K-Nearest Neighbours (KNN)

Aprendizaje Automático INF-393 II-2018

Ricardo Ñanculef

UTFSM Campus San Joaquín

Table of contents

Método Básico

KNN en Clasificación

- Consideremos un problema de clasificación estándar, con datos representados en $\mathbb{X} \subset \mathbb{R}^d$, categorías $\mathbb{Y} = \{c_1, c_2, \ldots, c_K\}$, y conjunto de ejemplos $S = \{(x^{(\ell)}, y^{(\ell)})\}_{\ell=1}^n$. Sea además $S_x = \{x^{(\ell)}\}_{\ell=1}^n$
- Supongamos que $\mathbb{X}$ está equipado con una métrica o distancia, es decir con una función $m : \mathbb{X} \times \mathbb{X} \to \mathbb{R}_0^+$ que satisface las siguientes propiedades

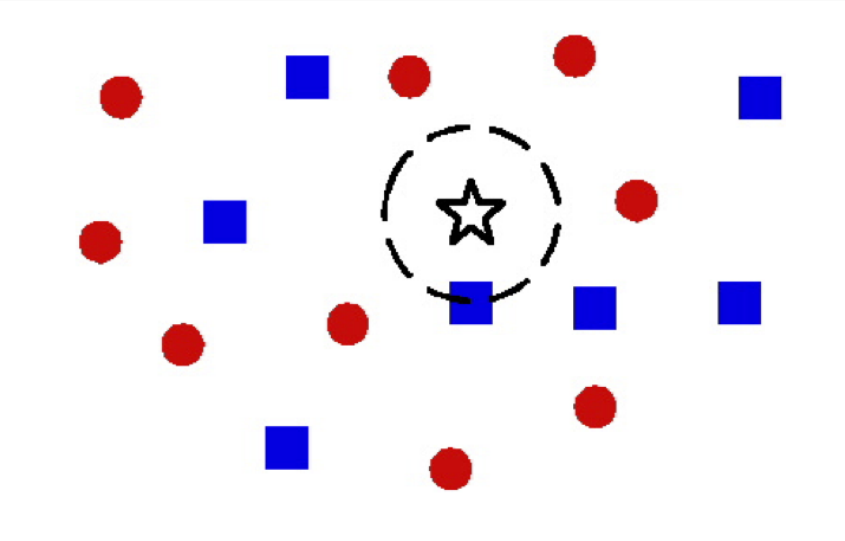
$$
\begin{aligned}
& m(a,b) = m(b,a)\ \forall a, b, \in \mathbb{X} \qquad (1)\\
& m(a,b) = 0, \Rightarrow\ \ a = b\ \forall a \in \mathbb{X}\\
& m(a,b) \leq m(a,c) + m(c,b)\ \forall a, b, c \in \mathbb{X}
\end{aligned}
$$

- Definamos además, $c(x^{(\ell)}) = y^{(\ell)}$, $\forall \ell = 1, \ldots, n$.

1NN en Clasificación

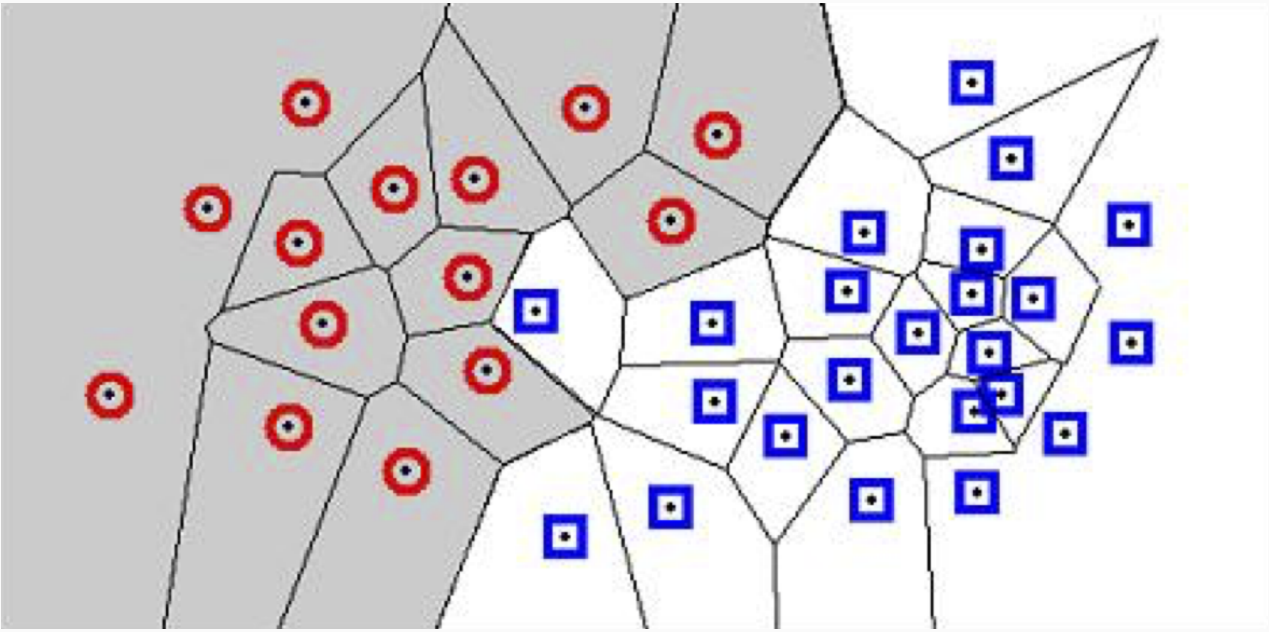
- Para clasificar un nuevo dato x, el clasificador de vecino más cercano (1NN) implementa la siguiente regla

$$f(x) = c\left(\underset{x^{(\ell)} \in S_x}{\arg\min}\ m(x, x^{(\ell)})\right) \tag{2}$$



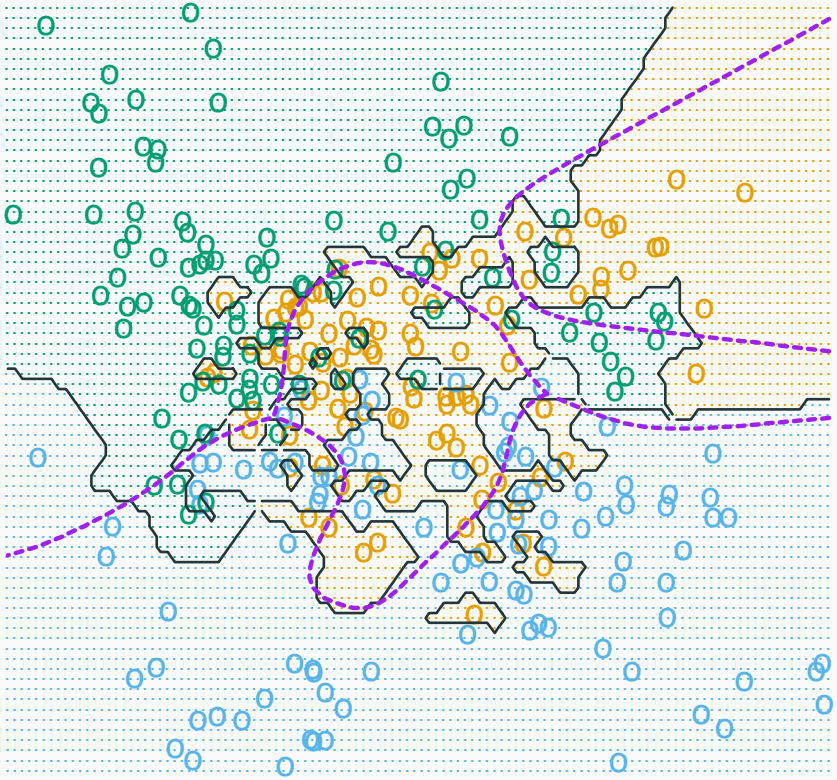
Fronteras de 1NN

• Las fronteras correspondientes a esta regla forman parte de un diagrama de Voronoi construido sobre el conjunto de puntos.



