Técnicas Supervisadas
Aproximación no paramétrica

2016

- *Notas basadas en el curso Reconocimiento de Formas de F.Cortijo, Univ. de Granada*
- *Pattern Classification de Duda, Hart y Storck*
- *The Elements of Statistical Learning de Hastie, Tibshirani y Friedman*
- *Parte del material se extrajo de las notas:Técnicas Supervisadas II: Aproximación no paramétrica de F.Cortijo, Univ. de Granada*

Contenido

- Estimación no paramétrica de la función de densidad
  - Estimadores de Parzen
  - Estimación mediante los k-vecinos más próximos

- Método de clasificación del vecino más próximo

Metodología Bayesiana

- **Idea:** Estudiar probabilidades de tomar decisiones incorrectas para cuantificar los costos y compromisos de esas decisiones y diseñar las estrategias de menor costo

- Supuestas **conocidas** todas las probabilidades en juego estudiaremos como establecer las reglas de decisión.

Bayes

Supuesto conocidas las prioris y las densidades condicionales

Para inferir la naturaleza del pixel de vector de características **x**,

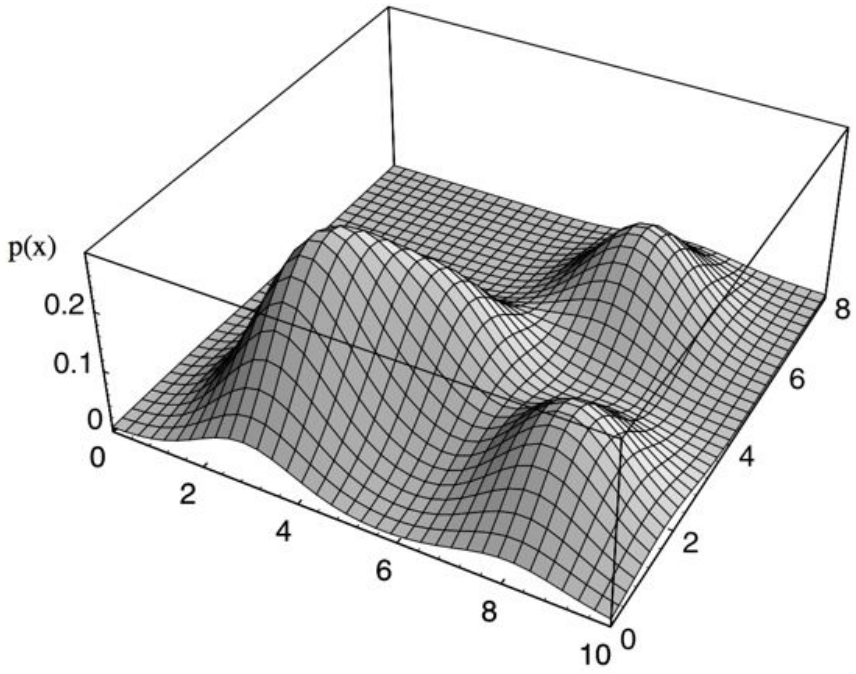
usamos Bayes: $p(\mathbf{x}, w_i) = P(w_i / \mathbf{x})\, p(\mathbf{x}) = p(\mathbf{x} / w_i) P(w_i)$

$$\Rightarrow \quad P(w_i / \mathbf{x}) = \frac{p(\mathbf{x} / w_i) P(w_i)}{p(\mathbf{x})}$$

- $P(w_i / \mathbf{x})$- *posterior:* probabilidad de que la clase sea w_i dado que se midió **x** .
- $P(w_i)$ – *prior:* conocimiento previo del problema
- $p(\mathbf{x} / w_i)$- *verosimilitud* : de la clase w_i respecto a **x**, cuanto mayor más probable que la verdadera clase sea w_i.
- $p(\mathbf{x})$- *evidencia:* factor de escala, normaliza a 1.

Estimación de densidades no paramétricas

- Formas paramétricas raramente ajustan densidades encontradas en la práctica (multimodales)



Estimación de densidades no paramétricas

- Formas paramétricas raramente ajustan densidades encontradas en la práctica (multimodales)

- Métodos no paramétricos:
  - Procedimientos que estiman funciones densidades $p(\mathbf{x}|w_j)$ a partir de las muestras.

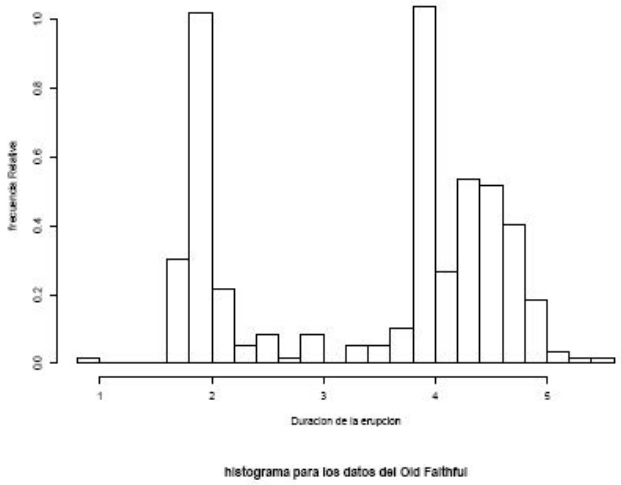
Estimación de densidades no paramétricas

- Formas paramétricas raramente ajustan densidades encontradas en la práctica (multimodales)

- Métodos no paramétricos:
  - Procedimientos que estiman funciones densidades $p(\mathbf{x}|w_j)$ a partir de las muestras.
  - Procedimientos que estiman directamente las probabilidades a posteriori $P(w_j|\mathbf{x})$.
  - *Relacionado con métodos que proponen directamente reglas de decisión (*Regla del vecino más cercano)

Histograma

- Estimador de densidades más sencillo y antigüo
- Realiza partición del espacio en intervalos (bins).
- Estimo la densidad por el número de muestras que caen en un intervalo (bin).



$$p_i = \frac{n_i}{n\Delta_i} \quad \Delta_i : \text{tamaño del intervalo}$$

fig: Cañette

Histograma

- Estimación depende del tamaño del intervalo:
  - Δ : pequeño aproximación ruidosa
  - Δ : grande aproximación promediada, ej: falla en capturar bimodal.
  - Mejores resultados Δ intermedio.

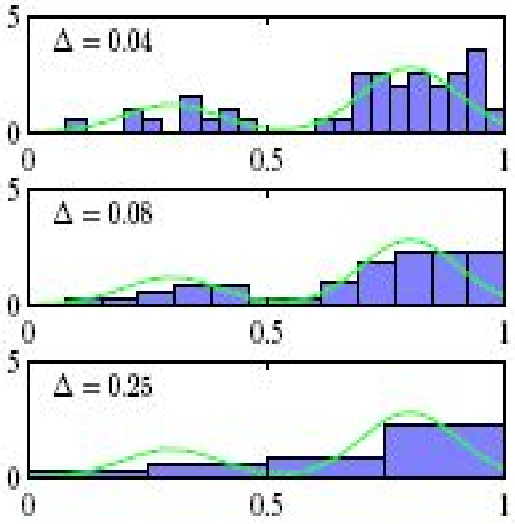


Fig: Bishop
Mezcla de
Gaussianas

Histograma

- Estimación depende del tamaño del intervalo.
  - Δ : pequeño aproximación ruidosa
  - Δ : grande aproximación promediada, ej: falla en capturar bimodal.
  - Mejores resultados Δ intermedio.
- Depende de la localización de los bins.
- Limitaciones:
  - Densidad estimada es discontinua
  - Escalado con la dimensionalidad.
  - M bins por dimensión, espacio D-dimensional M^D bins $\Rightarrow$ cantidad de datos necesaria para aproximación local significativa es prohibitiva.

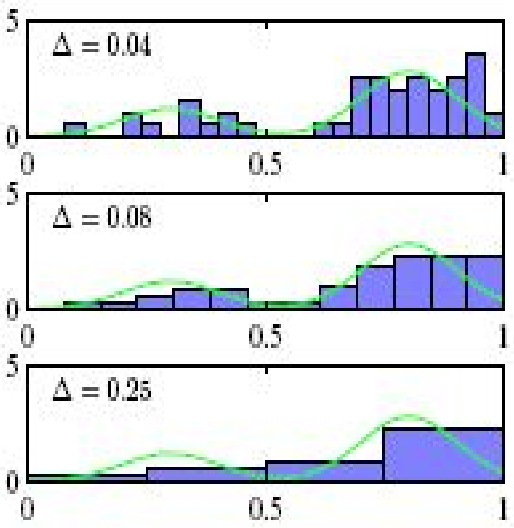


Fig: Bishop
Mezcla de
Gaussianas

Clasificación con Histograma

- Supongamos conjunto de n muestras de entrenamiento.
- En el bin B tengo k muestras, donde k_i son de la clase w_i
- Dada una nueva muestra "x" que cae en el bin B a qué clase le asigno para minimizar el error?

Clasificación con Histograma

- Supongamos conjunto de *n* muestras de entrenamiento.
- En el bin B tengo *k* muestras, donde k_i son de la clase w_i
- Dada una nueva muestra "x" que cae en el bin B a qué clase le asigno para minimizar el error?

$$p(\mathbf{x}) = \frac{k/n}{\Delta} \quad p(\mathbf{x}|w_i) = \frac{k_i/n_i}{\Delta} \quad p(w_i) = \frac{n_i}{n}$$

Clasificación con Histograma

- Supongamos conjunto de *n* muestras de entrenamiento.
- En el bin B tengo *k* muestras, donde k_i son de la clase w_i
- Dada una nueva muestra "x" que cae en el bin B a qué clase le asigno para minimizar el error?

$$p(\mathbf{x}) = \frac{k/n}{\Delta} \quad p(\mathbf{x}|w_i) = \frac{k_i/n_i}{\Delta} \quad p(w_i) = \frac{n_i}{n}$$

$$p(w_i|\mathbf{x}) = \frac{p(\mathbf{x}|w_i)p(w_i)}{p(\mathbf{x})} = \frac{k_i/n_i}{\Delta}\frac{n_i}{n}\frac{\Delta}{k/n} = \frac{k_i}{k}$$

- Al bin B se le asocia la etiqueta de la clase más frecuente.

Método del Histograma

- p(x) depende de vecindad.
- Vecindad en histograma = Definición de Bins
- Parámetro de suavizado: ancho del bin.
- Escalabilidad: El número de celdas crece en forma exponencial con la cantidad de características M^d .

- Veremos métodos que tienen mejor propiedades de escalabilidad que el histograma.

