Klasyfikacja w oparciu o przykłady

(ang. instance based learning)
Wykład 2, 14/10/2003

Plan wykładu

- Wprowadzenie
- Metoda *k* najbliższych sąsiadów
  - Miary podobieństwa
  - Praktyczne problemy
    - Redukcja zbędnych przykładów
    - Redukcja szumu w danych
    - Wyznaczanie wag atrybutów
  - Lokalna regresja
- Naiwna metoda wnioskowania Bayesowskiego
- Sieć Bayesowska

Lazy vs. eager learning

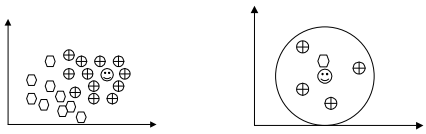
- Eager learning model:
  - Np. metody drzew decyzyjnych, reguł decyzyjnych, czy grupowania danych:

    „Konstruuje się jasny opis funkcji docelowej na podstawie przykładów trenujących„
- Lazy learning model:
  - Np. klasyfikacja w oparciu o przykłady:

    „Nie konstruuje się wcześniej opisu funkcji docelowej. Ta konstrukcja odbywa się w momencie klastrowania nowego obiektu”

Przykład

Dwuwymiarowy zbiór danych: każdy obiekt jest opisany dwoma atrybutami (x, y).
Są dwie klasy ⊕ lub ○



Algorytm najbliższego sąsiada (algorytm 1-NN)

- Parametr wejściowy:
  - Zbiór obiektów $P = \{<x_i, f(x_i)>\}$, gdzie f – funkcja docelowa, np. opis klas decyzyjnych.
  - x_q - obiekt do klasyfikowania
- Parametr wyjściowy:
  - wartość $f(x_q)$ (np. klasa decyzyjna, do której należy x_q)

Algorytm najbliższego sąsiada (algorytm 1-NN)

- **Ogólny schemat:**

  Krok 1: Poszukaj obiektu x_n najbliższego x_q.

  Krok 2: Wyznacz wartość $f(x_q)$ na podstawie wartości $f(x_n)$
- Zaleta: Prosty, szybki algorytm
- Wada: Nieodporny na szumy!!!

Algorytm *k* najbliższych sąsiadów (algorytm *k*-NN)

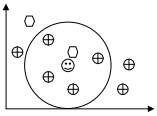
- **Ogólny schemat:**

  Krok 1: Poszukaj *k* najbliższych obiektów (sąsiadów) dla x_q.

  Krok 2: Głosuj wśród *k* najbliższych sąsiadów w celu wyznaczania klasy, do której należy x_q.

1-NN, decyzja jest ⬡

5-NN, decyzja jest ⊕



Zaleta: Bardziej odporny na szumy

Algorytm *k*-NN: Problemy

- Wyznaczanie miary podobieństwa (funkcji odległości) między obiektami.
- Głosowanie w celu wyznaczania klasy, do której należy nowy obiekt.
- Wyznaczanie liczby *k* najbliższych obiektów potrzebnych dla klasyfikowania nowego obiektu.

Miary podobieństwa

- Niech każdy obiekt *x* będzie zdefiniowany wektorem wartości: $<a_1(x),...,a_n(x)>$
- **Lokalna odległość** :
  $d_i(x,y) = |(a_i(x) - a_i(y)|$
- **Odległość między obiektami** :
  distance(x,y) = $F(d_1(x,y),..., d_n(x,y))$
  - **Odległość euklidesowa:**
    Eu-distance(x,y) = $[\Sigma_i (a_i(x) - a_i(y))^2]^{½}$
  - **Odległość miejska (Manhattan):**
    distance(x,y) = $\Sigma_i |a_i(x) - a_i(y)|$
- **Miary podobieństwa:**
  sim(x,y) = 1/(1+distance(x,y))

