Algorytmy Eksploracji Danych

Marcin Korzeń

Zakład Metod Sztucznej Inteligencji,
Wydział Informatyki, Zachodniopomorski Uniwersytet Technologiczny

e-mail: `mkorzen@wi.zut.edu.pl`
konsultacje: Czwartki N $10^{15} - 11^{45}$, Piątki P $8^{30} - 10^{00}$
pok.: 107a

Literatura

1. J. Koronacki, J. Mieleniczuk, *Statystyka dla kierunków systemy uczące*, WNT,2006
2. J. Koronacki, J. Ćwik, *Statystyczne systemy uczące*, WNT,2005
3. D. Hand, H. Mannila, P. Smyth, *Eksploracja danych*, WNT,2005
4. Ian H. Witten, Eibe Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, 2005

Część I

Wprowadzenie

Co warto wiedzieć

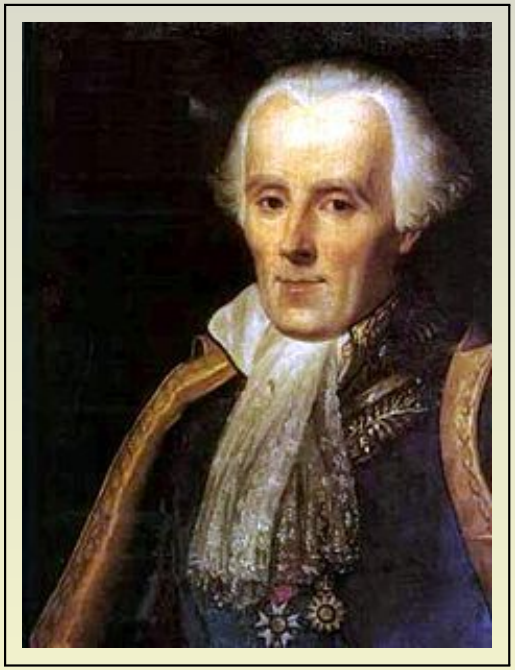
- Podstawy statystyki i rachunku prawdopodobieństwa (średnia, wariancja, i inne podstawowe statystyki, rozkład normalny – identyfikacja parametrów, korelacja regresja, testowanie hipotez, niezależność)
- Podstawy algebry liniowej (przestrzeń liniowa, wektor, iloczyn skalarny, wartości i wektory własne, działania na macierzach, metoda najmniejszych kwadratów)
- Podstawy baz danych (struktury danych, relacja, klucz, indeks)

Trochę historii

- 1763, twierdzenie Bayes'a
- 1774-1812, P. Laplace

  (nierozróżnialność, prawo następstw,...)
- 1801, C. F. Gauss, MNK, *Ceres*
- 1922, R. A. Fisher, Metoda największej wiarygodności
- 1927, K. Pearson, E. Peasron, J. S. Neyman, wnioskowanie statystyczne
- 1950, Metody bayesowskie

  (Wnioskowanie bayesowskie, sieci bayesowskie, klasyfikator Bayesa, filtry Bayesa, regularyzacja Bayesowska, metoda MAP)



Gromadzenie danych

Dawno dawno temu

- 1662, 1838 – pierwsze tablice trwania życia, początki demografii i ubezpieczeń
- 1790 pierwszy spis ludności w USA (3 929 326 ludności w tym 697 681 niewolników)
- 1865, Mendel przebadał ok 29 000 próbek roślin (grochu)
- 1936, Zbiór irysów 150 przykładów, (R. A. Fisher analiza dyskryminacyjna)

Współcześnie

- 1998, Wal-Mart 11 TB, $20 \cdot 10^6$ transakcji detalicznych dziennie
- Zapis ludzkiego genomu $3.3 \cdot 10^9$ nukleotydów
- 2008, `www.google.com` zawierał 10^{12} unikalnych adresów url, $63 \cdot 10^9$ stron

Kilka pojęć

- Statystyka
- Uczenie maszynowe (ang. *Machine Learning*)
- Eksploracja danych (ang. *Data mining*)
- Wydobywanie wiedzy z baz danych (ang. *Knowledge Discovery in Databases – KDD*)
- Inteligencja Biznesowa ? (ang. *Business Intelligence*)
- Systemy wspomagania decyzji (ang. *Decision Support System – DSS*)

Uczenie maszynowe, Eksploracja danych i Statystyka

Uczenie maszynowe

- Większy nacisk na rozwiązywanie konkretnego zagadnienia
- Trochę bliżej statystyki

Eksploracja danych

- Większy nacisk na sam algorytm
- Trochę bliżej informatyki

Statystyka

- Większy nacisk na wsparcie teoretyczne

Generalnie synonimy i trudno wyznaczyć granice

Generalna zasada

Podstawą każdego wnioskowania (modelowania) jest czynienie założeń

Nie ma modeli, które byłyby poprawne w każdej sytuacji

Zbiór danych

Tabela z danymi – formalnie para (U, A'),

- $U = \{u_1, \ldots, u_p\}$ - zbiór obiektów (przykładów, próbek, rekordów)
- $A' = \{a_1, \ldots, a'_n\}$ - zbiór atrybutów (zmiennych, kolumn, cech)
- $A' = A \cup D$ czasami wyróżniamy atrybut decyzyjny

Zbiory danych

Ważne odstępstwa od schematu (U, A)

- relacja lub kilka relacji w bazie danych
- dyskografia, wideoteka, album ze zdjęciami
- zbiór dokumentów tekstowych
- `www`
- systemy informacji przestrzennej ((ang. *spatial*))
- i wiele innych ...

Zbiór danych, a próba losowa

i.i.d. szczególny ale dość częsty przypadek danych (ang. *independent and identically-distributed*)

stacjonarność rozumiana jako niezależność rozkładu od czasu

niestacjonarność zależność rozkładu od historii i czasu

próbki zależne proces losowy, szereg czasowy, strumień danych (ang. *data stream*)

Problemy z danymi

- Danych może być dużo (zarówno obiektów jak i atrybutów; Co jest gorsze ?)
- Dane numeryczne, symboliczne, opis słowne, teksty, obrazy itp...
- Selekcja cech jest często krytyczna – skróty z danych
- Przyrost ilości danych
- Braki w danych

Wnioski

- Sposób gromadzenia powinien ułatwiać: wstawianie, kasowanie, wyszukiwanie (DBMS)
- Sposób gromadzenia powinien ułatwiać zadania analizy danych
- Jeżeli rozmiar na to pozwala należy używać najlepszych znanych rozwiązań
- Jeżeli rozmiar jest zbyt duży należy stosować szybkie rozwiązania przybliżone lub próbkowanie

Rodzaje pamięci

Rejestry – czas dostępu rzędu kilku taktów zegara

Rodzaje pamięci

Rejestry – czas dostępu rzędu kilku taktów zegara

Cache – 10[ns]

Rodzaje pamięci

Rejestry – czas dostępu rzędu kilku taktów zegara

Cache – 10[ns]

Operacyjna(RAM) – 100[ns]

Rodzaje pamięci

Rejestry – czas dostępu rzędu kilku taktów zegara

Cache – 10[ns]

Operacyjna(RAM) – 100[ns]

Twardy dysk – 10[ms]

Rodzaje pamięci

Rejestry – czas dostępu rzędu kilku taktów zegara

Cache – 10[ns]

Operacyjna(RAM) – 100[ns]

Twardy dysk – 10[ms]

Kolejny poziom – 1[s] i więcej

Jakie są różnice pomiędzy dyskiem, a RAM ?

Małe i duże zbiory danych

Mały - zbiór danych mieści się w pamięci operacyjnej komputera

Duży - zbiór danych **nie** mieści się w całości w pamięci operacyjnej

Małe i duże zbiory danych

Mały - zbiór danych mieści się w pamięci operacyjnej komputera

Duży - zbiór danych **nie** mieści się w całości w pamięci operacyjnej

Mały - wybrany algorytm eksploracji wykonuje się szybko

Duży - wybrany algorytm eksploracji wykonuje się wolno

Wnioskowanie z danych

Statystyka

- Plan eksperymentu
- Organizacja danych bez znaczenia
- Hipotezy stawia ekspert
- Testujemy pojedyncze hipotezy

Data mining

- Dane spływają na bieżąco
- Sposób organizacji danych krytyczny (KNN)
- Dane dostarczają hipotez
- Testowane są tysiące hipotez jednocześnie (Apriori)

Eksploracja danych a Statystyka

W jaki sposób liczymy średnią ?

- tak ?

$$\overline{x}_n = \frac{x_1 + x_2 + \ldots + x_n}{n}$$

Eksploracja danych a Statystyka

W jaki sposób liczymy średnią ?

- tak ?

$$\overline{x}_n = \frac{x_1 + x_2 + \ldots + x_n}{n}$$

- może tak ?

$$\overline{x}_n = \frac{(x_1 + \ldots + x_{\lfloor n/2 \rfloor}) + (x_{\lfloor n/2 \rfloor + 1} + \ldots + x_n)}{n}$$

Eksploracja danych a Statystyka

W jaki sposób liczymy średnią ?

- tak ?

$$\overline{x}_n = \frac{x_1 + x_2 + \ldots + x_n}{n}$$

- może tak ?

$$\overline{x}_n = \frac{(x_1 + \ldots + x_{\lfloor n/2 \rfloor}) + (x_{\lfloor n/2 \rfloor + 1} + \ldots + x_n)}{n}$$

- czy może tak ?

$$\overline{x}_{i+1} = \frac{1}{i+1}\left(\overline{x}_i \cdot i + x_{i+1}\right)$$

Narzędzia analizy danych

Matlab Optimization Toolbox, Neural Network Toolbox, Bayes Net Toolbox,

`http://www.mathworks.com/`

`http://www.cs.ubc.ca/~murphyk/Software/BNT/bnt.html`

WEKA `http://www.cs.waikato.ac.nz/~ml/weka/`

R, Rattle `http://togaware.com/`

Python, orange `http://www.ailab.si/orange/`

Python z pakietem Orange

Zalecana konfiguracja

python 26.x, Python 2.7.x Biblioteki: NumPy, SciPy, matplotlib, orange

```
import orange
data = orange.ExampleTable("ionosphere")
logistic = orange.LogRegLearner(data)
for ex in data[:-1]:
    print ex.getclass(), logistic(ex)
for v in data.domain.variables:
    if logistic.P[i]<0.05/30:
        print v, logistic.wald_Z[i], logistic.P[i]
```

Zalecana konfiguracja

R, rattle

```
install.packages("rattle", dependencies=TRUE)
library("foreign")
setwd("[WORKPATH]")
d <- read.csv("dane.csv")
d <- read.arff("dane.arff")
```

Matlab

```
load dane.txt
dane = csvread('dane.csv')
[dataSet,PA,nz]=readArff('dataset.arff');
```

Przykłady

Publiczne repozytoria

- http://mlr.cs.umass.edu/ml/datasets.html
- http://weka.sourceforge.net/wiki/index.php/Datasets
- http://lib.stat.cmu.edu/
- http://www.inf.ed.ac.uk/teaching/courses/dme/html/datasets0405.html

Komponenty zadania eksploracji danych

1. Rodzaj zadania eksploracji
2. Struktura i parametry modelu
3. Funkcja oceny modelu (może mierzyć jakość i złożoność modelu)
4. Metoda przeszukiwania (zarówno dobór struktury jak i optymalizacja funkcji oceny)
5. Sposób zarządzania danymi (zwykle pomijany, dla dużych zbiorów krytyczny, zwykle pamięć operacyjna lub DBMS)

Rodzaje zadań eksploracji danych

Identyfikacja rozkładu podstawe zadanie statystyki

Klasyfikacja przyporządkowanie obiektu do jednej z klas decyzyjnych

Regresja prognozowanie wartości zmiennej ciągłej

Wyszukiwanie reguł i wzorców wyszukiwanie zależności/warunkowych zależności w danych

Selekcja i ekstrakcja atrybutów

Klasteryzacja podział zbioru danych na grupy obiektów podobnych

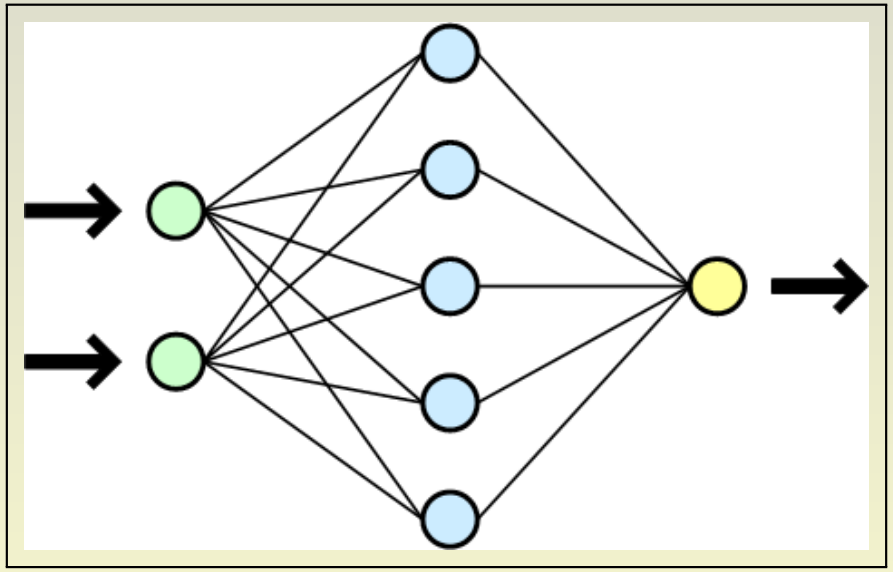
Struktura modelu - przykłady

Model liniowy

$$y(a, x) = a_0 + \sum_{i=1}^{n} a_i x_i$$

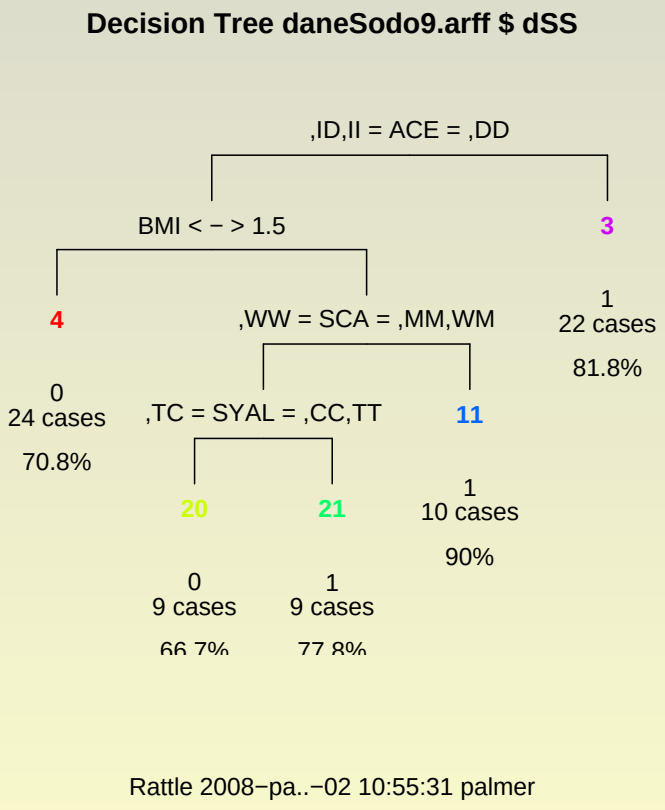
Struktura modelu - przykłady

Sieć neuronowa



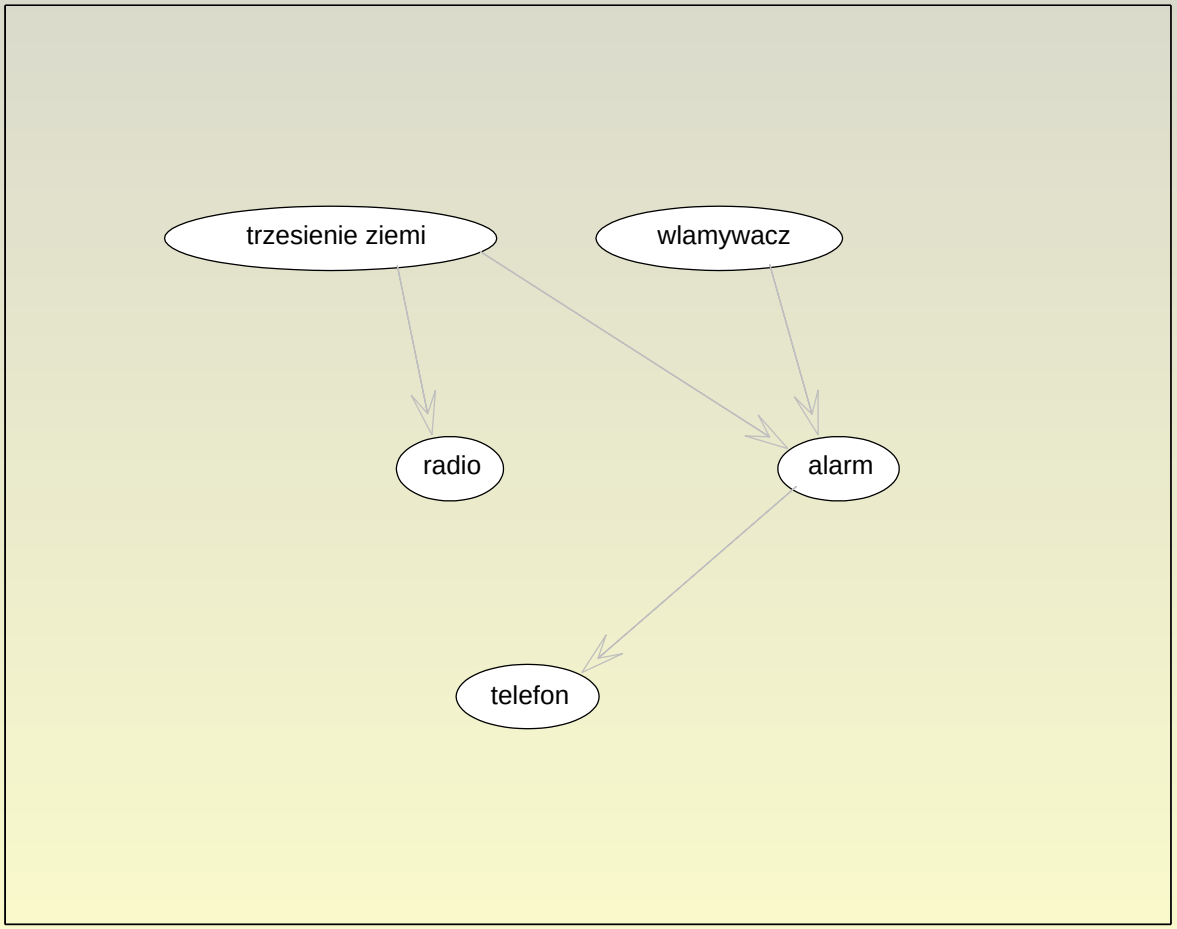
Struktura modelu - przykłady

Drzewo klasyfikacyjne



Struktura modelu - przykłady

Sieć bayesowska - graficzny model zależności



Funkcja oceny - przykłady

błąd średniokwadratowy ocena modelu linowego jak i innych

dokładność klasyfikacji miara jakości klasyfikatora

błąd kwantyzacji mierzy jakość grupowania danych

BIC miara uwzględnia złożoność modelu (ang. *Bayesian information criterion*)

Metoda przeszukiwania - przykłady

Programowanie kwadratowe bez lub z ograniczeniami (metoda najmniejszych kwadratów, SVM)

Metody gradientowe (sieci neuronowe, regresja logistyczna)

Przeszukiwanie wszerz/ w głąb, (z przycinaniem) (wyszukiwanie reguł i wzorców)

Przeszukiwanie zachłanne w wykorzystaniem heurystyki (drzewa decyzyjne)

Część II

Przetwarzanie wstępne i wizualizacja

Rozkład normalny

Jednowymiarowy - funkcja gęstości

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Wielowymiarowy - funkcja gęstości

$$f(\mathbf{x}) = \frac{1}{\sqrt{2\pi^n \det(\Sigma)}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1} (\mathbf{x}-\mu)\right)$$

(gdzie: μ – jest wektorem wartości oczekiwanych wektora losowego, Σ – jest macierzą kowariancji.)

Zadania

- Normalizacja
- Dyskretyzacja
- Binaryzacja
- Analiza składowych głównych
- Selekcja atrybutów
- Uzupełnianie braków
- Wizualizacja danych

Normalizacja

- Zachowująca zero

$$x_{norm} = \frac{x}{\max|x|}$$

- Skalowanie

$$x_{norm} = \frac{x - \min(x)}{\max(x) - \min(x)}$$

- Standaryzacja zmiennej (Z-score)

$$x_{norm} = \frac{x - \operatorname{mean} x}{\operatorname{std} x}$$

- Standaryzacja wektora zmiennych ? – (próba wielowymiarowa)

Dyskretyzacja

Zastąpienie atrybutu ciągłego atrybutem dyskretnym (przyjmującym skończoną liczbę wartości) niosącym zbliżoną informację do oryginału. Wyróżniamy zasadniczo dwa podejścia do dyskretyzacji:

bez nadzoru uwzględniana jest jedynie informacja rozkładzie dyskretyzowanego atrybutu (klasteryzacja)

nadzorowane uwzględniana jest informacja o rozkładzie łącznym atrybutu dyskretyzowanego i decyzji

Binaryzacja

Zastąpienie atrybutu dyskretnego – przyjmującego m wartości, m atrybutami binarnymi, jedynkującymi się dla odpowiednich wartości:

color
green
red
red
green
blue

$\rightarrow$

color=green	color=red	color=blue
1	0	0
0	1	0
0	1	0
1	0	0
0	0	1

Analiza składowych głównych (ang. *principal component analysis*)

Technika rzutowania zbioru danych lub zmiany układu odniesienia w ten sposób aby:

- wektor zmiennych losowych zastąpić wektorem zmiennych nieskorelowanych
- uporządkować zmienne wynikowe w kolejności istotności
- możliwa była względnie prosta redukcja wymiarowości oddzielając informacje od szumu.

Zastosowania:

- wizualizacja danych wielowymiarowych
- selekcja atrybutów, redukcja wymiarowości,
- kompresja danych
- analiza czynnikowa

Przypomnienie

- Wektory własne odpowiadające różnym wartością własnym są ortogonalne, ponadto można wybrać je tak, aby były dodatkowo unormowane.
- Wartości własne macierzy symetrycznej – a taką jest macierz kowariancji – są liczbami rzeczywistymi.
- Dla każdej macierzy symetrycznej A istnieje ortogonalna macierz P (zbudowana z wektorów własnych P_i) taka, że $P^T AP = \Lambda$, gdzie Λ jest macierzą diagonalną. Dokładniej jest to macierz zawierająca na głównej przekątnej kolejne wartości własne macierzy A.

Przypomnienie

- Dla macierzy symetrycznej można napisać rozkład spektralny postaci:

$$A = P\Lambda P^T = \lambda_1 P_1 P_1^T + \ldots + \lambda_n P_n P_n^T \tag{1}$$

Zależność (1) w przypadku, gdy występuje tylko kilka dominujących wartości własnych, daje możliwość aproksymacji macierzy A, biorąc pod uwagę jedynie wektory własne odpowiadające dominującym wartością własnym.

