

K-Nearest Neighbours (KNN)

Aprendizaje Automático INF-398 II-2021

Ricardo Ñanculef

UTFSM Campus San Joaquín

Table of contents

1. Método Básico
2. Algunas Propiedades
3. Algunas Mejoras
4. Edición & Condensación

Método Básico

KNN en Clasificación

- Consideremos un problema de clasificación estándar, con datos representados en $\mathbb{X} \subset \mathbb{R}^d$, categorías $\mathbb{Y} = \{c_1, c_2, \dots, c_K\}$, 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 $m : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}_0^+$ que satisface las siguientes propiedades

$$m(a, b) = m(b, a) \quad \forall a, b \in \mathbb{X} \tag{1}$$

$$m(a, b) = 0, \Rightarrow a = b \quad \forall a \in \mathbb{X}$$

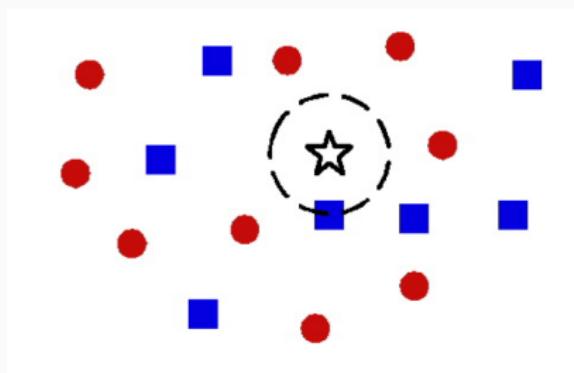
$$m(a, b) \leq m(a, c) + m(c, b) \quad \forall a, b, c \in \mathbb{X}$$

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

1NN en Clasificación

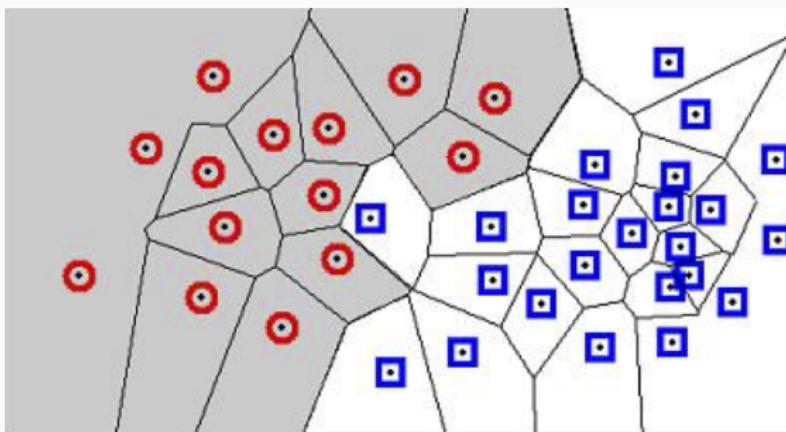
- Para clasificar un nuevo dato x , el clasificador de vecino más cercano (1NN) implementa la siguiente regla

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



Fronteras de 1NN

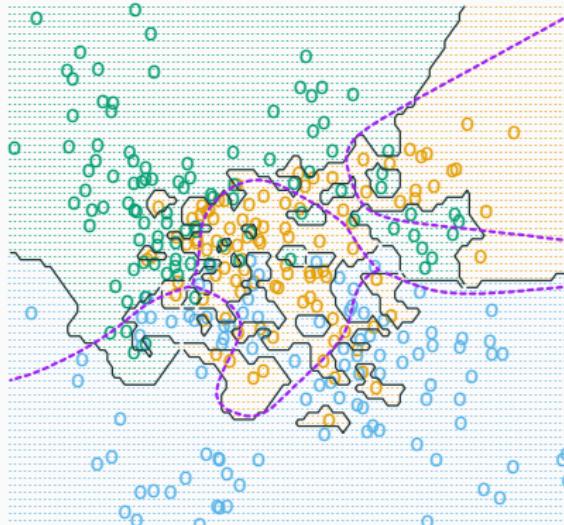
- Si consideramos las celdas de Voronoi correspondientes al conjunto de entrenamiento, la frontera de decisión de este método es la **unión de las líneas en que se encuentran celdas de clases diferentes**.



(La celda de Voronoi de un punto $p \in S \subset \mathbb{X}$ es el conjunto de todos los puntos $q \in \mathbb{X}$ que están más cerca de p que de cualquier otro punto de S .

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:
 - Encontrar los K vecinos más cercanos de x : $x_{(1)}, x_{(2)}, \dots, x_{(K)}$, es decir un conjunto de K elementos $N(x)$ tal que
$$\forall x_{(i)} \in N, \forall x_{\dagger} \in S_x - N, m(x, x_{(i)}) \leq m(x, x_{\dagger}). \quad (3)$$
 - Contar el número de veces que cada clase aparece entre las etiquetas de los vecinos:

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

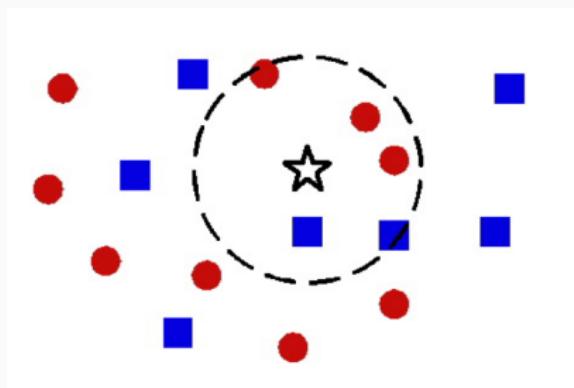
- Elegir la clase más popular¹:

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

¹los empates se rompen aleatoriamente.

