

Práctica 4

K-NN y ventanas de Parzen



Departament de Teoria
del Senyal i Comunicacions



ÍNDICE

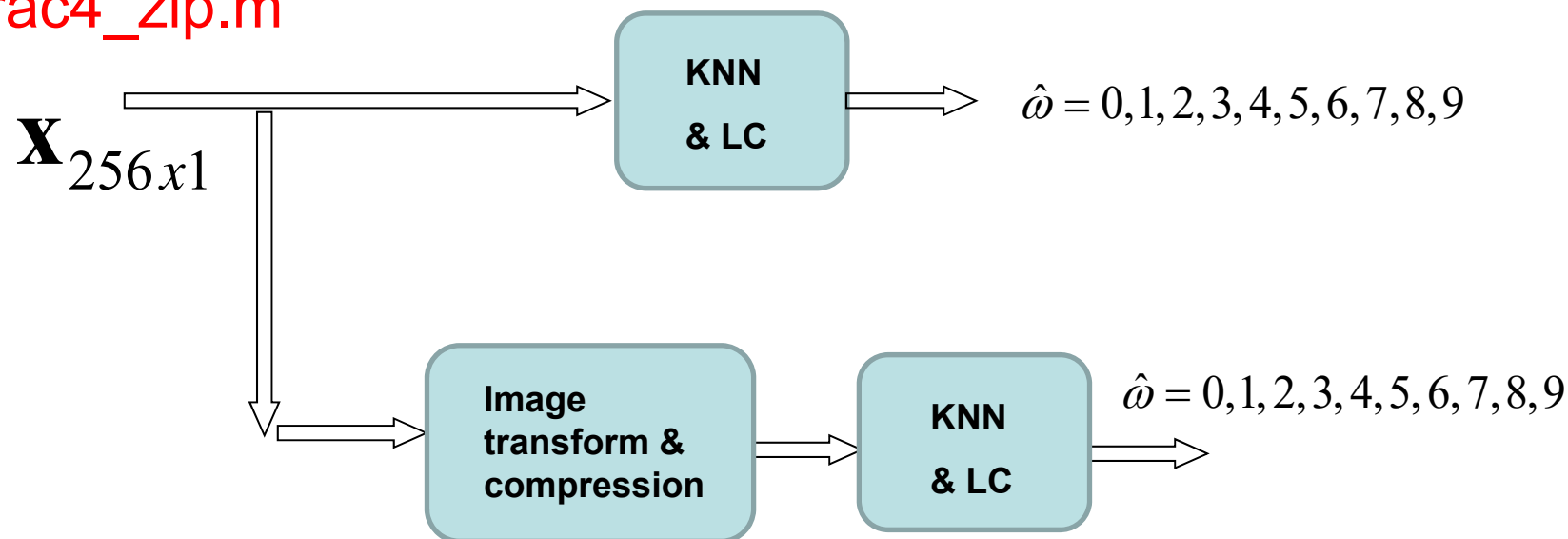
- 4.1 Esquema de la práctica, Clasificadores: KNN, LC
- 4.2 Transformaciones lineales para compresión de imagen
- 4.3 Base de datos ZIP
- 4.4 Clasificador basado en ventanas de Parzen

1 Esquema de la práctica

KNN busca los “k” vectores de train más cercanos al vector en cuestión, y la clase mayoritaria dentro de los “k” es la predicción del vector

Cada una de las 10 clases corresponde a un dígito escrito a mano

prac4_zip.m



$\mathbf{y}_{d'}$

$d' = N_dim \times N_dim$

$N_dim=8,4,2$

2 Transformaciones lineales para compresión

- Imagen:
$$\mathbf{X} = \begin{pmatrix} x(1,1) & x(1,2) & : & x(1,N) \\ x(2,1) & x(2,2) & : & x(2,N) \\ \vdots & \vdots & : & \vdots \\ x(N,1) & x(N,2) & : & x(N,N) \end{pmatrix}$$
- Transformada unitaria (la matriz de transformación es unitaria)

$$\mathbf{Y} = \mathbf{A}\mathbf{X}\mathbf{A}^T, \text{ with } \mathbf{A}^T\mathbf{A} = \mathbf{I} \Rightarrow \mathbf{X} = \mathbf{A}^T\mathbf{Y}\mathbf{A}$$

