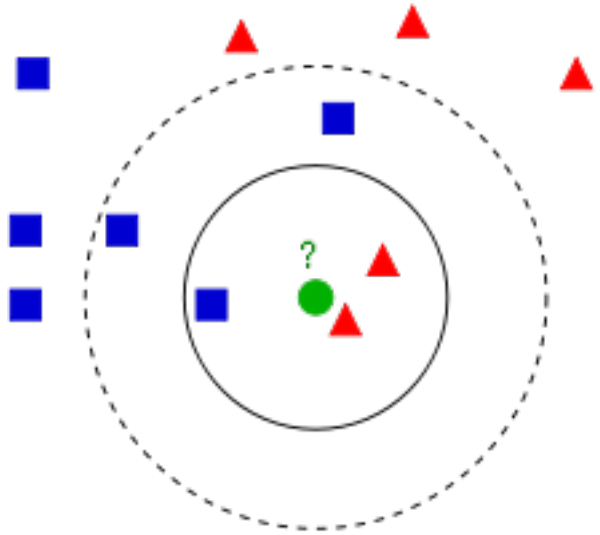



# Klasyfikator k-NN

Jerzy Stefanowski

Wykład uzupełniony 2020

Uczenie Maszynowe

dla ISWD i ITI PP

# Plan wykładu

1. Zasada najbliższego sąsiedztwa
2. Podstawowy algorytm k-NN
3. Dobór miary podobieństwa lub odległości
4. Strojenie liczby sąsiadów

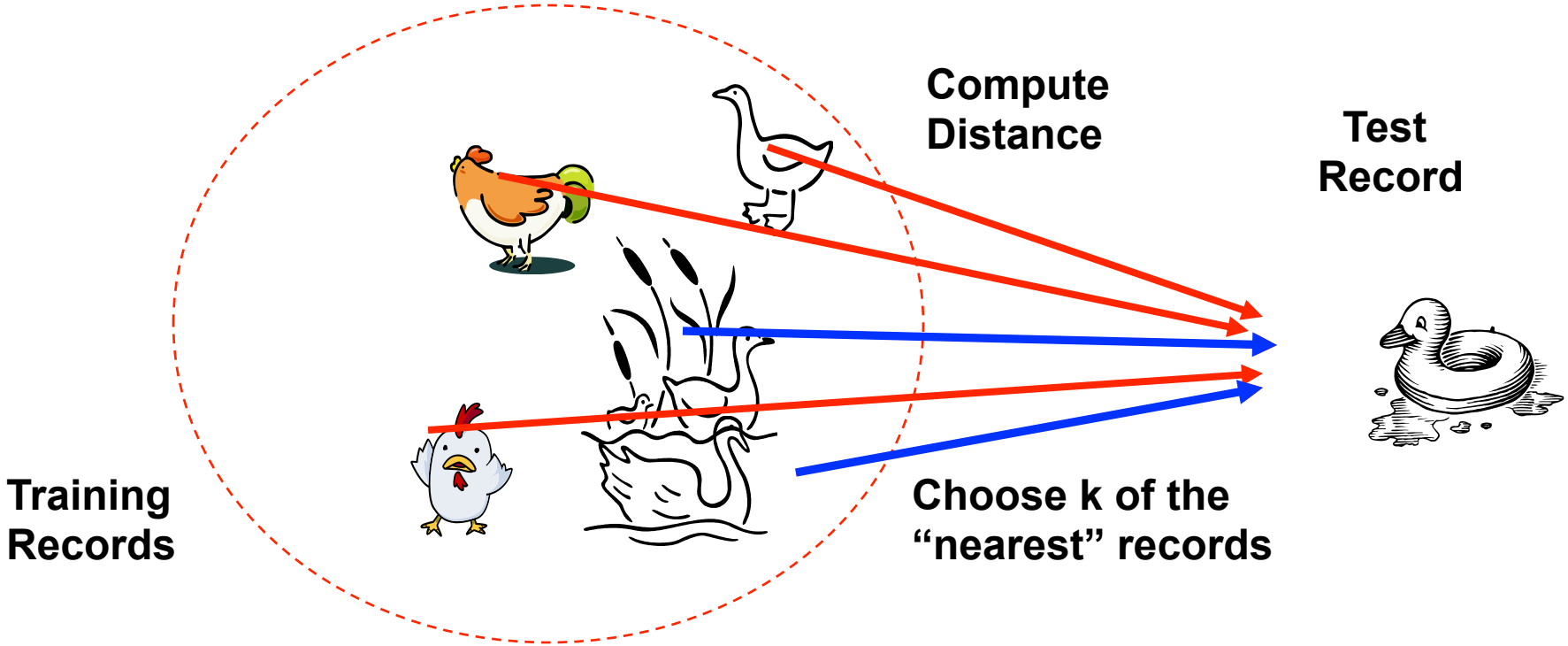
Rozszerzenia i powiązane zagadnienia:

5. Ograniczenia podstawowego algorytmu
6. Dobór przykładów uczących / tzw. Edited K-NN
7. Wybór przestrzeni atrybutów
8. Rozszerzenia algorytmu (np. regresja)
9. Podejścia minimalno-odległościowe w innych kontekstach

# Podobieństwo przykładów w klasyfikacji

Intuicja tzw. Nearest Neighbor Classifiers (bardzo prosta):

- "If it walks like a duck, quacks like a duck, then it's probably a duck"

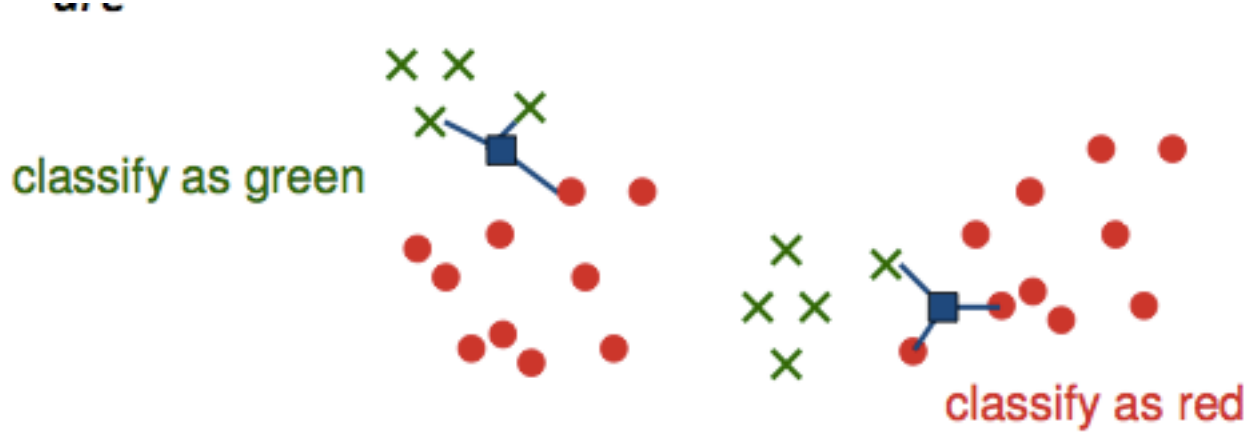


# Klasyfikator najbliższego sąsiada

- Prosta idea NN:
  - najbliższe przykłady mogą mieć tą samą etykietę
  - klasyfikuj nowy przykład na podstawie najbliższych (najbardziej podobnych) przykładów (uczących)

Algorytm (za chwilę)

- Decyzje do podjęcia:
  - Jak oceniać podobieństwo?
  - Jak wielu sąsiadów rozważać?



# *k*-Nearest-Neighbor Algorithm

**The case of discrete set of classes.**

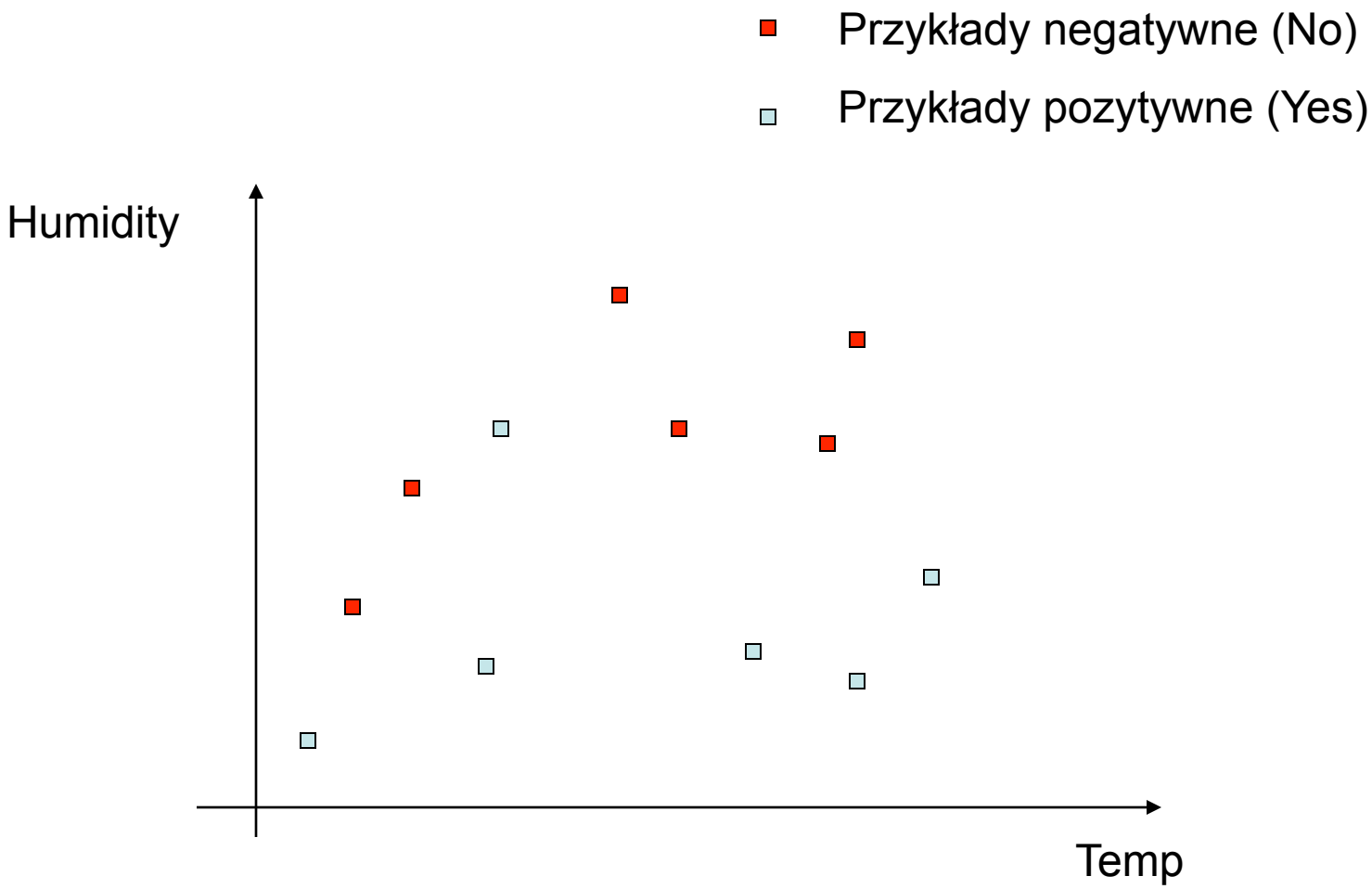
1. Take the instance *x* to be classified
2. Find *k* nearest neighbors of *x* in the training data.
3. Determine the class *c* of the majority of the instances among the *k* nearest neighbors.
4. Return the class *c* as the classification of *x*.

The distance functions are composed from difference metric $d_a$ defined for each two instances $x_i$ and $x_j$.