Estimación de densidades

$$P \text{ probabilidad de } \mathbf{x} \in R : \quad P = \int_R p(x')dx'$$

Estimación de densidades

$$P \text{ probabilidad de } \mathbf{x} \in R : \quad P = \int_R p(x')dx'$$

$$\mathbf{x}_1\, \mathbf{x}_2 \ldots \mathbf{x}_n : \text{ iid tomadas de una densidad } p(\mathbf{x}).$$

Estimación de densidades

$$P \text{ probabilidad de } \mathbf{x} \in R : \quad P = \int_R p(x')dx'$$

$$\mathbf{x}_1\, \mathbf{x}_2 \ldots \mathbf{x}_n : \text{ iid tomadas de una densidad } p(\mathbf{x}).$$

- Probabilidad que K (de las N) caigan en R:

$$\mathrm{Bin}(K|N,P) = \frac{N!}{K!(N-K)!} P^K (1-P)^{1-K}.$$

- E(K/N) = P var(K/N) = P(1-P)/N

Estimación de densidades

Si $p(\mathbf{x})$ continua en R y R tan pequeña que $p(\mathbf{x}) = cte$

$P = \int_R p(\mathbf{x}')dx' \approx p(\mathbf{x})V$ con V volumen envuelve a R

$$p(\mathbf{x}) \approx \frac{k/n}{V}$$

Estimación de densidades

Si $p(\mathbf{x})$ continua en R y R tan pequeña que $p(\mathbf{x}) = cte$

$P = \int_R p(\mathbf{x}')dx' \approx p(\mathbf{x})V$ con V volumen envuelve a R

$$p(\mathbf{x}) \approx \frac{k/n}{V}$$

si fijo V y $n \to \infty$ $\frac{P}{V} = \frac{\int_R p(\mathbf{x}')d\mathbf{x}'}{\int_R d\mathbf{x}'}$ versión promediada de $p(\mathbf{x})$

Estimación de densidades

- Si queremos obtener $p(\mathbf{x})$ y no una versión promediada necesitamos $V \rightarrow 0$.
- Si fijamos n puede que la estimación $p(\mathbf{x}) = 0$ o ∞ (cuando no cae ninguna o alguna muestra).
- V no puede ser arbitrariamente pequeño tenemos que admitir varianza en k/n y un cierto promediado en $p(\mathbf{x})$.

Estimación de densidades

- Queremos estimar p(x) y para eso formamos una secuencia de regiones $\{R_1, R_2, ..., R_n\}$ que contenga a x (V_1 para n=1, V_2 para n=2,..)
- V_n : volumen de la región R_n.
- k_n : cantidad de muestras que caen en R_n al usar n muestras,
- Construimos el estimador $p_n(x)$ de p(x) como:

$$p_n(\mathbf{x}) = \frac{k_n/n}{V_n}$$

Estimación de densidades

- k_n : cantidad de muestras que caen en R_n al usar n muestras,
- V_n : volumen de la región R_n.
- Definimos el estimador $p_n(x)$ de p(x) como:

$$p_n(\mathbf{x}) = \frac{k_n/n}{V_n}$$

- Si queremos $p_n(x) \rightarrow p(x)$ se requiere:

  - $\lim_{n\to\infty} V_n = 0$ ➢ Asegura P/V → p(x) (si p(x) continua en x)
  - $\lim_{n\to\infty} k_n = \infty$ ➢ Asegura $k_n/n \rightarrow P$ (si p(x)>0)
  - $\lim_{n\to\infty} k_n/n = 0.$ ➢ Se necesita para que $p_n(x) \rightarrow p(x)$

(infinitas muestras en Vn pero una ínfima parte de las n)

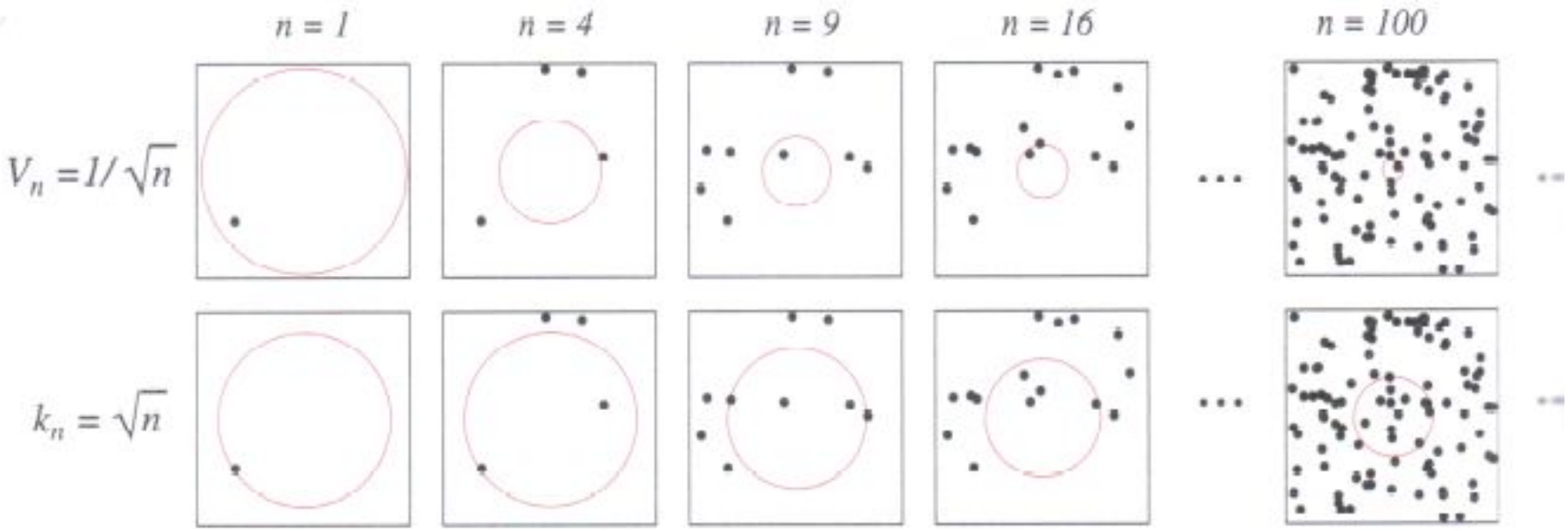
Estimación de densidades

$$p_n(\mathbf{x}) = \frac{k_n/n}{V_n}$$

- Dado un conjunto de n muestras existen 2 formas de analizar la estimación de p(x):
  - Fijar V_n , por ejemplo V_n = 1/sqrt(n) y determinar k_n empíricamente (Ventanas de Parzen)
  - Fijar k_n, por ejemplo k_n = sqrt(n) y determinar V_n empíricamente para que k_n muestras esten en V_n (Estimación de los k-vecinos más cercanos.)
- Ambos métodos convergen al valor verdadero de p(**x**) (respetando hipótesis claro) aunque es difícil hacer aseveraciones del comportamiento para *n* finito.

Ventanas de Parzen vs k- vecinos

Ejemplo: Dos métodos para estimar la densidad en el centro del cuadrado



Duda Hart - Stork

Estimación por ventanas de Parzen

- Sea R_n un hipercubo (d-dimensional) de lado h_n ($V_n = h_n^d$)

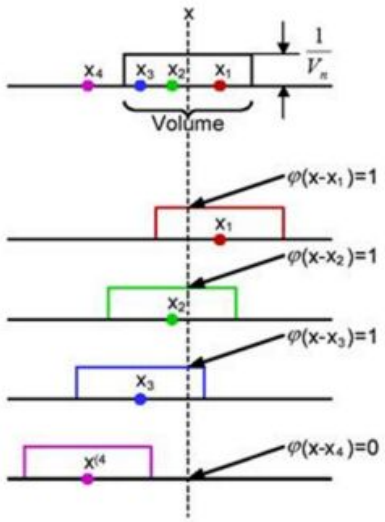
Estimación por ventanas de Parzen

- Sea R_n un hipercubo (d-dimensional) de lado h_n ($V_n = h_n^d$)
- Sea $\varphi(\mathbf{u})$ la ventana:

$$\varphi(\mathbf{u}) = \begin{cases} 1 & |u_j| \leq 1/2 \quad j = 1, ..., d \\ 0 & \text{otherwise.} \end{cases}$$

Podemos calcular k_n (cantidad de muestras que caen en R_n) como:

$$k_n = \sum_{i=1}^{n} \varphi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n}\right),$$



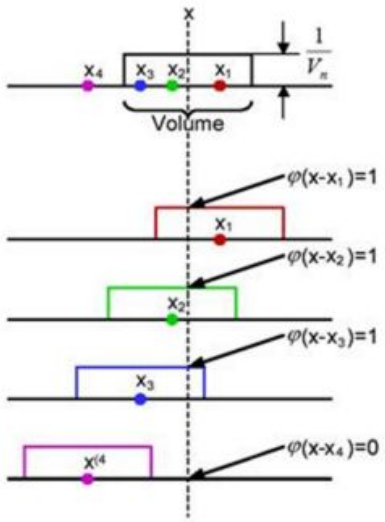
Estimación por ventanas de Parzen

- Sea R_n un hipercubo (d-dimensional) de lado h_n ($V_n = h_n^d$)
- Sea $\varphi(\mathbf{u})$ la ventana:

$$\varphi(\mathbf{u}) = \begin{cases} 1 & |u_j| \leq 1/2 \quad j = 1, ..., d \\ 0 & \text{otherwise.} \end{cases}$$

Podemos calcular k_n (cantidad de muestras que caen en R_n) como:

$$k_n = \sum_{i=1}^{n} \varphi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n}\right),$$



Por lo que

$$p_n(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{V_n} \varphi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n}\right)$$

Estimación por ventanas de Parzen

$$p_n(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{V_n}\varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right)$$

- Esta ecuación sugiere una manera más general para estimar funciones de densidad definiendo una función de ventana
- La ventana $\varphi(\mathbf{u})$ cumple un rol de *interpolador*

Estimación por ventanas de Parzen

$$p_n(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{V_n}\varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right)$$

- Esta ecuación sugiere una manera más general para estimar funciones de densidad definiendo una función de ventana
- La ventana $\varphi(\mathbf{u})$ cumple un rol de *interpolador*

- Qué necesitamos para que $p_n(x)$ sea una densidad de probabilidad? Si

$$\varphi(\mathbf{x}) \geq 0 \qquad \int \varphi(\mathbf{u})\, d\mathbf{u} = 1,$$

y además se mantiene que $V_n = h_n^d$ entonces $p_n(x)$ hereda estas condiciones.

Estimación por ventanas de Parzen

Qué efecto tiene el ancho de la ventana h_n en la estimación de $p(x)$?

Estimación por ventanas de Parzen

Qué efecto tiene el ancho de la ventana h_n en la estimación de p(x)?

Sea $$\delta_n(\mathbf{x}) = \frac{1}{V_n}\varphi\left(\frac{\mathbf{x}}{h_n}\right)$$

Podemos reescribir $p_n(x)$ como:

$$p_n(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\delta_n(\mathbf{x}-\mathbf{x}_i).$$

Estimación por ventanas de Parzen

Qué efecto tiene el ancho de la ventana h_n en la estimación de p(x)?

Sea $$\delta_n(\mathbf{x}) = \frac{1}{V_n}\varphi\left(\frac{\mathbf{x}}{h_n}\right)$$

Podemos reescribir $p_n(x)$ como:

$$p_n(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\delta_n(\mathbf{x}-\mathbf{x}_i).$$

- h_n afecta el ancho de $\boldsymbol{\delta}_n(x)$ tanto como su amplitud de manera de que su área sea uno (approx. masa de Dirac)

Estimación por ventanas de Parzen

Qué efecto tiene el ancho de la ventana h_n en la estimación de p(x)?

