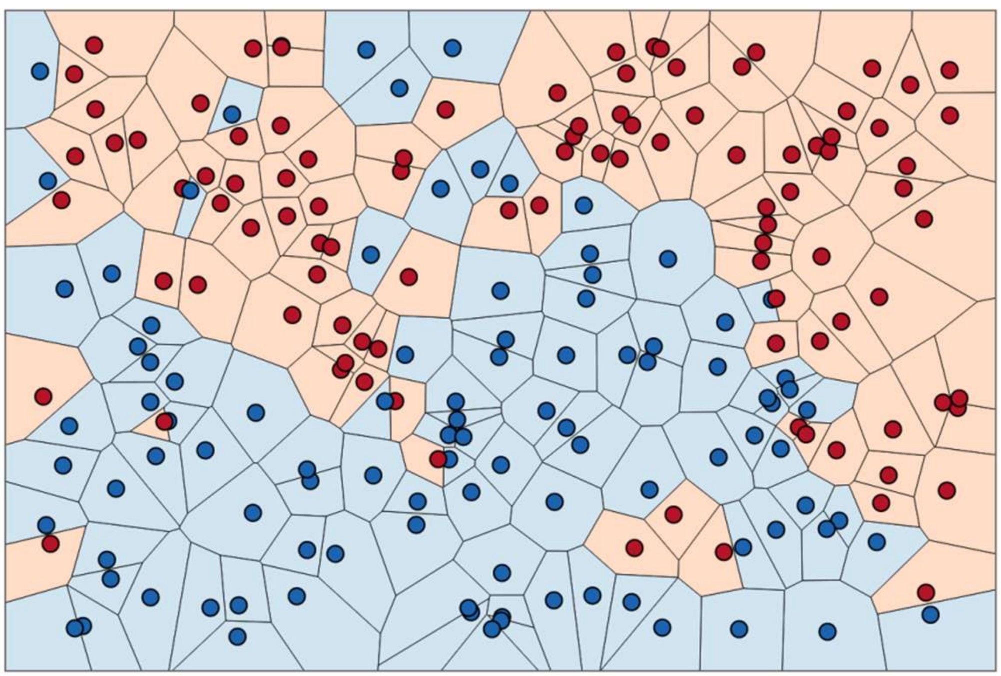
Aprendizaje Supervisado: Clasificación usando K-Vecinos Cercanos.

Agenda

- K-Vecinos Cercanos.
  - Teoría
    - # de Vecinos
    - Funciones de distancia
  - Algoritmo

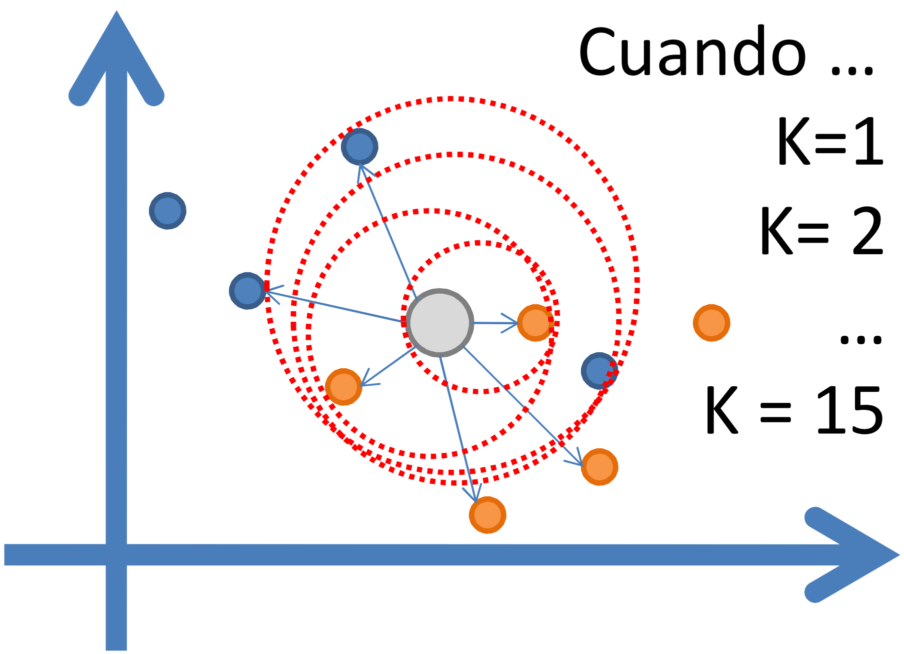
K-Vecinos Cercanos KNN)

