Sistemas de Visión Artificial

Tema 5. Técnicas de reconocimiento (3ª parte)

Sira E. Palazuelos Cagigas

Luis M. Bergasa Pascual

Índice

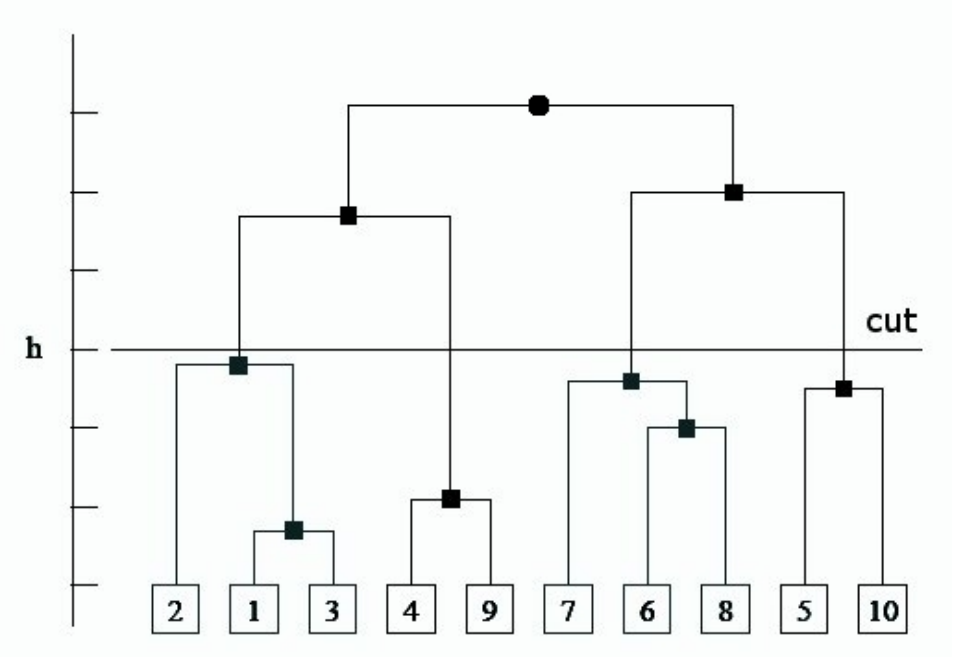
4. Técnicas de *clustering* basadas en distancias
5. Clasificadores estadísticos. Teoría estadística de la decisión.
6. Máquinas de vectores de soporte (*support vector machines)*

Universidad de Alcalá

4. Técnicas de *clustering* basadas en distancias

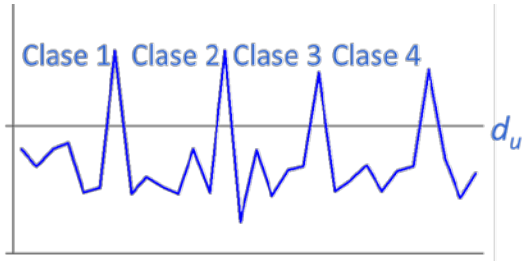
Clustering basado en distancias

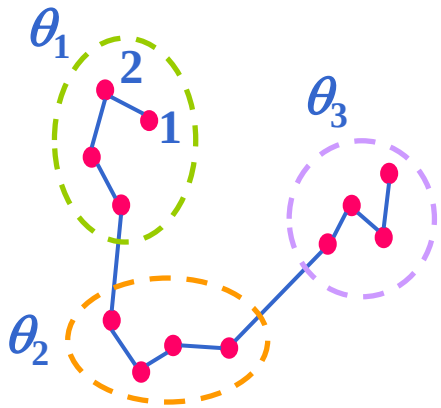
- ***Clustering:*** métodos de clasificación no supervisada o agrupamiento.
- Algunos algoritmos de ***clustering*** **basados en distancias**:
  - **Distancias encadenadas.**
  - **K-medias.**
  - **Máx-mín.**
  - **Algoritmos de *clustering* jerárquico** o los **dendrogramas** (pueden resultar ineficientes para tamaños grandes de muestra).



Clustering basado en distancias

- **Algoritmo de las distancias encadenadas**
  - **No** precisa de información sobre el **número de clases existente**.
  - Se realiza en un solo paso, por lo puede ser muy **rápido**.
  - Inconvenientes: necesita **fijar umbrales de clasificación** y la **solución final puede depender del punto de inicio del algoritmo**.
  - Algoritmo:
    - Se elije al azar uno de los datos de entrenamiento $X_{i(0)}$.
    - Se ordenan los vectores según la sucesión: $X_{i(0)}$, $X_{i(1)}$, …,$X_{i(P-1)}$, de forma que el siguiente vector de la cadena es el más próximo al anterior (que no esté ya en la cadena).
    - Se establecen las clases: se analizan las distancias d_i entre cada elemento y el siguiente:
      - Si $d_i > d_u$ (umbral) → comienza una nueva clase.
      - Si $d_i < d_u$, el elemento analizado pertenece a la misma clase que el elemento anterior.





www.vision.uji.es/~sotoca/docencia/rfv1-master/clustering.ppt

Clustering basado en distancias

- **Algoritmo de las K-medias (*k-means*)**
  - El **algoritmo de las K-medias** busca formar K *clusters* (clases), que serán representados por sus centroides o prototipos.
  - Cada **centroide** es el valor medio de los datos que pertenecen a su clase.
  - El algoritmo trata de **minimizar suma de las distancias** de los objetos al centroide de la clase a las que pertenezcan:

$$J_k = \sum_{x \in \alpha_k(t)} \left\| X - z_k(t) \right\|^2 \quad k = 1,2,\ldots,K$$

  - **Necesita conocer a priori el número de clases, K.**

Clustering basado en distancias

- El **algoritmo k-medias básico** consta de los siguientes pasos (I):
  1. **Elegimos aleatoriamente** K vectores, que serán los **centroides (z) iniciales** de las K clases (α). Pueden ser valores aleatorios o los podemos elegir entre los elementos a agrupar.

$$\alpha_1 : z_1(0); \alpha_2 : z_2(0); ....; \alpha_K : z_K(0)$$

  2. Se calcula la **distancia** de cada objeto (dato) a cada uno de los **centroides**. El **dato se asigna al grupo cuyo centroide esté más cerca** (distancia mínima).

$$For \quad i = 1 \quad to \quad N \quad \text{(N=número de datos de entrenamiento)}$$

$$x_i \in \alpha_j(t) \qquad si \qquad \| x_i - z_j(t) \| < \| x_i - z_n(t) \|$$

$$\forall n = 1, 2, ..., K / n \neq j$$

$$End$$

Clustering basado en distancias

- El **algoritmo k-medias básico** consta de los siguientes pasos (II):

3. Se **actualizan los centroides** con el valor medio de todos los objetos asignados a ese grupo.

$$z_j(t+1) = \frac{1}{N_j(t)} \sum_{i=1}^{N_j(t)} x_i$$

j=1,2,...,K x_i: objetos asignados a $\alpha_j(t)$ en esta iteración

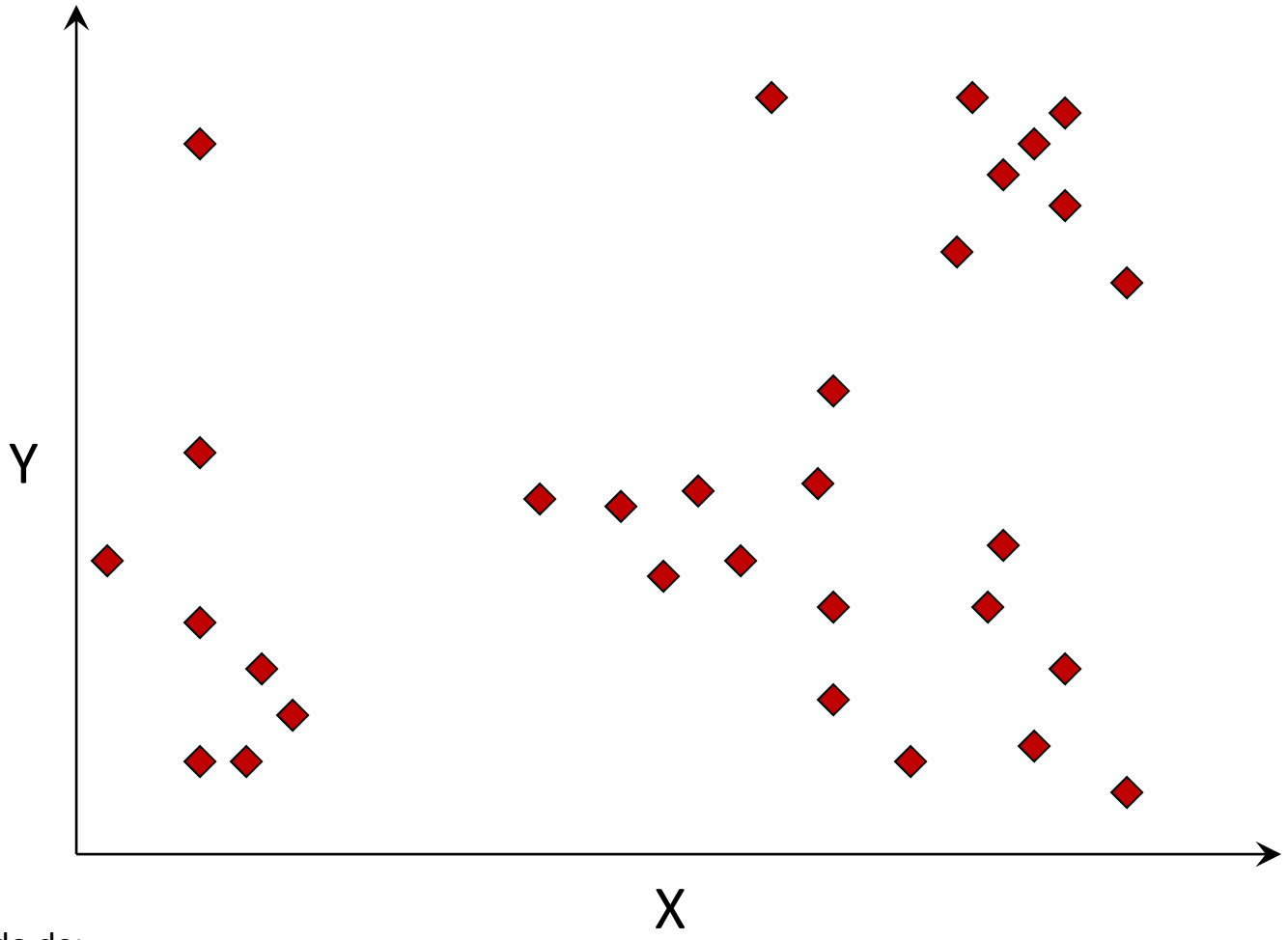
$N_j(t)$: número de objetos asignados a $\alpha_j(t)$ en esta iteración

4. Se **ejecutan los pasos 2 y 3** hasta que las clases **estabilizan** (p.ej. los centroides no se desplazan de su posición anterior o se desplazan menos de un umbral).

$$\sum_{j=1}^{K} \| z_j(t+1) - z_j(t) \| < \in$$

Clustering basado en distancias

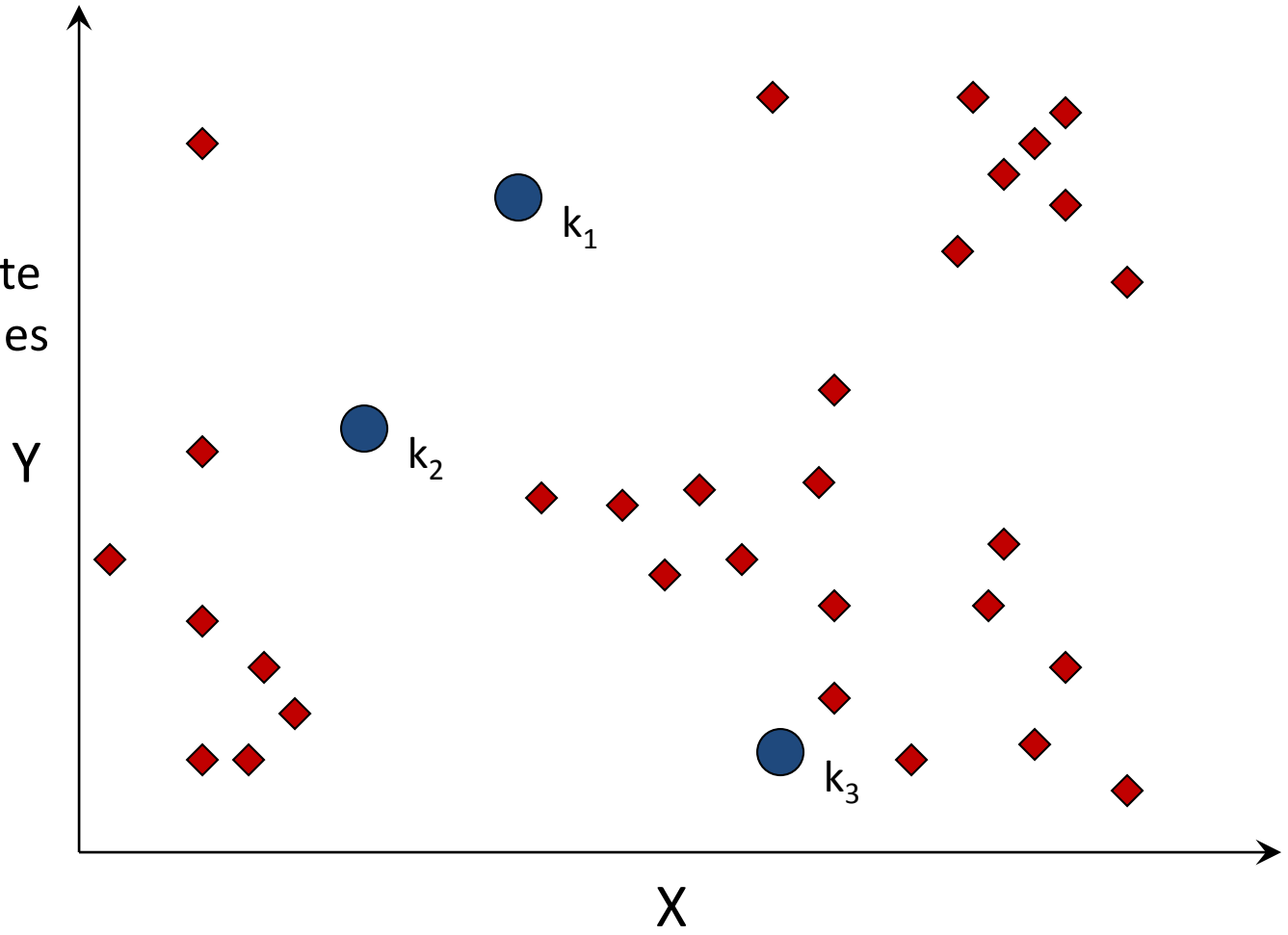
- Ejemplo k-medias (I): queremos clasificar estos datos en 3 clases con el algoritmo k-medias.



