Image Analysis in Neuroinformatics

Pattern Classification

Pavan Ramkumar Xavier Mínguez Bernal

Table of contents

Introduction

Classification of objects into a number of categories

Aim

classification of objects into a number of categories or classes

based on

statistical information priori knowledge

extracted from the classified or described patterns.

The design cycle

Detection

Modeling	Sensoring

Pre-processing

Segmentation	Denoising

Feature extraction

Feature generation	Feature selection

Classifier design

Training samples	Generalization

Post-processing

System evaluation	Cost & risks

Modeling

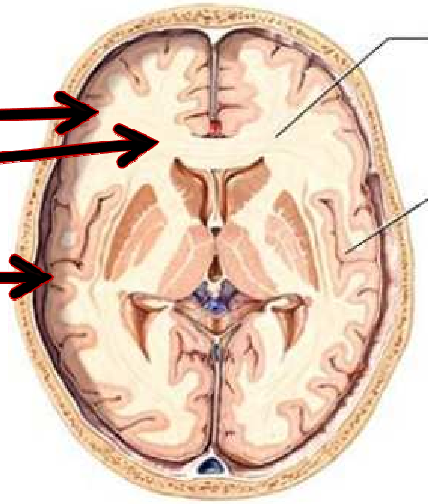
1. Definition of the different classes whose the objects are going to be fit in

Example:

Segmentation of brain structure using structural MRI and PET

4 classes

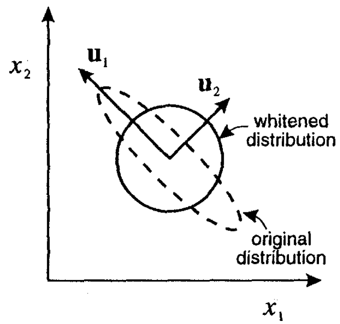
- Gray matter
- White Matter
- CSF (Cerebro-Spinal Fluid)
- Background pixels



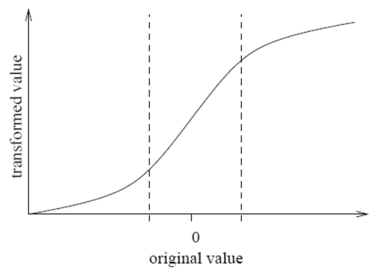
Pre-processing

- Segmentation
- Denoising
- Outliers
- Values spread in different dynamic ranges

Normalization



- Linear rescaling
- Whitening
- Softmax



Features

- Identify the measurable quantities that make the classes distinct from the others

- All the features used in the classification form the feature vector

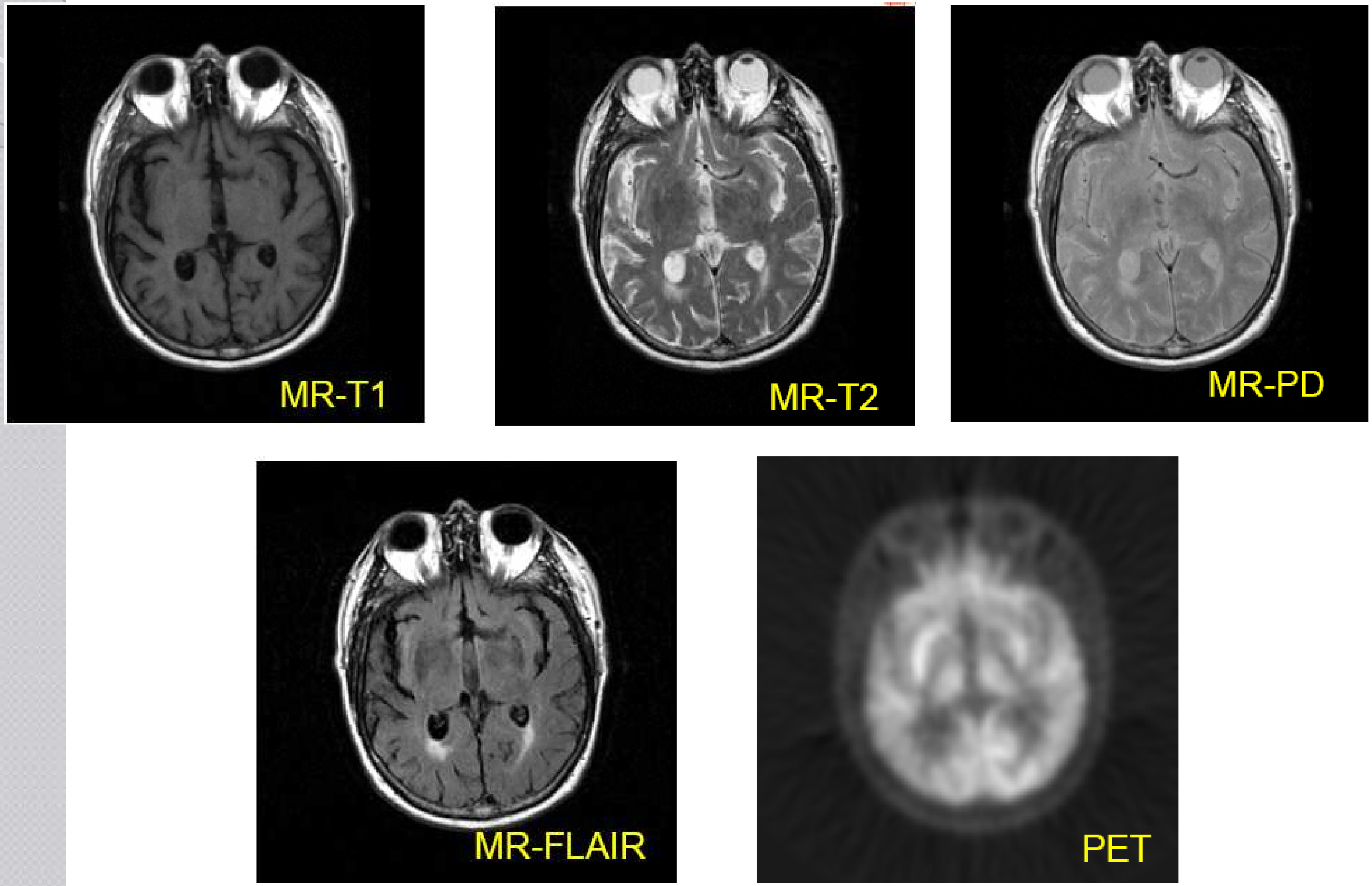
$$\underline{x} = [x_1, x_2, x_3 \dots x_l]^T$$