Analiza składowych głównych

- zbiór danych: $X = [x_{ij}], i = 1, \ldots, p, j = 1, \ldots, n$
- Szukamy kierunku (wersora $\mathbf{a}$), dla którego rozkład rzutu danych: $(X - \mu)\mathbf{a}$ – ma największą wariancję tzn.:

$$\begin{aligned} \sigma_{\mathbf{a}} &= ((X-\mu)\mathbf{a})^T((X-\mu)\mathbf{a}) \\ &= \mathbf{a}^T(X-\mu)^T(X-\mu)\mathbf{a} = \mathbf{a}^T \operatorname{cov} X\mathbf{a} \rightarrow_{\mathbf{a}} \max. \end{aligned}$$

- $\mathbf{a}$ jest wersorem osi, tzn. $\|\mathbf{a}\| = \sqrt{\mathbf{a}^T\mathbf{a}} = 1$

Analiza składowych głównych

Zatem szukamy takiego $\mathbf{a}$, który maksymalizuje:

$$\mathbf{a}^T \operatorname{cov} X \mathbf{a}$$

przy ograniczeniach:

$$\mathbf{a}^T \mathbf{a} = 1$$

Analiza składowych głównych

Wprowadzając mnożnik Lagrange'a dostajemy problem optymalizacji bez ograniczeń.

$$L(\mathbf{a}, \lambda) = \mathbf{a}^T \operatorname{cov} X \mathbf{a} - \lambda(\mathbf{a}^T \mathbf{a} - 1)$$

Różniczkując po $\mathbf{a}$ dostajemy:

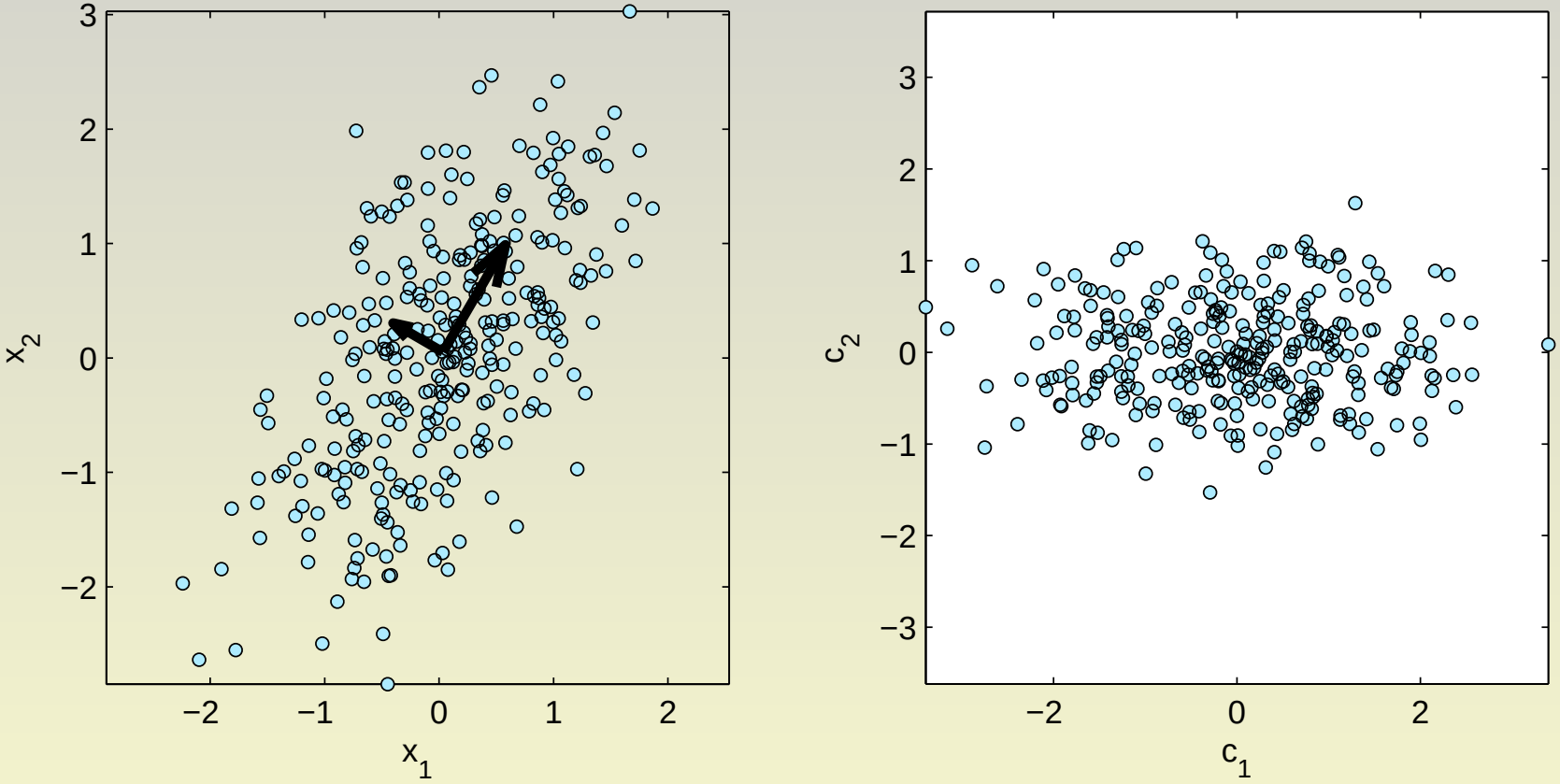
$$\frac{\partial L}{\partial \mathbf{a}} = 2 \operatorname{cov} X \mathbf{a} - 2\lambda \mathbf{a} = 0 \quad (2)$$

$$(\operatorname{cov} X - \lambda \mathbb{I})\mathbf{a} = 0 \quad (3)$$

Zatem musi być: $\det(\operatorname{cov} X - \lambda \mathbb{I}) = 0$

oraz λ jest największą liczbą spełniającą to równanie.

Analiza składowych głównych



Rys. Próba dwuwymiarowa zmienne skorelowane, wektory własne macierzy kowariancji przemnożone przez wartości własne, transformacja do zmiennych niezależnych.

Odległośc Mahalanobis'a

Odległości pomiędzy próbkami:

Manhattan

$$d(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n} |x_i - y_i)|$$

euklidesowa

$$d(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Mahalanobisa[1]

$$d(\mathbf{x},\mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})^T \Sigma^{-1} (\mathbf{x}-\mathbf{y})}$$

[1] porównaj statystykę H^2 Hotelling'a oraz związki z analizą wariancji

Odległośc Hamniga (dla danych nienumerycznych)

Odległość Hamminga

$$d(\mathbf{x},\mathbf{y}) = \begin{cases} 1, & \text{dla } \mathbf{x} = \mathbf{y} \\ 0, & \text{dla } \mathbf{x} \neq \mathbf{y} \end{cases}$$

Braki w danych

Brak wartości wybranych atrybutów dla pewnej liczby obiektów.
Rodzaje braków:

braki losowe (ang. *missing at random*) rozkład braków w danym atrybucie jest niezależny od wartości atrybutu oraz pozostałych atrybutów.

braki nielosowe (ang. *missing not at random, informative missing*) rozkład braków w danym atrybucie jest zależny od wartości atrybutu lub pozostałych atrybutów. (przekroczenie skali pomiarowej przyrządu, prawdopodobieństwo niepodania wielkości zarobków zależy od ich wysokości)

Braki w danych

Sposoby radzenia sobie z brakami

- Niektóre metody (naiwny klasyfikator Bayesa) działają bez większych problemów zarówno z brakami jak i bez braków w danych
- Zawsze można usunąć z tabeli wszystkie wiersze zawierające braki, lub kolumny zawierające braki (po tej operacji możemy dostać zbiór pusty)
- Zastosować metodę uzupełniania braków (np. wstawić typową wartość, algorytm EM)

Selekcja atrybutów

Wybór atrybutów skorelowanych ze zmienną decyzyjna — umożliwiający tak dokładne przewidywanie zmiennej decyzyjnej jak pełen zbiór atrybutów. Usuniecie zmiennych bez znaczenia dla podejmowanej decyzji

Wizualizacja i podsumowanie danych

- Histogramy
- Wizualizacja próbek
- Średnie, kowariancje, macierz korelacji
- Testy statystyczne, statystyki

Szereg rozdzielczy, rozkład częstości

Rozkład częstości

Przedstawia łączny rozkład częstości/liczności jednowymiarowej dyskretnej próby lub pojedynczej zmiennej

Szereg rozdzielczy

W przypadku zmiennej ciągłej równomierna dyskretyzacja zwana jest szeregiem rozdzielczym, lub rozkładem empirycznym pojedynczej zmiennej

Część III

Przykłady

Zbiór danych – `challenger ring`

Temporal order of flight	Number of O-rings at risk on a given flight	Number experiencing thermal distress	Launch temperature (degrees F)	Leak-check pressure (psi)
1	6	0	66	50
2	6	1	70	50
3	6	0	69	50
4	6	0	68	50
5	6	0	67	50
6	6	0	72	50
7	6	0	73	100
8	6	0	70	100
9	6	1	57	200
10	6	1	63	200
11	6	1	70	200
12	6	0	78	200
13	6	0	67	200
14	6	2	53	200
15	6	0	67	200
16	6	0	75	200
17	6	0	70	200
18	6	0	81	200
19	6	0	76	200
20	6	0	79	200
21	6	2	75	200
22	6	0	76	200
23	6	1	58	200

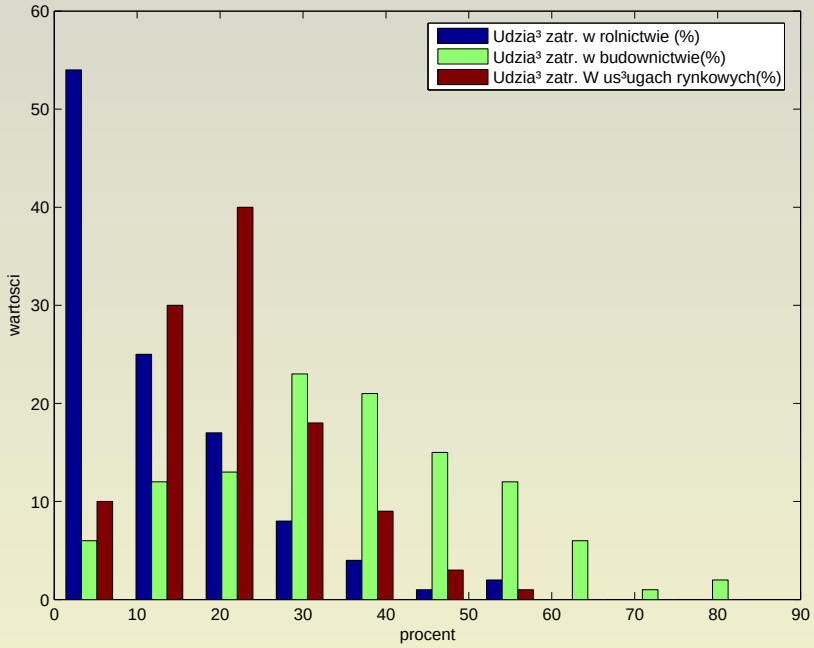
Zbiór danych – autos

fuel-type	body-style	drive-wheels	length	width	curb-weight	engine-type	num-of-cylinders	engine-size	fuel-system	horsepower	city-mpg	highway-mpg	price
gas	hatchback	fwd	177.80	66.50	2385	ohc	four	122	2bbl	84	26	32	8845
gas	sedan	fwd	177.80	66.50	2410	ohc	four	122	2bbl	84	26	32	8495
gas	hatchback	fwd	177.80	66.50	2385	ohc	four	122	2bbl	84	26	32	10595
gas	sedan	fwd	177.80	66.50	2410	ohc	four	122	2bbl	84	26	32	10245
gas	hatchback	fwd	177.80	66.50	2425	ohc	four	122	2bbl	84	26	32	11245
gas	sedan	rwd	175	66.10	2670	ohc	four	140	mpfi	120	19	27	18280
diesel	sedan	rwd	175	66.10	2700	ohc	four	134	idi	72	31	39	18344
diesel	sedan	rwd	190.90	70.30	3515	ohc	five	183	idi	123	22	25	25552
diesel	wagon	rwd	190.90	70.30	3750	ohc	five	183	idi	123	22	25	28248
diesel	hardtop	rwd	187.50	70.30	3495	ohc	five	183	idi	123	22	25	28176
diesel	sedan	rwd	202.60	71.70	3770	ohc	five	183	idi	123	22	25	31600
gas	sedan	rwd	202.60	71.70	3740	ohcv	eight	234	mpfi	155	16	18	34184
gas	convertible	rwd	180.30	70.50	3685	ohcv	eight	234	mpfi	155	16	18	35056
gas	sedan	rwd	208.10	71.70	3900	ohcv	eight	308	mpfi	184	14	16	40960
gas	hardtop	rwd	199.20	72	3715	ohcv	eight	304	mpfi	184	14	16	45400
gas	hatchback	rwd	178.40	68	2910	ohc	four	140	mpfi	175	19	24	16503
gas	hatchback	fwd	157.30	64.40	1918	ohc	four	92	2bbl	68	37	41	5389
gas	hatchback	fwd	157.30	64.40	1944	ohc	four	92	2bbl	68	31	38	6189
gas	hatchback	fwd	157.30	64.40	2004	ohc	four	92	2bbl	68	31	38	6669
gas	hatchback	fwd	157.30	63.80	2145	ohc	four	98	spdi	102	24	30	7689
gas	hatchback	fwd	173	65.40	2370	ohc	four	110	spdi	116	23	30	9959

Preprocesing przykład ZOO

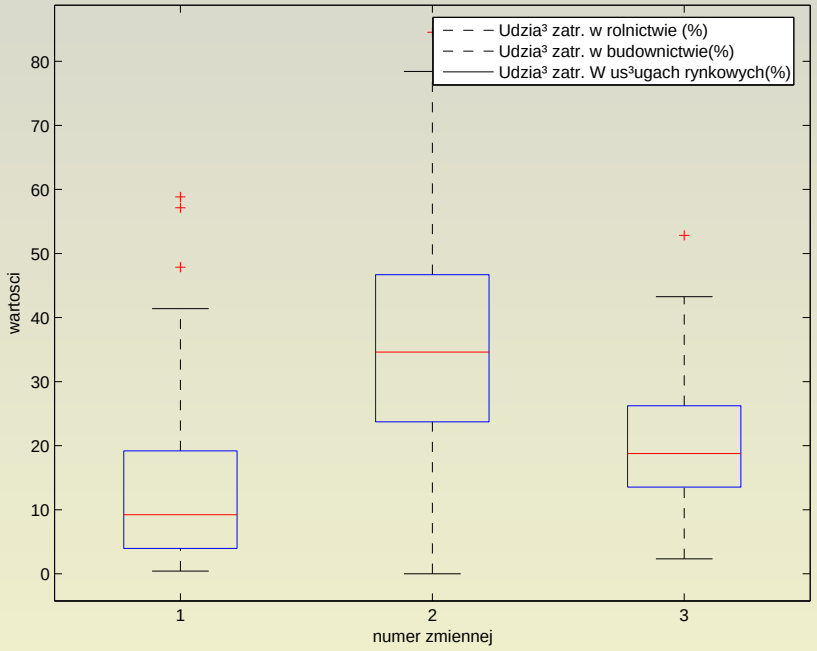
animal	hair	feathers	eggs	milk	airborne	aquatic	predator	toothed	backbone	breathes	venomous	fins	legs	tail	domestic	catsize	type
aardvark	true	false	false	true	false	false	true	true	true	true	false	false	4	false	false	true	mammal
antelope	true	false	false	true	false	false	false	true	true	true	false	false	4	true	false	true	mammal
bass	false	false	true	false	false	true	true	true	true	false	false	true	0	true	false	false	fish
bear	true	false	false	true	false	false	true	true	true	true	false	false	4	false	false	true	mammal
boar	true	false	false	true	false	false	true	true	true	true	false	false	4	true	false	true	mammal
buffalo	true	false	false	true	false	false	false	true	true	true	false	false	4	true	false	true	mammal
calf	true	false	false	true	false	false	false	true	true	true	false	false	4	true	true	true	mammal
carp	false	false	true	false	false	true	false	true	true	false	false	true	0	true	true	false	fish
catfish	false	false	true	false	false	true	true	true	true	false	false	true	0	true	false	false	fish
cavy	true	false	false	true	false	false	false	true	true	true	false	false	4	false	true	false	mammal
cheetah	true	false	false	true	false	false	true	true	true	true	false	false	4	true	false	true	mammal
chicken	false	true	true	false	true	false	false	false	true	true	false	false	2	true	true	false	bird
chub	false	false	true	false	false	true	true	true	true	false	false	true	0	true	false	false	fish
clam	false	false	true	false	false	false	true	false	false	false	false	false	0	false	false	false	invertebrate
crab	false	false	true	false	false	true	true	false	false	false	false	false	4	false	false	false	invertebrate
crayfish	false	false	true	false	false	true	true	false	false	false	false	false	6	false	false	false	invertebrate
crow	false	true	true	false	true	false	true	false	true	true	false	false	2	true	false	false	bird
deer	true	false	false	true	false	false	false	true	true	true	false	false	4	true	false	true	mammal
dogfish	false	false	true	false	false	true	true	true	true	false	false	true	0	true	false	true	fish

Graficzne przedstawienie rozkładu zmiennej



Histogram – rozkład liczności

(zbiór `dochod2000`)



Wykres pudełkowy (ang. *box plot*)

(zbiór `dochod2000`)

Tablice kontygencji (tablice dwudzielcze)

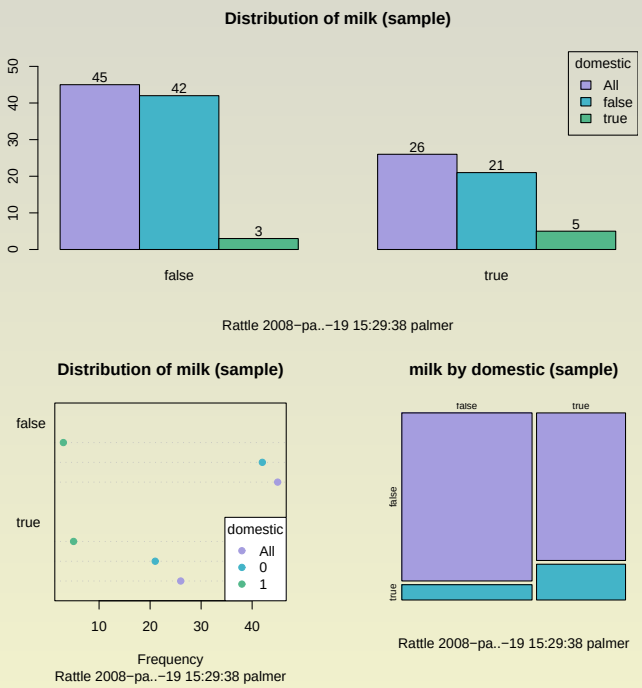
Tablice kontygencji

Przedstawiają łączny rozkład liczności dwuwymiarowej próby losowej.

Tablica: Przykładowa tablica kontygencji (zbiór ZOO slajd 55)

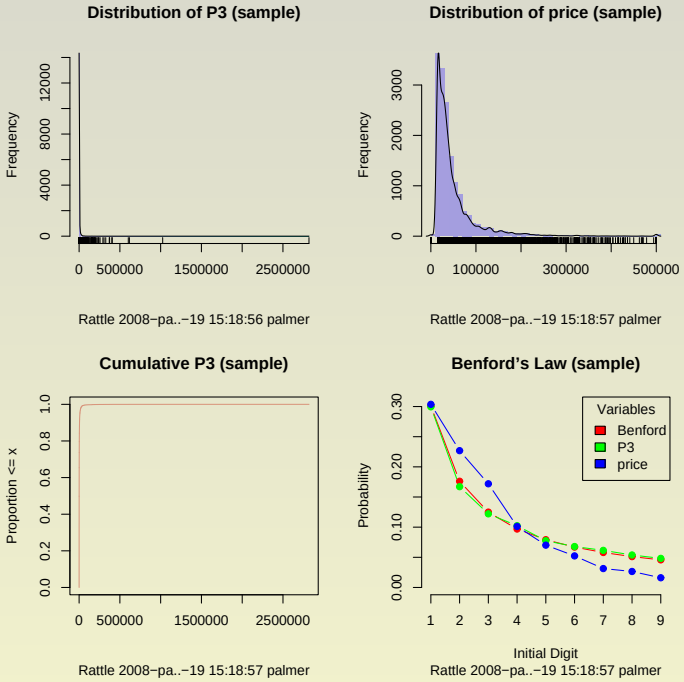
pióra \ domowe	false	true	łącznie:
false	71	10	81
true	17	3	20
łącznie:	88	13	101

Preprocesing przykład



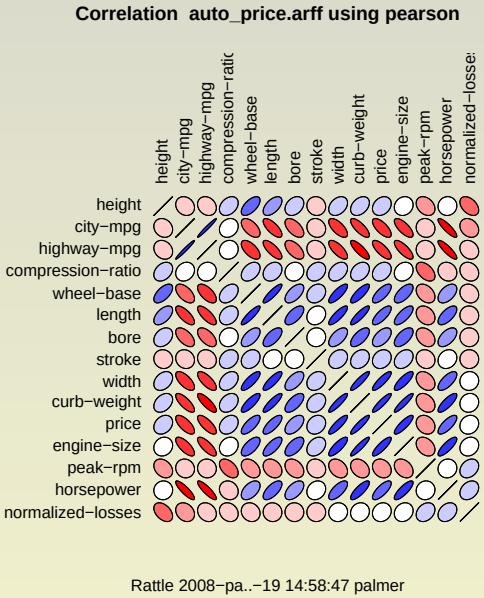
Rys. Histogramy częstości, wizualizacja macierzy kontygencji. (wydruk z **R**)

Preprocesing przykład (zbiór house)



Rys. Atrybuty ciągłe, histogramy, dystrybuanta empiryczna, ocena równomierności rozkładu atrybutów na podstawie wykresu Benford'a. (wydruk z **R**)

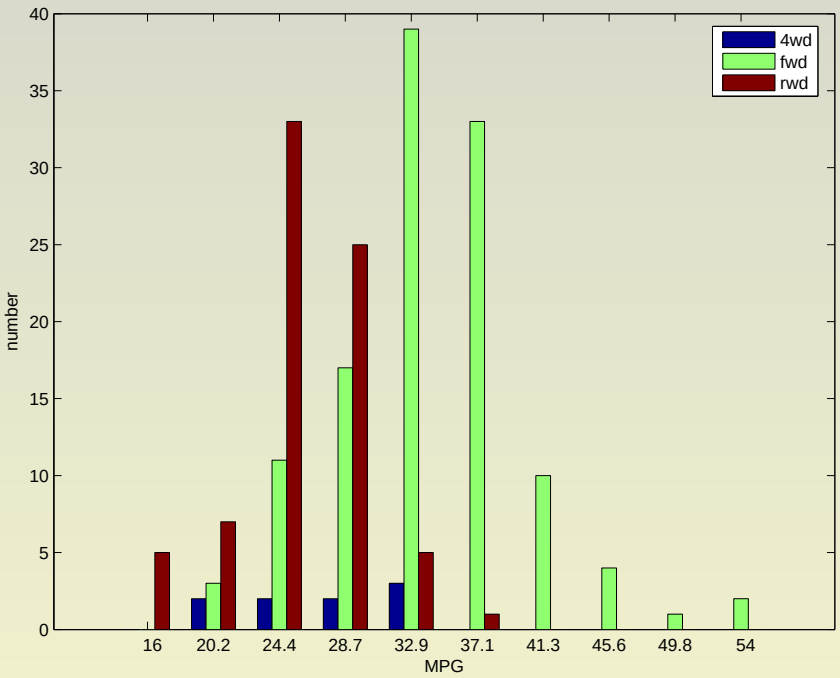
Preprocesing przykład

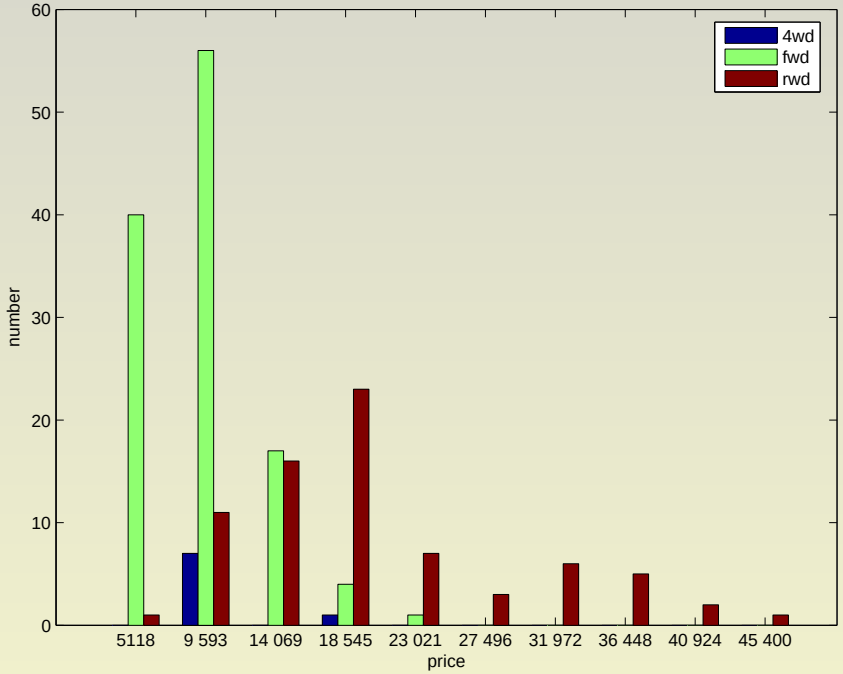


Rys. Wizualizacja macierzy korelacji.

