Statistical Classification

Statistical Classification vs. Regression and FF nets

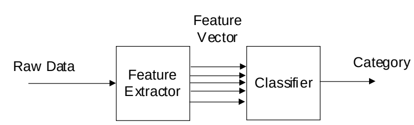
- In **regression or feed-forward nets**, it is assumed that there is an underlying **functional** mechanism, although the exact formulation and parameters of the mechanism may be unknown.
- An attempt is made to find the formulation and parameters that **minimize an error function**, such as MSE.

Statistical Classification vs. Regression and FF nets

- In **statistical classification**, no assumption is made that there is a single mechanism.
- The outcome of classification is always **discrete** (each datum is assigned to one of several classes), whereas with regression or FF nets, it may be continuous.
- Consequently, a **functional** classification is still being assumed, although we know the classification will be **wrong** some of the time.

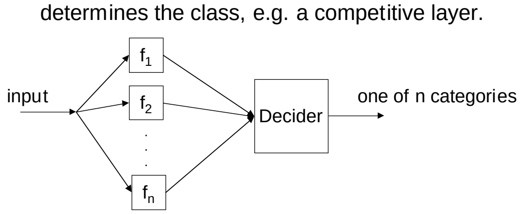
Source of Input

- In general, input is a vector.
- For pattern recognition, it may be a vector of **features** that have been obtained by pre-processing



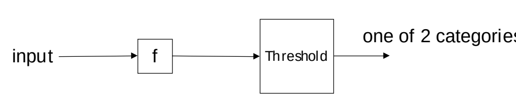
Statistical Classification vs. Regression and FF nets

- Regression and FF nets can be **part of** a statistical classification scheme. But may have to be a final decision layer that determines the class, e.g. a competitive layer.



Statistical Classification vs. Regression and FF nets

- For two classes, we have also used a single regression function plus a threshold.



Statistical Classification Objective

- We know the classification will be **wrong** some of the time.
- The goal is to minimize wrongness, in some sense, which is referred to as the **optimal classification**.

Optimal Classification

- The **optimal classifier** has been shown by statistician R.A. Fisher in 1936 to be one that assigns to each sample x the class c_i with the **highest posteriori probability** $P(c_i \mid x)$:

  $$(\forall j \neq i)\ P(c_i \mid x) > P(c_j \mid x)$$

- Reference: R.A. Fisher, *The use of multiple measurements in taxonomic problems*, Annals of Eugenics, **7**, part II, 179-188, 1936. Also in *Contributions to Mathematical Statistics*, Wiley, 1950.

Optimal Classification

- It can be shown (see CDROM examples) that the neural network or regression approach does ***not*** give and optimal classification in the sense described.

Computing $P(c_j \mid x)$

- $P(c_i \mid x)$ is typically computed using **Bayes' rule**:

  $$P(c_i \mid x) = P(x \mid c_i)\, P(c_i) / P(x)$$

  where $P(c_i)$ and $P(x)$ are the *prior probabilities* of being in class c_i and of the sample being x, respectively and $P(x \mid c_i)$ is the likelihood of drawing sample x as a random member of class c_i.

- For a given x, $P(x)$ can be dropped when comparing across classes for the optimal classification.

Bayes' Rule Interpretation

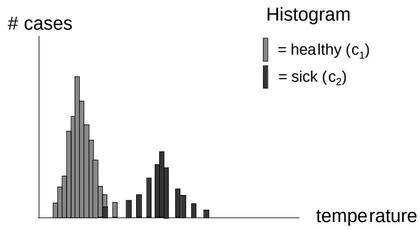
- $P(c_i)$ is the probability of a sample being in class without no information about the sample. This can be estimated from the relative frequency of class occurrences.
- $P(c_i \mid x)$ is the probability of a sample being in class with the identity of the sample known. This is what we'd like to know (compute).
- $P(x \mid c_i)$ is the probability of the sample within a given class. This can be determined from the probability distribution for the class c_i.