- Each of the feature vectors identifies uniquely a single pattern (object)

Features selection

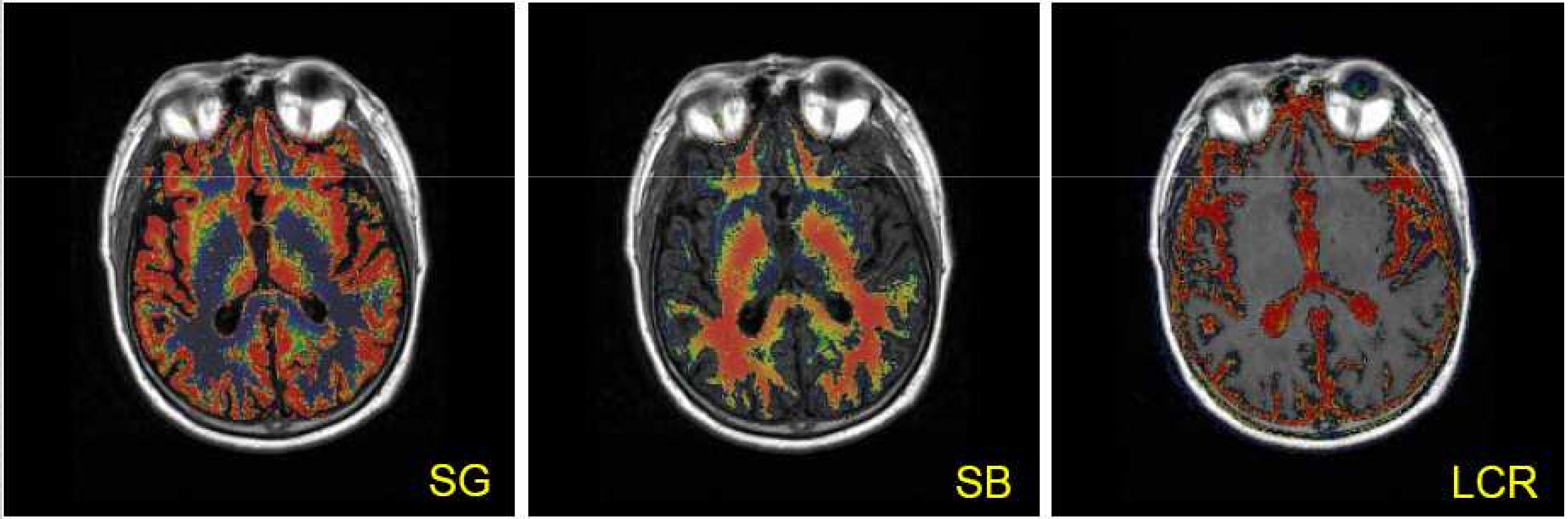
- How many features to select for a given problem, given set of training patterns?
- Curse of dimensionality: Over-training. Classifier may learn more than generalizable aspects of the dataset.
- Features may be ranked by PCA and selected based on classification accuracy