Metody głosowania

- **Zasada większościowa**:

$$\hat{f}(x_q) \leftarrow \underset{v \in V}{argmax} \sum_{i=1}^{k} \delta(v, f(x_i))$$

gdzie $\delta(x, y) = \begin{cases} 1 & jesli\ x = y \\ 0 & wpp. \end{cases}$

- **Ważona odległość**:

$$\hat{f}(x_q) \leftarrow \underset{v \in V}{argmax} \sum_{i=1}^{k} w_i \delta(v, f(x_i)) \qquad gdzie\ w_i = \frac{1}{1 + d(x_q, x_i)^2}$$

Wyznaczanie parametru *k*

- Jeśli *k* jest małe, algorytm nie jest odporny na szumy → jakość klasyfikacja jest niska.
- Jeśli *k* jest duże, koszt obliczenia jest większy → algorytm jest czasochłonny.
- Jak wybierać odpowiednią wartość *k*?
- **Idea:**
  - Wykonuj test typu kroswalidacji dla kilku różnych wartości *k*.
  - Wybierz wartość *k*, która daje najwyższą jakość klasyfikacji.

Praktyczne problemy w algorytmie *k*-NN

1. Dane zawierają szumy:

Rozwiązanie: usuwanie szumów

2. Atrybuty w różnym stopniu są ważne

Rozwiązanie:

wyznaczanie wag dla atrybutów
(lub selekcja istotnych atrybutów)

3. Funkcja docelowa nie jest dyskretna (wartości są rzeczywiste)

Rozwiązanie:

- modyfikacja algorytmu k-NN
- lokalna regresja

Usuwanie szumu (noisy examplars)

- **I stategia**:
  - wyznacz odpowiedni parametr *k*.
- **II strategia**:
  - Oceń jakość klasyfikacji każdego obiektu trenującego.
  - Usuń „słabe” obiekty.

Usuwanie szumu (c.d.)

- Dane: s_{min}, s_{max} – dolna i górna granica dokładności klasyfikacji.

- **Oceniać jakość klasyfikacji obiektu**:

  Krok 1. Wykonuj kroswalidację danych trenujących.

  Krok 2. Dla każdego obiektu *x* zanotuj procent obiektów dobrze klasyfikowanych s_x.

  Krok 3. Jeśli
  - $s_x < s_{min}$ to obiekt jest „słaby” (szum, trzeba usunąć)
  - $s_{min} < s_x < s_{max}$ obiekt jest „średnio dobry”

Wyznaczanie wag dla atrybutów

- **Motywacja**:
  - Niektóre atrybuty są ważniejsze niż inne
  - Niektóre atrybuty są ważne dla jednej klasy ale nie są ważne dla innej klasy.
- Jeśli w_1, w_2,...,w_n – wagi atrybutów to

$$distance(x, y) = \sqrt{w_1^2(a_1(x) - a_2(x))^2 + ... + w_n^2(a_n(x) - a_n(x))^2}$$

Wyznaczanie wag dla atrybutów (c.d.)

- **I model**: Każda klasa jest związana z jednym wektorem wag.
- **II model**: Jeden wspólny wektor wag dla wszystkich klas.
- **Idea**: Układ wag jest obliczony na podstawie wyniku klasyfikacji obiektów w zbiorze trenującym.

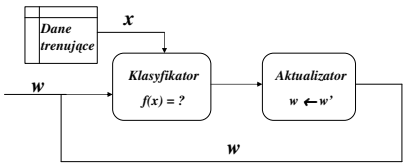
- **Algorytm sekwencyjnego poprawiania!**

Algorytm sekwencyjnego poprawiania

- **Ogólny schemat**

  Krok 1. zacznij od dowolnego wektora wag $w = [w_1,...,w_n]$.

  Krok 2. aktualizuj *w*, kiedy nowy obiekt trenujący *x* jest klasyfikowany.