- Funciones Base:
$$\mathbf{A} = (\mathbf{a}_1 \quad \mathbf{a}_2 \quad \cdots \quad \mathbf{a}_N) = \begin{pmatrix} a_1(1) & a_2(1) & : & a_N(1) \\ a_1(2) & a_2(2) & : & a_N(2) \\ \vdots & \vdots & : & \vdots \\ a_1(N) & a_2(N) & : & a_N(N) \end{pmatrix}$$

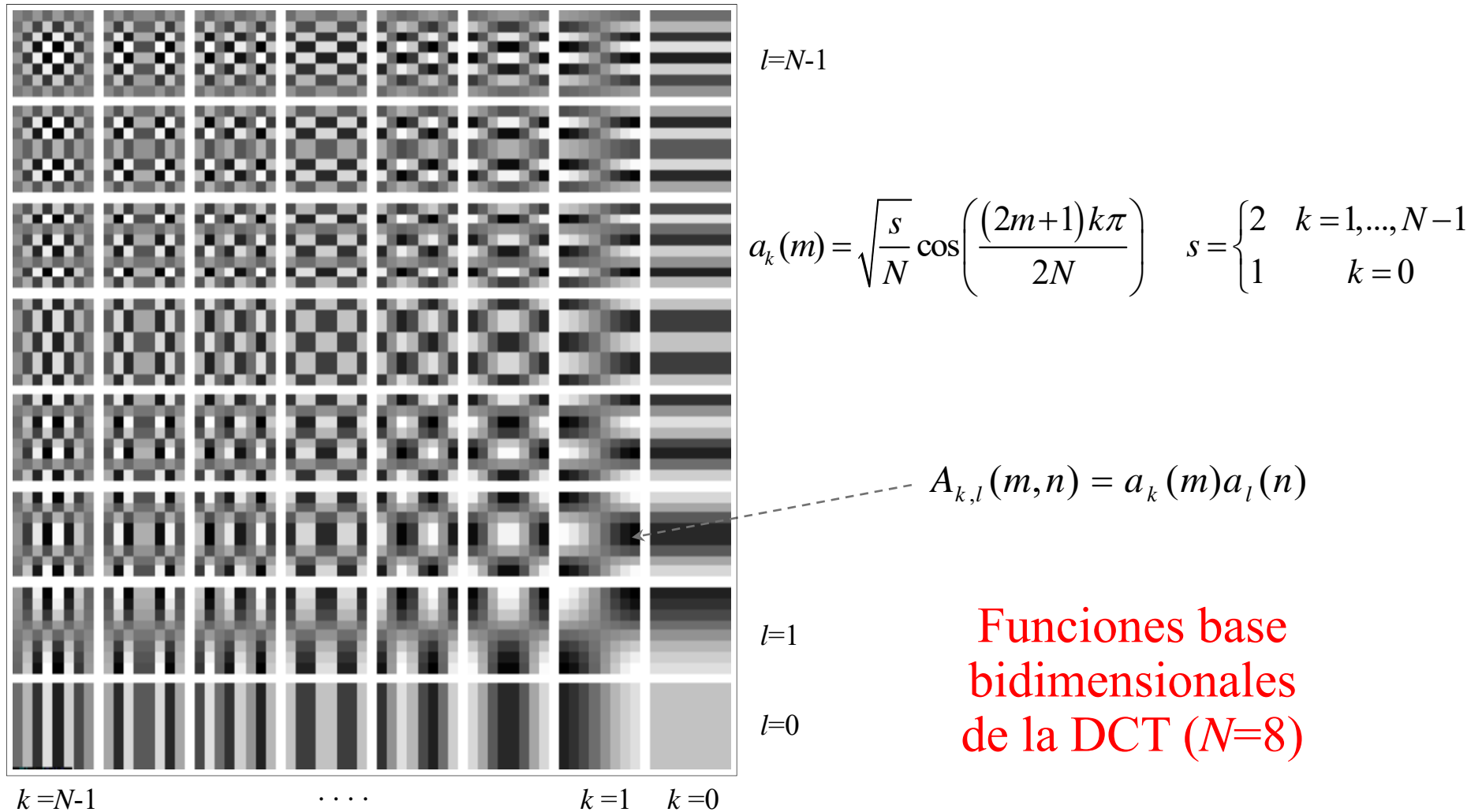
- Transformada

$$\mathbf{Y} = \begin{pmatrix} y(1,1) & y(1,2) & : & y(1,N) \\ y(2,1) & y(2,2) & : & y(2,N) \\ : & : & : & : \\ y(N,1) & y(N,2) & : & y(N,N) \end{pmatrix}$$