Fronteras de 1NN

- En problemas reales estas fronteras pueden ser arbitrariamente no-lineales.



KNN en Clasificación

- Para clasificar un dato x, KNN implementa el siguiente algoritmo:

1. Encontrar los K vecinos más cercanos de x: $x_{(1)}, x_{(2)}, \ldots, x_{(K)}$, es decir un conjunto de K elementos $N(x)$ tal que

$$\forall x_{(i)} \in N,\ \forall x_{\dagger} \in S_x - N,\ m(x, x_{(i)}) \leq m(x, x_{\dagger})\,. \tag{3}$$

2. Contar el número de veces que cada clase aparece entre las etiquetas de los vecinos:

$$r(c_j) = \sum_{i=1}^{K} I(c(x_{(i)}) = c_j)\,. \tag{4}$$

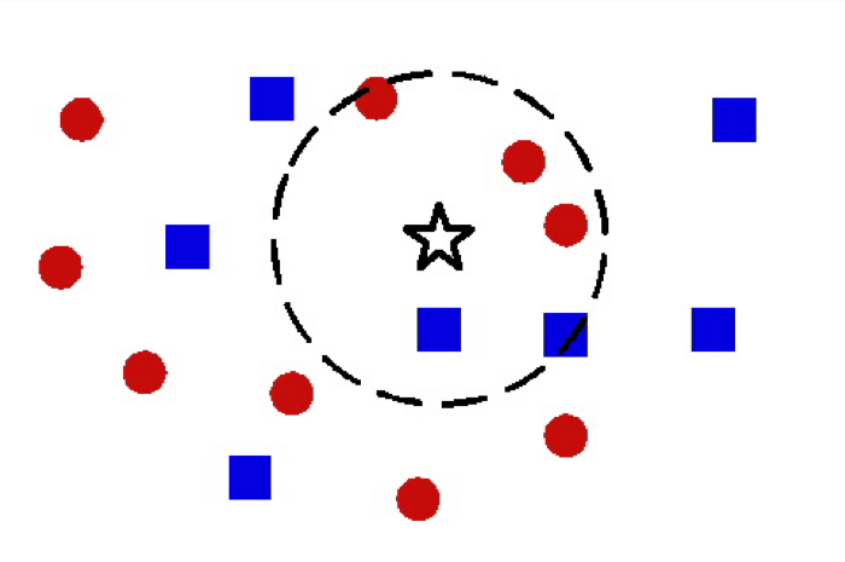
3. Elegir la clase más popular[1]:

$$f(x) = \arg\max_{c_i}\ r(c_i) \tag{5}$$

[1] los empates se rompen aleatoriamente.

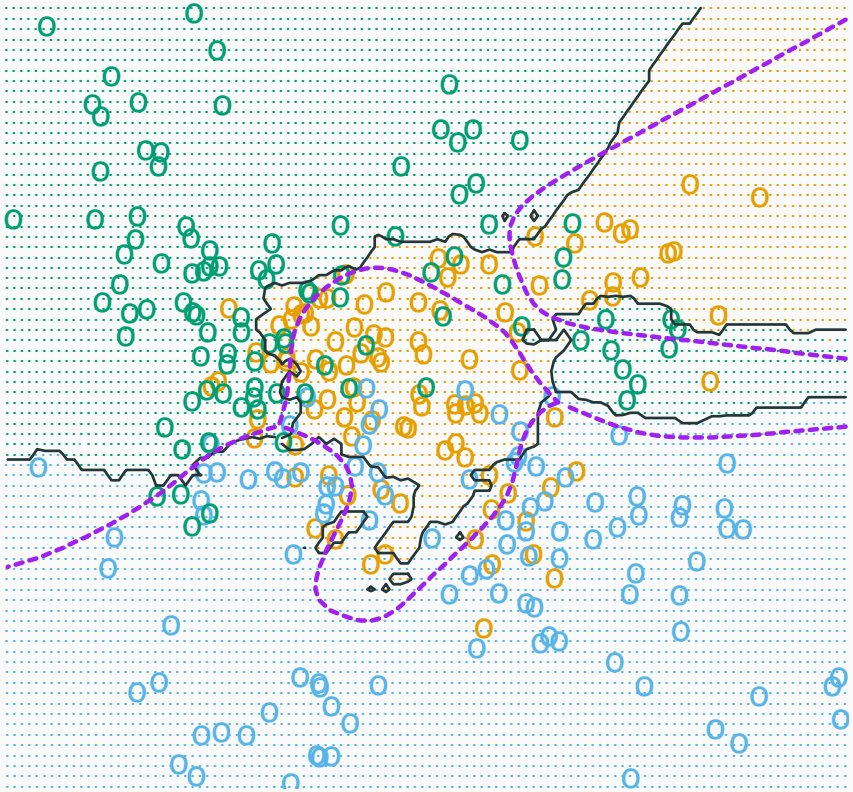
KNN en Clasificación

- En vez de considerar 1 vecino, considerar K, eligiendo la clase más popular del vecindario.



