

# WYKŁAD: Klasyfikacja logistyczna

Zaawansowane Metody Uczenia Maszynowego

Rozpatrywane dotąd metody klasyfikacji:

- LDA Fishera (liniowa reguła klasyfikacyjna);
- Reguła Bayesowska (jej wersja empiryczna dla rozkładów normalnych ze wspólną macierzą  $\Sigma$  pokrywa się z LDA).

Inne metody liniowe ?

Klasyfikacja logistyczna oparta na modelu regresji logistycznej.

Jedno z zastosowań: reklamy pojawiające się na stronie są dobierane na podstawie modelu regresji logistycznej/probitowej gdzie zmiennymi objaśniającymi są słowa kluczowe.

# Regresja logistyczna

$$P(Y = 1|\mathbf{x}) = \pi(\mathbf{x}) = E(Y|\mathbf{x}) \quad \mathbf{x} \in R^P$$

$\pi(\mathbf{x})$  – modelujemy zależność  $\pi$  od  $\mathbf{x}$ , a nie  $Y$  od  $\mathbf{x}$ .

$$\text{logit}(\pi(\mathbf{x})) := \log\left(\frac{\pi(\mathbf{x})}{1 - \pi(\mathbf{x})}\right) = \beta' \mathbf{x},$$

$$\pi(\mathbf{x}) = \frac{\exp(\beta' \mathbf{x})}{1 + \exp(\beta' \mathbf{x})}$$

# Metody estymacji

- Metoda największej wiarygodności (iteracyjna procedura IWLS: dopasowanie modeli liniowych metodą WNK)
- Estymator Blyth'a zmniejszający obciążenie estymatora  $\hat{\beta}_{NW}$ ;
- regularyzowany estymator Lasso

$$\operatorname{argmin}_{\mathbf{b}} \left\{ -2 \times \log \mathcal{L} + \lambda \sum_{i=1}^p |b_i| \right\}$$

i różne jego warianty.

## Model logistyczny dla $Y$ o $g$ wartościach

Wybieramy populację referencyjną np. ostatnią (o numerze  $g$ )

$$\log \frac{p(1|\mathbf{x})}{p(g|\mathbf{x})} = \beta'_1 \mathbf{x}$$

$$\log \frac{p(2|\mathbf{x})}{p(g|\mathbf{x})} = \beta'_2 \mathbf{x}$$

.....

$$\log \frac{p(g-1|\mathbf{x})}{p(g|\mathbf{x})} = \beta'_{g-1} \mathbf{x}$$

$$\beta'_i = (\beta_{i0}, \beta_{i1}, \dots, \beta_{ip})'$$

Nieznane parametry:  $\beta'_1, \dots, \beta'_{g-1} \in R^{p+1}$ , łącznie  $(g-1)(p+1)$  parametrów jednowymiarowych.

**Uwaga** W pakiecie R jako populacja referencyjna wybierana jest ta, której nazwa jest pierwsza w porządku leksykograficznym.

Estymacja parametrów metodą NW:  $\hat{\beta}_1, \dots, \hat{\beta}_{g-1} \longrightarrow \hat{p}(1|\mathbf{x}), \dots, \hat{p}(g|\mathbf{x})$

$$\hat{p}(k|\mathbf{x}) = \frac{\exp(\hat{\beta}'_k \mathbf{x})}{1 + \sum_{i=1}^{g-1} \exp(\hat{\beta}'_i \mathbf{x})} \quad k = 1, \dots, g-1$$

$$\hat{p}(g|\mathbf{x}) = \frac{1}{1 + \sum_{i=1}^{g-1} \exp(\hat{\beta}'_i \mathbf{x})}$$

### Reguła dyskryminacyjna

Reguła bayesowska oparta na estymatorach otrzymanych w modelu logistycznym:

Klasyfikuj do populacji  $l$  gdzie  $l = \underset{i}{\operatorname{argmax}} \hat{p}(i|\mathbf{x})$

Zauważmy, że w modelu logistycznym w naturalny sposób otrzymujemy oszacowania interesujących prawdopodobieństw aposteriori i nie ma potrzeby oddzielnej estymacji  $\pi_i$  i  $p(x|i)$ .