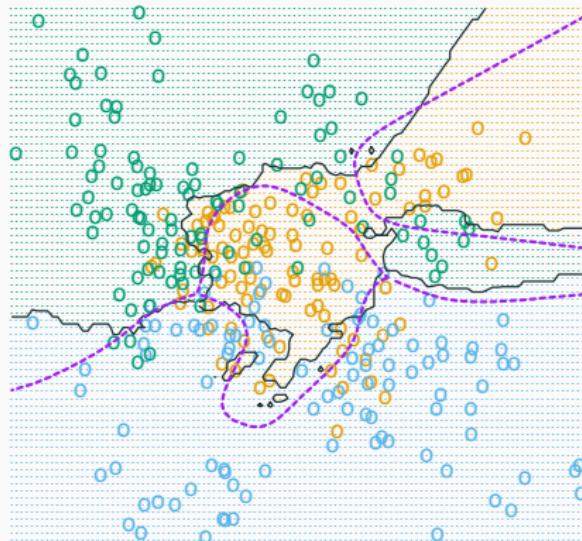
KNN en Clasificación

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



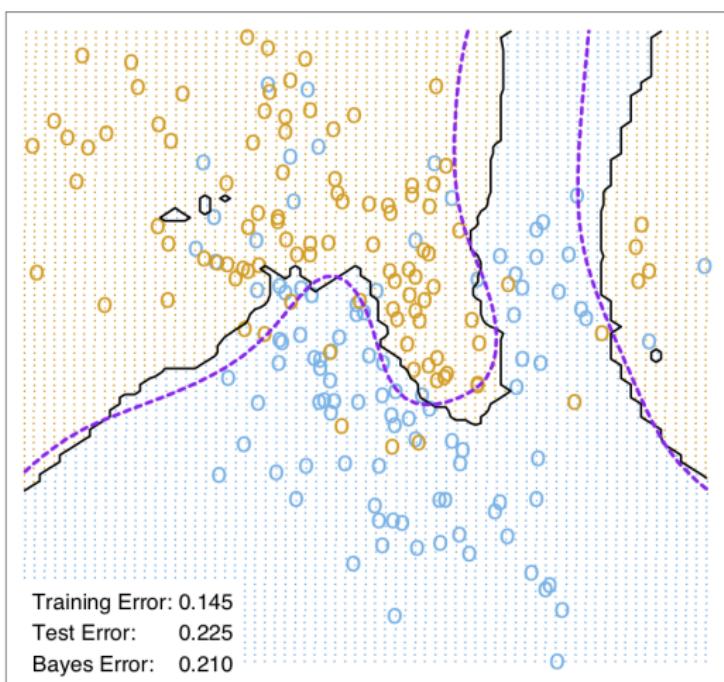
Fronteras de KNN

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



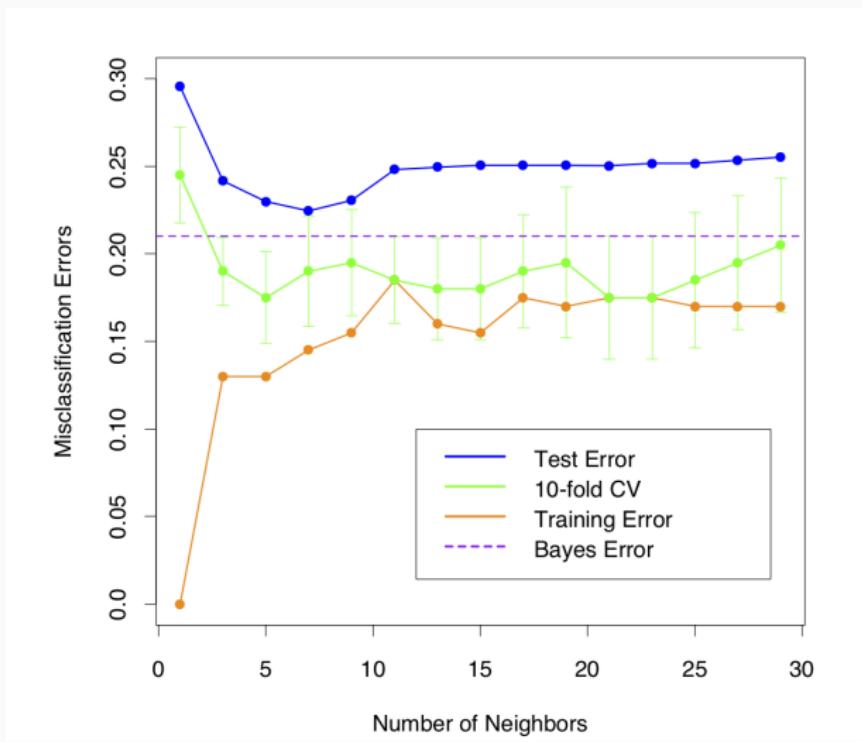
KNN versus K

- 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 : $x_{(1)}, x_{(2)}, \dots, x_{(K)}$ y sus respectivas etiquetas $y_{(1)}, y_{(2)}, \dots, y_{(K)}$.
 2. Predecir

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

Complejidad como Función de K

- Las predicciones que KNN para regresión hace sobre el conjunto de entrenamiento se pueden escribir como

$$\hat{\mathbf{Y}} = \frac{1}{K} H \mathbf{Y}, \quad (7)$$

donde $H_{ij} = 1$ si y solo si $x^{(j)} \in N(x^{(i)})$. • De acá se sigue que la dimensionalidad efectiva (complejidad) de KNN

$$\text{tr}\left(\frac{1}{K} H\right) = n/K, \quad (8)$$

es proporcional a n e inversamente proporcional a K tal como ocurre en problemas de clasificación.

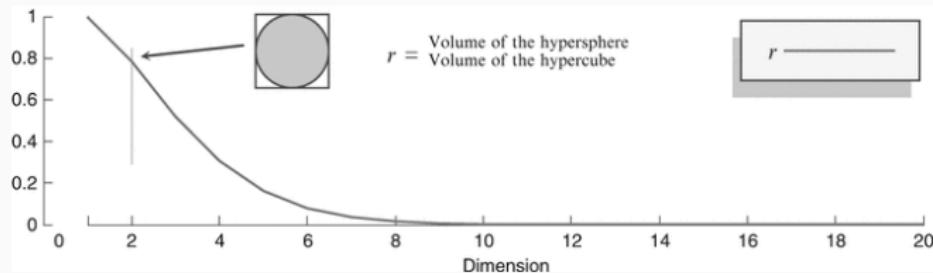
- Conceptualmente muy simple. Es muy conocido y utilizado en la práctica.
- El entrenamiento es muy eficiente en términos de tiempo de cómputo: sólo consiste en almacenar (memorizar) los ejemplos.
- Funciona bastante bien cuando la dimensionalidad es baja.
- Es un método naturalmente no-paramétrico (se 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.

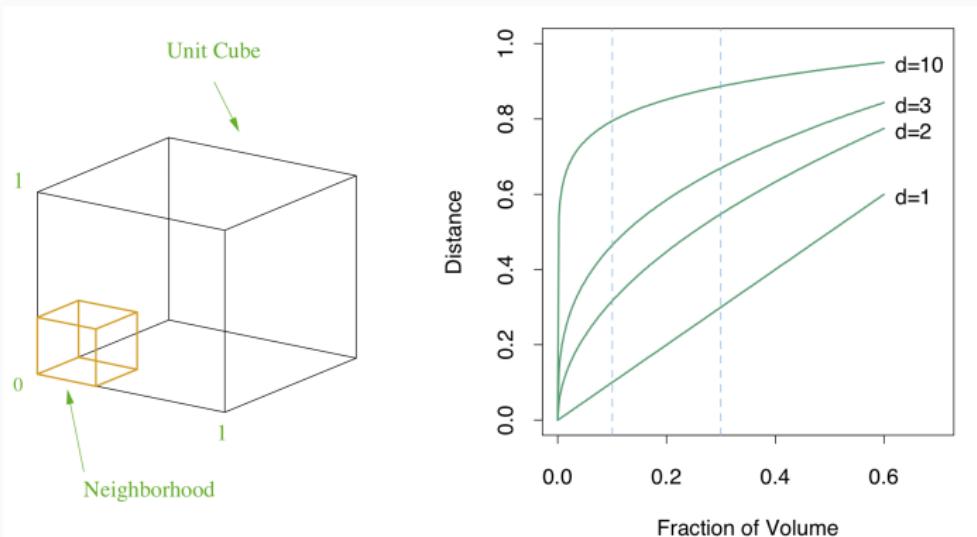
Maldición de la Dimensionalidad

- La probabilidad de que encontrar vecinos **cerca** de x decrece exponencialmente rápido en la dimensionalidad d del espacio de características.