Features in the example

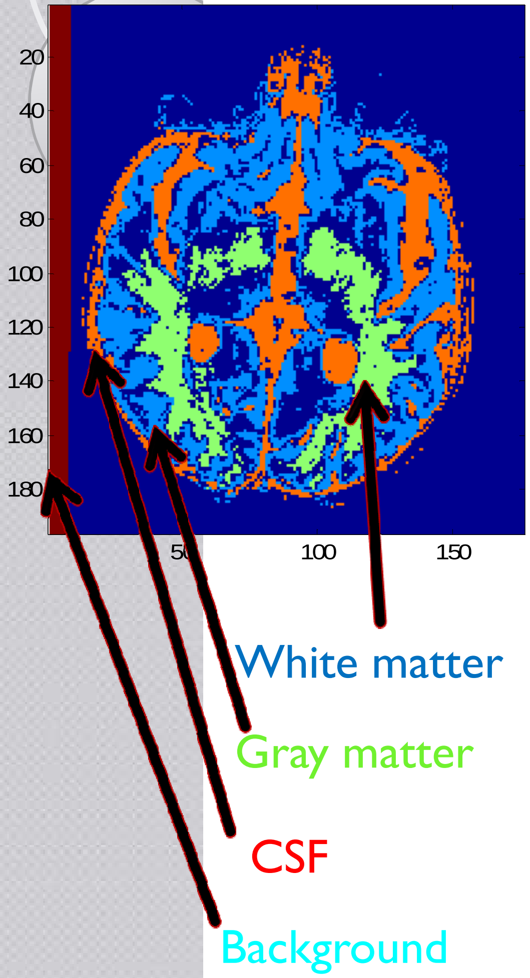


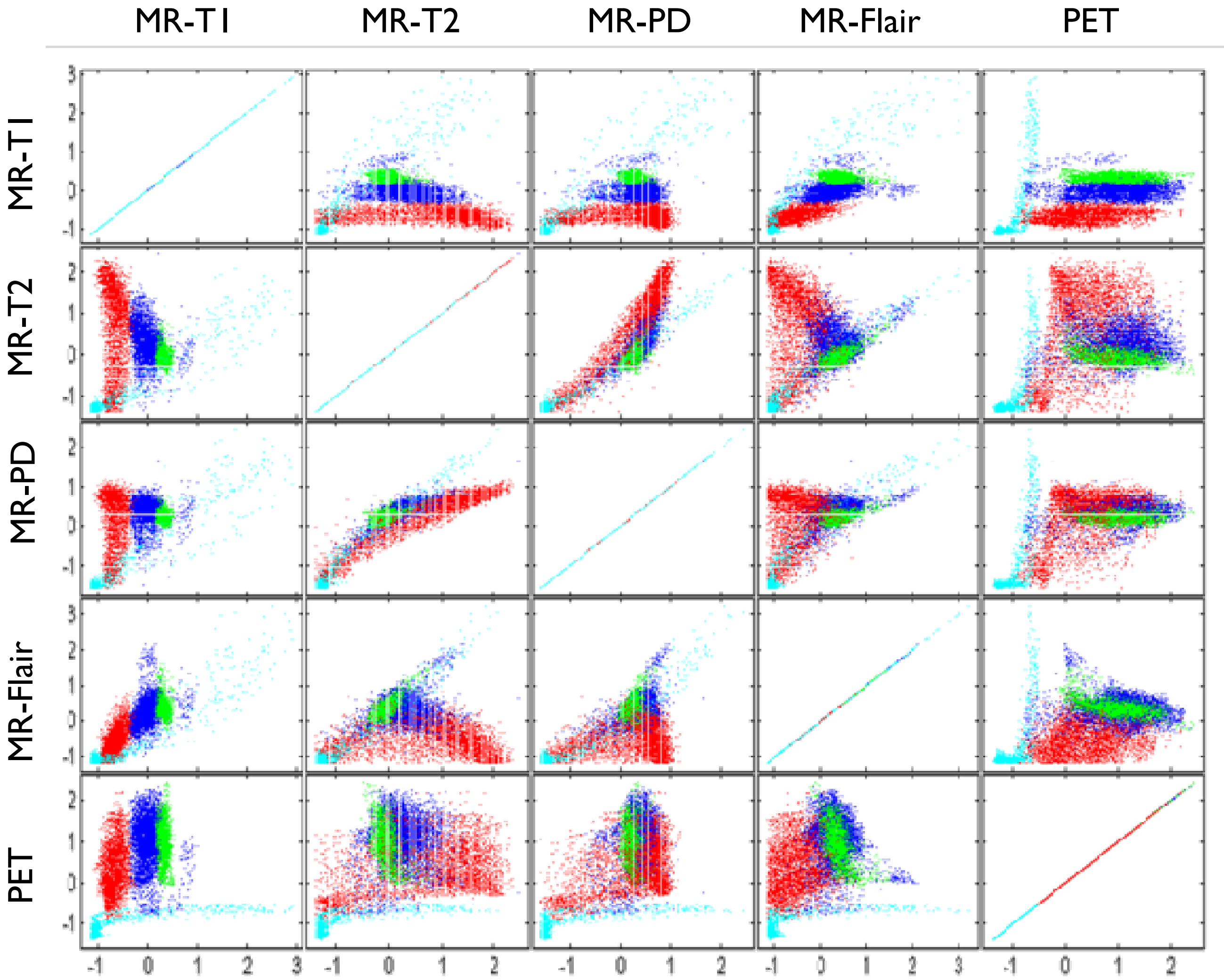
Database construction: labeling

Likelihood for each pixel to belongs to each class



Database and space of features





Discriminant functions

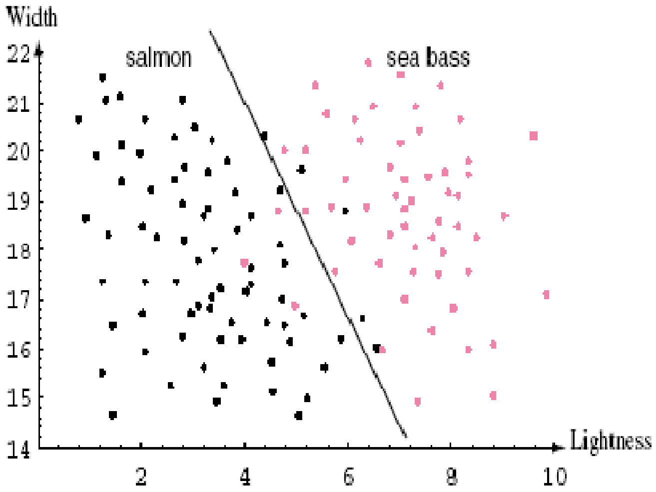
- Generate optimal decision boundaries
- Apply discriminant functions to a feature vectors of unknown classes to classify them

