# Leren aan de hand van voorbeelden (instance-Based Learning)

https://www.biostat.wisc.edu/~dpage/cs760/

# Naaste buur bepaling

## Learning task:

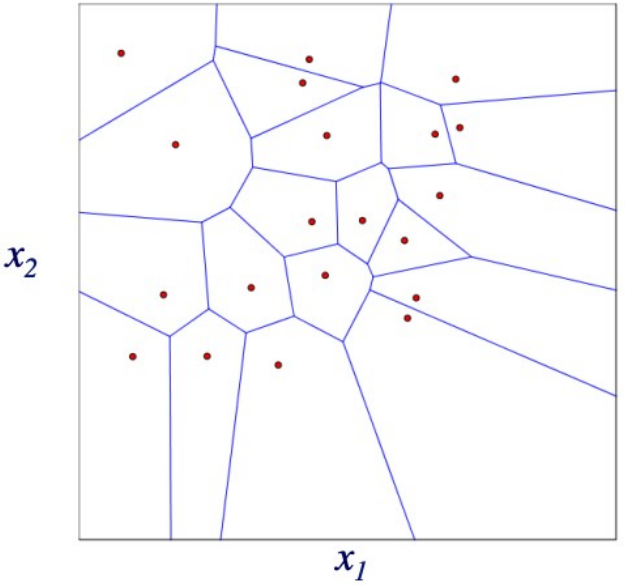
- Onderneem niets met de gegeven training-set $(x_1, y_1) \ldots (x_n, y_n)$ ook soms luie leerling genoemd

## Classification task:

- **Gegeven**: een te rangschikken voorbeeld $x_q$
- Zoek het training-set voorbeeld $x_i$ dat het meest gelijkenis vertoont met $x_q$
- Bepaal de rangschikkingswaarde $y_i$

# De beslissingsvelden voor naaste buur rangschikking

Voronoi Diagram: Elke polyhedron duidt het werkveld aan dat het dichtst in de buurt ligt van elk voorbeeld in de oefening



# *K*-naaste buur bepaling

Classification task:Gegeven:een te rangschikken voorbeeld $\boldsymbol{x}_q$

- Zoek het *k* training-set voorbeeld $(\boldsymbol{x}_1, y_1) \ldots (\boldsymbol{x}_k, y_k)$ dat het meeste gelijkenis vertoont met $\boldsymbol{x}_q$
- Bepaal de rangschikkingswaarde

$$\hat{y} \leftarrow \underset{v \in \text{values}(Y)}{\arg\max} \sum_{i=1}^{k} \delta(v, y_i) \qquad \delta(a,b) = \begin{cases} 1 \text{ if } a = b \\ 0 \text{ otherwise} \end{cases}$$

- (d.w.z. Duid de klasse aan waartoe de meeste buren behoren)

# Hoe kunnen we gelijkvormigheid/afstand bepalen?

Veronderstel dat alle kenmerken nominale(discrete)waarde hebben

- Hamming afstand: tel het aantal kenmerken die in de twee voorbeelden verschillend zijn

Veronderstel dat alle kenmerken continu zijn

- Euclidean afstand:
-

$$d(\boldsymbol{x}_i,\boldsymbol{x}_j) = \sqrt{\sum_f \left(x_{if} - x_{jf}\right)^2}$$ where $x_{if}$ represents the $f^{th}$ feature of $\boldsymbol{x}_i$

- Som de verschillende waarden op-in vergelijking met de Hamming afstand

# Hoe kunnen we gelijkvormigheid/afstand bepalen?

- Indien we zowel discrete/continous kenmerken hebben:

$$d(\boldsymbol{x}_i,\boldsymbol{x}_j) = \sum_f \begin{cases} \left|x_{if} - x_{jf}\right| & \text{if } f \text{ is continuous} \\ 1 - \delta\left(x_{if}, x_{jf}\right) & \text{if } f \text{ is discrete} \end{cases}$$

- Indien alle kenmerken even belangrijk zijn willen we op de continue kenmerken een soort van normalisatie toepassen(waarden van rang 0 tot 1) of een standaardisatie ((waarden toegekend volgens de normenstandaard)

# *K*-naaste buur regressie

Learning task:

- Onderneem niets met de gegeven training-set

$$(x_1, y_1) \ldots (x_n, y_n)$$

*Prediction task:*

**Gegeven:** maak een voorspelling voor instantie $x_q$

Vind de k-training set instanties $(x_1, y_1) \ldots (x_k, y_k)$ die het meest gelijken op $x_q$

Geef de waarde weer:

$$\hat{y} \leftarrow \frac{1}{k} \sum_{i=1}^{k} y_i$$

# Afstand-gewogen naaste buur bepaling

We kunnen waarden voorspellen op grond van hun afstand van $x_q$

## Classificatie:

$$\hat{y} \leftarrow \underset{v \in \text{values}(Y)}{\arg\max} \sum_{i=1}^{k} w_i \, \delta(v, y_i) \qquad w_i = \frac{1}{d(x_q, x_i)^2}$$

## Regressie:

$$\hat{y} \leftarrow \frac{\sum_{i=1}^{k} w_i \, y_i}{\sum_{i=1}^{k} w_i}$$

# Voordelen van leren aan de hand van voorbeelden

- Eenvoudig toepasbaar
- “Training” is  zeer efficiënt
- Geschikt voor on-line onderwijs
- Zeer sterk voor trainingsdata (“met veel ruis”)
- (als $k$ >1)
- Werkt dikwijls goed in de praktijk

# Beperking van leren aan de hand van voorbeelden

- Gevoelig voor irrelevante en verwante voorbeelden alhoewel.....
- - Er zijn varianten die gewicht toekennen
- – We zullen het later hebben over feature selectie methoden
- De rangschikking kan inefficiënt zijn, alhoewel uitgewerkte methoden en *k-d* bomen de zwakheden kunnen verlichten
- Verstrekt niet veel inzicht in het probleem domein omdat geen expliciet model voorhanden is