Aktualizacja wektora wag

- Wejście: zbiór obiektów *P*
- Wyjście: wektor wag ***w***

Krok 1. Dla $x \in P$ klasyfikuj *x* za pomocą pozostałych obiektów

Krok 2. Dla *x*, znajdź najbliższy obiekt *y* (wśród obiektów w zbiorze trenującym)

Krok 3. Dla każdego atrybutu a_i wyznacz $|a_i(x)-a_i(y)|$

Krok 4. Jeśli klasyfikacja jest prawidłowa to zwiększ wagę w_i , wpp. zmniejsz wagę w_i o Δw_i

(Δw_i jest odwrotnie proporcjonalne do $|a_i(x)-a_i(y)|$)

Krok 5. Jeśli wszystkie obiekty nie zostały testowane powróć do **Krok 1.**

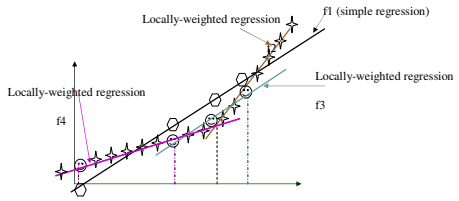
Wyznaczanie wartości rzeczywistej funkcji docelowej

- Modyfikacja algorytmu *k*-NN:

$$\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^{k} w_i \, f(x_i)}{\sum_{i=1}^{k} w_i} \qquad gdzie\ w_i = \frac{1}{1 + d(x_q, x_i)^2}$$

- Lokalna regresja:
  Znajdź lokalną aproksymację dla funkcji *f*, która najbardziej pasuje do obiektów w otoczeniu x_q.
- Liniowa regresja
- Kwadratowa regresja

Lokalna regresja



Trenujące dane

Wartość *f* obliczona za pomocą globalnej regresji

Wartość *f* obliczona za pomocą lokalnej regresji

Metody Bayesowskie

- Naiwna metoda klasyfikacji Bayesowkiej
- Sieć Bayesowka
- Kombinacja z wiedzą dziedzinową

Podstawowa teoria

- Reguła Bayesowska: $P(h \mid D) = \frac{P(D \mid h) P(h)}{P(D)}$
  gdzie
  - $P(h)$ = prawdopodobieństwo zajścia hipotezy h
  - $P(D)$ = prawdopodobieństwo otrzymania zbioru treningowego D
  - $P(h|D)$ = prawdopodobieństwo h podwarunkiem, że D jest dany
  - $P(D|h)$ = prawdopodobieństwo D przy założeniu, że h zachodzi

Zasada Bayesowkiego uczenia się:
Szukanie najbardziej prawdopodobnej hipotezy mając zadany zbiór treningowy: (maksymalizacja hipotezy aposteriori h_{map})

$$\begin{aligned} h_{map} &= \max_{h \in H} P(h \mid D) \\ &= \max_{h \in H} \frac{P(D \mid h) P(h)}{P(D)} \\ &= \max_{h \in H} P(D \mid h) P(h) \end{aligned}$$

Podtawowe twierdzenia probabilistyczne

- Prawdopodobieństwo koniunkcji dwóch zdarzeń:

$$P(A, B) = P(A \mid B) P(B) = P(B \mid A) P(A)$$

- Prawdopodobieństwo sumy dwóch zdarzeń:

$$P(A + B) = P(A) + P(B) - P(AB)$$

- Wzór na prawdopodobieństwo całkowite:
  jeśli zdarzenia A_1, …., A_n tworzą rozłączny podział przestrzeni probabilistycznej, to:

$$P(B) = \sum_{i=1}^{n} P(B \mid A_i) P(A_i)$$

Przykład

- Czy pacjent jest chory na raka?

Pacjent poddał testowi na obecność pewnego raka i dostał pozytywny wynik. Wynik testu jest
•prawidłowy (pozytywny) w 98% wsród chorujących na raka i
•prawidłowy (negatywny) w 97% wsród tych, którzy nie chorują na tego raka.
•Poza tym, 0.8% populacji choruje na badanego raka.

$$P(cancer) = .008, P(\neg cancer) = .992$$
$$P(+ \mid cancer) = .98, P(- \mid cancer) = .02$$
$$P(+ \mid \neg cancer) = .03, P(- \mid \neg cancer) = .97$$
$$P(cancer \mid +) = \frac{P(+ \mid cancer) P(cancer)}{P(+)}$$
$$P(\neg cancer \mid +) = \frac{P(+ \mid \neg cancer) P(\neg cancer)}{P(+)}$$