(wydruk z **R**)

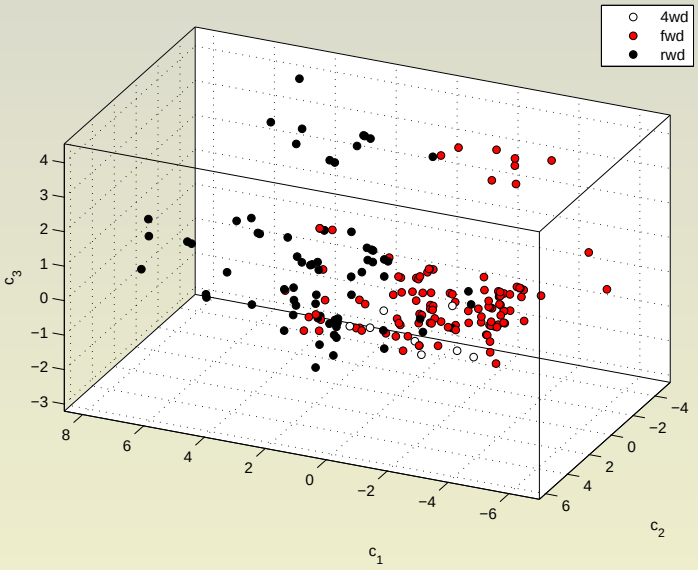
Zbiór autos





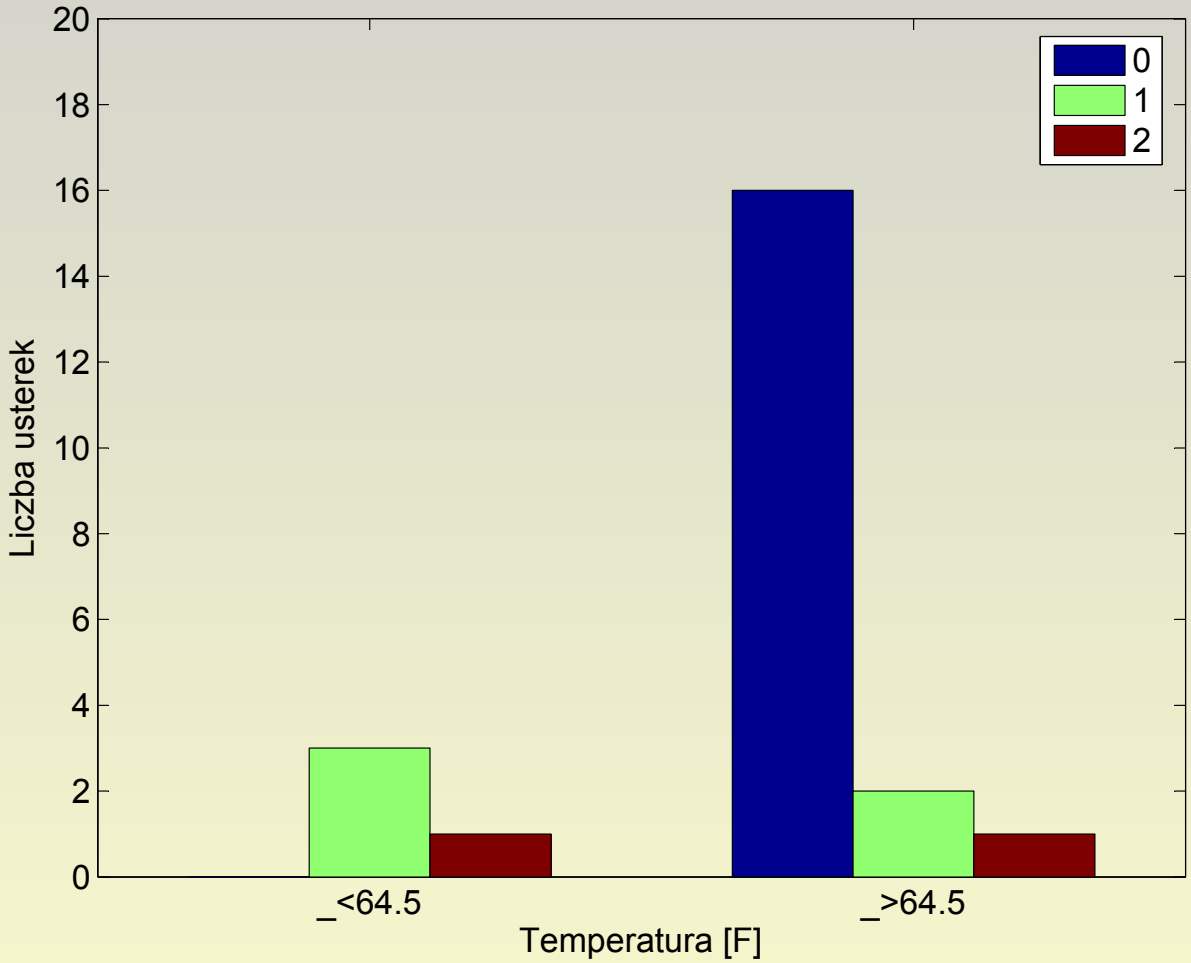
Przykład samochodów o różnym rodzaju napędu w zależności od ceny oraz zużycia paliwa. (zbiór autos).

Zbiór autos



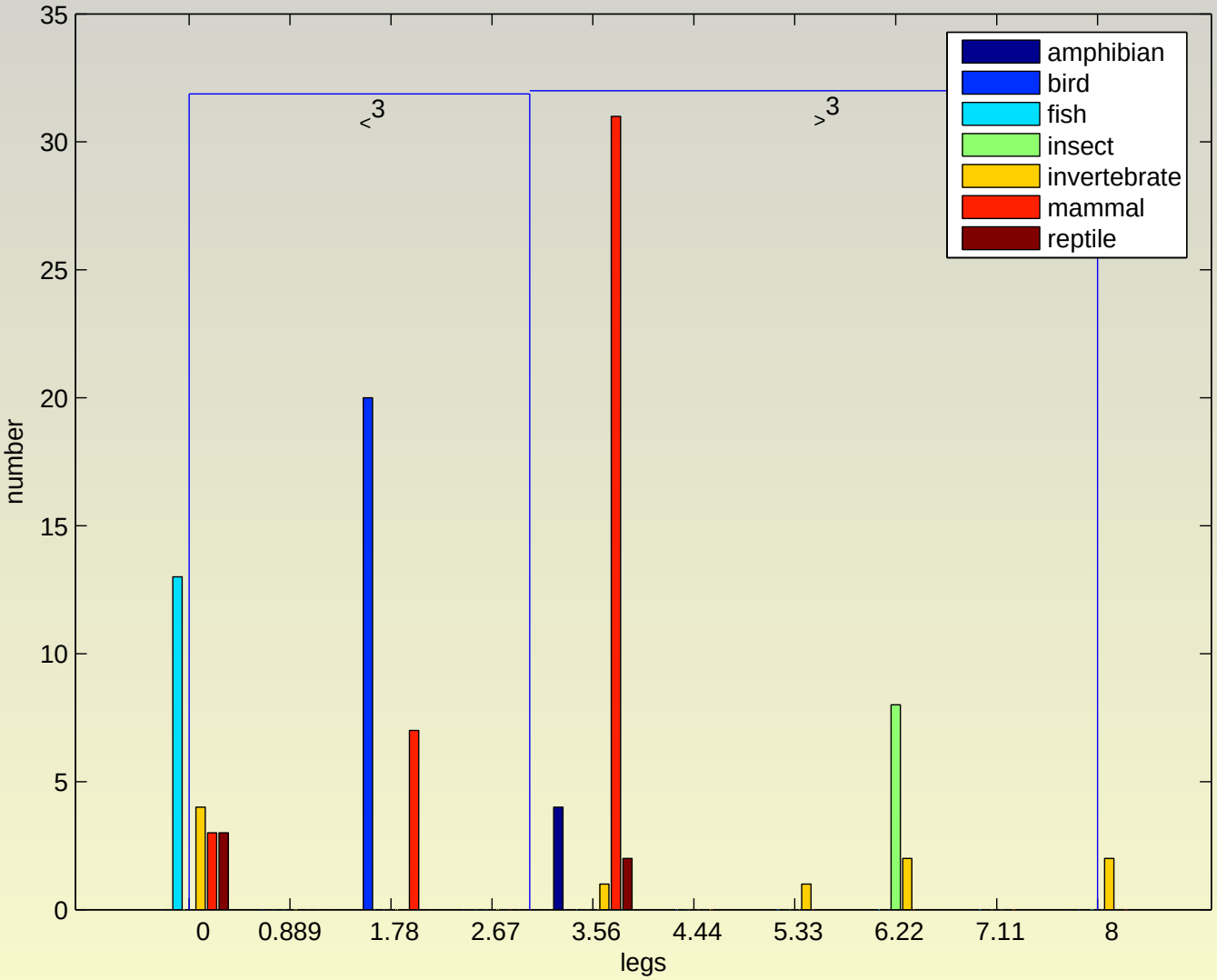
Rys. Wizualizacja danych w rzucie na składowe główne z zaznaczonymi klasami decyzyjnymi (typ napędu) (zbiór autos).

Zbiór `challengerRing`



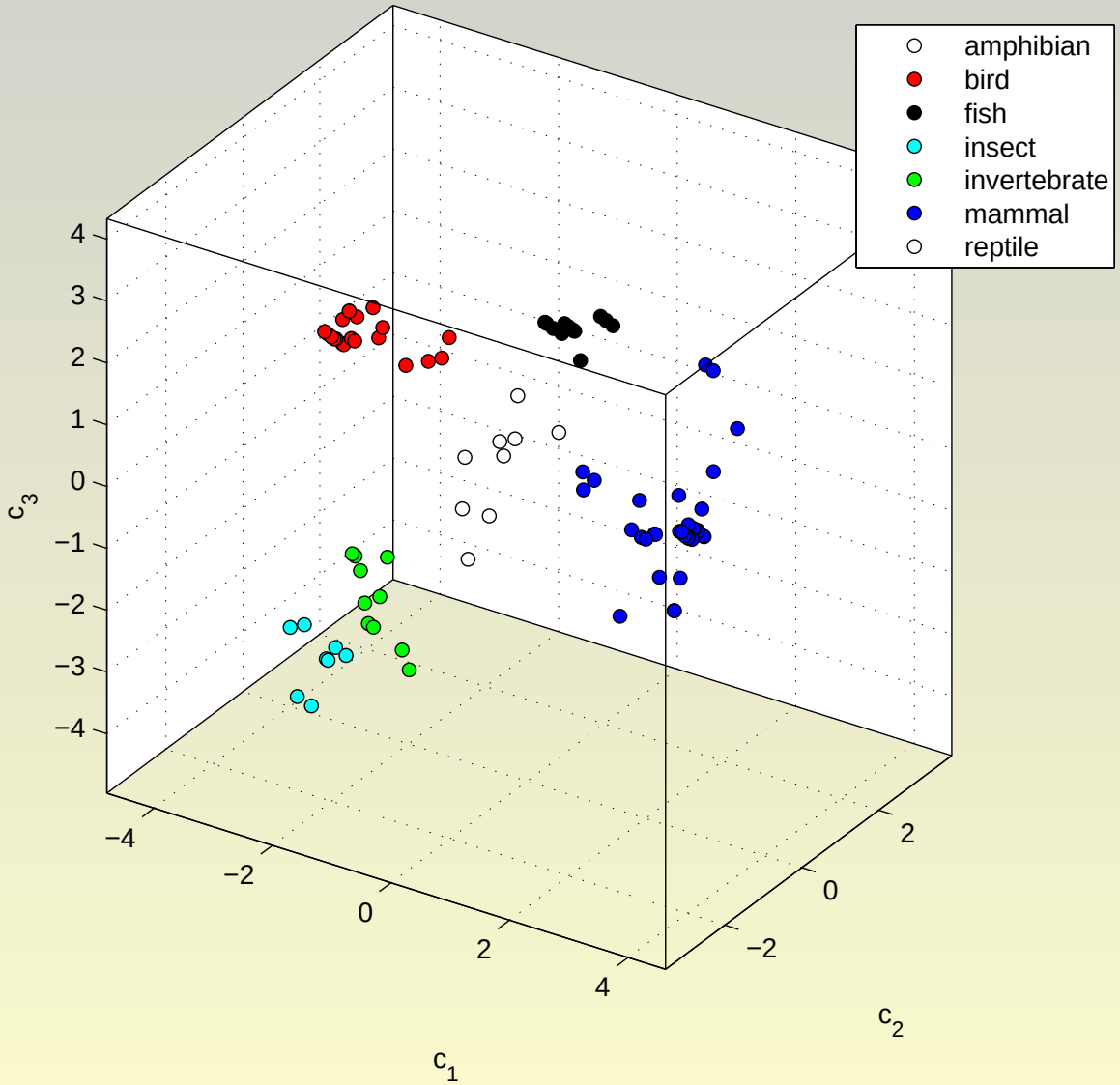
Histogram liczby usterek pod warunkiem temperatury (zbiór `challengerRing`).

Preprocesing przykład



Rys. Dyskretyzacja zmiennej (ktryterium entropia)

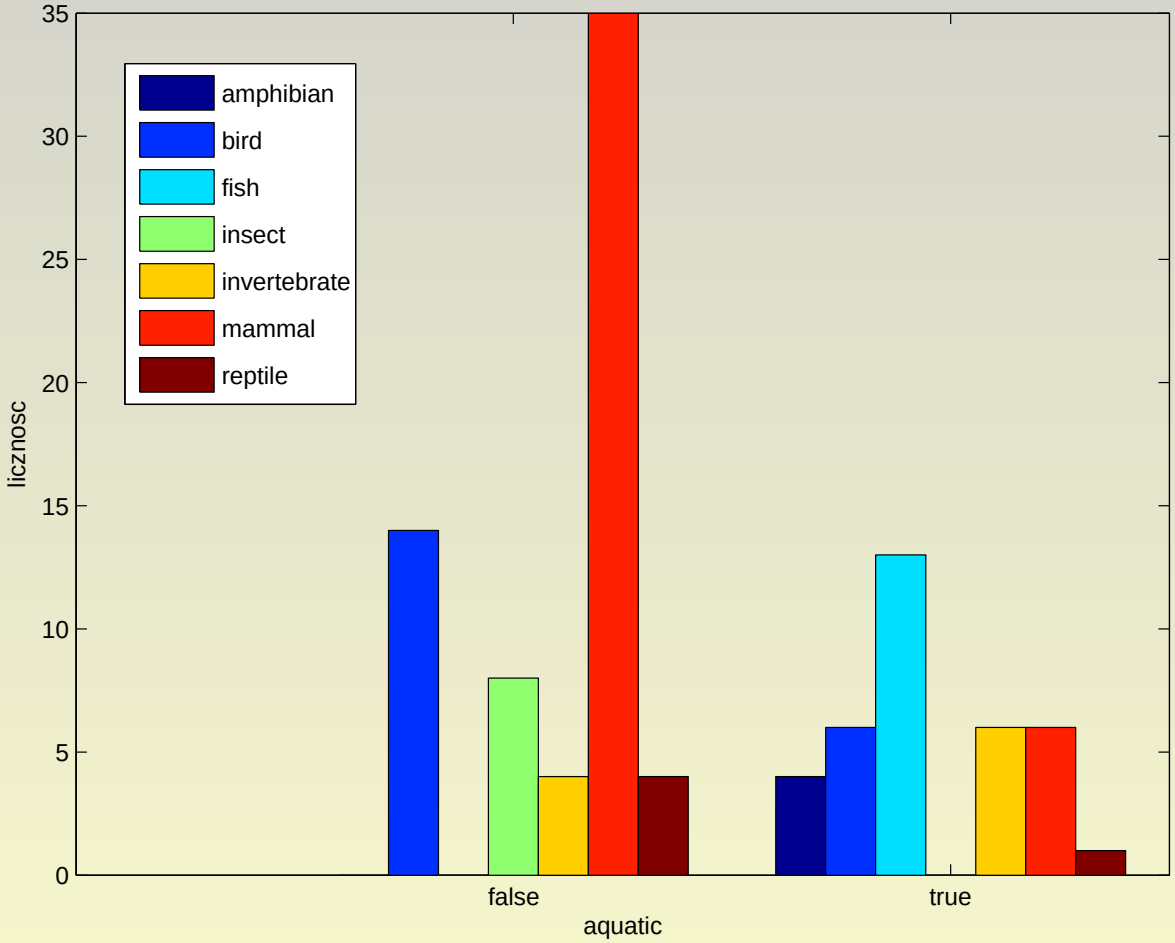
Zbiór ZOO rzut na składowe główne



Klasy kolejno:

1. płazy
2. ptaki
3. ryby
4. owady
5. bezkręgowce
6. ssaki
7. gady

Zbiór ZOO



Rozkład częstość w klasach decyzyjnych dla zmiennej `aquatic`

Część IV

Uczenie maszynowe

Paradygmaty uczenia maszynowego

Uczenie pod nadzorem modele prognozujące: klasyfikacja i regresja

Uczenie bez nadzoru identyfikacja rozkładu, algorytmy samoorganizacji, algorytmy grupowania

Uczenie ze wzmocnieniem uczenie zachowań i strategii, wykorzystywane w algorytmach sztucznej inteligencji

Paradygmaty uczenia maszynowego

Zbiór uczący

- uczenie pod nadzorem

$$P = \{(X^i, d^i)\} i = 1, \ldots, p$$

- uczenie bez nadzoru

$$P = \{X^i\}_{i=1,\ldots,p}$$

- $x_i \in \mathbb{R}^n$, $d_i \in \{-1, 1\}$, lub $d_i \in \mathbb{R}$
- p - liczba próbek uczących

Uczenie maszynowe — informacje ogólne

Zasadniczy podział metod uczenia

Metody wsadowe (off-line), kroki algorytmu są zależne od całego zbioru uczącego, (zwykle w takim trybie wyprowadzane są ścisłe reguły uczące):

- PCA
- algorytm K-środków

Metody bezpośrednie (on-line) kroki zależą od jednej lub kilku próbek uczących (szybkie przybliżenia metod wsadowych):

- uczenie perceptronu
- algorytmy samorganizacji: WTA, WTM

Iteracyjne metody uczenia

Metody bezpośrednie

1. **for** $k = 1$ **to** *liczbaKrokow*
   - $w_{k+1} = w_k + \Delta_k(X^i, d^i)$; (zmiana parametrów zależy od pojedynczej próbki uczącej)

Iteracyjne metody uczenia

Metody bezpośrednie

1. **for** $k = 1$ **to** *liczbaKrokow*
   - $w_{k+1} = w_k + \Delta_k(X^i, d^i)$; (zmiana parametrów zależy od pojedynczej próbki uczącej)

Metody wsadowe

1. **for** $k = 1$ **to** *liczbaKrokow*
   - $w_{k+1} = w_k + \Delta_k(X, d)$; (zmiana parametrów zależy od całego zbioru uczącego)

Część V

Algorytmy Prognozujące - Klasyfikacja

Klasyfikacja

Problem klasyfikacji

Jest to problem decyzyjny, w którym zadaniem jest przyporządkowanie pewnego obiektu, opisanego atrybutami x_j, do jednej z klas decyzyjnych c_i

Problem budowy klasyfikatora

Mając próbę uczącą $\{(X^i, d^i)\}_i = 1, \ldots, p$ pochodzącą z zadanego systemu klasyfikacji chcemy zbudować klasyfikator, który obiekty spoza próby uczącej będzie przyporządkowywał do właściwych klas decyzyjnych

Ocena jakości klasyfikatora

Uwagi

- Nie można ocenić jakości klasyfikatora na próbie uczącej (dlaczego?)
- Jakość klasyfikacji to nie to samo co błąd klasyfikatora

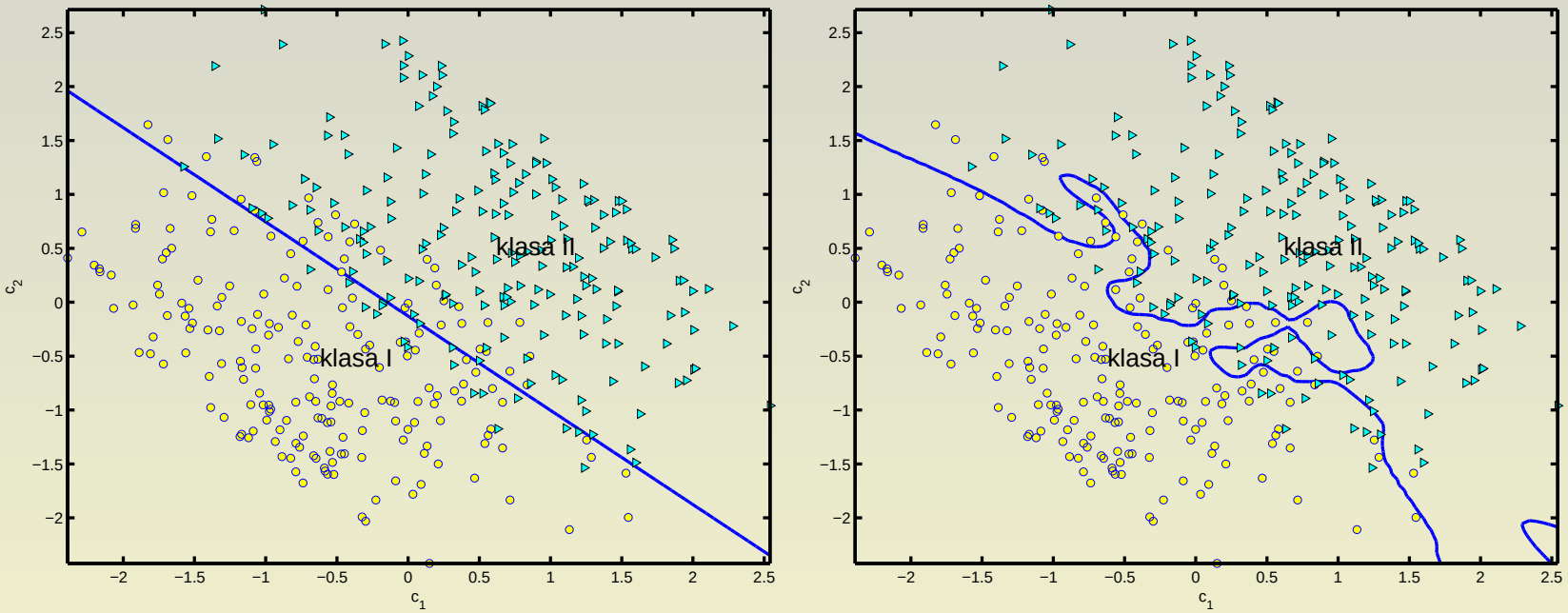
Ocena jakości klasyfikatora

Uwagi

- Nie można ocenić jakości klasyfikatora na próbie uczącej (dlaczego?)
- Jakość klasyfikacji to nie to samo co błąd klasyfikatora

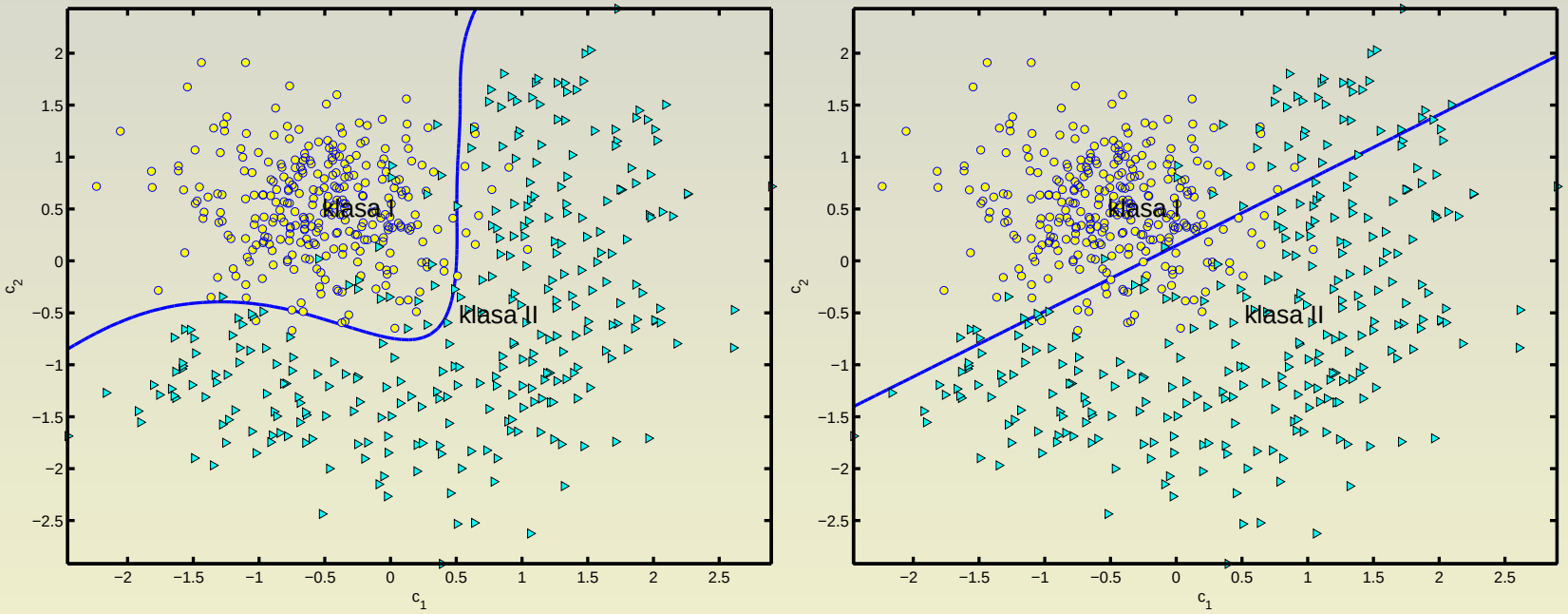
Do oceny klasyfikatora należy wykorzystać dane, które nie brały udziału w procesie uczenia

Zjawisko przeuczenia i niedouczenia



Rys. Model nauczony prawidłowo (a) oraz model przeuczony (b) – błąd obu modeli jest taki sam

Zjawisko przeuczenia i niedouczenia



Rys. Model złożony nauczony prawidłowo (a) oraz model niedouczony (b); dokładność klasyfikacji: (a) 95%, (b) 84%

Ocena jakości klasyfikatora

Podział danych

- Dzielimy zbiór wzorców $P = U \cup V \cup T$
- U - próba ucząca (na tej próbie wykonujemy algorytm uczenia)
- V - próba walidująca (służy do oceny klasyfikatora w trakcie procesu uczenia)
- T - próba testowa (na niej określamy błąd klasyfikatora)

Uwaga

Gdy nie mamy próby testowej lub próba ucząca jest zbyt mała dokładność klasyfikatora możemy wyznaczyć metodą **kroswalidacji**

Ocena jakości klasyfikatora

klasa \ klasyfikowany jako:	negatywny	pozytywny
negatywny	TN	FP
pozytywny	FN	TP

- TP, TF - prawdziwe pozytywy, prawdziwe negatywy
- FP - fałszywie pozytywy - negatywny klasyfikowany jako pozytywny
- FN - fałszywie negatywny - pozytywny klasyfikowany jako negatywny

T- true, F- false

Ocena jakości klasyfikatora

Miary jakości klasyfikatora

- dokładność klasyfikacji

$$\frac{TP+TN}{TP+TN+FP+FN}$$

- błąd klasyfikacji

$$\frac{FP+FN}{TP+TN+FP+FN}$$

- czułość

$$\frac{TP}{TP+FN}$$

- specyficzność

$$\frac{TN}{TN+FP}$$

Ocena jakości klasyfikatora

Miary jakości klasyfikatora

- F–miara

$$\frac{2TP}{2TP+FP+FN}$$

- zbalansowana dokładność

$$\frac{1}{2}\frac{TP}{TP+FN}+\frac{1}{2}\frac{TN}{TN+FP}$$

Uwaga $TP+FN$ jest to prawdopodobieństwo wystąpienia przypadku pozytywnego, natomiast $TN+FP$ - negatywnego.