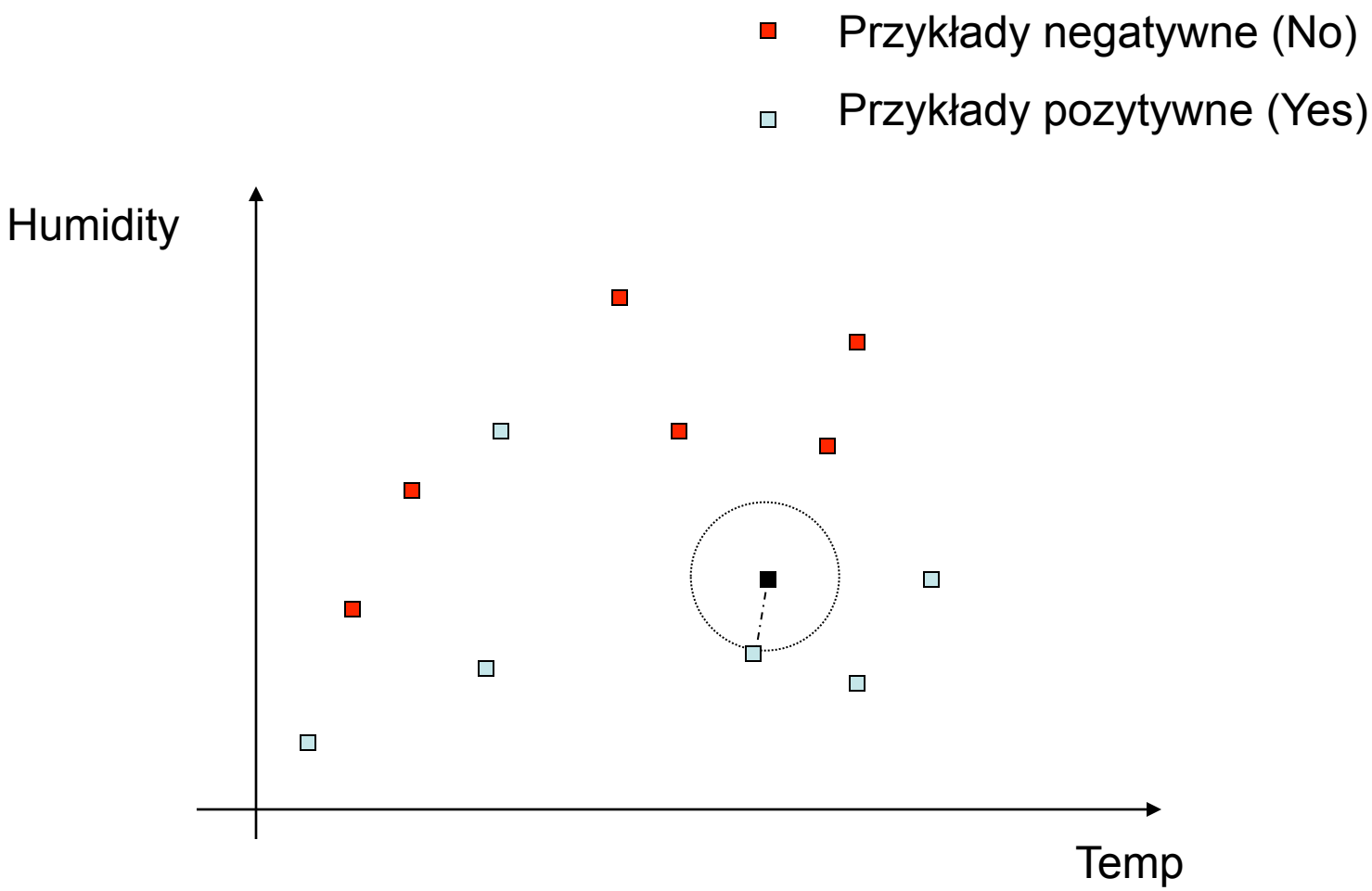
Ang. także - instance based learning

Inna polska nazwa – klasyfikator minimalno odległościowy

# Działanie klasyfikatora NN



# Działanie klasyfikatora NN



# Pytania przy działaniu z k-NN

- Jak dobrać ocenę podobieństwa przykładów?
- Dobór paramaterów (k –> hyper-parametr)
- Jak głosować (prosto, ważone,...)?
- Właściwości klasyfikatora, interpretacja geometryczna?
- Klątwa wymiarowości – selekcja cech?
- Dobór przykładów uczących.

# Określanie podobieństwa

- Różne spojrzenie – co to jest podobny do
- Funkcja odległości jako odwrotność podobieństwa
- Typowo odległośc euklidesowa - *Euclidean distance*:

$$\sqrt{(a_1^{(1)} - a_1^{(2)})^2 + (a_2^{(1)} - a_2^{(2)})^2 + \ldots + (a_k^{(1)} - a_k^{(2)})^2}$$

  $\mathbf{a}^{(1)}$ i $\mathbf{a}^{(2)}$: dwa przykłady opisane *k* atrybutami
- Może mieć różne warianty (np. ważone cechy)
- Inne tzw. metryki: *city-block (Manhattan) metric*
  - Proste dodawanie wartości odległości
  - Metryka Minkowskiego Lq
  - Miara Czebyszewa

$$\sum_{i=1}^{k} | x_i - y_i |$$

$$\max_{i=1}^{k} | x_i - y_i |$$

# Metryki

Miara odległości x od y - D(x, y)

Metryka matem. musi spełniać aksjomaty:

- $D(x, x) = 0$
- $D(x, y) = D(y, x)$
- $D(x, y) \leq D(x, z) + D(z, y)$ prawo trójkąta

# Normalizacja wartości atrybutów

- Atrybuty definiowane na skalach o różnym zakresie (rozstępie) ⇒ potrzeba *normalizacji*:

$$a_i = \frac{v_i - \min v_i}{\max v_i - \min v_i} \qquad \text{lub} \qquad a_i = \frac{v_i - Avg(v_i)}{StDev(v_i)}$$

  $v_i$ : wartość atrybutu *i*

- Nominalne atrybuty: najprościej odległość 0 lub 1 -> oraz b. zaawansowane miary
- Co z nieznanymi wartościami atrybutów (missing attribute values):
  - Pomijamy w obliczeniach
  - Zakładamy maksymalne odległości (z uwagi na normalizację)

# Przykład obliczeń 1NN

Brak normalizacji

| Age | Loan | Class | Distance |
|---|---|---|---|
| 25 | $40,000 | N | 102000 |
| 35 | $60,000 | N | 82000 |
| 45 | $80,000 | N | 62000 |
| 20 | $20,000 | N | 122000 |
| 35 | $120,000 | N | 22000 |
| 52 | $18,000 | N | 124000 |
| 23 | $95,000 | Y | 47000 |
| 40 | $62,000 | Y | 80000 |
| 60 | $100,000 | Y | 42000 |
| 48 | $220,000 | Y | 78000 |
| 33 | $150,000 | Y | 8000 |
| **48** | **$142,000** | **?** | |

Euclidean Distance

$$D = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

# KNN Classification – Standardized Distance

| Age | Loan | Default | Distance |
|---|---|---|---|
| 0.125 | 0.11 | N | 0.7652 |
| 0.375 | 0.21 | N | 0.5200 |
| 0.625 | 0.31 | N | 0.3160 |
| 0 | 0.01 | N | 0.9245 |
| 0.375 | 0.50 | N | 0.3428 |
| 0.8 | 0.00 | N | 0.6220 |
| 0.075 | 0.38 | Y | 0.6669 |
| 0.5 | 0.22 | Y | 0.4437 |
| 1 | 0.41 | Y | 0.3650 |
| 0.7 | 1.00 | Y | 0.3861 |
| 0.325 | 0.65 | Y | 0.3771 |
| **0.7** | **0.61** | **?** | |

Standardized Variable

$$X_s = \frac{X - Min}{Max - Min}$$

# „Świat” miar odległości

**Minkowsky:**

$$D(\boldsymbol{x},\boldsymbol{y}) = \left(\sum_{i=1}^{m} |x_i - y_i|^r\right)^{1/r}$$

**Euclidean:**

$$D(\boldsymbol{x},\boldsymbol{y}) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2}$$

**Manhattan / city-block:**

$$D(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{m} |x_i - y_i|$$

**Camberra:**

$$D(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{m} \frac{|x_i - y_i|}{|x_i + y_i|}$$

**Chebychev:**

$$D(\boldsymbol{x},\boldsymbol{y}) = \max_{i=1}^{m} |x_i - y_i|$$

**Quadratic:**

$$D(\boldsymbol{x},\boldsymbol{y}) = (\boldsymbol{x} - \boldsymbol{y})^T Q(\boldsymbol{x} - \boldsymbol{y}) = \sum_{j=1}^{m}\left(\sum_{i=1}^{m}(x_i - y_i)q_{ji}\right)(x_j - y_j)$$

Q is a problem-specific positive definite $m \times m$ weight matrix

**Mahalanobis:**

$$D(\boldsymbol{x},\boldsymbol{y}) = [\det V]^{1/m}(\boldsymbol{x} - \boldsymbol{y})^{\mathrm{T}} V^{-1}(\boldsymbol{x} - \boldsymbol{y})$$

$V$ is the covariance matrix of $A_1..A_m$, and $A_j$ is the vector of values for attribute $j$ occuring in the training set instances $1..n$.

**Correlation:**

$$D(\boldsymbol{x},\boldsymbol{y}) = \frac{\sum_{i=1}^{m}(x_i - \overline{x_i})(y_i - \overline{y_i})}{\sqrt{\sum_{i=1}^{m}(x_i - \overline{x_i})^2 \sum_{i=1}^{m}(y_i - \overline{y_i})^2}}$$

$\overline{x_i} = \overline{y_i}$ and is the average value for attribute $i$ occuring in the training set.

**Chi-square:**

$$D(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{m} \frac{1}{sum_i}\left(\frac{x_i}{size_{\boldsymbol{x}}} - \frac{y_i}{size_{\boldsymbol{y}}}\right)^2$$

$sum_i$ is the sum of all values for attribute $i$ occuring in the training set, and $size_x$ is the sum of all values in the vector $\boldsymbol{x}$.

**Kendall's Rank Correlation:**

$$D(\boldsymbol{x},\boldsymbol{y}) = 1 - \frac{2}{n(n-1)} \sum_{i=1}^{m} \sum_{j=1}^{i-1} \mathrm{sign}(x_i - x_j)\,\mathrm{sign}(y_i - y_j)$$

sign($x$)=-1, 0 or 1 if $x < 0$, $x = 0$, or $x > 0$, respectively.

# Odległość dla zmiennych jakościowych

| X | Y | Distance |
|---|---|---|
| Male | Male | 0 |
| Male | Female | 1 |

$$x = y \Rightarrow D = 0$$

$$x \neq y \Rightarrow D = 1$$

# K-NN dla innych sytuacji

| | Food (3) | Chat (2) | Fast (2) | Price (3) | Bar (2) | BigTip |
|---|---|---|---|---|---|---|
| 1 | great | yes | yes | normal | no | yes |
| 2 | great | no | yes | normal | no | yes |
| 3 | mediocre | yes | no | high | no | no |
| 4 | great | yes | yes | normal | yes | yes |

Similarity metric: Number of matching attributes (k=2)

- New examples:
  - Example 1 (great, no, no, normal, no) Yes

    → most similar: number 2 (1 mismatch, 4 match) → yes

    →Second most similar example: number 1 (2 mismatch, 3 match) → yes

  - Example 2 (mediocre, yes, no, normal, no) Yes/No

    → Most similar: number 3 (1 mismatch, 4 match) → no

    →Second most similar example: number 1 (2 mismatch, 3 match) → yes

# Inne metryki na mieszanych typów atrybutów (Wilson,Martinez – studium z HVDM)

- Heterogenuous distance measures

In this section, we define a heterogeneous distance function *HVDM* that returns the distance between two input vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. It is defined as follows:

$$HVDM(\boldsymbol{x},\boldsymbol{y}) = \sqrt{\sum_{a=1}^{m} d_a^2(x_a,y_a)} \qquad (11)$$

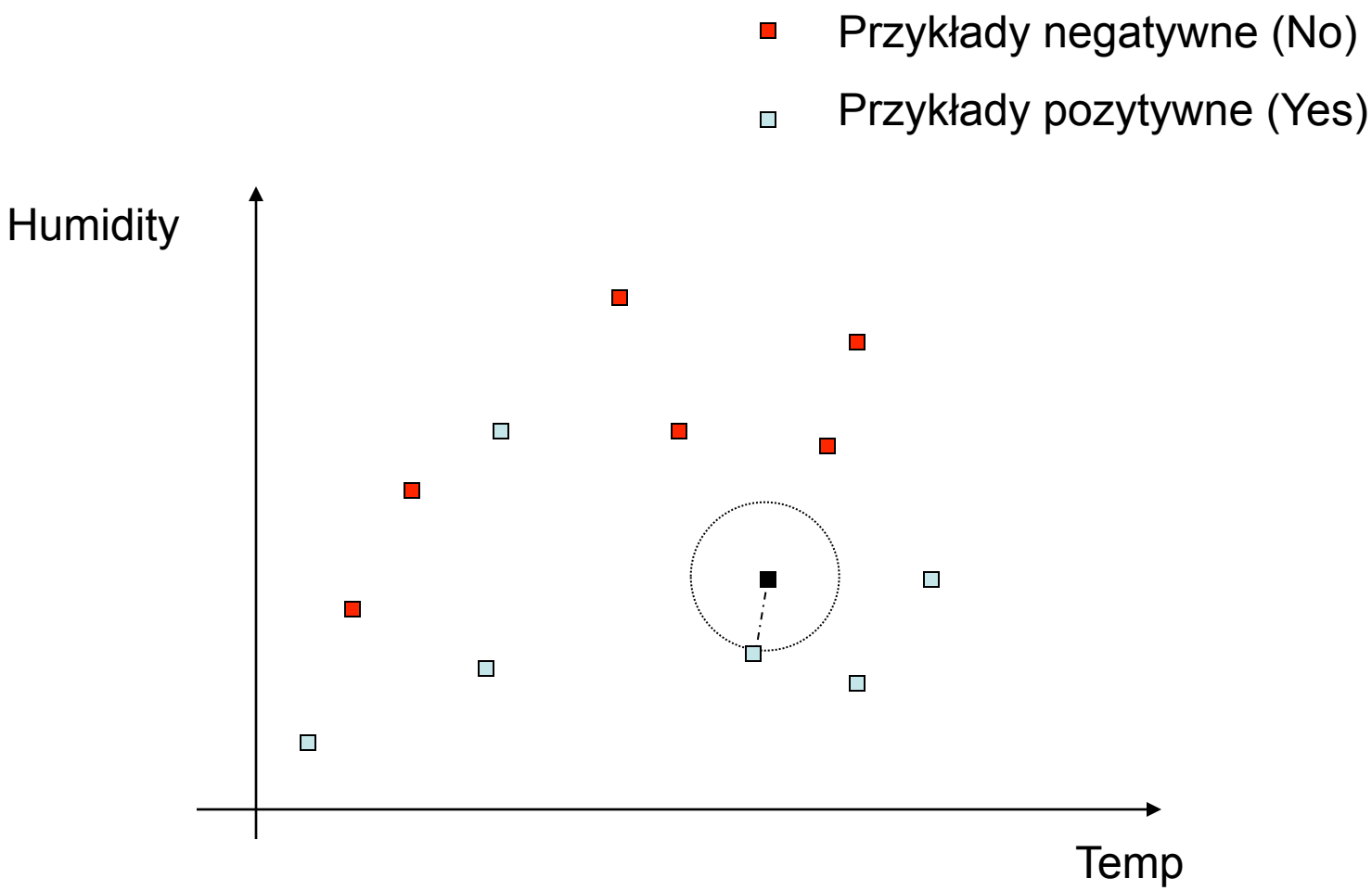
where $m$ is the number of attributes. The function $d_a(x,y)$ returns a distance between the two values $x$ and $y$ for attribute $a$ and is defined as:

$$d_a(x,y) = \begin{cases} 1, & \text{if } x \text{ or } y \text{ is unknown; otherwise...} \\ normalized\_vdm_a(x,y), & \text{if } a \text{ is nominal} \\ normalized\_diff_a(x,y), & \text{if } a \text{ is linear} \end{cases} \qquad (12)$$

$$normalized\_vdm3_a(x,y) = \sqrt{C * \sum_{c=1}^{C} \left| \frac{N_{a,x,c}}{N_{a,x}} - \frac{N_{a,y,c}}{N_{a,y}} \right|^2}$$

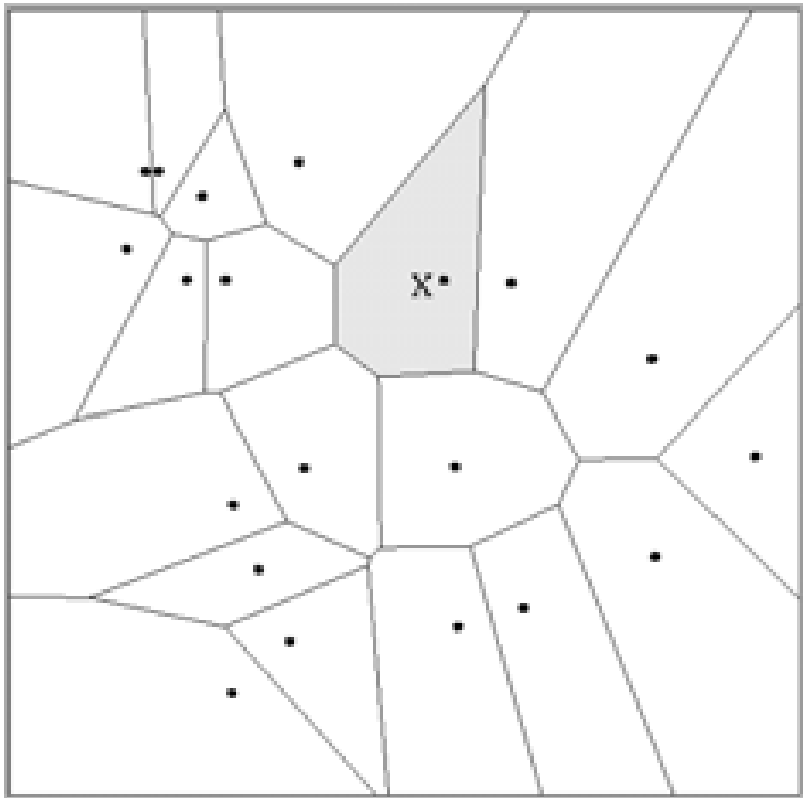
| Database | Euclid. | HOEM | HVDM |
|---|---|---|---|
| Anneal | **94.99** | 94.61 | 94.61 |
| Audiology | 60.50 < | 72.00 < | **77.50** |
| Audiology.Test | 41.67 < | 75.00 | **78.33** |
| Australian | 80.58 | 81.16 | **81.45** |
| Bridges | 58.64 | 53.73 | **59.64** |
| Crx | 78.99 | **81.01** | 80.87 |
| Echocardiogram | **94.82** | **94.82** | **94.82** |
| Flag | 48.95 < | 48.84 | **55.82** |
| Heart.Cleveland | 73.94 | 74.96 | **76.56** |
| Heart.Hungarian | 73.45 < | 74.47 | **76.85** |
| Heart.Long-Beach-Va | **71.50** | 71.00 * | 65.50 |
| Heart.More | **72.09** | 71.90 | **72.09** |
| Heart.Swiss | **93.53** * | 91.86 | 89.49 |
| Hepatitis | **77.50** | 77.50 | 76.67 |
| Horse-Colic | **65.77** | 60.82 | 60.53 |
| House-Votes-84 | 93.12 < | 93.12 < | **95.17** |
| Image.Segmentation | 92.86 | **93.57** | 92.86 |
| Led+17 | 42.90 < | 42.90 < | **60.70** |
| Led-Creator | **57.20** * | **57.20** * | 56.40 |
| Monks-1.Test | **77.08** | 69.43 | 68.09 |
| Monks-2.Test | 59.04 < | 54.65 < | **97.50** |
| Monks-3.Test | 87.26 < | 78.49 < | **100.00** |
| Mushroom | **100.00** | **100.00** | **100.00** |
| Promoters | 73.73 < | 82.09 < | **92.36** |
| Soybean-Large | 87.26 < | 89.20 | **90.88** |
| Soybean-Small | **100.00** | **100.00** | **100.00** |
| Thyroid.Allbp | 94.89 | 94.89 | **95.00** |
| Thyroid.Allhyper | **97.00** | **97.00** | 96.86 |
| Thyroid.Allhypo | **90.39** | **90.39** * | 90.29 |
| Thyroid.Allrep | **96.14** | **96.14** | 96.11 |
| Thyroid.Dis | **98.21** | **98.21** | **98.21** |
| Thyroid.Hypothyroid | **93.42** | **93.42** | 93.36 |
| Thyroid.Sick-Euthyroid | **68.23** | **68.23** | **68.23** |
| Thyroid.Sick | **86.93** * | 86.89 * | 86.61 |
| Zoo | 97.78 | 94.44 | **98.89** |

# Działanie klasyfikatora NN

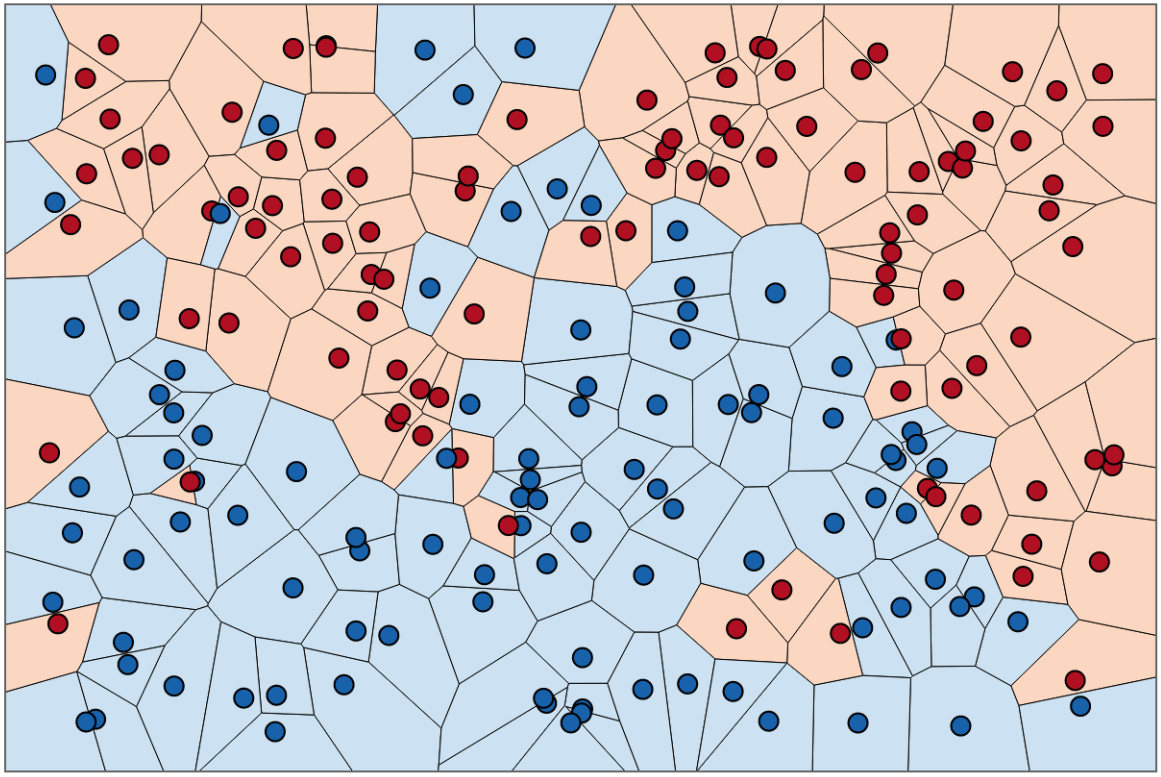


# Interpretacja1-Nearest Neighbor

Voronoi diagrams w przestrzeni atrubutów R − kształt odcinkami liniowy (wygląd zależy od wybranej miary odległości)



# Voronoi Cell Visualization of Nearest Neighborhoods



1-NN może przybliżać b. trudne rozkłady przykładów

# Cechy klasyfikatora 1-NN

- Zalety:
  - Prostota
  - Bardzo szybki proces uczenia
- Wady:
  - Duże zapotrzebowanie na pamięć
  - Powolne odpytywanie
  - Wiedza = przykłady uczące (brak reprezentacji wiedzy)
  - Wszystkie atrybuty są tak samo istotne
  - Znaczna podatność na przeuczenie
- Liczne modyfikacje

# Własności k-NN

- K-NN jako specjalny przypadek tzw. a variable-bandwidth, kernel density "balloon" estimator
- Oszacowane teoretyczne błędu w odniesieniu do optymalnego klasyfikatora Bayesowskiego

Cover i Hart [2] opublikowali w 1967 roku **oszacowanie ryzyka klasyfikatora najbliższy sąsiad** $R_{NN}$ za pomocą ryzyka $R^*$ optymalnego algorytmu Bayesa w asymptotycznym przypadku, gdy długość ciągu uczącego $N \to \infty$

$$R^* \leq R_{NN} \leq R^* \left(2 - \frac{M}{M-1} R^*\right),$$

$M$ to liczba klas.

W przypadku problemu dwuklasowego ($M = 2$) otrzymujemy oszacowanie

$$R^* \leq R_{NN} \leq 2R^* (1 - R^*).$$

# Jak dobrać k liczbę sąsiadów?

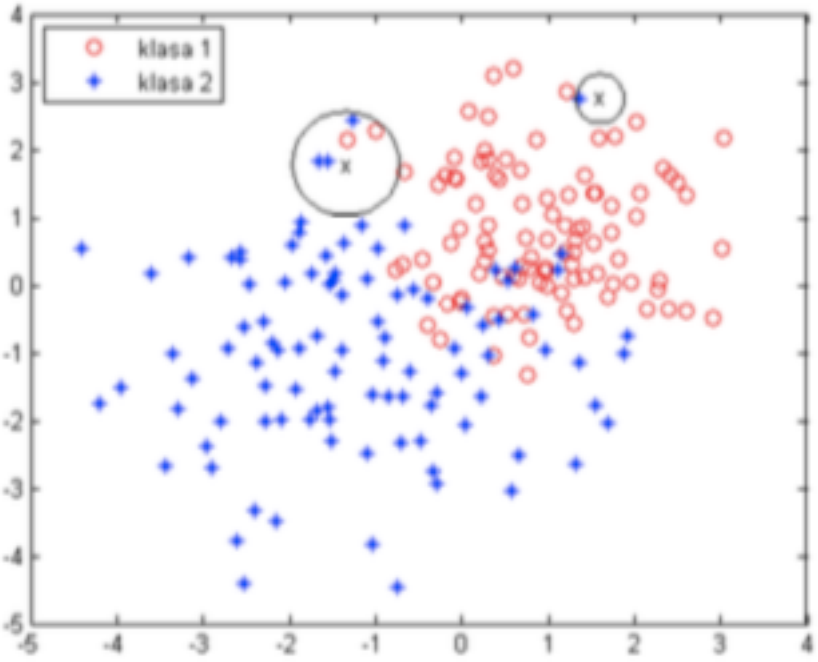
Liczba *k* musi być:

— na tyle duża, by redukować wrażliwość algorytmu na trudne przykładu

— na tyle mała, by nie wybierać sąsiadów mocno osadzonych w innych klasach

Jeśli k jest duże, koszt obliczenia jest większy → algorytm jest czasochłonny

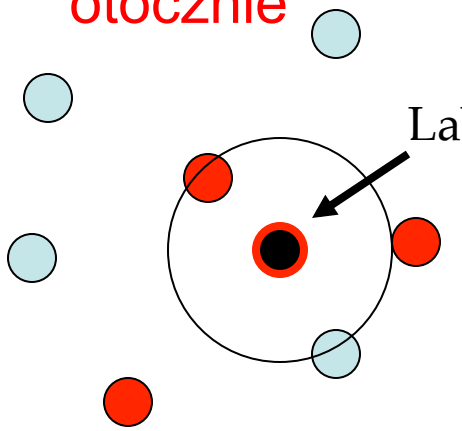
— trzeba także uwzględniać wielkość zbioru uczącego lub stosować specjalne procedury strojenia (wewnętrzna ocena krzyżowa)



# k – Nearest Neighbor

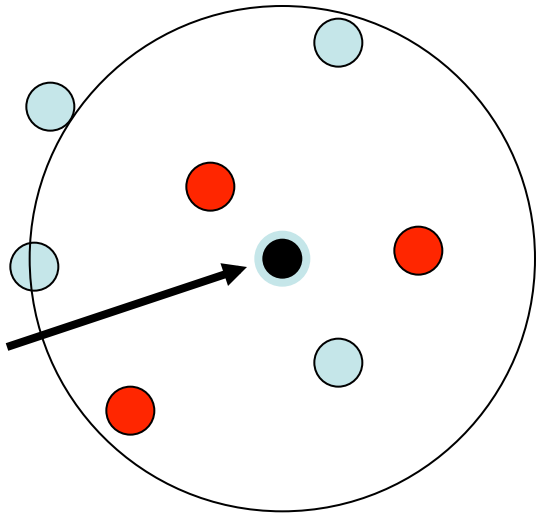
Wybrać większą wartość *k*

- Pozwala lepiej uwzględniąć trudne lokalne rozkłady przykładów (over-lapping, noise)
- Nowy obiekt etykietowany na podstawie najczęściej występujących etykiet wśród *k* najbliższych sąsiadów
- Jest to bezpośrednie oszacowanie warunkowego prawdopodobieństwa a posteriori p(j|x) przynależności zaobserwowanej wartości x do klasy j !!! Tzw. bezpośrednie otocznie

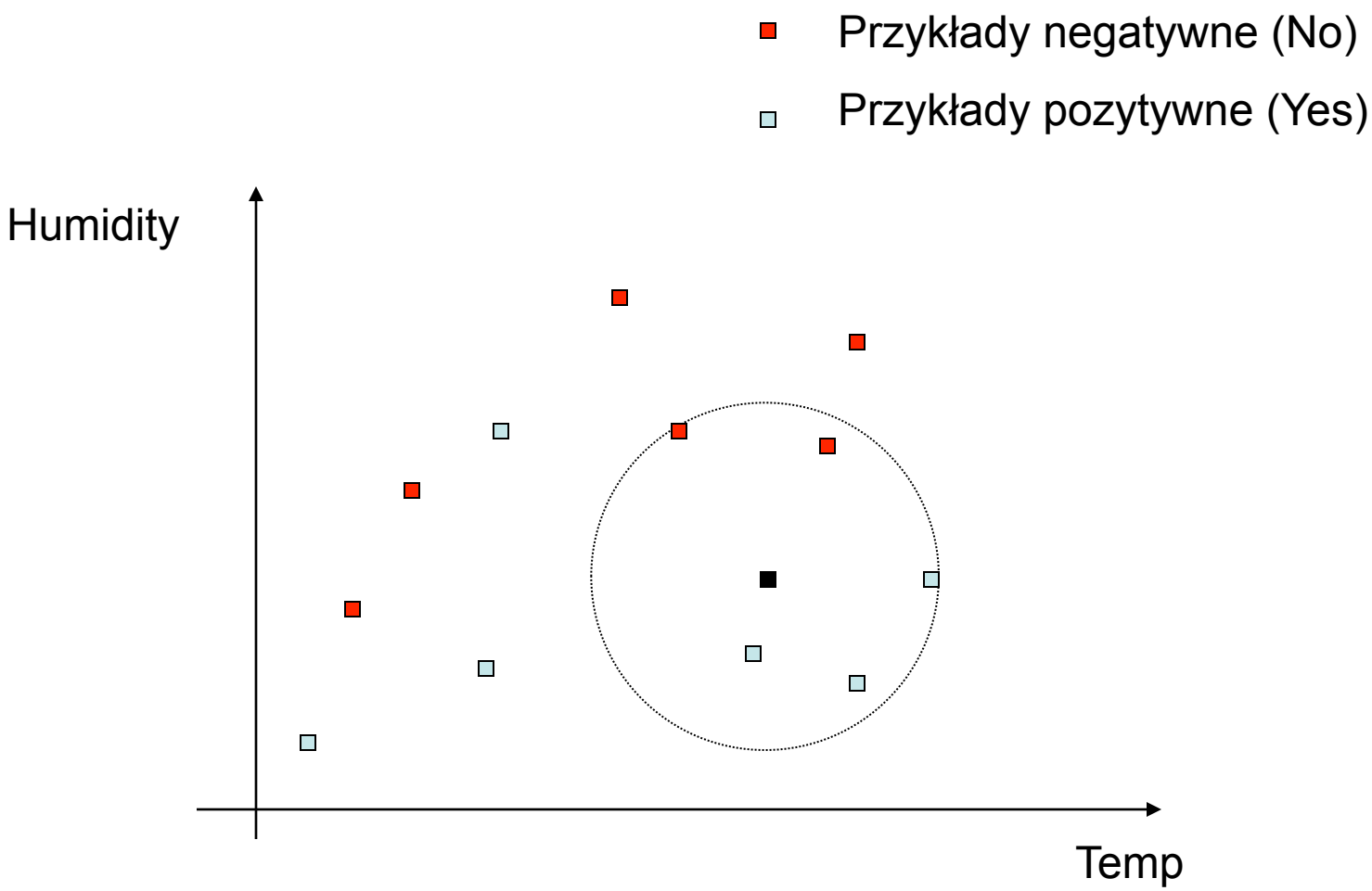


Label it red, when k = 3

Label it blue, when k = 7



# Klasyfikator k-NN, k=5




Jeśli byłoby więcej k równo-odległych obserwacji − rozważamy wszystkie

# Oszacowanie prawdopodobieństwa

$k$ obserwacji ($k$-nearest neighbors, k-nn) najbliższych $\mathbf{x}_i$ spośród wszystkich $\mathbf{x}_1, ..., \mathbf{x}_n$

Czyli **bezpośrednim otoczeniem** punktu $\mathbf{x}$ jest kula w $R^p$, o środku w $\mathbf{x}$ i promieniu takim, żeby znalazło się w nim dokładnie $k$ obserwacji $\mathbf{x}_i$

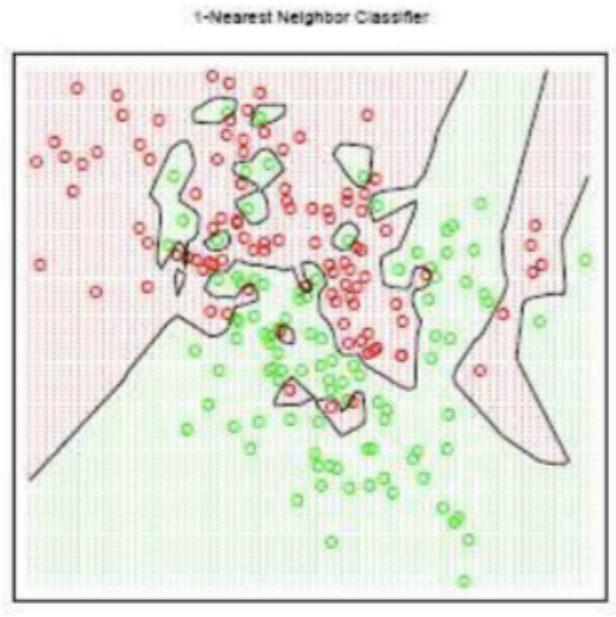
$$p(j|\mathbf{x}) = \frac{\sum_{\langle k \rangle} \mathbf{x} \in j}{k}$$

Obserwacja $\mathbf{x}$ zostanie sklasyfikowana do tej klasy $j$, z której pochodzi najwięcej spośród $k$ najbliższych punktowi $\mathbf{x}$ obserwacji próby uczącej

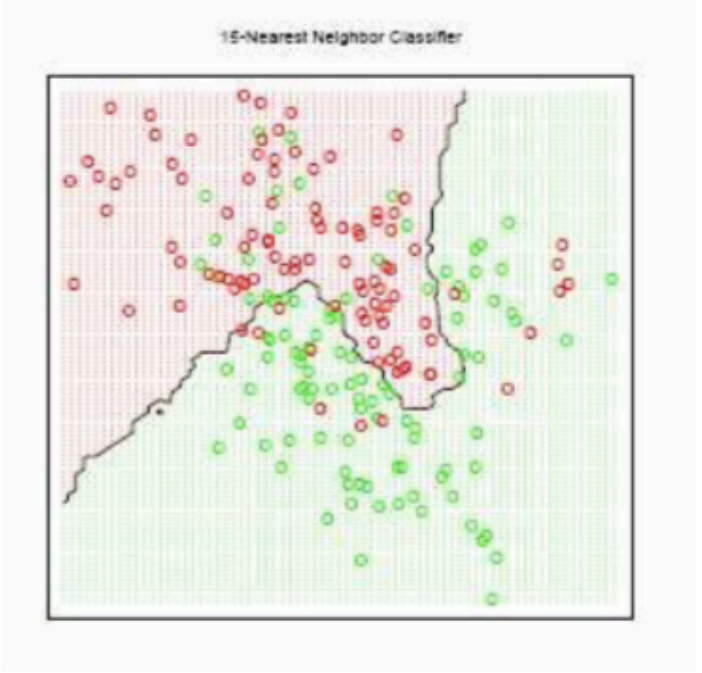
Za wykład J. Sienkiewicz

# Interpretacja granicy decyzyjnej

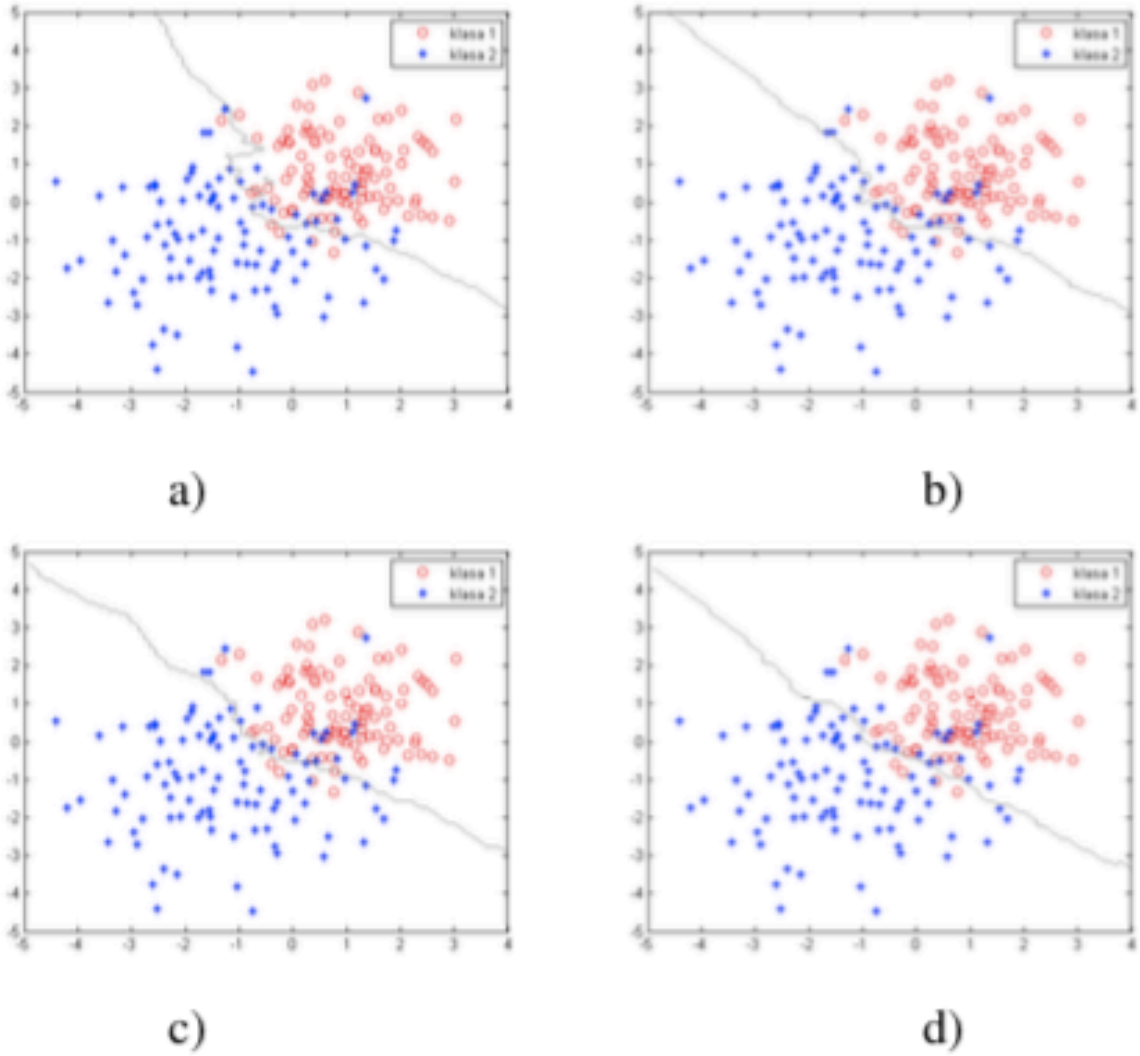
K=1




K=15



# Różnice w interpretacji geometrycznej




Rysunek 4. Klasyfikator k-najbliższych sąsiadów: (a) $k = 9$, (b) $k = 19$, (c) $k = 29$, (d) $k = 59$.