## Nota bene

Jeśli  $p(\mathbf{x}|i)$ : gęstość rozkładu  $N(\mathbf{m}_i, \Sigma)$   $i = 1, \dots, k$ .  
to

$$\log \frac{p(k|\mathbf{x})}{p(g|\mathbf{x})} = \frac{1}{2}(\mathbf{x} - \mathbf{m}_g)' \Sigma^{-1} (\mathbf{x} - \mathbf{m}_g) - \frac{1}{2}(\mathbf{x} - \mathbf{m}_k)' \Sigma^{-1} (\mathbf{x} - \mathbf{m}_g) + \log \frac{\pi_k}{\pi_g} =$$
$$\frac{1}{2}(\mathbf{m}_k - \mathbf{m}_g)' \Sigma^{-1} \mathbf{x} - \frac{1}{2}(\mathbf{m}_k + \mathbf{m}_g)' \Sigma^{-1} (\mathbf{m}_k - \mathbf{m}_g) + \log \frac{\pi_k}{\pi_g}$$

ma postać  $\beta'_k \mathbf{x}$ .

Ta zależność była również wykorzystywana w metodzie LDA.

Czym zatem różnią się te dwie metody?

Sposobem estymacji parametrów.

W regresji logistycznej maksymalizujemy ( $g = 2$ )

$$L = \prod_{i=1}^n P(Y = 1|X = \mathbf{x}_i)^{y_i} (1 - P(Y = 1|X = \mathbf{x}_i))^{1-y_i}$$

To jest warunkowa funkcja wiarygodności  $p(y_1, \dots, y_n | \mathbf{X} = \mathbf{x})$  wykorzystująca jedynie warunkowy rozkład  $Y$  pod warunkiem  $\mathbf{X}$ . (brzegowy rozkład  $\mathbf{X}$  nie odgrywa tu roli, nic o nim nie zakładamy!)

W przypadku LDA gęstość  $p(\mathbf{X} = \mathbf{x}, Y = k)$  ma postać

$$p(\mathbf{x}, k) = \phi(\mathbf{x}, \boldsymbol{\mu}_k, \boldsymbol{\Sigma})\pi_k$$

Maksymalizacja pełnej funkcji wiarygodności o postaci

$$\tilde{L} = \prod_{i=1}^n p(\mathbf{x}_i, y_i)$$

prowadzi do rozpatrywanych poprzednio estymatorów

$$\hat{\mu}_i = \bar{x}_i$$

$$\hat{\Sigma} = \frac{1}{n - g} \sum_{k=1}^g (n_k - 1) S_k$$

$$\hat{\pi}_i = \frac{n_i}{n}$$

Można spodziewać się, że regresja logistyczna nie jest tak czuła na duże odstępstwa od normalności i równość macierzy kowariancji jak LDA.

# Odporność na złą specyfikację modelu

Regresja logistyczna dobrze estymuje kierunek prawdziwego wektora  $\beta$ , nawet gdy model jest źle wyspecyfikowany dla predyktorów normalnych.

Twierdzenie (Ruud(83), Brillinger (1983)) Niech dane będą generowane przez model binarny  $P(Y = 1|X = x) = q(\alpha_0 + \beta_0'x)$ , gdzie  $q$  jest niekoniecznie funkcją logistyczną. Jeśli predyktory mają rozkład normalny i  $\hat{\beta}_{ML}$  estymator w dopasowanym modelu logistycznym, to

$$\hat{\beta}_{ML} \approx \eta\beta_0$$

( $\hat{\beta}_{ML}$  w przybliżeniu współliniowy z  $\beta_0$ ). Podobny wynik prawdziwy dla dopasowania modelu liniowego do modelu binarnego !