Maldición de la Dimensionalidad

- La probabilidad de que encontrar vecinos **cerca** de x decrece exponencialmente rápido en la dimensionalidad d del espacio de características.



Tarea

- Demuestre que la distancia de un punto x a su vecino más cercano de entre n ejemplos está dada por

$$\tilde{R} = (1 - 1/2^{1/n})^{1/d}, \quad (9)$$

(asuma que los ejemplos se distribuyen de modo uniforme en un hiper-cuadrado unitario en torno al punto).

Algunas Propiedades

This Week's Citation Classic

Cover T M & Hart P E. Nearest neighbor pattern classification.
IEEE Trans. Inform. Theory IT-13:21-7, 1967.
[Dept. Electrical Engineering, Stanford Univ., Stanford,
and Stanford Res. Inst., Menlo Park, CA]

CC/NUMBER 13
MARCH 29, 1982

Thomas M. Cover
Departments of Statistics and
Electrical Engineering
Stanford University
Stanford, CA 94305

March 5, 1982

"Early in 1966 when I first began teaching at Stanford, a student, Peter Hart, walked into my office with an interesting problem. He said that Charles Cole and he were using a pattern classification scheme which, for lack of a better word, they described as the nearest neighbor procedure. This scheme assigned to an as yet unclassified observation the classification of the nearest neighbor. Were there any good theoretical properties of this procedure? Of course the motivation for such a classification rule must go back to prehistoric times. The idea is that 'things that look alike must be alike.'

"The problem seemed extremely inviting from a theoretical point of view. We began meeting for two or three hours every afternoon in an attempt to find some distribution-free properties of this classification rule. By distribution-free, I mean properties that are true regardless of the underlying joint distribution of the categories and observations. Obviously, we could not hope to prove that a procedure always has, for example, a zero probability of error, because

is less than twice the Bayes risk for all reasonable distributions and for any number of categories. Thus ancient man was proved right—"things that look alike are alike"—with a probability of error that is no worse than twice the probability of error of the most sophisticated modern day statistician using the same information. Moreover, we were soon able to prove that our bound was the best possible. So the search was over.

"The simplicity of the bound and the sweeping generality of the statement, combined with the obvious simplicity of the nearest neighbor rule itself, have caused this result to be used by others, thus accounting for the high number of citations. Since the properties of the nearest neighbor rule can be easily remembered, the bound yields a benchmark for other more sophisticated data analysis procedures, which sometimes actually perform worse than the nearest neighbor rule. This is probably due to the fact that more ambitious rules have too many parameters for the data set.

"It should be mentioned that we had to exclude a certain technical set of joint distributions from the proof of our theorem. The attendant measure-theoretic difficulties in eliminating the so-called singular distributions almost delayed the publication of our paper. It was wise that we did not hold up publication, because the theorem was not proved in total generality until ten years later in Charles Stone's 1977 paper in the *Annals of Statistics*.¹ The result remains the same, but now it applies to all possible probability distributions."

1. Stone C J. Consistent nonparametric regression. *Ann. Statist.* 5:595-645, 1977.

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

$$m = \arg \max_{j \in \{1, \dots, K\}} p(y = c_j | x) \quad (10)$$

- El error de Bayes es entonces

$$B^*(x) = P(\text{Bayes se equivoque} | x) = 1 - p(c_m) \quad (11)$$

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

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.

$$P_n(e|x) = P(\text{1NN se equivoca} \mid x) \quad (12)$$

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

- Los errores asintóticos del clasificador se definen como

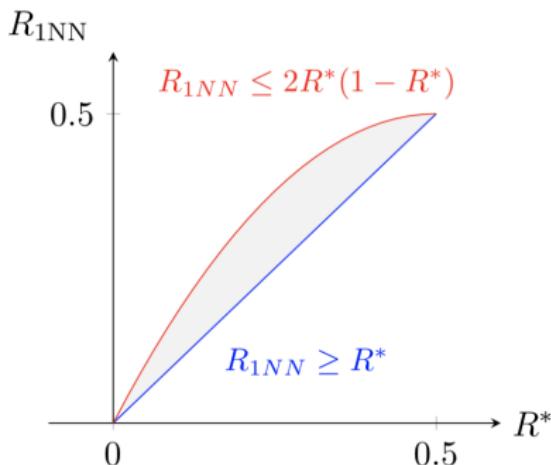
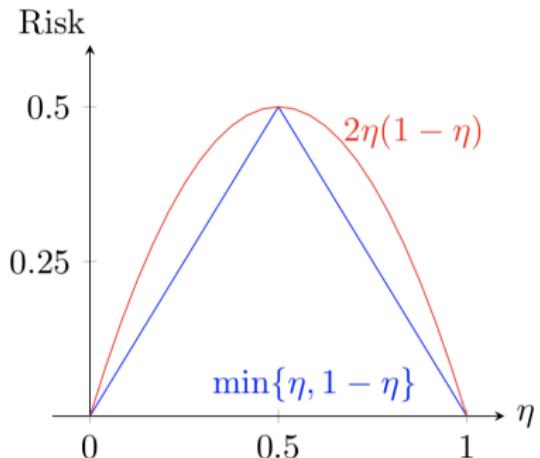
$$P(e|x) = \lim_{n \rightarrow \infty} P_n(e|x) \quad (13)$$

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

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

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

- El punto de partida es esta simple observación

$$P_n(e|x) = 1 - \sum_k P_n(\bar{e}|x, y_{(1)} = c_k) P_n(y_{(1)} = c_k|x) , \quad (14)$$

que nos lleva a

$$P(e|x) = 1 - \sum_k P^2(c_k|x) \quad (15)$$

$$P(e) = \sum_x \left(1 - \sum_k P^2(c_k|x) \right) p(x) , \quad (16)$$

Error Asintótico de 1NN

- Si denotamos por j^* la clase que prefiere Bayes, obtenemos (invocando Cauchy-Swartz) que

$$\begin{aligned} P(e|x) &= 1 - P^2(c_{j^*}|x) - \sum_{k \neq j^*} P^2(c_k|x) \\ &\leq 1 - (1 - B^*(x))^2 - B^{*2}(x) \\ &\leq 2B^*(x) - B^{*2}(x) \end{aligned} \tag{17}$$

de modo que

$$\begin{aligned} P(e) &= 2 \sum_x B^*(x)p(x) - \sum_x B^{*2}(x)p(x) \\ &= 2B^* - \sum_x B^{*2}(x)p(x). \end{aligned} \tag{18}$$

Error Asintótico de 1NN

- Si ahora observamos que

$$\text{Var}(B^*(x)) \geq 0 \Rightarrow \sum_x B^{*2}(x)p(x) > \left(\sum_x B^*(x)p(x) \right)^2, \quad (19)$$

y reemplazamos en nuestra última desigualdad

$$P(e) \leq 2B^* - \sum_x B^{*2}(x)p(x). \quad (20)$$

obtenemos la cota superior del Teorema:

$$P(e) < 2B^* - B^{*2} \quad (21)$$

Error Asintótico de KNN (con $K > 1$)