(Jak wypada klasyfikator bezregułowy w świetle tej miary ?)

Macierz kosztów

Domyślne macierze kosztów

klasa \ klasyfikowany jako:	negatywny	pozytywny
negatywny	0	1
pozytywny	1	0

klasa \ klasyfikowany jako:	A	B	C
A	0	1	1
B	1	0	1
C	1	1	0

W praktyce prawie zawsze mamy asymetrię – ponosimy różne koszty błędnych decyzji

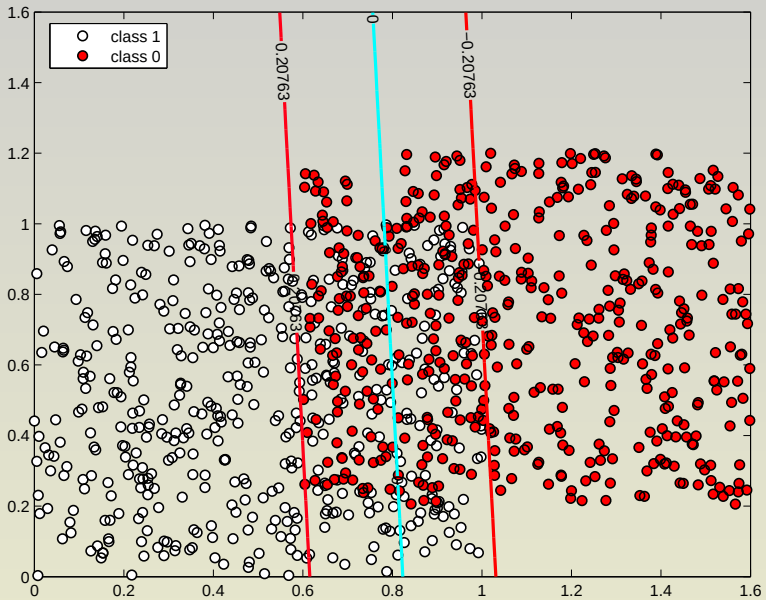
Krzywa ROC

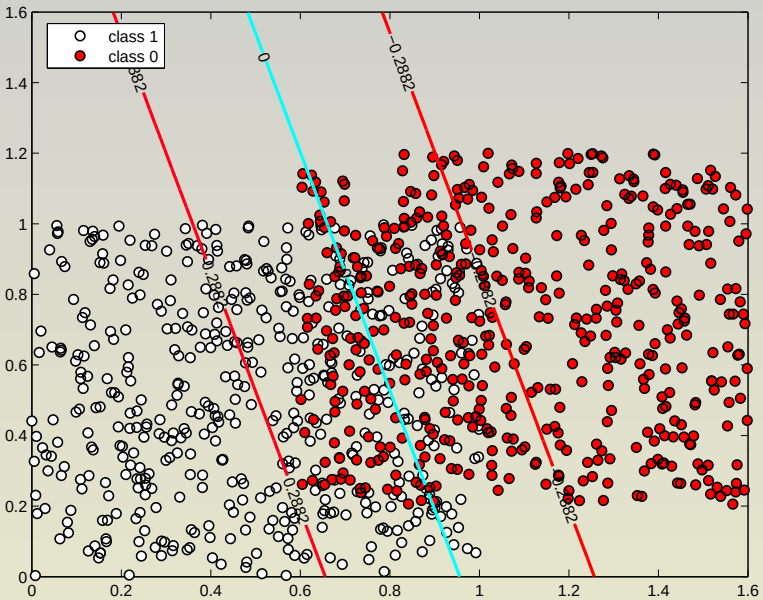
- Stosujemy gdy dysponujemy prawdopodobieństwem decyzji (a nie decyzja typu 0, 1)
- Obieramy próg, przy którym klasę uznajemy za pozytywną lub negatywną
- Przy ustalonym progu liczymy TP, TN, FP, FN
- Zaznaczamy na wykresie czułość i $1-$specyficzność
- Powtarzamy dla różnych wartości progu

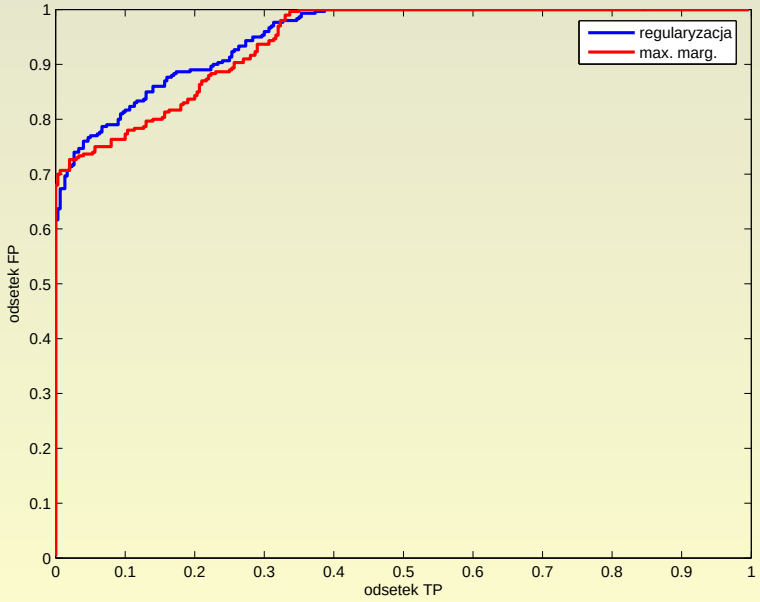
Uwagi

- Odsetek błędnych klasyfikacji (bezpośrednia maksymalizacja NP–trudna nawet gdy szukamy prostej separującej)
- Krzywa ROC, AUC - bezpośrednia minimalizacja powoduje pewne problemy
- Maksimum TP, TN lub coś pośredniego zależnie od asymetrii klasy decyzyjnych
- Maksimum funkcji jakości np.: błędu, lub wiarygodności (nie można zastosować w każdej sytuacji)

Krzywa ROC przykład







	regularyzacja	max. margin
dokładność	$\frac{515}{600} = 0.86$	$\frac{512}{600} = 0.853$
AUC	0.947	0.938
margines	-0.28	-0.20

Przegląd metod klasyfikacji

- Funkcyjne (MLP, SVM, regresja logistyczna)
  - Klasyfikatory liniowe
- Bazujące na próbkach (ang. *instance based classifier*) (k-NN)
- Naiwny klasyfikator bayesa
- Klasyfikatory regułowe (drzewa decyzyjne, zbiory reguł)
- Meta–klasyfikatory

Ograniczenia uczenia maszynowego

Nie ma nic za darmo (ang. *no-free-lunch theorem*)

Wszystkie klasyfikatory są tak samo dobre (lub złe) jeżeli wziąć średnią po wszystkich możliwych zbiorach danych

Wnioski

1. Raczej nie należy poszukiwać najlepszego klasyfikatora w ogólności (w szczególności błędem jest wzięcie kilku zbiorów danych i wybranie na ich postawie najlepszego klasyfikatora)
2. Jeżeli chcemy powiedzieć że klasyfikator jest dobry musimy powiedzieć w jakiej sytuacji (zwykle zakładając co nieco na temat danych)
3. Zależnie od danych (patrząc na rozkład jak i na ilość), które posiadamy o pewnych klasyfikatorach można powiedzieć, że są lepsze lub gorsze (co nie wynika z powyższego twierdzenia)

Część VI

Naiwny klasyfikator Bayes'a

Naiwny klasyfikator Bayes'a (ang. *Naïve Bayes*)

Optymalny klasyfikator Bayesa:

$$Pr(c_i|\mathbf{x}) = \frac{Pr(c_i)\,p(\mathbf{x}|c_i)}{p(\mathbf{x})}$$

gdzie:

c_i	oznacza i–tą klasę decyzyjną
$Pr(c_i)$	oznacza prawdopodobieństwo wyboru decyzji c_i niezależnie od danych
$p(x), p(x\|c_i)$	oznaczają funkcje gęstości rozkładu próbek pomiarowych odpowiednio: łączną, oraz warunkową w klasie c_i.

Naiwny klasyfikator Bayes'a (ang. *Naïve Bayes*)

Naiwne założenie

Atrybuty $x_j, j = 1, \ldots, n$ są warunkowo niezależne w klasach decyzyjnych (pod warunkiem zmiennej decyzyjnej), to znaczy

$$p(\mathbf{x}|c_j) = \prod_{a=1}^{n} p(x_j|c_j)$$

Zatem

$$Pr(c_i|\mathbf{x}) = \frac{Pr(c_i) \prod_{j=1}^{n} p(x_j|c_i)}{p(\mathbf{x})}$$

Naiwny klasyfikator Bayes'a (uwagi)

Uwaga 1. Mianownik jest zawsze stały (nie zależy od klasy decyzyjnej c_i) zatem nie można go łatwo dobrać tak, aby $Pr(c_i|x)$ sumowały się, ze względu na i do jedności. (co zapisujemy: $p(c_i|\mathbf{x}) \propto Pr(c_i) \prod_{j=1}^{n} p(x_j|c_j)$)

Uwaga 2. Klasę decyzyjna dla danej obserwacji wybieramy tę, która maksymalizuje prawdopodobieństwo klasyfikacji to znaczy:

$$c_* = \text{argmax}_{c_i} Pr(c_i) \prod_{j=1}^{n} p(x_j|c_i)$$

Uwaga 3. W przypadku atrybutów ciągłych posługujemy się jednowymiarowymi rozkładami warunkowymi w klasach decyzyjnych (najczęściej zakładamy, że są normalne lub stosujemy algorytm EM).

Naiwny klasyfikator Bayes'a (zastosowania)

Filtr spamu atrybuty binarne są to słowa występujące w mailach (1 wystąpiło, 0 nie wystąpiło)

$$P(spam|wyrazy) \propto Pr(spam) \prod_{j=1}^{n} Pr(wyraz|spam)$$

$$P(\neg spam|wyrazy) \propto Pr(\neg spam) \prod_{j=1}^{n} Pr(wyraz|\neg spam)$$

Problem: gdy pewne słowa się nie pojawią w wiadomościach — estymowane prawdopodobieństwo wyniesie zero.

Część VII

Drzewa klasyfikacyjne

Drzewa klasyfikacyjne

- Drzewa binarne
- Działają na danych symbolicznych jak i numerycznych
- Dobrze radzą sobie z brakami
- Stosunkowo szybkie jeżeli wykonują się w pamięci
- Zachłanny sposób wyboru i podziału kolejnych atrybutów
- Wybór atrybutu zależny od przyrostu informacji / miary zanieczyszczenia

Drzewa klasyfikacyjne

Miary zanieczyszczenia (ang. *impurity*)

entropia miara rozproszenia atrybutu

$$H(a) = -\sum_{p=1}^{k} Pr\{a = p\} \log(Pr\{a = p\})$$

index Giniego

$$G_{CART}(a) = 1 - \sum_{p=1}^{k} Pr\{a = p\}^2$$

błąd klasyfikacji

$$Q(a) = 1 - \max_{p=1,\ldots,k} Pr\{a = p\}^2$$

Drzewa klasyfikacyjne

Miary warunkowe

entropia

$$H(d|a=a_i) = -\sum_{p=1}^{k} Pr\{d=p|a=a_i\}\log(Pr\{d=p|a=a_i\})$$

index Giniego

$$G_{CART}(d|a=a_i) = 1 - \sum_{p=1}^{k} Pr\{d=p|a=a_i\}^2$$

błąd klasyfikacji

$$Q(d|a=a_i) = 1 - \max_{p=1,\ldots,k} Pr\{d=p|a=a_i\}^2$$

Drzewa klasyfikacyjne

Miary warunkowe

entropia

$$H(d|a) = \sum_i Pr(a = a_i) H(d|a = a_i)$$

index Giniego

$$G_{CART}(d|a) = \sum_i Pr(a = a_i) G_{CART}(d|a = a_i)$$

błąd klasyfikacji

$$Q(d|a) = \sum_i Pr(a = a_i) Q(d|a = a_i)$$

Informacja wzajemna (przyrost informacji)

Miary warunkowe

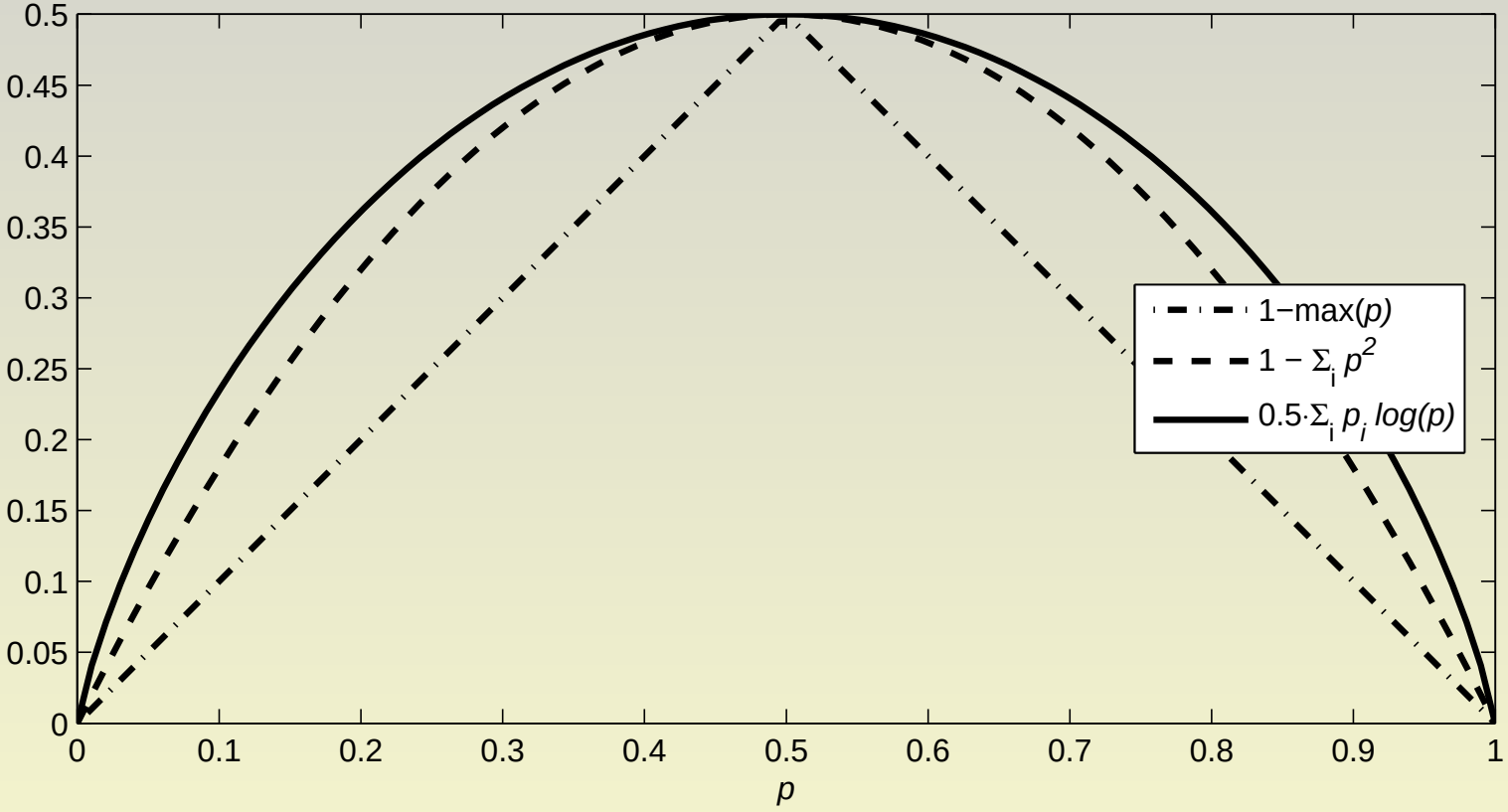
entropia

$$
\begin{aligned}
I_H(d,a) &= H(a) + H(d) - H(ad) \\
I_H(d,a) &= H(d) - H(d|a)
\end{aligned}
$$

index Giniego

$$
\begin{aligned}
I_{G_{CART}}(d,a) &= \cdots ? \cdots \\
I_{G_{CART}}(d,a) &= G_{CART}(d) - G_{CART}(d|a)
\end{aligned}
$$

Miary rozproszenia



Porównanie miar H, G, Q.

Algorytm budowy drzewa (do końca)

```
function T=TREE({A, d}, @Q, T, lastNode)
    for all a ∈ A do
        I_a := I_Q(d, a) = Q(d) − Q(d|a)
    end for
    a* := argmax_{a∈A}(i_a)
    if I_{a*} = 0 then
        return T
    end if
    [{A, d}', {A, d}''] = split({A, d}, a*)
    T.addEdge(lastNode, node(a*, v_1))
    T.addEdge(lastNode, node(a*, v_2))
    T=tree({A, d}', @Q, T, node(a*, v_1))
    T=tree({A, d}'', @Q, T, node(a*, v_2))
    return T
end function
```

Drzewa decyzyjne (uwagi)

Uwaga 1. W przypadku atrybutów ciągłych dokonujemy tzw. podziałów monotonicznych (sortujemy rekordy względem wybranego atrybutu oraz dla każdego rekordu sprawdzamy przyrost $I(d, a)$)

Uwaga 2. Przycinanie drzewa wybranie kompromisu pomiędzy koszem (jakością klasyfikacji), a złożonością drzewa (liczbą krawędzi)

Uwaga 3. Do oceny kosztu używamy próby walidującej lub kroswalidacji

Część VIII

Regresja logistyczna

Troche historii

1838 Pierre François Verhulst (1804-1849), funkcja logistyczna jako rozwiązanie równania: $\frac{dn}{dt} = rn\left(1 - \frac{n}{k}\right)$, N - liczba osobników, k - pojemność środowiska. rozwiązanie:

$$n(t) = \frac{k}{1 + Cke^{-rt}}, \quad C = 1/N(0) - 1/K$$

1920 Reed & Pearl ponownie wzrost populacji

1930 Bliss i inni (w tym R. A. Fisher, probit, estymacja parametrów metodą ML)

1944 J. Berkson (logit, minimum χ^2)

1957 (tu mamy perceptron)

1966 Cox (dalsze uogólnienia) - modele Cox'a

1977 Pakiet BDMP (pierwsze powszechne użycie w pakiecie statystycznym)

Funkcja logistyczna

funkcja logistyczna

$$y(x; w, b) = \frac{1}{1 + e^{-(x \cdot w + b)}}$$

model do przewidywania prawdopodobieństwa klasyfikacji

$$p(1|x) = y(x)$$

$$p(0|x) = 1 - y(x)$$

Logit

szansa (ang. *odds*):

$$o = \frac{p}{1-p}$$

$$\text{logit}(p) = \log o = \log\left(\frac{p}{1-p}\right) = \log(p) - \log(1-p)$$

$$p = \frac{1}{1+e^{-u}}, \qquad u = log\left(\frac{1-p}{p}\right)$$

$$\log\left(\frac{1-p}{p}\right) = b + x_1 w_1 + \ldots + x_n w_n$$

Probit

$$\text{probit}(p) = \Phi^{-1}(p)$$

Estymacja parametrów

$$
\begin{aligned}
L(w,b) &= \prod_{i=1}^{p} y(x_i;w,b)^{d_i}(1-y(x;w,b))^{1-d_i} \\
l(w,b) &= -\sum_{i=1}^{p} d_i \log y(x_i;w,b) + (1-d_i)\log(1-y(x_i;w,b))
\end{aligned}
$$

$$
\frac{\partial l}{\partial w_j} = -\sum_{i=1}^{p} d_i \frac{y_i(1-y_i)x_j}{y_i} + (1-d_i)\frac{1}{1-y_i} = -\sum_{i=1}^{p}(d_i-y_i)x_j
$$

$$
\frac{\partial l}{\partial b} = -\sum_{i=1}^{p}(d_i-y_i)
$$

Kilka uwag

- Perceptron ma sporo wspólnego z regresją logistyczną
- SVM również jeżeli porówna się funkcje straty: $Q(z) = (1 - z)_+$ oraz $Q(z) = \log(1 + e^{-z})$.
- regresja logistyczna sama w sobie nie jest odporna na atrybuty zupełnie losowe (są one w stanie zaburzać wyniki)
- W przypadku, gdy klasy mają rozkład normalny to MLE i SSE dają jednakowe wyniki, w przeciwnym wypadku tak być nie musi.
- W przypadku zwykłej regresji liniowej oraz normalnej postaci szumów dostajemy SSE stosując MLE

Algorytm uczenia

- IRLS - w kolejnych krokach rozwiązujemy równanie $\frac{\partial l}{\partial w_j} = 0$
  (złożoność o($\mathbf{n^3} + n * p$), odwracanie macierzy)
- WLS + liniowy CG
- CG
- inne

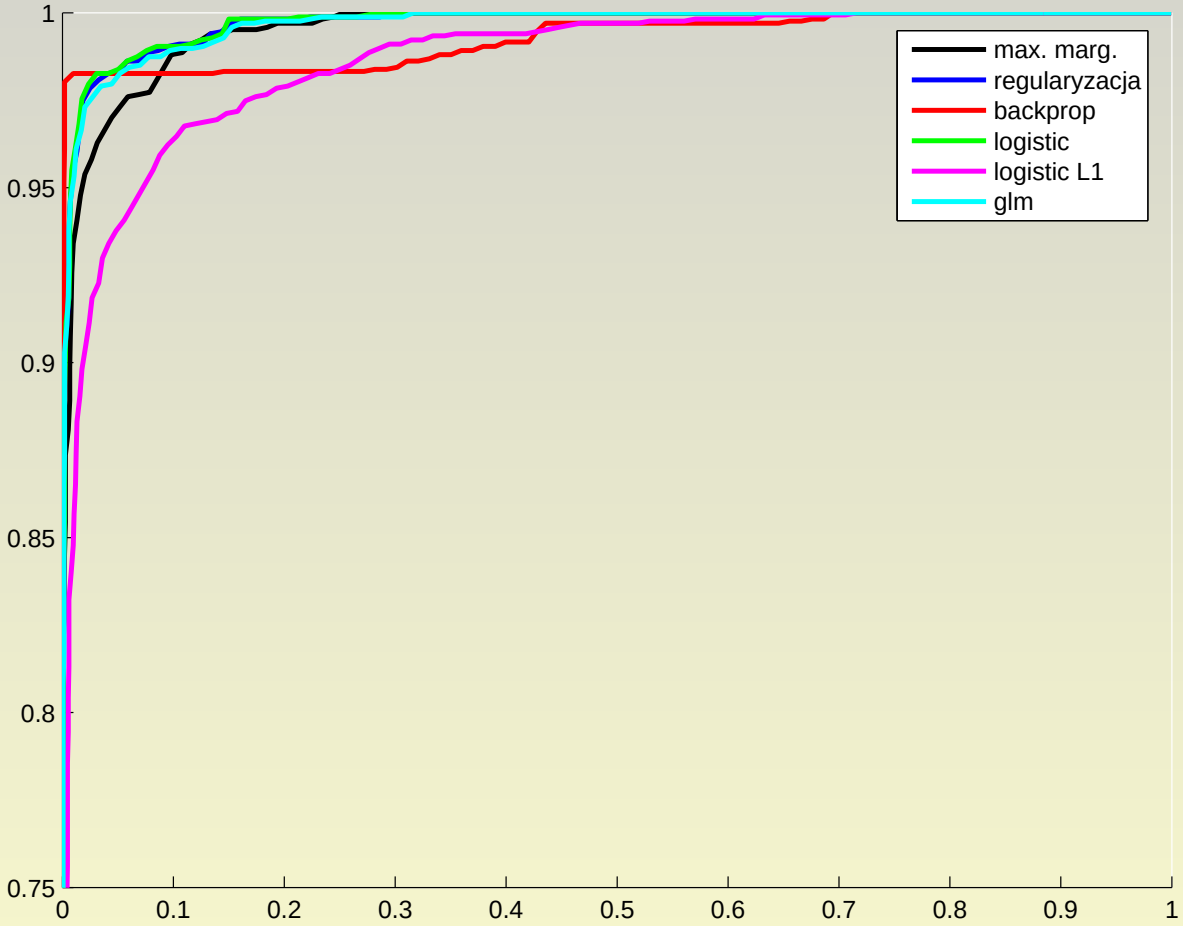
Z test, Wald'a ,(istotnośc współczynników, R)

```
 Estimate Std.  Error z value Pr(>|z|)
(Intercept) -1.54004 0.26915 -5.722 0.0000000105 ***
oddzial21 0.53683 0.19337 2.776 0.005500 **
lokalizacja11 0.13925 0.19648 0.709 0.478491
lokalizacja21 -0.19685 0.24446 -0.805 0.420681
wiekpon40.51 0.59341 0.15513 3.825 0.000131 ***
plec11 0.19880 0.17836 1.115 0.265035
wykszt11 0.55800 0.17271 3.231 0.001234 **
wykszt121 -0.07831 0.70598 -0.111 0.911677
wykszt31 -0.23314 0.31479 -0.741 0.458928
wykszt41 -0.47791 0.21855 -2.187 0.028764 *
miejsce11 -0.30177 0.25917 -1.164 0.244276
miejsce21 -0.40837 0.23165 -1.763 0.077929 .
miejsce31 -0.01204 0.23865 -0.050 0.959771
miejsce41 NA NA NA NA
rozp21 0.80688 0.15781 5.113 0.0000003173 ***
stom21 -0.35270 0.15166 -2.326 0.020041 *
oper21 0.75903 0.15351 4.945 0.0000007632 ***
transf11 0.49115 0.19632 2.502 0.012356 *
injek21 0.54147 0.24742 2.188 0.028633 *
tatuaz11 0.84862 0.20783 4.083 0.0000444170 ***
```