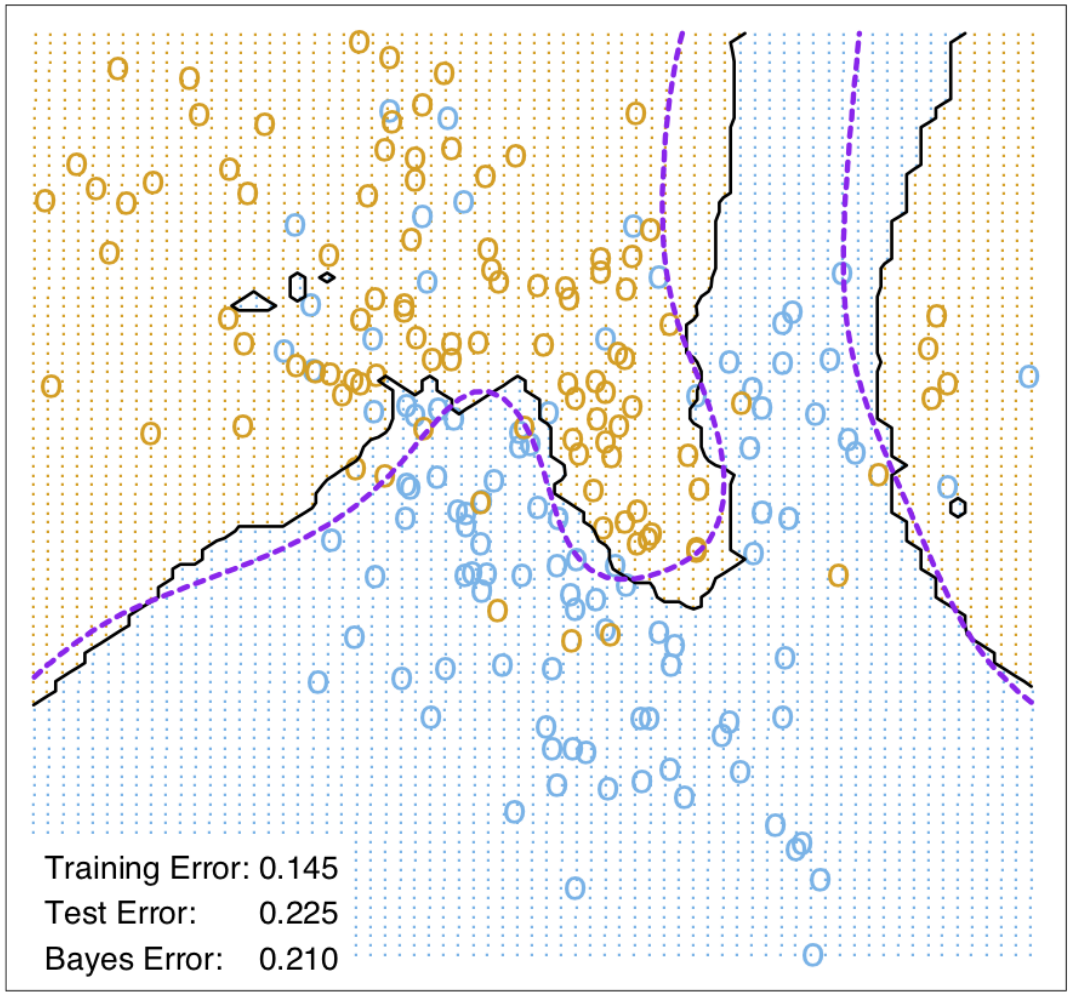
Fronteras de KNN

- En general, las fronteras se suavizan a medida que usamos un K más grande. En este sentido, a mayor K, la solución "se regulariza".



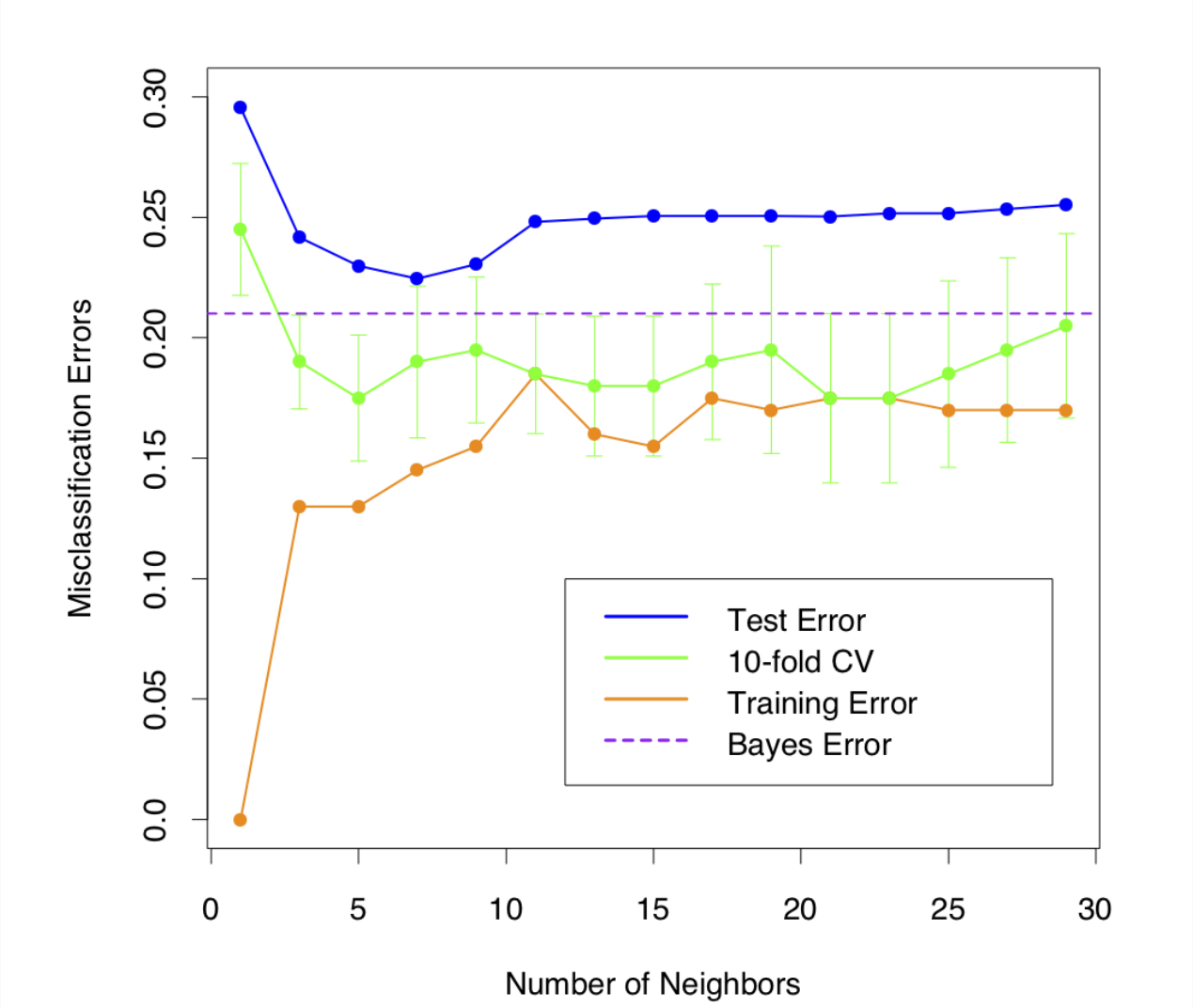
KNN versus K

- El error de predicción (test) óptimo no necesariamente disminuye al aumentar de K.



KNN versus K

- El error de predicción (test) óptimo no necesariamente disminuye al aumentar de K.



KNN en Regresión

- Para predecir una respuesta continua, KNN simplemente cambia la "estadística" aplicado a las etiquetas de los vecinos.
- Lo más común es usar la media.

1. Encontrar los K vecinos más cercanos de x: $x_{(1)}, x_{(2)}, \ldots, x_{(K)}$ y sus respectivas etiquetas $y_{(1)}, y_{(2)}, \ldots, y_{(K)}$.
2. Predecir

$$f(x) = \frac{1}{K} \sum_{i=1}^{K} y_{(1)} \,. \tag{6}$$

Lo Bueno

- Conceptualmente muy simple. Es muy conocido y utilizado en la práctica.
- El entrenamiento es muy eficiente en términos de tiempo de cómputo: sólo consiste en almacenar (memorizar) los ejemplos.
- Funciona bastante bien cuando la dimensionalidad es baja.
- Es un método naturalmente no-paramétrico (se adapta a la complejidad del problema) y no asume una forma específica de las fronteras de clasificación.
- Puede acomodar métricas especializadas, si se conocen (e.g. conjuntos, textos, imágenes).

Lo Malo

- El costo computacional en fase de decisión es muy alto y puede hacerlo inviable como solución.
  - $\mathcal{O}(nd)$ (tiempo y espacio) sin estructuras de datos especializadas o algoritmos especializados de búsqueda por similaridad.
  - Estructuras de datos especializadas (KDtree,BallTree) tienden a escalar muy mal en memoria con $d > 3$.
- Es extremadamente sensible a la maldición de la dimensionalidad. Reducción de dimensionalidad es aconsejable, aunque le hace perder simplicidad.
- La métrica utilizada podría no ser adecuada.