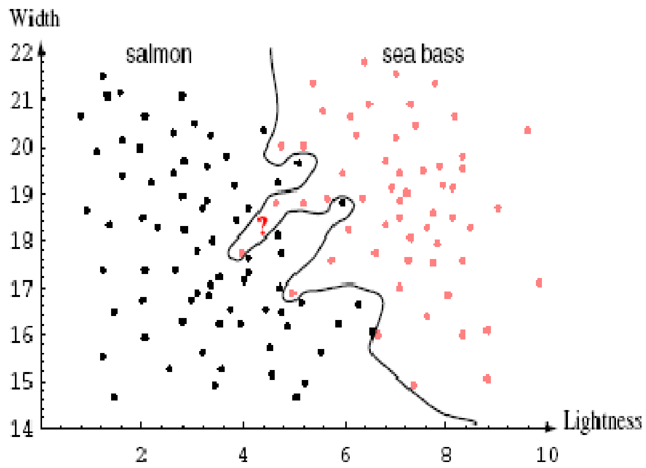
$$g_i(\underline{x})$$

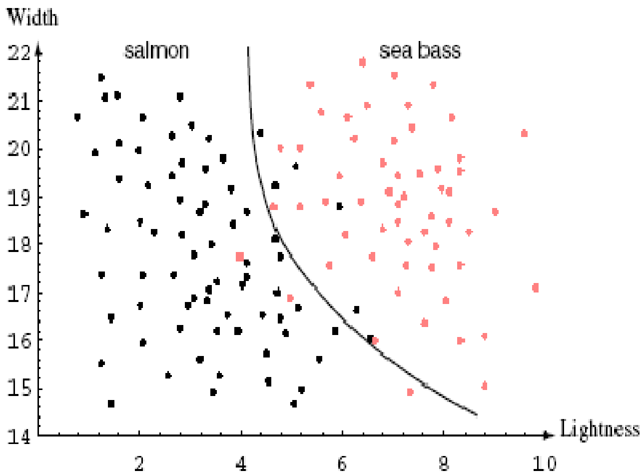
- The classifier is said to assign a feature vector $\underline{\mathbf{x}}$ to class C_i if

$$g_i(\underline{x}) > g_j(\underline{x}) \quad for\, all\; j \neq i$$

Discriminant functions II







Training and Testing (Bootstrapping)

- Training: Process by which classifier learns the properties of each class so that it may discriminate an arbitrary pattern. A set of patterns with apriori known classes are used for this purpose.

- Testing: Process by which performance of the classifier is evaluated. Patterns with known classes are given to the trained classifier in order to test it's accuracy.

- How does splitting into training and testing sets affect classifier design?
- Leave one out method
  - (N-1) training, 1 test. Repeated with each pattern as test pattern
  - Gives least biased estimate of classification accuracy

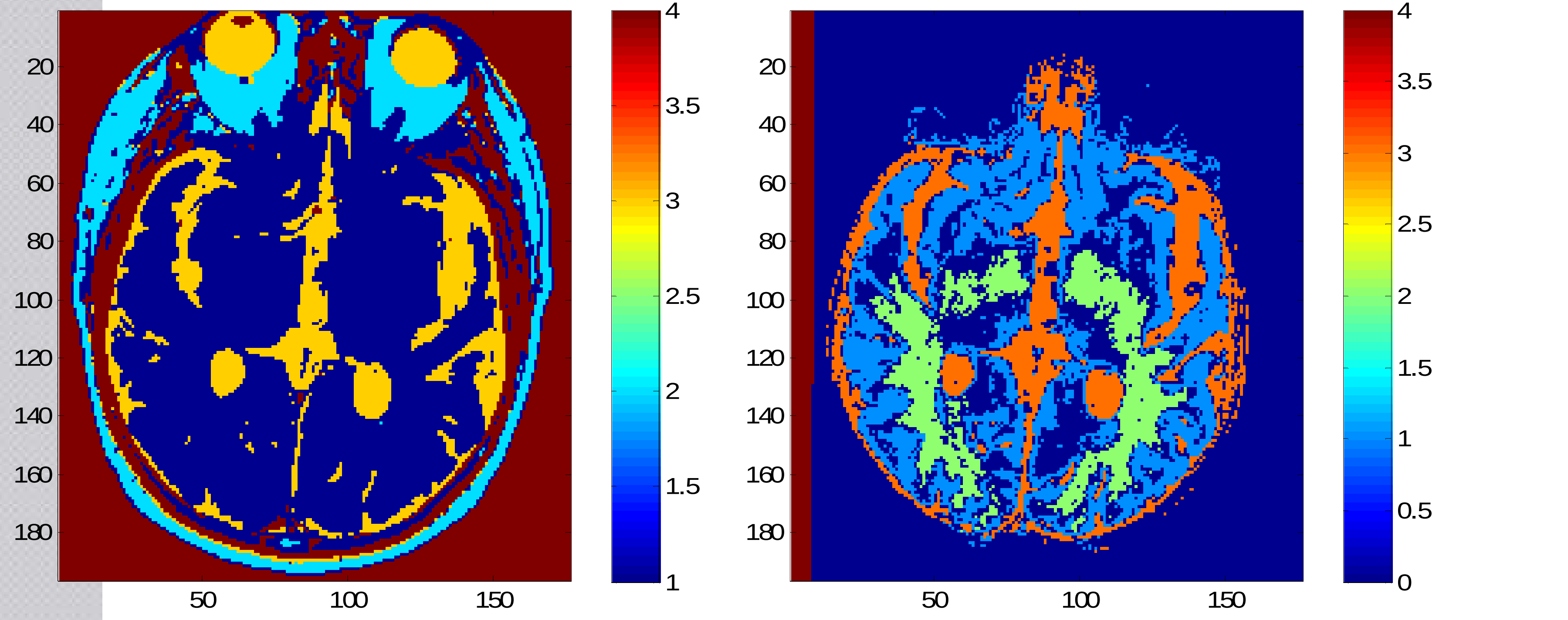
Unsupervised classification: k-means

- No prior training information is available and we have to classify the set of feature vectors into unknown number of categories

K-means algorithm

1. Assign each object randomly to one of the clusters
   $k = 1, \ldots K$.
2. Compute the means of each of the clusters
3. Reassign each object to the cluster with the closest mean
4. Return to step 2 until the means of the clusters do not change anymore

Unsupervised classification: k-means results



Probabilistic Models

- How do we know that heuristic methods are optimal?
- We use a (multivariate) statistical model for the distribution of patterns
- Training dataset is used to estimate model parameters

Prior, Likelihood, Posterior

- Prior: Apriori probability of occurrence of class. Given by

$$P(C_i)$$

- Likelihood: Given that a pattern ***x*** belongs to class C_i what is the probability of occurrence of ***x***. Given by

$$P(x / C_i)$$

- Posterior: Given a pattern x, what is the probability that it belongs to class C_i

$$P(C_i / x)$$

Statistical Decision

- Classification penalty: A loss L_{ij} is defined as the loss incurred by a classifier by labelling a pattern coming from class i into class j. L_{ii} = 0, or some fixed operational cost.

