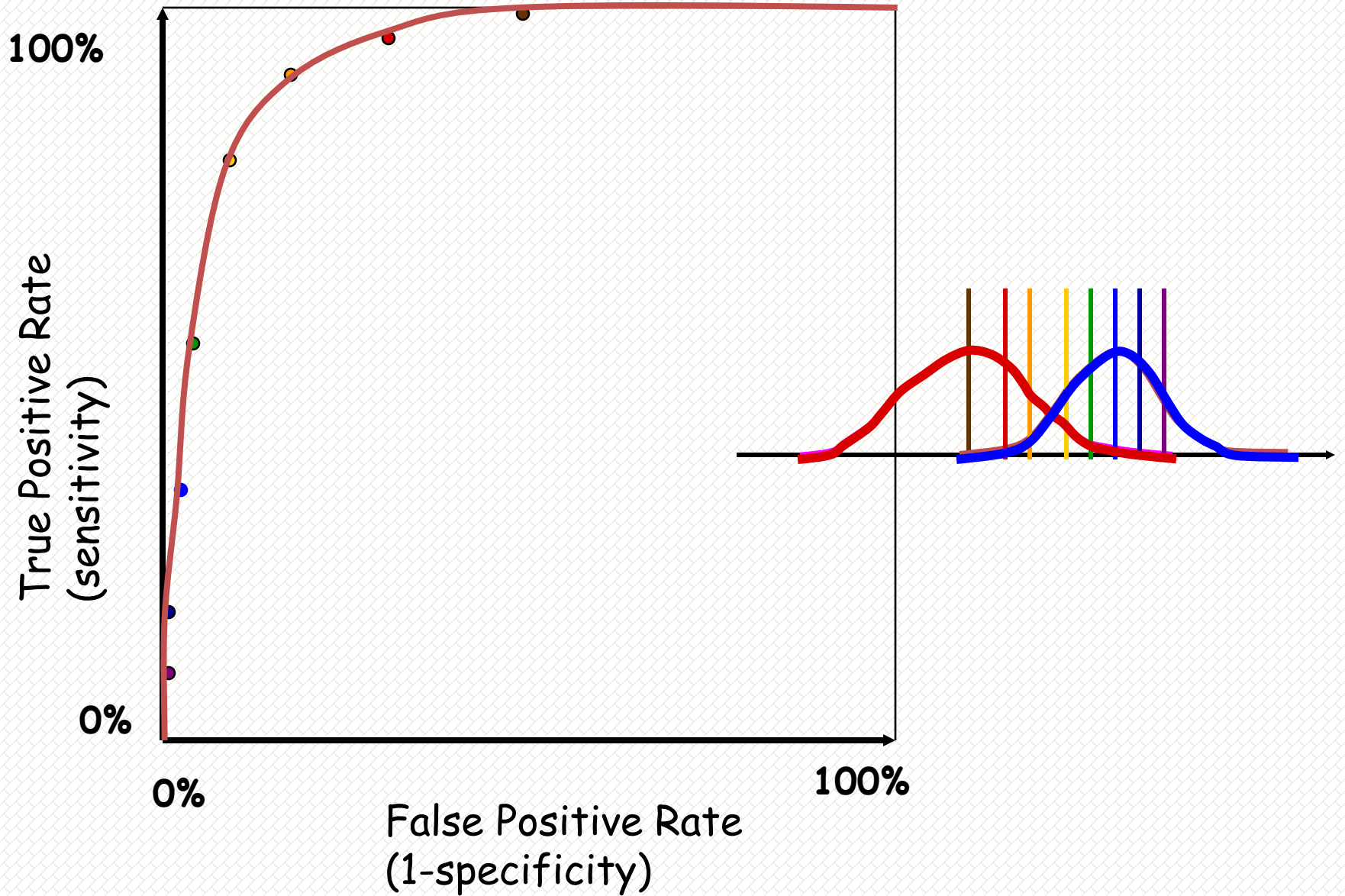
without the disease
with the disease

Moving the Threshold: left



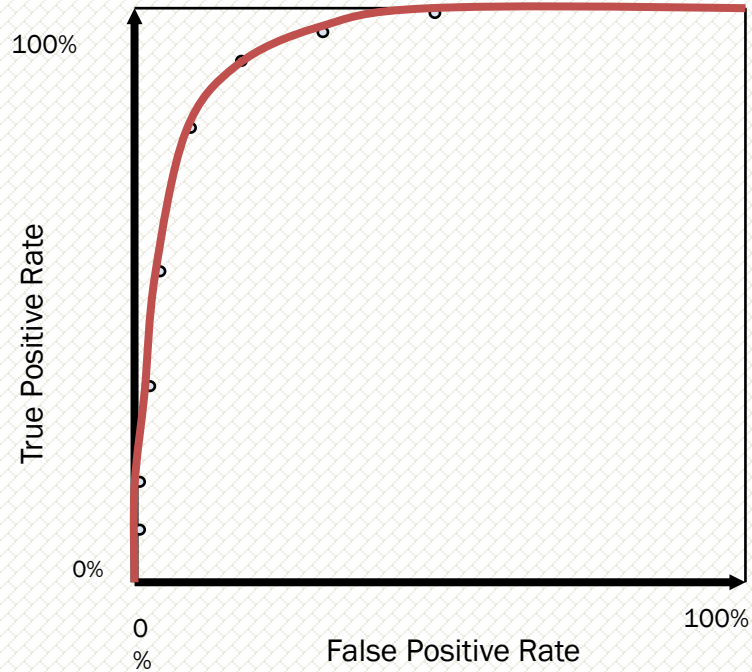
without the disease