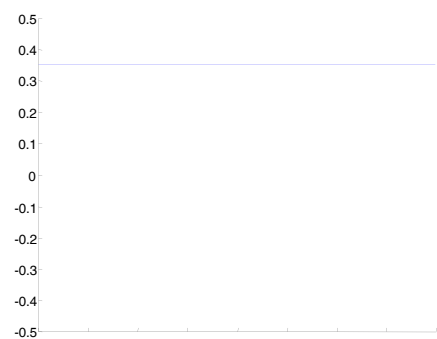
$$x(m,n) = \sum_{i=1}^N a_m(i) \sum_{j=1}^N a_n(j) y(i,j) = \sum_{i=1}^N \sum_{j=1}^N y(i,j) \mathbf{A}_{i,j}(m,n)$$

- Es decir, la imagen \mathbf{X} es una combinación lineal de imágenes $\mathbf{A}_{i,j}$ tal que los valores $y(i,j)$ representan los pesos respecto a dicha función.

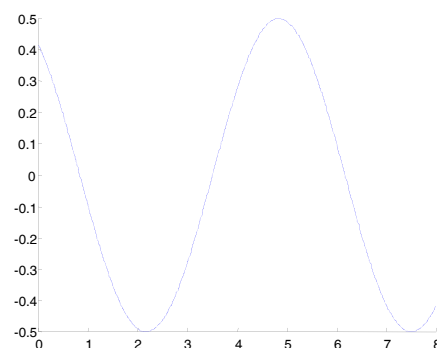
Transformada discreta de coseno



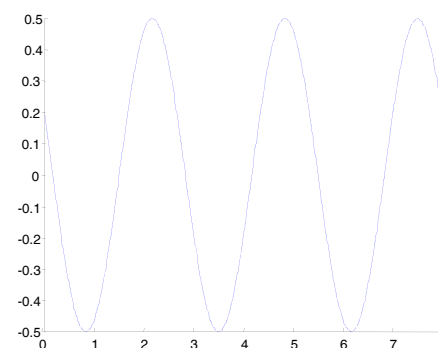
Funciones base unidimensionales de la DCT para N=8