- Conditional Average Risk: Average loss in classifying pattern ***x*** to class C_j. Given by:

$$R_j(x) = \sum_{i=1}^{M} L_{ij} P(C_i / x)$$

Bayes' Classifier

- Minimizes the conditional average risk
- Posterior probability can be calculated from the Bayes' formula

$$P(C_i / x) = \frac{P(C_i)P(x / C_i)}{P(x)}$$

- Denominator is independent of C_i and may be left out in the minimization.

Bayes' classifier II

- In the two class problem, the two risks may be written as:

$$r_1(x) = L_{11}P(C_1/x) + L_{21}P(C_2/x)$$
$$r_2(x) = L_{12}P(C_1/x) + L_{22}P(C_2/x)$$
$$x \in C_1 \text{ if } r_1(x) < r_2(x)$$

- Thus, the Bayes' Classifier can be written as a series of discriminant functions

Naïve Bayes' Classifier

- If the loss is assumed to be the same for all erroneous classifications, then we can define

  $L_{ii} = 0, L_{ij} = 1$, i.e. $L_{ij} = 1 - \delta_{ij}$

- Then, the Bayes' classifier essentially minimizes

$$p(x) - p(x / C_i) P(C_i)$$

- Or maximizes

$$p(x / C_i) P(C_i)$$

- Which is the posterior probability. This is the same as MAP estimate. However, by Bayes' theorem this is the same as

$$p(C_i / x) P(x)$$

- Since p(x) is not dependent on C_i, this is the same as the ML estimate.

Naïve Bayes' with normal pdfs

- For class with multivariate normal pdfs with mean m_i and covariance $\mathbf{C_i}$

$$p(x / C_i) = \frac{1}{(2\pi)^{n/2} |C_i|^{1/2}} \exp\left[-\frac{1}{2}(x - m_i)^T C_i^{-1} (x - m_i) \right]$$

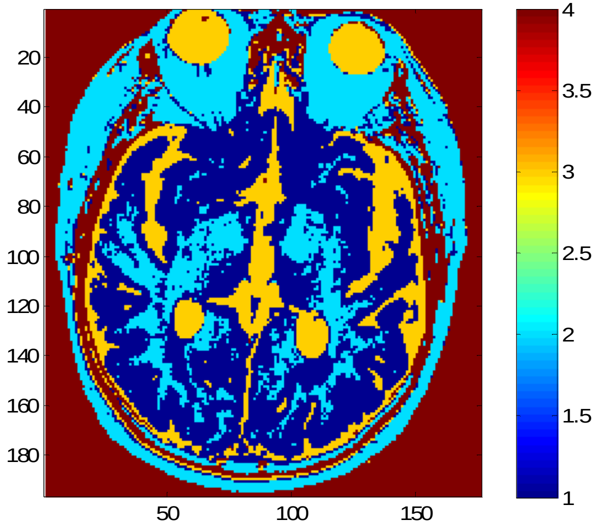
- The log-posterior densities can be written as

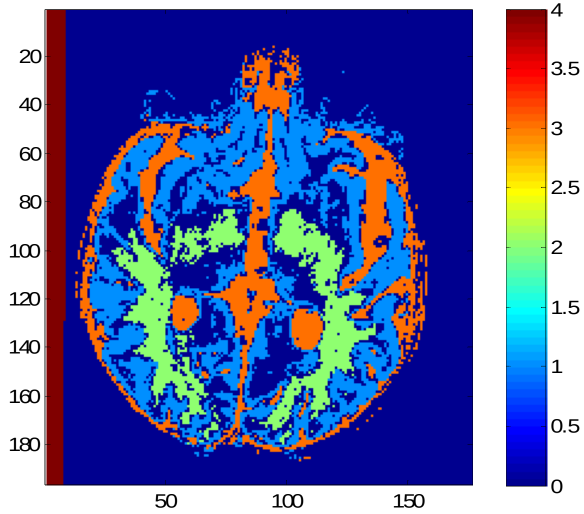
$$d_i(x) = \ln P(C_i) - \frac{1}{2}\ln(C_i) - \frac{1}{2}\left[(x - m_i)^T C_i^{-1} (x - m_i) \right]$$

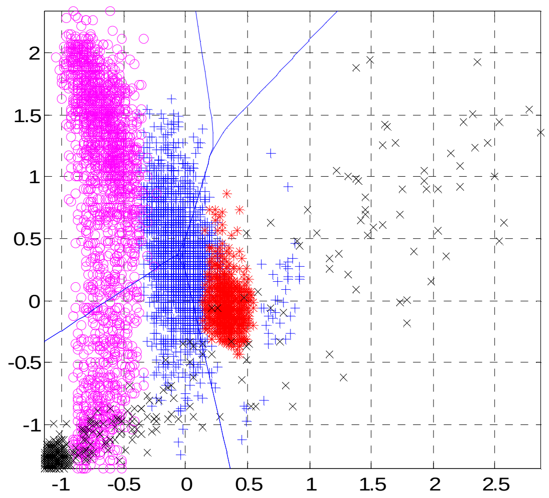
- This takes on a quadratic form (called hyperquadric). If the covariances are equal, i.e. Ci = C for all i, it reduces to a linear

$$d_i(x) = \ln P(C_i) + x^T C^{-1} m_i - \frac{1}{2} m_i^T C^{-1} m_i$$

Naïve Bayes' Results







Error Train= 0.1628

Error Test= 0.1582

Logistic Regression

- Typically used in two class problems, though may be extended to multi class problems.
- Does not directly assign a class, but a membership value to all classes
- Likelihood (or posterior) is expressed as a sigmoid function of the parameter vector, which is then maximized using a training set

$$P(C_i / x) = \frac{1}{1 + \exp(-b_i^T x)}$$

- Since it is non-linear, an iterative algorithm is reqd. to estimate the co-efficients of the model