Ejemplo modificado de:
http://www.wiphala.net/courses/datamining/KAS_DM/2007-0/class/04_clustering/class_04_clustering.ppt

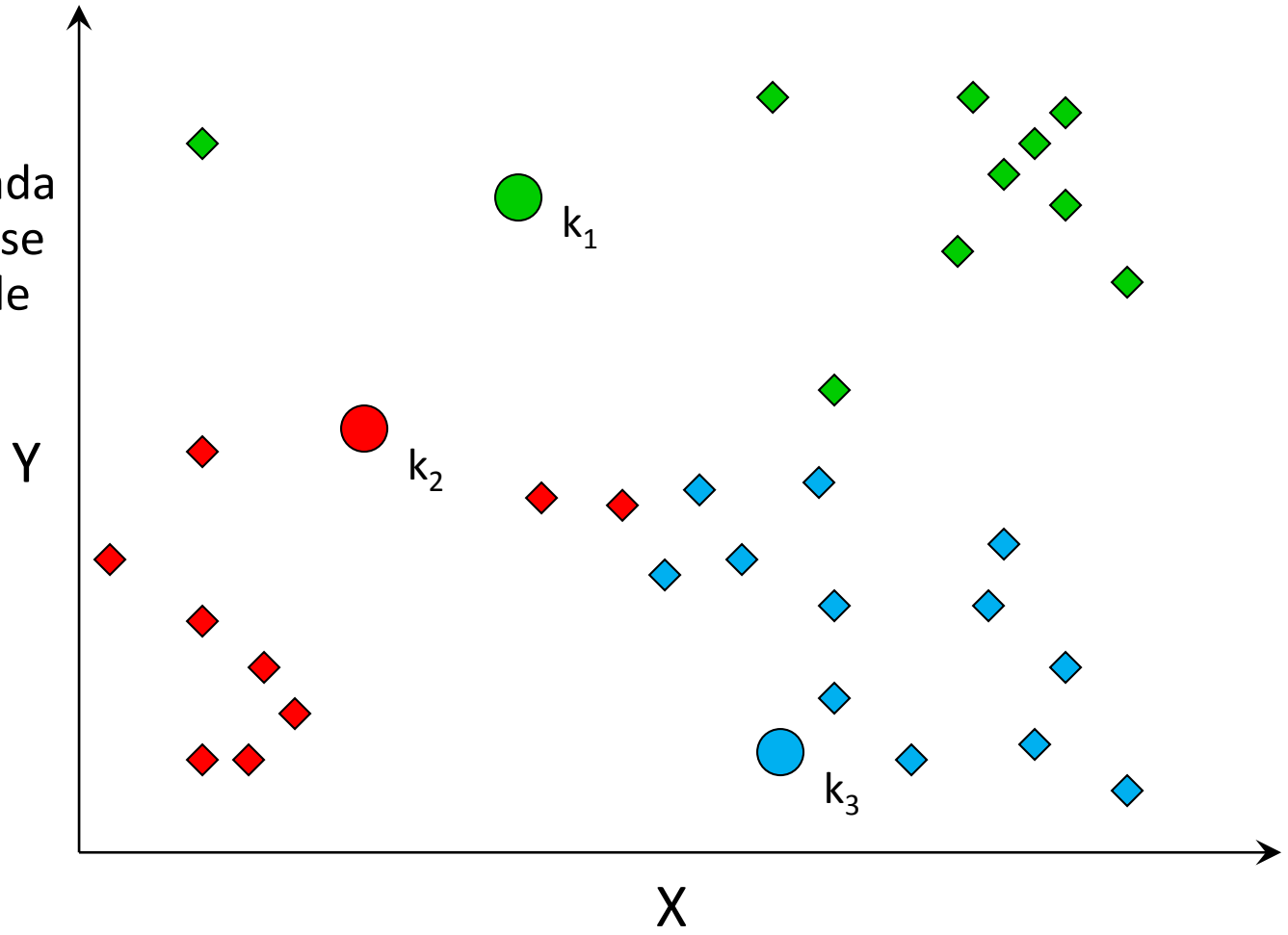
Clustering basado en distancias

- Ejemplo k-medias (II): 1º Elegimos aleatoriamente los 3 centroides iniciales.



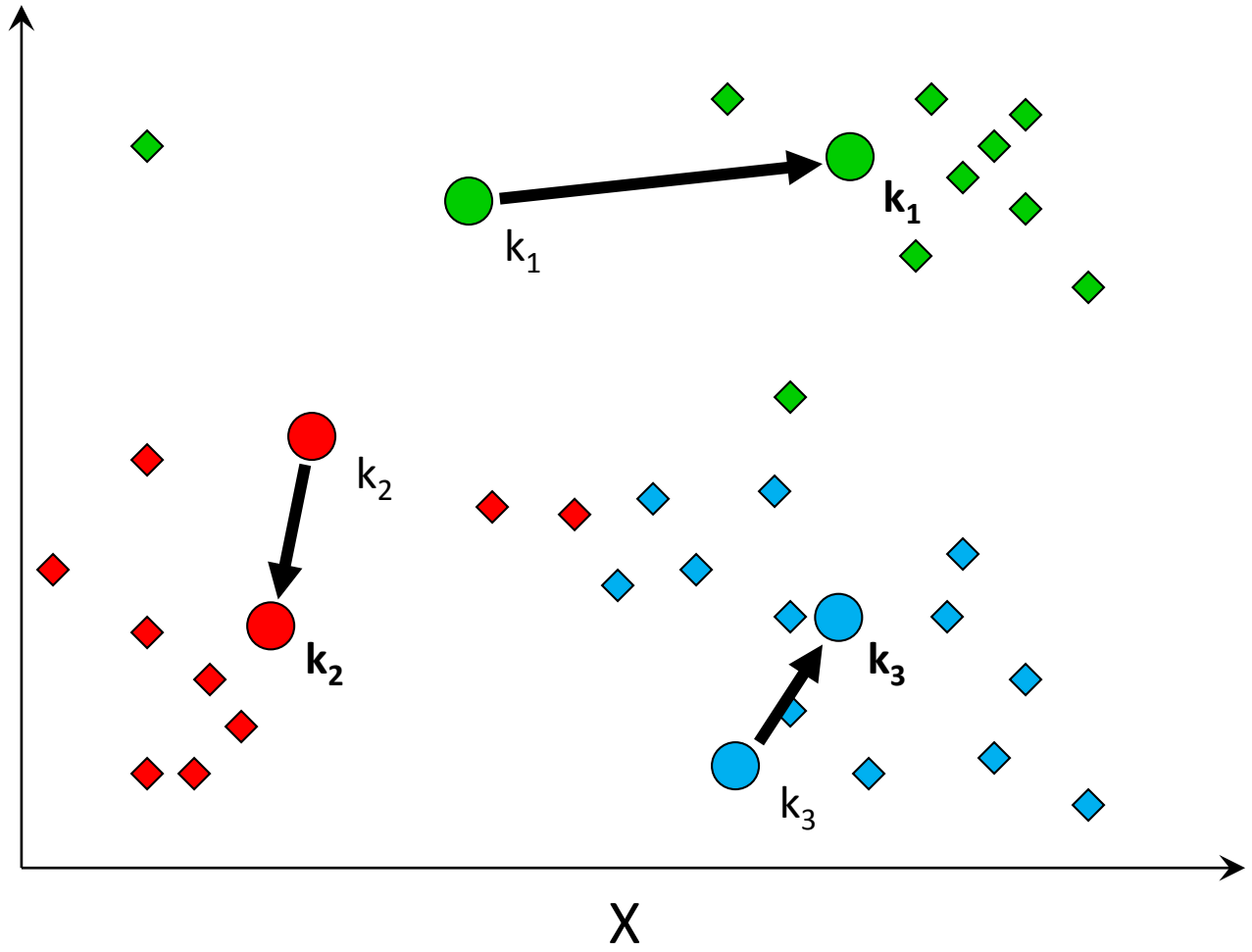
Clustering basado en distancias

- Ejemplo k-medias (III): 2º
  Asignamos cada punto a la clase cuyo centroide esté más cercano.



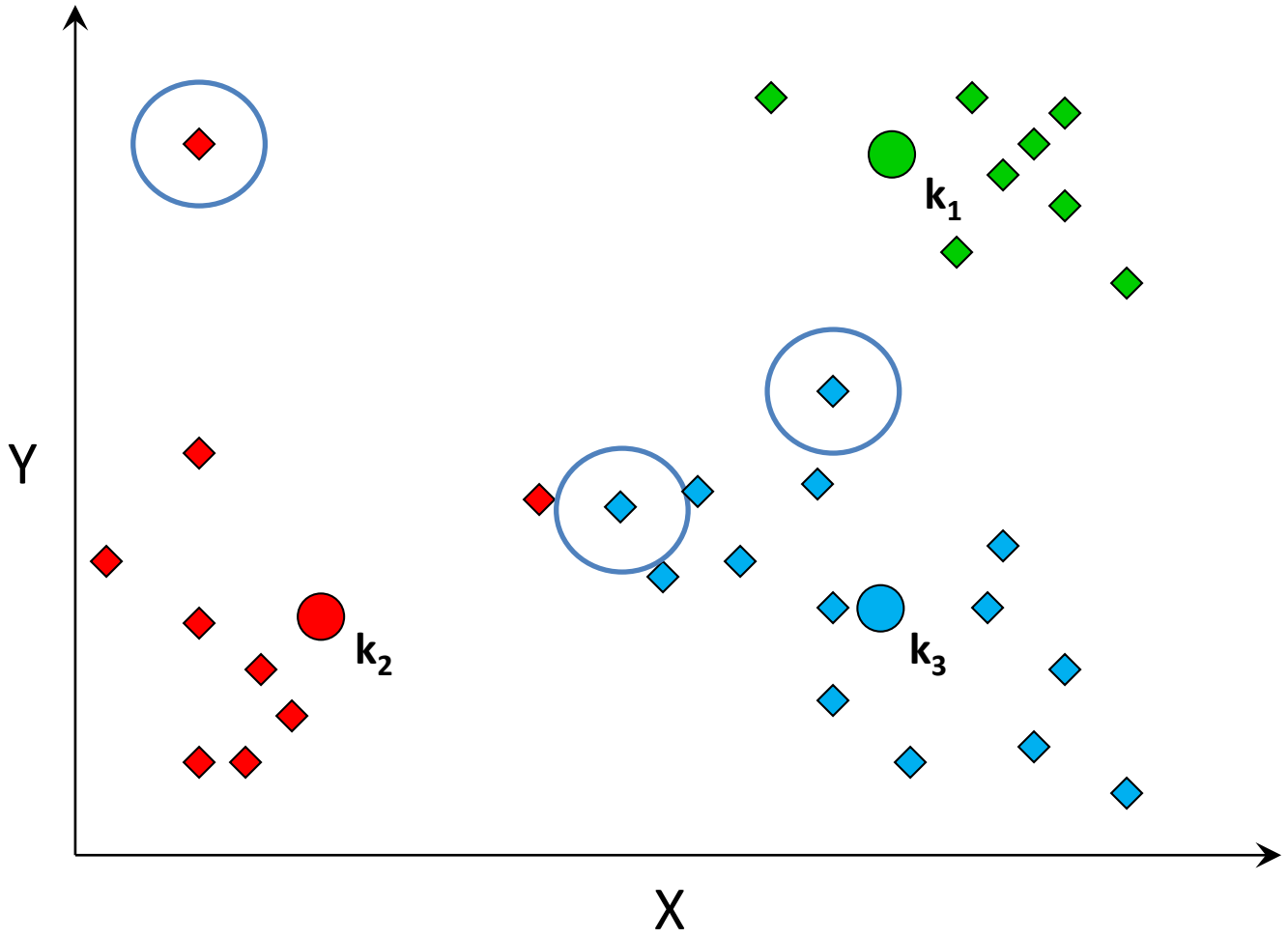
Clustering basado en distancias

- Ejemplo k-medias (IV): 3º Calculamos la nueva posición del centroide: la media de los objetos clasificados en su clúster.