Algunas Propiedades

Error de Bayes

- Recordemos que error de Bayes representa el menor error que se puede conseguir en un problema de clasificación.
- Si $p(x, y)$ es la distribución real de las observaciones, la regla óptima de decisión es $f(x) = c_m$ con

$$m = \underset{j \in \{1,\ldots,K\}}{\arg\max}\ p(y = c_j | x) \tag{7}$$

- El error de Bayes es entonces

$$B^*(x) = P(\text{Bayes se equivoque} \,|\, x) = 1 - p(c_m) \tag{8}$$

$$B^* = P(\text{Bayes se equivoque}) = \sum_x B^*(x)\, p(x)\,.$$

Error Asintótico de 1NN

- Sea $P_n(e|x)$ el error condicional del clasificador 1NN cuando se "entrena" con n ejemplos y $P_n(x)$ su valor esperado.

$$P_n(e|x) = P(\text{1NN se equivoque} \,|\, x) \tag{9}$$

$$P_n(e) = P(\text{1NN se equivoque}) = \sum_x P_n(e|x)p(x)\,.$$

- Los errores asintóticos del clasificador se definen como

$$P(e|x) = \lim_{n\to\infty} P_n(e|x) \tag{10}$$

$$P(e) = \sum_x P(e|x)p(x)\,.$$

Teorema (Cover & Hart)

Bajo ciertas condiciones de regularidad bastante generales,

$$B^* \leq P(e) \leq B^* \left(2 - \frac{K}{K-1} B^*\right) \leq 2B^*\,.$$

Error Asintótico de 1NN

- Si denotamos por $Q_n(x_{(1)}|x)$ la distribución de los vecinos de x sobre una muestra de tamaño n, la demostración del Teorema anterior requiere

$$\lim_{n\to\infty} Q_n(x_{(1)}|x) = \delta(d(x, x_{(1)}))\,.$$

- El punto de partida es esta simple observación

$$P_n(e|x) = 1 - \sum_k P_n(e|x, y_{(1)} = c_k) P_n(y_{(1)} = c_k|x)\,, \tag{11}$$

que nos lleva a

$$P(e|x) = 1 - \sum_k P^2(c_k|x) \tag{12}$$

$$P(e) = \sum_x \left(1 - \sum_k P^2(c_k|x)\right) p(x)\,, \tag{13}$$

Error Asintótico de 1NN

- Para $K = 2$ es muy simple mostrar que

$$1 - \sum_k P^2(c_k|x) \geq 2B^*(x) - B^{*2}(x)\,, \tag{14}$$

Observando que

$$Var(B^*(x)) \geq 0 \Rightarrow \sum_x B^{*2}(x)p(x) > \left(\sum_x B^*(x)p(x)\right)^2 \tag{15}$$

obtenemos

$$P(e) \leq 2B^* - B^{*2}\,. \tag{16}$$

- Generalizar a $K > 2$.

Error Asintótico de KNN

- Definiendo

$$P_n^{(K)}(e|x) = P(\text{KNN se equivoque} \,|\, x) \tag{17}$$
$$P_n^{(K)}(e) = P(\text{KNN se equivoque}) = \sum_x P_n^{(K)}(e|x)p(x)\,.$$

- Es posible mostrar que existe una función convexa C_K tal que

$$P^{(K)}(e|x) = \lim_{n\to\infty} P_n^{(K)}(e|x) \leq C_K(B^*(x)) \tag{18}$$

- Usando Jensen

$$P^{(K)}(e) = \mathbb{E}\left(P^{(K)}(e|x)\right) \leq \mathbb{E}\left(C_K(B^*(x))\right) \leq C_K(B^*)\,.$$

- Además,

$$B^* \leq P^{(K)}(e) \leq C_K(B^*) \leq C_{K-1}(B^*) \leq \ldots \leq C_1(B^*) \leq 2B^*(1-B^*)\,.$$

Algunas Mejoras

KNN con Pesos

- Una idea antigua, que conecta KNN con métodos no paramétricos denominados *Kernel Smoothers* (e.g. Nadaraya–Watson) es usar pesos para cada vecino.
- Regla modificada:

1. Encontrar los K vecinos más cercanos de x: $x_{(1)}, x_{(2)}, \ldots, x_{(K)}$ y las respectivas distancias $m_{(1)}, m_{(2)}, \ldots, m_{(K)}$.
2. Definir $w_i = k(m_{(i)})$, donde $k()$ es el kernel elegido.
3. Cada vecino "vota" por una clase con su peso w_i

$$r(c_j) = \sum_{i=1}^{K} w_i I(c(x_{(i)}) = c_j)\,. \tag{19}$$

4. Elegir la clase más popular[2]:

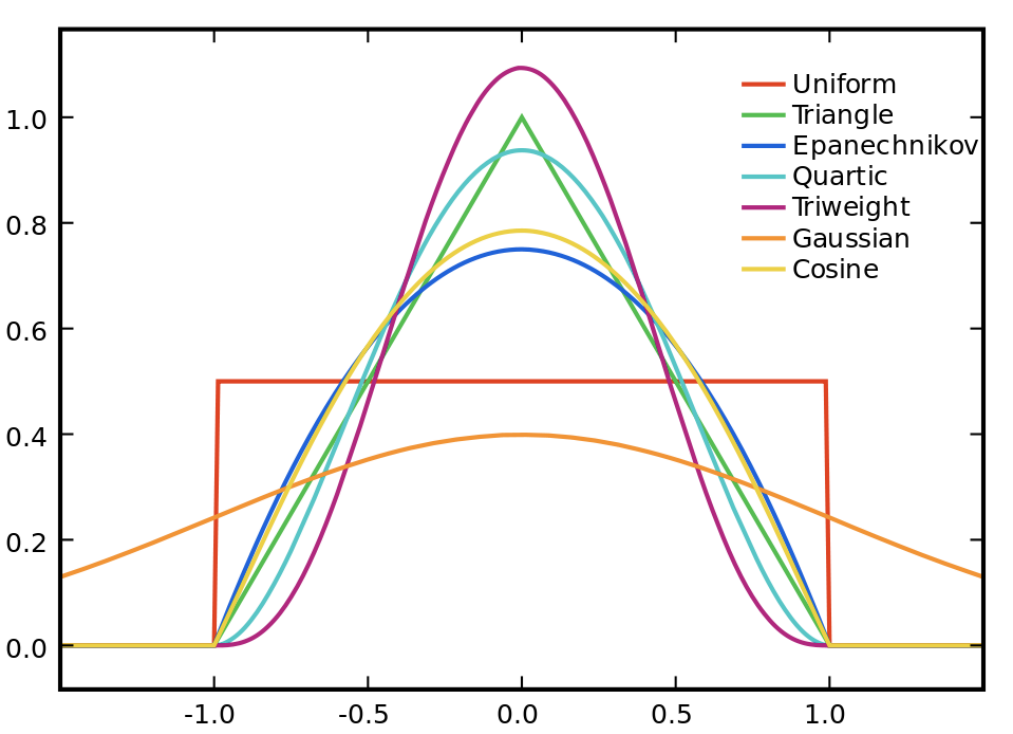
$$f(x) = \arg\max_{c_i} r(c_i) \tag{20}$$

[2] los empates se rompen aleatoriamente.

Kernels

- La función de kernel debe satisfacer

  1. $k(d) \geq 0, \ \forall d$.
  2. $k(0)$ es máximo.
  3. $k(d)$ es decreciente en d.



KNN con Pesos

- La elección del kernel suele no ser gravitante.
- Es importante normalizar las distancias para obtener números en $[0, 1]$.

El método típico es calcular la distancia al vecino $K + 1$-ésimo.