- Definiendo

$$P_n^{(K)}(e|x) = P(\text{KNN se equivoca} \mid x) \quad (22)$$

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

- Para K impar, es posible mostrar que existe una función concava C_K tal que

$$P^{(K)}(e|x) = \lim_{n \rightarrow \infty} P_n^{(K)}(e|x) \leq C_K(B^*(x)) \quad (23)$$

- Usando Jensen

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

- Además,

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

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 : $x_{(1)}, x_{(2)}, \dots, x_{(K)}$ y las respectivas distancias $m_{(1)}, m_{(2)}, \dots, m_{(K)}$.
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

$$r(c_j) = \sum_{i=1}^K w_i I(c(x_{(i)}) = c_j). \quad (24)$$

4. Elegir la clase más popular²:

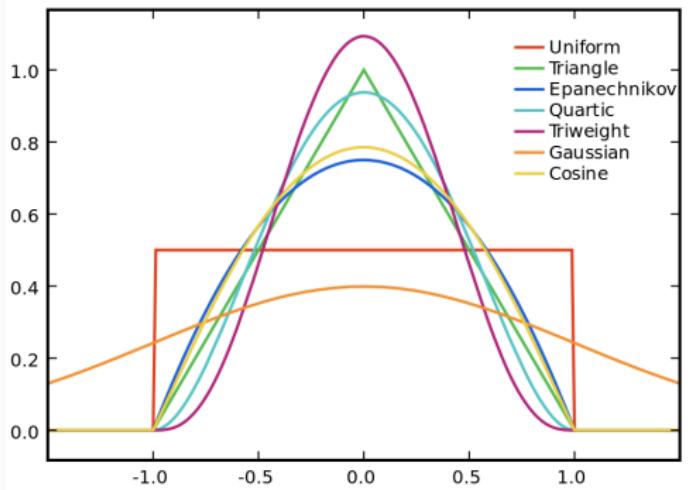
$$f(x) = \arg \max_{c_i} r(c_i) \quad (25)$$

²los 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 :

$x_{(1)}, x_{(2)}, \dots, x_{(K)}, x_{(K+1)}$ y las respectivas distancias
 $m_{(1)}, m_{(2)}, \dots, m_{(K)}, m_{(K+1)}$.

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

$$r(c_j) = \sum_{i=1}^K w_i I(c(x_{(i)}) = c_j). \quad (26)$$

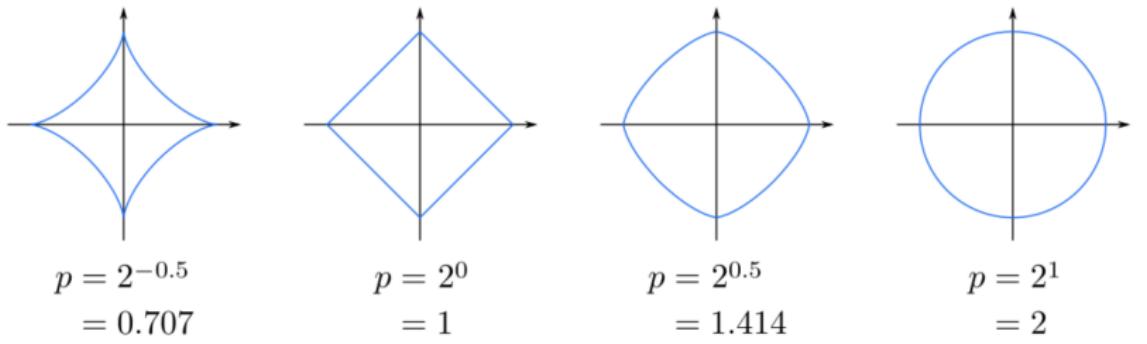
4. Elegir la clase más popular:

$$f(x) = \arg \max_{c_i} r(c_i) \quad (27)$$

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

$$m(a, b) = \left(\sum_i |a_i - b_i|^p \right)^{1/p}. \quad (28)$$



KNN con Métricas Adaptativas

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

$$m(a, b) = (a - b)^T M(a - b), \quad (29)$$

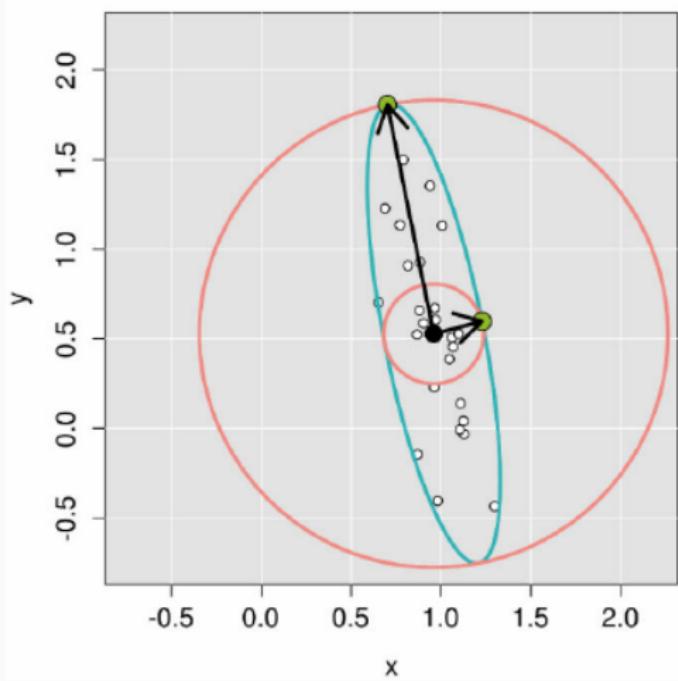
donde $M \in \mathbb{R}^{d \times d}$ es una matriz que se determina con algún criterio a partir del conjunto de entrenamiento.

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

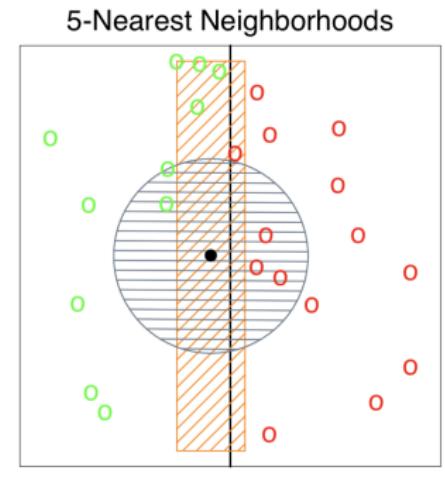
$$\Sigma \approx \mathbb{E}(xx^T) - \mathbb{E}(x)\mathbb{E}(x)^T. \quad (30)$$

la métrica se denomina **Métrica de Mahalanobis**.