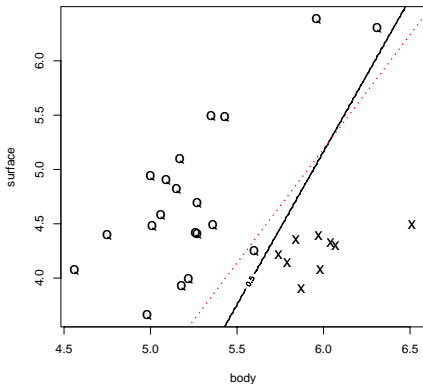
## Dane earthquake

Dopasujemy model logistyczny  $\text{popn} \sim \text{body} + \text{surface}$ . Tworzymy nową ramkę danych z zero-jedynkową zmienną  $y$  zamiast  $\text{popn}$ . Dla dopasowania modelu logistycznego  $y \sim \text{body} + \text{surface}$  wykorzystywana funkcja `glm`.

Opcja `family=binomial` specyfikuje model logistyczny.

```
earthquake = read.table("earthquake.txt", header=TRUE)
equake = data.frame(y=ifelse(earthquake$popn=="equake", 0, 1),
body=earthquake$body, surface=earthquake$surface)
g2 = glm(y~ body + surface, data=equake, family=binomial)
```

Wykres rozproszenia danych earthquake z obszarami klasyfikacji wyznaczonymi przy użyciu klasyfikacji logistycznej (linia ciągła) i LDA (linia przerywana).



Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-1151.09	1000874.08	-0.001	0.999
body	276.21	190711.00	0.001	0.999
surface	-98.01	118520.53	-0.001	0.999

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 3.5924e+01 on 28 degrees of freedom  
Residual deviance: 3.7043e-09 on 26 degrees of freedom  
AIC: 6

Wartość odchylenia resztowego (residual deviance) jest bardzo mała i wskazuje na bardzo dobre dopasowanie, gdy jednocześnie wyniki testu  $t$  mówią o nieistotności obu zmiennych. Taka paradoksalna sytuacja występuje często przy liniowej separowalności klas, gdy estymatory współczynników w modelu regresji logistycznej i ich błędy standardowe zachowują się niestabilnie ( $p(1|\mathbf{x}_i) \approx 1, 0, i = 1, \dots, n \Rightarrow \|\hat{\beta}\|$ -duża).

## Tabela i procent poprawnych reklasyfikacji.

```
Ypred =ifelse(g2$fitted.values < 0.5, 0, 1)

# klasyfikacja do klasy 1 dla prawd. aposteriori klasy 1 < 0,5.

print(kl =table(equake$y, Ypred))
print(procent= sum(diag(kl)) / sum(kl))
```

```
      Ypred
      0  1
0  20  0
1   0  9
```

```
[1] 1
```

Działanie klasyfikatora logistycznego różni się od klasyfikatora LDA: pierwszy z nich klasyfikuje bezbłędnie wszystkie elementy próby uczącej (sytuacja liniowo separowalnych klas). *Nie* należy wyciągać stąd wniosku, że klasyfikator logistyczny będzie działał lepiej dla nowych obserwacji.

Dane **urine**, wybór zmiennych w klasyfikacji.