with the disease

ROC curve

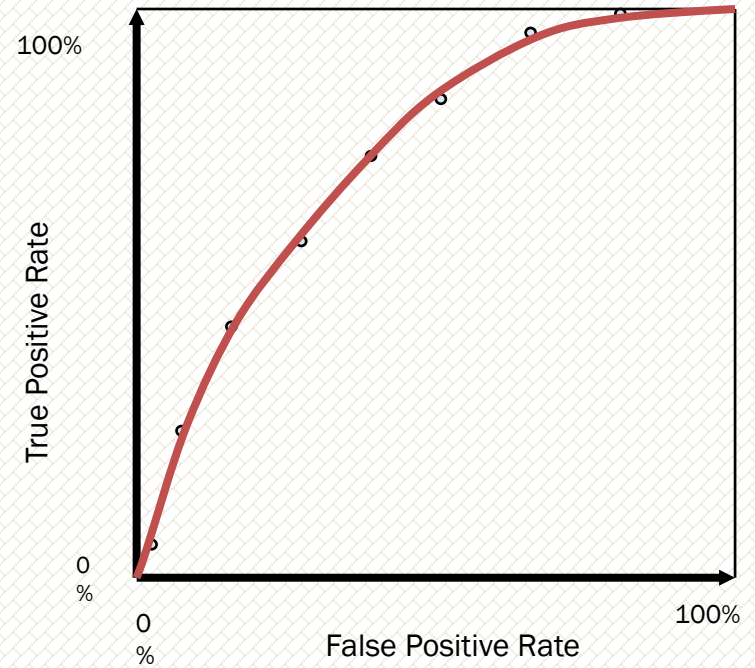


ROC curve comparison

A good test:

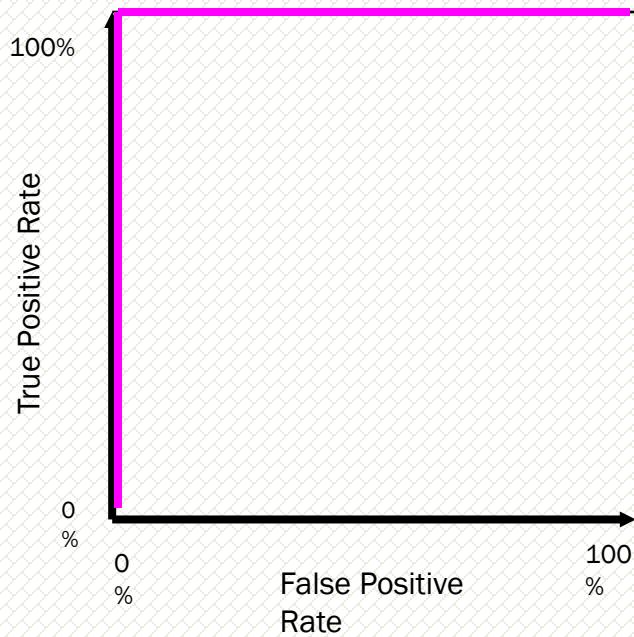


A poor test:



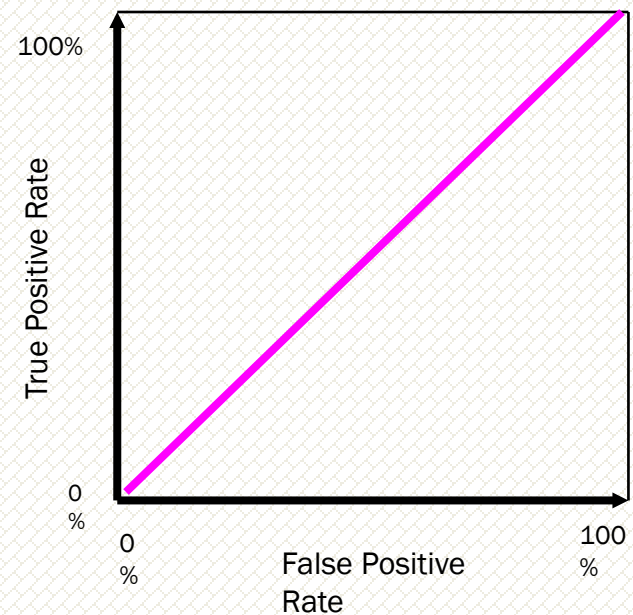
ROC curve extremes

Best Test:



The distributions don't overlap at all

Worst test:

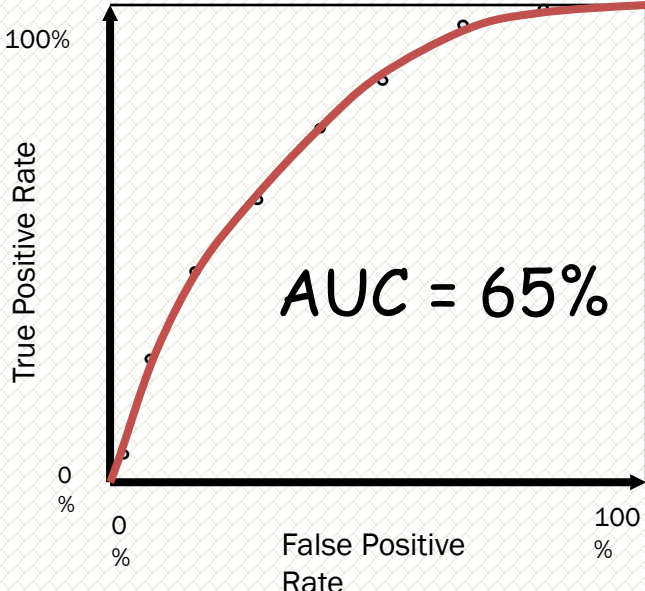
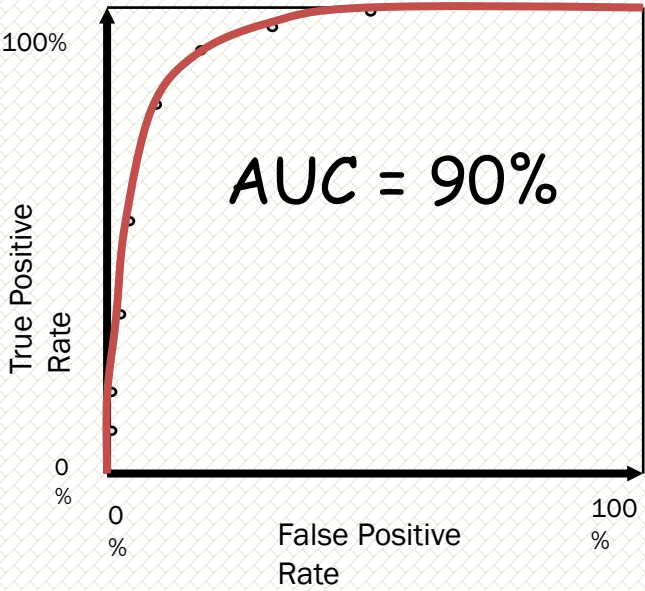
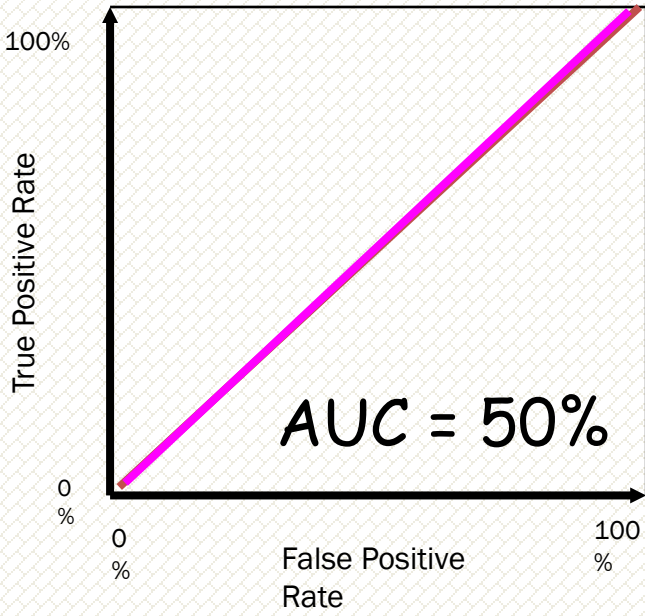
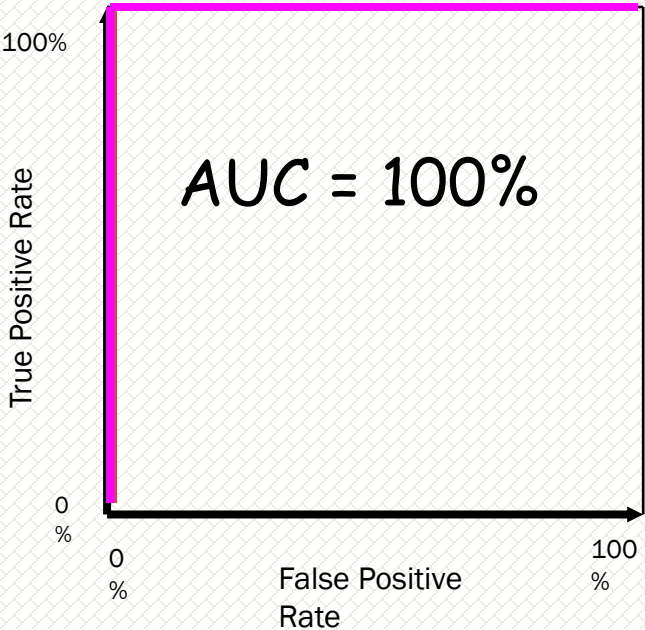


The distributions overlap completely

Area under ROC curve (AUC)

- × *Overall measure* of test performance
- × *Comparisons* between two tests based on differences between (estimated) AUC
- × For continuous data, AUC equivalent to *Mann-Whitney U-statistic* (nonparametric test of difference in location between two populations)

AUC for ROC curves



Interpretation of AUC

- ✘ “AUC can be interpreted as the probability that the test result from a randomly chosen diseased individual is more indicative of disease than that from a randomly chosen nondiseased individual”:

$$P(X_i \geq X_j | D_i = 1, D_j = 0)$$

- ✘ So can think of this as a nonparametric distance between disease/nondisease test results

Problems with AUC

- × *No clinically relevant meaning*
- × A lot of the area is coming from the range of *large false positive* values, no one cares what's going on in that region (need to examine restricted regions)
- × The curves might *cross*, so that there might be a meaningful difference in performance that is not picked up by AUC

Embedded Feature selection with SVM

Recursive Feature Elimination: a type of backward feature elimination

1. Train the classifier (optimize the weights w_i with respect to error function J).
2. Compute the ranking criterion $((w_i)^2)$ for all features
3. Remove the feature with smallest ranking criterion.