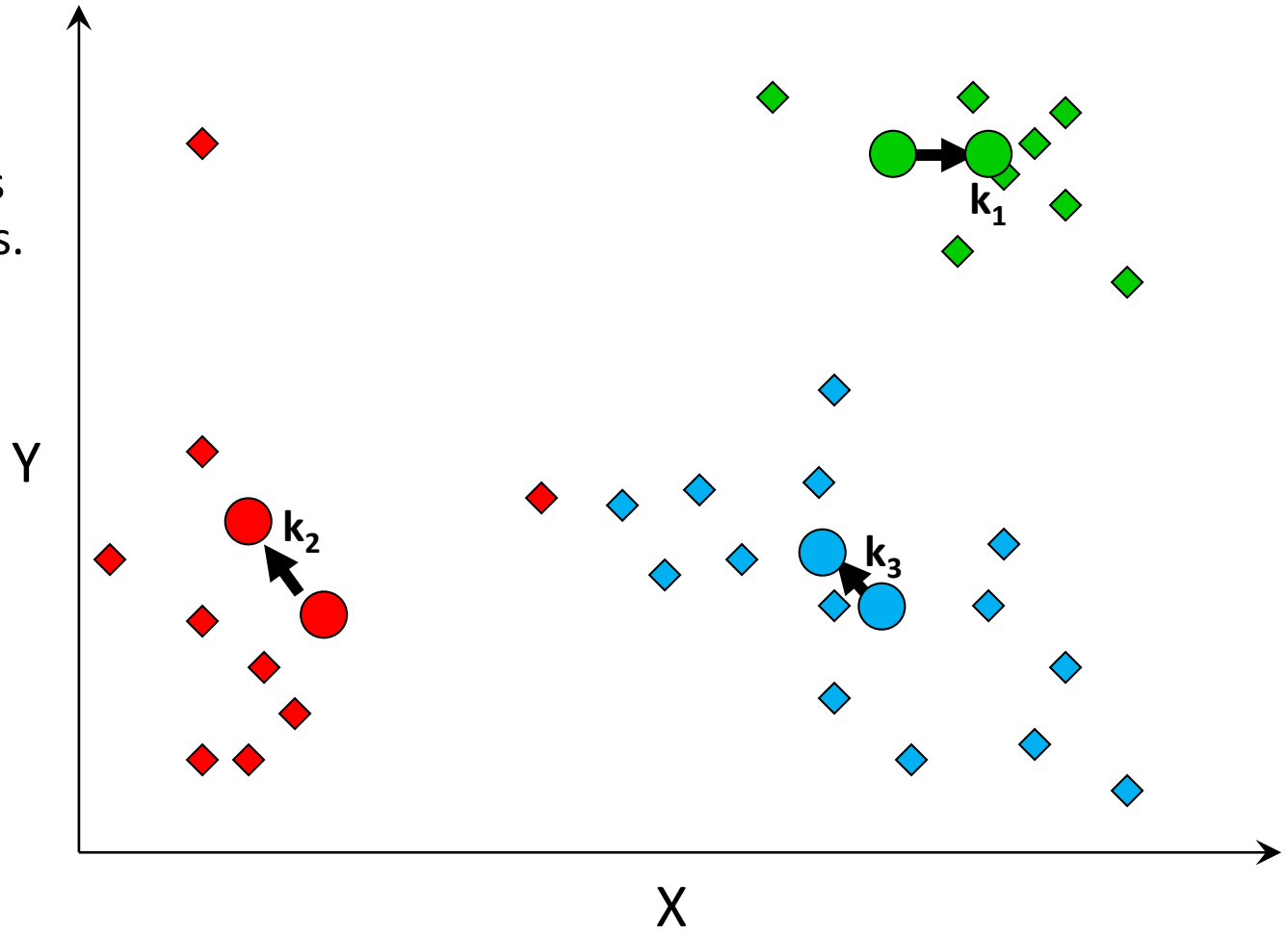
Clustering basado en distancias

- Ejemplo k-medias (V): 4º Reasignamos los puntos a las clases cuyo centroide esté más cerca.



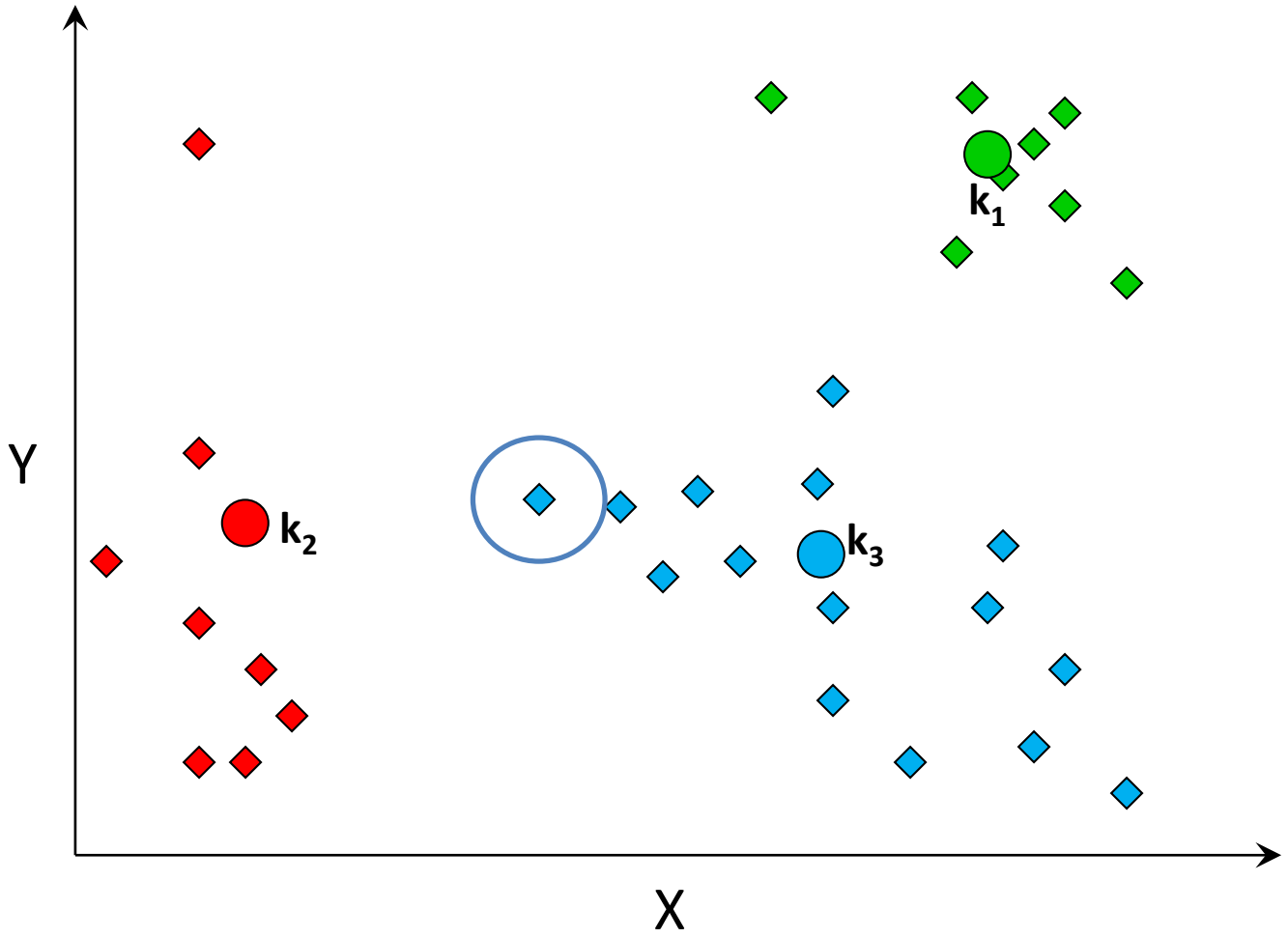
Clustering basado en distancias

- Ejemplo k-medias (VI): 5º Recalculamos los centroides.



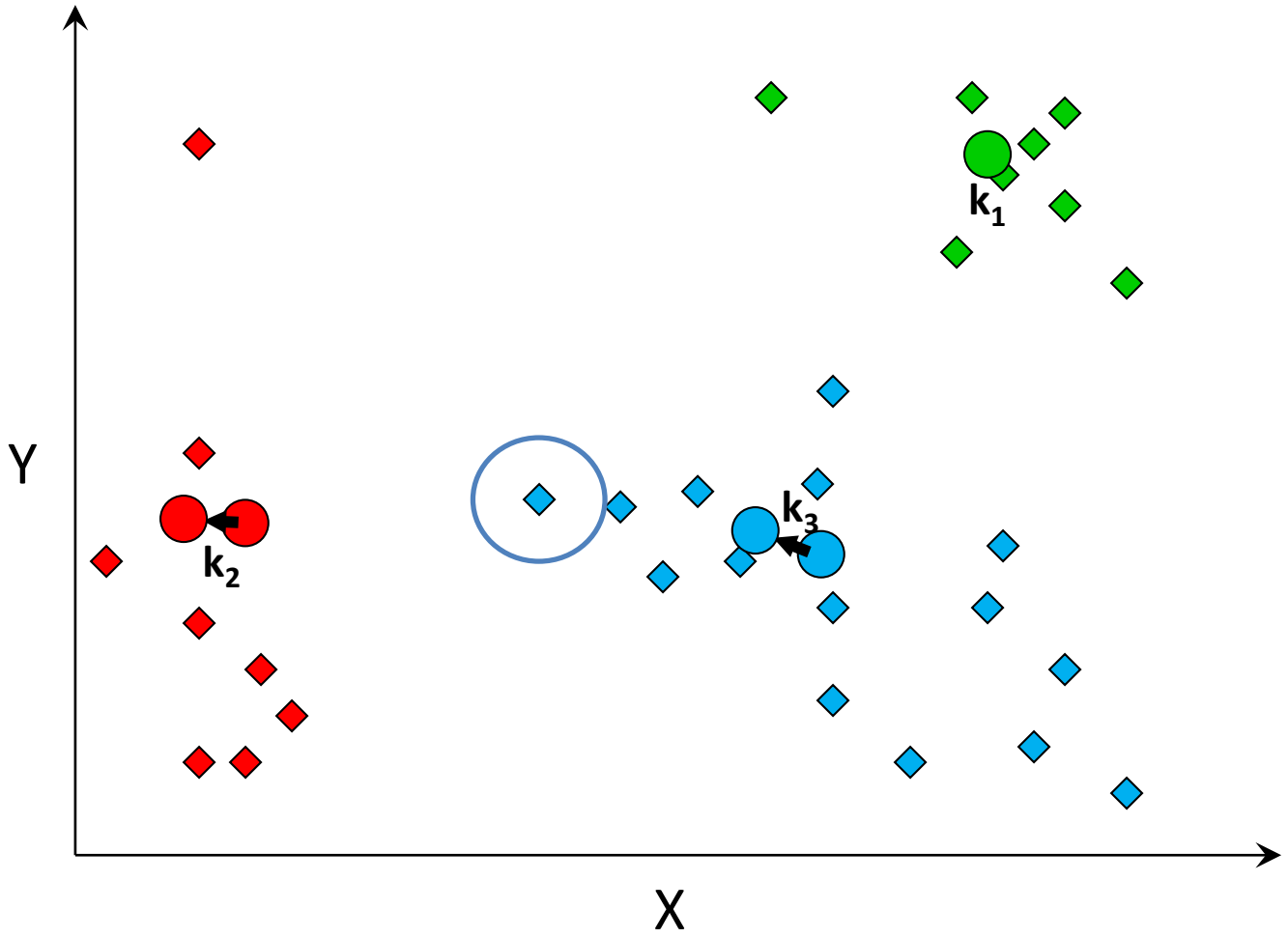
Clustering basado en distancias

- Ejemplo k-medias (VII): 6º Reasignamos puntos.



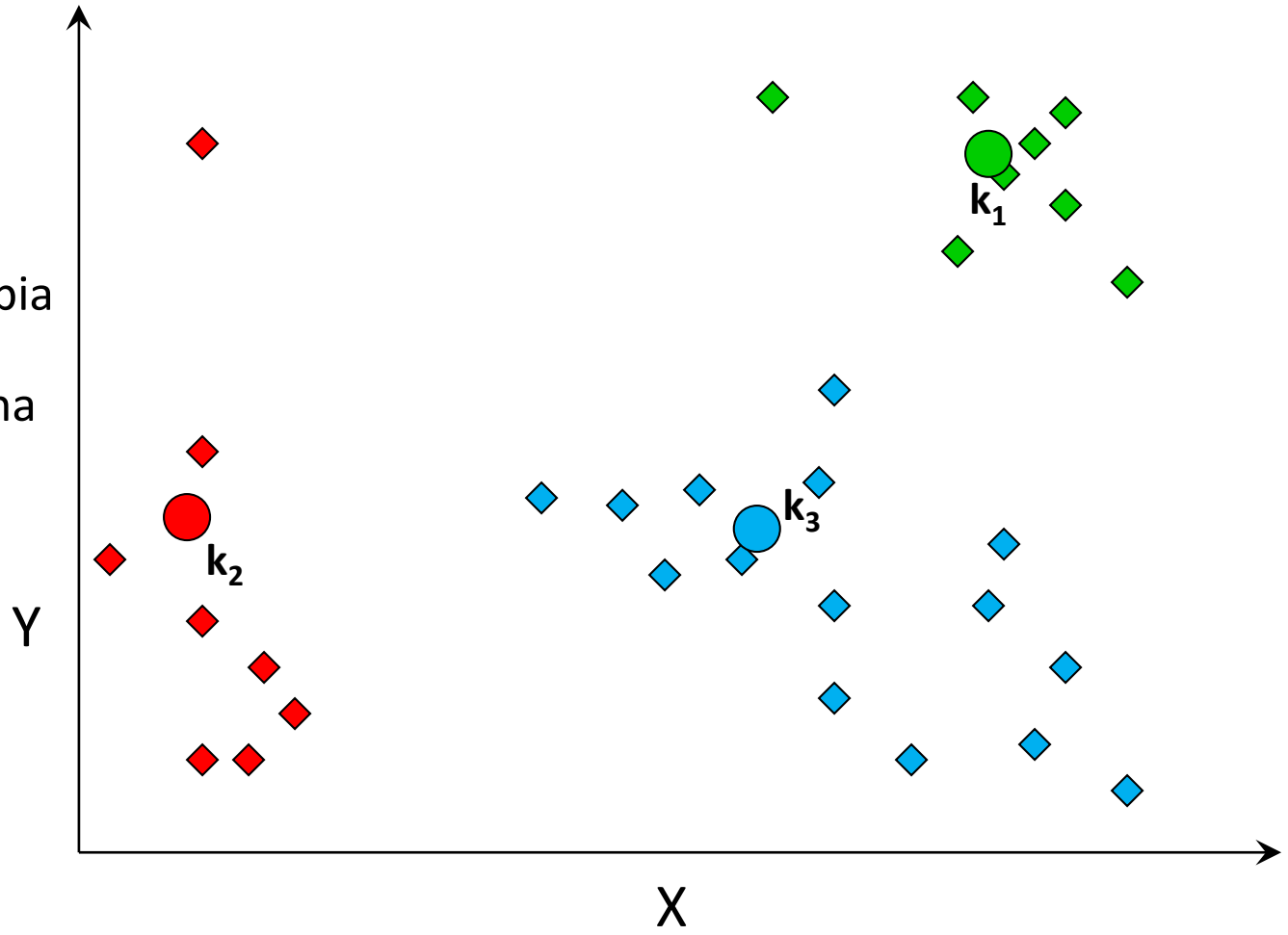
Clustering basado en distancias

- Ejemplo k-medias (VIII): 7º Recalculamos centroides.