Zmienna `presence` jest zmienną grupującą, pozostałe atrybuty: wartości pomiarów fizyko-chemicznych moczu. Model logistyczny `presence ~ sg+ph+mosm+mmho+urea+calcium`

```
urine.glm=glm(presence ~ ., family = binomial, data = urine)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-355.33771	222.76696	-1.595	0.11069
sg	355.94379	222.11004	1.603	0.10903
ph	-0.49570	0.56976	-0.870	0.38429
mosm	0.01681	0.01782	0.944	0.34536
mmho	-0.43282	0.25123	-1.723	0.08493
urea	-0.03201	0.01612	-1.986	0.04703
calcium	0.78369	0.24216	3.236	0.00121

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 105.17 on 76 degrees of freedom  
Residual deviance: 57.56 on 70 degrees of freedom  
AIC: 71.56

$$Dev_{null,\omega} = Dev_{null} - Dev_{resid}.$$

Duża różnica odchylenia zerowego (null deviance) i resztowego (residual deviance) wskazuje na występowanie istotnych zmiennych w modelu, odpowiednia  $p$ -wartość, uzyskana na podstawie rozkładu chi kwadrat z 6 stopniami swobody wynosi  $pchisq(105.17 - 57.56, 7-1, lower=F)$ .  
i jest mniejsza od 0.001. Tabelka reklasyfikacji i procent poprawnej reklasyfikacji

	k1	
	0	1
no	40	4
yes	8	25

[1] 0.8441558

Dokonajmy redukcji zmiennych w modelu metodą eliminacji wstecz, sprawdźmy, czy mniejszy model można uznać za adekwatny i jak wygląda reklasyfikacja.

```
urine.glm = glm(presence ~ ., data=urine, family=binomial)
urine.step = step(urine.glm, direction="backward")
print(urine.step)
```

```
. . . . .
```

```
Call:  glm(formula = presence ~ sg + mmho + urea + calcium, family = binomial,
          data = urine)
```

```
Coefficients:
```

(Intercept)	sg	mmho	urea	calcium
-500.01090	497.12038	-0.20547	-0.01783	0.72232

```
Degrees of Freedom: 76 Total (i.e. Null); 72 Residual
```

```
Null Deviance: 105.2
```

```
Residual Deviance: 59.07 AIC: 69.07
```

Otrzymany podzbiór zmiennych objaśniających: calcium, mmho, sg, urea uzyskuje się również stosując metodę dołączania.

Przetestujmy teraz, czy model mniejszy jest adekwatny, przy użyciu statystyki równej różnicy odchyłeń, która przy hipotezie  $H_0$  (model mniejszy jest adekwatny) ma dla dużych licznosci w przybliżeniu rozkład  $\chi^2$  z  $7 - 5 = 2$  stopniami swobody

```
print(anova(u2.glm, u.glm, test="Chi"))
```

Analysis of Deviance Table

Model 1: presence ~ sg + mmho + urea + calcium

Model 2: presence ~ sg + ph + mosm + mmho + urea + calcium

	Resid. Df	Resid. Dev	Df	Deviance	P(> Chi )
1	72	59.071			
2	70	57.560	2	1.511	0.470

Porównując model mniejszy i większy nie odrzucamy hipotezy, że model mniejszy jest adekwatny. Dopasowujemy mniejszy model i przeprowadzamy reklasyfikację.

```
u2.glm = glm(presence ~ sg + mmho + urea + calcium, data=urine,  
family=binomial)
```