NOTE: for the Abeel et al. paper, drop 20% features at each iteration by default.

- If features are removed one at a time, there is also a corresponding **feature ranking**.
- However, the features that are top ranked (eliminated last) are not necessarily the ones that are individually most relevant. Only taken together the features of a subset F_m are optimal in some sense.

Embedded Feature selection with SVM

A linear SVM essentially consists of a separating hyperplane in the input space.

-> The absolute values of the weights of each dimension in the hyperplane can be regarded as the contribution (importance) of each dimension (feature) to the multivariate decision of the hyperplane.

Feature ranking with Support Vector Machine- Recursive Feature Elimination

Algorithm SVM-train

Inputs: Training examples $\{x_1, x_2, \dots, x_k, \dots, x_\ell\}$ and class labels $\{y_1, y_2, \dots, y_k, \dots, y_\ell\}$.

$$\left\{ \begin{array}{l} \text{Minimize over } \alpha_k: \\ J = (1/2) \sum_{hk} y_h y_k \alpha_h \alpha_k (x_h \cdot x_k + \lambda \delta_{hk}) - \sum_k \alpha_k \\ \text{subject to:} \\ 0 \leq \alpha_k \leq C \quad \text{and} \quad \sum_k \alpha_k y_k = 0 \end{array} \right. \quad \Rightarrow \quad \begin{array}{l} D(x) = w \cdot x + b \\ w = \sum \alpha_k y_k x_k \quad \text{and} \\ b = \langle y_k - w \cdot x_k \rangle \end{array}$$

Outputs: Parameters α_k .

Feature ranking with Support Vector Machine- Recursive Feature Elimination

Algorithm SVM-RFE:

Inputs:

Training examples

$$X_0 = [x_1, x_2, \dots, x_k, \dots, x_\ell]^T$$

Class labels

$$y = [y_1, y_2, \dots, y_k, \dots, y_\ell]^T$$

Initialize:

Subset of surviving features

$$s = [1, 2, \dots, n]$$

Feature ranked list

$$r = []$$

Repeat until $s = []$

Restrict training examples to good feature indices

$$X = X_0(:, s)$$

Train the classifier

$$\alpha = SVM\text{-train}(X, y)$$

Compute the weight vector of dimension length(s)

$$w = \sum_k \alpha_k y_k x_k$$

Compute the ranking criteria

$$c_i = (w_i)^2, \quad \text{for all } i$$

Find the feature with smallest ranking criterion

$$f = \text{argmin}(c)$$

Update feature ranked list

$$r = [s(f), r]$$

Eliminate the feature with smallest ranking criterion

$$s = s(1:f-1, f+1:\text{length}(s))$$

Output:

Feature ranked list r .

Ensemble Feature Selection

Idea: Aggregate the feature rankings provided by the single feature selectors into a final consensus ranking.

Consider an ensemble E consisting of s feature selectors,

$$E = \{F_1, F_2, \dots, F_s\},$$

Assuming that each F_i provides a feature ranking

$$\mathbf{f}_i = (f_i^1, \dots, f_i^N),$$

which are aggregated into a consensus feature ranking \mathbf{f} by weighted voting:

Ensemble feature selection

weighted voting:

$$f^l = \sum_{i=1}^t w(f_i^l)$$

where $w(\cdot)$ denotes a weighting function. If a *linear aggregation* is performed using $w(f_i^l) = f_i^l$, this results in a sum where features contribute in a linear way with respect to their rank.

- Weights can be used to incorporate prior knowledge.

Ensemble of linear SVM and SVM-RFE feature selection methods

Ensemble feature selection applied by Abeel et al.

Starting from a particular training set, i.e. one of the 500 subsamplings containing 90% of the data,

- Generate a diverse set of RFE feature selections.
 - -> Because the RFE procedure is deterministic, this is done by generating different sample sets using the particular training set.
 - -> random sampling with replacement from the the particular training

Ensemble feature selection applied by Abeel et al.