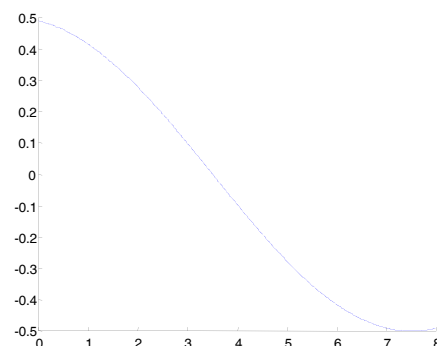
$a_0(n)$



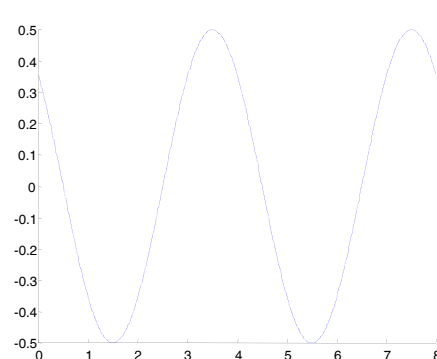
$a_3(n)$



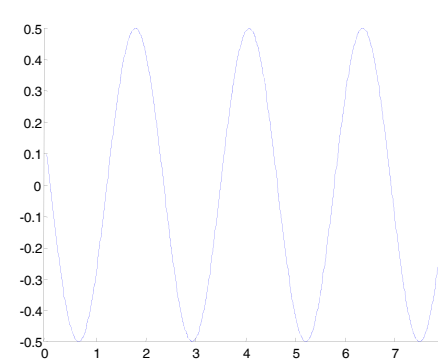
$a_6(n)$



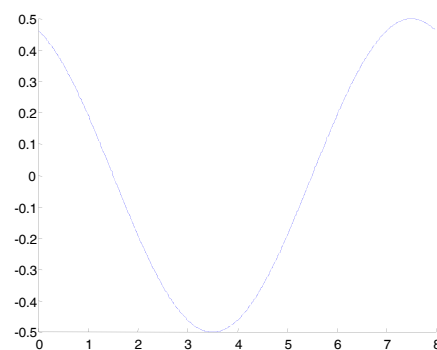
$a_1(n)$



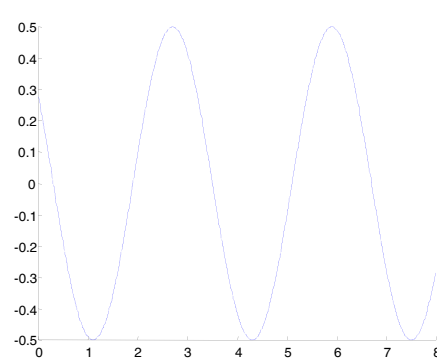
$a_4(n)$



$a_7(n)$



$a_2(n)$



$a_5(n)$

Transformada de Hadamard

- Matrices de Transformación $N=2^L$

$$\mathbf{A}_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

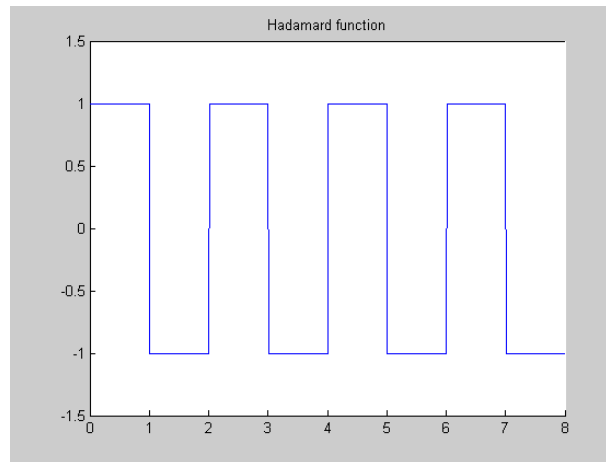
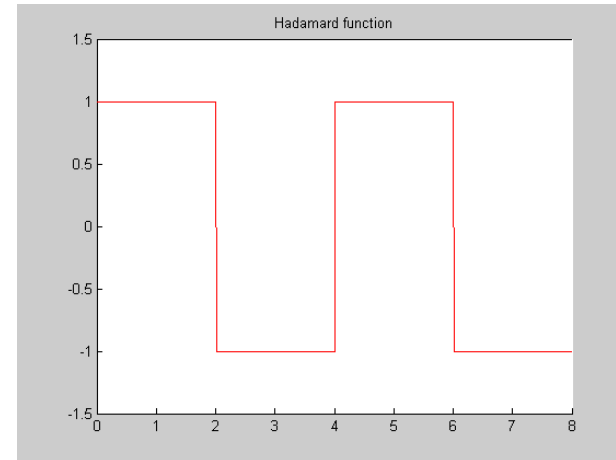
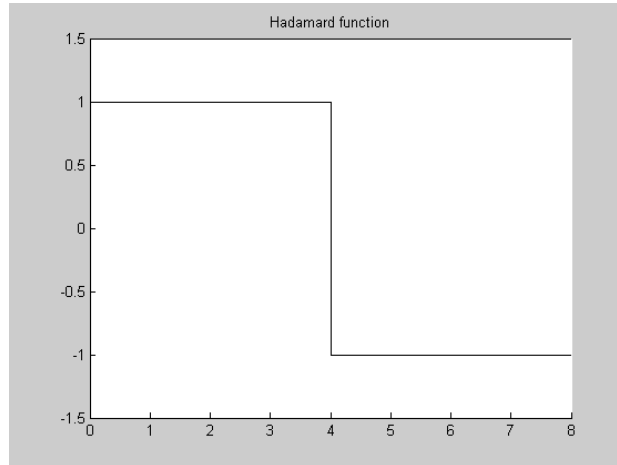
- Construcción recursiva:

$$\mathbf{A}_L = \frac{1}{\sqrt{2}} \begin{pmatrix} \mathbf{A}_{L-1} & \mathbf{A}_{L-1} \\ \mathbf{A}_{L-1} & -\mathbf{A}_{L-1} \end{pmatrix}$$

