PRAC1: MAP Y DATOS GAUSSIANOS
CRITERIOS DE CLASIFICACIÓN BASADOS EN MAXIMIZAR LA PROBABILIDAD A POSTERIORI.

Asignatura: CLP: Clasificación de Patrones.
Optativa de 2º Ciclo
ETSETB-UPC
TSC-UPC

Profesores:
Margarita Cabrera
Josep Vidal
UPC-TSC-D5

Marzo-2006

1 Introducción a PR-TOOLS

PR-TOOLS consiste en un paquete de software de Matlab, especialmente diseñado para realizar aplicaciones de Clasificación de patrones y trabajar con bases de datos experimentales.
PR-Tools ha sido creado por: *Pattern Recognition Group, Delta University of Technology (Holanda). http://www.ph.tn.tudelft.nl/prtools.*

A continuación se presentan algunas características importantes del trabajo con este tipo de software:

- Conjuntos de objetos etiquetados.
- Un **objeto** es un vector k-dimensional. En cada coordenada se almacena un valor de una característica.
- El espacio definido por todos los valores posibles de características se denomina **espacio de características**.
- Los objetos se representan como puntos o vectores en este espacio.
- Una **función de clasificación asigna etiquetas a nuevos objetos** que aparecen dentro del espacio de características.

Conjuntos de objetos: Los conjuntos de objetos similares dentro del espacio de características **se agrupan en "Clusters**".

Problema fundamental: Hallar una **medida de distancia**. La distancia euclídea suele cambiarse a través de transformaciones de escalado o de movimientos o rotaciones del vector de características que representa un objeto.

La dimensionalidad del espacio de características se puede reducir por la selección de sub-conjuntos de características, mediante diferentes tipos de algoritmos que se denominan como estrategias de "Feature Selection". Ventajas:

- Mejora de la Velocidad Computacional
- Ocasionalmente mejora la precisión de los algoritmos de clasificación.
- Menor complejidad de los algoritmos de clasificación.

Los clasificadores pueden ser:

- Lineales
- No lineales

Y se hallan basados en dos estrategias distintas:

- Minimizar el error de clasificación a partir de la estimación de las f.d.p (Regla de Bayes, en el caso de que se conociera exactamente la f.d.p.)
- Optimizar determinadas funciones de clasificación.

En el entrenamiento de un clasificador, es interesante realizar estrategias de

- Cross-Validation
- Rotación

Programación PR-Tools con MATLAB:

Clases (Equivalen a 2 tipos de estructuras de datos): Dataset y Mapping

DATASET (A): Conjunto de objetos representados por una matriz de vectores de características. Adjunto a esta matriz se tiene información adicional de los datos:

DATASET Dataset class constructor

a = dataset(d,labels,featlist,prob,lablist,imheight)

A dataset object is constructed from:
d size [m,k], a set of m datavectors of k features
labels size [m,n] labels for each of the datavectors either in string or in numbers
featlist size [k,f] defines the labels for the k features (Los nombres de las características)
prob size [c,1], apriori probabilities for each of the c classes
prob = 0: all classes have equal probability 1/c
prob = []: all datavectors are equally probable
lablist size [c,n] classlabels (Los nombres de las clases)

Alternativamente se puede utilizar esta subrutina para testear todos los parámetros que forman el objeto de este tipo:

[nlab,lablist,m,k,c,prob,featlist,imheight] = dataset(a)

MAPPING (W): Almacena clasificadores entrenados, resultados de extracción de características, definiciones de escalado de datos, proyecciones no lineales, etc..
A*W es un dataset.

MAPPING Mapping class constructor

w = mapping(map,d,lablist,k,c,v,par)

A map/classifier object is constructed from:
d size (any), a set of weights defining the mapping
lablist size [c,n], defines the labels for the outputs ('classes') of the mapping, either in string or in numbers c is the number of classes. At least two labels should be supplied.
map type or name of routine used for learning or testing
k number of inputs (número de características)
c number of outputs (número de clases). Note that if c == 1, lablist should still have two labels, one for each 'direction'
v size [c] or size [1] Output multiplication factor, for all outputs simultaneously or for each output separately
par size (any), parameter vector describing the structure of the
mapping (type dependent)
classbit 0 | 1 if classc==1 this is a classifier: map by w*sigm*normm
which constructs normalized probabilistic outputs
[d,lablist,map,k,c,v,par] = mapping(w) Retrieves the parameters from a mapping w.