Ensemble EFS consisting of t feature selectors,

$$EFS = \{F_1, F_2, \dots, F_t\},$$

then we assume each F_i provides a

$$\text{feature ranking } \mathbf{f}_i = (f_i^1, \dots, f_i^N),$$

where f_i^j denotes the rank of feature j in bootstrap i .

A general formulation for the **ensemble ranking \mathbf{f}** , obtained by summing the ranks over all bootstrap samples is as follows:

$$\mathbf{f} = \left(\sum_{i=1}^t w_i(f_i^1), \dots, \sum_{i=1}^t w_i(f_i^N) \right)$$

Results

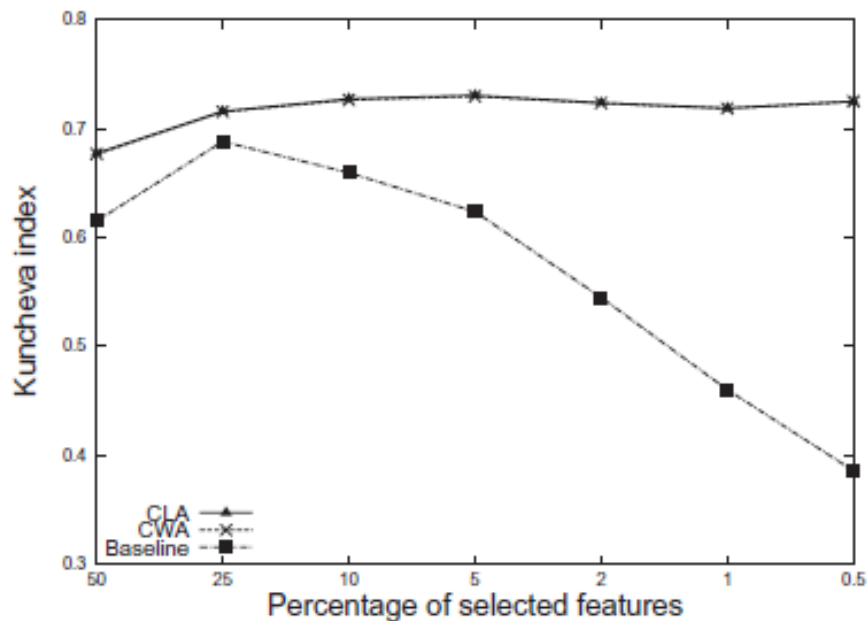


Fig. 1. Stability of the baseline method (original RFE) and the ensemble methods for prostate. We used 40 bootstraps and RFE with $E=20\%$.

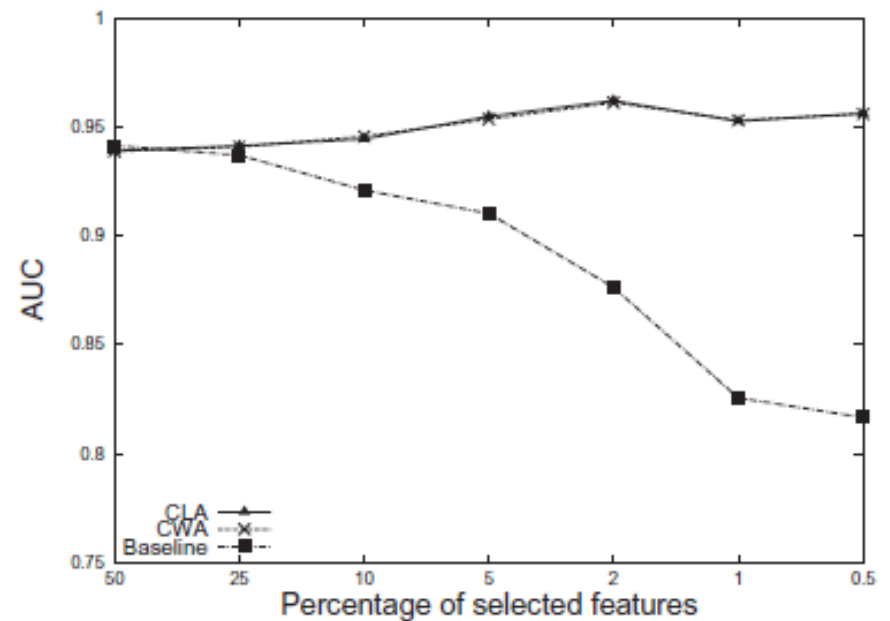


Fig. 2. Classification performances of the baseline method (original RFE) and the ensemble methods for prostate. We used 40 bootstraps and RFE with $E=20\%$.