```
      k12  
      0  1  
no    40  4  
yes   8 25
```

```
[1] 0.8441558
```

Otrzymaliśmy dokładnie takie same wyniki reklasyfikacji.

## Podjęcie teoriodecyzyjne

Zamieńmy indeksy klas na  $Y = \pm 1$ .

Wtedy:

dla  $Y = 1$

$$-\log p(x, y) = -y \log\left(\frac{\exp(\beta'x)}{1 + \exp(\beta'x)}\right) = \log(1 + \exp(-y\beta'x))$$

dla  $Y = -1$

$$-\log p(x, y) = -(-y) \log\left(\frac{1}{1 + \exp(\beta'x)}\right) = \log(1 + \exp(-y\beta'x))$$

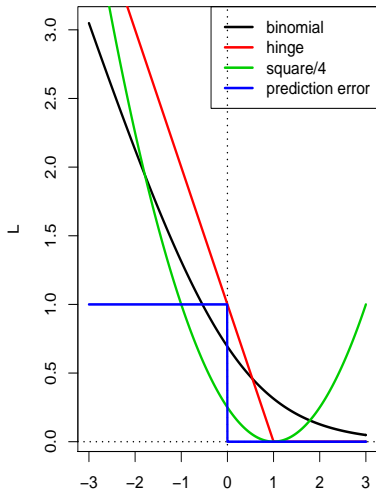
funkcja straty

$$l(y, f(x)) = \log(1 + \exp(-yf(x)))$$

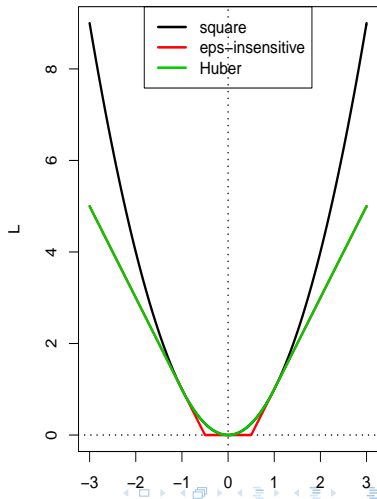
Najczęstsze funkcje straty: -logarytm wiarygodności  $p(x, y)$  dla dopasowywanego modelu.

$$\hat{\beta}_{ML} = \operatorname{argmin}_{\beta} \sum_{i=1}^n l(y_i, \beta'x_i)$$

### Classification loss functions



### Regression loss functions



## Model proporcjonalnych szans

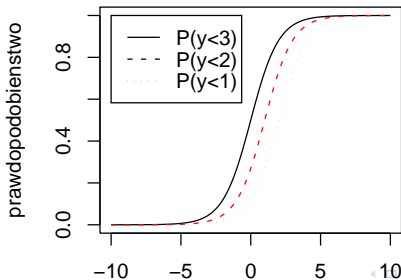
Przypuśćmy, że zmienna  $Y$  jest zmienną nominalną o  $g$  uporządkowanych kategoriach (np. kategorie wiekowe, kategorie klienta: spłaty terminowe, spłaty z opóźnieniem, brak spłat). Informacja o uporządkowaniu klas powinna być wykorzystana w modelu. Oznaczmy kategorie jako  $1, 2, \dots, g$ . W modelu proporcjonalnych szans dla  $j = 1, 2, \dots, g - 1$

$$\log \frac{Pr(y \leq j | \mathbf{x})}{1 - Pr(y \leq j | \mathbf{x})} = \alpha_j - \beta' \mathbf{x}, \quad (*)$$

gdzie  $\mathbf{x} = (x_1, \dots, x_p)'$  jest wektorem predyktorów. Funkcja logitowa  $\log(p/(1-p)) \uparrow$  dla  $p \uparrow$  i  $Pr(y \leq j | \mathbf{x}) \uparrow$  gdy  $j \uparrow \Rightarrow \alpha_1 \leq \alpha_2 \leq \dots \leq \alpha_{g-1}$ . Dla ustalonego  $j$  model (\*) jest modelem logistycznej regresji dla odpowiedzi binarnej 1 gdy  $\{y \leq j\}$ , i 0 gdy  $\{y > j\}$ . Dla  $g = 2$  otrzymujemy model regresji logistycznej.

Ważne ze zmianą  $j$  in  $(*)$  wyraz wolny  $\alpha_j$  się zmienia, podczas gdy wektor  $\beta$  pozostaje taki sam. Dla  $\gamma_j(\mathbf{x}) = Pr(y \leq j | \mathbf{x})$  założenie modelowe oznacza, że funkcja  $\gamma_j(\cdot)$  jest przesunięciem funkcji  $\gamma_k(\cdot)$ . Mianowicie, np. dla jednowymiarowego predyktora mamy

$$\gamma_k(x) = \frac{\exp(\alpha_j - \beta(x - (\alpha_k - \alpha_j)/\beta))}{1 + \exp(\alpha_j - \beta(x - (\alpha_k - \alpha_j)/\beta))} = \gamma_j(x - (\alpha_k - \alpha_j)/\beta).$$