2 Objetivos de la práctica 1

- Los objetivos de la presente práctica consisten en familiarizarse con el software prtools, con matlab y con el entorno de trabajo en general.
- Generar bases de datos Gaussianas y evaluar parámetros de las mismas.
- Aplicar clasificadores MAP (Lineales y Cuadráticos) sobre las clases generadas.

3 Laboratorio: Creación de Bases de datos Gaussianas.

3.1 PARTE 1:

Ejecute "prac1_gauss3.m"

Se trabaja con vectores de dimensión d=3 y C=2 clases. Las 3 coordenadas son estadísticamente independientes tal como describen las matrices de covarianza (Caso 1)

$$\mathbf{y}_i : N(\mathbf{s}_i, \boldsymbol{\Sigma}); \quad \boldsymbol{\Sigma} = \sigma^2 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Observe las probabilidades de error obtenidas mediante los clasificadores

ldc: Fronteras de decisión lineales. Presupone clases equiprobales y idéntica matriz de covarianza.
qdc: Fronteras de decisión cuadráticas. Presupone clases equiprobales y estima diferentes matrices de covarianza.

Anote las probabilidades de error anteriores en fichero Word G***_Prac1.doc:

distancia=1, SNR=3,0,-3,-10 dB, $SNR = 10\log_{10}\left(\frac{Energia\ promedio}{\sigma^2}\right)$

Observe las gráficas de la ROC para , SNR=3,0,-3,-10 dB y comente los resultados.

3.2 PARTE 2:

Se propone utilizar la modulación QPSK como base de trabajo. *Prac1_QPSK.m*
Por tanto los vectores de características son de dimensión d=2 y se tienen C=4 clases. Inicialmente se contemplan matrices de covarianza iguales para todas las clases (Caso 2) pero no diagonales.

$$\mathbf{y}_i : N(\mathbf{s}_i, \boldsymbol{\Sigma}); \quad \boldsymbol{\Sigma} = \sigma^2 \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$$

Se pide analizar para esta constelaciones:

- Mapas de Señal y fronteras de decisión mediante criterios ldc y qdc
- Distancia de Mahalanobis Inter-clases

La evaluación de las anteriores características se propone realizar para:

- Relación señal a ruido: SNR=10dB
- Distancia entre clases: dist=1
- Parámetro ro = $\rho = 0, 0.5, -0.5$

Observe que apenas hay diferencias entre ambos clasificadores.

Evalúe el mapa de señal al aplicar ldc y qdc, para el siguiente caso (Caso 3). Compare la forma de los clusters con los autovalores de las distintas matrices de covarianza, que puede obtener llamando a la subrutina "eig" de matlab.
SNR=5dB, 0dB

- Símbolo 1: ro = +0.5
- Símbolo 2: ro = 0
- Símbolo 3: ro = –0.5
- Símbolo 4: ro = +0.8

Para calcular autovalores, por ejemplo de la clase i:

Autoval=eig(squeeze(G_A1(:,:,i))) %Exact
Autoval=eig(squeeze(Cov(:,:,i))) %Estimated

Puede guardar los resultados anteriores en el documento Word y comentar los resultados.

A continuación reparta la potencia de ruido (ro=0) entre las 4 clases proporcionalmente a [1.4 0.5 0.05 0.05]/2 y observe las diferencias producidas sobre las fronteras de decisión al aplicar clasificador lineal (ldc) o cuadrático (qdc). Note en el programa que esta instrucción se halla programada pero comentada %.

3.3 PARTE 3:

Se propone utilizar como base de trabajo BPSK aunque en dos dimensiones. *Prac1_BPSK.m*
Por tanto los vectores de características son de dimensión d=2 y se tienen C=2 clases. Inicialmente se contemplan matrices de covarianza iguales para todas las clases (Caso 2) pero no diagonales.

$$\mathbf{y}_i : N(\mathbf{s}_i, \boldsymbol{\Sigma}); \quad \boldsymbol{\Sigma} = \sigma^2 \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}; \rho = -0.5 \text{, SNR=5dB}$$

Se pide analizar para esta constelación:

- Mapas de Señal y fronteras de decisión mediante criterios ldc y qdc.
- Errores de decisión mediante criterios ldc y qdc.
- Autovalores de las matrices de covarianza.