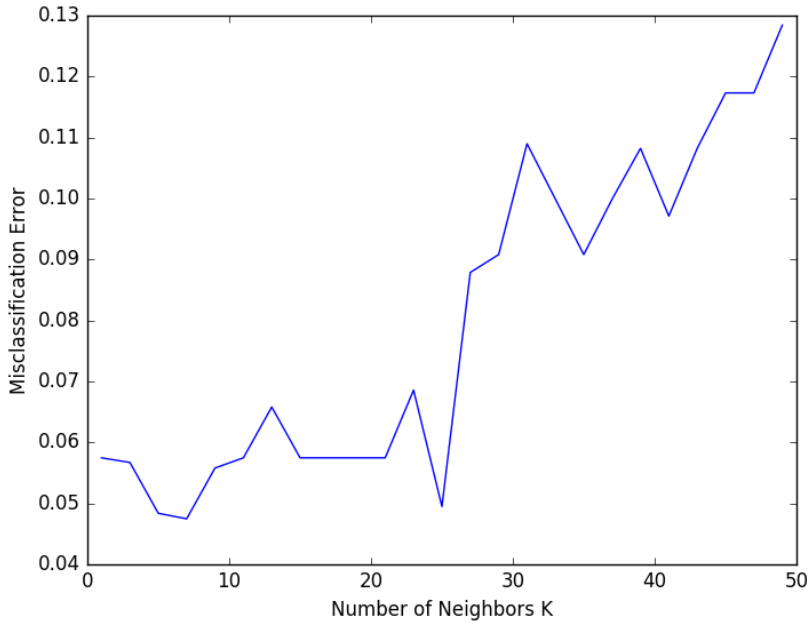
# Jak dobierać wielkość *k*?

Wyniki analizy z przeszłości - nieparzysta mała liczba

- *k* zależne od wielkości zbioru danych *n*
- np. *k* = sqrt(*n*) lub podobne funkcje nieliniowe

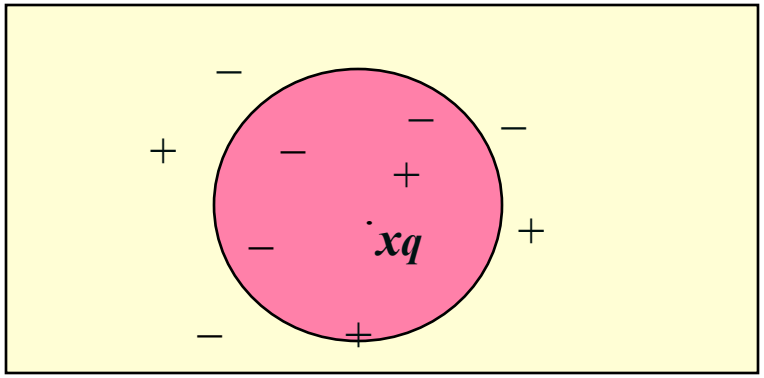
Dobór eksperymentalny

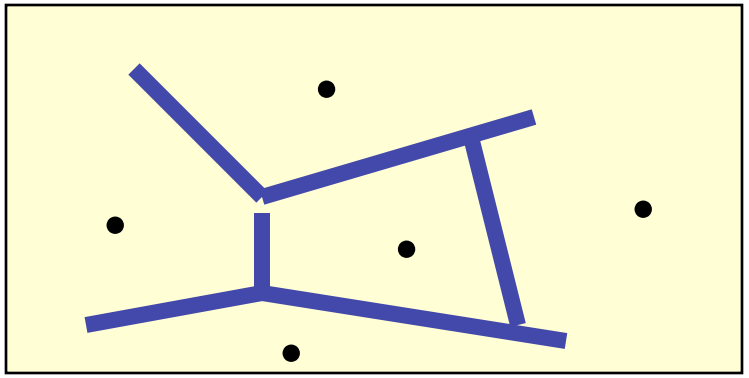
- Dodatkowa ocena walidacyjnego (techniki jak tzw. internal cross validation)



# Algorytm *k*-Nearest Neighbor (kNN)

- Przykłady – punkty w przestrzeni *R* *n*-wymiarowej.
- Najbliżsi sąsiedzi definiowani funkcją odległości → odwrotność podobieństwa
- Dyskretne, *k*-NN zwraca najczęstsza z decyzji dla *k* przykładów najbliższych $x_q$
  - Głosowanie większościowe
  - Głosowanie ważone (odwrotność odległości)
- Parametryzacja (odległość, k, …)



# Rozszerzenia *k*-NN

- The k-NN algorithm for continuous-valued target functions.
  - Calculate the mean values of the *k* nearest neighbors.
- Distance-weighted nearest neighbor algorithm.
  - Weight the contribution of each of the k neighbors according to their distance to the query point *xq*.
    - giving greater weight to closer neighbors: $w \equiv \frac{1}{d(x_q, x_i)^2}$
  - Similarly, we can distance-weight the instances for real-valued target functions.
- Robust to noisy data by reduction of the example basis
- Curse of dimensionality: distance between neighbors could be dominated by irrelevant attributes. To overcome it,
  - Selection of the least relevant attributes
  - Weighted distances

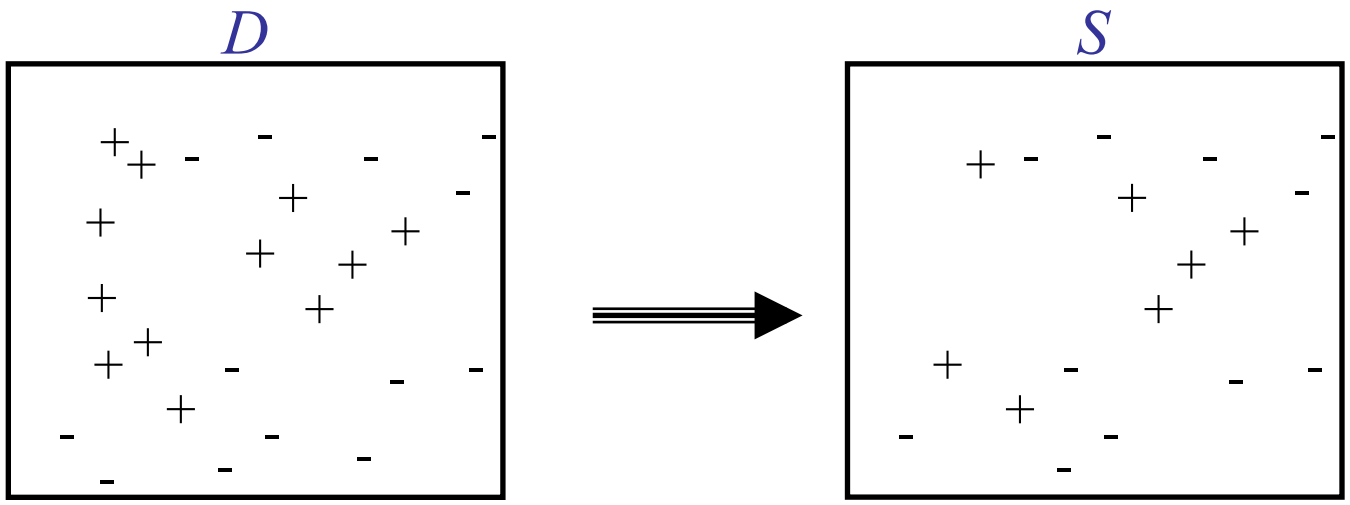
$$d(x_1, x_2) = \sqrt{\sum_A w_A \cdot d_A(v_{1,A}, v_{2,A})^2}$$

Za wykładem

# Condensed (edited) NN Algorithm

*The Condensed NN algorithm was introduced to reduce the storage requirements of the NN algorithm and improve classification for complex distribution.*

The algorithm finds a subset *S* of the training data *D* s.t. each instance in *D* can be correctly classified by the NN algorithm applied on the subset *S*.



# Condensed nearest neighbor (CNN, the Hart algorithm)

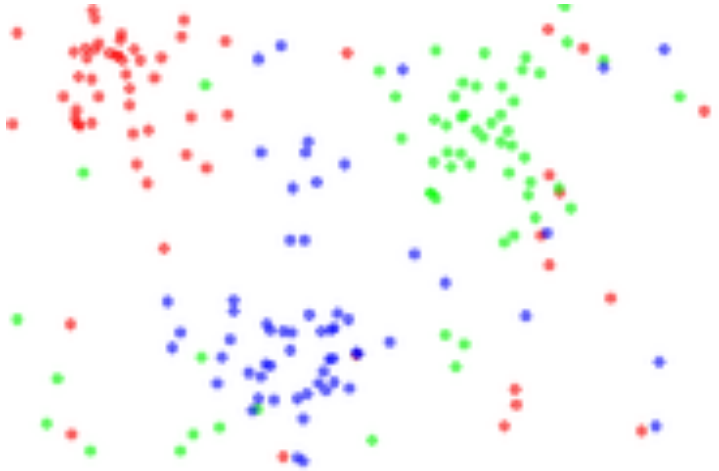
Given a training set X, CNN works iteratively:

- Scan all elements of X, looking for an element x whose nearest prototype from U has a different label than x.
- Remove x from X and add it to U
- Repeat the scan until no more prototypes are added to U.

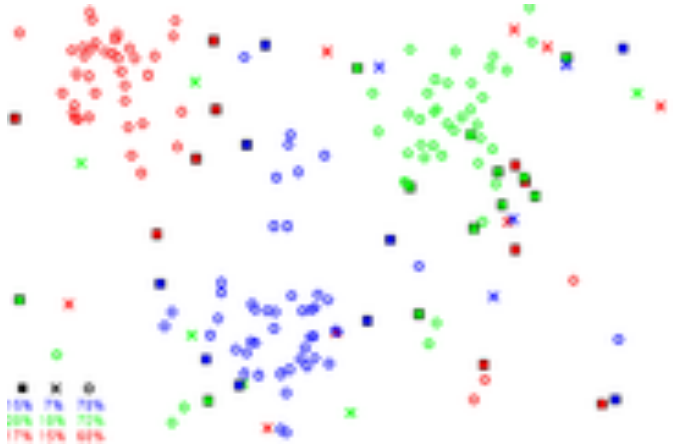
Use U instead of X for classification. The examples that are not prototypes are called "absorbed" points.

Inne wersje edytowanych K-NN spójrz do Wilson, D. & Martinez, Tony. (2000). Reduction Techniques for Instance-Based Learning Algorithms. Machine Learning.

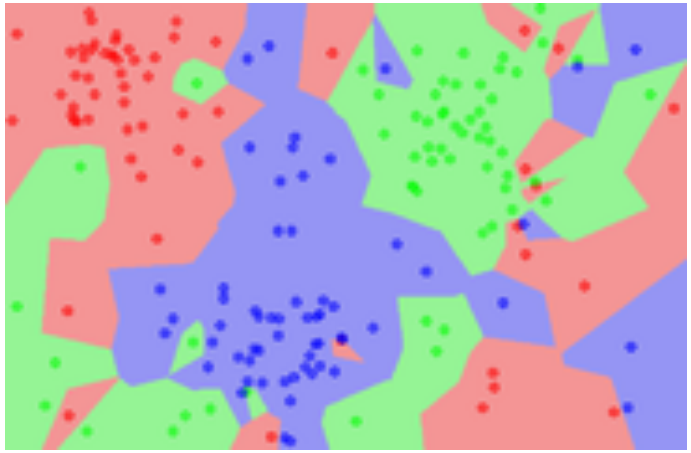
# CNN – przykład (1-NN)

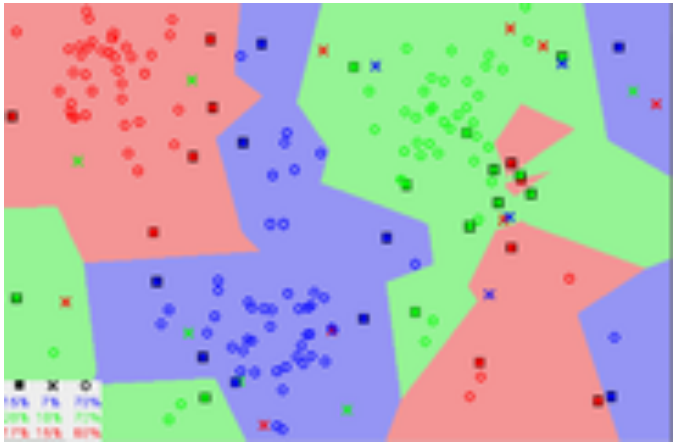


Oryginalny zbiór



Zredukowany CNN zbiór





Rysunki wikipedia

# Instance Based Reasoning (Aha)

- **IB1** is based on the standard KNN
- **IB2** is incremental KNN learner that only incorporates misclassified instances into the classifier.
- **IB3** discards instances that do not perform well by keeping success records.