Métrica de Mahalanobis



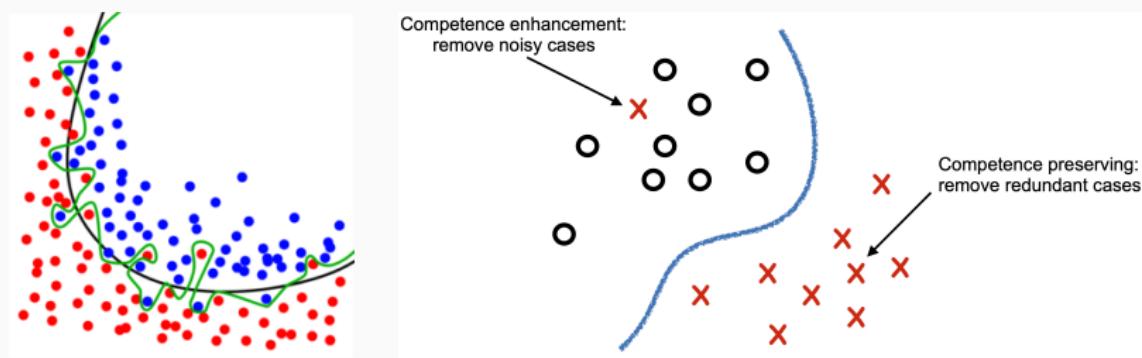
- La matriz M puede ser una matriz de reducción de dimensionalidad ($P^T P$ con P un proyector), que “pese más” las direcciones relevantes del espacio característico.
- Ambas técnicas pueden aplicarse localmente, aumentando significativamente el costo computacional del clasificador.



Edición & Condensación

Ideas Clave

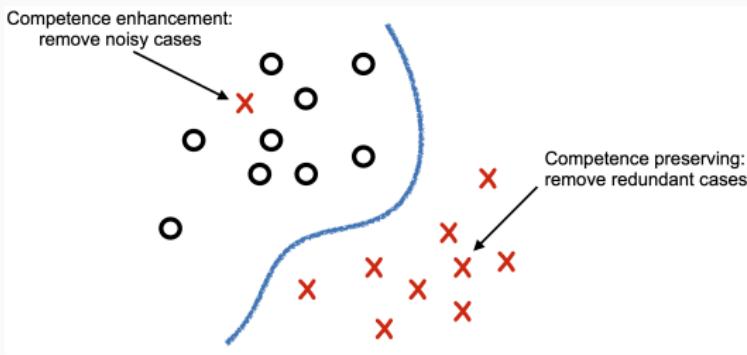
- Dos defectos importantes de KNN son: (i) su alta sensibilidad a irregularidades en el conjunto de entrenamiento (e.g. noise, outliers) y (ii) el alto costo computacional en tiempo de decisión.
- Gran parte de la investigación asociada a este método se concentra en aliviar esos problemas.



- Los métodos de **Edición** intentan mejorar la capacidad de generalización del estimador, eliminando ejemplos irregulares (noise) de la “memoria” del modelo. Ejemplos: Edición de Wilson, Edición Múltiple.
- Los métodos de **Condensación** intentan eliminar la mayor cantidad de ejemplos de la “memoria” del modelo, preservando la “forma” del estimador (e.g. las fronteras de clasificación). Ejemplos: Condensación de Hart (CNN), Fast CNN, Delaunay, Gabriel, RNG.
- Métodos Híbridos. Ejemplos: IFC.
- La mayoría de estos métodos están diseñados para clasificación.

Edición de Wilson

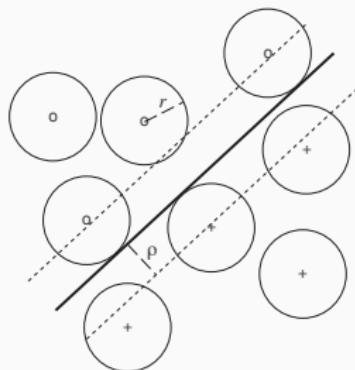
- Hipótesis: los ejemplos clasificados incorrectamente por KNN reducen su consistencia. Objetivo: Sustituir el conjunto de entrenamiento S por una versión más robusta M .
1. Entrenar KNN con S .
 2. Clasificar S e incluir todos los ejemplos mal clasificados en F .
 3. $M = S - F$.



Edición de Wilson - Variantes

- Variante 1:

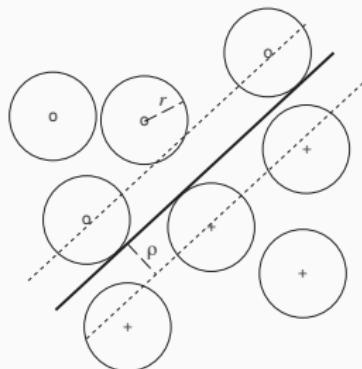
1. $M = S$:
2. Por cada ejemplo en $x \in S$:
 - 2.1 Entrenar KNN con $E - \{x\}$ y clasificar x .
 - 2.2 Si x está mal clasificado, $M = M - \{x\}$.



Edición de Wilson - Variantes

- Variante 2:

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



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

$$P_1^E(e|x) \leq \frac{P(e|x)}{2(1 - P(e|x))} \leq P(e|x), \quad (31)$$

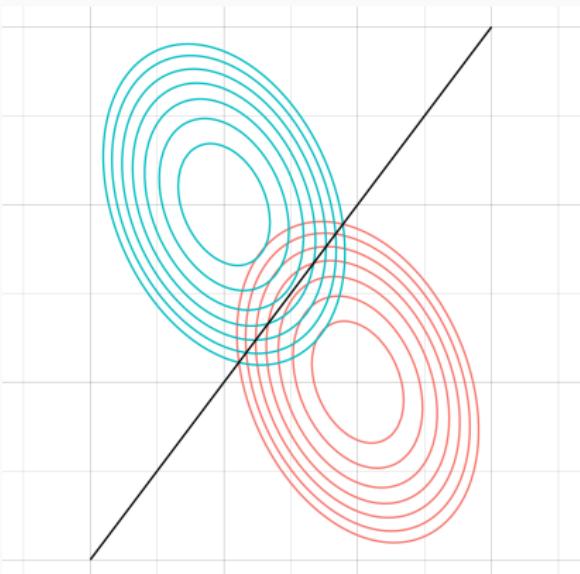
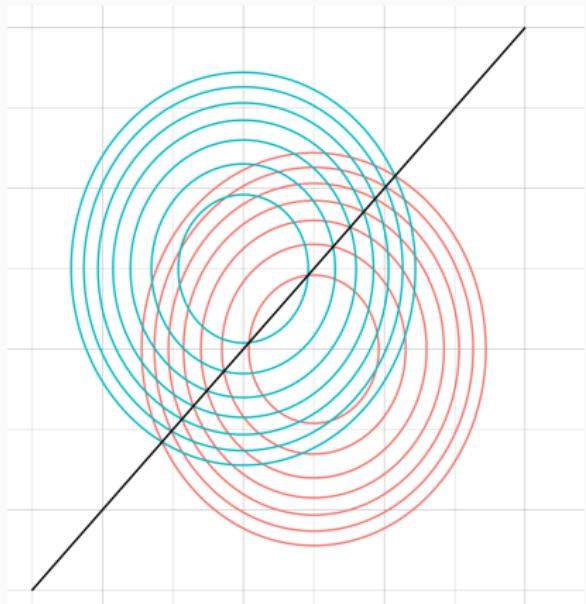
de modo que el método editado con $K = 1$, mejora sobre 1NN básico