Posterior se propone analizar el siguiente caso (Caso 3).

$$\mathbf{y}_i : N(\mathbf{s}_i, \mathbf{\Sigma}_i);\ \mathbf{\Sigma}_i = \sigma^2 \begin{pmatrix} 1 & \rho_i \\ \rho_i & 1 \end{pmatrix}$$

SNR=5dB

- Símbolo 1: ro = –0.8
- Símbolo 2: ro = +0.8

Se pide analizar para esta constelación:

- Mapas de Señal y fronteras de decisión mediante criterios ldc y qdc.
- Errores de decisión mediante criterios ldc y qdc.
- Autovalores de las matrices de covarianza.

Puede guardar los resultados anteriores en un documento Word y comentar los resultados.

Optative:
Adicionalmente puede cambiar, para este tercer caso, las probabilidades a priori de las dos clases, para ello cambie el número de muestras de las dos clase, ya que las probabilidades a priori se obtienen a partir de las proporciones de muestras de cada clase.

3.4 PARTE 4: OPCIONAL (Clasificador Cuadrático)

Ejecute prac1_QPSK_qdcr. Ver las diferencias de los parámetros que se guardan del clasificador entre las subrutinas. Interpretar a partir de la información obtenida del help de prtools. (help qdc; help quadrc);

- Qdc
- Quadrc

3.5 PARTE 5: OPCIONAL (Curva ROC para 2 distribuciones gaussianas monodimensionales)

Ejecute P1_ROC.

4 Subrutinas a Utilizar

Prac1_gauss3.m:
Prac1_QPSK.m:
Prac1_BPSK.m:
Prac1_QPSK_qdcr.m (OPCIONAL).
P1_ROC.m (OPCIONAL).

GAUSS Generation of multivariate Gaussian dataset.

```
A = gauss(n,U,G)

Generation of n k-dimensional Gaussian distributed vectors with
covariance matrices G (size k*k*c) and with means, labels and
prior probabilities defined by the dataset U with size (c*k).
Alternatively n can be a vector with length c.

Default:
        G   : eye(k)
        U   : zeros(1,k)
```

LDC Linear Discriminant Classifier

```
W = ldc(A,r,s)

Computation of a linear discriminant between the classes of the
dataset A assuming normal densities with equal covariance
matrices. The joint covariance matrix is the weighted (by apriori
probabilities) average of the class covariance matrices.

r and s (0 <= r,s <=1) are regularization parameters used for
finding the covariance matrix by

        G = inv((1-r-s)*G+r*diag(diag(G)))+
                s*mean(diag(G))*eye(size(G,1))
So,     r = 0 : (default) no regularization
        r = 1 : don't use data
```

QDC Quadratic Bayes Normal Classifier

```
W = qdc(A,r,s)

Computation of the quadratic classifier between the classes of the
dataset A assuming normal densities. r and s (0 <= r,s <=1) are
regularization parameters used for finding the  covariance matrix
by
```

```
G = (1-r-s)*G + r*diag(diag(G)) +
                    s*mean(diag(G))*eye(size(G,1))
```

Default: r = 0, s= 0.

The classification A*W is computed by normal_map. See there for details.

See also datasets, mappings, nmc, nmsc, ldc, udc, quadrc, normal_map

ROC Receiver-operator curve

```
e = roc(D,k)
```

Computes k points of the receiver-operator curve of the classifier W for the labeled data set D, which is typically the result of D = A*W*classc, The curve is computed for k thresholds of the aposteriori probabilities stored in D. The resulting error frequencies for the two classes are stored in the two columns of e, which may conveniently be plotted by plot2. Default k = 100

See also datasets, mappings, reject, plot2

DISTMAHA Mahalanobis distance

```
D = distmaha(A,U,G)
```

Computation of the Mahanalobis distances of all vectors in the dataset A to a dataset of points U, using the covariance matrix G. G should be either a 2-dimensional square matrix of the right size or a 3-dimensional matrix containing a covariance matrix for each point in U. If A contains m vectors and U n vectors, the size of D is m*n.

```
D = distmaha(A)
```

Estimation of the Mahalanobis distance matrix between all classes in the set of data vectors in A defined by labels.

See also datasets