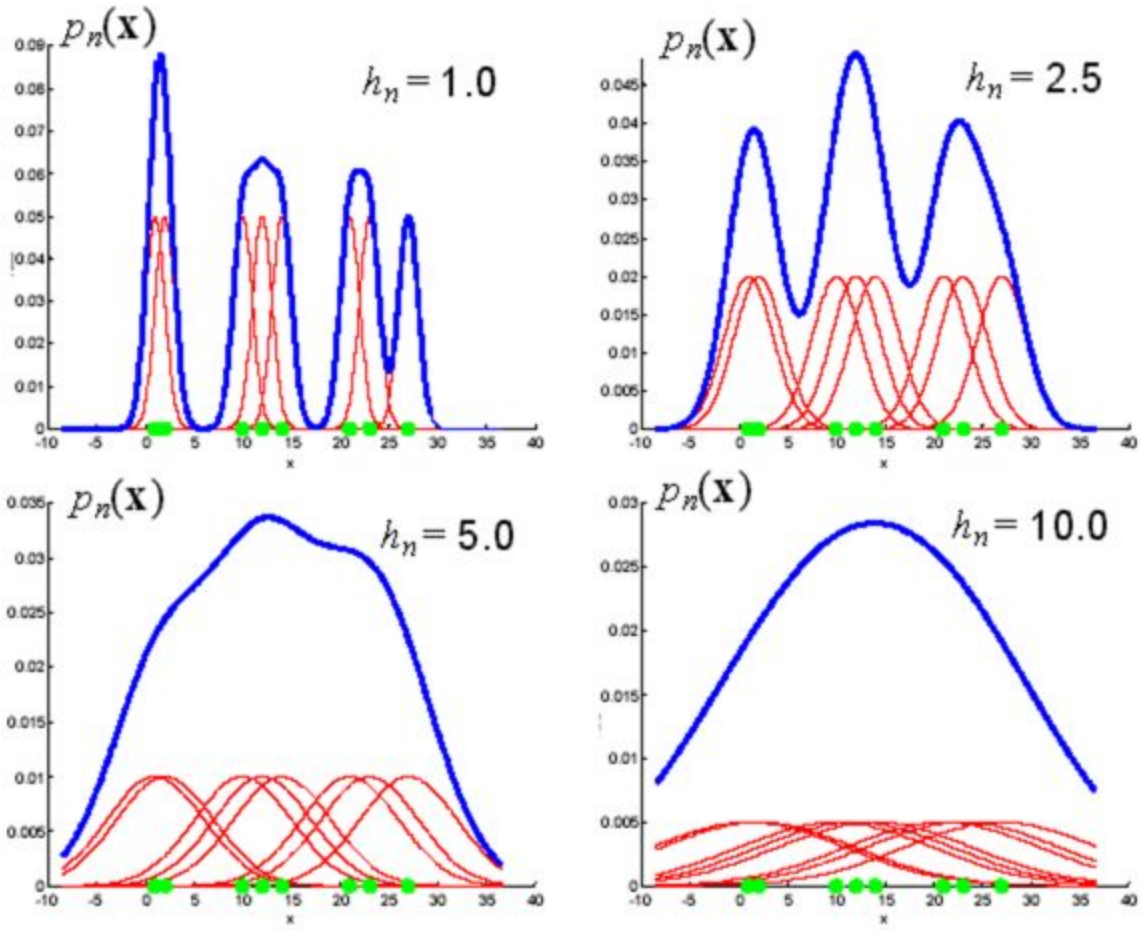
Sea $$\delta_n(\mathbf{x}) = \frac{1}{V_n}\varphi\left(\frac{\mathbf{x}}{h_n}\right)$$

Podemos reescribir $p_n(x)$ como:

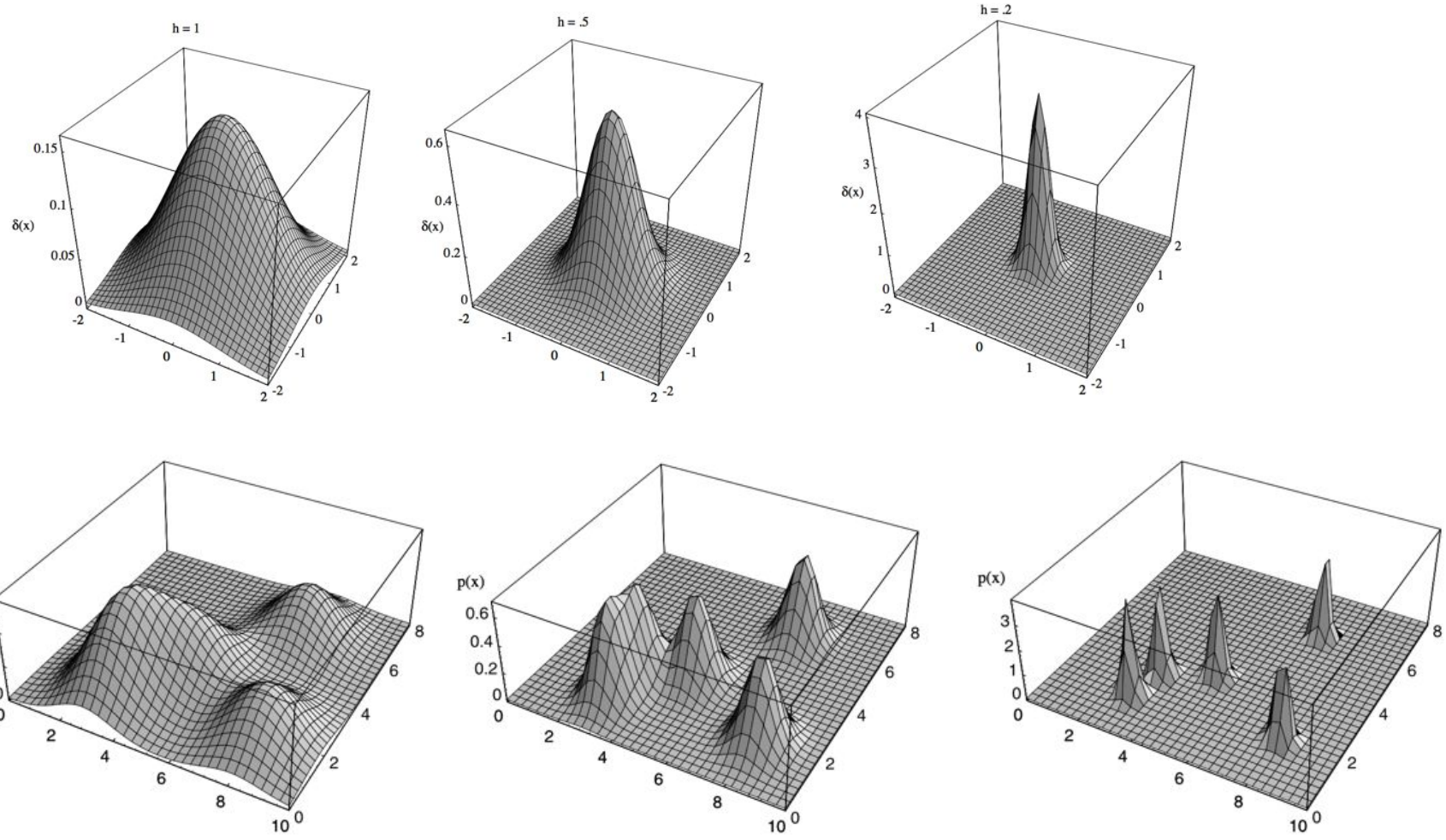
$$p_n(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\delta_n(\mathbf{x}-\mathbf{x}_i).$$

- h_n afecta el ancho de $\boldsymbol{\delta}_n(x)$ tanto como su amplitud de manera de que su área sea uno (approx. masa de Dirac)
- h_n es grande ⇒ amplitud de $\boldsymbol{\delta}_n(x)$ es pequeña y $\boldsymbol{\delta}_n(x\text{-}x_i)$ cambia lento ya que la ventana es ancha. Por lo tanto $p_n(x)$ es la superposición de funciones de cambio lento y el (estimador sobre-regularizado)
- h_n es pequeño ⇒ amplitud de $\boldsymbol{\delta}_n(x)$ es grande y $\boldsymbol{\delta}_n(x\text{-}x_i)$ cambia rápido ya que la ventana es angosta. Por lo tanto $p_n(x)$ es la superposición de funciones de cambio rápido centradas en las muestras (estimador ruidoso)