$$P_1^E(e) \leq P(e), \quad (32)$$

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

$$P_1^E(e) \approx \frac{P(e)}{2} \approx B^*, \quad (33)$$

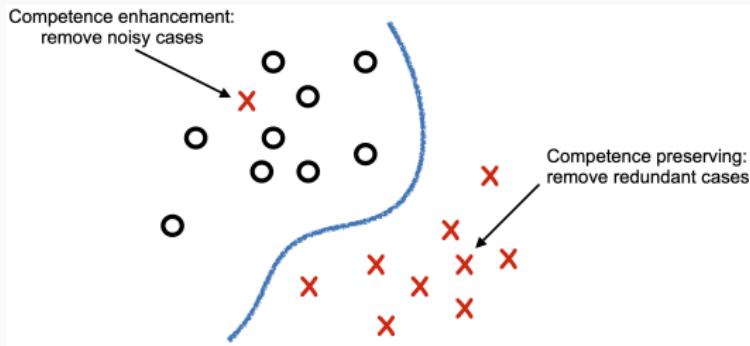
Edición de Wilson



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

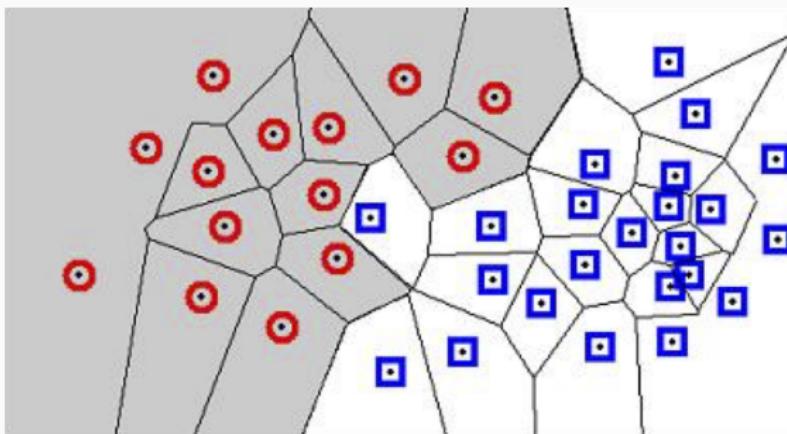
Condesación

- Objetivo: Eliminar puntos del conjunto de entrenamiento preservando la forma del estimador (típicamente la frontera).



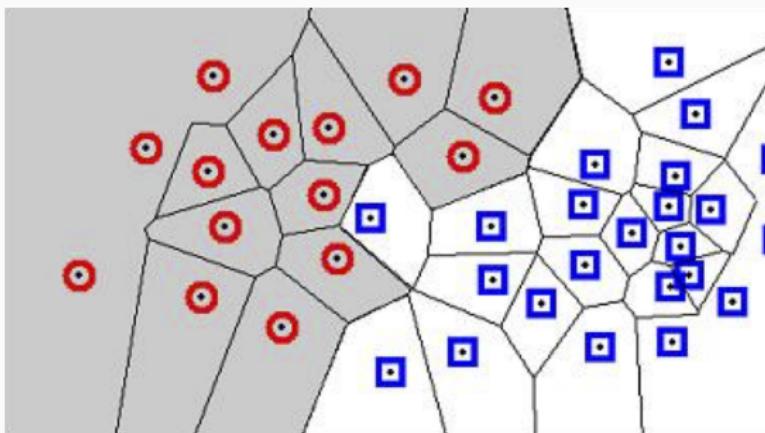
Condesación de Delaunay

- Una forma natural de identificar puntos superfluos en clasificación es buscar aquellos que no intervienen en la definición de la frontera.



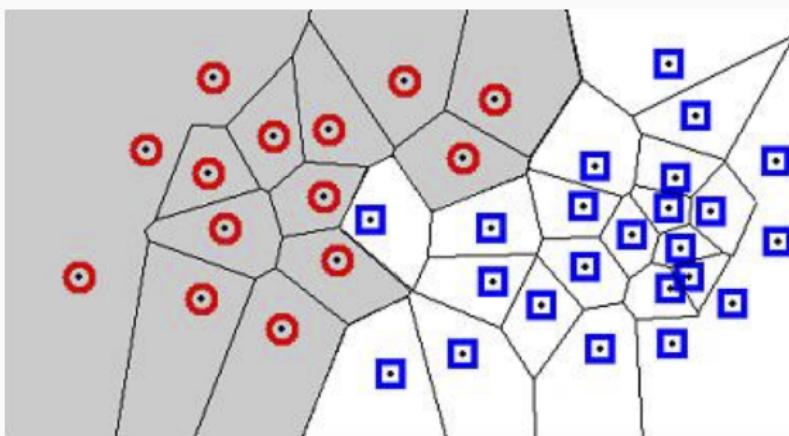
Condesación de Delaunay

- Si consideramos las celdas de Voronoi correspondientes al conjunto de entrenamiento S , la frontera de decisión de este método es la unión de las aristas en que se encuentran celdas de clases diferentes. La idea natural es **remover todos los puntos que no soporten aristas de frontera**.



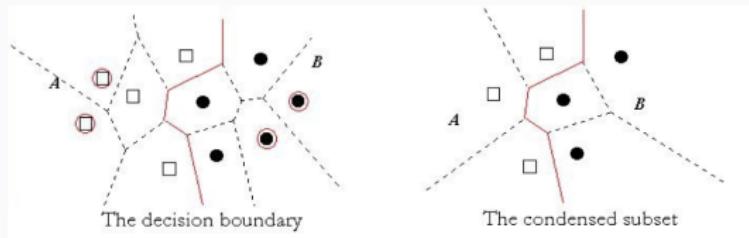
Condesación de Delaunay

- Más formalmente, el **Grafo de Delaunay** une dos puntos de S si y sólo si esos puntos comparten una arista en el diagrama de Voronoi.
- Un dato se denomina **punto de frontera** cuando tiene como vecino un punto de clase diferente y en otro caso como **punto interno**.
- La condesación de Delaunay elimina todos los puntos internos.



Condesación de Delaunay

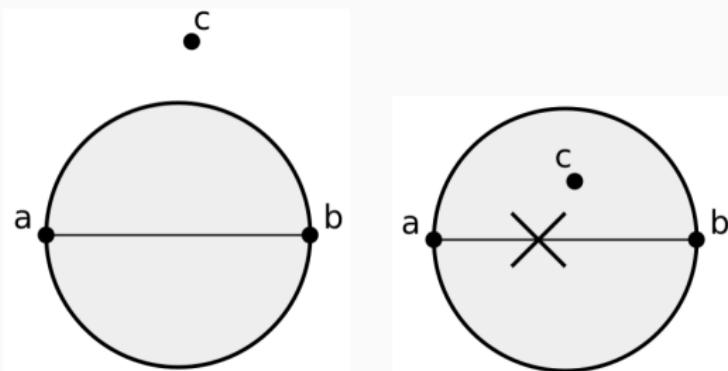
- Este método claramente garantiza la preservación de la frontera.
- Puede demostrarse además que es óptimo (genera la máxima reducción posible de S): un punto de S puede eliminarse sin alterar la frontera sólo si es un nodo interno de Delaunay.