1. Encontrar los $K + 1$ vecinos más cercanos de x:
   $x_{(1)}, x_{(2)}, \ldots, x_{(K)}, x_{(K+1)}$ y las respectivas distancias
   $m_{(1)}, m_{(2)}, \ldots, m_{(K)}, m_{(K+1)}$.
2. Definir $w_i = k(m_{(i)}/m_{(K+1)})$, donde $k()$ es el kernel elegido.
3. Cada vecino "vota" por una clase con su peso w_i

$$r(c_j) = \sum_{i=1}^{K} w_i I(c(x_{(i)}) = c_j)\,. \tag{21}$$

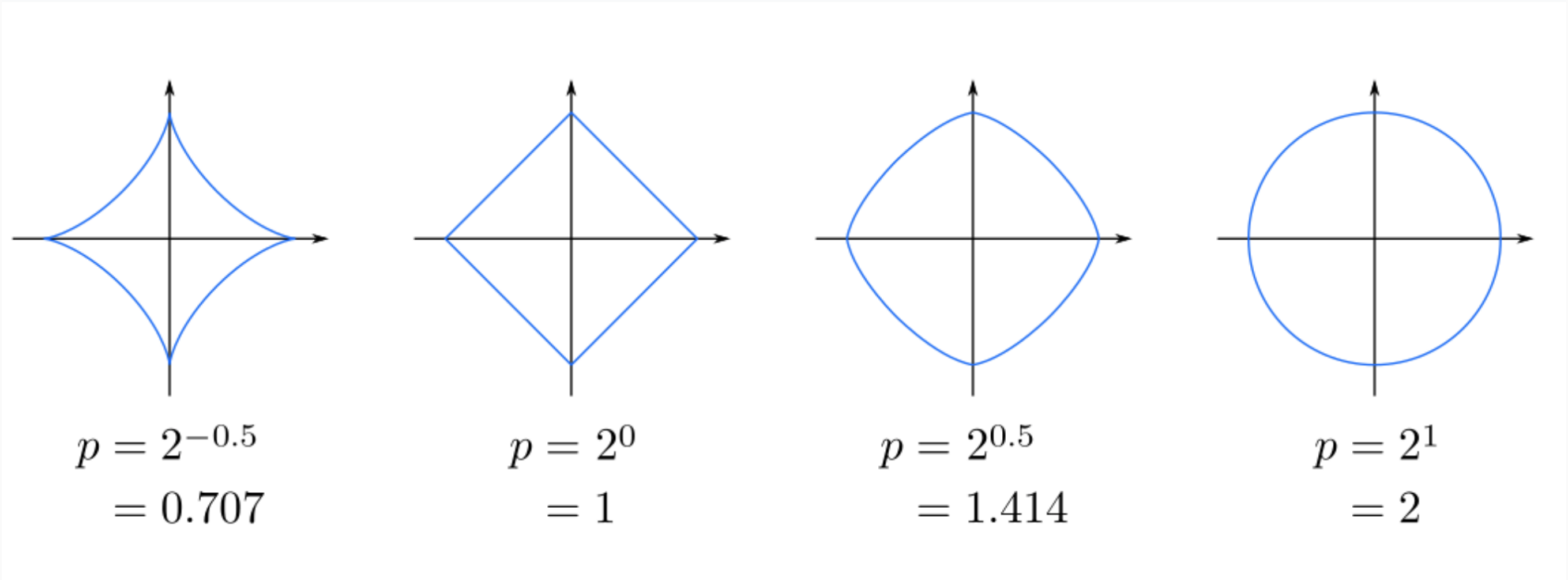
4. Elegir la clase más popular:

$$f(x) = \arg\max_{c_i} r(c_i) \tag{22}$$

KNN con Métricas Adaptativas

- La elección de la métrica es por supuesto fundamental para KNN.
- Típicamente $\mathbb{X} = \mathbb{R}^d$, los datos se "normalizan" de modo que los atributos tengan la misma escala y la métrica es de la forma

$$m(a, b) = (\sum_i |a_i - b_i|^p)^{1/p} . \tag{23}$$



KNN con Métricas Adaptativas

- Una alternativa es "adaptar" la métrica a los datos. Por ejemplo, es frecuente considerar una métrica de la forma

$$m(a, b) = (a - b)^T M(a - b)\,, \tag{24}$$

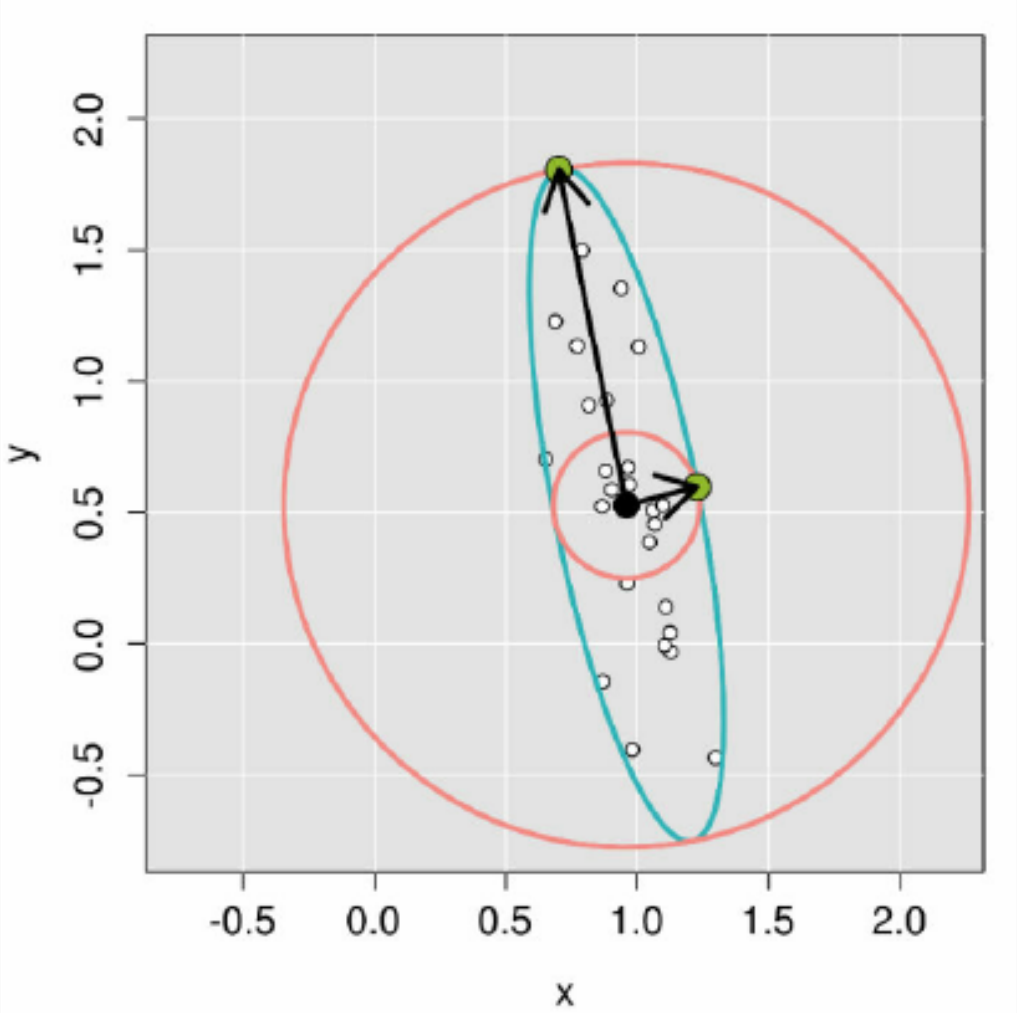
donde $M \in \mathbb{R}^{d \times d}$ es una matriz que se determina con algún criterio a partir del conjunto de entrenamiento.

- Cuando $M = \Sigma^{-1}$

$$\Sigma \approx \mathbb{E}\left(xx^T\right) - \mathbb{E}(x)\mathbb{E}(x)^T\,. \tag{25}$$

la métrica se denomina Métrica de Mahalanobis.