- Surge en 1951 (Fix & Hodges, 1951)
- Técnica no-paramétrica
  - Excepto K
- Almacena todos los posibles casos, y clasifica nuevos casos basados en una medida de similitud.
  - *"Aprendizaje Flojo"*



- Otros nombres:
  - Razonamiento basado en casos.
  - Aprendizaje basado en instancias.
- Suposiciones mínimas sobre los datos.
  - Aproximación local.
- Predicciones precisas pero inestables.

KNN a grandes rasgos



$$D(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sqrt{\sum_{q}^{N} \left(x_{iq} - x_{jq}\right)^2}$$

Existen Varios métodos de NN

NN

Dado $\boldsymbol{x}_{new}$ su clase es $y_i^* \rightarrow$
$\underset{\boldsymbol{x}_i \in X}{\operatorname{argmin}} D(\boldsymbol{x}_{new}, \boldsymbol{x}_i)\,, \exists(\boldsymbol{x}_i^*, y_i^*) \in X \times Y$

K-NN

Dado $\boldsymbol{x}_{new}$ su clase es $majority(y_i), \forall x_i \in K$ donde K indica el número de vecinos cercanos.

WK-NN (Weighted)

Dado $\boldsymbol{x}_{new}$ su clase es $majority(\boldsymbol{\beta} y_i), \forall x_i \in K$ donde K indica el número de vecinos cercanos, y $\boldsymbol{\beta}$ es un vector de pesos.

El algoritmo de clasificación usando KNN

Dado $\boldsymbol{x}_{new}$ podemos determinar su clase usando el siguiente algoritmo:

1. Especificar el tamaño de vecindario $\underline{k}$ ($\mathbb{Z}_+$).
2. Usando una $D(\boldsymbol{x}_{new}, \boldsymbol{x}_j)$ seleccionar *k* vecinos más cercanos.
3. Usando la función $mayoria(\cdot)$ determinar la moda entre los *k-nn*.
4. Asignar la clase más popular a $\boldsymbol{x}_{new}$.

Clasificación binaria vía KNN

Dado un conjunto

$$(\bar{x}_i, y_i) \in \Re^N \times \{\pm 1\}$$

Y un nuevo elemento $\boldsymbol{x}_{new}$

$$f(\bar{x})$$

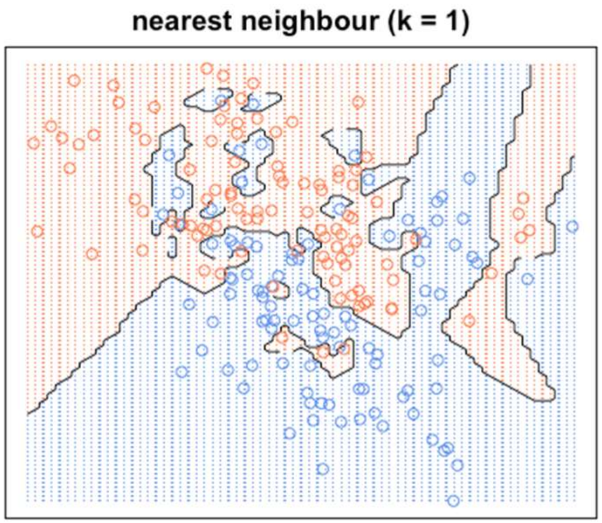
$$\hat{y} = \underset{x_i \in N_k}{\text{majority}}(y_i)$$