Results: Changing numbers of bootstrap

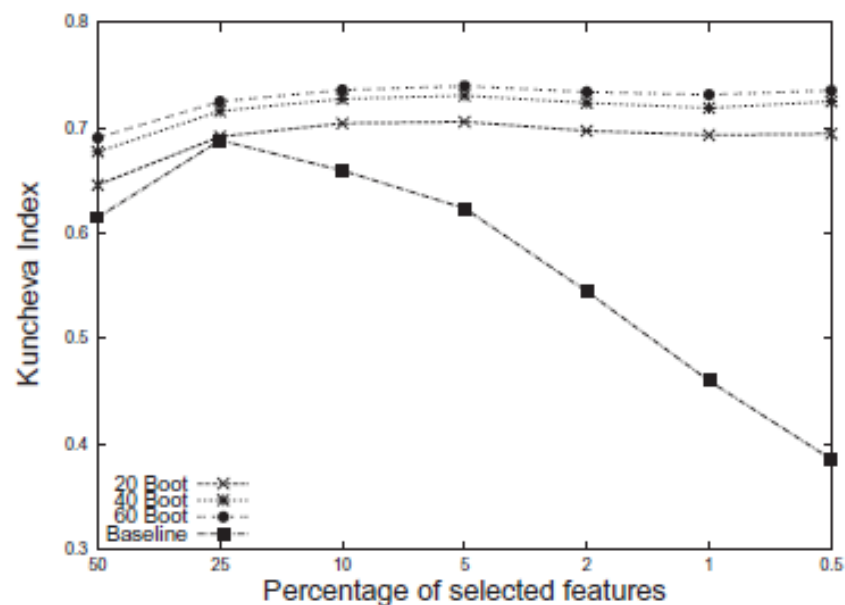


Fig. 4. Stability for several numbers of bootstrap rounds for the construction of an ensemble signature for prostate. We used the CLA aggregation method and eliminated 20% of the features at each iteration of RFE. The baseline is the original RFE on the full training sets without bootstrap.

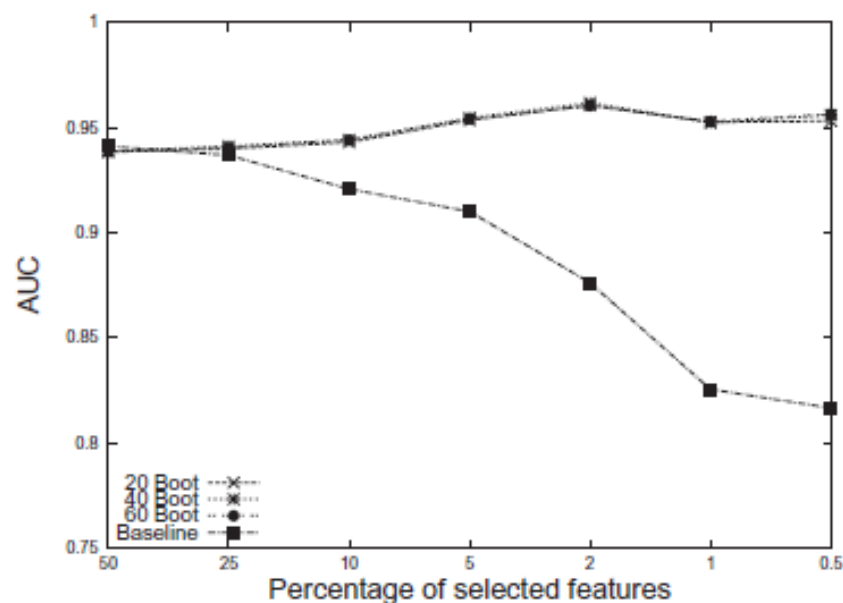


Fig. 5. Classification performances for several numbers of bootstrap rounds for the construction of an ensemble signature for prostate. We used the CLA aggregation method and eliminated 20% of the features at each iteration of RFE. The baseline is the original RFE on the full training sets without bootstrap.