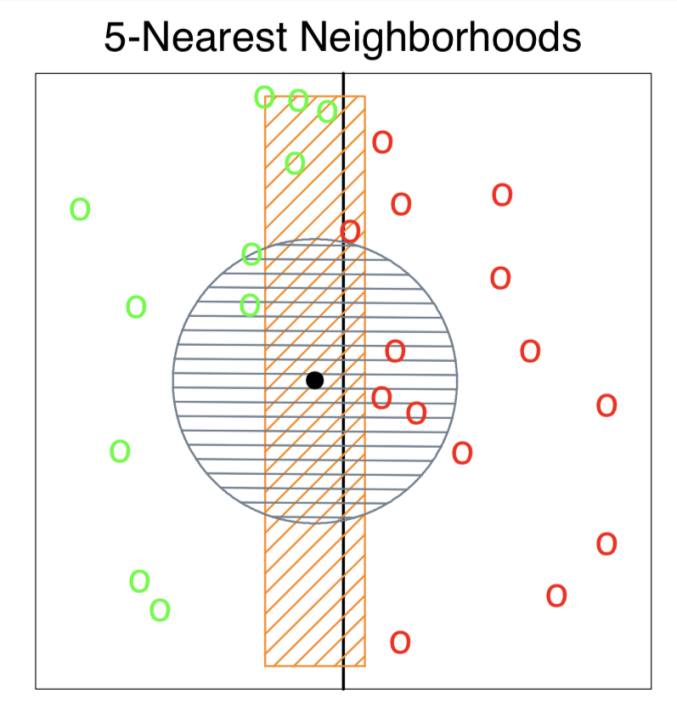
Métrica de Mahalanobis



PCA & LDA

- La matriz M puede ser una matriz de reducción de dimensionalidad ($P^T P$ con P un proyector), que "pese más" las direcciones relevantes del espacio característico.

- Ambas técnicas pueden aplicarse localmente, aumentando significativamente el costo computacional del clasificador.



Edición & Condensación

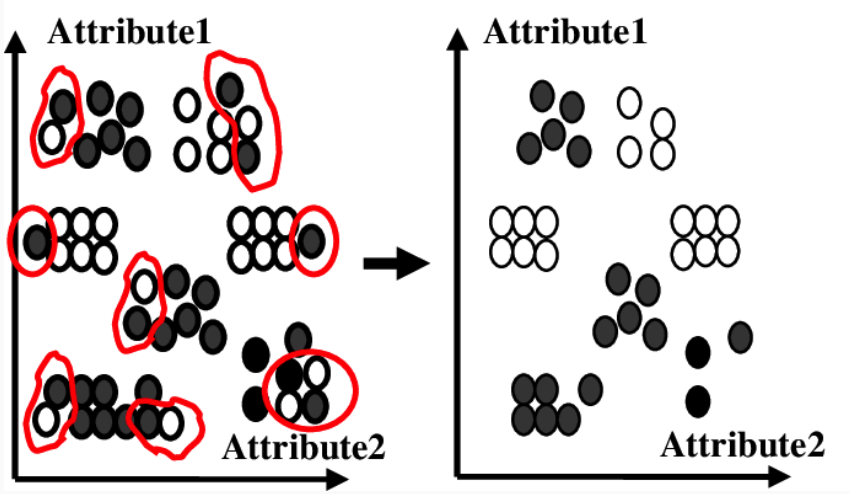
Ideas Clave

- El principal defecto de KNN es el gran costo computacional en tiempo de decisión. Gran parte de la investigación asociada a este método se concentra en atacar este problema.
- Un método de Edición intenta seleccionar un subconjunto del conjunto de entrenamiento que genere un clasificador más consistente (mejores cotas de generalización).
- Un método de Condensación intenta seleccionar un subconjunto del conjunto de entrenamiento que preserve las fronteras de KNN.
- Métodos de edición famosos: Wilson, Edición Múltiple de Devijer & Kittler.
- Métodos de condensación famosos: Delaunay, Gabriel, RNG.

Edición de Wilson

- Hipótesis: los ejemplos clasificados incorrectamente por KNN reducen su consistencia.
- Algoritmo:

1. Entrenar KNN con S.
2. Clasificar S e incluir todos los ejemplos mal clasificados en M.
3. Entrenar KNN con $S - M$.



Edición de Wilson

- Variante 1:

  1. $E = S$:
  2. Por cada ejemplo en $x \in S$:
     - 2.1 Entrenar KNN con $E - \{x\}$.
     - 2.2 Clasificar x.
     - 2.3 Si x está mal clasificado, $E = E - \{x\}$.

- Variante 2:

  1. Dividir el conjunto de entrenamiento en dos partes S_1 (entrenamiento) y S_2 (test).
  2. Entrenar KNN con S_1.
  3. Clasificar S_2 e incluir todos los ejemplos mal clasificados en M.
  4. Entrenar KNN con $S_2 - M$.

Edición de Wilson

- Sean $P_K^E(e|x)$, $P_K^E(e)$ los error asintóticos de KNN entrenado con el conjunto editado.
- En el caso de clasificación binaria, es posible demostrar que

$$P_1^E(e|x) \leq \frac{P(e|x)}{2(1 - P(e|x))} \leq P(e|x)\,, \tag{26}$$

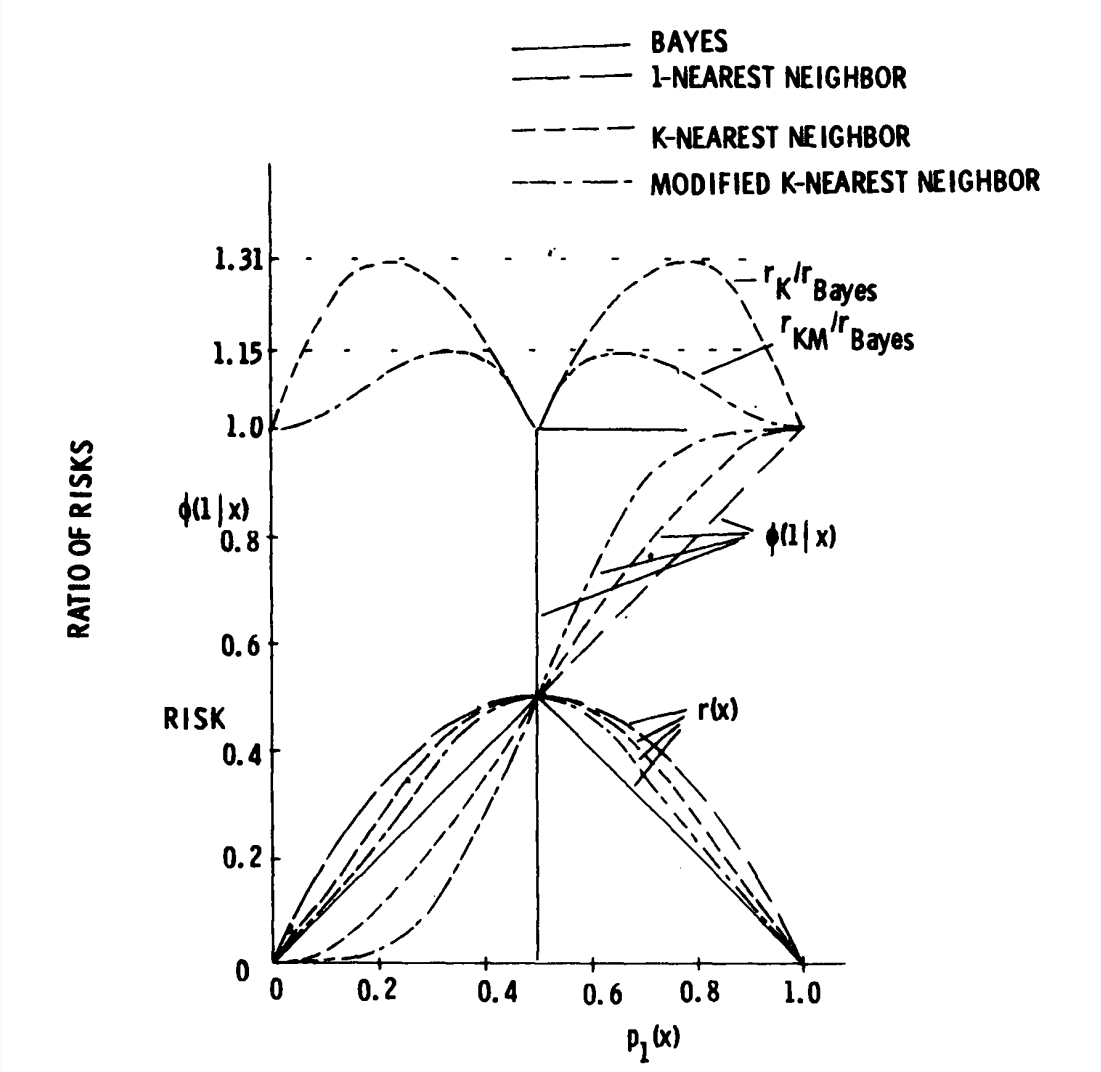
de modo que el método editado con $K = 1$, mejora sobre $1NN$ básico