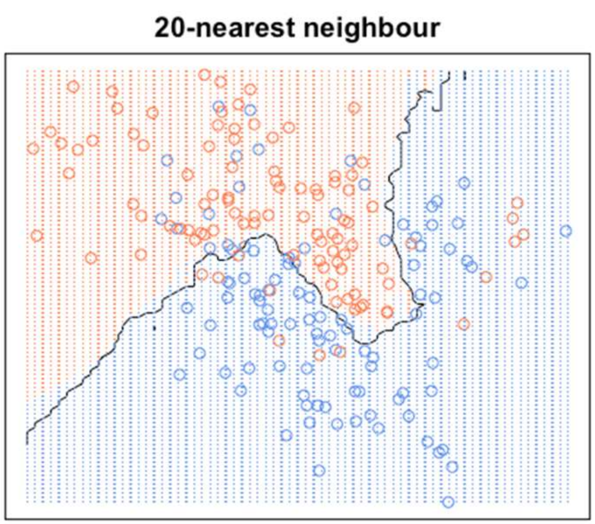
$$D(\boldsymbol{x}_{new}, \boldsymbol{x}_i) | x_i \in N_k$$

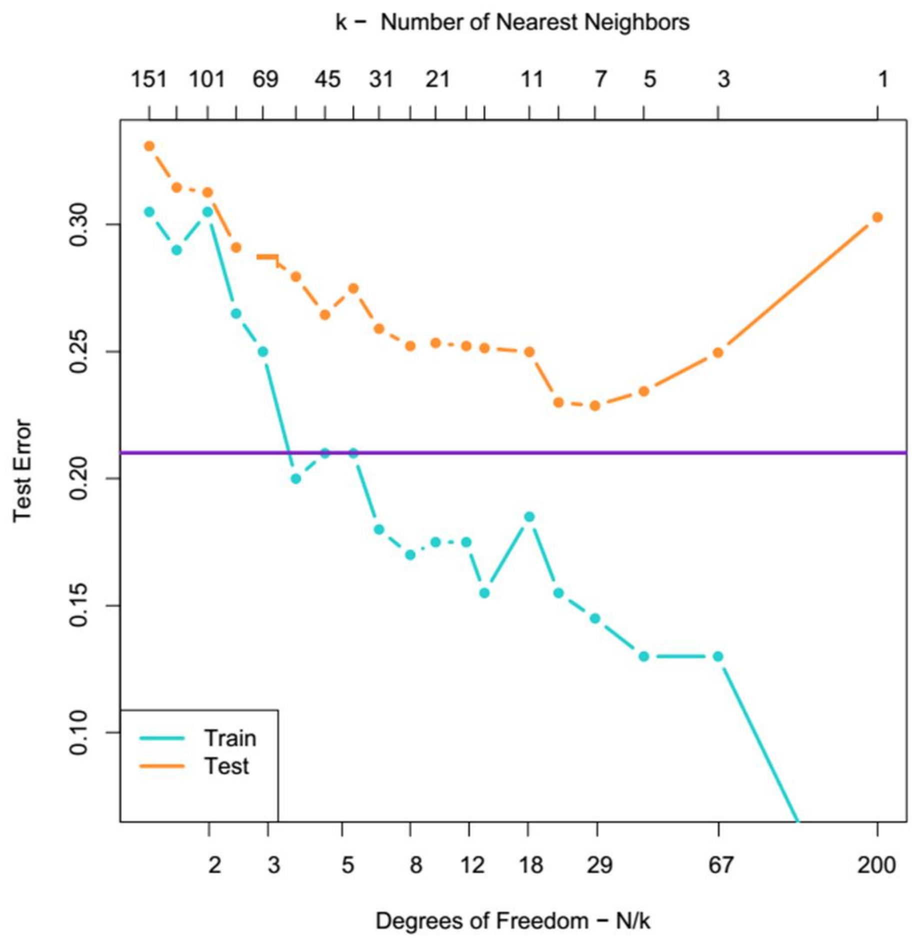
$$\hat{y}_{new}$$

N_k: es el vecindario de x_{new} con sus K vecinos más cercanos.