Nearest Neighbor Classifier

- A vector x of unknown class is assigned to the class of the nearest sample of known class.

$$x \in C_i \text{ if } D(s_i, x) = \min(D(s_j, x)), \text{ j=1,2...N}$$

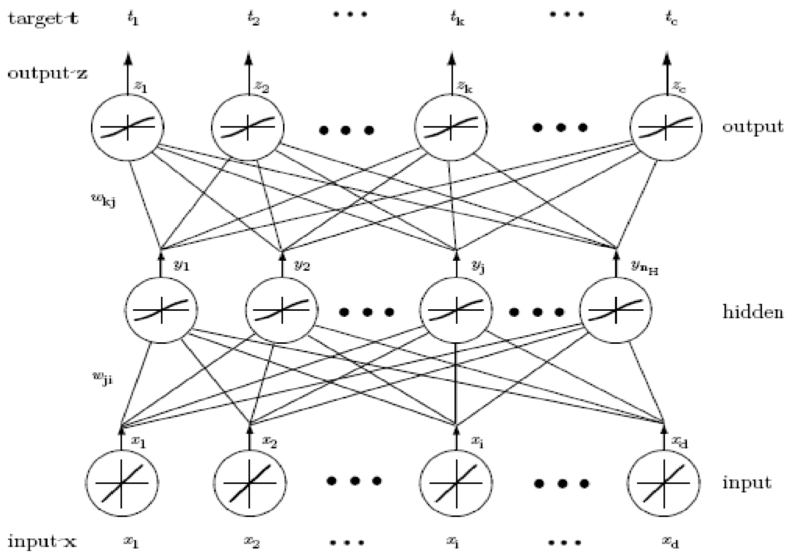
- We rely entirely on the class of one observation. It would be more reliable if we used k nearest samples, and assigned the majority class instead. This is the k-nearest neighbor classifier.

- There are many variants of the k-NN classifier such as the weighted k-NN, in which contributions of samples in the k-neighborhood towards determining the class of the sample are weighted by their distance from the sample.

Nearest Neighbor Classifier Results

Multilayer neural network

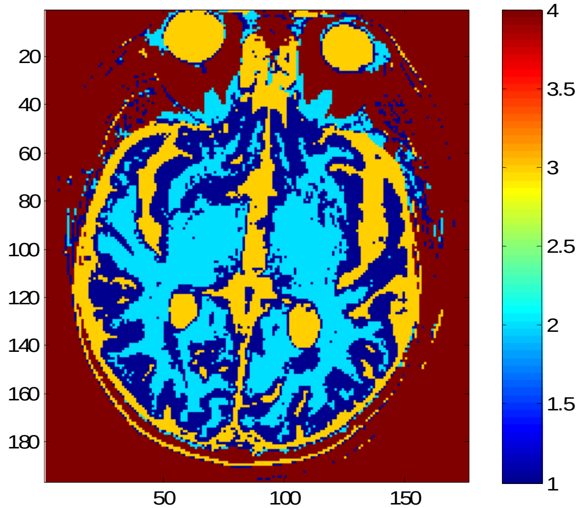
- Interconnected group of artificial neurons that uses a computational model for information processing

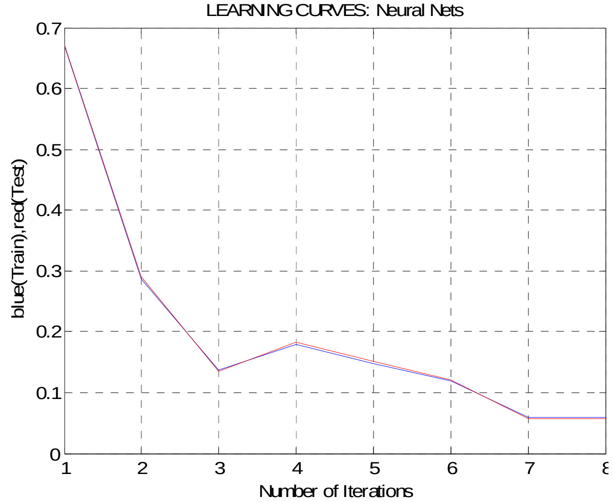


$$g_k(\underline{x}) \equiv \underline{z}_k = f\left(\sum_{j=1}^{nH} w_{kj} f\left(\sum_{i=1}^{d} w_{ji} x_i + w_{j0}\right) + w_{k0}\right) \quad J(\underline{\mathbf{w}}) \equiv \frac{1}{2}\sum_{k=1}^{c}(t_k - z_k)^2 = \frac{1}{2}(\underline{\mathbf{t}} - \underline{\mathbf{z}})^T(\underline{\mathbf{t}} - \underline{\mathbf{z}})$$