Results: varying number of features to eliminate during RFE

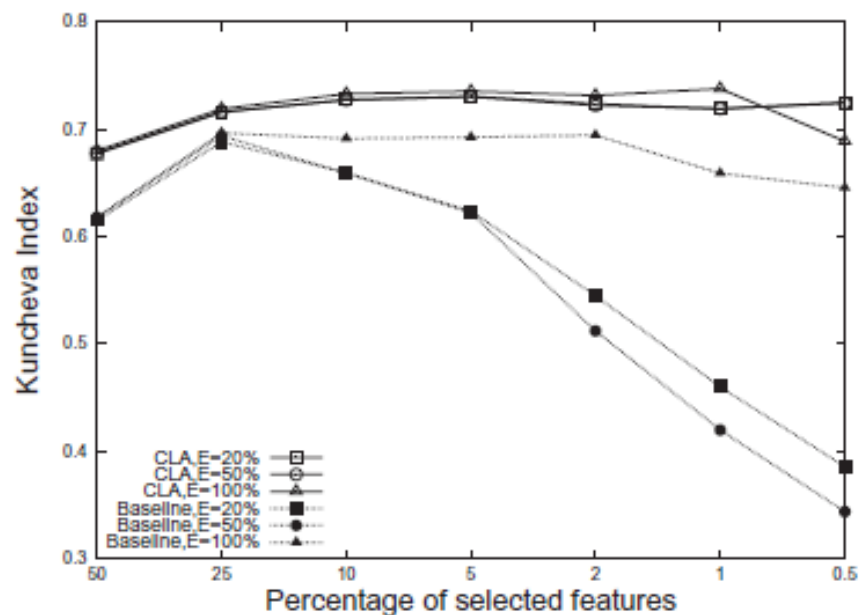


Fig. 6. Stability with regard to a varying number of features to eliminate during RFE. Results represent the CLA aggregation and constructed using 40 bootstrap samples.

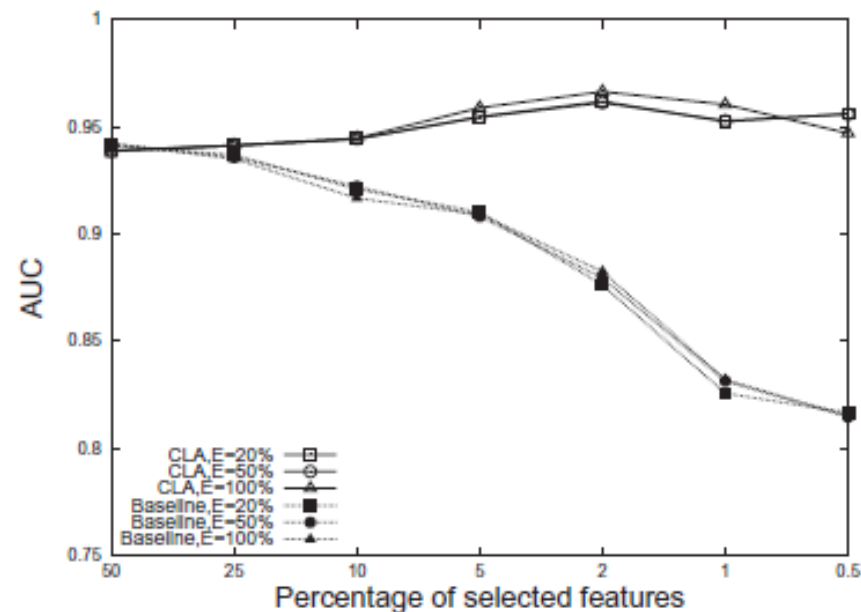


Fig. 7. Classification performance with regard to a varying number of features to eliminate during RFE. Results represent the CLA aggregation method and were constructed using 40 bootstrap samples.