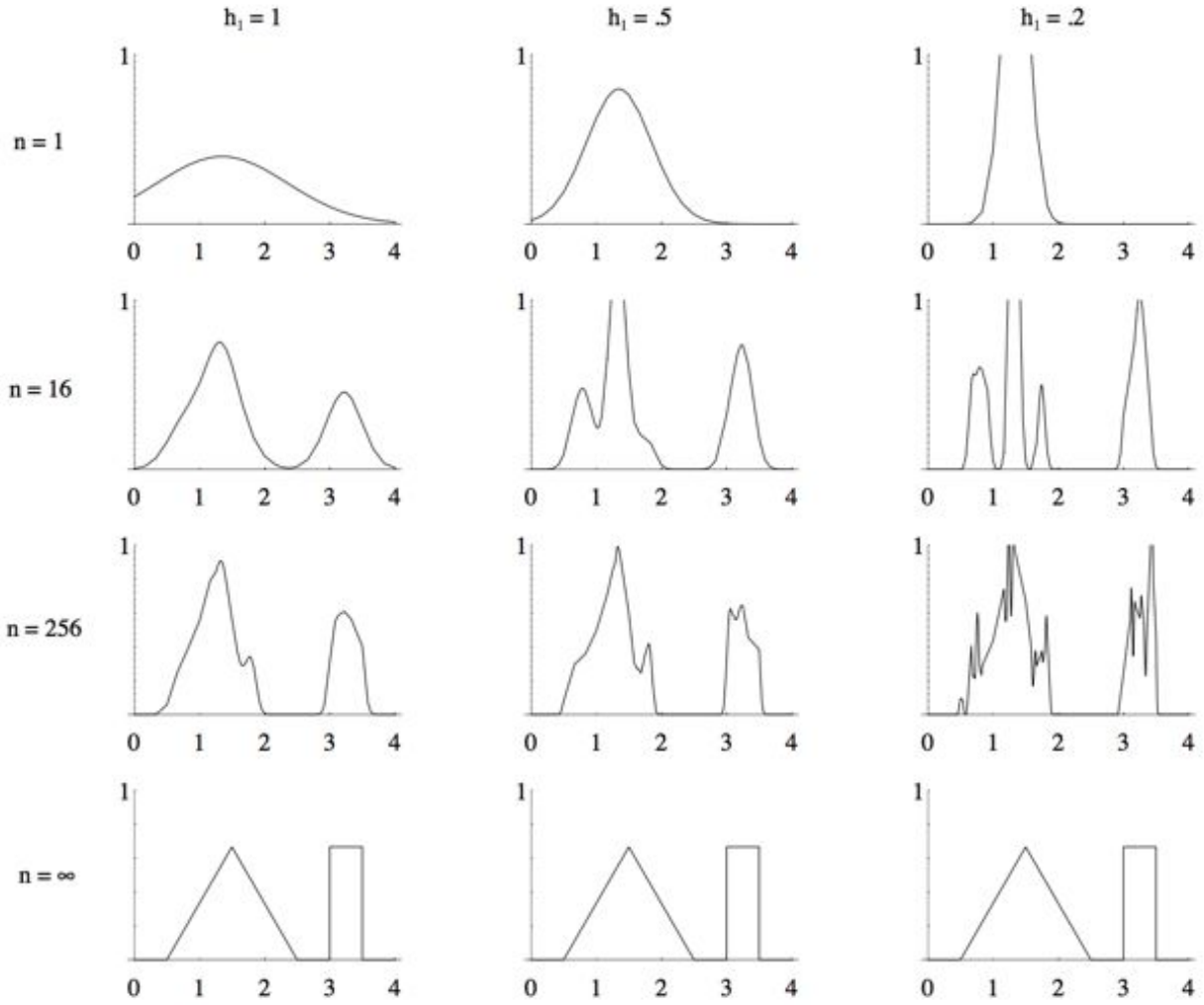
Estimación por ventanas de Parzen



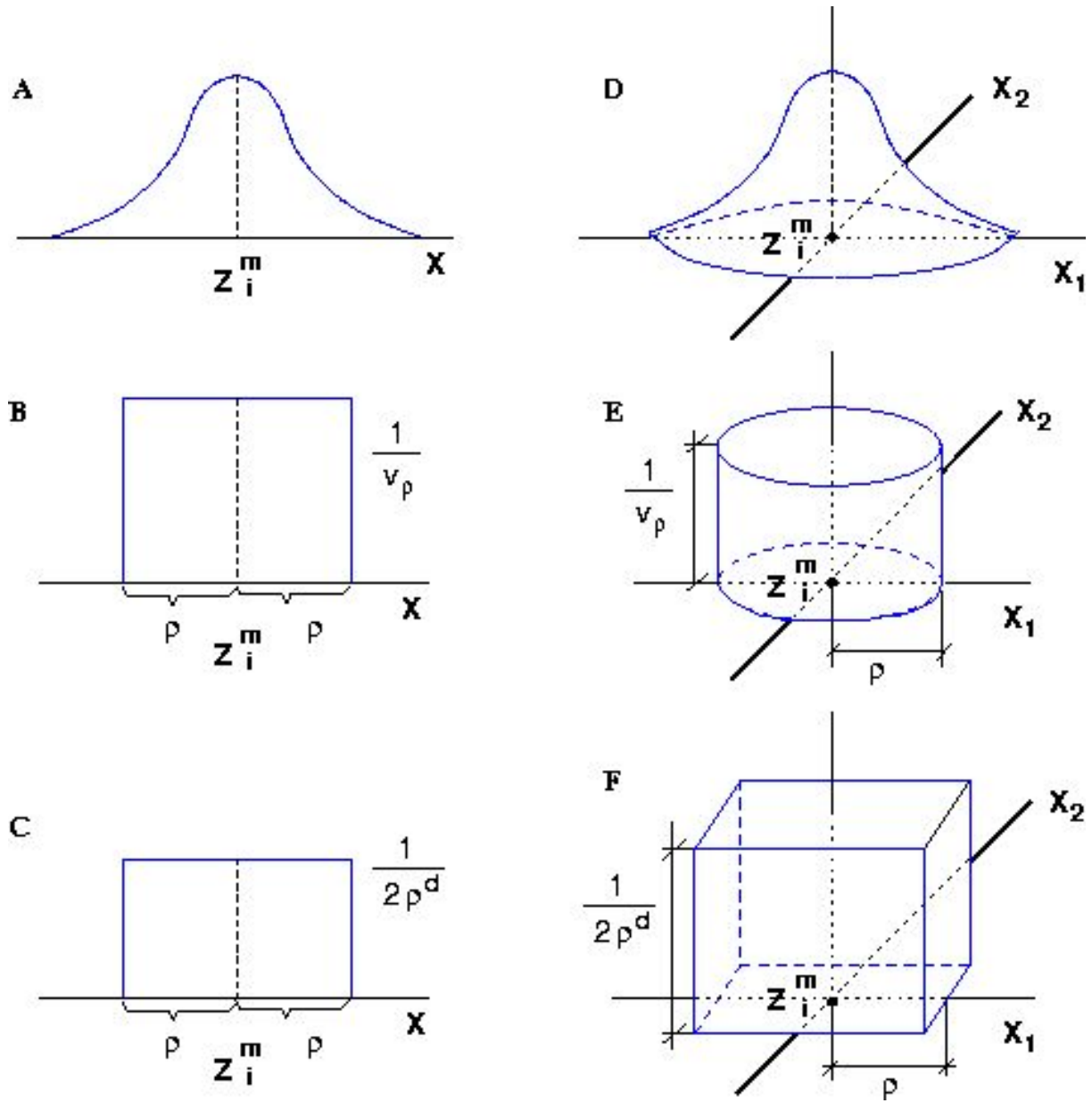
Estimación por ventanas de Parzen



Estimación por ventanas de Parzen



- Ventana Gaussiana
- $h_n = h_1/\mathrm{sqrt}(n)$


A
z_i^m
X
B
$\frac{1}{v_\rho}$
X
ρ
z_i^m
ρ
C
$\frac{1}{2\rho^d}$
X
ρ
z_i^m
ρ
D
X_2
z_i^m
X_1
E
X_2
$\frac{1}{v_\rho}$
z_i^m
X_1
ρ
F
X_2
$\frac{1}{2\rho^d}$
z_i^m
X_1
ρ

Núcleo Gaussiano

$$h(\mathbf{x}) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\mathbf{x}^2}$$

$$K(\mathbf{x}, \mathbf{Z}_i^m) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{Z}_i^m)^T \sum^{-1} (\mathbf{x}-\mathbf{Z}_i^m)}$$

- Distancia de Mahalanobis, ρ distinto en cada dimensión y se contempla correlación entre variables. Si matriz es diagonal, la correlación es 0 y la distancia es euclídea.
- Estimador suave, computacionalmente muy costoso

Núcleo hiperesférico

$$K(\mathbf{x}, \mathbf{Z}_i^m) = \begin{cases} \dfrac{1}{V} & \text{si } \left\{ d(\mathbf{x}, \mathbf{Z}_i^m) \leq \rho \right\} \\ 0 & \text{si } \left\{ d(\mathbf{x}, \mathbf{Z}_i^m) > \rho \right\} \end{cases}$$

- Ventaja: eficiencia computacional (cálculo de distancia y suma). Útil cuando tengo muchas muestras.
- Desventaja: estimación constante por tramos

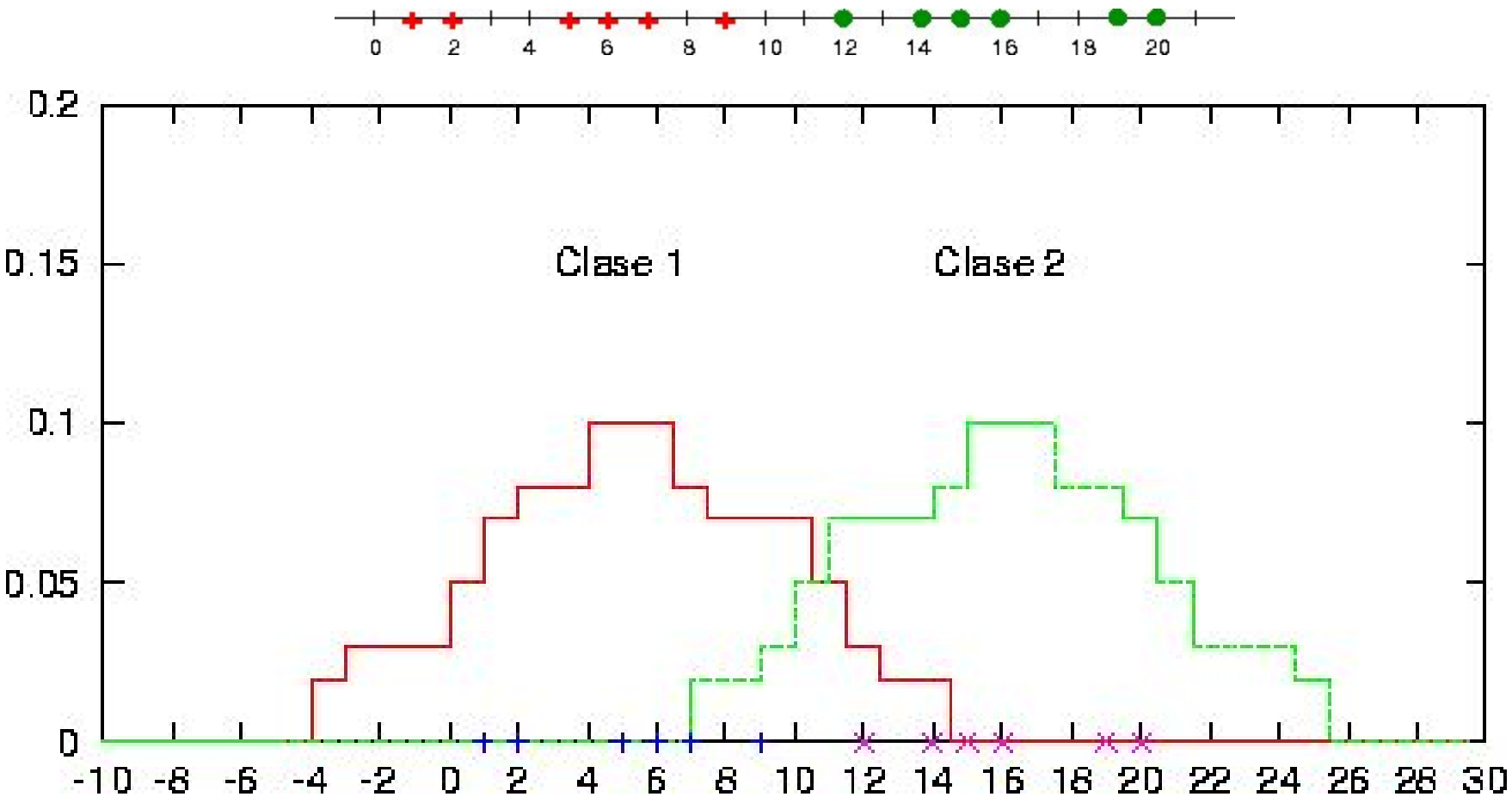
Núcleo hipercúbico

$$K(\mathbf{x}, \mathbf{Z}_i^m) = \begin{cases} (2\rho)^{-d} & \text{si} \quad \{\delta_T(\mathbf{x}, \mathbf{Z}_i^m) \leq \rho\} \\ 0 & \text{si} \quad \{\delta_T(\mathbf{x}, \mathbf{Z}_i^m) > \rho\} \end{cases}$$

$$\delta_T(\mathbf{x}, \mathbf{Z}_i^m) = \underset{j=1..d}{máx}\left\{\left|\mathbf{x}_j - \mathbf{Z}_i^m\right|\right\} \text{ Distancia de Chevyshev}$$