Nazwa 'model proporcjonalnych szans' ( proportional odds) związana z faktem, że założenie modelowe implikuje:

$$\frac{\gamma_i(\mathbf{x}_1)/(1 - \gamma_i(\mathbf{x}_1))}{\gamma_i(\mathbf{x}_2)/(1 - \gamma_i(\mathbf{x}_2))} = \exp(-\beta'(\mathbf{x}_1 - \mathbf{x}_2)).$$

Tak więc powyższy iloraz szans nie zależy od  $i$ . Konwencja znaków  $\beta$ : dla  $x_1 < x_2$  przy  $\beta > 0$  chcemy, aby powyższy stosunek był  $> 1$  ( Uwaga: procedura GENMOD (SAS) używa  $\beta$  zamiast  $-\beta$  w (\*).

Parametry modelu estymowane przy użyciu metody największej wiarygodności.

Procedury: polr w R i Genmod w SAS.

Inne modele: model proporcjonalnych hazardów

$$\log(-\log(1 - \gamma_j(\mathbf{x}))) = \alpha_j + \beta' \mathbf{x}$$

Założenie implikuje, że  $P(Y > j|\mathbf{x}_1) = P(Y > j|\mathbf{x}_2)\exp(\beta'(\mathbf{x}_1 - \mathbf{x}_2))$ .

Inne metody dyskryminacji liniowej:

- perceptron Rosenblatta (liniowe sieci neuronowe);
- metoda oparta na regresji wielowymiarowej.

Klasyfikacja oparta o regresję liniową:

Odpowiedź: modyfikowana etykieta. Etykieta klasy kodowana jest jako wektor  $g$ -wymiarowy

$$y = (y^{(1)}, \dots, y^{(g)})'$$

dla klasy  $k$ ,  $y = (0, 0, \dots, 0, 1, 0, \dots, 0)'$  (1 na  $k$ -tym miejscu)

# Regresja wielowymiarowa

$\mathbf{X}$  – macierz eksperymentu  $n \times (p + 1)$

$\mathbf{Y}$  – macierz (a nie wektor) odpowiedzi

$$\begin{pmatrix} y_1^{(1)} & \cdots & y_1^{(g)} \\ \vdots & & \vdots \\ y_n^{(1)} & \cdots & y_n^{(g)} \end{pmatrix}$$

Szukamy macierzy  $\hat{\mathbf{B}}_{(p+1) \times g}$  minimalizującej

$$CRIT(\mathbf{B}) = \sum_{i=1}^n \|y_i - [1, \mathbf{x}'_i]\mathbf{B}\|^2$$

# Regresja wielowymiarowa

Problem równoważny rozwiązaniu  $g$  problemów regresji wielokrotnej.

$$\begin{aligned} CRIT(\mathbf{B}) &= \sum_{i=1}^n \|y_i - [1, \mathbf{x}'_i]\mathbf{B}\|^2 = \\ &= \sum_{i=1}^n \sum_{k=1}^g (y_i^{(k)} - [1, \mathbf{x}'_i]\mathbf{b}_k)^2 = \\ &= \sum_{k=1}^g \sum_{i=1}^n (y_i^{(k)} - [1, \mathbf{x}'_i]\mathbf{b}_k)^2 = \end{aligned}$$

$\mathbf{b}_1, \dots, \mathbf{b}_g$  - kolumny  $\mathbf{B}$ .

Minimalizacja  $CRIT(\mathbf{B})$  minimalizacja wszystkich wewnętrznych sum oddzielnie.  $g$  problemów regresji jednowymiarowej !

Macierz  $\hat{\mathbf{B}}$  składa się z kolumn parametrów dla kolejnych problemów regresji ( $k$ -ty problem: odpowiedź 1 dla elementu  $k$ -tej klasy) dla pozostałych klas: 'jeden przeciw wszystkim').