Test ilorazu wiarygodności

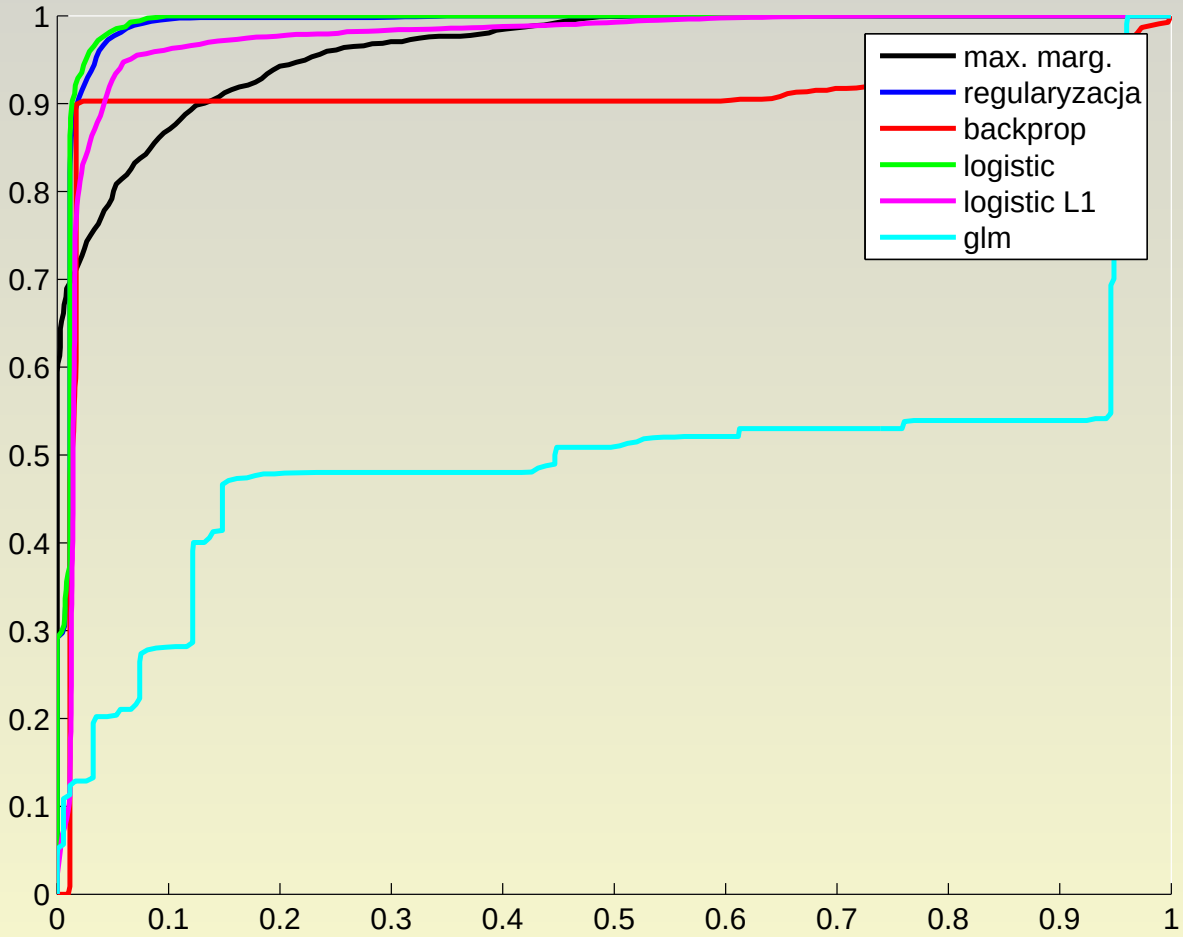
```
 Df Deviance Resid.  Df Resid.  Dev P(>|Chi|)
NULL 1068 1390.15
oddzial2 1 14.14 1067 1376.01 0.0001694481
lokalizacja1 1 0.05 1066 1375.96 0.83
lokalizacja2 1 6.69 1065 1369.27 0.01
wiekpon40.5 1 21.37 1064 1347.90 0.0000037833
plec1 1 5.97 1063 1341.93 0.01
wykszt1 1 18.40 1062 1323.53 0.0000178989
wykszt12 1 0.15 1061 1323.38 0.70
wykszt3 1 0.26 1060 1323.12 0.61
wykszt4 1 4.62 1059 1318.50 0.03
miejsce1 1 0.01 1058 1318.49 0.90
miejsce2 1 5.56 1057 1312.92 0.02
miejsce3 1 0.0017571245 1056 1312.92 0.97
miejsce4 0 0.00 1056 1312.92
rozp2 1 29.37 1055 1283.55 0.0000000598
stom2 1 4.93 1054 1278.62 0.03
oper2 1 26.32 1053 1252.30 0.0000002895
transf1 1 7.73 1052 1244.58 0.01
injek2 1 5.79 1051 1238.79 0.02
tatuaz1 1 16.76 1050 1222.03 0.0000424593
```

eksperymenty



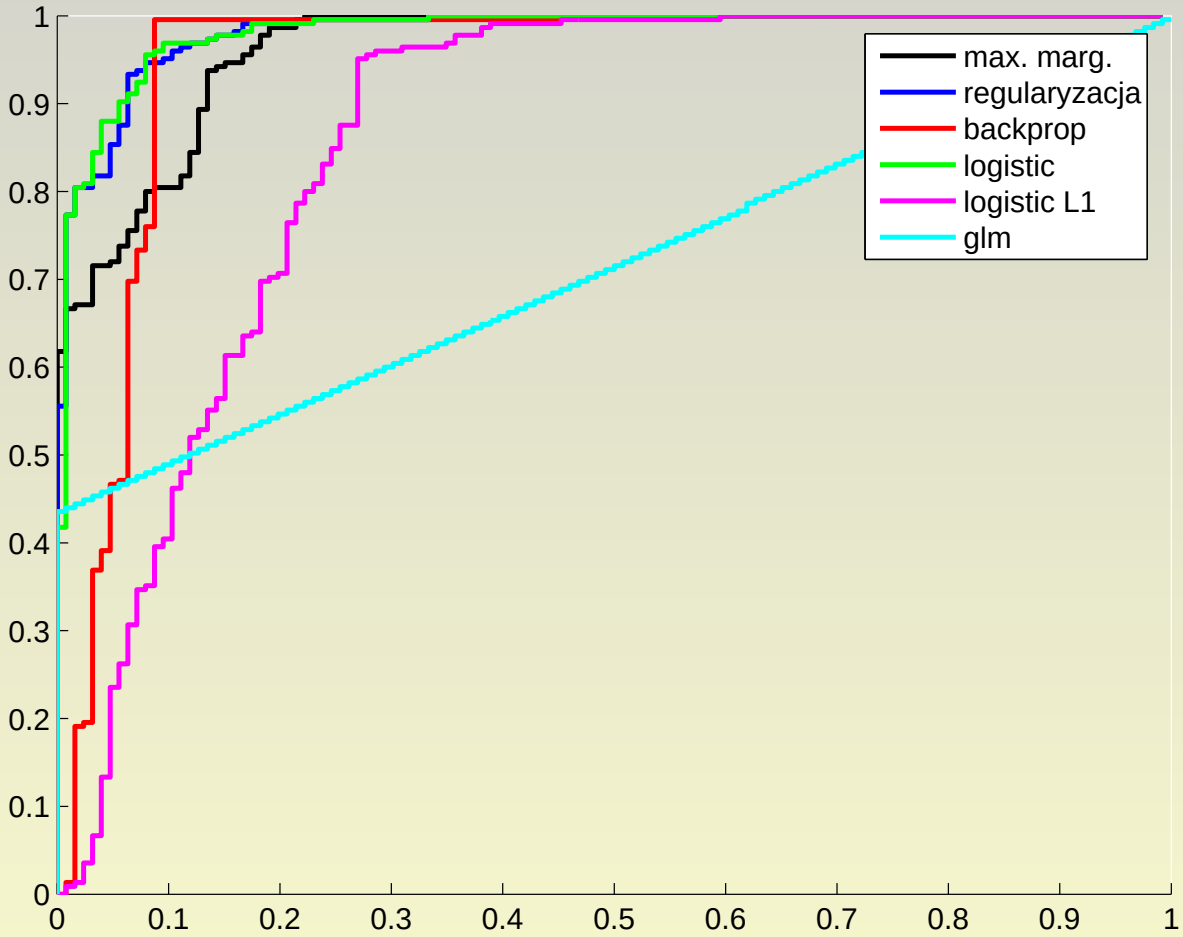
Krzywa ROC dane `krvskp`

eksperymenty



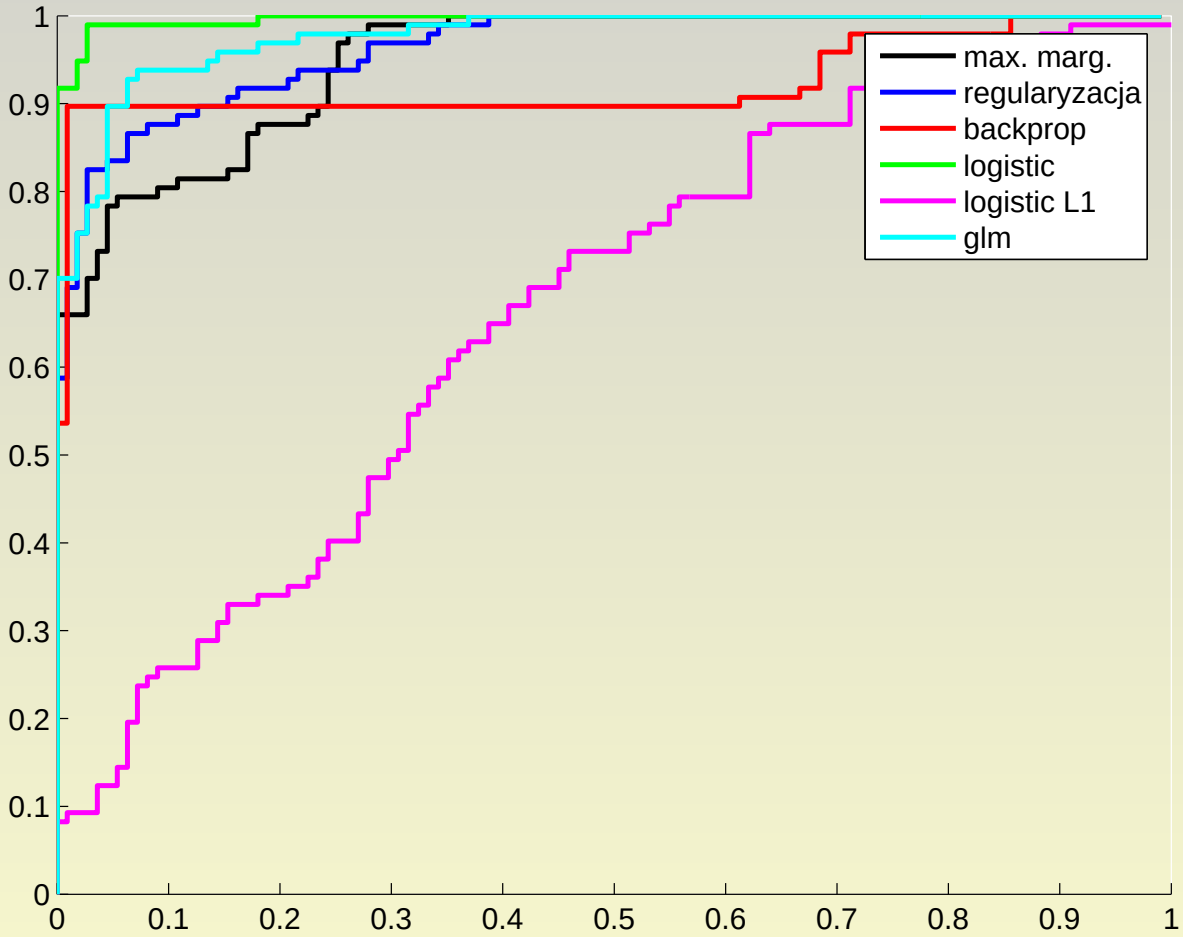
Krzywa ROC dane `mushroom`

eksperymenty



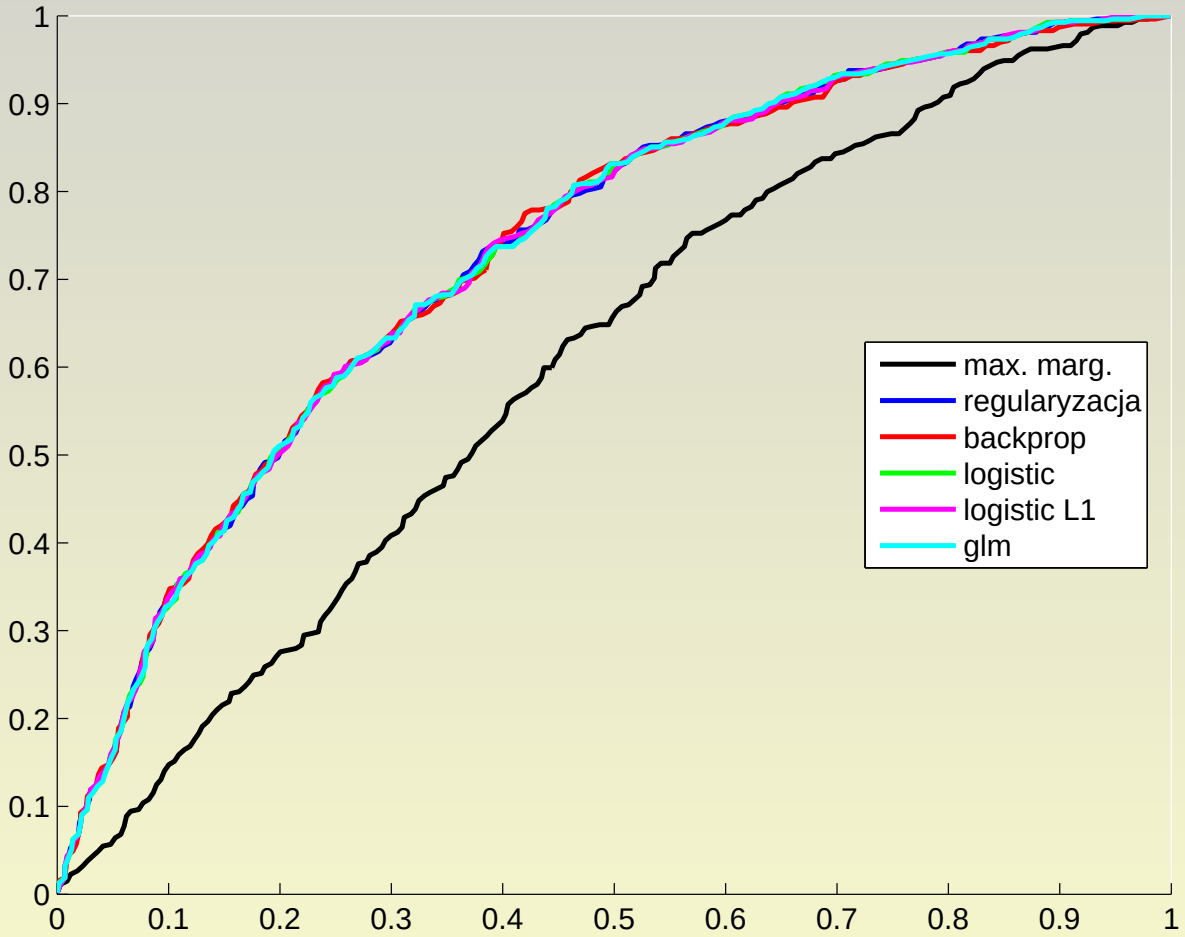
Krzywa ROC dane `ionosphere`

eksperymenty



Krzywa ROC dane `sonar`

eksperymenty



Krzywa ROC dane `PAM szczepienie HBV`

Część IX

Rzadka regresja logistyczna

Algorytm podstawowy wady i zalety

- Niejednoznaczne rozwiązanie w przypadku atrybutów warunkowych $1:1$ zależnych
- Brak wbudowanego mechanizmu selekcji atrybutów
- Nieodporny na dodawanie szumów – atrybutów niezależnych z decyzją
- Nie uwzględnia wiedzy *a priori* na temat rozkładu wag

Kiedy regularyzacja jest konieczna ?

- Numeryczna niestabilność algorytmów
- Niejednoznaczność rozwiązania
- Mamy pewną wiedzę na temat możliwych rozwiązań

Przykład (A. N. Tichonow 1943, 1963, 1977):
Szukamy $\|A\mathbf{x} - \mathbf{b}\|^2 \to \min$, w przyp. gdy A jest osobliwa mamy dużą liczbę rozwiązań. Możemy wybrać jedno dodając term z karą za wielkość rozwiązania:

$$\|A\mathbf{x} - \mathbf{b}\|^2 + \|\Gamma\mathbf{x}\|^2 \to \min$$

Troche historii

A. N. Tichonow 1943, 1963, 1977 regularyzacja zastosowana do problemów numerycznych

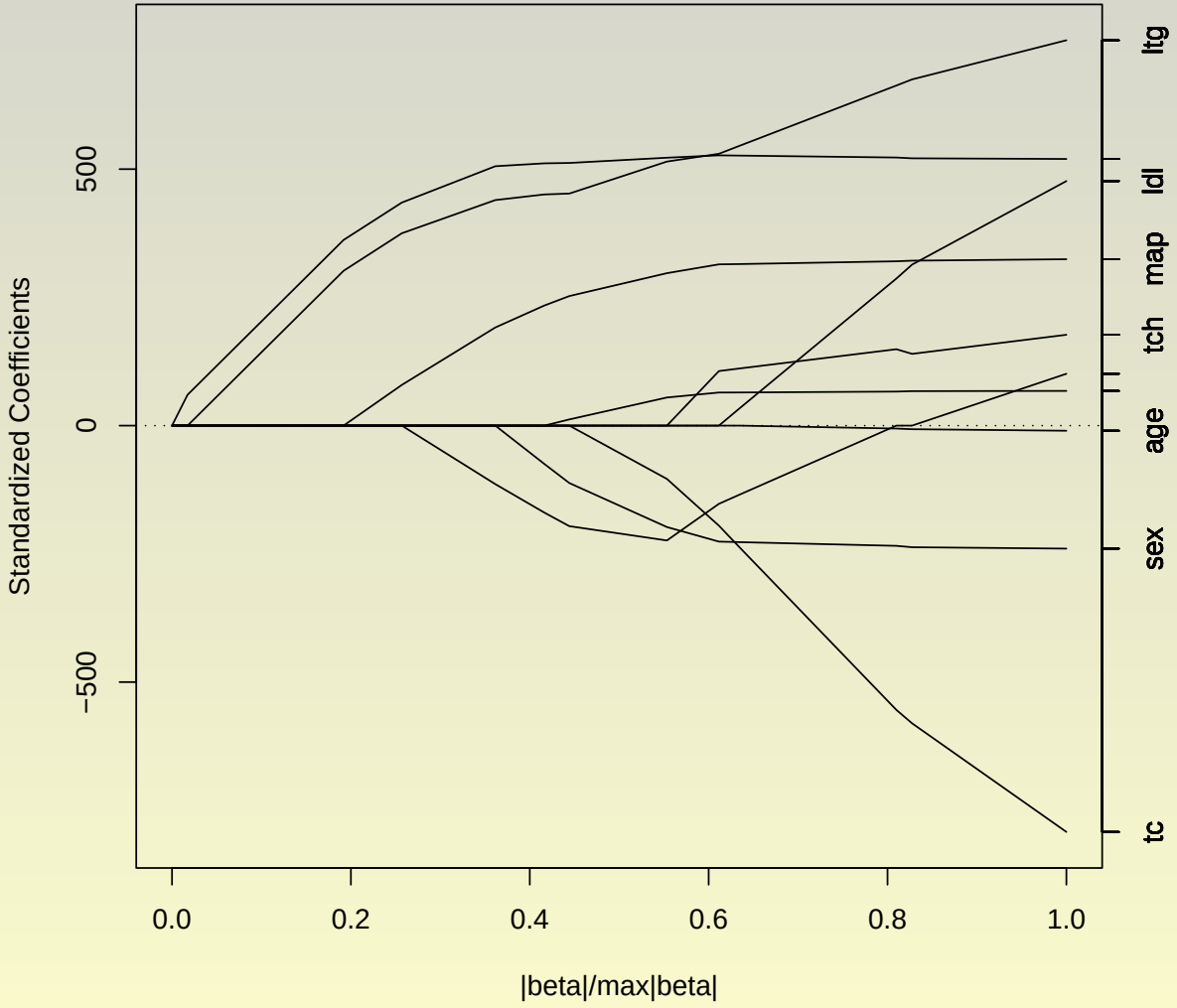
A. E. Hoerl 1962 regresja (liniowa) grzebieniowa (ang. *ridge regression*)
kara: $\gamma \| w \|_2$

R. Tibshirani 1996 lasso $\beta \| w \|_1$

H. Zou, T. Hastie 2003 regresja liniowa *elastic net*
$(1-\beta) \| w \|_2 + \beta \| w \|_1$

B. Efron 2004 *LARS* *(ang. least angle regression)*

elastic net



Działnie *elastic net* dla różnych wartości parametru β, dane `diabetes`

Dlaczego regularyzacja działa ?

Motywacja geometryczna

$$\max_w l(w) = \max_w \sum_{i=1}^{p} \log(p(d|x, w))$$

przy ograniczeniach:

$$\|w\|_q < B$$

Równoważnie używając mnożników Lagrange'a:

$$\max_w \sum_{i=1}^{p} \log(p(d|x, w)) - \gamma \sum_i |w_i|$$

Dlaczego regularyzacja działa ?

Motywacja bayesowska

Interesuje nas rozkład wag pod warunkiem danych
$D = \{X, d\}_{i=1,\ldots,p}, X \in R^n$

$$p(w|D,\gamma) \propto P(D|w,\gamma)\, p(w|\gamma)$$

Założenie *a priori*: wagi są niezależne i maja rozkład Lapace'a:
$\frac{\gamma}{2} e^{-\gamma|w_i|}$ łączny rozkład wag:

$$p(w|\gamma) = \prod_{i=1}^{n} \frac{\gamma}{2} e^{-\gamma|w_i|} = \frac{\gamma}{2} e^{-\gamma \Sigma_i |w_i|}$$

Dlaczego regularyzacja działa ?

Motywacja bayesowska

Maksymalizujemy logarytmiczną wiarygodność modelu:

$$\log(p(w|D,\gamma)) \propto \log(P(D|w,\gamma)\,p(w|\gamma)) =$$

$$l(w) - \gamma \sum_i |w_i| + c$$

Dlaczego regularyzacja działa ?

Algorytmy obortowo–niezmiennicze

Niech M będzie macierzą ortogonalną ($M \cdot M' = M' \cdot M = \mathbb{1}$), wynik działania algorytmu dla danych $\{X, d\}$ jest identyczny jak dla danych $\{X \cdot M, d\}$, m. in.:

- SVM z jądrami liniowym, wielomianowymi, gaussowskimi
- MLP + backprop
- zwykła regresja logistyczna, (również z regularyzacją L^2!)
- reguła perceptronu
- PCA

Dlaczego regularyzacja działa ?