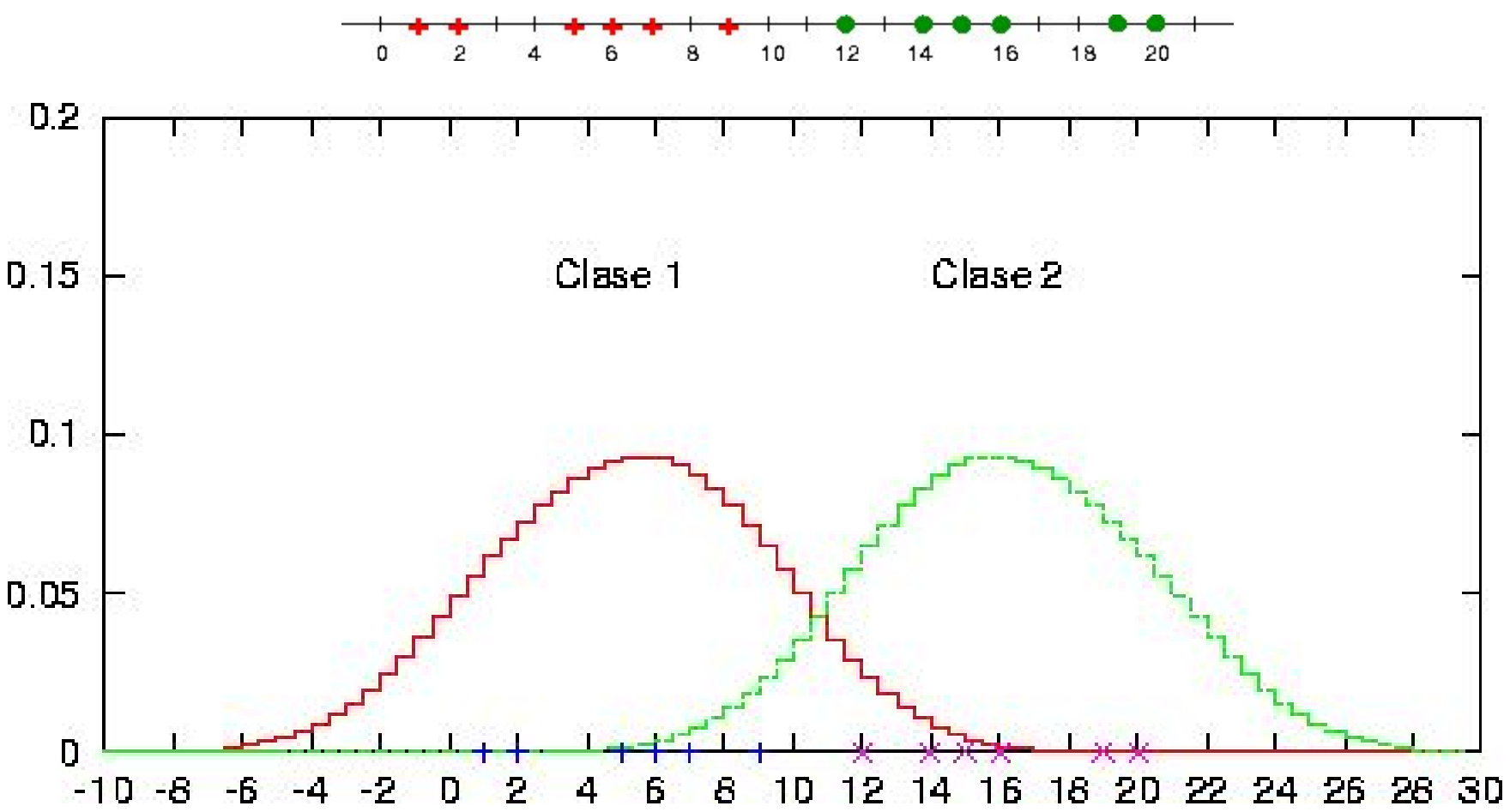
- Ventaja: eficiencia computacional .Cálculo de la distancia más eficiente.
- Desventaja: estimación constante por tramos