Probability Distribution within Classes

- The actual distribution of samples within a class might not be known.
- It is common to make assumptions, such as:
  - Gaussian distribution of $P(x \mid c_i)\, P(c_i)$
  - Distribution mean = sample mean
  - Distribution variance = sample variance

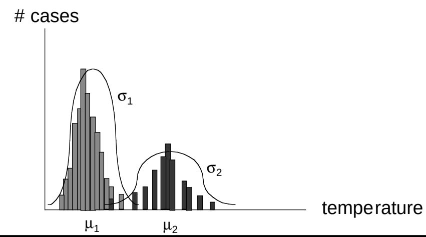
Example
2 classes: healthy & sick
depending on temperature

- Measured samples (temperature, class) :

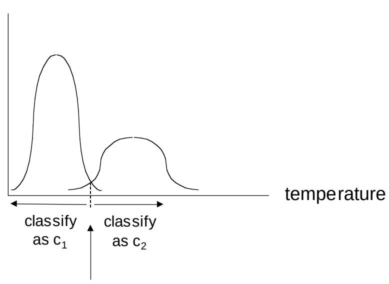


Example
2 classes: healthy & sick
depending on temperature

- Gaussian Estimates (use density fn as mass)
  $P(t \mid c_i)\ P(c_i) = \exp(-(t-\mu_i)^2 / 2\sigma_i^{\ 2})/\mathrm{sqrt}(2\pi\sigma_i)$, i = 1,2



Example
Classification

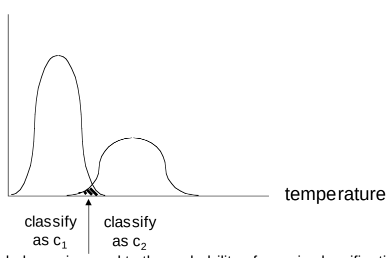


crossover point (called the “Bayesian threshold”)

Classification

- There are two ways to compute the classification for an input:
  - Compute the value of the Gaussian pdf for each class and choose the one that is greater, **or**
  - Solve for the crossover point and compare the value to it. Solving can be done by taking ln of both functions; monotonicity of ln means that the relative order is preserved.

Probability of Error in Classification



The shaded area is equal to the probability of error in classification (in particular, no overlap means no error).

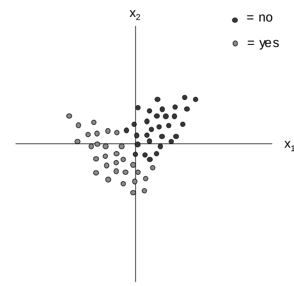
Discriminant Functions

- The role of the Gaussian Estimates
  $P(t \mid c_i)\ P(c_i) = \exp(-(t-\mu_i)^2 / 2\sigma_i^{\ 2})/\mathrm{sqrt}(2\pi\sigma_i)$
  as means for comparison to determine classification can be **generalized** to a set of functions, one for each class. The class is determined as the function having the largest value for the given argument.
- As a set, such functions are called **discriminant functions**.
- One way to compute such a set of functions is by neural networks.

>1 Dimension

- The previous example was one-dimensional (temperature).
- The extension of discriminant functions to >1 dimension is obvious.
- However, it is less obvious to see what is going on graphically:
  - There is a **joint probability** distribution for each class.
  - The classes can **overlap** in multiple dimensions.

2-D Classification



Joint Probability Distributions

- **Example**: product of two Gaussians:

  $P((x,y) \mid c_i)\, P(c_i) = \exp(-(x-\mu_{i1})^2 / 2\sigma_{i1}{}^2)/\mathrm{sqrt}(2\pi\sigma_{i1})$
  $\qquad * \exp(-(x-\mu_{i2})^2 / 2\sigma_{i2}{}^2)/\mathrm{sqrt}(2\pi\sigma_{i2})$

- **Example**: general multivariate Gaussian:

  $P(\mathbf{x}) \mid c_i)\, P(c_i) =$

  $\exp(-(\mathbf{x}-\mu_i)^T \Sigma^{-1}(\mathbf{x}-\mu_i)/2)/(\mathrm{sqrt}(2\pi)^{N/2} |\Sigma|^{1/2})$

  where **x** and $\mu_\mathbf{i}$ are vectors, Σ is the covariance matrix, and N is the dimension.