Clustering basado en distancias

- Ejemplo k-medias (IX): 8º AI reasignar los puntos, ninguno cambia de clase → la clasificación ha estabilizado.

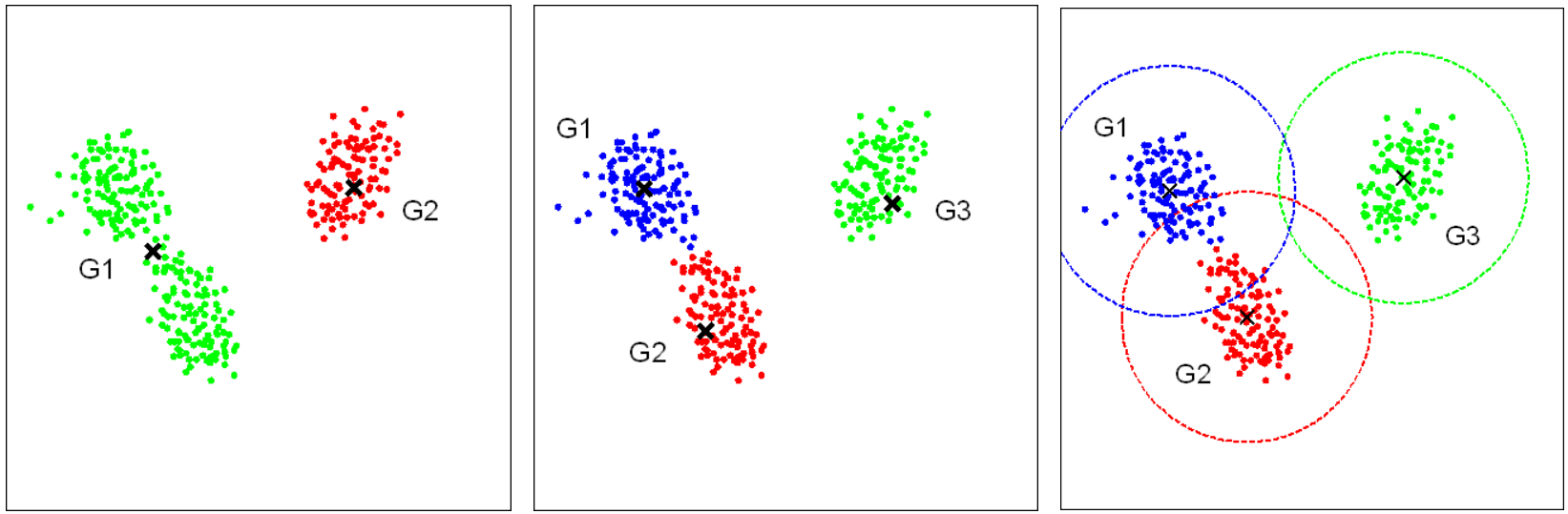


Clustering basado en distancias

- **Ventajas:**
  - **Bajo tiempo de ejecución por iteración.**
  - Algoritmo **sencillo** de programar.
- **Inconvenientes y soluciones disponibles (I):**
  - El **tiempo de cómputo total** depende del **número de iteraciones** hasta la convergencia.
  - Se necesita saber el **número de clases a priori** → Si **no se conoce el número de clases**: **k-medias extendido,** que permite actualizar el número de clases en tiempo de ejecución. El algoritmo compara la distancia de cada objeto a su centroide más cercano (d) con una distancia límite (d_{Max}). Si $d>d_{Max}$ se crea un nuevo grupo y se continua la ejecución del algoritmo → **Problema**: seleccionar d_{Max}.

Clustering basado en distancias

- **Inconvenientes y soluciones disponibles (II):**
  - Ejemplo de k-medias extendido:

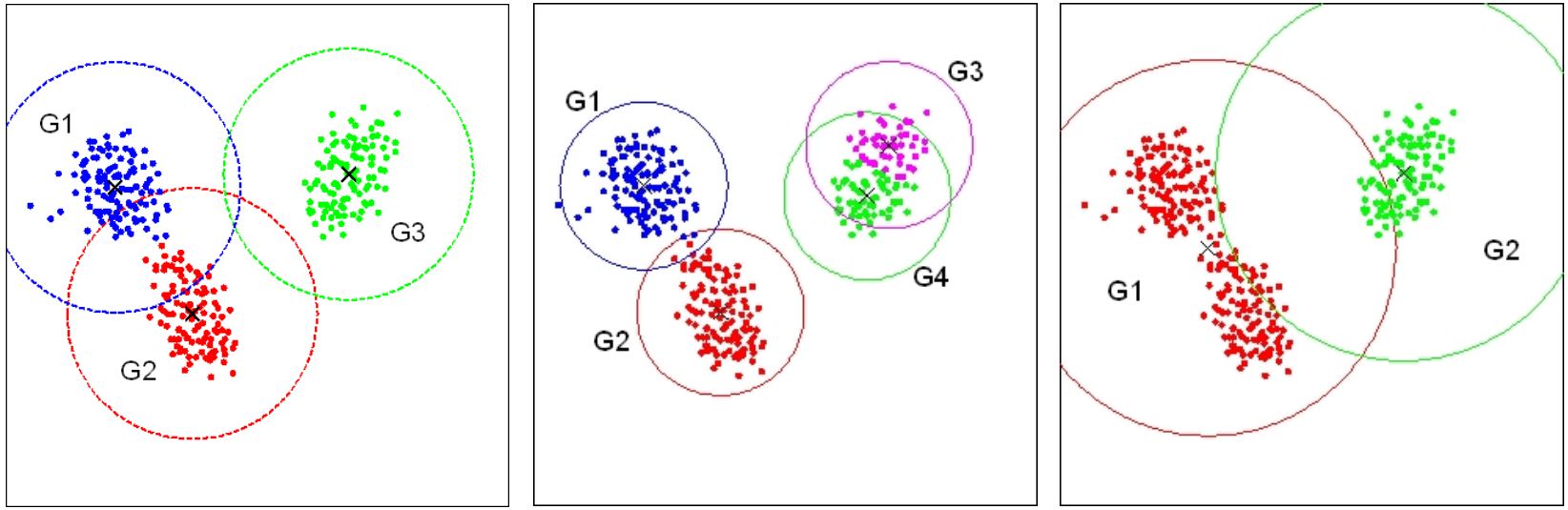


(a) (b) (c)

(a) Resultado del algoritmo k-medias básico con k=2.
(b) Resultado del algoritmo k-medias básico con k=3.
(c) Resultado del algoritmo k-medias extendido. Las circunferencias representan el valor de d_{Max} en el espacio de clasificación 2D.

Clustering basado en distancias

- **Inconvenientes y soluciones disponibles (III):**
  - Ejemplo de k-medias extendido con distintos valores de d_{Max}:



(a) (b) (c)

(a) $d_{Max} = d_{Max0}$ las tres clases se detectan correctamente.

(b) $d_{Max} = d_{Max0}/1.5$ una de las clases se divide.

(c) $d_{Max} = 1.5 \cdot d_{Max0}$ dos de las clases se unen.

Clustering basado en distancias

- **Inconvenientes y soluciones disponibles (IV):**
  - El **agrupamiento final puede depender de los centroides elegidos inicialmente**: la convergencia a un mínimo global no está garantizada, puede estabilizar en un **mínimo local** → Para incrementar la probabilidad de encontrar el **mínimo global** repetimos el proceso varias veces con **distintos centroides iniciales**.
  - Muy **sensible a los *outliers*** (valores atípicos) → Para que los **valores atípicos** influyan menos en el resultado: algoritmo **k-medianas** (*k-medoids*) que utiliza la **mediana** (el valor de la variable de posición central en el conjunto de datos ordenados) en vez de la media para calcular la posición del clúster: no se ve afectado por los valores extremos.

Clustering basado en distancias

- **Inconvenientes y soluciones disponibles (V):**
  - El **aprendizaje es local**: en cada iteración únicamente se mueve el centroide de la clase ganadora → se puede solucionar con el **k-medias difuso**, que considera la probabilidad de cada dato de pertenecer a cada clase.
  - El método de cálculo de clases no es óptimo, ya que considera que todas tienen la **misma dispersión estadística para cada dimensión** de X: fallan cuando los grupos tienen diferentes tamaños y formas y cuando los puntos de un grupo están muy cerca del centroide de otro grupo.

Universidad de Alcalá

5. Clasificadores estadísticos. Teoría estadística de la decisión.

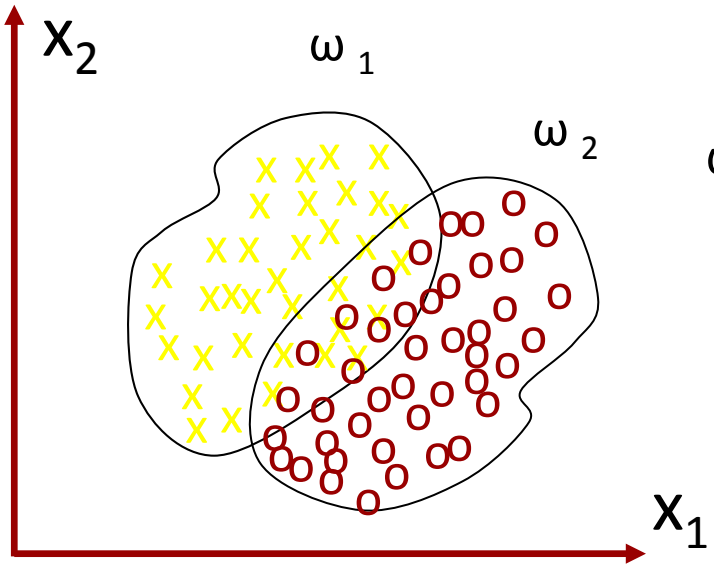
Clasificadores estadísticos

- Los **clasificadores estadísticos** se emplean en casos en los que las clases tienen una **gran dispersión** respecto a la media y además hay **solapamiento** entre clases.
- Considera que el problema puede ser definido en **términos estadísticos** y necesita que todos los **parámetros estadísticos relevantes estén definidos**.
- Ejemplo de clasificación de tuercas y tornillos:

Características, p. ej.:

x_1: circularidad

x_2: área



ω_1: tuercas

ω_2: tornillos

Clasificadores estadísticos

- Algunas definiciones (I):
  - **Población**: Conjunto de todos los elementos bajo estudio.
  - **Muestra**: Subconjunto finito de elementos de una población.
  - **Variable aleatoria:** Cualquier propiedad o característica X que se pueda obtener de cada objeto de una población.
  - **Variable aleatoria n-dimensional**: Conjunto de n propiedades extraídas de cualquier objeto de una población.
  - **Probabilidad** de que X tome un cierto valor x:

$$P(X=x) = p(x)$$

$$0 \leq p(x) \leq 1$$

  - **Función de distribución de probabilidad** de una variable aleatoria X:

$$F(x) = Prob(X \leq x)$$

$$F(-\infty) = 0; \quad x_1 < x_2 \rightarrow F(x_1) \leq F(x_2); \quad F(+\infty) = 1$$

  - **Función densidad de probabilidad** de una variable aleatoria X (definida si F es continua y derivable): $p(x) = dF(x)/dx$

$$F(x) = \int_{-\infty}^{x} p(y)dy \qquad \int_{-\infty}^{+\infty} p(x)dx = 1$$

Clasificadores estadísticos

- Algunas definiciones (II):
  - **Probabilidad incondicional** de una variable aleatoria:

$$P(X=x) = p(x) \qquad \int_{-\infty}^{+\infty} p(x)dx = 1$$

  - **Probabilidad conjunta** de dos variables aleatorias X, Y**:**

$$P(X = x;\ Y = y) = p(x, y) \qquad \int_x \int_y p(x, y)dxdy = 1$$

  - Dos sucesos son **independientes** si $p(x,y) = p(x)\cdot p(y)$
  - **Probabilidad marginal:** probabilidad de una variable obtenida a partir de cierta información sobre las demás variables**.** Las probabilidades incondicionales p(x) y p(y) son las probabilidades marginales de la probabilidad conjunta p(x,y)

$$p(x) = \int_{-\infty}^{+\infty} p(x, y)dy \qquad p(y) = \int_{-\infty}^{+\infty} p(x, y)dx$$

  - **Probabilidad condicionada** o probabilidad de X dado Y, es la probabilidad de que ocurra un evento X, sabiendo que también sucede otro evento Y.