Ejemplo



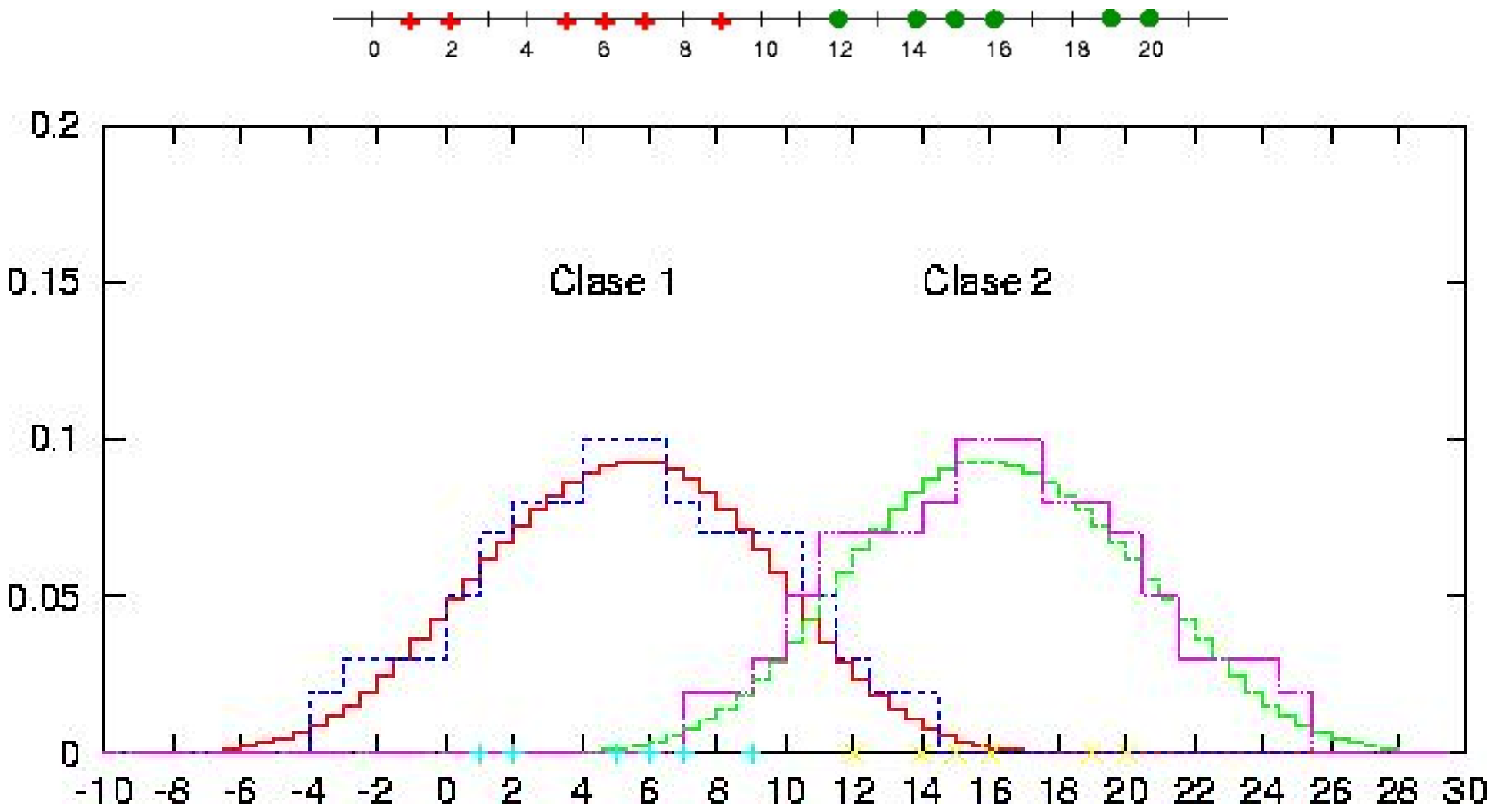
Hipercubo $\rho=5$

Ejemplo



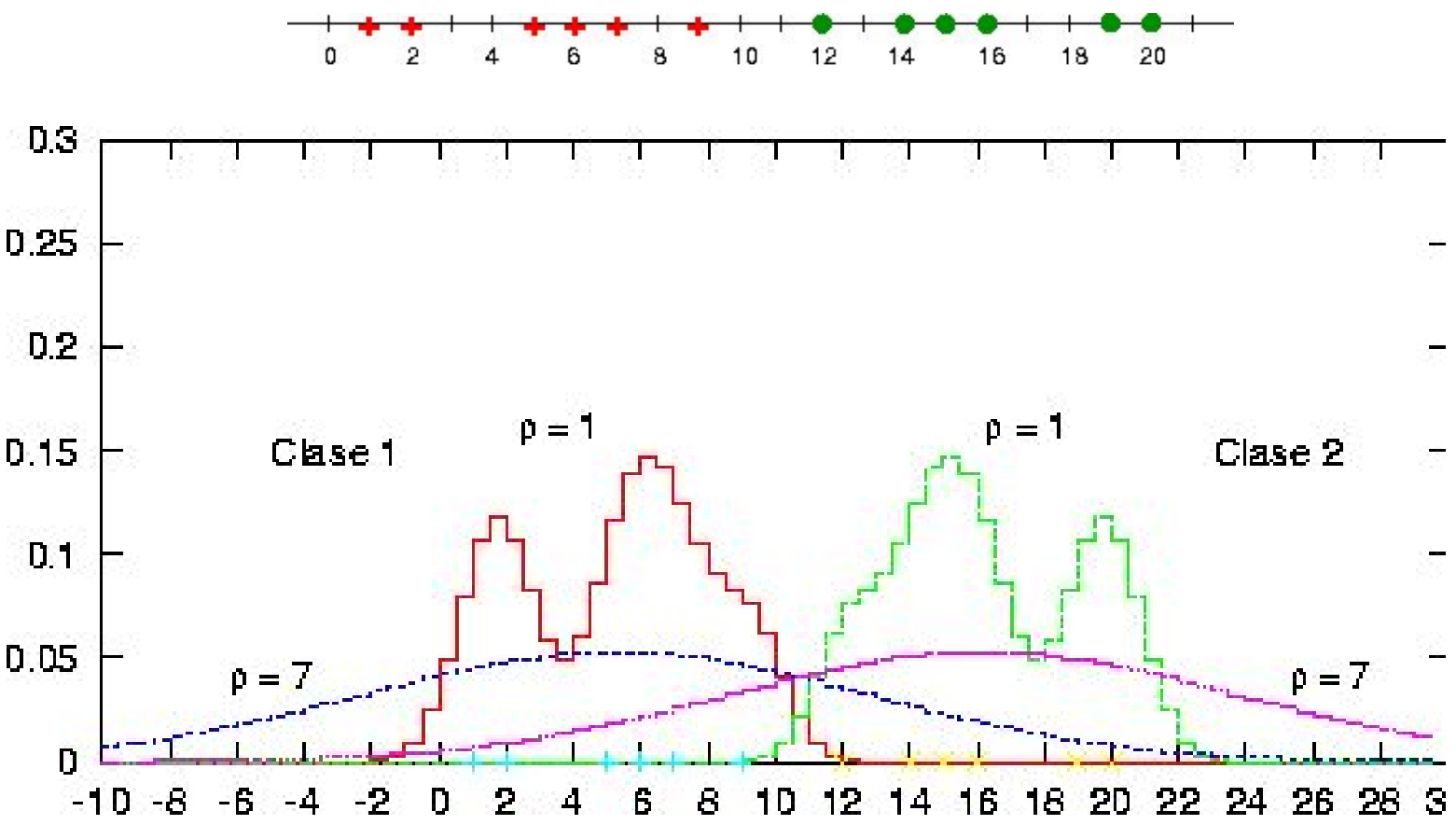
Gaussiano $\rho=3$

Ejemplo



Comparación

Ejemplo



Comparación para un mismo núcleo y distintos anchos

Ejemplo Clasificación

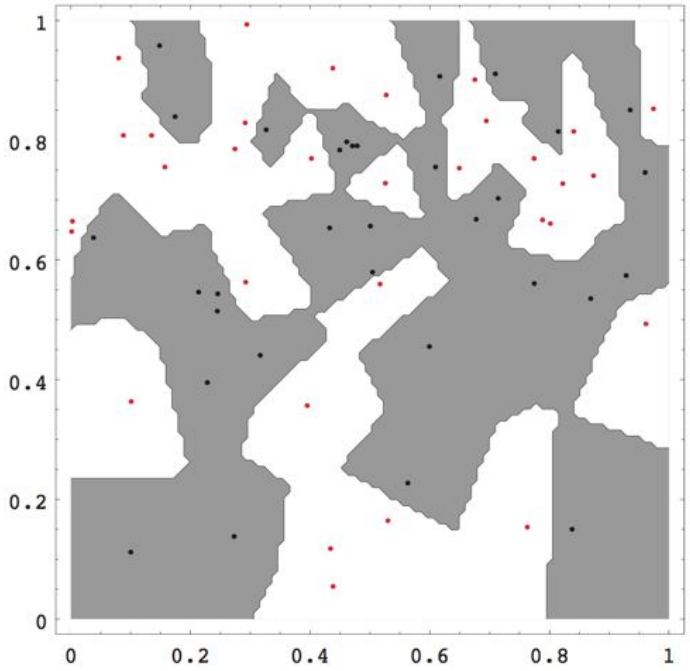
- Se estiman las densidades $p(x/w_i)$ usando el conjunto de entrenamiento de cada clase. Si las clases no son equiprobables, se estiman los términos $p(w_i)$

Ejemplo Clasificación

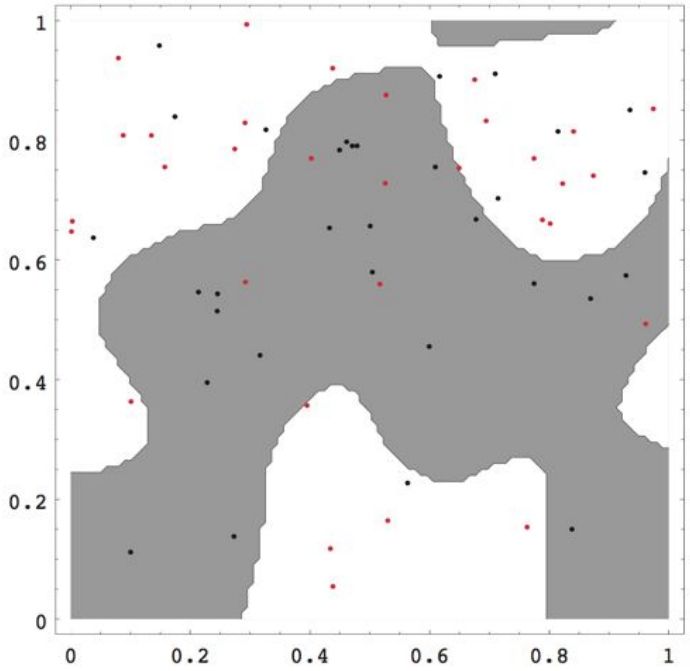
- Se estiman las densidades $p(x/w_i)$ usando el conjunto de entrenamiento de cada clase. Si las clases no son equiprobables, se estiman los términos $p(w_i)$
- Las fronteras de decisión con un estimador que usa ventanas de Parzen dependen de la ventana elegida (y de su ancho)
- El error en entrenamiento puede ser arbitrariamente pequeño al elegir ventanas suficientemente pequeñas. Importante: Controlar el error en etapa de testeo (nuevas muestras).

Ejemplo Clasificación

h pequeño



h grande



- Se puede ajustar el ancho de la ventana mediante validación cruzada.
- Se parte el conjunto de muestras disponibles en dos conjuntos y se entrena con uno y se calcula el error con el otro.
- Se busca el ancho que minimiza el error de clasificación en validación.

Estimación mediante los k vecinos más próximos

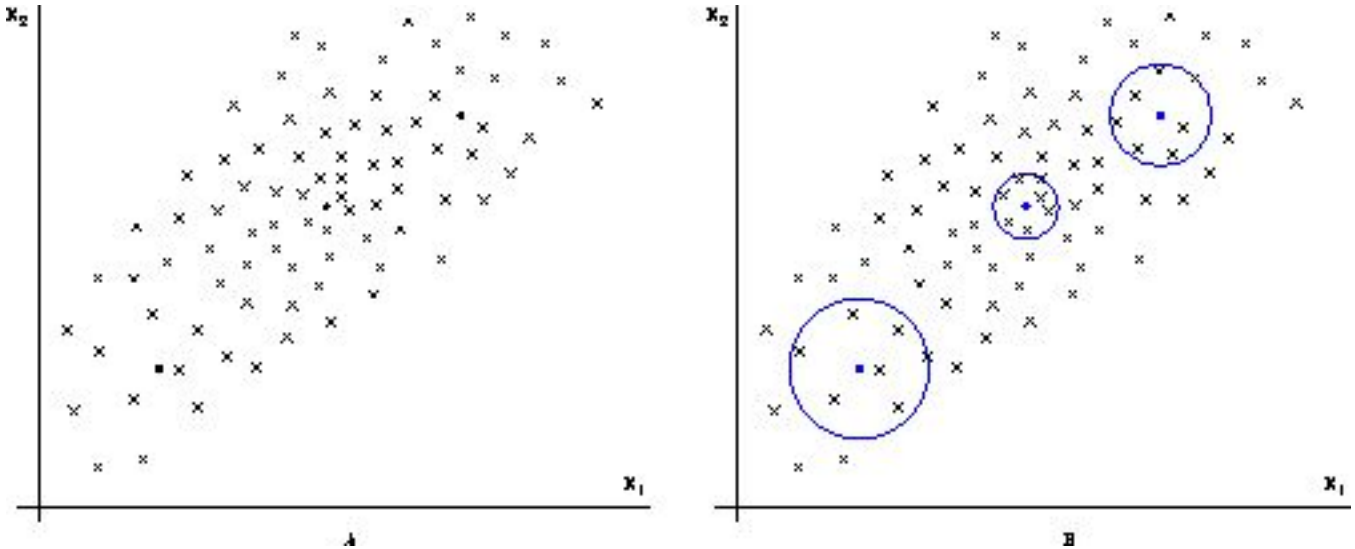
- Una manera de eliminar el problema de la elección de la función ventana es dejar que sea determinada por el conjunto de entrenamiento.

- Para estimar p(x) a partir de un conjunto de n muestras, podemos centrar una celda en x y hacerla crecer hasta que capturar las k_n muestras más cercanas a x
  (k_n es un parámetro).

Estimación mediante los k vecinos más próximos

- Si la densidad p(x) es muy alta en x, entonces la celda será pequeña (buena resolución)
- Si la densidad p(x) es muy baja en x, entonces la celda será grande.

$$p_n(\mathbf{x}) = \frac{k_n/n}{V_n}$$

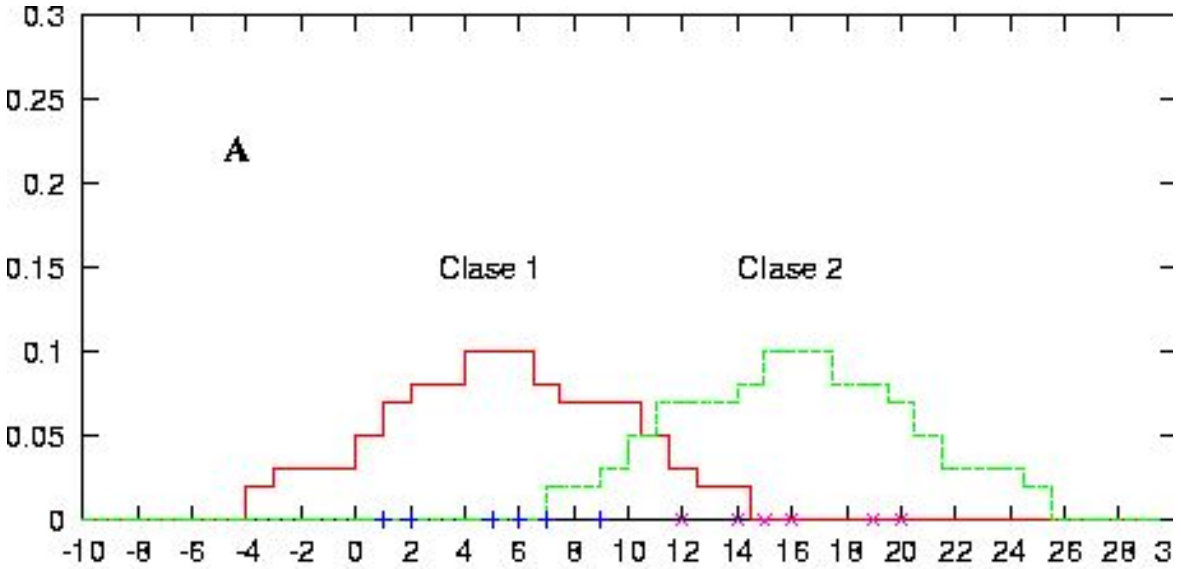


Elección de k

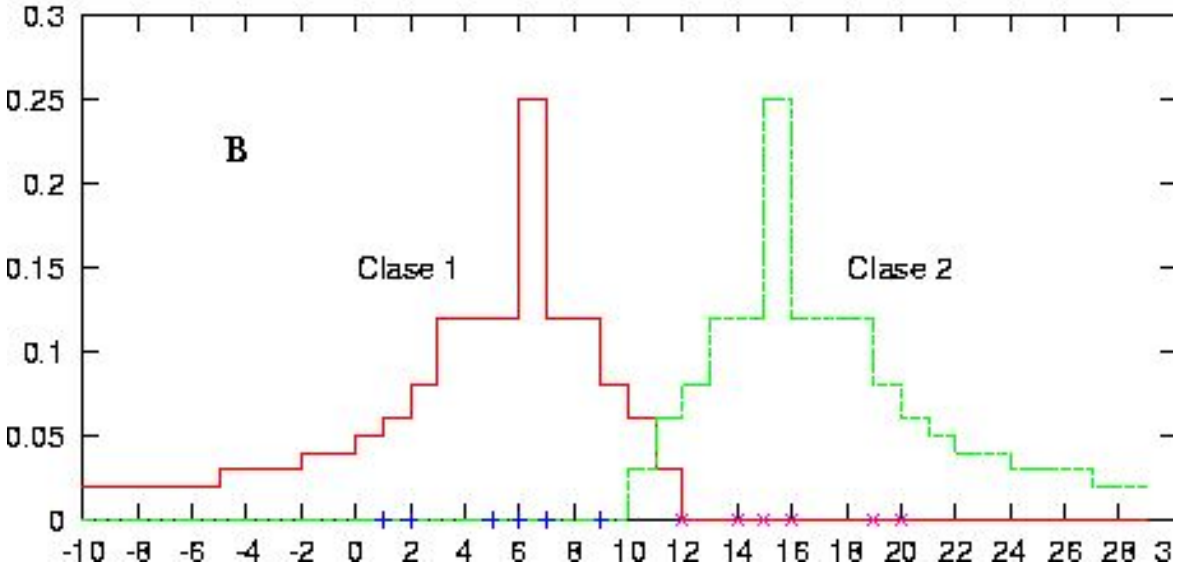
$$p_n(\mathbf{x}) = \frac{k_n/n}{V_n}$$

- k debe depender de n (es decir k_n)
- (i) $k_n \rightarrow \infty$ (con $n \rightarrow \infty$) , para asegurar $k_n/n \rightarrow$ Prob x en V_n
- (ii) $k_n/n \rightarrow 0$ (con $n \rightarrow \infty$) es necesario para que $Vn \rightarrow 0$
- Se puede probar que (i) y (ii) son condiciones necesarias y suficientes para que $p_n(x)$ converja a p(x) en probabilidad en los puntos donde p(x) es continua.
- Podemos elegir por ejemplo k_n = c/sqrt(n)