Algorytmy nie obortowo–niezmiennicze

- naiwny klasyfikator bayesowski
- drzewa decyzyjne z rozcięciami wzgledem atrybutów ((bez mutli–rozcięć)
- regresja logistyczna z regularyzacją L^1
- większość algorytmów selekcji cech (np. używając przyrostu informacji czy indeksu Giniego)

Algorytmy obortowo–niezmiennicze

Zasadnicze różnice

- Dla dowolnego algorytmu obrotowo niezmienniczego istnieją problemy liniowo separowalne, które wymagają liczby próbek liniowo zależnej od liczby atrybutów. [A. Y. Ng]
- Regresje z regularyzacją L_1 oraz naiwny klasyfikator Bayes'a można z powodzeniem uczyć, gdy liczba próbek jest w przybliżeniu logarytmem z liczby atrybutów
- Gdy próba jest wystarczająco duża SVM, LR dają rozwiązania dokładne

Część X

Regresja logistyczna z regularyzacją bayesowską

Regularyzacja bayesowską

Maksymalizujemy logarytmiczną wiarygodność modelu:

$$\log(p(w|D,\theta)) \propto \log(P(D|w,\theta)\,p(w|\theta))$$

$p(w|\theta)$ – rozkład apriori

θ – wektor parametrów rozkładu apriori

Co optymalizujemy

$$
\begin{aligned}
M(w,b,\theta) &= \prod_{i=1}^{l} y(x_i;w,b)^{d_i}(1-y(x;w,b))^{1-d_i}p(w|D,\theta) \\
Q(w,b;c;\gamma) &= \log(M(w,b,c|\gamma) \\
&= \underbrace{\sum_{i=1}^{l} d_i \log y(x_i;w,b) + (1-d_i)\log(1-y(x_i;w,b))}_{Q_D} + \\
& \underbrace{\log\big(p(w|\theta)\big)}_{Q_w} = Q_D(x,w) + Q_w(w,\theta)
\end{aligned}
$$

Q_D term ze zwykłej regresji logistycznej wspólny dla wszystkich metod zależny od danych

Q_w term zależny od rozkładu a-priori wag, (θ – wektor parametrów tego rozkładu)

Założenia a-priori

$\emptyset$ brak założeń:

$$p(w) \; =? \; 1$$

(przed uczeniem każda możliwa wartość wag jest dla nas jednakowo prawdopodobna/możliwa?)

G Gauss:

$$p_G(w|\sigma) = \frac{1}{2\pi\sqrt{\sigma}} \exp\left(-\frac{w^2}{2\sigma^2}\right)$$

L Laplace:

$$p_L(w|\lambda) = \frac{1}{\lambda} \exp\left(-\frac{|w|}{\lambda}\right)$$

pochodne loglikelihood'a względem w

$$\frac{\partial Q}{\partial w_j} = \sum_{i=1}^{l}(d_i - y_i)x_{ij} + \frac{\partial}{\partial w_j}\log\big(p(w|c,\gamma)\big) \tag{4}$$

$$\frac{\partial Q}{\partial b} = -\sum_{i=1}^{p}(d_i - y_i)$$

$Q_w = \log(p(w,\theta))$

$$\log(p_G) = \sum_{i=1}^{n} \frac{{w_i}^2}{2\sigma^2} = \gamma \sum_{i=1}^{n} {w_i}^2 + const \quad (5)$$

$$\log(p_L) = \sum_{i=1}^{n} \frac{|w_i|}{\lambda} = \gamma \sum_{i=1}^{n} |w_i| + const \quad (6)$$

pochodne f. kary (Q_w) względem w

$$\frac{\partial \log(p_G)}{\partial w_i} = 2\gamma w_i \tag{7}$$

$$\frac{\partial \log(p_L)}{\partial w_i} = \gamma \operatorname{sign}(w_i) \tag{8}$$

$$\tag{9}$$

Algorytm uczenia – szczegóły

- Zauważmy, że w minimum Q dostajemy:

$$\frac{\partial \log(Q)}{\partial w_i} = \frac{\partial \log(Q_D)}{\partial w_i} + \gamma \operatorname{sign}(w_i) = 0, \tag{10}$$

- zatem, $w_i = 0$, jeżeli $\left|\frac{\partial \log(Q_D)}{\partial w_i}\right| < \gamma$ oraz $w_i \neq 0$ o ile $\left|\frac{\partial \log(Q_D)}{\partial w_i}\right| = \gamma$.
- praktycznie zerujemy składową gradientu jeżeli:

$$|w_i| < 1E-10 \tag{11}$$

$$\left|\frac{\partial \log(Q_D)}{\partial w_i}\right| < \gamma. \tag{12}$$

Część XI

Metoda największej wiarygodności

Metoda największej wiarygodności

- Służy estymacji parametrów rozkładu
- Wykorzystuje funkcję wiarygodności

$$l(x, w) = \prod_{j=1}^{p} p(x_j, w)$$

- Rozwiązaniem jest

$$w^* = \text{argmax}_w l(x, w)$$

- Estymatory parametrów uzyskane MNW maja dobre własności

Metoda największej wiarygodności (przykład)

x_i -próbki brane z populacji o rozkładzie normalnym

$$p(x,\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$\begin{aligned}
l(x,\mu) &= \prod_j \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{(x_j-\mu)^2}{2\sigma^2}} \\
L(x,\mu) &= \sum_j \log\left(\frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{(x_j-\mu)^2}{2\sigma^2}}\right) \\
L(x,\mu) &= \log\left(\frac{1}{\sigma\sqrt{2\pi}}\right) - \sum_j \frac{(x_j-\mu)^2}{2\sigma^2} \\
L'(x,\mu) &= \sum_{j=1}^{p} \frac{(x_j-\mu)}{\sigma^2} = 0 \\
\mu &= \frac{\sum_{j=1}^{p} x_j}{p}
\end{aligned}$$

Część XII

Algorytmy Klasteryzacji

Algorytm EM (ang. *Expectation-Maximization*)

Cechy algorytmu

- Służy estymacji nieznanych parametrów rozkładu z danych (np. klasteryzacja, uzupełnianie braków)
- Wykorzystuje logarytmiczną funkcję wiarygodności
- Bezpośrednia maksymalizacja tej funkcji jest często trudna ze względu na dużą liczbę parametrów
- Stosuje się cyklicznie na przemian dwa kroki E i M, które zapewniają zbieżność do optimum

Algorytm EM

Inicjacja

Wyznaczanie początkowych parametrów rozkładu

Iteracje

Krok E (szacowanie) obliczenie "prawdopodobieństw" (przynależności, wartości funkcji gęstości) warunkowych i łącznych dla próbek pomiarowych

Krok M (maksymalizacja) wyznaczanie nowych parametrów rozkładów (zwykle stosując metodę największej wiarygodności – sumując po danych)

Oznaczenia

$Pr(i), \pi_i$ prawdopodobieństwo zajścia zdarzenia i (np. i–ta składowa)

$Pr(i|x)$ prawdopodobieństwo klasy pod warunkiem danej (stopień przynależności) (prawdopodobieństwo posteriori, wyznaczane przy pomocy tw. Bayes'a)

$p(x), p(x|\mu_i, \sigma_i), p(x|i)$ wartość funkcji gęstości prawdopodobieństwa w danych oraz i–tej składowej — warunkowej funkcji gęstości

$\boldsymbol{p}$ liczba przykładów, rozmiar zbioru danych

$\boldsymbol{n}$ liczba składowych w mieszance (funkcji przynależności)

parametry modelu

$$
\begin{aligned}
M &= [\mu_1, \ldots, \mu_n] \\
\Sigma &= [\sigma_1, \ldots, \sigma_n] \\
\Pi &= [\pi_1, \ldots, \pi_n]
\end{aligned}
$$

Mieszanka rozkładów gaussowskich (ang. *Gaussian Mixture*)

Zakładamy łączny rozkład danych:

$$p(x, M, \Sigma, \Pi) = \sum_{i=1}^{n} p(x|\mu_i, \sigma_i)\pi_i = \sum_{i=1}^{n} \pi_i \frac{1}{\sigma_i\sqrt{2\pi}} e^{-\frac{(x-\mu_i)^2}{2\sigma_i^2}}$$

Funkcja wiarygodności i jej logarytm:

$$l(x, M, \Sigma, \Pi) = \prod_{j=1}^{p} \sum_{i=1}^{n} p(x|\mu_i, \sigma_i)\pi_i \tag{13}$$

$$L(x, M, \Sigma, \Pi) = \sum_{j=1}^{p} \log\left(\sum_{i=1}^{n} p(x|\mu_i, \sigma_i)\pi_i\right) \tag{14}$$

Żądamy maksimum przy ograniczeniach: $\sum_i \pi_i = 1$

Mieszanka rozkładów gaussowskich (EM)

$$
\begin{aligned}
L(x, M, \Sigma, \Pi) &= \sum_{j=1}^{p} \log\Big(\sum_{i=1}^{n} p(x_j|\mu_i, \sigma_i)\pi_i\Big) \\
&= \sum_{j=1}^{p} \log\Big(\sum_{i=1}^{n} Pr(i|x_j)\frac{p(x|\mu_i, \sigma_i)\pi_i}{Pr(i|x_j)}\Big) \\
&\geq \sum_{j=1}^{p}\sum_{i=1}^{n} Pr(i|x_j)\log\Big(\frac{p(x|\mu_i, \sigma_i)\pi_i}{Pr(i|x_j)}\Big) \\
&= \sum_{j=1}^{p}\sum_{i=1}^{n} Pr(i|x_j)\log\big(p(x|\mu_i, \sigma_i)\pi_i\big) \\
&\quad -Pr(i|x_j)\log\big(Pr(i|x_j)\big)
\end{aligned}
$$

Mieszanka rozkładów gaussowskich (EM)

Maksymalizujemy L przy ograniczeniach z użyciem mnożnika Lagrange'a λ:

$$\begin{aligned} Q(x, M, \Sigma, \Pi) \quad = \quad & \sum_{j=1}^{p} \sum_{i=1}^{n} Pr(i|x_j) \log\big(p(x|\mu_i, \sigma_i)\pi_i\big) - \\ & Pr(i|x_j) \log\big(Pr(i|x_j)\big) - \\ & \lambda\Big(\sum_i \pi_i - 1\Big) \end{aligned}$$

oraz (krok E)

$$Pr(i|x) = \frac{\pi_i p(x|\mu_i, \sigma_i)}{\sum_{i=1}^{n} \pi_i p(x|\mu_i, \sigma_i)}$$

Mieszanka rozkładów gaussowskich (EM)

$\mu_i, \sigma_i, \pi_i, i = 1, \ldots n$ wyznaczamy z warunków:

$$\begin{aligned} \frac{\partial Q}{\partial \mu_i} &= 0 \\ \frac{\partial Q}{\partial \sigma_i} &= 0 \\ \frac{\partial Q}{\partial \pi_i} &= 0 \end{aligned}$$

Mieszanka rozkładów gaussowskich (EM)

Krok E:

$$Pr(i|x) = \frac{\pi_i p(x|\mu_i, \sigma_i)}{\sum_{i=1}^{n} \pi_i p(x|\mu_i, \sigma_i)}$$

Krok M[2]:

$$\begin{aligned}
\mu_i &= \frac{\sum_{j=1}^{p} Pr(i|x_j) x_j}{\sum_{j=1}^{p} Pr(i|x_j)} \\
\sigma_i &= \frac{\sum_{j=1}^{p} Pr(i|x_j)(x_j - \mu_i)^2}{\sum_{j=1}^{p} Pr(i|x_j)} \\
\pi_i &= \frac{1}{n} \sum_{j=1}^{p} Pr(i|x_j)
\end{aligned}$$

[2] por. defuzyfikację metodą środka ciężkości – zwłaszcza μ_i)

Algorytm K-Środków definicje

Punkt należy do sąsiedztwa

Mówimy, że punkt X^i należy do sąsiedztwa środka C^{j^*} jeżeli:

$$j^* = \text{argmin}_{j=1,\ldots,K} d(X^i, C^j)$$

Sąsiedztwo

Sąsiedztwem środka C^j nazywamy zbiór wszystkich punktów X^i, które należą do jego sąsiedztwa.

Algorytm Kmeans

```
function C=KMEANS(X)
    C^i(0) ∈ X, i = 1,2,...,K; (losujemy początkowe środki z danych)
    while ∃_{i=1,...,K} C^i(n−1)! = C^i(n) do (nastąpiła zmiana K-środków ?)
        for all j = 1,...,K do
            S(C_i(n−1)) = {x ∈ X: x = argmin_{x∈X}(d(C_i, x)}
            C_i(n) = mean S(C_i(n−1))
        end for
    end while
end function
```

Algorytm K-środków (ang. *K-means*)

Analogia do algorytmu EM

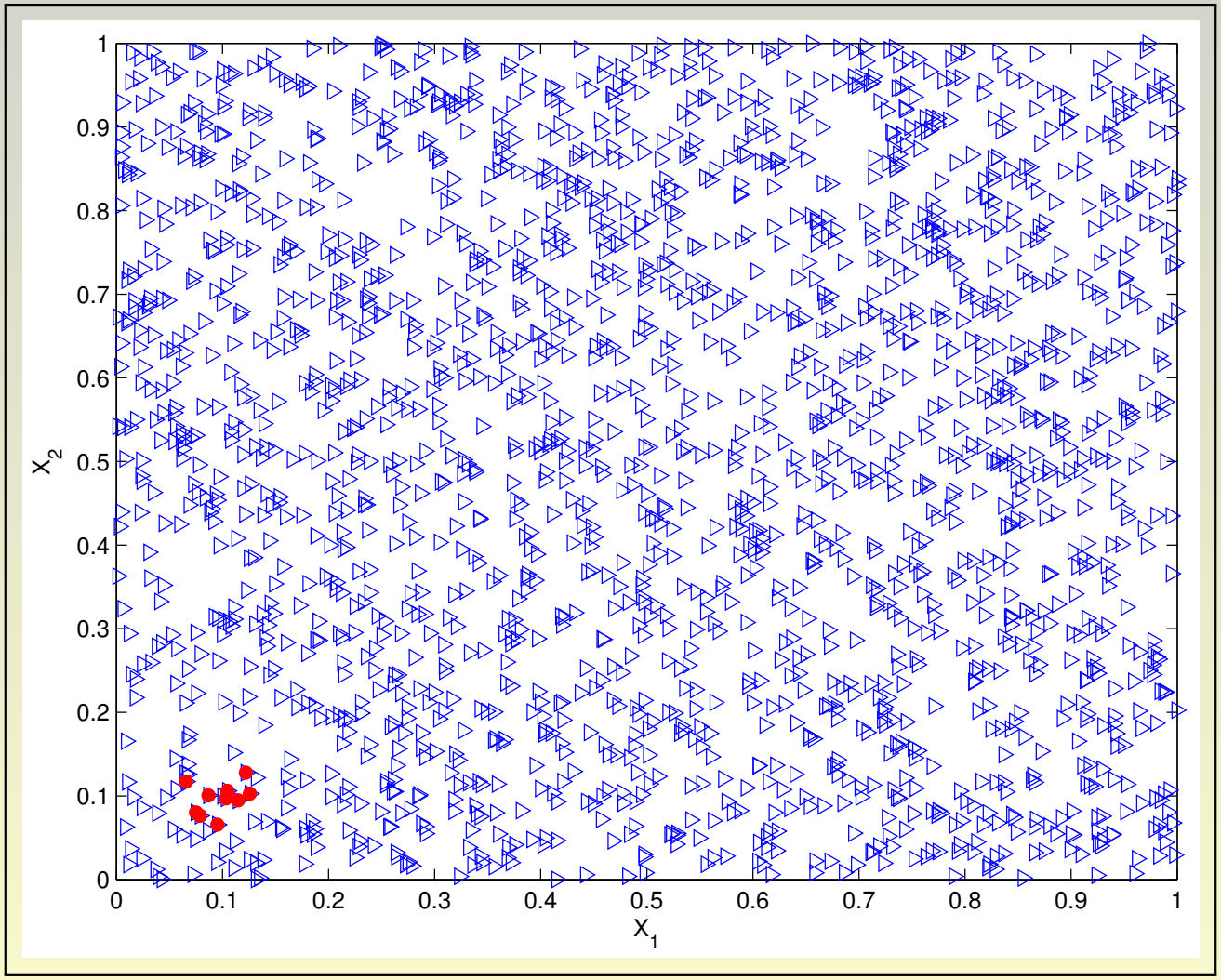
Krok E wyznaczenie sąsiedztwa $S(C_i)$ (funkcja charakterystyczna sąsiedztwa jest odpowiednikiem prawdopodobieństwa klasyfikacji)

$$Pr(C_i|x_j) = \chi_{C_j}(x_j)$$

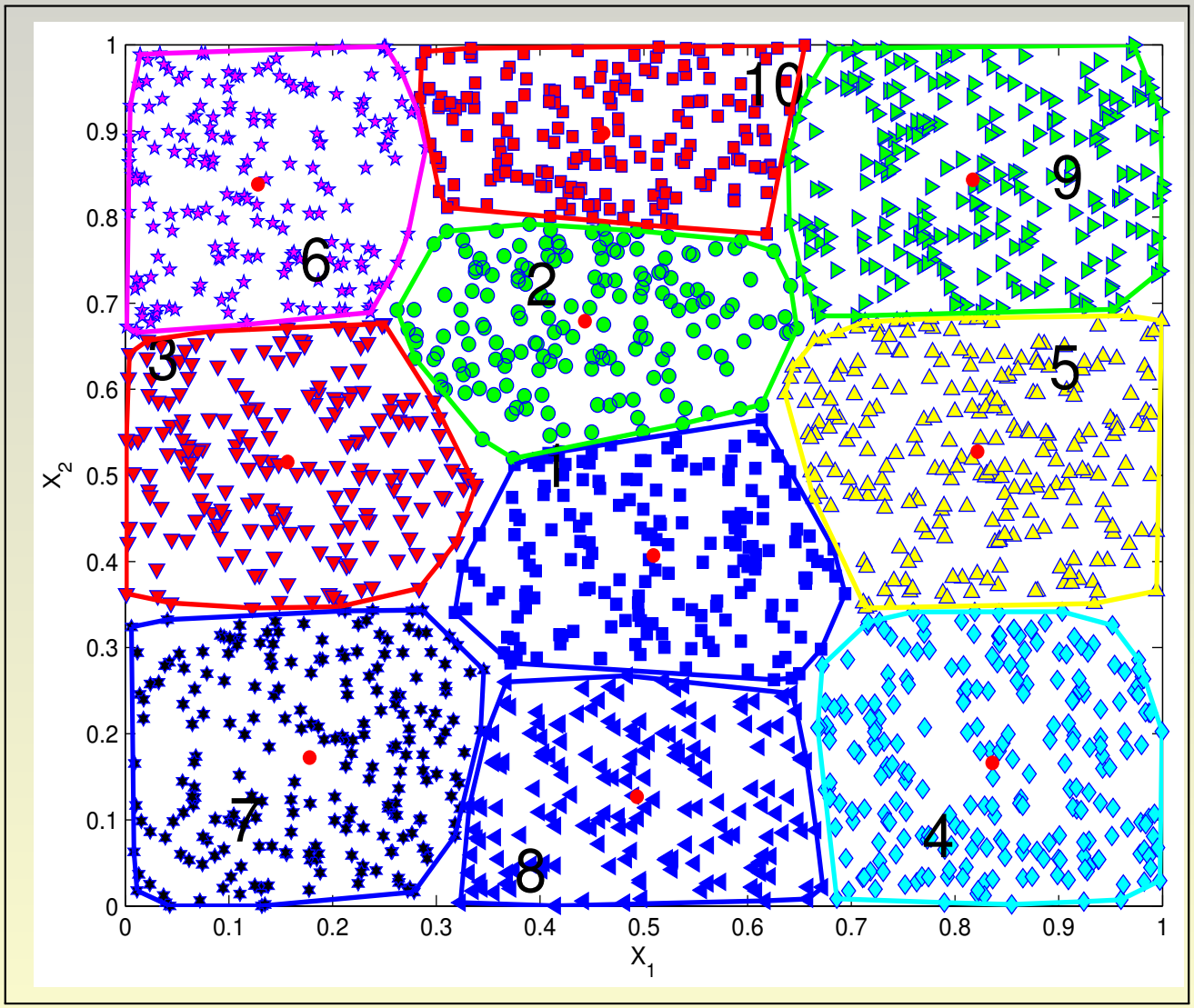
Krok M wyznaczenie centrów

$$C_i = \frac{\sum_{j=1}^{p} \chi_{C_i}(x_j) x_j}{\sum_{j=1}^{p} \chi_{C_i}(x_j)} = \text{mean}\, S(C_j)$$