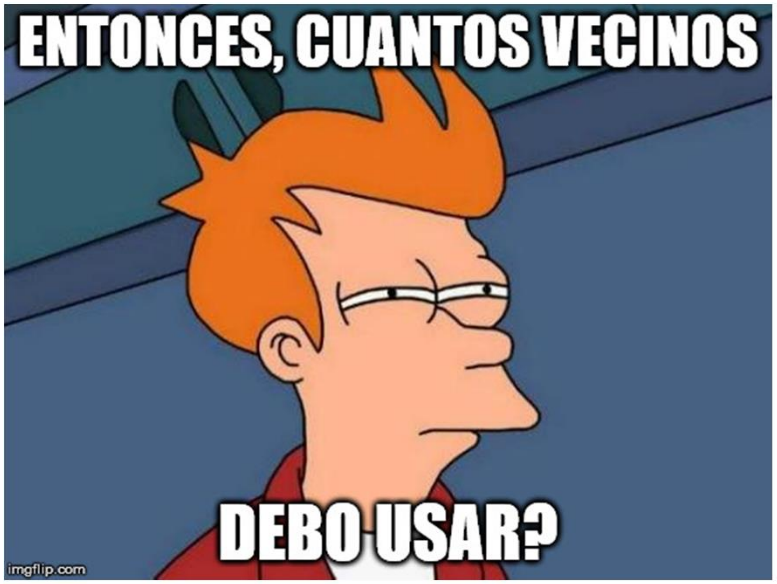
Más acerca del número de Vecinos: K





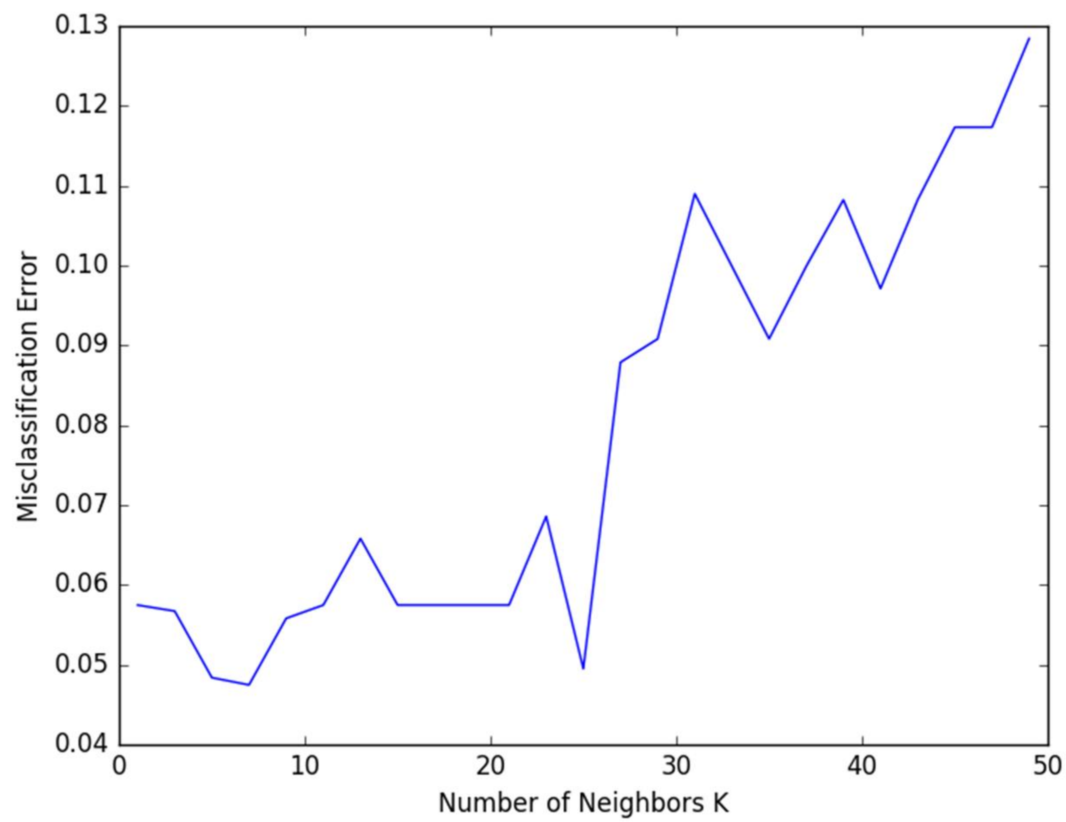


The Elements of Statistical Learning, Capitulo 2.



Reglas de dedo:
- $\sqrt{N}$ donde N es el número de elementos en la muestra
- K=3, K=5, K=7, ...?!?!?

La mejor forma es a través de validación cruzada.