$$P_1^E(e) \leq P(e)\,, \tag{27}$$

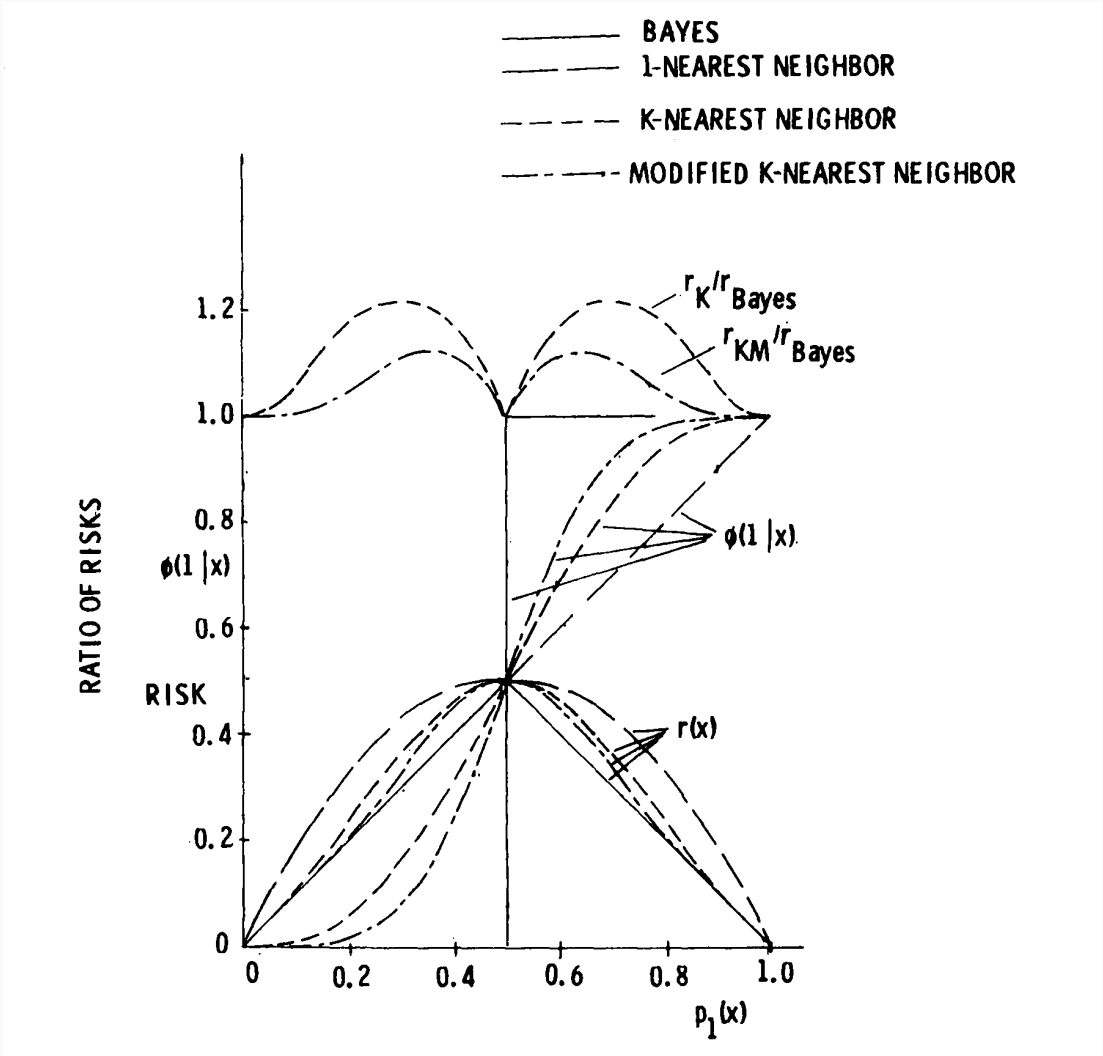
- Si $P(e|x)$ es pequeño, el método editado es prácticamente óptimo