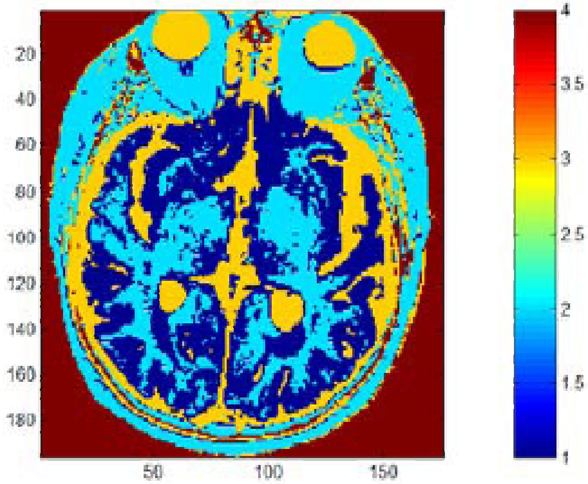
Multilayer neural network: results

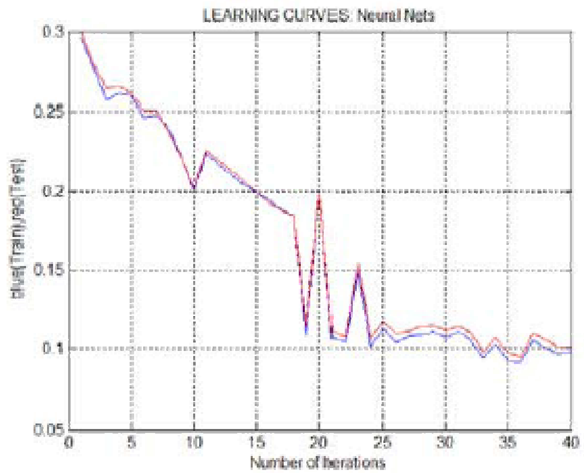
Levenberg-Marquad one hidden layer 5 neurons per layer





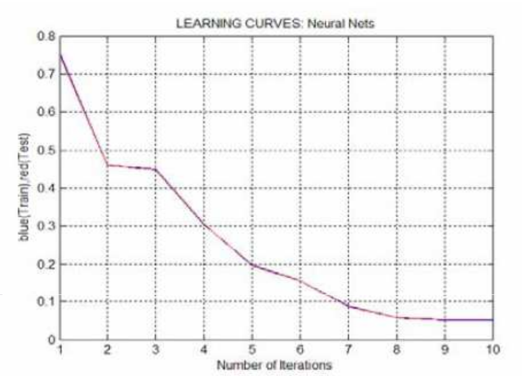
Backpropagation one hidden layer 5 neurons per layer



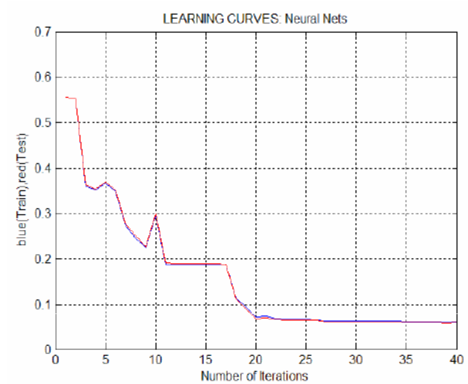


MNN:Topology discussion

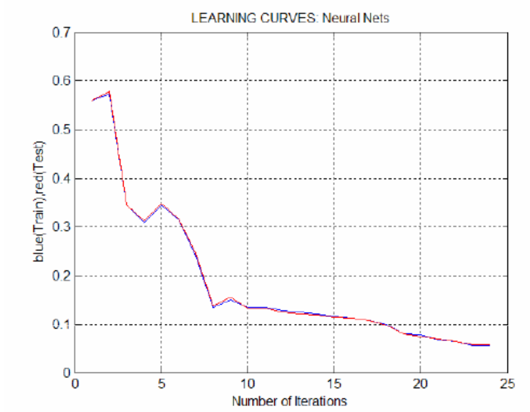
- Network architecture. Choices in the number of hidden layers, units and feedback connections