- Lamentablemente, los mejores algoritmos conocidos para construir el Grafo de Delaunay en \mathbb{R}^d tienen costo $\mathcal{O}(d^3 n^{d//2} \log n)$.
- Se conoce un algoritmo $\mathcal{O}(n \log n)$ para el plano y es un problema abierto diseñar algoritmo eficiente para d arbitrario. **Se cree que esto es imposible.**

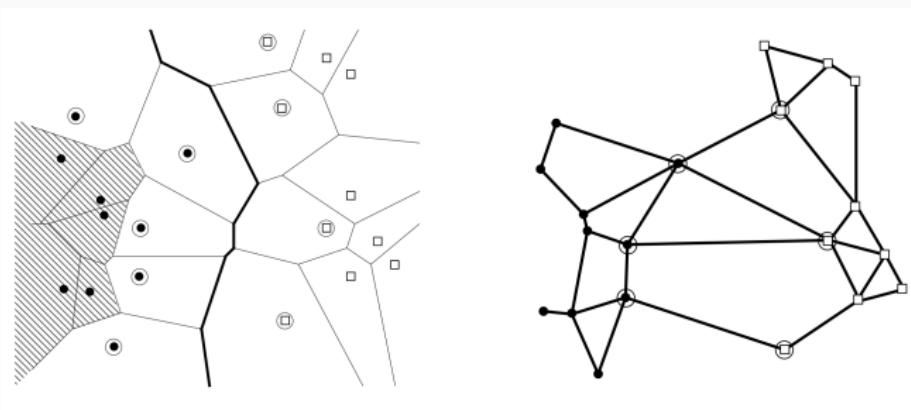
Condesación de Gabriel

- Idea: relajar la definición de Delaunay.
- **Grafo de Gabriel:** dos puntos son mutuamente vecinos si la circunferencia de menor radio que contiene ambos puntos, no contiene otros puntos.
- Se sabe que el Grafo de Gabriel es un sub-grafo del Grafo de Delaunay.



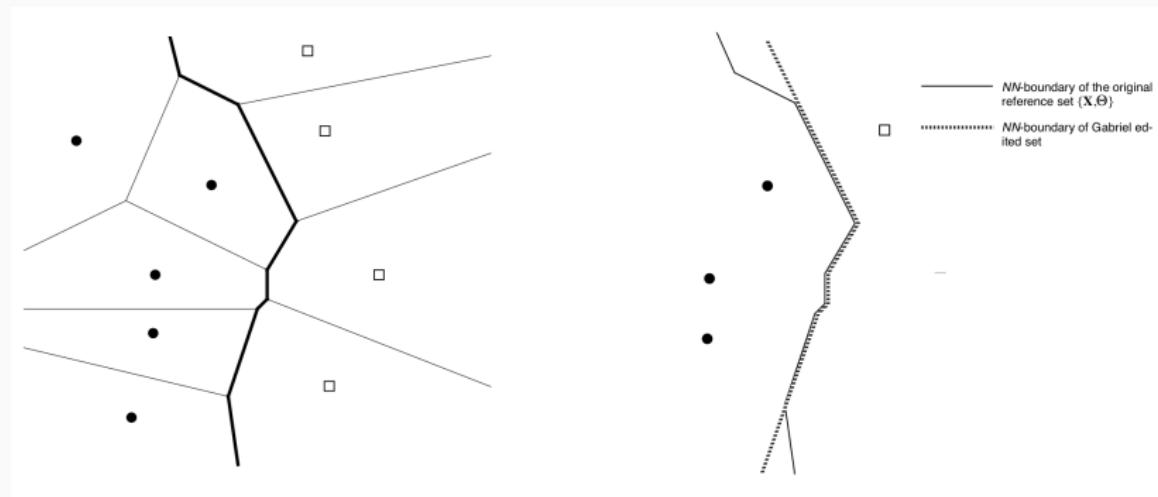
Condesación de Gabriel

- La nueva condesación consiste en sustituir el Grafo de Delaunay por la relajación de Gabriel.
- Un dato se denomina **punto de frontera** cuando tiene como vecino un punto de clase diferente en el grafo de y en otro caso como **punto interno**.
- La condesación elimina todos los puntos internos.



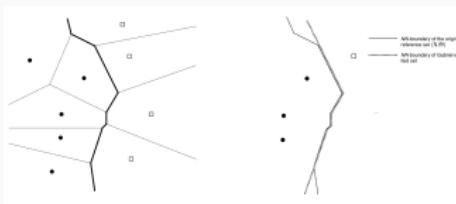
Condesación de Gabriel

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



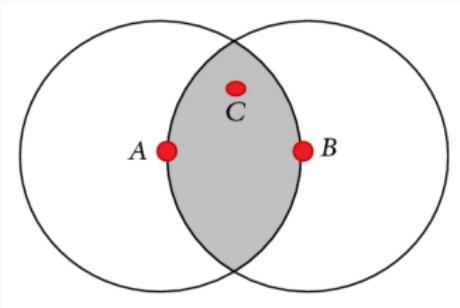
Condesación de Gabriel

- El método NO garantiza la preservación de la frontera, pero los cambios ocurren fuera de la envoltura convexa del conjunto de puntos.
- El método elimina siempre al menos tantos puntos como Delaunay (dos vecinos de Gabriel son vecinos de Delaunay).
- Es posible ejecutarlo de modo relativamente eficiente: $\mathcal{O}(dn^3)$ (exacto) ó $\mathcal{O}(dn^2)$ (heurístico).



Condesación vía RNG (Relative Neighbour Graph)

- Definición de vecinos: Dos puntos a, b son mutuamente vecinos en el grafo RNG 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).



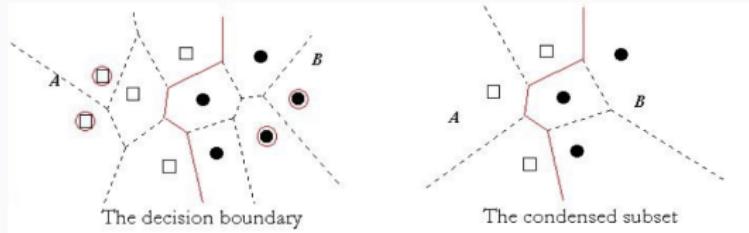
- Cambia significativamente la frontera.
- Es posible demostrar que el clasificador obtenido no es consistente.

Condesación de Hart (CNN)

- Idea: relajar el objetivo de preservar la frontera,
garantizando consistencia (clasificación idéntica a la que se tenía antes de
condensar) sobre el conjunto de entrenamiento S (en vez de \mathbb{X}).

Condesación de Hart (CNN)

- Idea: relajar el objetivo de preservar la frontera
- En vez de garantizar consistencia sobre todo \mathbb{X} , hacerlo sobre el conjunto de entrenamiento S .



(consistencia en S : clasificación de S es idéntica a la que se tenía antes de condensar).

Condesación de Hart (CNN)

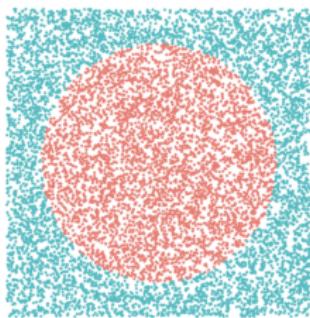
- **Algoritmo:** Inicializar M (conjunto de puntos a preservar) con un sólo ejemplo y luego iterar:
 1. Entrenar KNN sobre M , clasificar $S - M$ y pasar todos los puntos mal clasificados a M .
 2. Si no hay cambios en M , terminar.
- Costo $\mathcal{O}(dn^3)$ en el peor caso.
- Sensible al orden de presentación de los ejemplos.