Naiwna metoda Bayesa

- Załóżmy, że uczymy się funkcji celu f: X→V, gdzie każdy obiekt jest opisany wektorem <a_1, a_2, …., a_n>.
- Najbardziej prawdopodobna wartość f(x) wynosi:

$$v = \max_{vj \in V} P(v_j \mid a_1, a_2 .... a_n)$$
$$= \max_{vj \in V} \frac{P(a_1, a_2 .... a_n \mid v_j) P(v_j)}{P(a_1, a_2 .... a_n)}$$
$$= \max_{vj \in V} P(a_1, a_2 .... a_n \mid v_j) P(v_j)$$

- Naiwne założenie: „atrybuty są warunkowo niezależne”, tzn.

$$P(a_1, a_2 ... a_n \mid v_j) = \prod_i P(a_i \mid v_j)$$

Przykład: naiwna metoda

Zgadnij, czy odbędzie się gra w tenisa w dniu o warunkach pog.: <sunny, cool, high, strong> na podstawie nast. Zbioru danych:

Outlook	Temperature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

$$p(y)\,p(sun \mid y)\,p(cool \mid y)\,p(high \mid y)\,p(strong \mid y) = .005$$
$$p(n)\,p(sun \mid n)\,p(cool \mid n)\,p(high \mid n)\,p(strong \mid n) = .021$$

przykład

Outlook	Temperature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

P(p) = 9/14
P(n) = 5/14

outlook	
P(sunny|p) = 2/9	**P(sunny|n) = 3/5**
P(overcast|p) = 4/9	**P(overcast|n) = 0**
P(rain|p) = 3/9	**P(rain|n) = 2/5**
temperature	
P(hot|p) = 2/9	**P(hot|n) = 2/5**
P(mild|p) = 4/9	**P(mild|n) = 2/5**
P(cool|p) = 3/9	**P(cool|n) = 1/5**
humidity	
P(high|p) = 3/9	**P(high|n) = 4/5**
P(normal|p) = 6/9	**P(normal|n) = 2/5**
windy	
P(true|p) = 3/9	**P(true|n) = 3/5**
P(false|p) = 6/9	**P(false|n) = 2/5**

Algorytm „Naive Bayes”

1. Uczenie(zbiór przykładów)
  for (każda klasa decyzyjna v_i)
    oszacuj $P(v_j)$
    for (każda wartość a_i na atrybucie a)
      oszacuj $P(a_i \mid v_j)$

2. Klasyfikacja nowego obiektu(x)

$$v = \max_{v_j \in V} P(v_j) \prod_{a_i \in x} P(a_i \mid v_j)$$

typowe oszacowanie dla $P(a_i \mid v_j)$

$$P(a_i \mid v_j) \leftarrow \frac{n_c + mp}{n + m}$$

Gdzie:
n: liczba przykładów z klasy v_j;
n_c: liczba przykładów z $a=a_i$,
p: wstępne oszacowanie dla $P(a_i|v_j)$
m: waga przekonań dla p

Sieć Bayesowska

- Założenie „naive Bayes” jest zbyt ograniczone i prymitywne!
- Bez niego, obliczenia nie są wykonalne
- Sieć Bayesowska:
  - opisuje *warunkową niezależność między zbiorami atrybutów*, uwzględniając
    - *wiedzę ekspercką* o zależności między atrybutami i
    - *dane treningowe.*
  - DAG (direct acyclic graph)

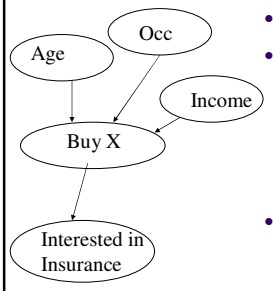
Sieć Bayesowska

- Jest to acykliczny graf skierowany, gdzie
  - Wierzchołki: atrybuty
  - Krawędzie: zależność
  - Kierunki krawędzi: relacja przyczynowo-skutkowa
  - Do każdego atrybutu A, dołączona jest tablica prawdopodobieństw

    P (A | B_1, …., B_n),

    gdzie B_1, …., B_n są bezpośrednimi poprzednikami atrybutu A w tym grafie

Przykład sieci



- Age, Occupation oraz Income decyduje, czy klient kupuje dany produkt.
- Jeśli klient kupuje produkt, to jego zainteresowanie ubezpieczeniem (interest in insurance) jest niezależne od Age, Occupation, Income.