Spójrz do Aha, David W., Dennis Kibler, Marc K. Albert (1991). Instance-Based Learning Algorithms. Machine Learning, 6, pp. 37-66.

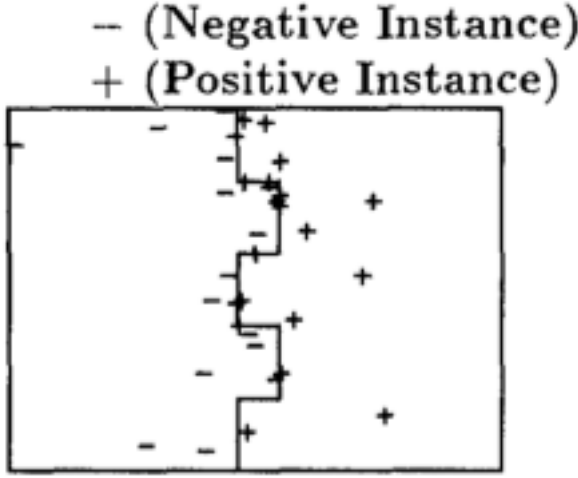
# IBL 2

*Table 2.* The IB2 algorithm (*CD* = Concept Description).

---

$CD \leftarrow \emptyset$
**for each** $x \in$ Training Set **do**
1. **for each** $y \in CD$ **do**
   Sim[$y$] ← Similarity($x$, $y$)
2. $y_{max} \leftarrow$ some $y \in CD$ with maximal Sim[$y$]
3. **if** class($x$) = class($y_{max}$)
   **then** classification ← **correct**
   **else**
   3.1 classification ← **incorrect**
   3.2 $CD \leftarrow CD \cup \{x\}$

---




After 5 Instances After 25 Instances After 100 Instances

*Table 5.* The IB3 algorithm (*CD* = Concept Description).

$CD \leftarrow \emptyset$
**for each** $x$ in Training Set **do**
1. **for each** $y \in CD$ **do**
   Sim[$y$] ← Similarity($x$, $y$)
2. **if** $\exists\{y \in CD |$ acceptable($y$)$\}$
   **then** $y_{max}$ ← some acceptable $y \in Cd$ with maximal Sim[$y$]
   **else**
      2.1 $i$ ← a randomly-selected value in [1,$|CD|$]
      2.2 $y_{max}$ ← some $y \in CD$ that is the $i$-th most similar instance to $x$
3. **if** class($x$) $\neq$ class($y_{max}$)
   **then** classification ← **correct**
   **else**
      3.1 classification ← **incorrect**
      3.2 $CD \leftarrow CD \cup \{x\}$
4. **for each** $y$ in $CD$ **do**
   **if** Sim[$y$] $\geq$ Sim[$y_{max}$]
   **then**
      4.1 Update $y$'s classification record
      4.2 **if** $y$'s record is significantly poor
         **then** $CD \leftarrow C - \{y\}$

Wykorzystuje tzw. rekord klasyfikacyjny przykładu oraz spec. test statystyczny

# Porównanie IBL (D.Aha)

*Table 6.* Percent accuracy ± standard error and percent storage requirements. IB3 recorded higher classification accuracies and lower storage requirements than IB2. It also compares favorably with the other algorithms' classification accuracies.

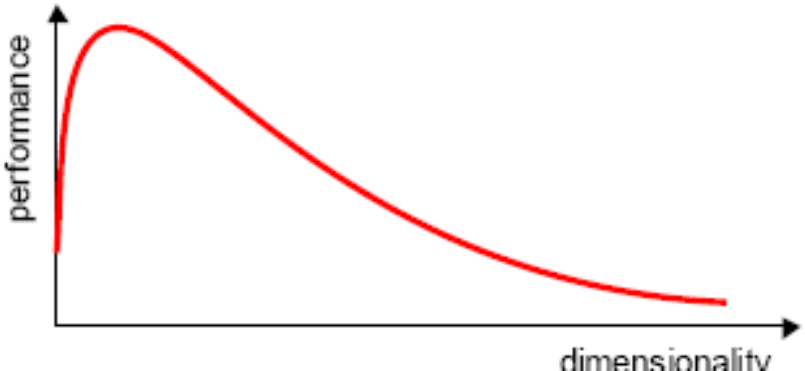
| Database | IB1 | IB2 | IB3 | C4 |
|---|---|---|---|---|
| Voting | 91.8 ± 0.4 100 | 90.9 ± 0.5 11.1 | 91.6 ± 0.5 7.4 | 95.5 ± 0.3 |
| Tumor | 34.7 ± 0.8 100 | 32.9 ± 0.8 71.3 | 38.6 ± 0.9 16.4 | 37.8 ± 0.9 |
| LED Display | 70.5 ± 0.4 100 | 62.4 ± 0.6 41.5 | 71.7 ± 0.4 28.7 | 68.3 ± 0.3 |
| Waveform | 75.2 ± 0.3 100 | 69.6 ± 0.4 32.5 | 73.8 ± 0.4 14.6 | 70.7 ± 0.3 |
| Cleveland | 75.7 ± 0.8 100 | 71.4 ± 0.8 30.4 | 78.0 ± 0.8 7.7 | 75.5 ± 0.7 |
| Hungarian | 58.7 ± 1.5 100 | 55.9 ± 2.0 36.0 | 80.5 ± 0.9 7.5 | 78.2 ± 0.9 |

Za: Aha, David W., Dennis Kibler, Marc K. Albert (1991). Instance-Based Learning Algorithms. Machine Learning, 6, pp. 37-66.

# Przekleństwo wymiarowości

„Curse of dimensionality” [Bellman 1961]

- W celu dobrego przybliżenia funkcji (także stworzenia klasyfikatora) z danych:
  - The number of samples required per variable increases exponentially with the number of variables
  - Liczba obserwacji żądanych w stosunku do zmiennej wzrasta wykładniczo z liczbą zmiennych
- Oznacza to konieczność zdecydowanego wzrostu niezbędnych obserwacji przy dodawaniu kolejnych wymiarów



# Różne podejścia do redukcji wymiarów

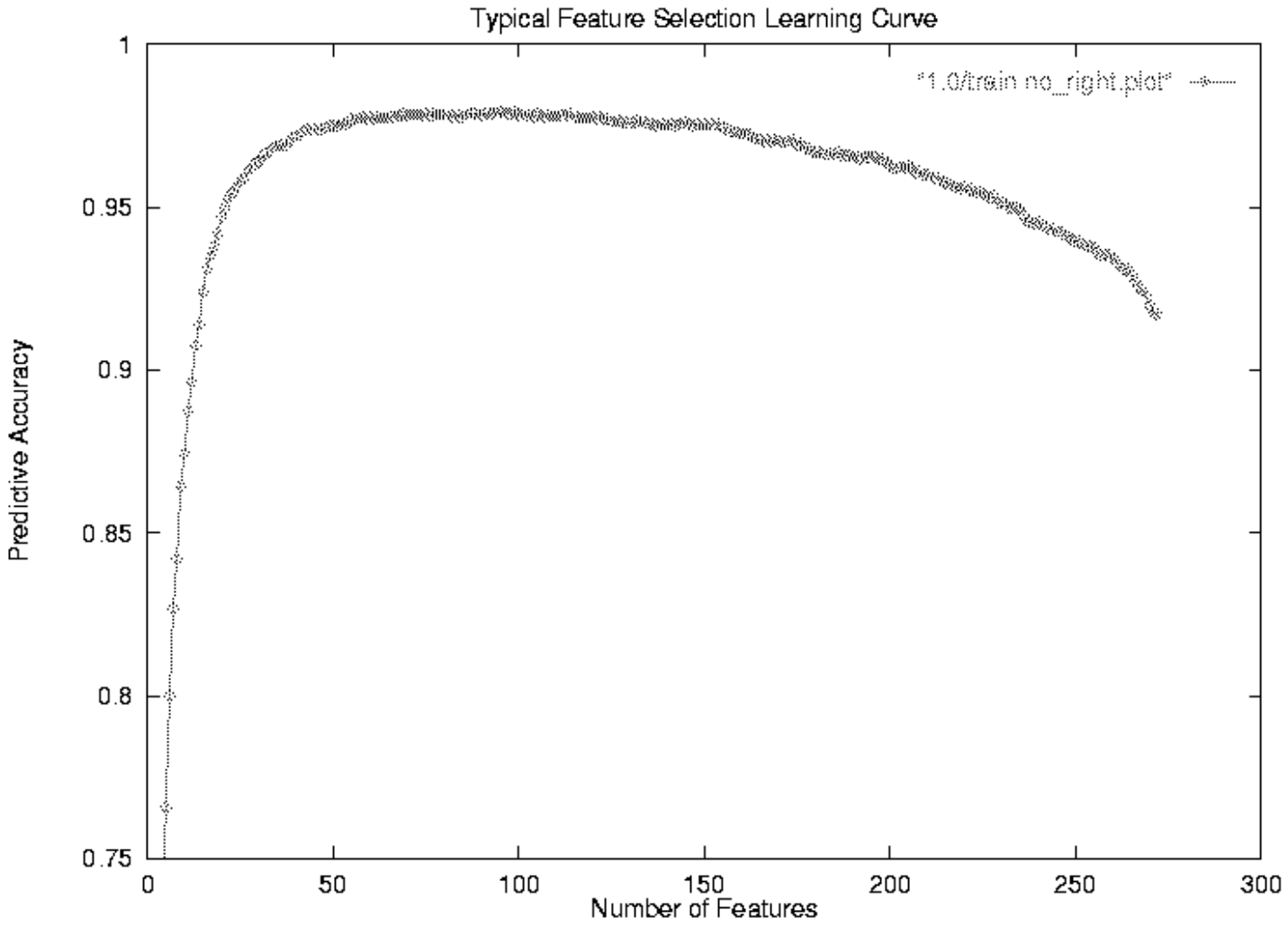
- **Selekcja cech**, zmiennych (cech, atrybutów, ..)
  - Wybierz podzbiór zmiennych $F' \subset F$
- **Konstrukcja** nowych zmiennych
  - Metody projekcji – nowe cechy zastępują poprzednie;
  - Statystyczne PCA, MDS vs. sztuczne sieci neuronowe (SOM, GN,…)

Więcej podczas innych przedmiotów:
Selekcja cech – za rok na Projekt eksploracji dancych – spójrz na
http://www.cs.put.poznan.pl/jstefanowski/PSE.html

# Redukcja rozmiarów danych – Selekcja atrybutów

- Dany jest *n* elementowy zbiór przykładów (obiektów). Każdy przykład *x* jest zdefiniowany na $V1 \times V2 \times Vm$ gdzie *Vi* jest dziedziną *i*-tego atrybutu. W przypadku nadzorowanej klasyfikacji przykłady zdefiniowane są jako <*x*,*y*> gdzie *y* określa pożądaną odpowiedź, np. klasyfikację przykładu.
- **Cel selekcji atrybutów:**
  - *Wybierz minimalny podzbiór atrybutów, dla którego rozkład prawdopodobieństwa różnych klas obiektów jest jak najbliższy oryginalnemu rozkładowi uzyskanemu z wykorzystaniem wszystkich atrybutów.*
- **Nadzorowana klasyfikacja**
  - *Dla danego algorytmu uczenia i zbioru uczącego, znajdź najmniejszy podzbiór atrybutów dla którego system klasyfikujący przewiduje przydział obiektów do klas decyzyjnych z jak największą trafnością.*