- Propuesto por Angiulli³ hacia 2005.
- Inicializar M con los centroides de cada clase y luego iterar:
 1. Por cada elemento $p \in M$, intentar elegir un punto $q \in S$ que tenga p como vecino más cercano en M y que sea de clase diferente de p .
 2. Si lo anterior es posible, agregar q a M y seguir. Sino, terminar.
- Costo $\mathcal{O}(dn^2)$ en el peor caso.
- Garantiza consistencia.
- Insensible al orden de presentación de los ejemplos.

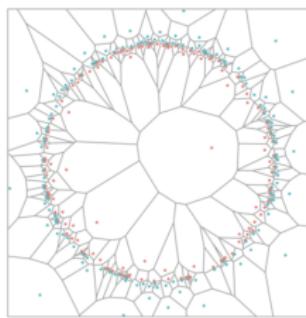
³Fabrizio Angiulli. “Fast condensed nearest neighbor rule”, International conference on Machine learning, 2005.

Fast CNN

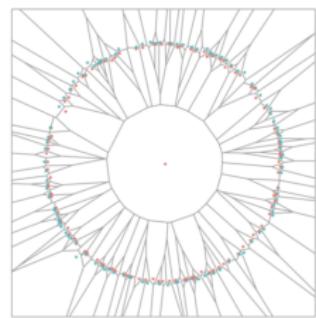
- En vez de garantizar consistencia sobre todo \mathbb{X} , hacerlo sobre el conjunto de entrenamiento S .



(a) Dataset P (10^4 points)



(b) CNN (281 points)



(c) FCNN (222 points)

- Muchos otros métodos. Algunos populares: DROP1, DROP5, ICF, etc.

***k*-Nearest Neighbour Classifiers**
2nd Edition (with Python examples)

Pádraig Cunningham
School of Computer Science
University College Dublin
padraig.cunningham@ucd.ie

Sarah Jane Delany
School of Computing
Technological University Dublin
sarahjane.delany@tudublin.ie

Abstract

Perhaps the most straightforward classifier in the arsenal of machine learning techniques is the Nearest Neighbour Classifier – classification is achieved by identifying the nearest neighbours to a query example and using those neighbours to determine the class of the query. This approach to classification is of particular importance because issues of poor run-time performance is not such a problem these days with the computational power that is available. This paper presents an overview of techniques for Nearest Neighbour classification focusing on: mechanisms for assessing similarity (distance), computational issues in identifying nearest neighbours and mechanisms for reducing the dimension of the data.

This paper is the second edition of a paper previously published as a technical report [14]. Sections on similarity measures for time-series, retrieval speed-up and intrinsic dimensionality have been added. An Appendix is included providing access to Python code for the key methods.

Literatura

- Distinguir métodos que garantizan consistencia de muchos otros que no.

The screenshot shows a research paper on arXiv.org. The header includes the arXiv logo, a search bar, and links for Help and Advanced Search. The main content area has a red header bar with the arXiv categories: arXiv.org > cs > arXiv:1904.12142. Below this, the title 'Computer Science > Computational Geometry' is displayed, along with the submission date 'Submitted on 27 Apr 2019'. The main title of the paper is 'Guarantees on Nearest-Neighbor Condensation heuristics'. The authors are listed as 'Alejandro Flores-Velasco, David Mount'. The abstract begins with: 'The problem of nearest-neighbor (NN) condensation aims to reduce the size of a training set of a nearest-neighbor classifier while maintaining its classification accuracy. Although many condensation techniques have been proposed, few bounds have been proved on the amount of reduction achieved. In this paper, we present one of the first theoretical results for practical NN condensation algorithms. We propose two condensation algorithms, called RSS and VSS, along with provable upper-bounds on the size of their selected subsets. Additionally, we shed light on the selection size of two other state-of-the-art algorithms, called MSS and FCNN, and compare them to the new algorithms.' The footer contains standard arXiv subject and citation information.

arXiv.org > cs > arXiv:1904.12142

Computer Science > Computational Geometry

[Submitted on 27 Apr 2019]

Guarantees on Nearest-Neighbor Condensation heuristics

Alejandro Flores-Velasco, David Mount

The problem of nearest-neighbor (NN) condensation aims to reduce the size of a training set of a nearest-neighbor classifier while maintaining its classification accuracy. Although many condensation techniques have been proposed, few bounds have been proved on the amount of reduction achieved. In this paper, we present one of the first theoretical results for practical NN condensation algorithms. We propose two condensation algorithms, called RSS and VSS, along with provable upper-bounds on the size of their selected subsets. Additionally, we shed light on the selection size of two other state-of-the-art algorithms, called MSS and FCNN, and compare them to the new algorithms.

Subjects: Computational Geometry (cs.CG)

Cite as: [arXiv:1904.12142 \[cs.CG\]](https://arxiv.org/abs/1904.12142)
(or [arXiv:1904.12142v1](https://arxiv.org/abs/1904.12142v1) [cs.CG] for this version)

- Poco para regresión.

The screenshot shows a research article page. At the top left is the Elsevier logo, which includes a tree icon and the word 'ELSEVIER'. In the center, the journal title 'Neurocomputing' is displayed above 'Volume 251, 16 August 2017, Pages 26-34'. To the right is a small thumbnail image of the journal cover. Below the header, the article title 'An efficient instance selection algorithm for k nearest neighbor regression' is shown. Underneath the title, the authors' names are listed: 'Yunsheng Song ^a✉, Jiye Liang ^a✉, Jing Lu ^b, Xingwang Zhao ^a✉'. There is a 'Show more ▾' link. Below the authors, the DOI is provided: 'https://doi.org/10.1016/j.neucom.2017.04.018'. To the right of the DOI is a 'Get rights and content' button. A horizontal line separates this section from the abstract. The abstract begins with the heading 'Abstract' and a descriptive paragraph about the k-Nearest Neighbor algorithm (kNN). The text reads: 'The k-Nearest Neighbor algorithm(kNN) is an algorithm that is very simple to understand for classification or regression. It is also a lazy algorithm that does not use the training data points to do any generalization, in other words, it keeps all the training data during the testing phase. Thus, the population size becomes a major concern for kNN, since large population size may result in slow execution speed and large memory requirements. To solve this problem, many efforts have'.

Neurocomputing
Volume 251, 16 August 2017, Pages 26-34

An efficient instance selection algorithm for k nearest neighbor regression

Yunsheng Song ^a✉, Jiye Liang ^a✉, Jing Lu ^b, Xingwang Zhao ^a✉
Show more ▾

<https://doi.org/10.1016/j.neucom.2017.04.018> [Get rights and content](#)

Abstract

The k-Nearest Neighbor algorithm(kNN) is an algorithm that is very simple to understand for classification or regression. It is also a lazy algorithm that does not use the training data points to do any generalization, in other words, it keeps all the training data during the testing phase. Thus, the population size becomes a major concern for kNN, since large population size may result in slow execution speed and large memory requirements. To solve this problem, many efforts have