P(Age, Occ, Inc, Buy, Ins) =
= P(Age)P(Occ)P(Inc)
P(Buy|Age,Occ,Inc)P(Int|Buy)

- **Stan wiedzy:** przy zadanej strukturze i warunkowych prawdopodobieństwach, istniejące algorytmy mogą wnioskować o atrybuty symboliczne i dyskretyzowane atrybuty ciągłe.

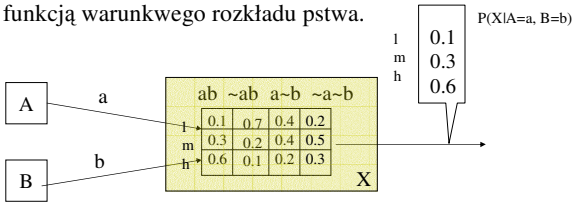
Podstawowy wzór:

$$P(x_1,....x_n \mid M) = \prod_{i=1}^{n} P(x_i \mid Pa_i, M)$$

$$Pa_i = parent(x_i)$$

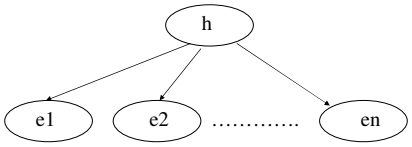
Wierzchołki jako funkcje

Każdy wierzchołek w sieci Bayesowskiej jest funkcją warunkwego rozkładu pstwa.



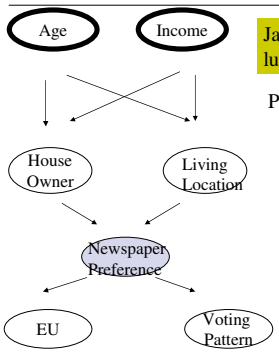
•wejście: wartość rodziców
•wyjście: rozkład pstwa własnych wartości

Przypadek szczególny: „naive Bayes"



P(e1, e2, ……en, h) = P(h) P(e1 | h) …….P(en | h)

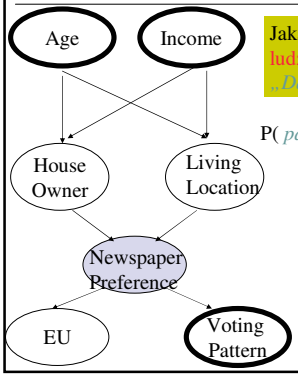
Wnioskowanie o sieci Bayesowskie:



Jaka jest szansa, że bogaci i starzy ludzie kupują *„Sun"*?

P(*paper = Sun* | Age>60, Income > 60k)

Wnioskowanie o sieci Bayesowskie:

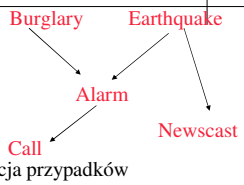


Jaka jest szansa, że bogaci i starzy ludzie głosujący na partię X kupują *„Daily Mail"*?

P(*paper = DM* | Age>60, Income > 60k, Vote = X)

Uczenie Bayesowskie

B	E	A	C	N
~b	e	a	c	n
b	~e	~a	~c	n
…….....				



Dane: pełna lub częściowa obserwacja przypadków
Szukane: parametry i struktura

Metody uczenia się:

EM (Expectation Maximisation)

-Uzupełnić brakujące dane za pomocą bieżącej aproksymacji parametrów;
-Aproksymować parametry za pomocą wypełnionych danych

Gradient Ascent Training

Gibbs Sampling (MCMC)

Bibliografia

- Lenz M., Bartsch-Sporl B., Burkhard H., Wess S. (1998). *Case-based resoning technology. From fundations to applications*. Prinfer-Verlag Berlin Heidelberg. LNAI, Vol. 1400.
- Aha D. (1992). *Tolerating noisy, irrelevant, and novel attributes in instance-based learning algorithms*. International Journal of Man-Machine Studies 36(2): 267-287.