$$P(X = x|Y = y) = p(x|y)$$

  - Dos sucesos son **independientes** si: $p(x|\ y) = p(x)$ y $p(y|x) = p(y)$

Clasificadores estadísticos

- Algunas definiciones (III):
  - **Teorema de Bayes:** relaciona la probabilidad conjunta, y las probabilidades condicionales e incondicionales:

$$p(x,y) = p(x)\,p(y|x) = p(y)\,p(x|y)$$

$$p(x|y) = \frac{p(x)\,p(y|x)}{p(y)} \qquad p(y|x) = \frac{p(y)\,p(x|y)}{p(x)}$$

  - **Probabilidad a priori** de una clase c: P(c). Es la probabilidad de que un objeto arbitrario pertenezca a c. Puede estimarse como la proporción de objetos de c respecto al total, y se supone que se conoce de antemano.

$$0 \le P(c) \le 1, \quad 1 \le c \le C; \qquad \sum_{c=1}^{C} P(c) = 1$$

  - **Función de densidad de probabilidad condicional de una clase c**, $p(x|c)$, $p_c(x)$.
  - **Densidad incondicional de x:** $p(x)$

Clasificadores estadísticos

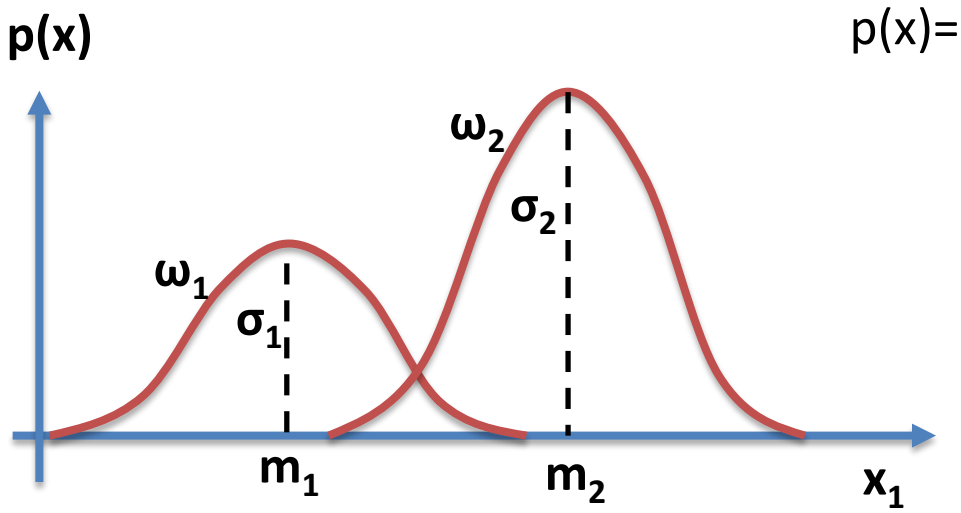
- Algunas definiciones (IV):
  - La **probabilidad a posteriori** de la clase c (hipótesis) si se ha observado x (evidencia), p(c|x), se puede calcular utilizando el teorema de Bayes:

$$p(c|x)=\frac{P(c)\,p(x|c)}{p(x)} \quad \text{donde} \quad p(x)=\sum_{c'=1}^{C} p(x|c')P(c')$$

  - Nuestro problema de **clasificación** consiste en saber, para un valor x de la característica del objeto, a qué clase pertenece:
    - **Calcularemos la probabilidad a posteriori de cada clase c si se ha observado x**, aplicando el **teorema de Bayes** a partir de:
      - La probabilidad a priori de cada clase, y
      - Las funciones de densidad de probabilidad condicionales de cada clase.
    - La probabilidad de la característica (evidencia), p(x): probabilidad de que se presente un elemento a clasificar con un vector de características x. Opera como un factor de escala ya que aparece en todas las clases→ se puede eliminar de la decisión.
    - Asignaremos el objeto a la **clase con mayor probabilidad a posteriori**.

Clasificadores estadísticos

- Suponemos que los valores de las características de los objetos presentes en una imagen siguen una **distribución normal o de Gauss**. Cada curva viene caracterizada por su **media** (m_i) y por su **desviación típica** (σ_i).
- Los valores de las características totales de la imagen se corresponderán con la suma de las distribuciones de todos los objetos presentes en ella ponderada por el número de veces que aparece cada uno.
- Ej: suponemos que hay 2 objetos, 1 y 2 y tenemos la siguiente distribución de valores para la característica x_1:



$$p(x)=P_1p_1(x)+P_2p_2(x)=d_1(x)+d_2(x)$$

$$p_1(z)=\frac{1}{\sigma_1\sqrt{2\pi}}\exp\left[-\frac{(z-m_1)^2}{2\sigma_1^2}\right]$$

$$p_2(z)=\frac{1}{\sigma_2\sqrt{2\pi}}\exp\left[-\frac{(z-m_2)^2}{2\sigma_2^2}\right]$$

$$P_1+P_2=1$$

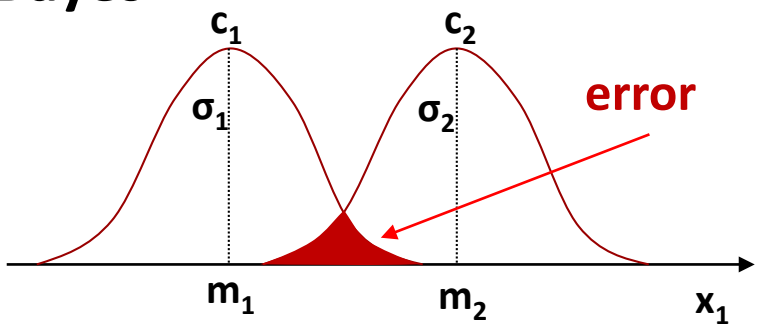
Clasificadores estadísticos

Clasificación de mínimo riesgo o de Bayes

- Probabilidad de error:

$$p(\text{error}/X)=\begin{cases} p(c_1/X) & \text{si se clasifica como } c_2 \\ p(c_2/X) & \text{si se clasifica como } c_1 \end{cases}$$

$$P_C(\text{error}|X)=1-P(c|X)$$



- Probabilidad mínima de error para un punto dado X:

$$\text{mín}(P(\text{error}|X))=\min_{1\le c\le C}(P_c(\text{error}|X))=1-\text{máx}(P(c|X))$$

Es decir, la probabilidad de error se minimiza si asignamos X a la clase con mayor probabilidad a posteriori.

- El riesgo global de error de un clasificador es:

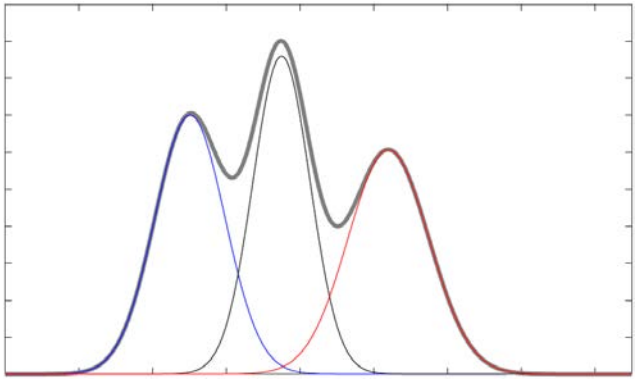
$$p(error)=\int_{-\infty}^{\infty} p(error,X)\,dX=\int_{-\infty}^{\infty} p(error/X)\,p(X)\,dX$$

- **Clasificador de mínimo riesgo de error o de Bayes, será el que minimice p(error), asignando X a la clase de mayor probabilidad a posteriori.**

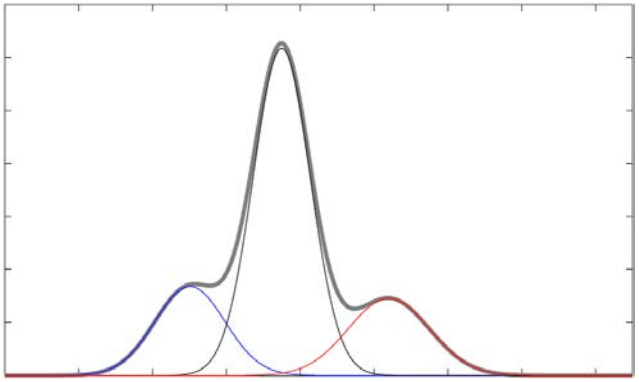
Clasificadores estadísticos

Ejemplos: probabilidades a priori y densidad incondicional (y conjunta de todas las clases)

(P1 = P2 = P3 = 1/3)

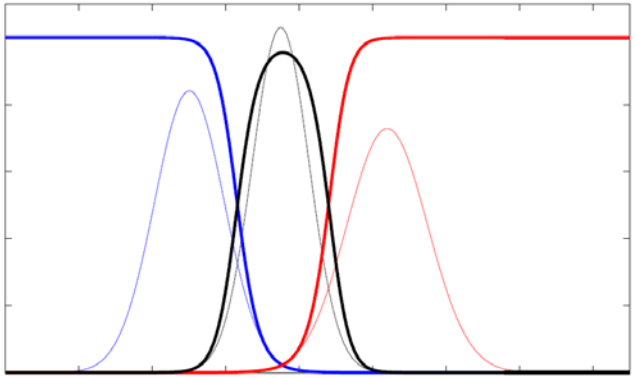


(P1=0.2, P2=0.6, P3=0.2)



Densidades por clase y probabilidad a posteriori

(P1 = P2 = P3 = 1/3)



Probabilidad de error

(P1=0.2, P2=0.6, P3=0.2)

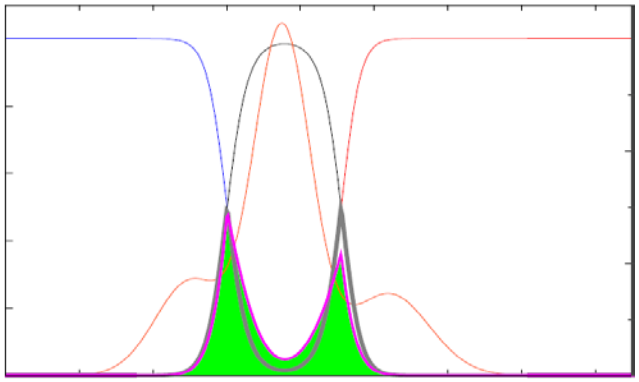
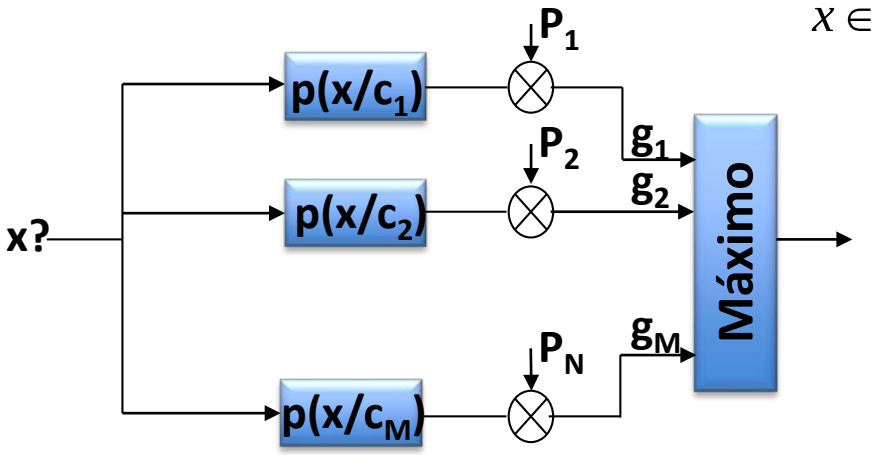


Gráfico de http://web.iti.upv.es/~evidal/students/app/tema5/t3app2p.pdf

Clasificadores estadísticos

- Problema de clasificación: dado un UT con M clases UT=(c_1, c_2, …c_M) y un vector de características x=(x_1, x_2, …,x_N)



$$x \in c_j \quad si \quad P_j \cdot p(x/c_j) > P_i \cdot p(x/c_i)$$
$$\forall i \neq j, \quad i = 1, 2, ..., N$$

x Є c_j | $p(x/c_j)$ P_j máximo

- El umbral óptimo se selecciona para aquel nivel z =T que cumple:

$$d_1(z) = d_2(z) \rightarrow P_1 * p_1(z) = P_2 * p_2(z)$$

- Si $\sigma_1=\sigma_2=\sigma$ (histograma bimodal modelado con funciones gaussianas de la misma varianza):

$$T = \frac{m_1 + m_2}{2} + \frac{\sigma^2}{m_1 - m_2} \ln \frac{P_2}{P_1}$$

- Estas ecuaciones (criterio de clasificación y cálculo del umbral) pueden generalizarse para el caso de M clases y N dimensiones.

Clasificadores estadísticos

- Un **clasificador de Bayes o de mínimo error** puede utilizar cualquiera de las siguientes **funciones discriminantes equivalentes** con los mismos resultados, pero costes de computación diferentes:

$$g_i(x) = \frac{p(x / c_i)p(c_i)}{p(x)}$$ Probabilidad a posteriori.

$$g_i(x) = p(x / c_i)p(c_i)$$ Eliminamos $p(x)$ porque no ayuda a discriminar ya que aparece en todos los términos.

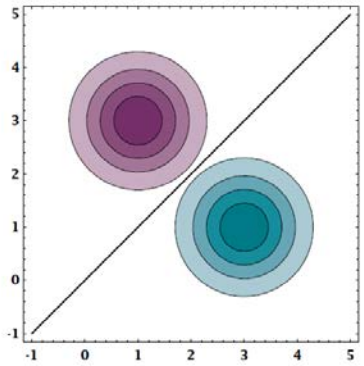
$$g_i(x) = \ln(p(x / c_i)) + \ln(p(c_i))$$ Hacemos el logaritmo y cambiamos la multiplicación por una suma. Es el que se suele utilizar en la práctica.