Okazuje się, że prognoza  $\hat{y}(\mathbf{x}) = [1, \mathbf{x}']\mathbf{B}$  ma własność

$$\sum_{i=1}^n \hat{y}^{(k)}(\mathbf{x}) = 1$$

Reguła klasyfikacyjna

$$\delta(\mathbf{x}) = \operatorname{argmax}_{k=1,2,\dots,g} \hat{y}^{(k)}(\mathbf{x})$$

Komentarz:  $\delta(\cdot)$  dopuszcza uogólnienie nieliniowe  
– dyskryminacja giętka (flexible discrimination)

# Maskowanie

## Kwestia skal pomiarowych atrybutów

Dotąd milcząco zakładaliśmy, że atrybuty przyjmują wartości rzeczywiste. Nie ma problemu dla zmiennych ilościowych dyskretnych ze stosowaniem LDA, dyskryminacji logistycznej, empirycznej metody bayesowskiej.

Wartości nominalne

$x$  – przyjmuje  $r$  wartości

$i$ –ta wartość  $\rightarrow (0, 0, \dots, 0, 1, 0, \dots, 0)'$  (1 na  $i$ –tym miejscu)

musimy mieć dane zawierające obserwacje dla każdego układu atrybutów,

aby metoda była stabilna

wartości nominalne na skali porządkowej: metoda *ad hoc*

$i$ -ta wartość  $\rightarrow (i - 1)/n$

## Naiwna metoda bayesowska

Dal atrybutów nominalnych o odpowiednio  $k_1, \dots, k_p$  wartościach potrzebujemy  $n \gg g \times k_1 \times \dots \times k_p$  wartości aby wszystkie komórki  $X^{(1)} = x^{(1)}, \dots, X^{(p)} = x^{(p)}$  we wszystkich klasach zostały wypełnione. Metoda postępowania dla atrybutów nominalnych unikająca problemu przekleństwa nieskończoności (*curse of dimensionality*) oparta na **naiwnej metodzie bayesowskiej** (zakładającej niezależność atrybutów w klasach)

$$\frac{p(2|\mathbf{x})}{p(1|\mathbf{x})} = \frac{\pi_2 p(\mathbf{x}|2)}{\pi_1 p(\mathbf{x}|1)} = \frac{\pi_2}{\pi_1} \prod_{i=1}^p \frac{p(x^{(i)}|2)}{p(x^{(i)}|1)}$$

$$\mathbf{x} = (x^{(1)}, \dots, x^{(p)})'$$

$$\log \frac{p(2|\mathbf{x})}{p(1|\mathbf{x})} = \log \frac{\pi_2}{\pi_1} + \sum_{i=1}^p \log \frac{p(x^{(i)}|2)}{p(x^{(i)}|1)}$$

atrybut  $\mathbf{x}^{(i)}$  – poziomy  $l = 1, \dots, m_i$

$$\hat{P}(\mathbf{x}^{(i)} = l|k) = \frac{n_{ik}(l)}{n_k}$$

$n_{ik}(l)$  – # elementów klasy  $k$ , dla których  $i$ -ty atrybut jest równy  $l$

$$\frac{p(\mathbf{x}^{(i)} = l|2)}{p(\mathbf{x}^{(i)} = l|1)} \text{ estymujemy przez } \frac{n_{i2}(l)}{n_{i1}(l)} \cdot \frac{n_1}{n_2}$$

## Uwaga

- Naiwna metoda bayesowska działa często dobrze nawet w przypadku, gdy atrybuty są zależne !
- (ii) Poprawka Laplace'a

$$\hat{P}(\mathbf{x}^{(i)} = l|k) = \frac{n_{ik}(l) + \lambda}{\sum_w (n_{ik}(w) + \lambda)}$$

$\lambda = 0.5, 1.$