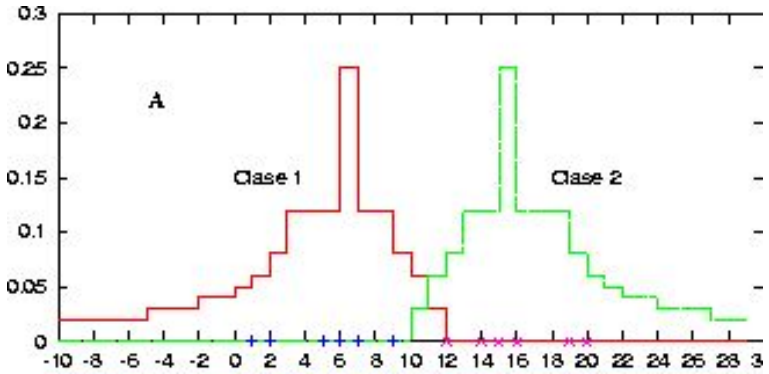
Hipercubo ρ=5



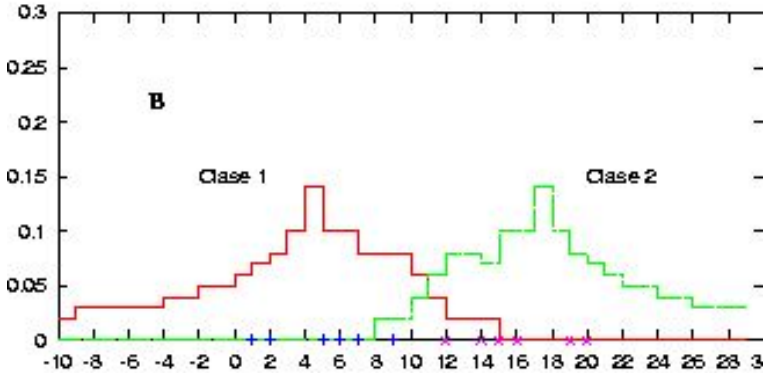
k-vecinos *k*=3



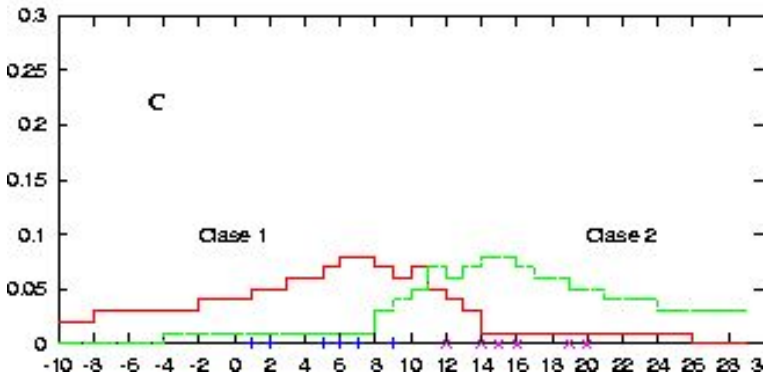
k-vecinos *k*=3



k-vecinos *k*=5



k-vecinos *k*=7



Estimación directa de las probabilidades a posteriori

$$p_n(\mathbf{x}, w_i) = p_n(\mathbf{x} / w_i) P(w_i) = \frac{k_i}{n_i V} \frac{n_i}{n} = \frac{k_i}{nV}$$

$$P_n(w_i / \mathbf{x}) = \frac{p_n(\mathbf{x}, w_i)}{\sum_{j=1}^{c} p_n(\mathbf{x}, w_j)} = \frac{k_i}{k}$$

Regla de clasificación de los k-vecinos más cercanos k-NN

- La regla de decisión que minimiza el error: elegimos para **x** la clase más frecuente de la celda.
- Seleccionamos w_j si $k_j = \max_{i=1..c} \{k_i(\mathbf{x})\}$
- $k_i(\mathrm{x})$: número de muestras de la clase w_i entre los k vecinos más cercanos a **x**.

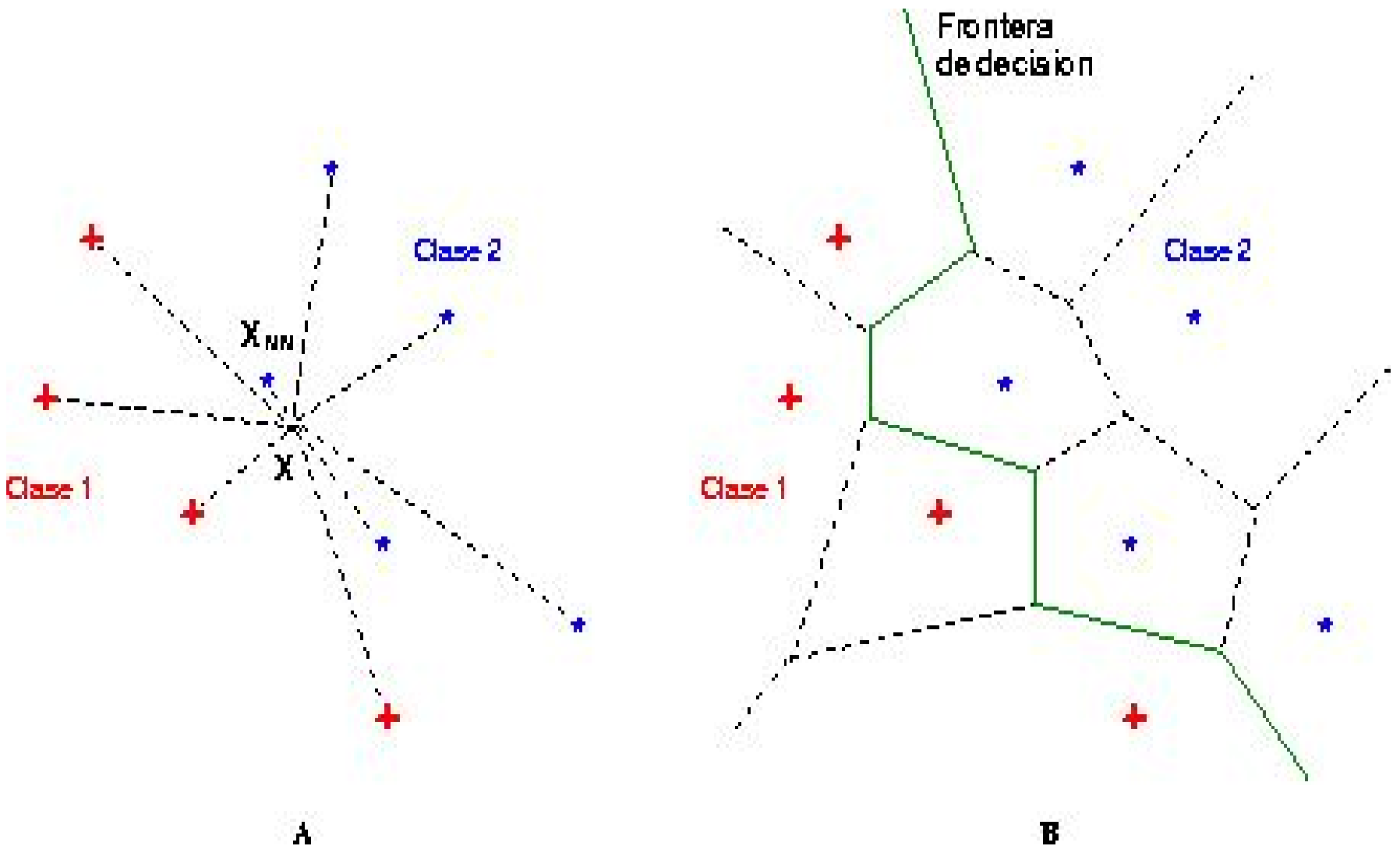
Regla del vecino más cercano 1-NN

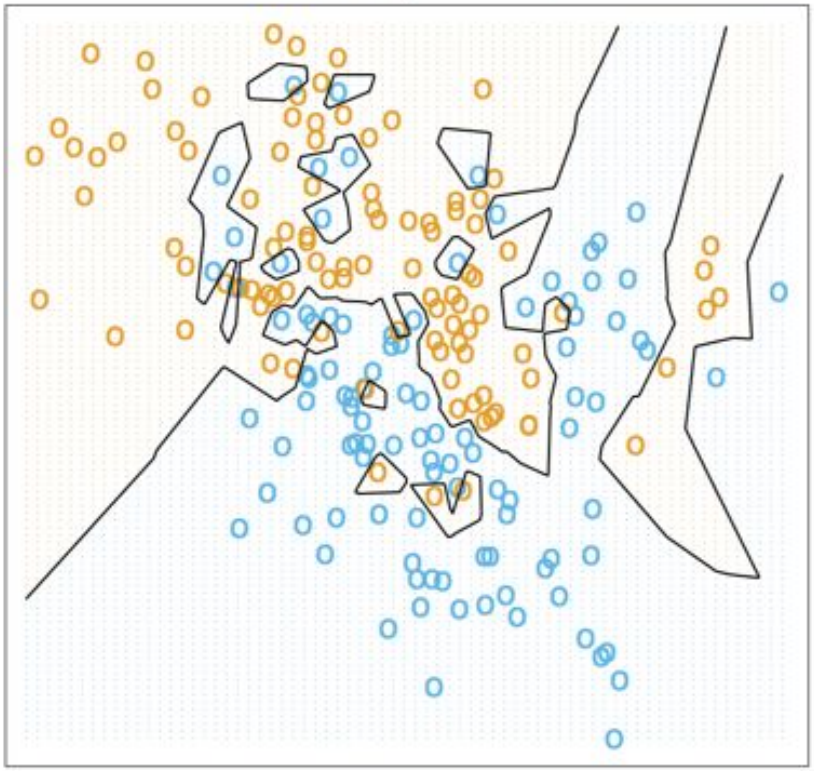
- Se selecciona w_j si:

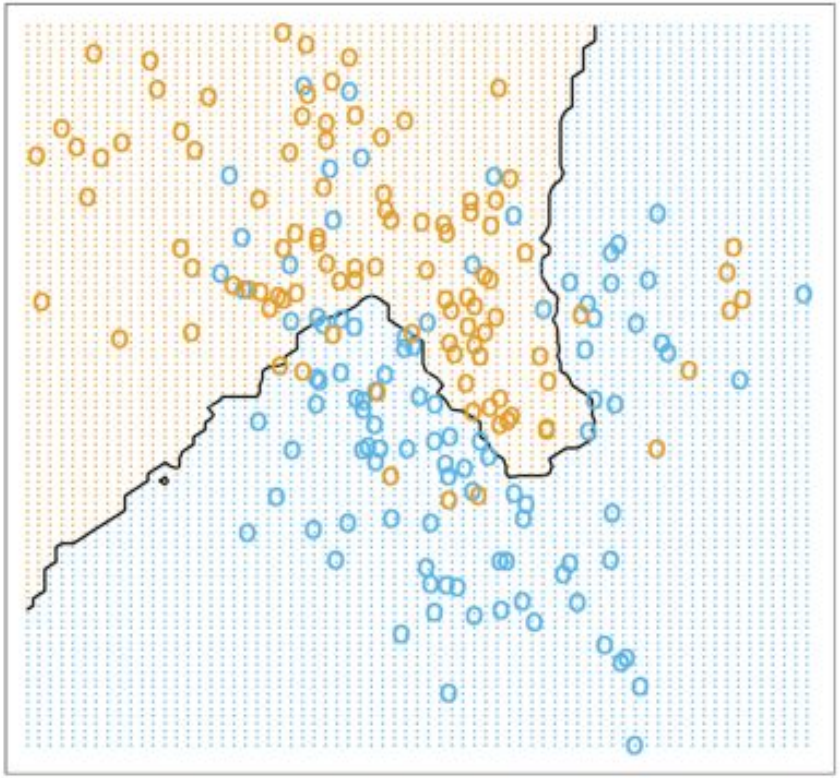
$$d(x, x_{NN}) = \min_{i=1..n} \{\delta(x, x_i)\}$$

$$x_{NN} \in w_j$$

- Interpretación:
  - El espacio es dividido en n celdas (regiones de Voronoi)


Frontera
de decisión
Clase 2
Clase 1
x_{NN}
x
A
B


1−Nearest Neighbor Classifier


15-Nearest Neighbor Classifier

Cotas de Error de 1-NN

- Procedimiento subóptimo, tasa de error mayor que la de Bayes.
- Con un número ilimitado de prototipos el error nunca es mayor que 2 veces Bayes

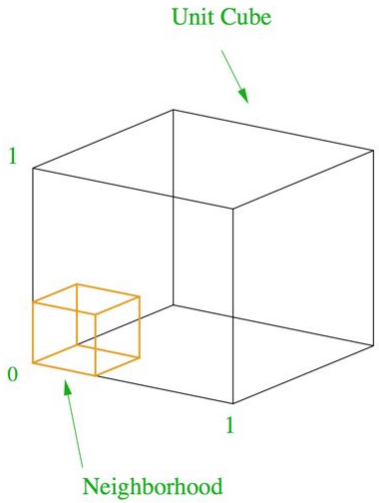
$$E^* < E_1 \leq E^*\left(2 - \frac{c}{c-1}E^*\right)$$

E^*: Error óptimo - Clasificador Bayesiano,

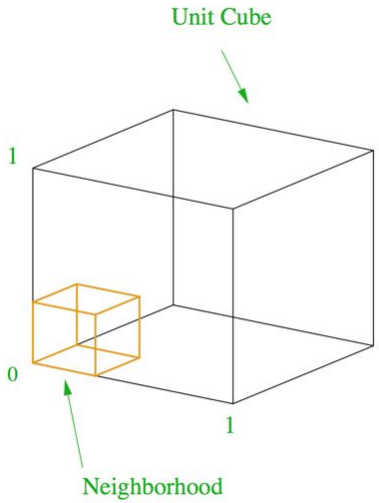
c: número de clases. [Prueba en Duda-Hart-Stork]

Curse of Dimensionality
(La maldición de la dimensionalidad)

- Sean N puntos distribuidos uniformemente en el Hipercubo de lado 1 en dimensión p,
- R: Entorno hipercubo alrededor de un punto x

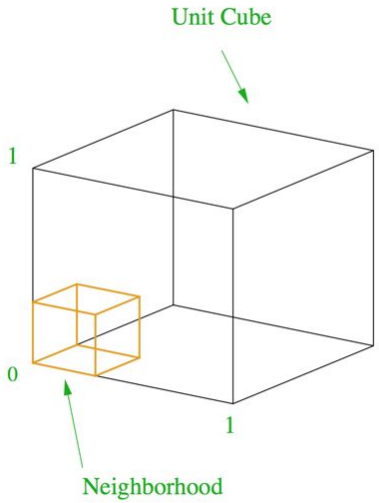


Curse of Dimensionality
(La maldición de la dimensionalidad)



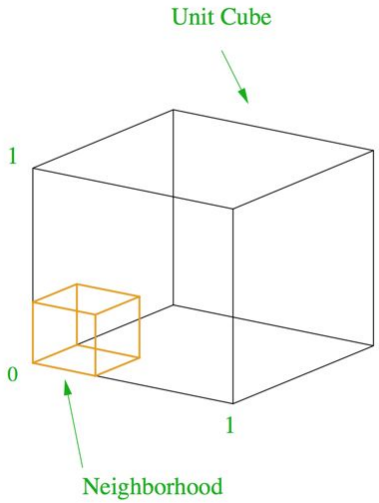
- Sean N puntos distribuidos uniformemente en el Hipercubo de lado 1 en dimensión p,
- R: Entorno hipercubo alrededor de un punto x
- Deseo que una fracción r de los N puntos estén contenidos en R ==> Vol(R) = r

Curse of Dimensionality
(La maldición de la dimensionalidad)



- Sean N puntos distribuidos uniformemente en el Hipercubo de lado 1 en dimensión p,
- R: Entorno hipercubo alrededor de un punto x
- Deseo que una fracción r de los N puntos estén contenidos en R ==> Vol(R) = r
- El lado de ese entorno tiene largo $e_p = r^{1/p}$.

Curse of Dimensionality
(La maldición de la dimensionalidad)



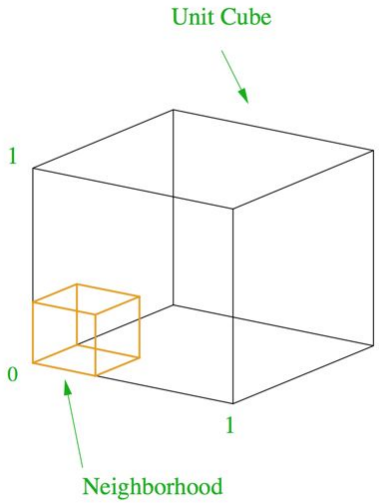
- Sean N puntos distribuidos uniformemente en el Hipercubo de lado 1 en dimensión p,
- R: Entorno hipercubo alrededor de un punto x
- Deseo que una fracción r de los N puntos estén contenidos en R ==> Vol(R) = r
- El lado de ese entorno tiene largo $e_p = r^{1/p}$.

- Si queremos que el 1% de los puntos dentro, entonces:

  Dimensión p=10 → $e_{10} = 0.63$!!!

  Dimensión p=100→ $e_{100} = 0.95$!! No parece muy local :)

Curse of Dimensionality
(La maldición de la dimensionalidad)



- Sean N puntos distribuidos uniformemente en el Hipercubo de lado 1 en dimensión p,
- R: Entorno hipercubo alrededor de un punto x
- Deseo que una fracción r de los N puntos estén contenidos en R ==> Vol(R) = r
- El lado de ese entorno tiene largo $e_p = r^{1/p}$.

- Si queremos que el 1% de los puntos dentro, entonces:

  Dimensión p=10 → $e_{10} = 0.63$!!!

  Dimensión p=100→ $e_{100} = 0.95$!! No parece muy local :)

Reducir r tampoco sirve porque aumenta la varianza del estimador...

Costo Computacional k-NN

- Requiere explorar todo el conjunto de referencia O(n).
- Cálculo de la distancia euclídea lineal con d O(nd)
- Espacio de almacenamiento de todos los prototipos O(nd)
- **Inaplicable** si tengo un conjunto de referencia grande y alta dimensionalidad.

Vecino más cercano

- **Ventajas**: poderoso, fácil de implementar y pronto para trabajar una vez que se cargan los vectores de referencia. Puede paralelizarse.

- **Desventajas**: Puede establecer incontables pequeñas regiones en zonas de alta penetración con clases competitivas. La ambigüedad queda escondida por una decisión arbitraria que acarrea un gran esfuerzo computacional. *Solución:* Regla del vecino más cercano restringida (clase de rechazo).

Regla 1-NN(t)

$$d(\mathbf{x}) = \begin{cases} w_j & \delta(x, x_{NN}) = \min_{i=1..n}(\delta(x, x_i)) \le t \\ w_0 & \text{en otro caso} \end{cases}$$

- t se elige luego de examinar la distribución de las distancias de los patrones.

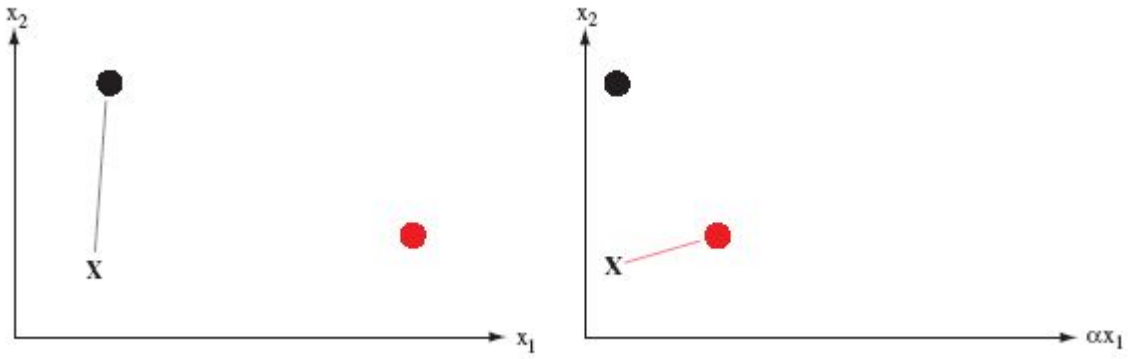
- Regla del vecino más cercano confía en la métrica o distancia utilizada.

Métrica de Minkowski $\quad \mathrm{L_k}(a,b) = \left(\sum_{i=1}^{d} |a_i - b_i|^k \right)^{1/k}$

Norm $\mathrm{L_k}$ $\qquad \mathrm{L_1}$: Manhatan, $\qquad \mathrm{L_2}$: Euclidea

Re-escalado

- En el cálculo de la distancia hay que tener en cuenta que diferentes componentes pueden tener diferentes escalas (características) por lo que los datos deben re-escalarse previamente



Re-escalado

Curse of Dimensionality
(La maldición de la dimensionalidad)

- Supongamos que tiro al azar (uniformemente) N puntos en la Esfera p dimensional de radio 1.
- Sea d la distancia del origen al punto más cercano.
- Obs. d es un variable aleatoria.
- Cuál es su mediana?

Nota la mediana de d es d_m si $P(d<=d_m) = P(d>=d_m) = 1/2$