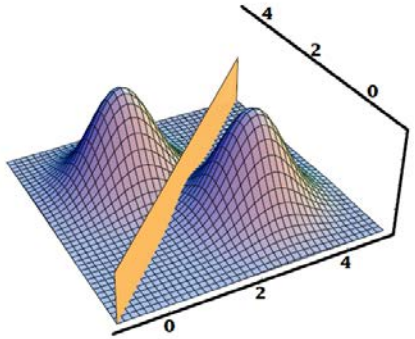
- El teorema de Bayes es **óptimo** si $p(x/c_i)$ y $p(c_i)$ son conocidas. Se estiman mediante **datos de entrenamiento:**
  - A veces es complejo (insuficiente número de muestras, alta dimensión de x)
  - Solución: forma paramétrica (Gaussianas) → estimación de los parámetros de la gaussiana (media y desviación típica).

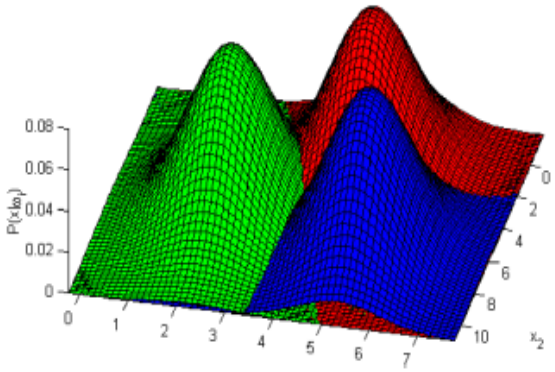
Clasificadores estadísticos

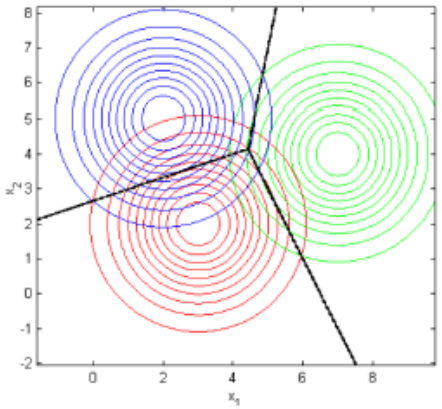
- **Funciones discriminantes** para funciones de densidad Gaussianas definen las **fronteras de decisión**: $g_i(X)=g_j(X)$.
- En función de estas se obtienen las distintas regiones en el **espacio de clasificación**.

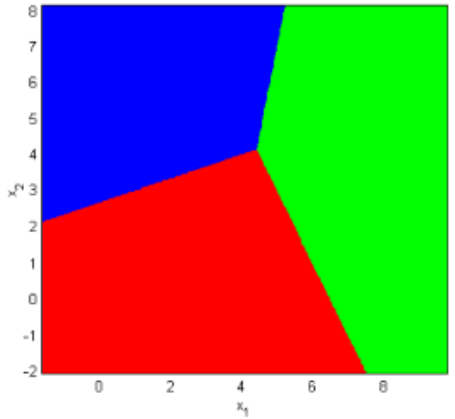
Por ejemplo, si las matrices de características son **incorreladas** con la misma varianza ($\Sigma_i=\sigma^2 I$ matriz diagonal), y **probabilidades a priori iguales** → las **fronteras de decisión** son **hiperplanos**.







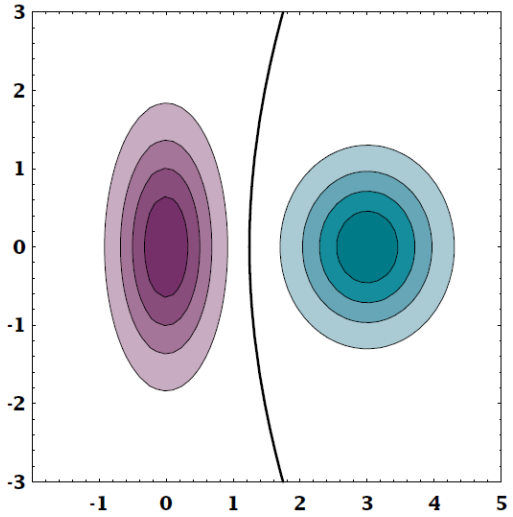


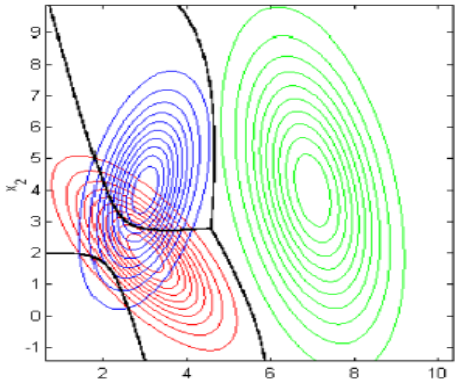


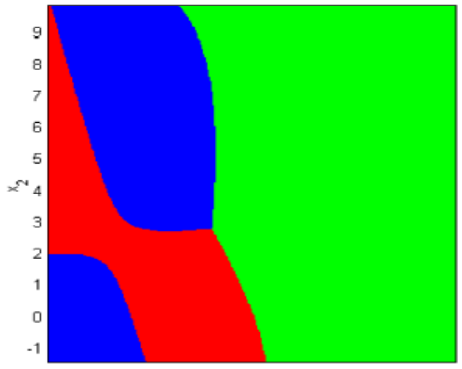
http://arantxa.ii.uam.es/~msanchez/docencia/maaa/transparencias/Bayes_1112.pdf http://mit.ocw.universia.net/6.345/NR/rdonlyres/Electrical-Engineering-and-Computer-Science/6-345Automatic-Speech-RecognitionSpring2003/C919167E-496F-4F65-9D99-CE5B97406A1C/0/lecture7.pdf

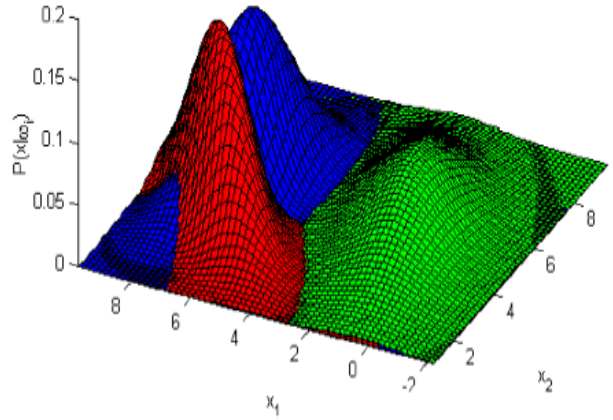
Clasificadores estadísticos

- Si suponemos un **caso general**, en el que las distribuciones tienen **matrices de covarianza (Σ_i) arbitrarias**. Para 2D:
  - Las **clases** son **elipses** de distinto tamaño y forma.
  - Las **fronteras de decisión** son **cuadráticas** (circunferencias, elipses, parábolas, hipérbolas, pares de planos).









http://arantxa.ii.uam.es/~msanchez/docencia/maaa/transparencias/Bayes_1112.pdf http://mit.ocw.universia.net/6.345/NR/rdonlyres/Electrical-Engineering-and-Computer-Science/6-345Automatic-Speech-RecognitionSpring2003/C919167E-496F-4F65-9D99-CE5B97406A1C/0/lecture7.pdf

Universidad de Alcalá

6. Máquinas de vectores de soporte (*Support vector machines*-SVMs)

SVMs

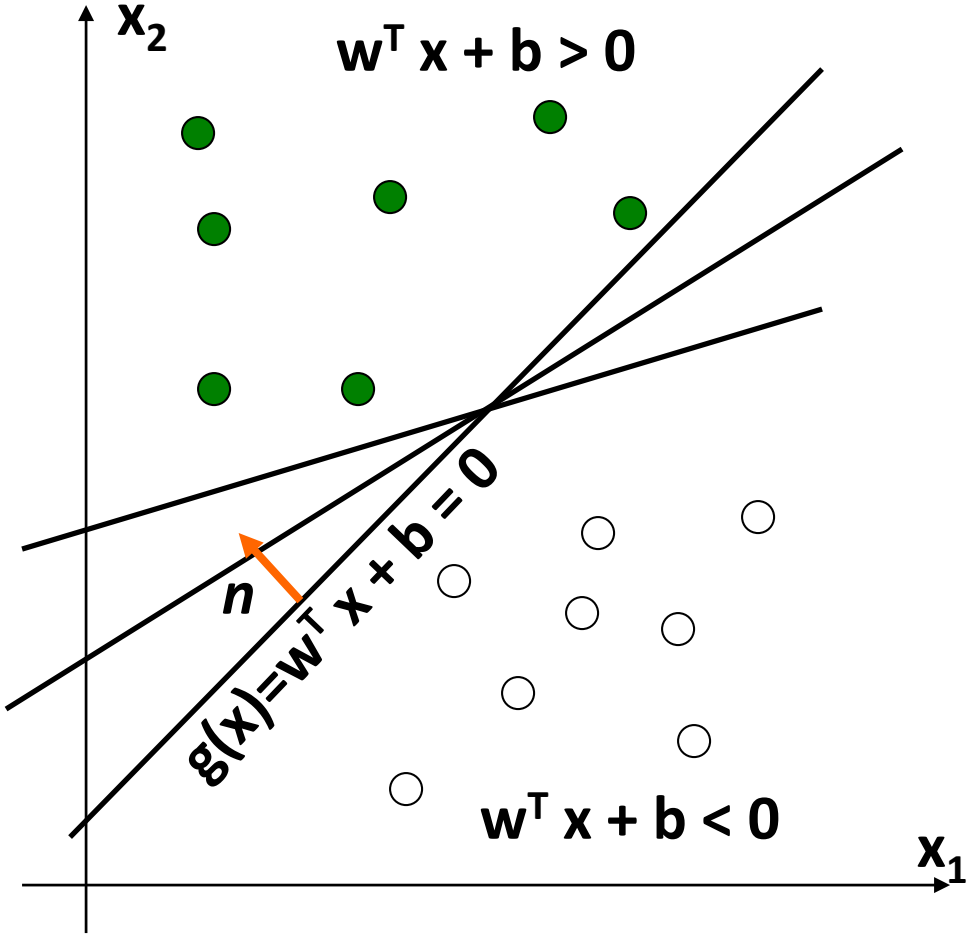
- Clasificadores lineales:
  - Se basan en una función lineal que define un hiperplano:

$$g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$$

"n" es el vector unitario normal al hiperplano:

$$\mathbf{n} = \frac{\mathbf{w}}{\|\mathbf{w}\|}$$

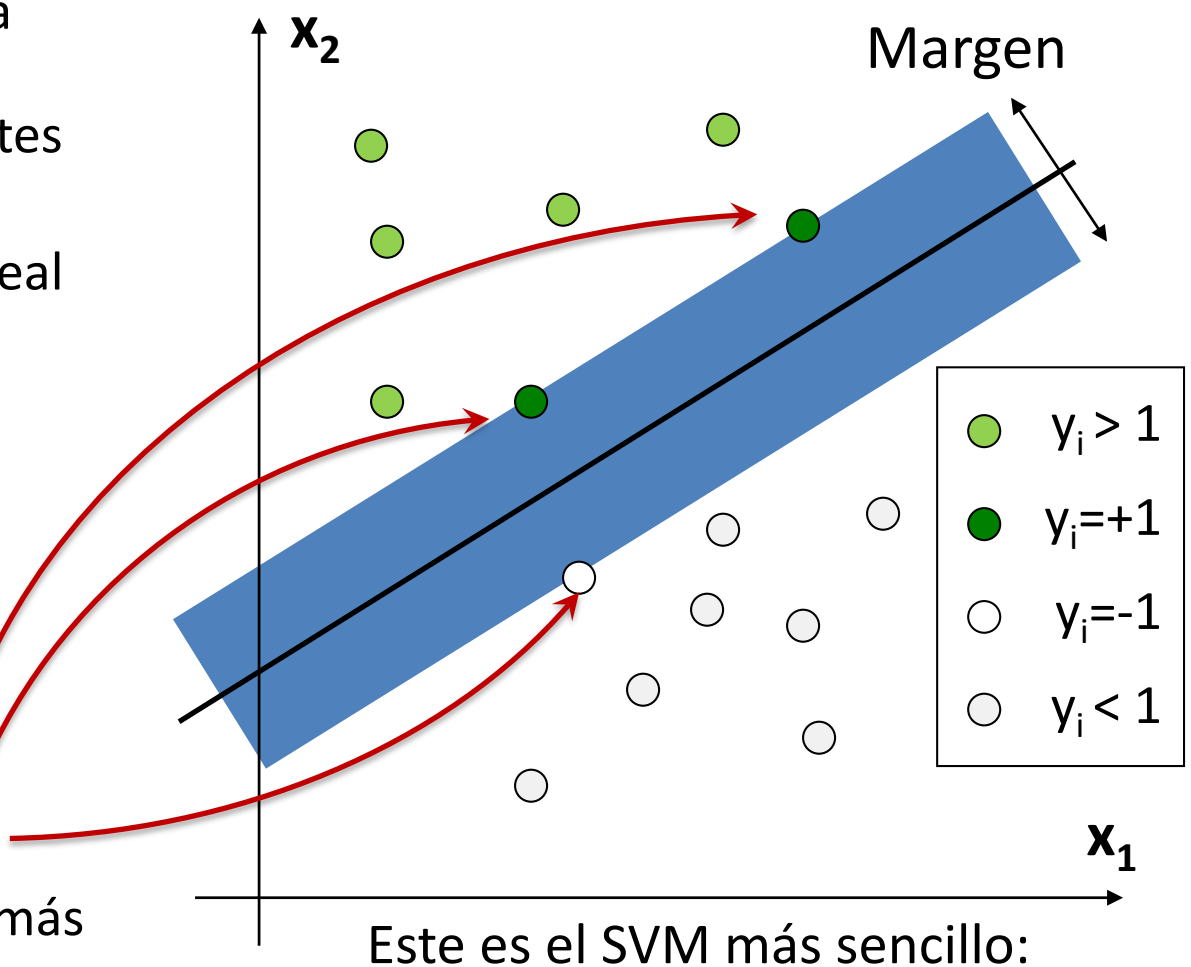
  - Hay infinitas soluciones... ¿Cuál es la mejor?