Funciones de Distancia

- Las funciones de distancia *D* funcionan con
  - Variables Nominales/Categóricas
  - Variables Continuas
  - Valores Faltantes
- ¿La clave?
  - Valores pequeños para objetos similares y viceversa.

$$D(\boldsymbol{x}_i, \boldsymbol{x}_j) = \begin{cases} \approx 0 & \boldsymbol{x}_i \sim \boldsymbol{x}_j \\ \infty & \boldsymbol{x}_i \neq \boldsymbol{x}_j \end{cases}$$

Propiedades de las Funciones de Distancia

$$D(\boldsymbol{x}_i, \boldsymbol{x}_j) = \begin{cases} \approx 0 & \boldsymbol{x}_i \sim \boldsymbol{x}_j \\ \\ \infty & \boldsymbol{x}_i \neq \boldsymbol{x}_j \end{cases}$$

- $D(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0$ No-negatividad
- $D(\boldsymbol{x}_i, \boldsymbol{x}_i) = 0 \Leftrightarrow \boldsymbol{x}_i = \boldsymbol{x}_j$ Identidad
- $D(\boldsymbol{x}_i, \boldsymbol{x}_j) = D(\boldsymbol{x}_j, \boldsymbol{x}_i)$ Simetría
- $D(\boldsymbol{x}_i, \boldsymbol{x}_j) + D(\boldsymbol{x}_j, \boldsymbol{x}_k) \geq D(\boldsymbol{x}_i, \boldsymbol{x}_k)$ Subaditividad

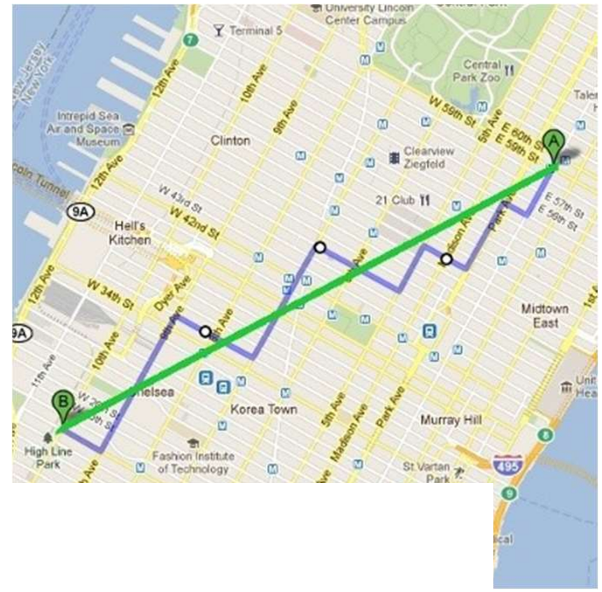
Funciones de Distancia

$$D(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sqrt{\sum_{q=1}^{N} (x_{iq} - x_{jq})^2}$$

Distancia Euclidiana

$$D(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{q=1}^{N} |x_{iq} - x_{jq}|$$

Distancia Manhattan*



$$D(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(\sum_{q=1}^{N} (x_{iq} - x_{jq})^q\right)^{1}$$

$$D_H = \sum_{q=1}^{N} |x_i - x_j|$$

$$x_i = x_j \Rightarrow 0$$
$$x_i \neq x_j \Rightarrow 1$$

Distancia de Hamming

Pregunta: _¿A que tipo de datos podemos aplicar esta distancia?_

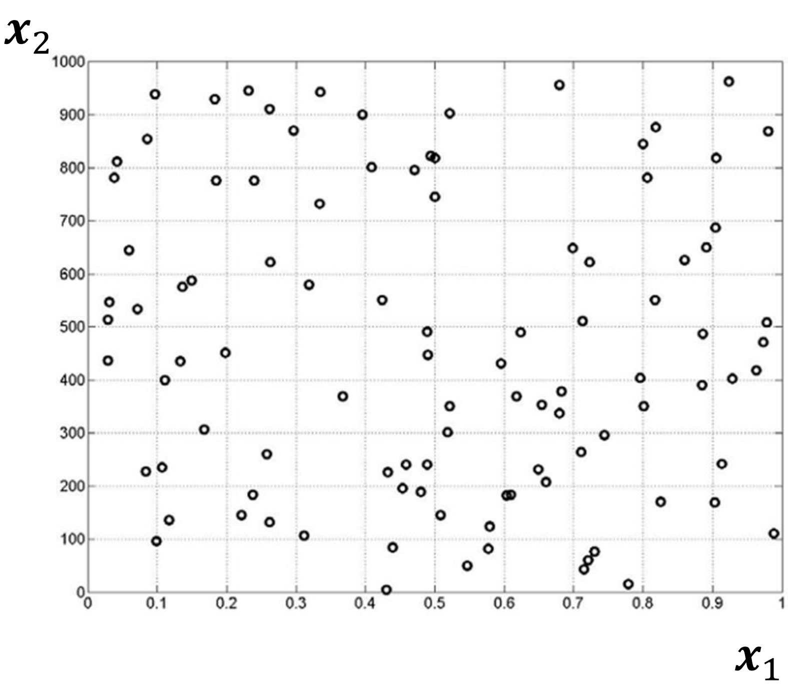
*La suma total de la diferencia entre las coordenadas *x* y *y*.