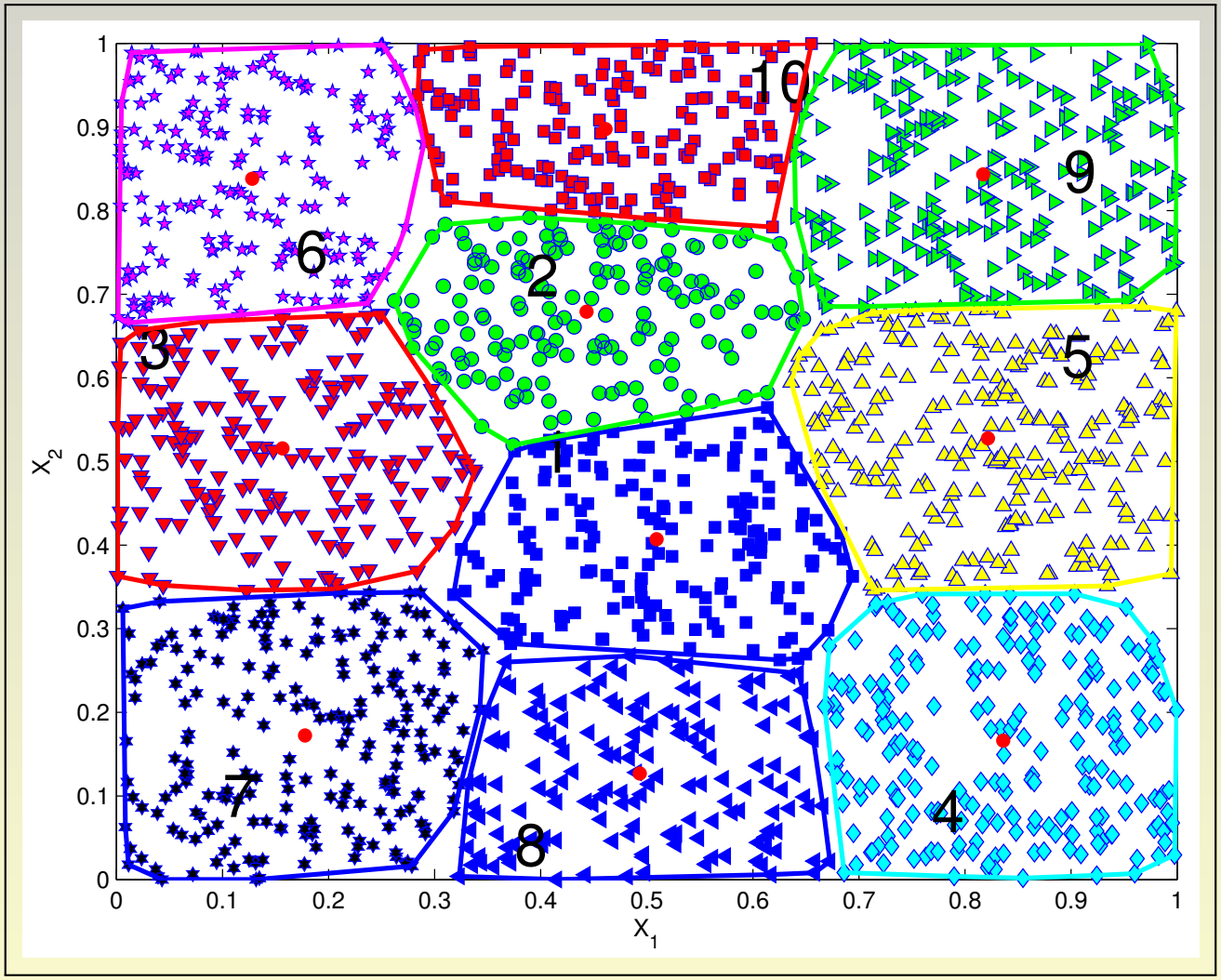
K-środki - przykłady: inicjacja



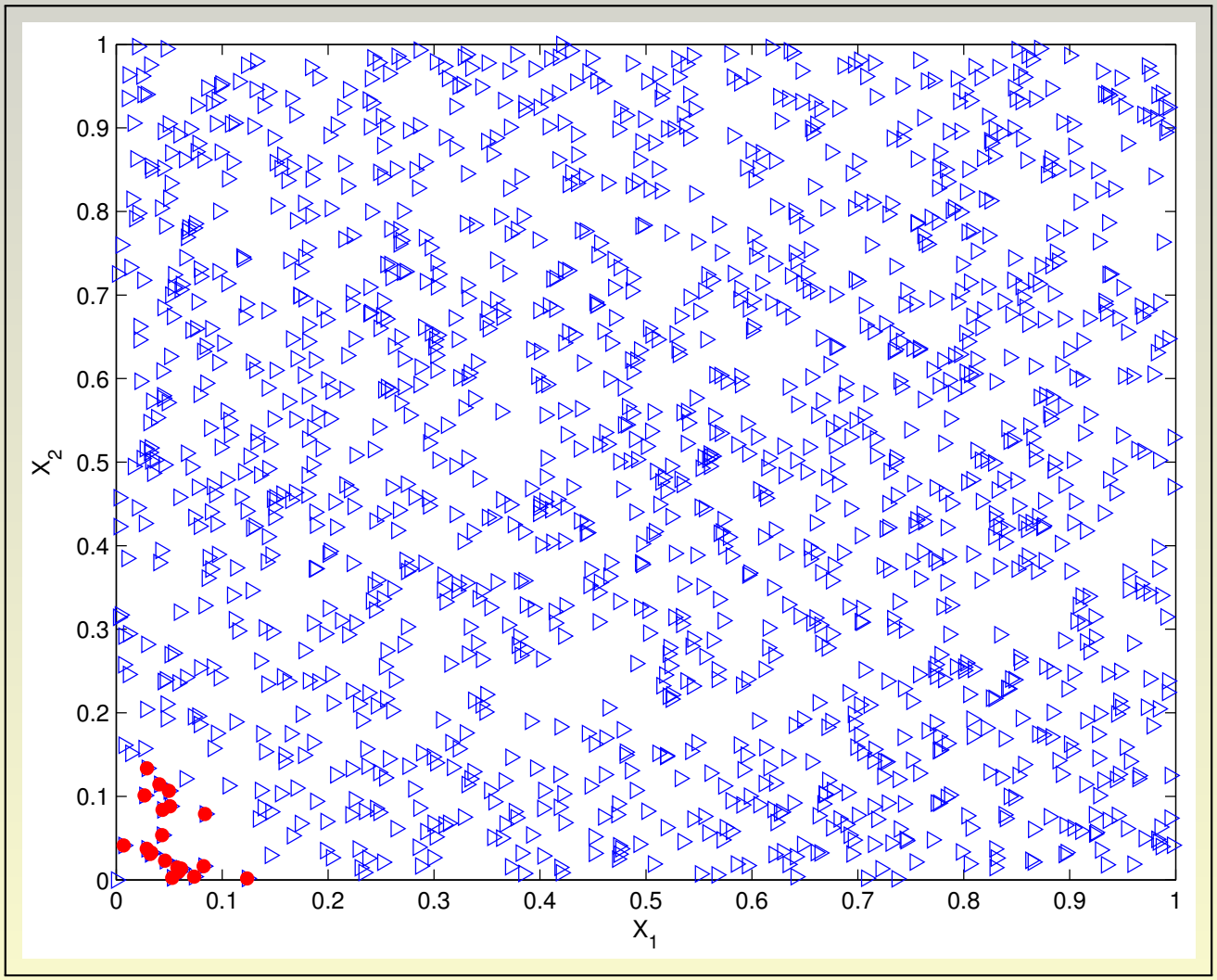
K-środki - przykłady: samoorganizacja



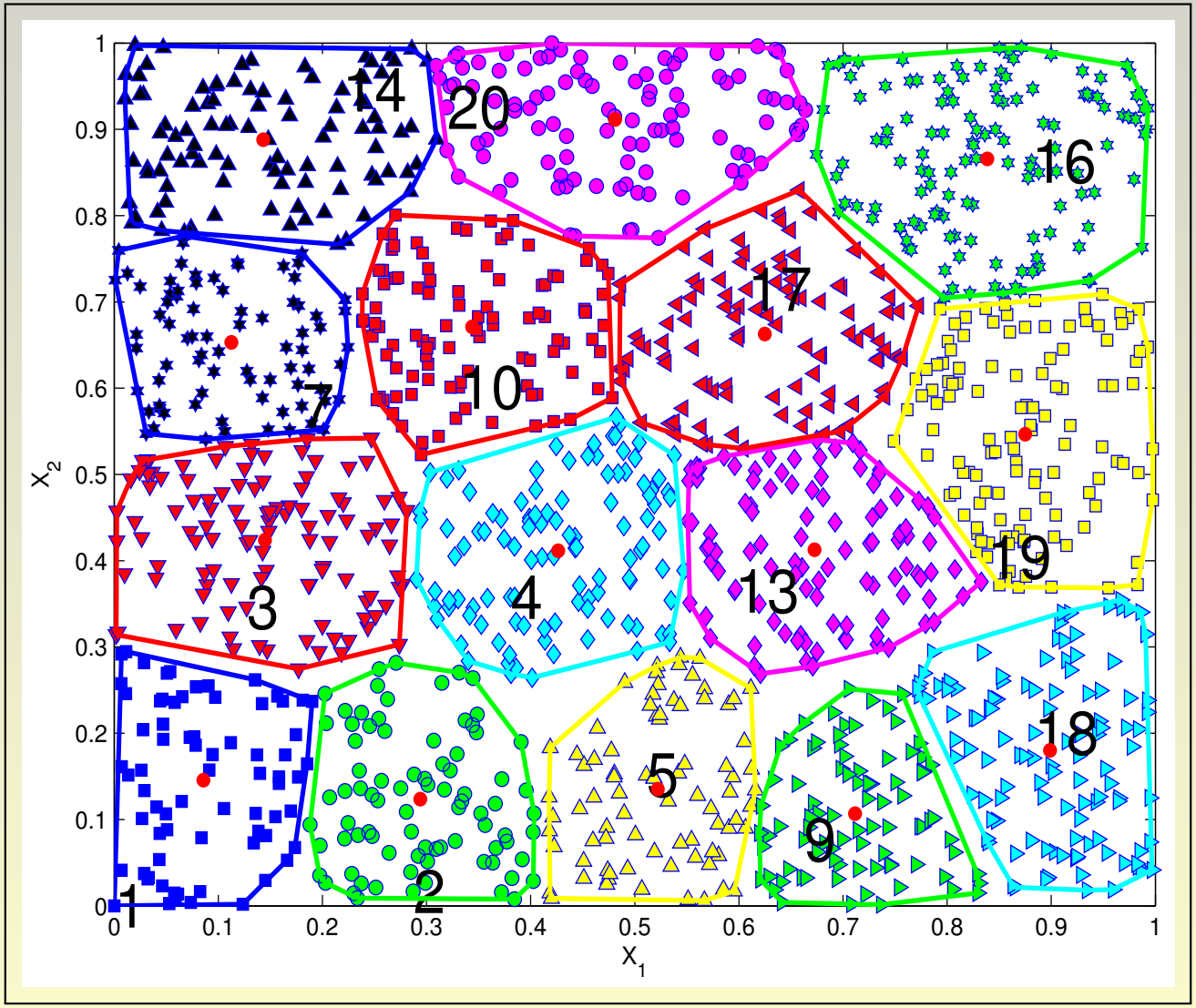
K-środki - przykłady: stan końcowy



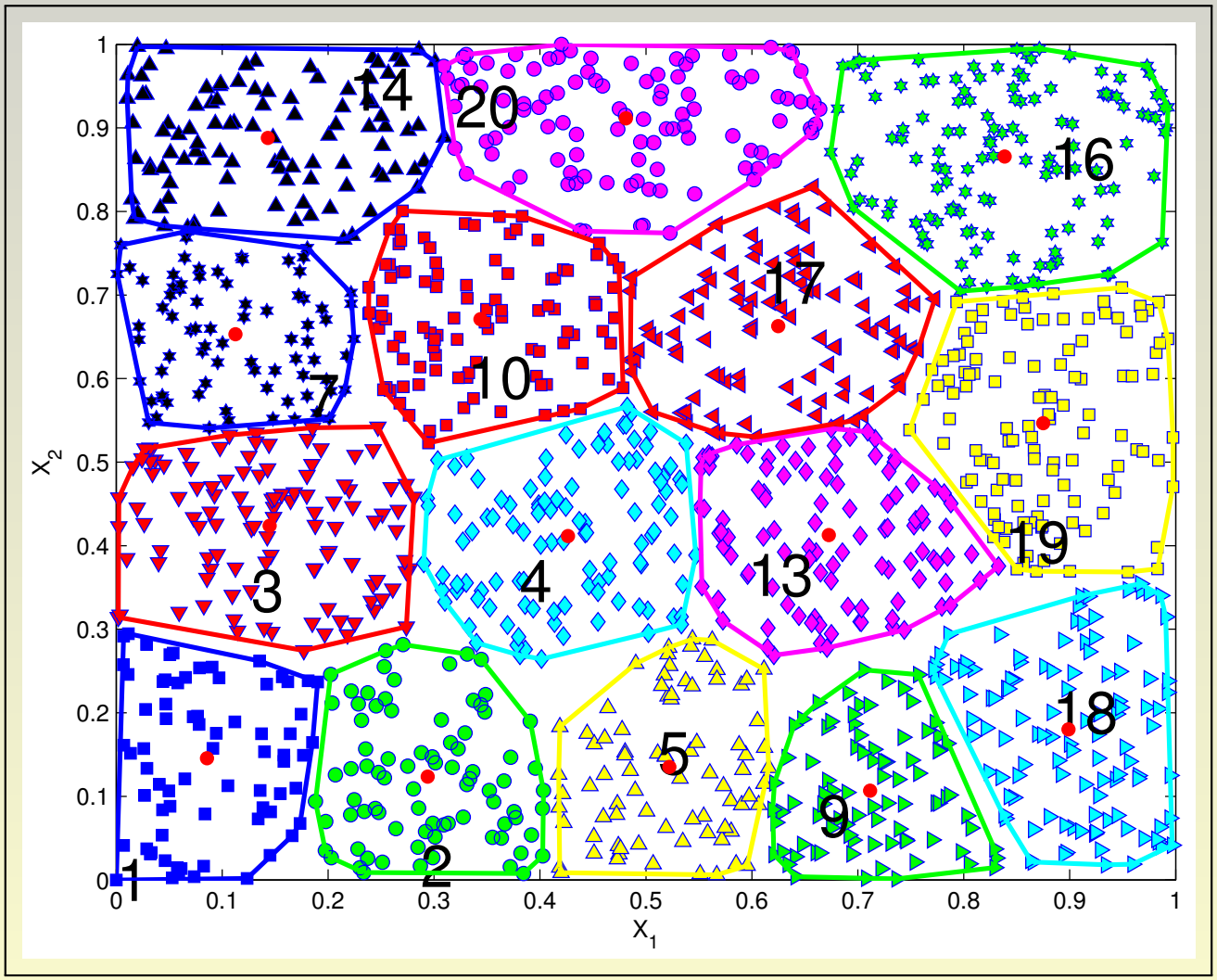
K-środki - przykłady: inicjacja



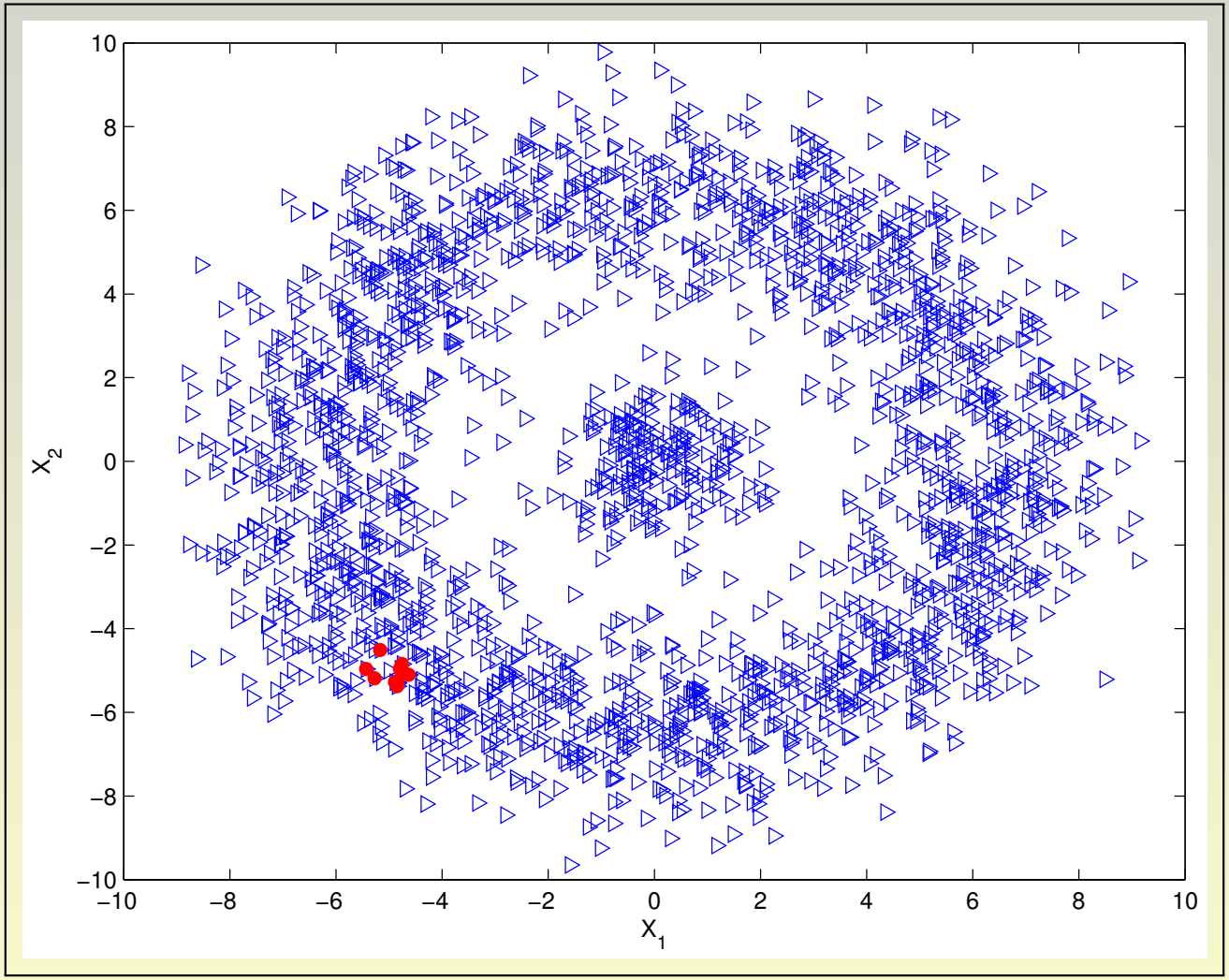
K-środki - przykłady: samoorganizacja



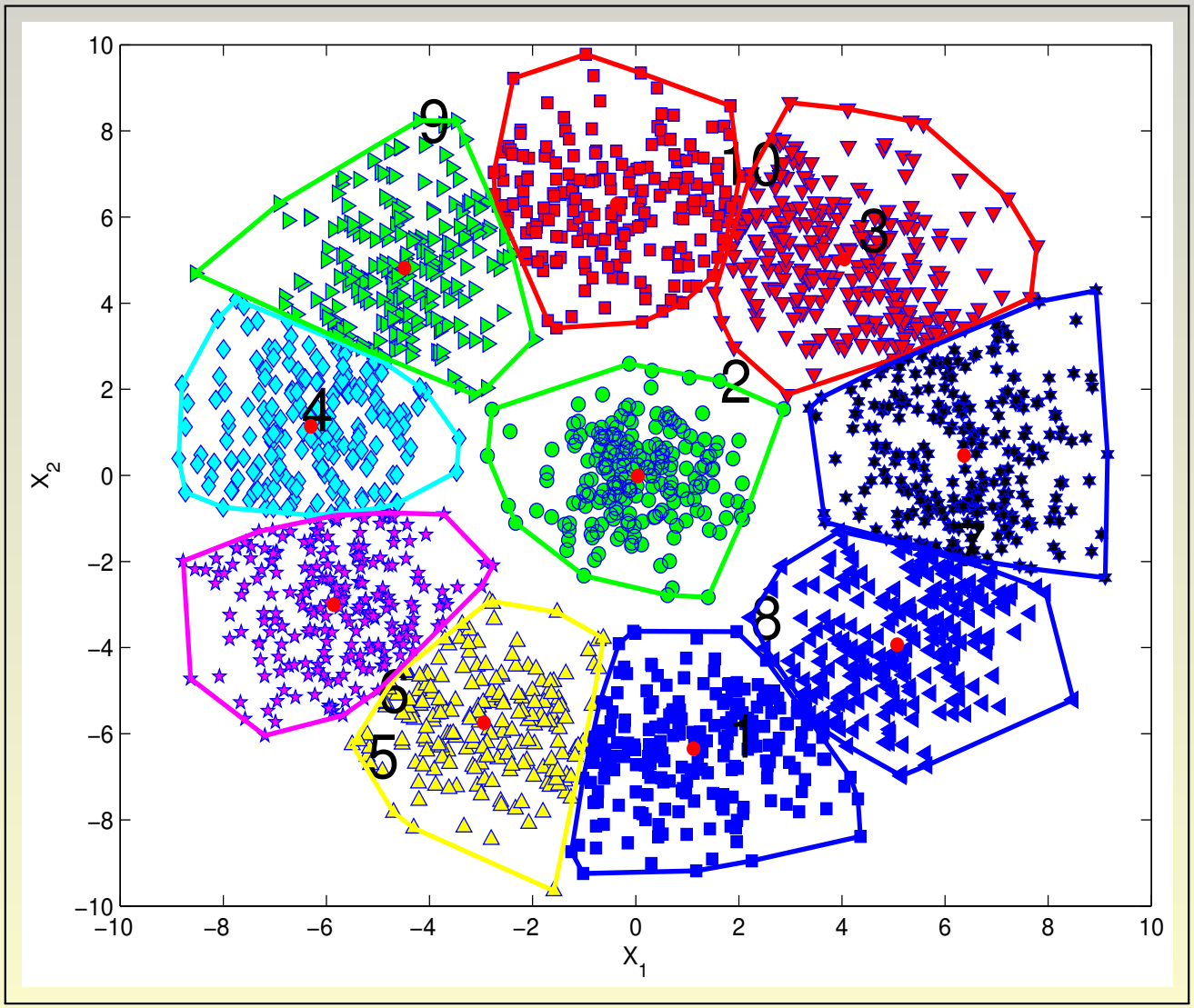
K-środki - przykłady: stan końcowy



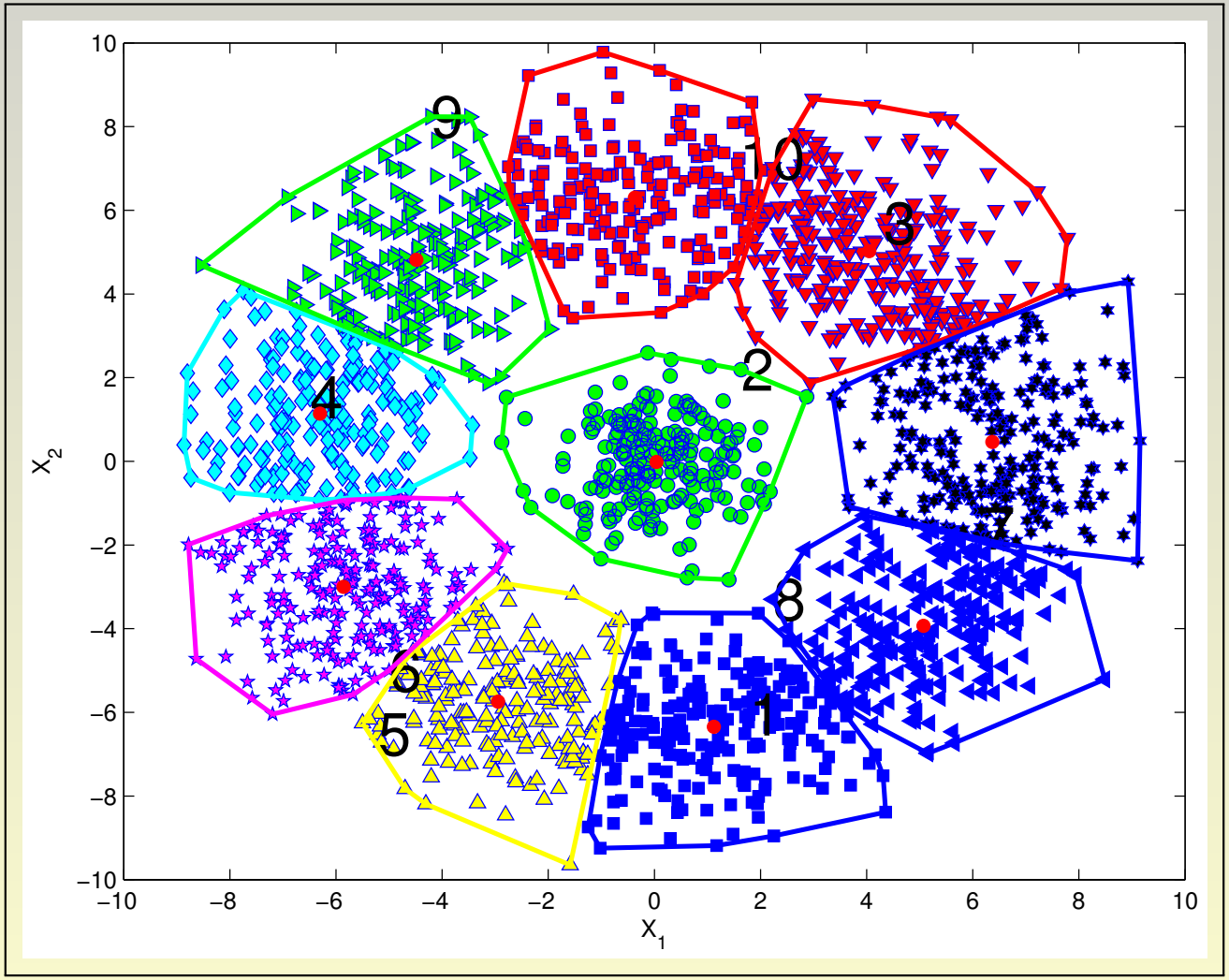
K-środki - przykłady: inicjacja



K-środki - przykłady: samoorganizacja



K-środki - przykłady: stan końcowy



Algorytm K-środków: problemy i uwagi

- Czy algorytm zatrzyma się dla każdych danych wejściowych?
- Jak dobrać liczbę centrów K?
- Jak wybrać początkowe położenia centrów?
- Czy podział na grupy jest uzasadniony z punktu widzenia rozkładu w danych? (odp.: ANOVA)

Grupowanie – ocena jakości

błąd kwantyzacji

$$e(\mathbf{c}) = \sum_{p=1}^{P} \|\mathbf{x}^{(p)} - \mathbf{c}^*(\mathbf{x}^{(p)})\|$$

gdzie

$\mathbf{c}^*(\mathbf{x})$ jest środkiem sąsiedztwa dla punktu $\mathbf{x}$

Problem

Jak wybrać liczbę klas / funkcji przynależności w przypadku bez nadzoru?