$$P_1^E(e) \approx \frac{P(e)}{2} \approx B^*\,, \tag{28}$$

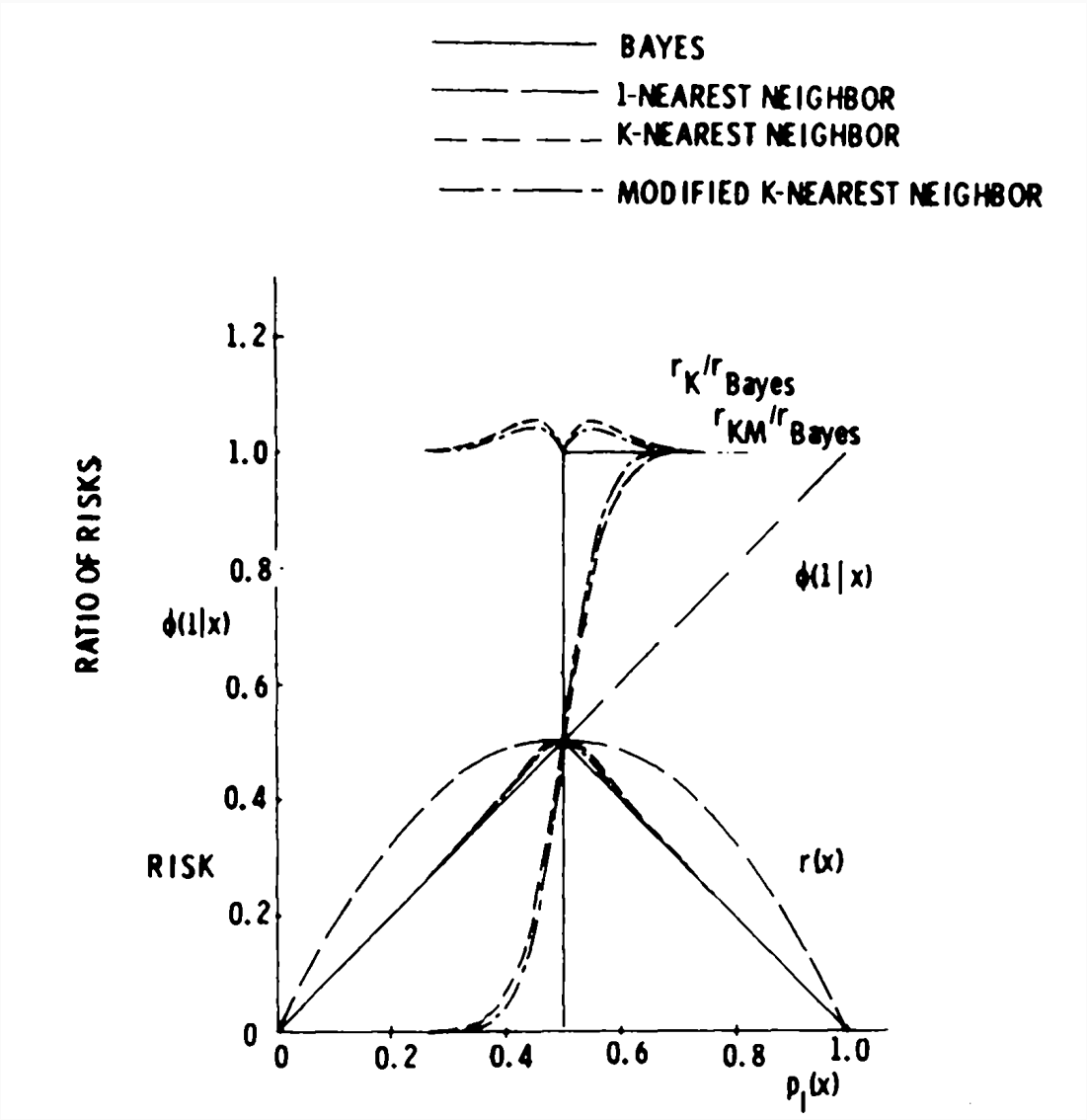
Edición de Wilson



Edición de Wilson



Edición de Wilson



Edición Múltiple de Devijer & Kittler

- Idea: Dividir el conjunto de ejemplos en varios sub-grupos y usar edición cruzada entre los sub-grupos.
- Algoritmo:

1. $E = S$.
2. Dividir E en $M > 2$ sub-grupos $E_1, E_2, \ldots, E_M$.
3. Clasificar los ejemplos de E_i con el clasificador entrenado en $E_{i+1|M}$ e incluir todos los ejemplos mal clasificados en M_i.
4. $E = E - M$.
5. Si no ha habido cambios en E deternerse. Sino, volver a 2.

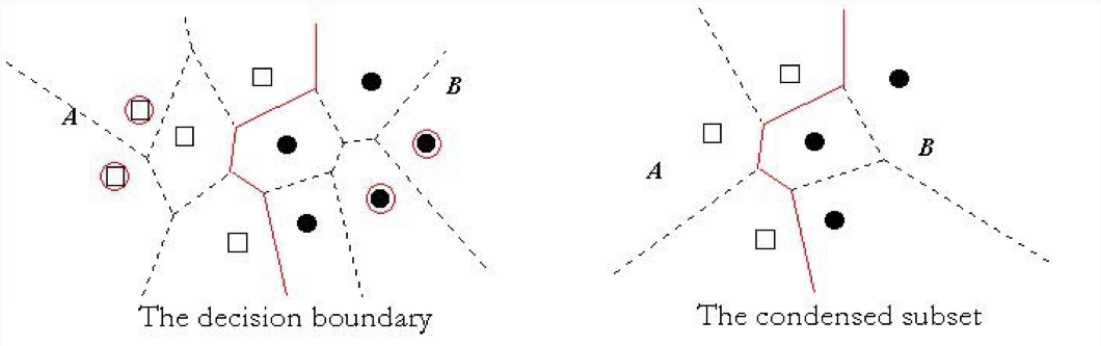
Condesación basada en Grafos de Proximidad

- Objetivo: Eliminar puntos del conjunto de entrenamiento preservando la frontera.
- Grafos de Proximidad: Casi todos los métodos se basan la re-definición del concepto de vecinos en el conjunto de entrenamiento.
- Método Genérico:

1. Para cada punto x del conjunto de entrenamiento, chequear si todos sus vecinos son de la misma clase. Si es así marcarlo.
2. Eliminar del conjunto, todos los puntos marcados.

Condesación de Delaunay

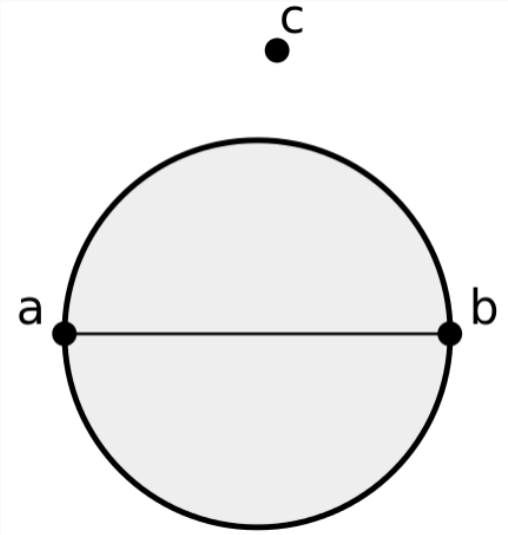
- Definición de vecinos: Tres puntos son mutuamente vecinos si la circunferencia circunscrita no contiene otros puntos.
- Alternativamente, dos puntos son vecinos si los une una arista en el grafo de Delaunay, es decir, el grafo que une un punto con todos aquellos puntos contiguos en un diagrama de Voronoi.

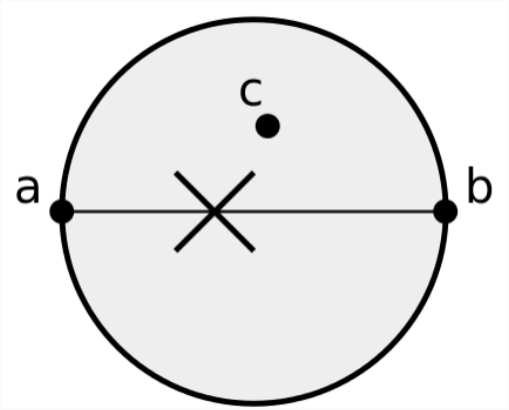


- El método garantiza la preservación de la frontera. - Extremadamente costoso para $d > 3$.

Condesación de Gabriel

- Definición de vecinos: Dos puntos son mutuamente vecinos si la circunferencia de menor radio que contiene ambos puntos, no contiene otros puntos.

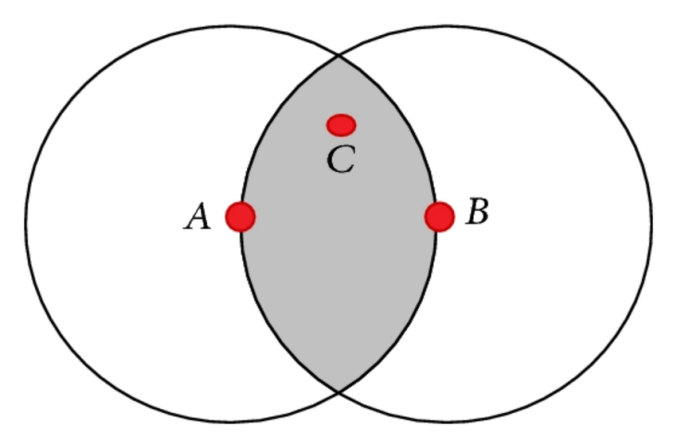




- El método NO garantiza la preservación de la frontera, pero los cambios ocurren fuera de la envoltura convexa del conjunto de puntos.
- Es posible ejecutarlo de modo eficiente.

Condesación vía RNG (Relative Neighbour Graph)

- Definición de vecinos: Dos puntos a, b son mutuamente vecinos si la circunferencia de radio $m(a, b)$ en torno a a no contiene otros puntos (fuera de b) y si la circunferencia de radio $m(a, b)$ en torno a b no contiene otros puntos (fuera de a).



- Cambia significativamente la frontera.
- Es posible demostrar que el clasificador obtenido no es consistente.