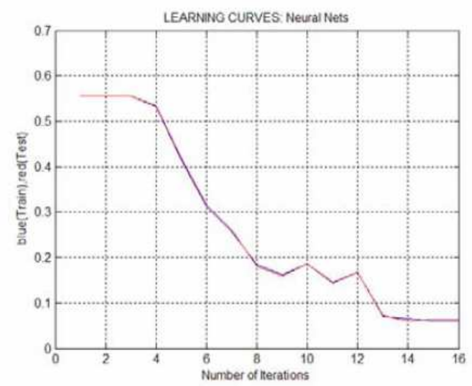
1 hidden layer 10 neurons



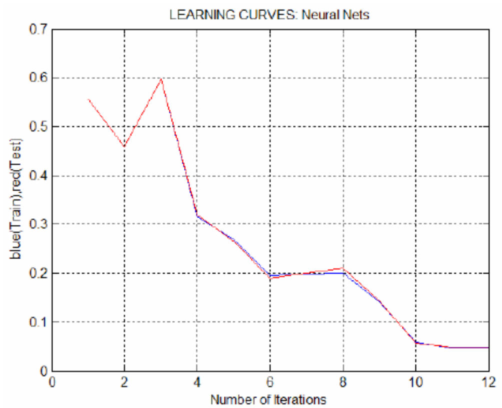
1 hidden layer 2 neurons



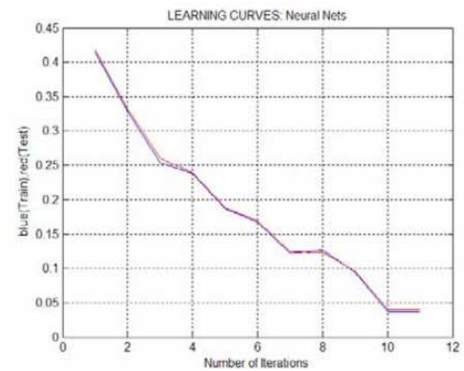
2 hidden layers 5 neurons



3 hidden layer 2 neurons



3 hidden layer 5 neurons



3 hidden layer 15 neurons

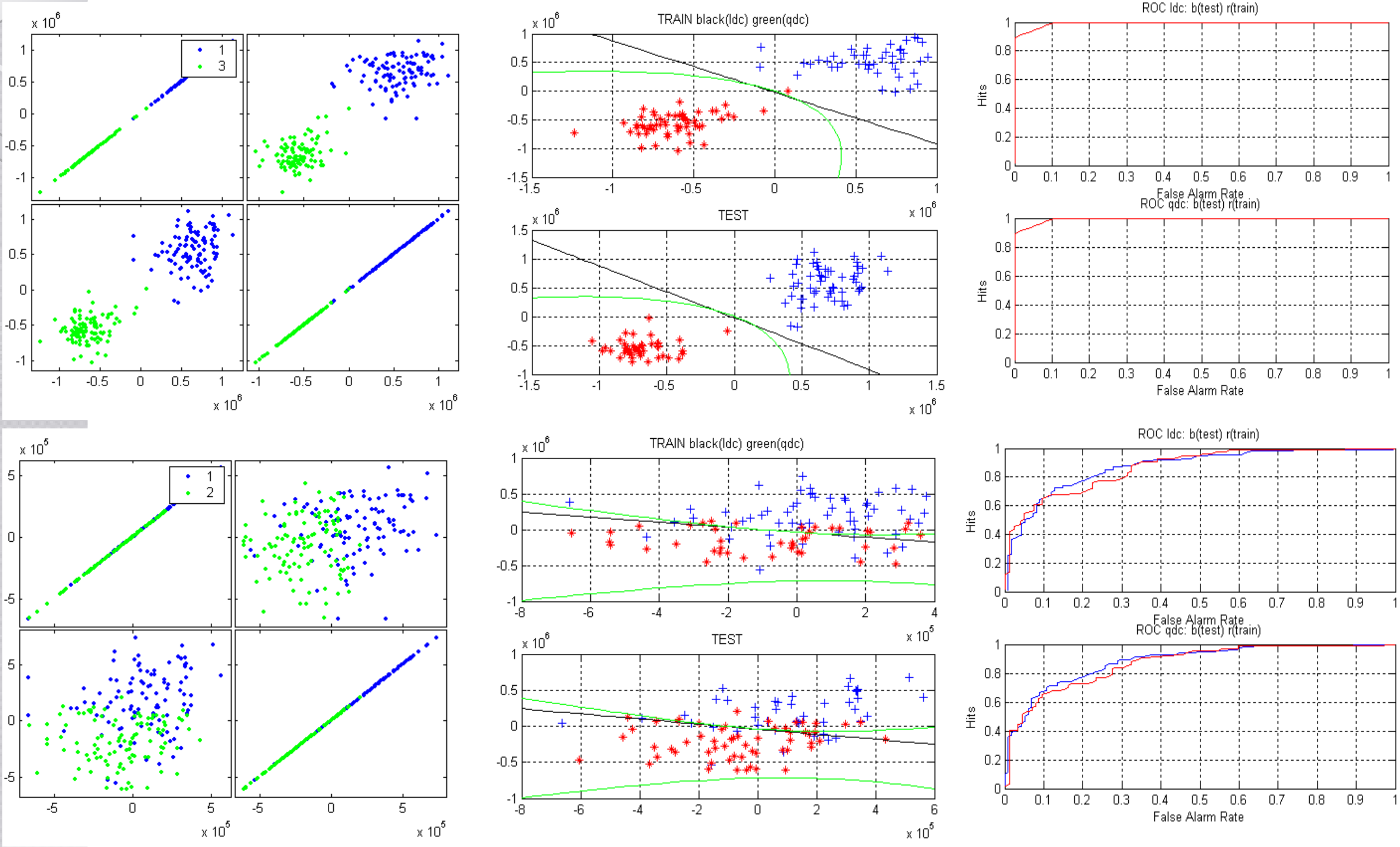
Measures of diagnostic accuracy

- Two class problem: Pathology detector: Abnormal (positive) vs Normal (negative)
- True Positive(TP): 'Hit'; Test is positive, subject has abnormality
- True Negative(TN): Test is negative, subject is normal
- False Negative(FN): 'Miss'; Test is negative, subject has abnormality
- False Positive(FP): Test is positive, subject is normal
- S^+ = True Positive Fraction = $P(T^+/A)$ = (No. of TP)/(No. of A)
- S^- = True Negative Fraction = $P(T^-/N)$ = (No. of TN)/(No. of N)
- $FNF = P(T^-/A)$
- $FPF = P(T^+/N)$
- $Accuracy = S^+P(A) + S^-P(N)$

Receiver Operating Characteristics

- S^+ (sensitivity): Represents how sensitive the classifier is to a TP
- S^- (specificity): Represents how specific the classifier is to a TP
- Ideally we want both to be high so that there are no false negatives or false positives
- The ROC curve maps (FPF, TPF) i.e. $(1 - S^-, S^+)$. The effectiveness of the test is measured by the area under the ROC curve
- More separable the data, closer to ideal ROC curve

ROC illustrations



Statistical Separability of classes

- How far apart are the classes statistically?

- Normalized distance between pdfs
  - D = |m1 – m2|/(σ1 + σ2)
  - In fact the Fischer Linear Discriminant is trained to maximize D
  - Vanishes for identical means

- Jeffries-Matusita Distance

$$Jij = \left\{ \int_x \left[\sqrt{p(x/C_i)} - \sqrt{p(x/C_j)} \right]^2 dx \right\}^{1/2}$$

- Gives theoretical upper and lower bounds on classification error

End of the presentation

Questions round