http://www.ece.uprm.edu/~manian/svm_tutorial.ppt

SVMs

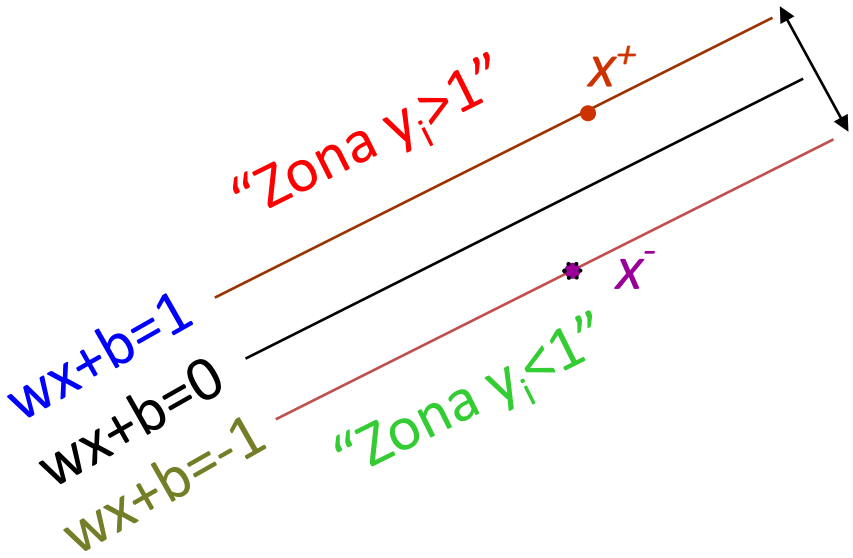
- El **margen** es la anchura que podría tener la frontera de decisión antes de tocar ningún dato.
- El mejor clasificador lineal será el que tenga el **máximo margen**.
- Es el más **robusto** a *outliers* y el que tiene mayor capacidad de **generalización**.
- **Vectores de soporte:** Son los únicos importantes en el entrenamiento. Los demás se pueden ignorar.



Este es el SVM más sencillo: **SVM lineal (LSVM)**.

http://www.ece.uprm.edu/~manian/svm_tutorial.ppt

SVMs



Anchura del margen $M = \frac{2}{\|w\|}$

Escalamos las rectas para que :

cuando $y_i = +1, \quad \mathbf{w}^T\mathbf{x}_i + b \geq 1$

cuando $y_i = -1, \quad \mathbf{w}^T\mathbf{x}_i + b \leq -1$

Sabemos que en los vectores de soporte:

- $\mathbf{w} \cdot \mathbf{x}^+ + b = +1$
- $\mathbf{w} \cdot \mathbf{x}^- + b = -1$
- $\mathbf{w} \cdot (\mathbf{x}^+ - \mathbf{x}^-) = 2$

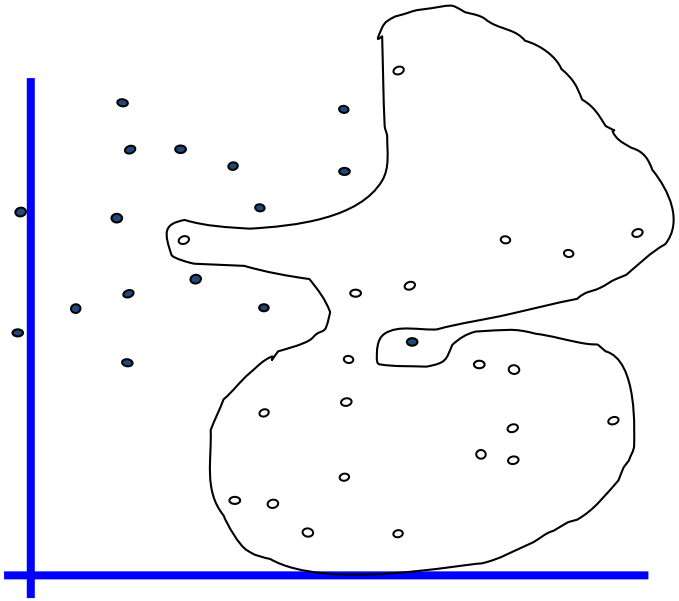
$$M = \frac{(x^+ - x^-)\cdot w}{\|w\|} = \frac{2}{\|w\|}$$

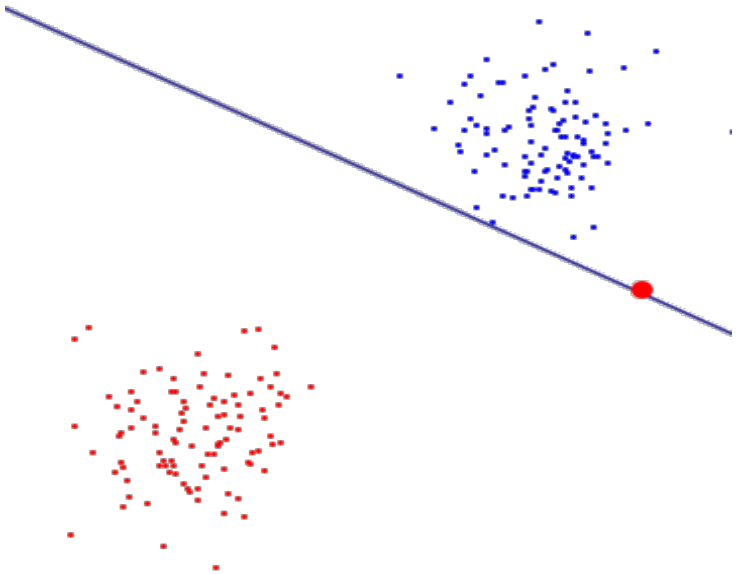
$$\begin{cases} \text{maximizar} & \frac{2}{\|\mathbf{w}\|} \\ \text{minimizar} & \frac{1}{2}\|\mathbf{w}\|^2 \end{cases}$$

http://www.ece.uprm.edu/~manian/svm_tutorial.ppt

SVMs

- El **entrenamiento con margen duro** (*hard margin*) requiere que todos los puntos sean clasificados correctamente: no se admiten errores en el entrenamiento.
- **Problema:** ¿qué pasa cuando la base de datos de entrenamiento tiene errores?: un solo pixel erróneo puede determinar la frontera de decisión, el margen. **Sobreajuste (*overfitting*)**





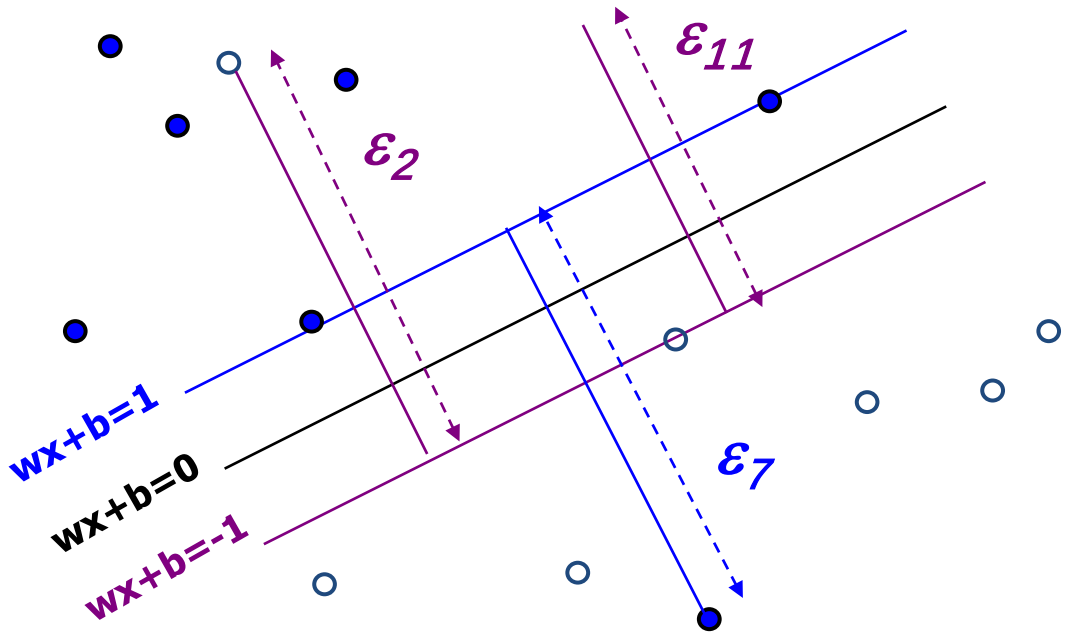
http://www.ece.uprm.edu/~manian/svm_tutorial.ppt

SVMs

- En el **entrenamiento con margen suave** *(soft margin)* incluimos variables (ξ_i *slack variables*) que permiten la clasificación errónea de ejemplos difíciles o ruidosos.
- ¿En qué consitirá la **optimización**?

Minimización de :

$$\frac{1}{2}\mathbf{w}.\mathbf{w} + C\sum_{k=1}^{R}\varepsilon_k$$

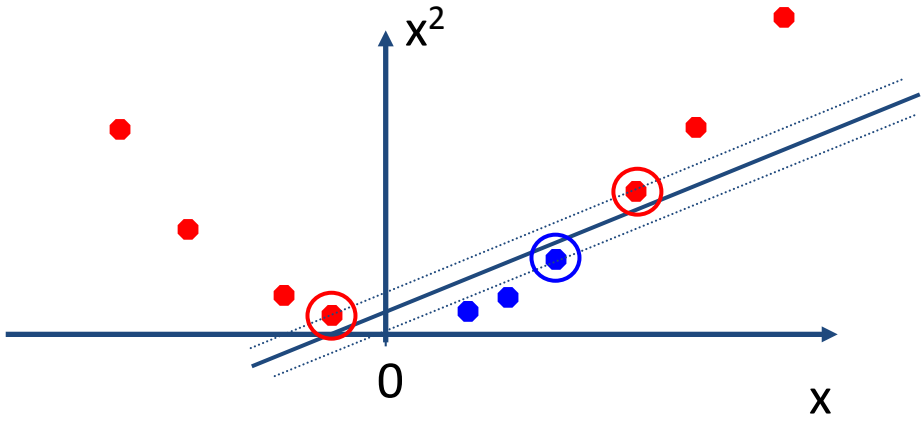


http://www.ece.uprm.edu/~manian/svm_tutorial.ppt

SVMs

- **SVMs no lineales**:
  - Algunos conjuntos de datos son **linealmente separables**, como mucho con alguna dificultad por el ruido, pero hay otros conjuntos que **no** lo son, como, p. ej.:

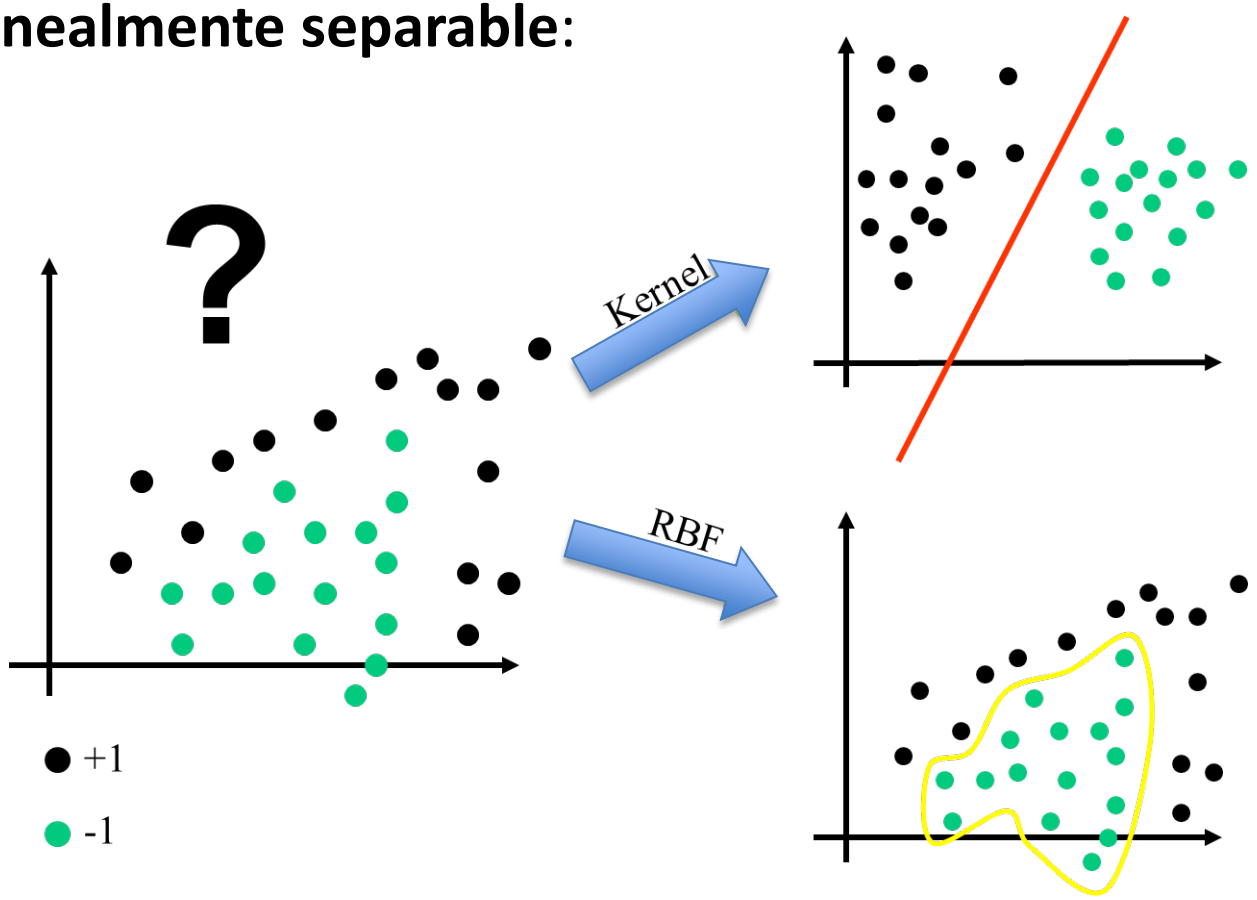
  - Los **mapeamos en un espacio de dimensión superior en el que sí sean linealmente separables**.