# Przykład działania K-NN nad wysoce wielowymiarowymi danymi



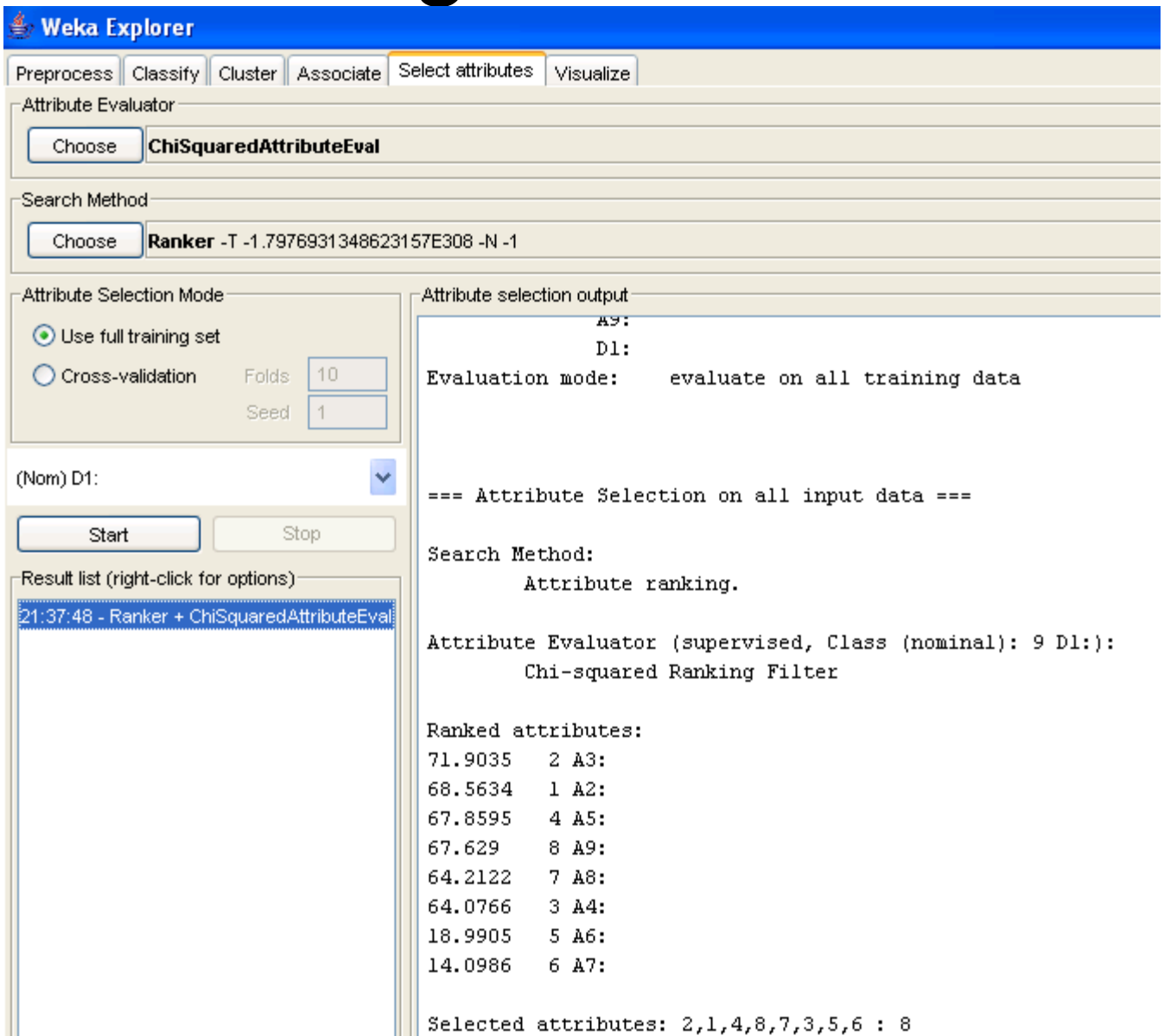
# Podejścia rankingowe

Wejście zbiór cech / atrybutów $F=f_1,f_2,\ldots,f_m$

- Dla każdej cechy $f \in F$ wylicz wartość miary oceny J(f)
- Uporządkuj cechy według wartości J(f).
- Weź k cech z góry rankingu albo cechy, dla których zachodzi $J(f) > \delta$, gdzie $\delta$ jest ustalonym progiem.

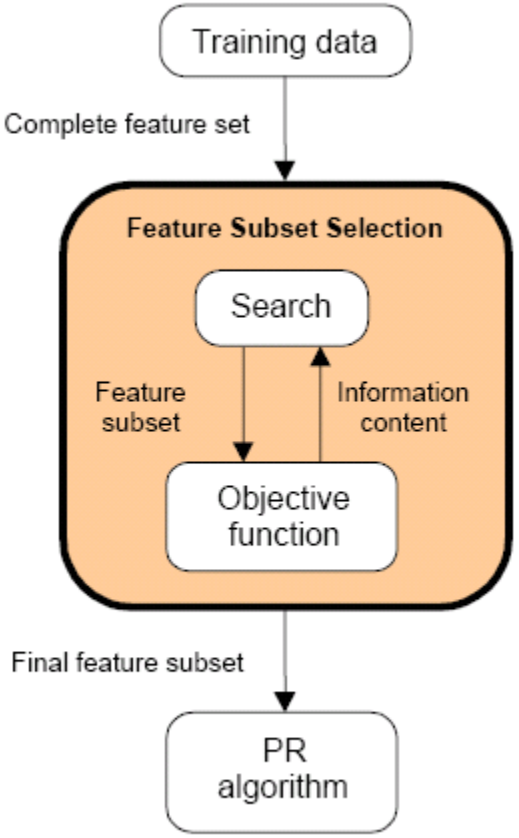
Wyjście: Podzbiór najlepiej ocenionych cech.

# Ranking with …? WEKA

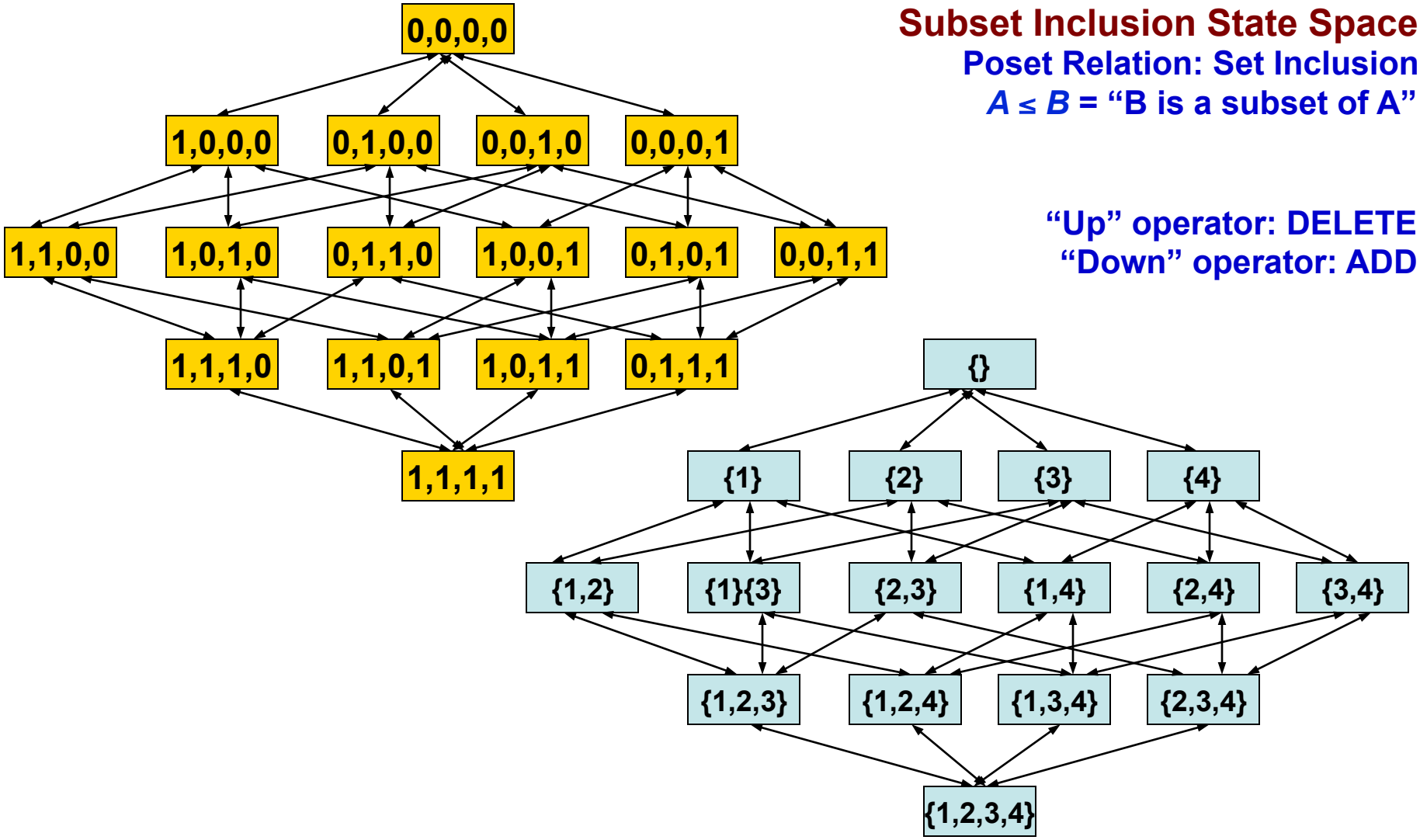


# Problem selekcji atrybutów (Feature Selection)

- Optymalne rozwiązanie – NP.
- *n* liczba atrybutów
  → przegląd przestrzeni z $2^n$ stanami.
- Przestrzeń podzbiorów częściowo uporządkowana (ang. lattice)
- Strategia przeszukiwania oraz
- Miara oceny *J*