BIC (ang. *Bayesian information criterion*)

$$BIC = Loglikelihood - liczbaParametrow \cdot \log(rozmiarDanych)$$

szukamy takiej liczby klas która maksymalizuje *BIC*

BIC (ang. *Bayesian information criterion*)

$$BIC = Loglikelihood - liczbaParametrow \cdot \log(rozmiarDanych)$$

szukamy takiej liczby klas która maksymalizuje *BIC*

Inny sposób oceny jakości klasteryzacji to analiza wariancji

Algorytm EM w Matlabie

Pakiet `Netlab` (autorzy: Ian Nabney, Christopher Bishop) dostępny pod adresem: `http://www.ncrg.aston.ac.uk/netlab/book.php`

Przydatne funkcje pakietu `netlab` (uczenie)

`[centres] = kmeans(centres, data)` algortym K–środków

`mix = gmm(dim, ncentres, covartype)` tworzy GMM model o zadanym typie macierzy kowariancji

`mix = gmminit(mix, x, options)` inicjuje parametry modelu GMM algorytmem K–środków

`mix = gmmem(mix, x, options)` dobór parametrów modelu GMM algorytmem EM

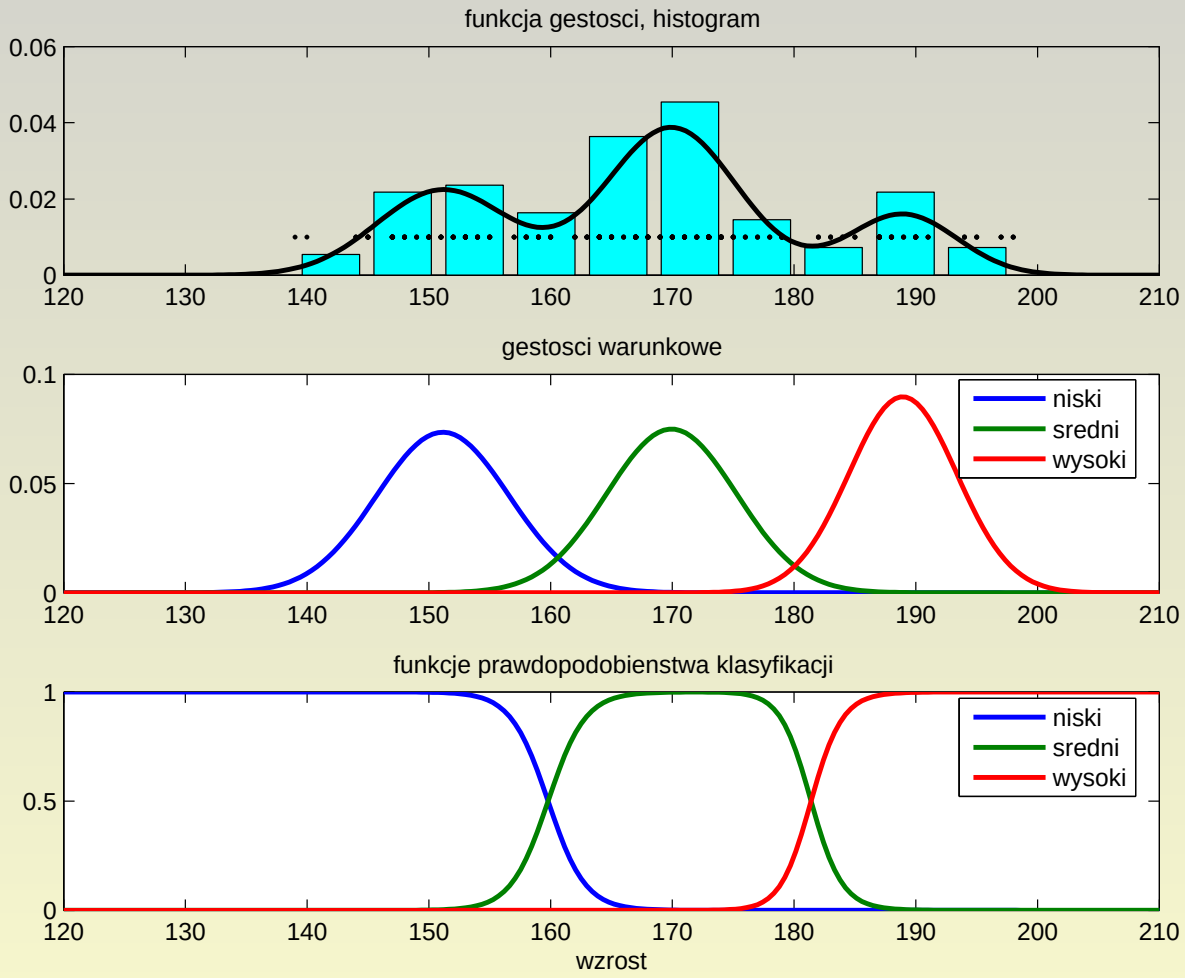
Przydatne funkcje pakietu `netlab` (wnioskowanie)

`pdata = gmmprob(mix, x)` funkcja gęstości mieszanki rozkładów gaussowskich

`pcidata = gmmpost(mix, x)` prowdopodobieństwa posteriori, klasy pod warunkiem obserwacji

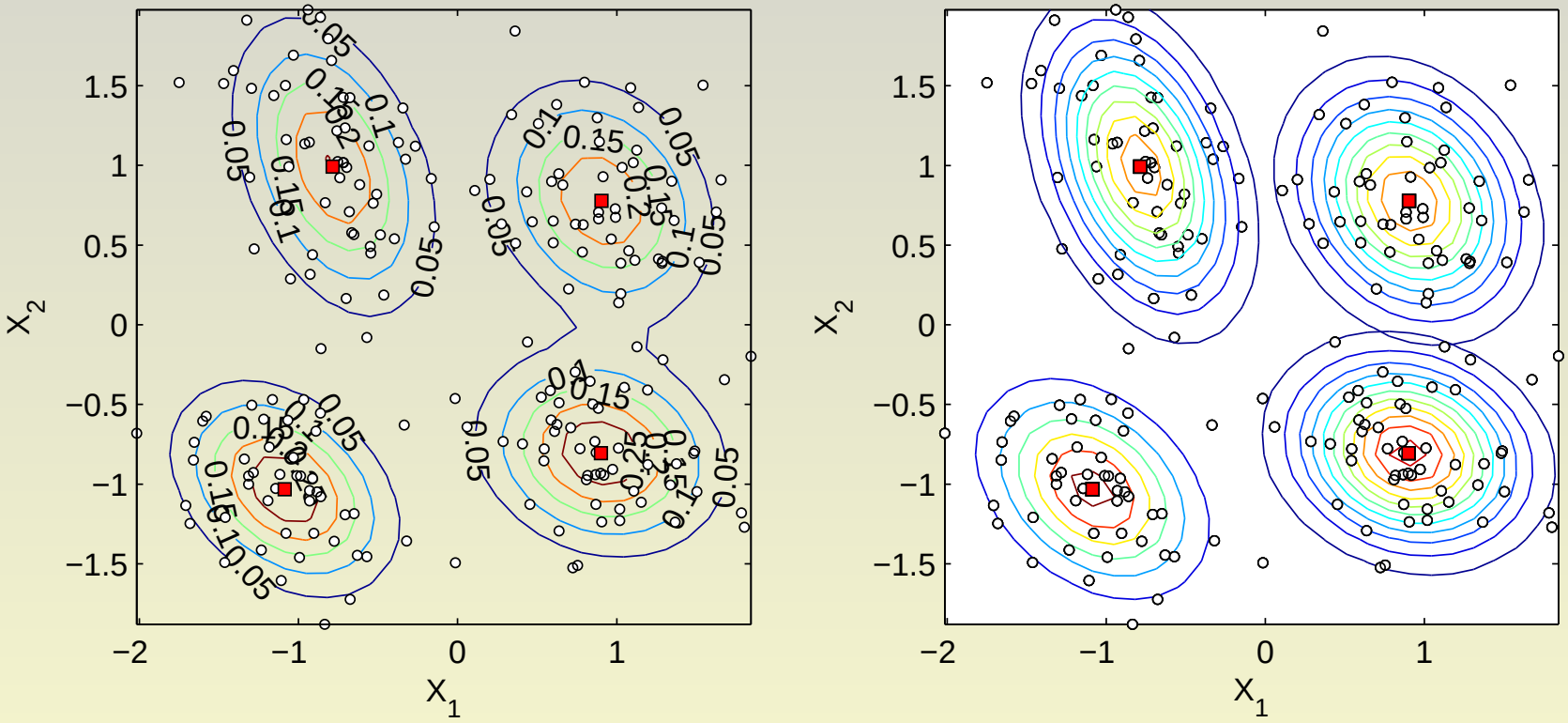
`mix = gmmactiv(mix, x)` funkcje aktywacji, gausowskie składowe mieszanki

Przypadek jednowymiarowy



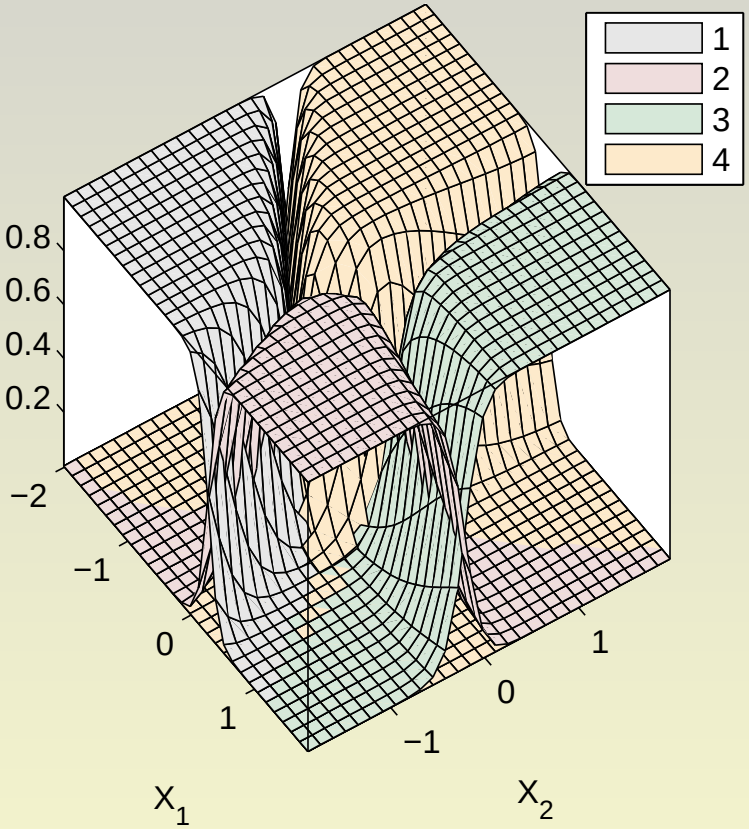
Zastosowanie algorytmu EM do dyskretyzacji

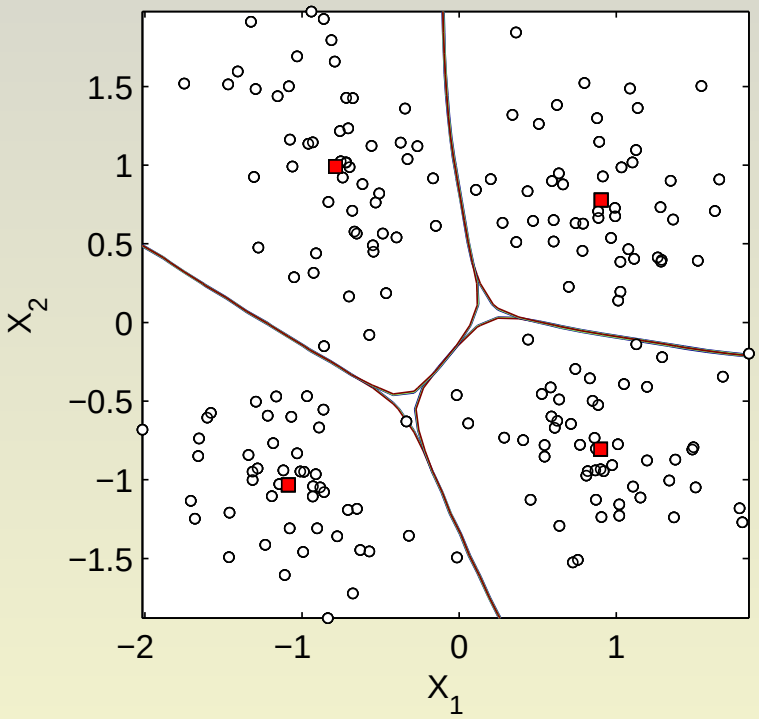
Przykład 2D



Rozkład łączny i rozkłady warunkowe klastrów.

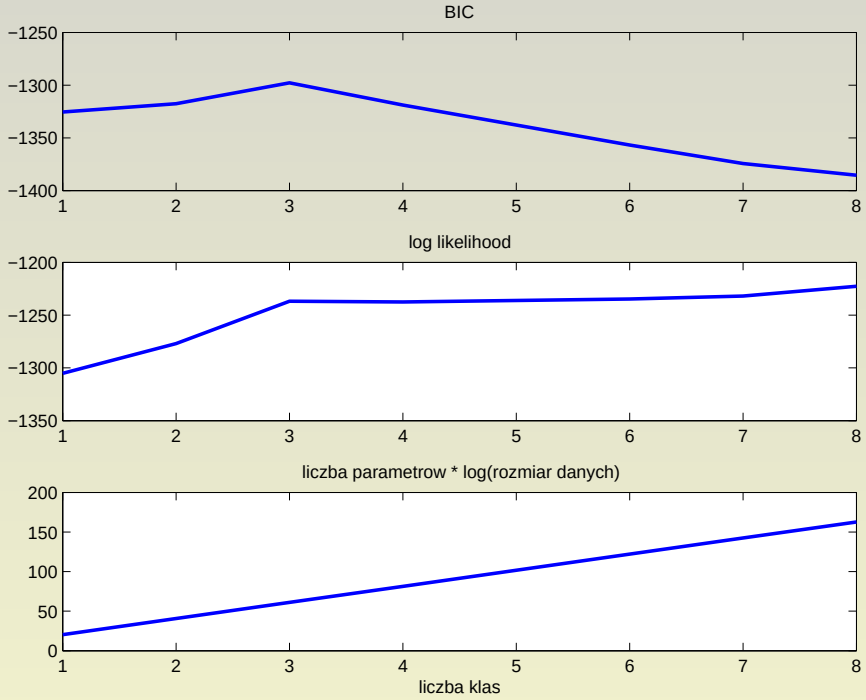
Przykład 2D



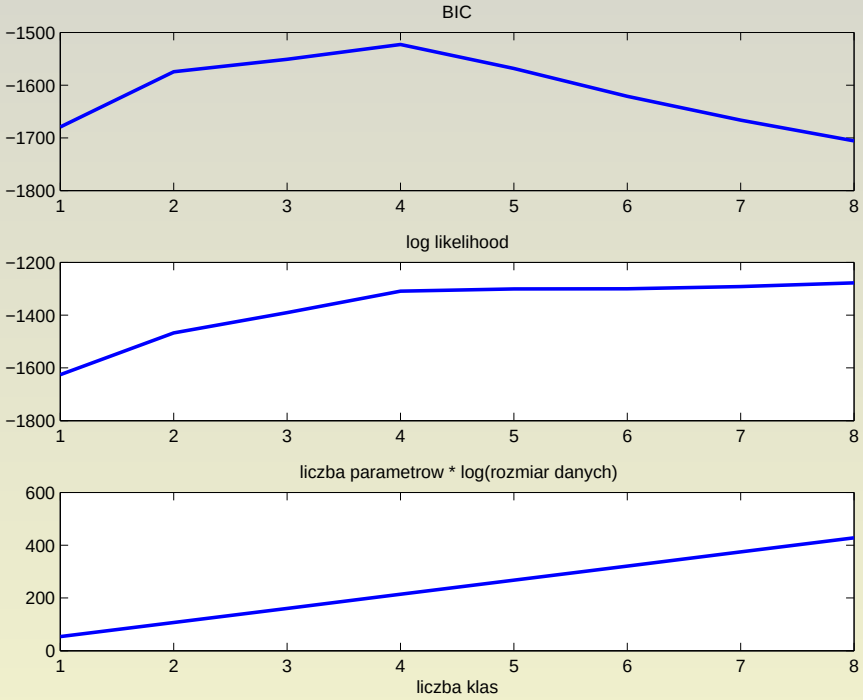


Prawdopodobieństwa posteriori

BIC (ang. *Bayesian information criterion*)



Rys. BIC przypadek jednowymiarowy



Rys. BIC przypadek dwuwymiarowy

Część XIII

Indukcja Reguł Decyzyjnych

Algorytm apriori

- Służy do wykrywania interesujących wzorców w danych
- Dedykowany bazom danych w postaci transakcji
- Kryteria interesującości: wsparcie i zaufanie
- Standardowo pracuje na danych binarnych
- Główna idea: wsparcie jest monotoniczną funkcją zbioru

Transakcje – koszyki sklepowe

T_{ID}	A	B	C	D	E	$\cdots$
t_1	1	1	1	0	0	$\cdots$
t_2	0	0	0	1	1	$\cdots$
t_3	0	0	0	1	0	$\cdots$
t_4	0	0	0	1	1	$\cdots$
t_5	0	0	0	0	1	$\cdots$
t_6	1	0	0	1	0	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$

Czy tak zapisujemy transakcje sklepowe?

T_{ID}	P_{ID}	ilosc
t_1	A	1
t_1	B	1
t_1	C	1
t_2	D	1
t_2	E	1
t_3	D	1
t_4	D	1
t_4	E	1
t_5	E	1
t_6	A	1
t_6	D	1
$\vdots$	$\vdots$	$\vdots$

Czy może tak lepiej?

Definicje

zbiór cech (ang. *itemset*) $X \subset P$ – dowolny podzbiór zbioru atrybutów

wsparcie (ang. *support*) $supp(X)$ liczba transakcji zawierających atrybuty $X \subset P$. (przykładowo liczba wszystkich koszyków w bazie, w których występują piwo i pieluchy)

zbiór częsty (frequent itemset) zbiór spełniający relację: $supp(X) > min_{supp}$

Definicje

reguła asocjacyjna implikacja $X \rightarrow Y$, gdzie X, Y dowolne podzbiory atrybutów

wsparcie reguły $supp(X \rightarrow Y) = supp(X \cup Y)$

zaufanie (ang. *confidence*) $conf(X \rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)}$ (przykładowo $conf(piwo \rightarrow pieluchy) = \frac{supp(piwo \cup pieluchy)}{supp(piwo)}$)

Definicje

Uwaga 1. Jeżeli $X \cup Y$ jest częsty to zarówno X jest częsty jak i Y, to znaczy:

$$A \subset B \Rightarrow supp(A) \geq supp(B)$$

Uwaga 2. $conf(XY \rightarrow Z) = \frac{supp(XYZ)}{supp(XY)} \geq \frac{supp(XYZ)}{supp(X)} = conf(X \rightarrow YZ)$

Uwaga 3. sensownie jest przyjąć, że

$$supp(\emptyset) = |D|$$

oraz

$$conf(\emptyset \rightarrow Y) = \frac{supp(Y)}{|D|}$$

Algorytm Apriori

Wejście: zbiór danych, min_{supp}, min_{conf}

zbiory częste znajdujemy wszystkie zbiory częste korzystając z monotoniczności wsparcia

generowanie reguł ze zbiorów częstych generujemy reguły o zaufaniu większym niż min_{conf}, dzieląc zbiór częsty na dwa podzbiory

Apriori - generowanie zbiorów częstych

- C_i - lista zbiorów kandydujących na zbiory częste składających się i atrybutów
- F_i - lista zbiorów częstych składających się i atrybutów

Apriori - generowanie zbiorów częstych

- $i = 0$, C_i - inicjujemy listę jednoelementowymi zbiorami atrybutów
- Powtarzaj (dopóki lista kandydatów C_i jest niepusta)
  1. wstawiamy do F_i wszystkie częste z C_i oraz zapamiętujemy ich wsparcia
  2. tworzymy zbiory kandydujące C_{i+1} (długości $i+1$) ze wszystkich $F_j, j = 1, \ldots, i$ (zbiór częsty musi być nadzbiorem zbiorów częstych, zestawiamy ze sobą wszystkie uzyskane zbiory częste w pary oraz sprawdzamy czy ich suma ma długość $i+1$ oraz czy wszystkie danej pary podzbiory są częste)
  3. $i++$
- zwracamy rodzinę $\{F_i\}$ zbiorów częstych

Apriori - generowanie reguł

- Iterujemy po wszystkich regułach $X \Rightarrow Y$, gdzie zarówno X i XY jest częsty
- Sprawdzamy czy reguły mają odpowiednie zaufania (ta część nie wymaga już danych bo odpowiednie wsparcia są już wyznaczone)

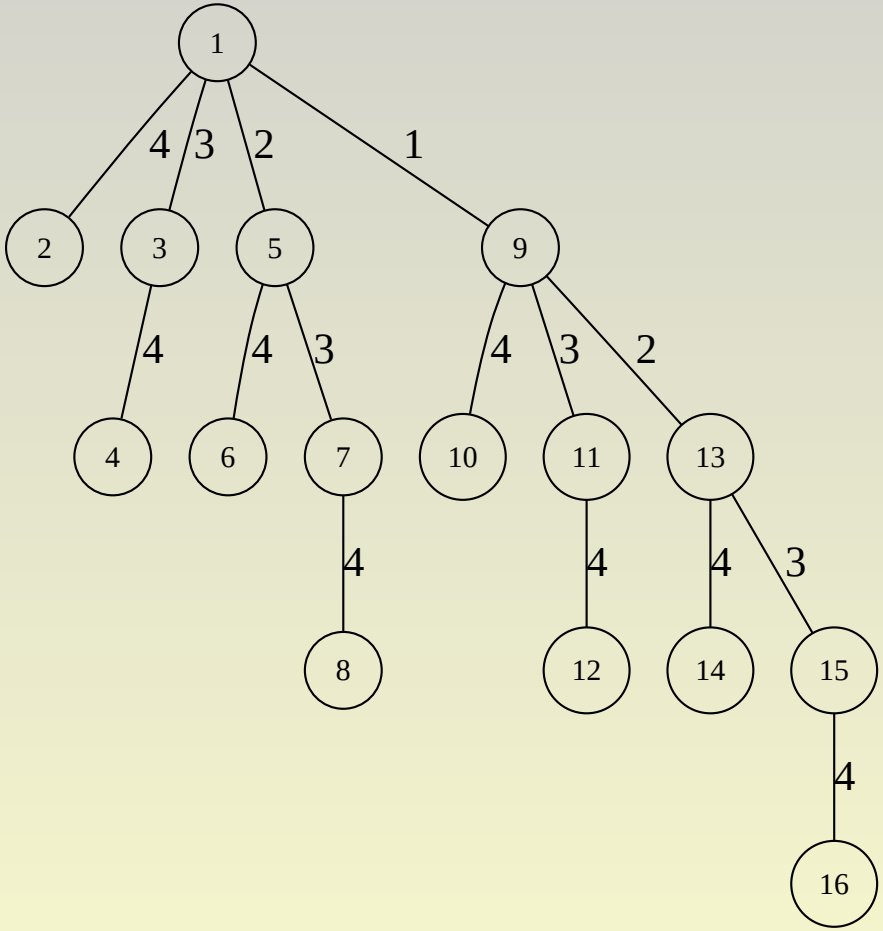
Algorytm z drzewkiem wyliczającym podzbiory

- Służy do generowania zbiorów częstych
- Prostsza w implementacji alternatywa dla algorytmu apriori
- Jeżeli oba algorytmy są napisane wydajnie jest trochę mniej wydajny niż apriori
- Główna idea:
  - Iterujemy po wszystki podzbiorach
  - Wszystkie częste zbiory atrybutów wstawiamy do listy S
  - Minimalne zbiory nieczęste wstawiamy do listy B
  - Przed wyznaczeniem wsparcia sprawdzamy czy rozważany zbiór atrybutów nie jest nadzbiorem któregoś zbioru z listy B

Algorytm budowy drzewa (do końca)

```
function T=GENTREE(atrs, N)
    for i = [N : −1 : max(atrs) + 1] do
        gentree([atrs i], N)
    end for
end function
```

Przeliczanie podzbiorów

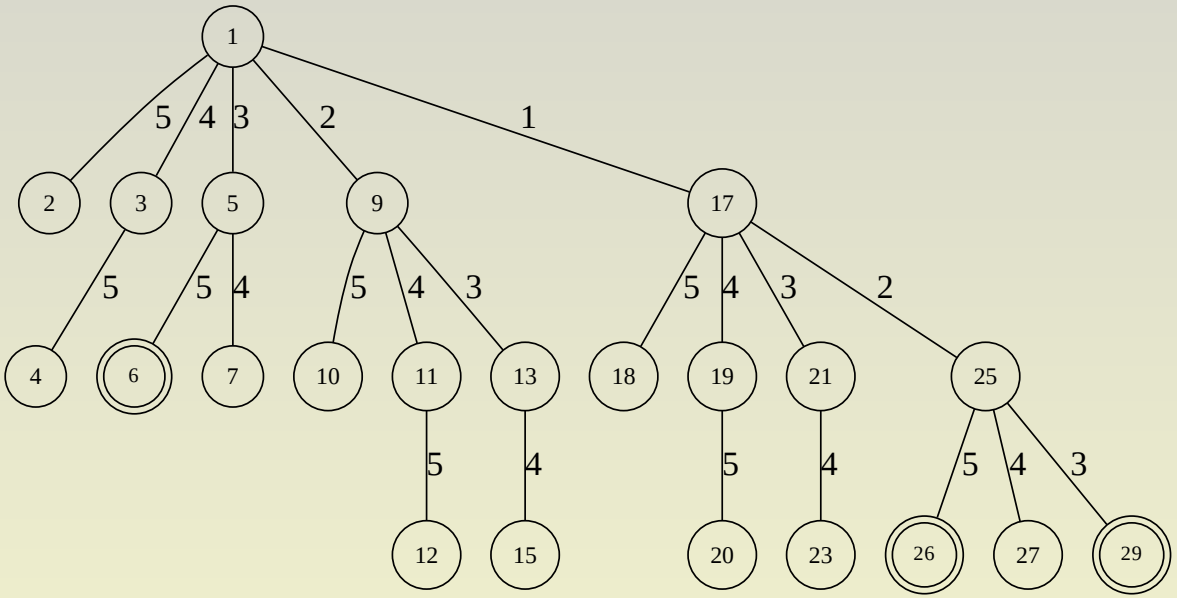


Pełne drzewo wyliczania podzbiorów

{}
{4}
{3}
{3, 4}
{2}
{2, 4}
{2, 3}
{2, 3, 4}
{1}
{1, 4}
{1, 3}
{1, 3, 4}
{1, 2}
{1, 2, 4}
{1, 2, 3}
{1, 2, 3, 4}

kolejność przeglądania

Generowanie zbiorów częstych



Drzewo poszukiwania atrybutów częstych wraz z elementami na brzegu negatywnym

{}
{5}
{4}
{4, 5}
{3}
{3, 4}
{2}
{2, 5}
{2, 4}
{2, 4, 5}
{2, 3}
{2, 3, 4}
{1}
{1, 5}
{1, 4}
{1, 4, 5}
{1, 3}
{1, 3, 4}
{1, 2}
{1, 2, 4}

zbiory częste

{3, 5}
{1, 2, 5}
{1, 2, 3}

brzeg minimalne – nieczęste

kolejność przeglądania

Testowanie statystyczne reguł

- Każda reguła jest potencjalnie interesującym wzorcem
- Testowanie istotności powinno być przeprowadzane na niezależnym zbiorze testowym
- Ze względu na dużą liczbę reguł a co za tym idzie testów trzeba korygować poziom istotności (np. stosując poprawkę Bonferroniego)
- Można ograniczyć liczbę reguł rozważając tzw. reguły pareto optymalne – takie które nie są zdominowane przez inne reguły na wykresie (wsparcie, zaufanie)