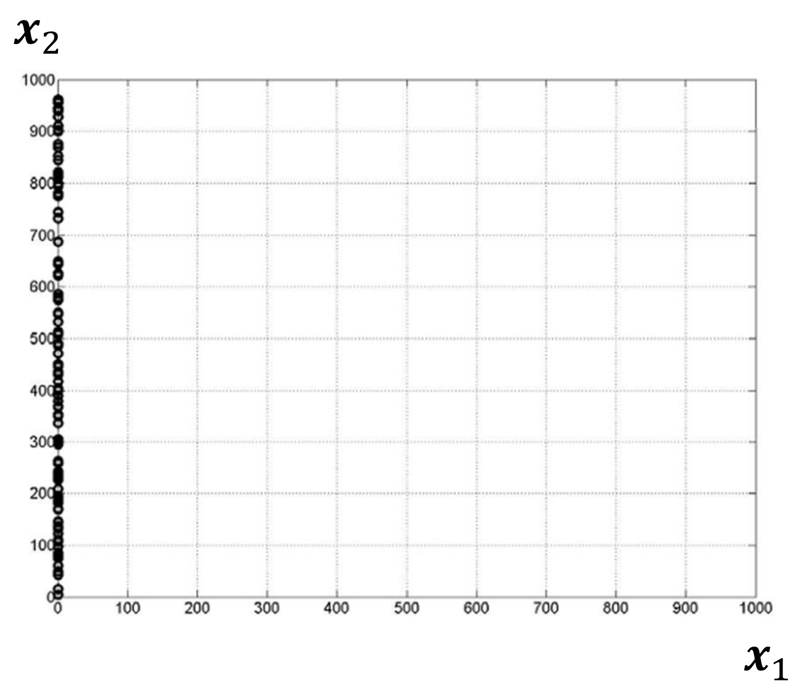
Distancias para Variables Continuas

¿Qué pasaría si las variables son medidas en diferentes escalas?

- $\boldsymbol{x}_1 \in [0,1]$
- $\boldsymbol{x}_2 \in [0,1K]$

¿Podemos anticipar un problema con D_E?





Distancias para Variables Continuas

Entre mayor es la escala de una variable, mayor es su influencia en *D*

La solución: NORMALIZAR LAS VARIABLES

¿Cómo?

Re-escalamiento $$x'^{j}_{i} = \frac{x^{j}_{i} - \min(x^{j})}{\max(x^{j}) - \min(x^{j})}$$

$x'^{j}_{i} \in [0,1]$

Normalización a la media $$x'^{j}_{i} = \frac{x^{j}_{i} - \mu^{j}}{\max(x^{j}) - \min(x^{j})}$$

Estandarización (z-score) $$x'^{j}_{i} = \frac{x^{j}_{i} - \mu^{j}}{\sigma^{j}}$$

Fortalezas y Debilidades de KNN

Fortalezas

- Construye un modelo local por cada instancia.
- Aprender el modelo no tiene costo.
- Buen desempeño w.r.t. otras técnicas de ML.

Debilidades

- No es aprendizaje.
- No hay un modelo global y el conocimiento es ilegible.
- No hay generalización.
- Maldición de la dimensionalidad.
- El ruido y los valores atípicos tienen un impacto negativo en el desempeño.