- Ejemplo (N=8):

$$\mathbf{A}_3 = \frac{1}{2\sqrt{2}} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{pmatrix}$$

Algunas funciones base unidimensionales de Hadamard para $N=8$



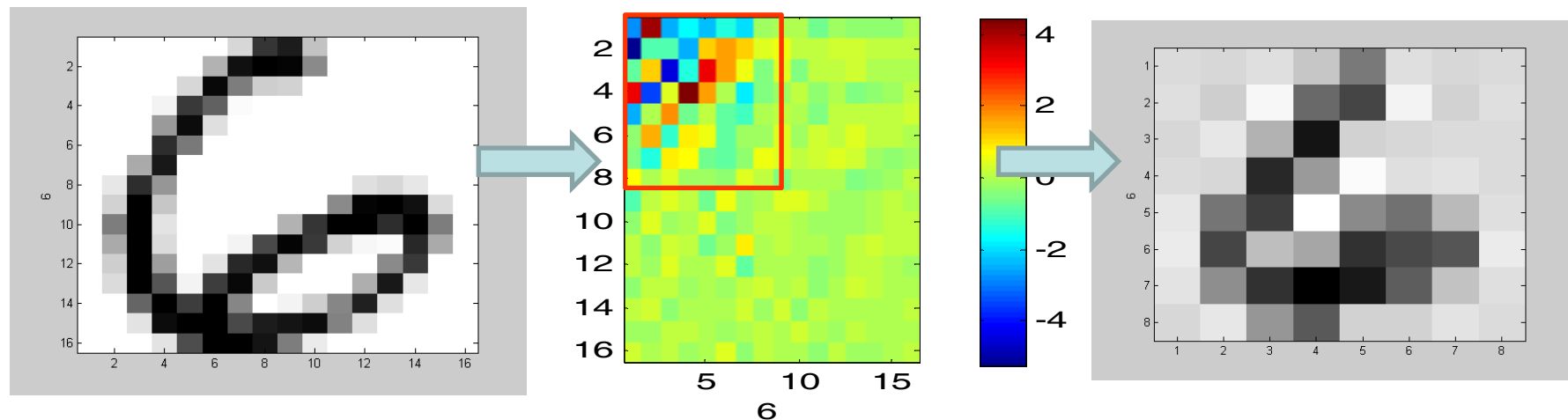
3 Base de datos ZIP

Cada vector de $d=256$ muestras representa una imagen de 16×16 píxeles, perteneciente a una de las 10 clases diferentes correspondientes a los dígitos (0,...,9) escritos a mano.

Parámetros de la base de datos:

- Total de vectores de la base de datos de Train: 7291
- Total de vectores de la base de datos de Test: 2007
- Dimensión inicial de los vectores: $d=256$
- Clases: 10

Ejemplo: reducción de 256 a 64 coeficientes transformados, usando la DCT



Una vez hecha la compresión, seremos capaces de seguir reconociendo?

4 CLASIFICADOR: VENTANAS DE PARZEN

La estimación de la pdf condicionada a la clase c usando n vectores de entrenamiento de dicha clase viene dada por

$$\hat{f}(\mathbf{x}|c) = \frac{1}{n} \sum_{i=1}^n \frac{1}{V_{R,n}} \varphi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n}\right) = \frac{1}{n} \sum_{i=1}^n \gamma_c(\mathbf{x} - \mathbf{x}_i)$$

Test Train

La función matlab proporcionada `predict_parzen.m`, utiliza ventana gaussiana:

$$\gamma_c(\mathbf{x} - \mathbf{x}_i) = \frac{1}{h_n^d} \frac{1}{\sqrt{(2\pi)^d |\mathbf{C}_c|}} \exp\left(-\frac{1}{2} \left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n}\right)^T \mathbf{C}_c^{-1} \left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n}\right)\right) \quad \int \gamma_c(\mathbf{x}) d\mathbf{x} = 1$$