# Przeszukiwanie przestrzeni podzbiorów




**Subset Inclusion State Space**

**Poset Relation: Set Inclusion**

***A* ≤ *B* = "B is a subset of A"**

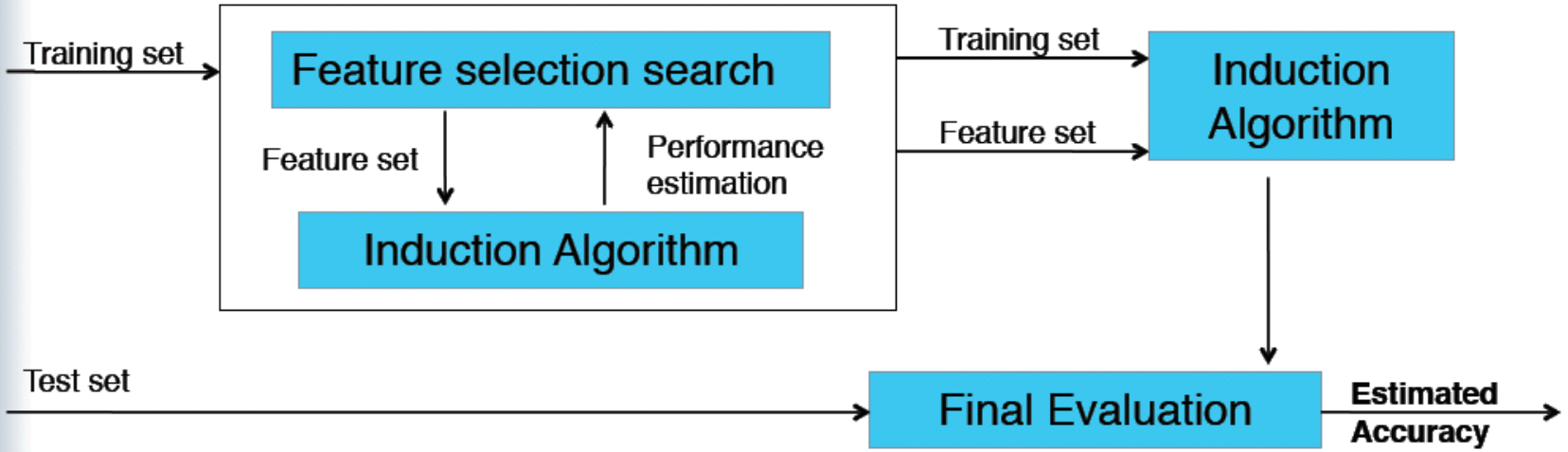
**"Up" operator: DELETE**

**"Down" operator: ADD**

# Wrapper [Kohavi et al.]



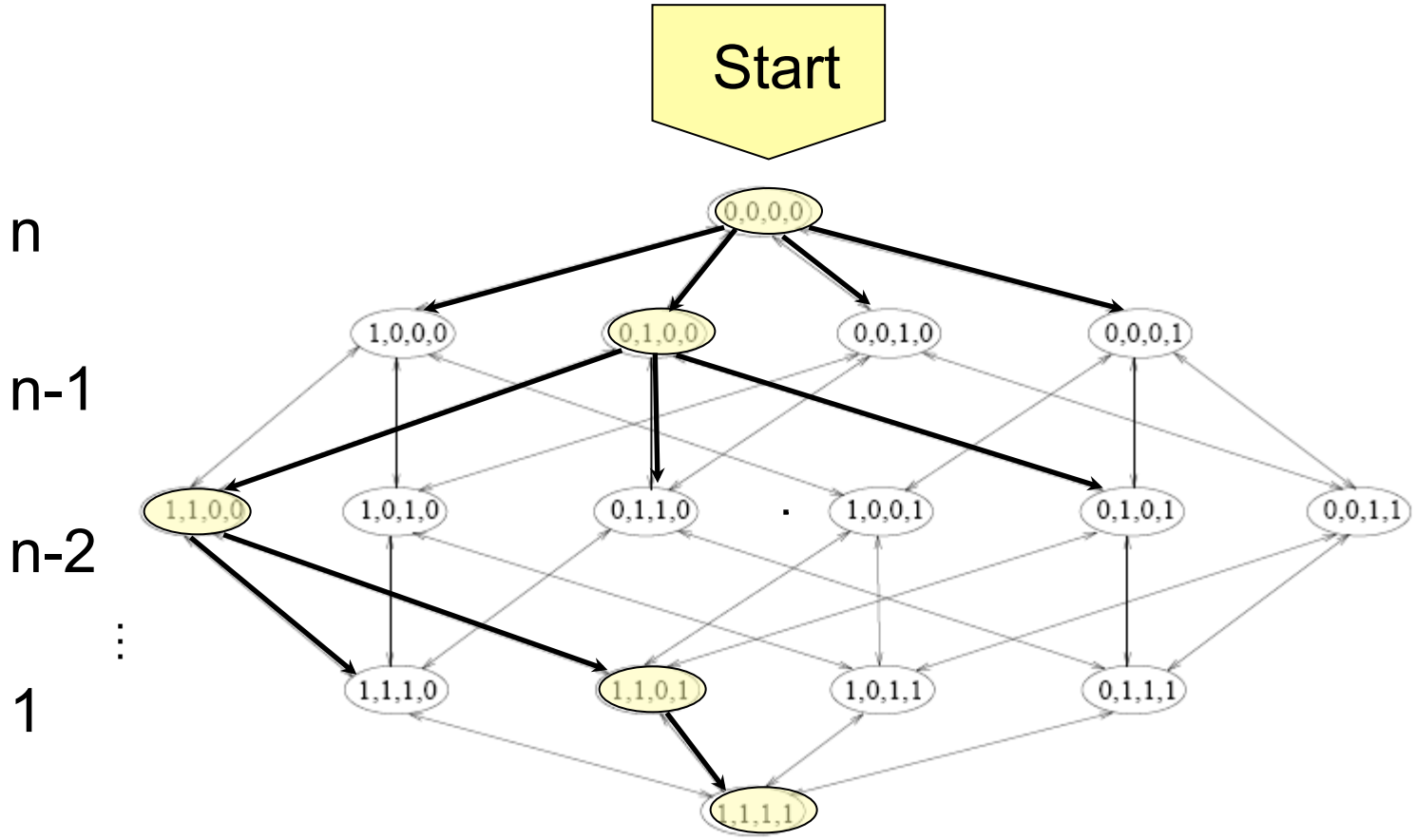
## Wrapper Approach




*Induction Algorithm is wrapped in the selection mechanism*

Strengths
- Takes bias of alg. into account
- Considers features in context

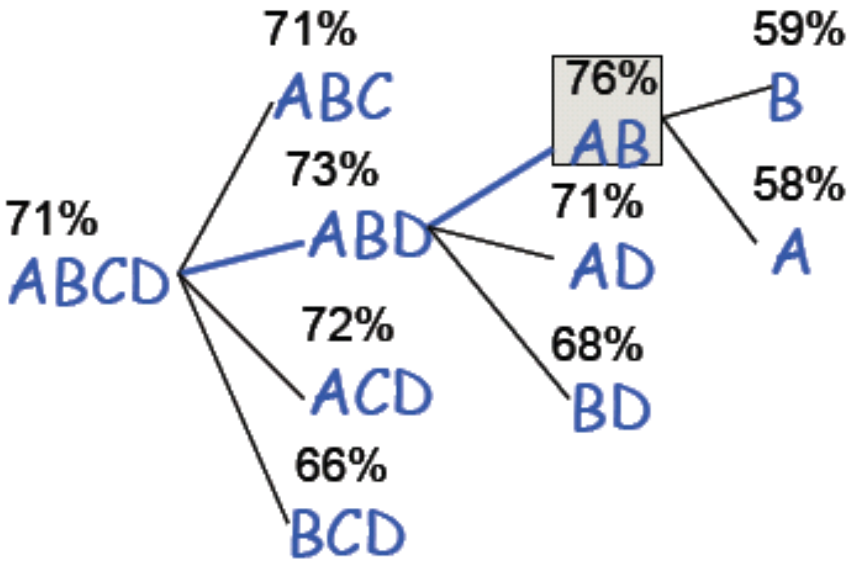
# Forward Selection (wrapper)




Also referred to as SFS: Sequential Forward Selection

# Wrapper – ocena trafności kieruje przeszukiwaniem

## Przykład




Backward Elimination:
Slightly better than FSS because it considers features in context

# Przykład użycia – dane medyczne

| Data set | Number of all features | Number of features selected by FBFS algorithm | Number of features selected by best-first algorithm | Accuracy for set of all features [%] | Accuracy for FBFS algorithm [%] | Accuracy for best-first choice algorithm [%] |
|---|---|---|---|---|---|---|
| HIST | 67 | 17 | 11 | 87.510.47 | 89.100.75 | 85.380.53 |
| COOC | 69 | 25 | 22 | 61.650.72 | 66.310.70 | 64.270.78 |
| DENS | 96 | 16 | 9 | 19.920.50 | 27.040.65 | 22.810.76 |

Tabela – ocena zdolności rozpoznawania klas obrazów (algorytm typu K-NN i wrapper)

Za: J.Jelonek, J.Stefanowski: Feature subset selection for classification of histological images, *Artificial Intelligence in Medicine*, vol. 9, 1997, 227-239.

# Kilka uwag o kosztach obliczeniowych

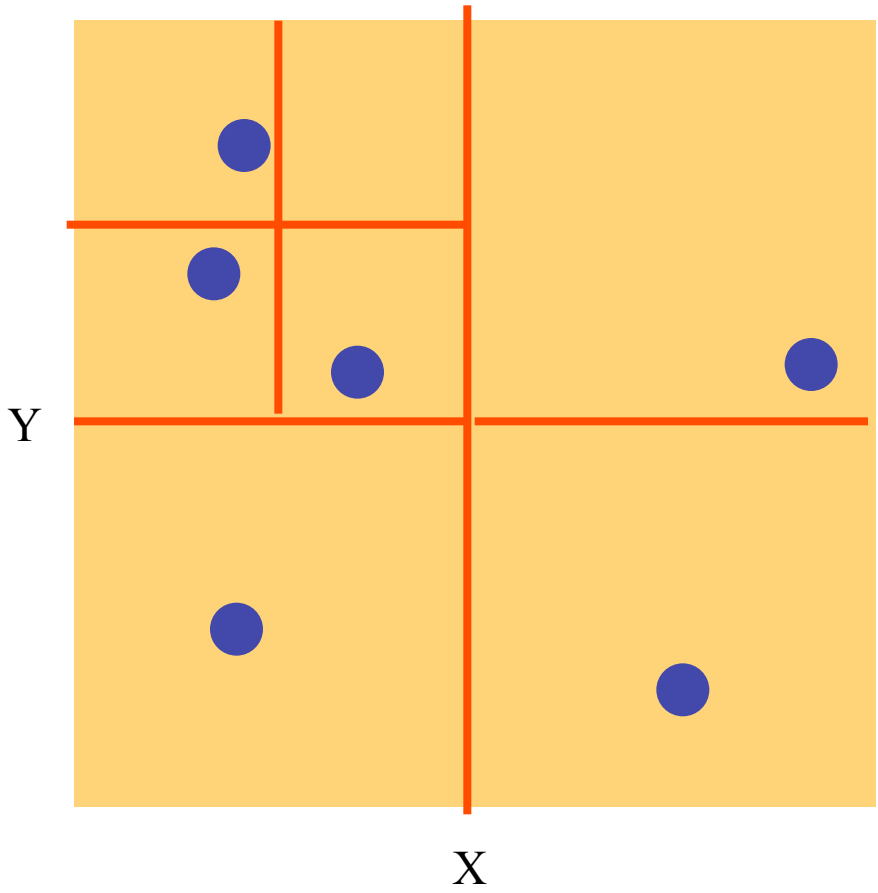
Załóżmy n przykładów opisanych d atrybutami

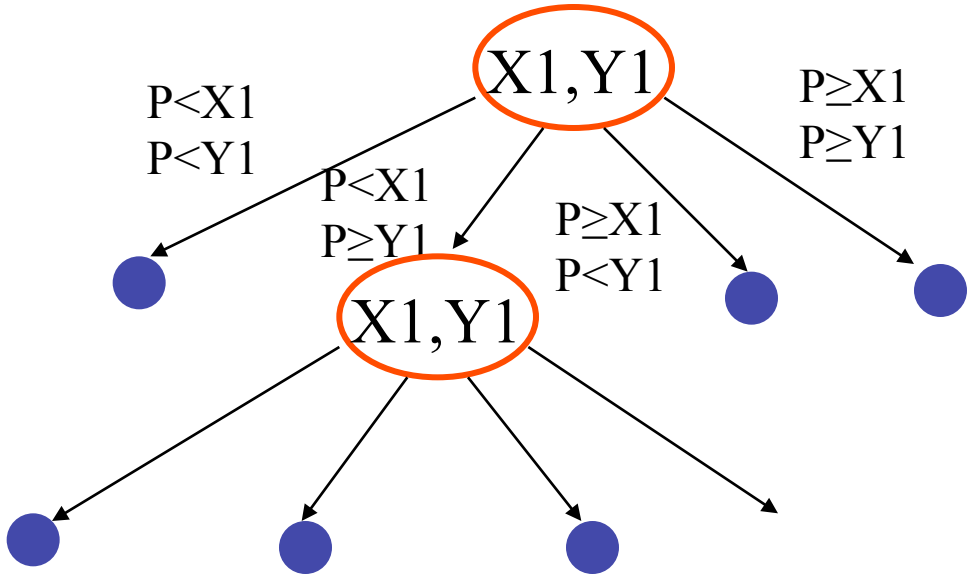
- O(d) to compute distance to one example
- O(nd) to find one nearest neighbor
- O(knd) to find k closest examples examples

Ocena complexity -- O(knd)

Dla danych o większych rozmiarach - nieakceptowalne

# Quad-tree – przyspiesz obliczania




http://www.wisdom.weizmann.ac.il/~mica/CVspring06/presentations/Dan_Tomer.ppt

# Niebezpieczeństwa wysokiej wielowymiarowości

- Trudności z klasyczną Euclidean measure:
  - Dane wysoko-wielowymiarowe
    - przekleństwo wielowymiarowości
  - Może prowadzić do zaskakujących sytuacji

| | | | |
|---|---|---|---|
| 1 1 1 1 1 1 1 1 1 1 1 0 | | 1 0 0 0 0 0 0 0 0 0 0 0 | |
| | vs | | |
| 0 1 1 1 1 1 1 1 1 1 1 1 | | 0 0 0 0 0 0 0 0 0 0 0 1 | |
| d = 1.4142 | | d = 1.4142 | |

- Rozwiązania?

odległość kątowa

$$p(x,y) = \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2 \sum_i y_i^2}}$$

# Inne miary odległości dla wysokowymiaroych przestrzeni

Odległość euklidesowa nie do zastowania -> alternatywy:
Miary “nie-euklidesowe” wykorzystujące inne własności punktów niż bezpośrednie położenie w przestrzeni

Popularne:

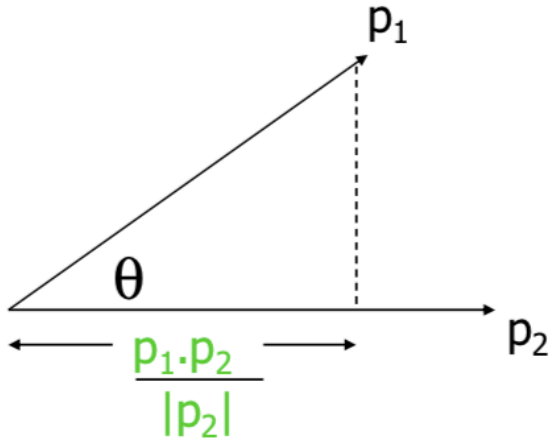
- Odległość kosinusowa / przestrzeń wektorowa -> kąt
- Tzw. Edit distance – ocena zmian w łańcuchach
- Odległość Hamming’a – najprostsza binarna ocena niezgodności
- Inne

# Odległość kosinusowa

Historzycznie wykorzystywana w przetwarzaniu tekstów oraz wyszukiwaniu informacji (rzadkie dane występowania termów)

Reprezentacja wektorowa − obserwacja -> wektor od zera (0,..,0) do położenia punktu w przestrzeni

Dwa punkty -> kąt między wektorami / kosinus odpowiada znormalizowanemu iloczynowi skalarnemu




$$d(p_1, p_2) = \theta = \arccos(p_1.p_2/|p_2||p_1|)$$

# “Lazy” vs. “Eager” Learning

- Instance-based learning: lazy evaluation
- Decision-tree and Bayesian classification: eager evaluation
- Key differences
  - Lazy method may consider query instance *xq* when deciding how to generalize beyond the training data *D*
  - Eager method cannot since they have already chosen global approximation when seeing the query
- Efficiency: Lazy - less time training but more time predicting
- Accuracy
  - Lazy method effectively uses a richer hypothesis space since it uses many local linear functions to form its implicit global approximation to the target function
  - Eager: must commit to a single hypothesis that covers the entire instance space

# Czy są pytania?

?

Kontakt:
Jerzy.Stefanowski@cs.put.poznan.pl