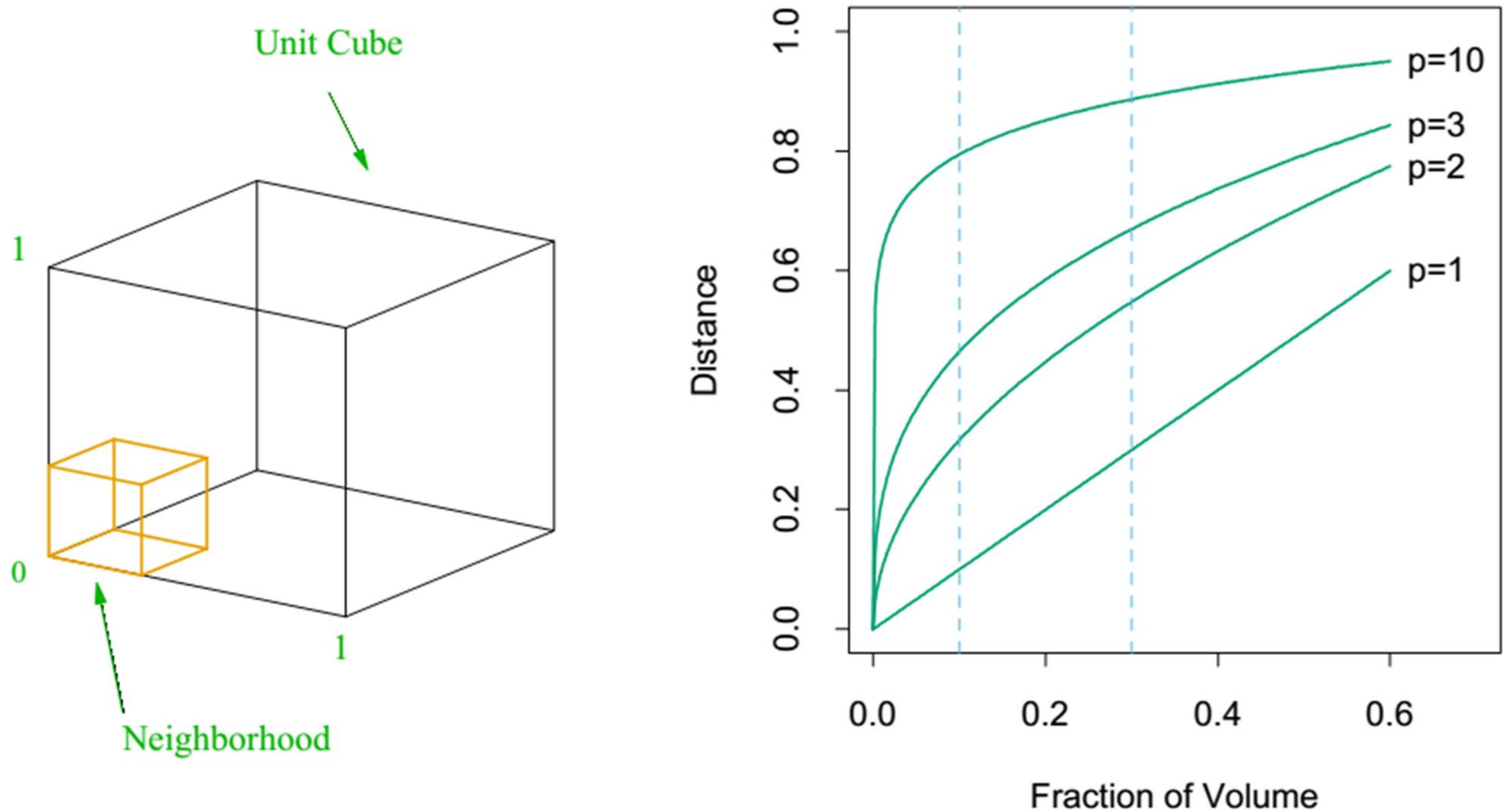


FIGURE 2.6. *The curse of dimensionality is well illustrated by a subcubical neighborhood for uniform data in a unit cube. The figure on the right shows the side-length of the subcube needed to capture a fraction r of the volume of the data, for different dimensions p. In ten dimensions we need to cover 80% of the range of each coordinate to capture 10% of the data.*

The Elements of Statistical Learning, Capitulo 2.