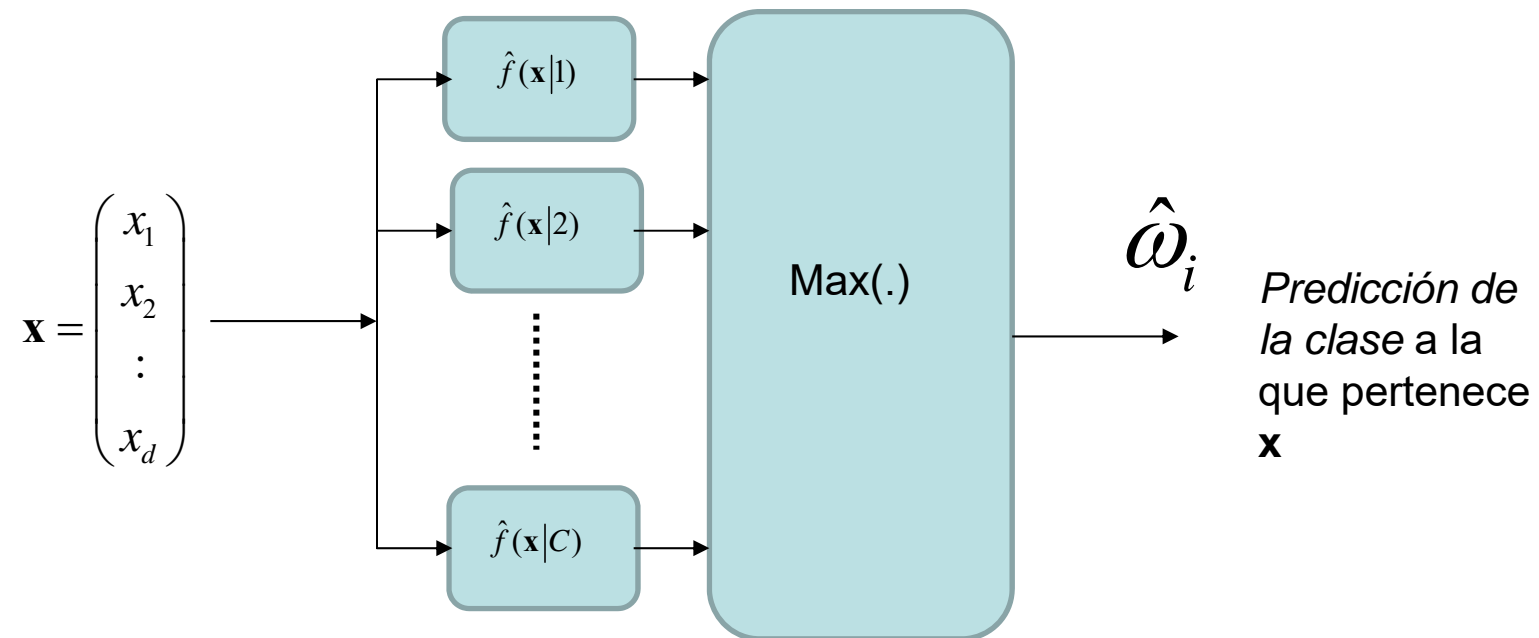
donde d es el número de coordenadas de cada vector y el parámetro que determina el ancho de la ventana es

$$h_n = \frac{h}{\sqrt{n}}$$

La función devuelve la predicción para cada vector de la base de test

Clasificador basado en ventanas de Parzen

Clasificador de c categorías:



```
function Predict_test = predict_parzen  
    (X_train, Labels_train, N_classes, h, X_test)
```

Parzen Windows are
computed from the
Train set

```
% Parzen classifier with gaussian window  
% X_train: matrix, rows contain train vectors  
% Labels_train: Assumed to be 0,1,2,...,N_classes-1  
% N_classes: Number of different classes  
% h: Window width parameter  
% X_test: matrix, rows contain vectors to be labeled  
% Predict_test: Labels given to the test set
```

Introduce
Train, Test
or any set
of vectors
to be
labeled

GUIÓN PRÁCTICA 4

- Clasificadores: KNN, LC, sobre imágenes no comprimidas.
- KNN y LC sobre transformaciones lineales para compresión de imagen, DCT y HT con niveles de compresión 8x8, 4x4 y 2x2.
Elegir la mejor combinación KNN como la “más eficiente computacionalmente” de las 6 transformaciones resultantes, tal que el error de test $< 10\%$.
- Validar parámetro “k” en KNN para **la mejor combinación.**
- Clasificador basado en ventanas de Parzen. Validar ancho de ventana gaussiana para **la mejor combinación.**