Covariance Matrix

- c(i,j) = { [x(1,i) - m(i)] [x(1,j) - m(j)]
  \+ ... +
  [x(n,i) - m(i)] [x(n,j) - m(j)] } / (n - 1)
  where m(i) and m(j) are the means of their respective sample variables and n is the number of samples.
- c(i,j) = 0: variables are uncorrelated
- c(i,j) = product of standard deviations: variables are perfectly correlated

Mahalanobis Distance

- **G**eneral multivariate Gaussian:

  $P(\mathbf{x}) \mid c_i)\, P(c_i) =$

  $\exp(-(\mathbf{x}-\mu_i)^T \Sigma^{-1}(\mathbf{x}-\mu_i)/2)/(\mathrm{sqrt}(2\pi)^{N/2} |\Sigma|^{1/2})$

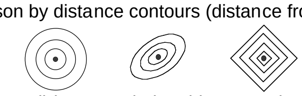
  the argument of exp:

  $-(\mathbf{x}-\mu_\mathbf{i})^T \Sigma^{-1}(\mathbf{x}-\mu_i)/2$

  is called the **Mahalanobis distance** from **x** to the mean $\mu_\mathbf{i}$.

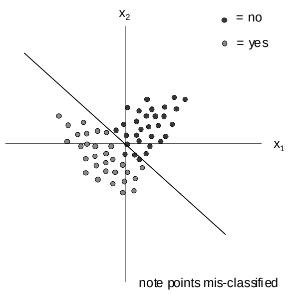
Mahalanobis Distance

- The Mahalanobis distance distance *generalizes* Euclidean distance.
- If the covariance matrix is diagonal, then the M-distance is the same as the E-distance.
- This gives rise to a **minimum Euclidean-distance classifier**.
- Comparison by distance contours (distance from center):

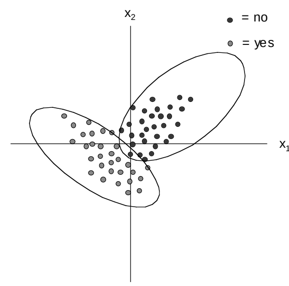


Euclidean Mahalanobis Hamming

Linear Discriminant Functions



Quadratic Discriminant Functions



Discriminant Functions for Multivariate Gaussian

- Quadratic discriminant functions are well-suited for multivariate Gaussian distributions.
- This is seen by using the ln's of the distribution functions for discrimination. The ln's are quadratic in x.
- Effectively the selected class for a data point x is the one for which x has the closest Mahalanobis distance to the class' mean.

2-D Example

- See text NAS, Figure 2-5
- x = (height, weight)
- classes = {man, woman}

	100 measurements [weight, height]	**1000 measurements [weight, height]**
woman	μ = [63.7385, 1.6084] Σ = [[77.1877, 0.0139], [0.0139, 0.0047]]	μ = [64.86, 1.62] Σ = [[90.4401, 0], [0, 0.0036]]
man	μ = [82.5278, 1.7647] Σ = [[366.3206, 0.4877], [0.4877, 0.0084]]	μ = [78.02, 1.75] Σ = [[310.1121, 0], [0, 0.0081]]

man classifier = $77.16h^2 - 233.95h + 0.0039w^2 - 0.4656w + 129.4 > 0$ (based on 1000)

Kernel-Based Classification

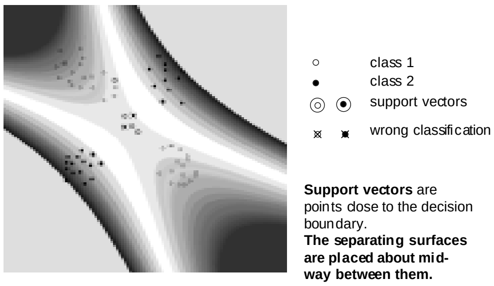
- A 3-layer network:
  - Non-linear functions of inputs (transform inputs), typically *symmetric* functions
  - Linear combinations of the outputs of the non-linear functions
  - Maximum selection
- In other words, a neural network
- The idea is motivated by **Cover's theorem**, which states that *any* classification problem is linearly-separable if transformed to a sufficiently-high dimensional space.

Related Topics

- Gaussian processes
- Support Vector Machines
  - Generalize Radial-Basis Function Networks, plus add a threshold at output.
  - See NAS, sections 5.8, 3.3.3, 3.3.4 (large-margin perceptron and Adatron algorithm)
- CMAC (Cerebellar Model Articulation Controller, J.S. Albus, 1975).

Support-Vector Machine Demo

http://svm.research.bell-labs.com/SVT/SVMsvt.html



Support vectors are points dose to the decision boundary.
The separating surfaces are placed about mid-way between them.