http://www.ece.uprm.edu/~manian/svm_tutorial.ppt

SVMs

- El espacio original **siempre** puede ser **mapeado** en otro espacio de características donde el conjunto de entrenamiento sea **linealmente separable**:

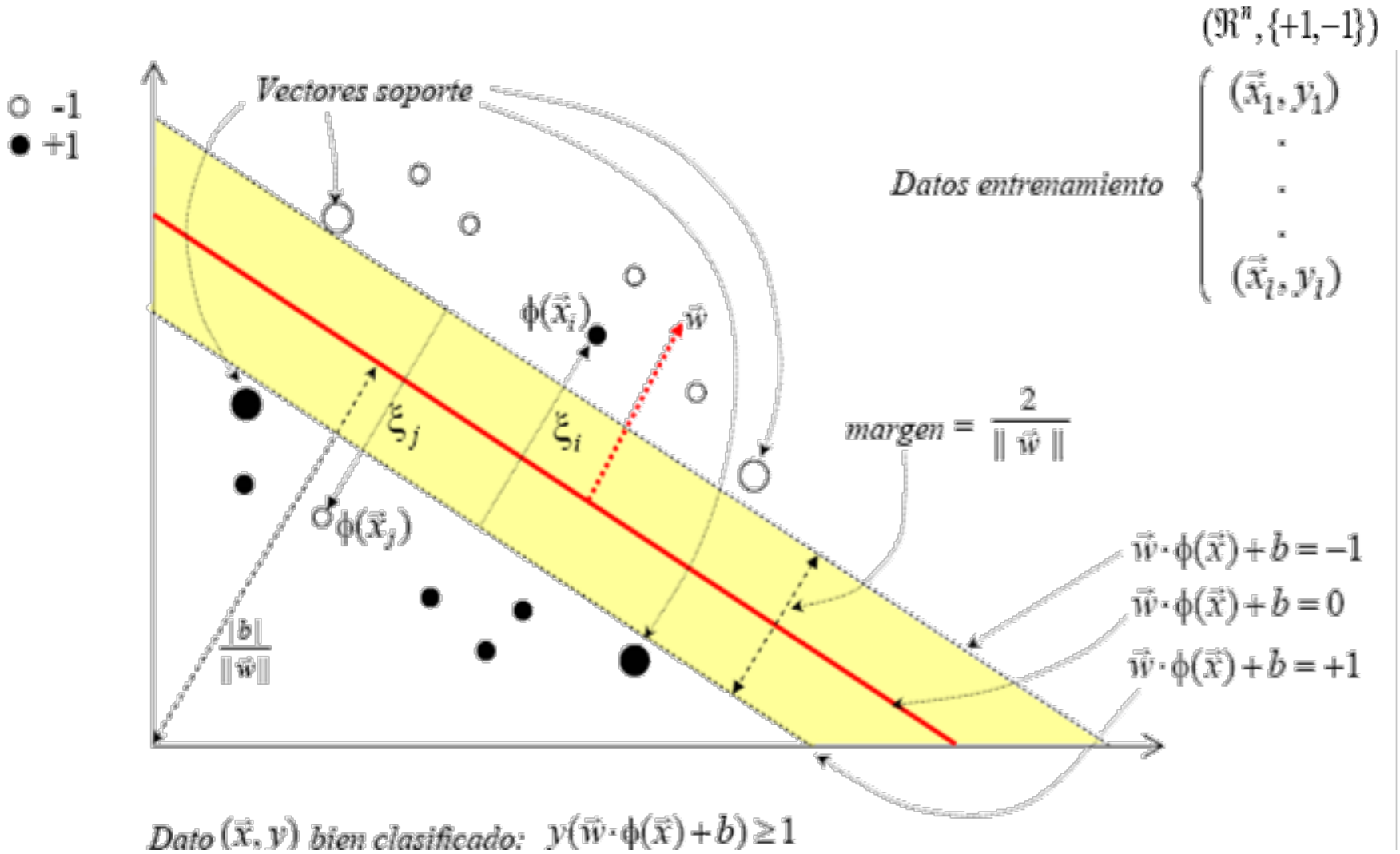


SVMs

- Para mapear los espacios de características se utilizan **funciones kernel**:
  - **Función kernel genérica**: $K(\vec{x}_i, \vec{x}_j) \equiv \langle \phi(\vec{x}_i) \cdot \phi(\vec{x}_j) \rangle$
  - **Funciones kernel utilizadas**:
    - **Lineal:** $K(x_i, x_j) = x_i \cdot x_j$
    - **Sigmoide:** $K(\vec{x}_i, \vec{x}_j) = \tanh(\gamma \vec{x}_i \cdot \vec{x}_j + coef)$
    - **Polinómica:** $K(x_i, x_j) = (x_i \cdot x_j + 1)^d$
    - **Gaussiana (RBF):** $K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$
- **Función de clasificación:**

$$f(\mathbf{x}) = \sum_{\substack{\text{vectores} \\ \text{soporte}}} \alpha_i y_i (\Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}_i)) + b = \sum_{\substack{\text{vectores} \\ \text{soporte}}} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b$$

SVMs



http://www.infor.uva.es/~isaac/doctorado/Cap07_SVM.ppt

SVMs

- Función a minimizar (sin permitir errores):

$$\frac{1}{2}\vec{w}\cdot\vec{w}$$

- sujeta a: $y_i(\vec{w}\cdot\phi(\vec{x}_i)+b)\geq 1 \quad (i=1,\ldots,N)$

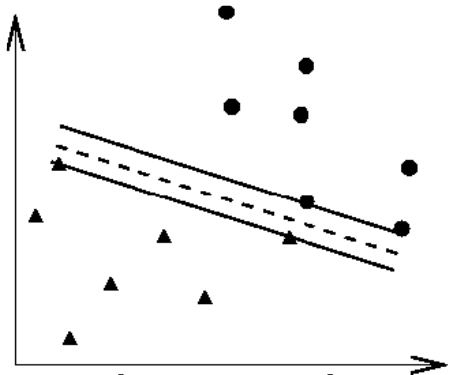
- Función a minimizar (permitiendo errores):

$$\frac{1}{2}\vec{w}\cdot\vec{w}+C\sum_{i=1}^{N}\xi_i$$

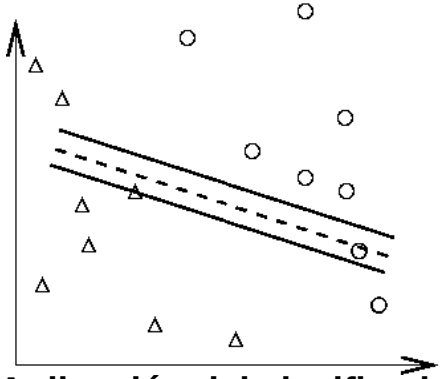
- sujeta a: $y_i(\vec{w}\cdot\phi(\vec{x}_i)+b)\geq 1-\xi_i \quad (\xi_i\geq 0,\ i=1,\ldots,N)$

SVMs

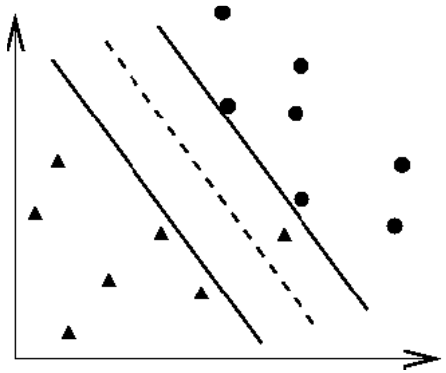
- Ejemplos de SVM con datos de entrenamiento erróneos



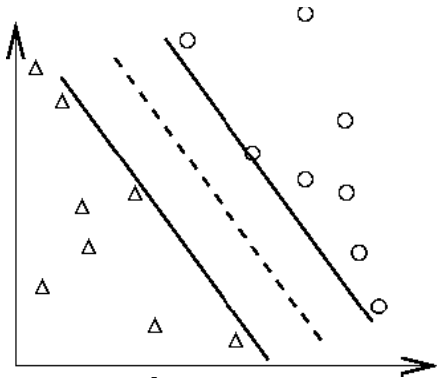
a) Datos de entrenamiento y un clasificador sobreentrenado



b) Aplicación del clasificador (a) sobre los datos de prueba



c) Datos de entrenamiento y un clasificador mejor entrenado



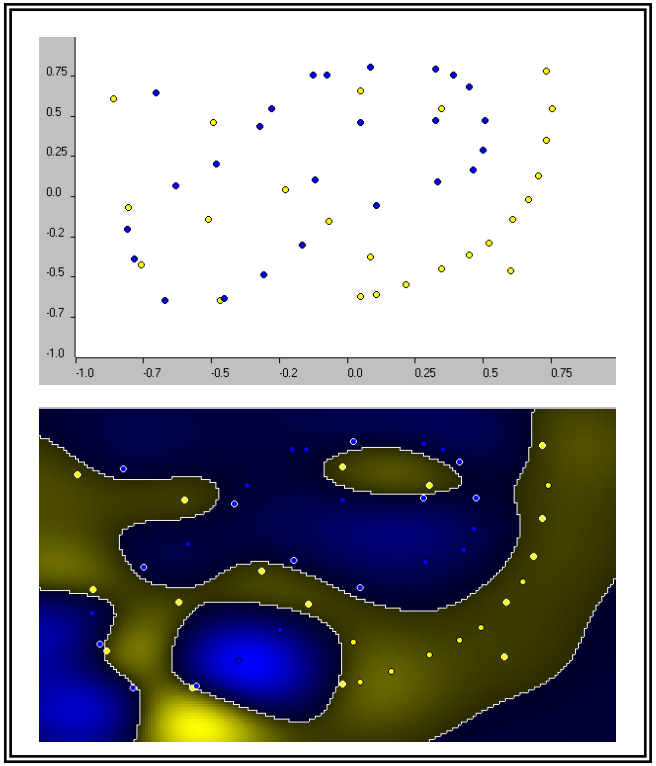
c) Aplicación de un clasificador mejor sobre los datos de prueba

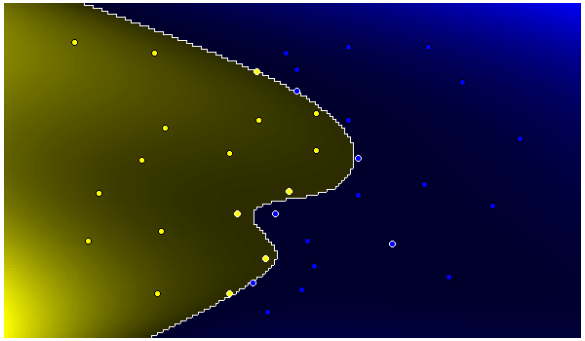
SVMs

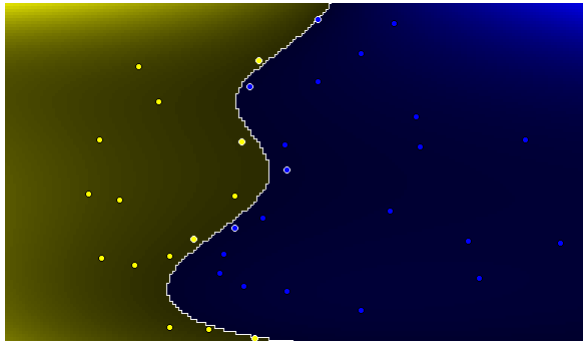
- Problemas de los SVMs:
  - Son muy sensibles al ruido.
  - Un número pequeño de ejemplos mal etiquetados puede hacer que el rendimiento se reduzca dramáticamente.
- Solamente considera 2 clases. ¿cómo se pueden utilizar SVMs para clasificar más de 2 clases?
  1. Entrenando varios SVMs en paralelo:
     - SVM 1 aprende “Output==1” vs “Output != 1”
     - SVM 2 aprende “Output==2” vs “Output != 2”
     - :
     - SVM m aprende “Output==m” vs “Output != m”
  2. La salida será el SVM cuyo resultado sea mayor.

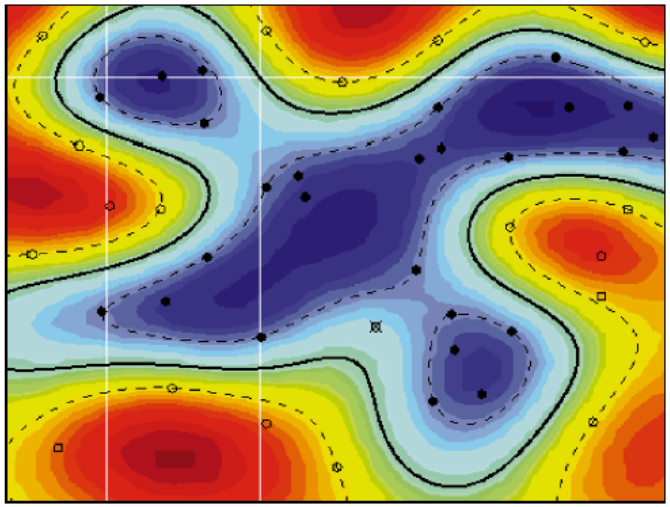
SVMs

- Separación polinómica y con RBF









http://dis.unal.edu.co/~fgonza/courses/2003/pmge/present/ExpoSVM.ppt