

Zaawansowane uczenie maszynowe:  
*wykład 14*

Paweł Cichosz

1 Selekcja atrybutów

2 Transformacja atrybutów

# Selekcja atrybutów

**Motywacja:** ograniczenie ryzyka nadmiernego dopasowania, uproszczenie modeli i zwiększenie możliwości ich interpretacji.

**Rodzaje metod:**

**filtry (*attribute selection filters*):** selekcja na podstawie pewnej miary predykcyjnej przydatności pojedynczych atrybutów lub (rzadziej) podzbiorów atrybutów, niezwiązanej z konkretną reprezentacją i algorytmem tworzenia modelu,

**opakowania (*attribute selection wrappers*):** selekcja przez przeszukiwanie przestrzeni podzbiorów atrybutów, ocenianych na podstawie jakości modeli tworzonych z ich wykorzystaniem przez ustalony docelowy algorytm uczenia się.

## Selekcja atrybutów przez filtrowanie

**Korzyści:** łatwość użycia, niezależność selekcji od docelowego algorytmu uczenia się.

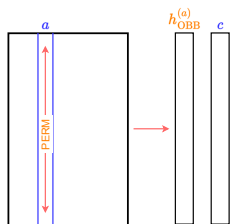
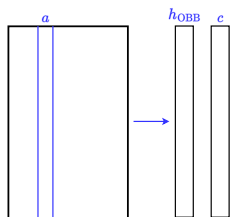
**Proste filtry statystyczne:** ocena predykcyjnej przydatności poszczególnych atrybutów na podstawie statystycznych miar zależności z atrybutem docelowym.

**Algorytm CFS (*correlation-based feature selection*):** ocena predykcyjnej przydatności podzbioru atrybutów  $\mathbf{A}$  na podstawie ich zależności z atrybutem docelowym i zależności wzajemnych:

$$\frac{\sum_{a \in \mathbf{A}} \kappa(a, b)}{\sqrt{|\mathbf{A}| + \sum_{a_1, a_2 \in \mathbf{A}, a_1 \neq a_2} \kappa(a_1, a_2)}}$$

gdzie  $b$  – atrybut docelowy,  $\kappa$  – miara zależności (np. współczynnik korelacji, informacja wzajemna, symetryczna niepewność itp.).

# Selekcja atrybutów przez filtrowanie



**Las losowy:** wpływ zaburzenia atrybutu przez losową permutację wartości na wzrost błędu OOB w lesie losowym (*ważność zmiennych*).

**Algorytm Boruta:** iteracyjny proces selekcji atrybutów wykorzystujący las losowy.

- 1 Tworzone randomizowane kopie każdego atrybutu (przez losową permutację).
- 2 Wyznaczana predykcyjna użyteczność wszystkich atrybutów za pomocą lasu losowego.
- 3 Predykcyjna użyteczność każdego atrybutu porównywana z maksymalną predykcyjną użytecznością jego randomizowanych kopii:

**atrybuty ważne:** istotnie częściej lepsze niż gorsze od najlepszej randomizowanej kopii,

**atrybuty nieważne:** istotnie częściej gorsze niż lepsze od najlepszej randomizowanej kopii.

# Selekcja atrybutów przez opakowywanie

**Korzyści:** dostosowanie zestawu atrybutów do specyfiki docelowego algorytmu uczenia się.

**Podzbiory atrybutów:** generowane przez zastosowanie pewnej strategii przeszukiwania przestrzeni możliwych podzbiorów.

**Ocena podzbiorów atrybutów:**

- ograniczenie danych do atrybutów z podzbioru,
- zastosowanie procedury oceny jakości modeli opartej na próbkowaniu (najczęściej  $n \times k$ -krotna walidacja krzyżowa).

# Przeszukiwanie przestrzeni podzbiorów

## Proste strategie lokalnego przeszukiwania:

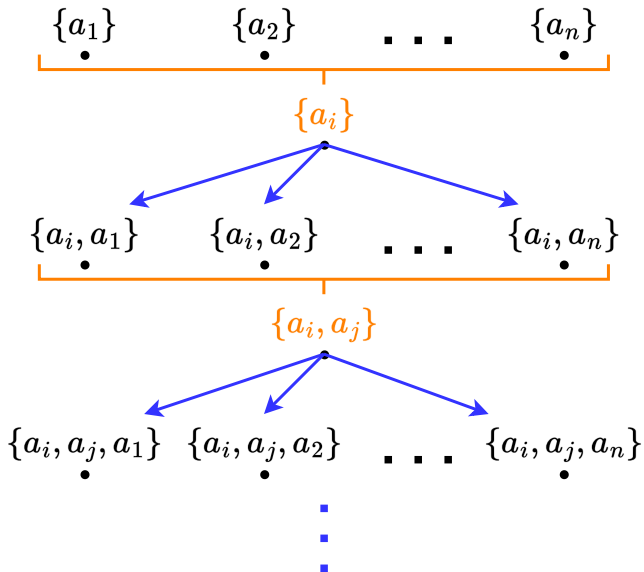
**w przód:** dodawanie kolejnych atrybutów zaczynając od podzbiorów jednoelementowych, zatrzymanie gdy kilka kolejnych kroków nie przynosi poprawy jakości.

**wstecz:** usuwanie kolejnych atrybutów zaczynając od pełnego zbioru, zatrzymanie gdy kilka kolejnych kroków nie przynosi poprawy jakości.

**mieszane:** przeplatanie kroków w przód i wstecz.

**Metaheurystyczne algorytmy przeszukiwania:** np. wspinaczkowe, symulowane wyżarzanie, ewolucyjne.

## Przeszukiwanie przestrzeni podzbiorów





# Rekurencyjna eliminacja atrybutów

**Podjęcie hybrydowe:** o wyborze podzbioru decyduje ocena jakości modelu (jak przy opakowywaniu), lecz proces przeszukiwania ukierunkowany przez miarę przydatności atrybutów (jak przy filtrowaniu).

- 1 zainicjalizuj  $\mathbf{A}$  jako zbiór wszystkich atrybutów;
- 2 **powtarzaj:**
  - 1 utwórz model  $h$  z użyciem atrybutów z  $\mathbf{A}$ ;
  - 2 wyznacz wartość miary przydatności każdego  $a \in \mathbf{A}$  dla modelu  $h$ ;
  - 3 oceń jakość modelu;
  - 4 usuń z  $\mathbf{A}$  ustaloną liczbę  $s \geq 1$  najmniej przydatnych atrybutów;

**jak długo** nie są spełnione kryteria stopu.

# Selekcja atrybutów jako część procesu modelowania

- Dane używane do selekcji atrybutów wchodzi w skład zbioru trenującego (w szerszym sensie) nawet jeśli nie są bezpośrednio wykorzystywane do tworzenia modelu po selekcji atrybutów.
- W celu nieobciążonej oceny jakości modelu konieczny podzbiór niewykorzystywany ani do jego tworzenia, ani do selekcji atrybutów.

- 1 Selekcja atrybutów

- 2 Transformacja atrybutów

# Transformacje skalujące

**Motywacja:** transformacje zalecane w przypadku algorytmów wykorzystujących miary niepodobieństwa i odległości (w tym SVM), stosujących regularyzację lub zakładających normalność atrybutów, mogą przyspieszać algorytmy gradientowe.

**Centrowanie:** sprowadzanie do rozkładu o średniej 0:

$$a'(x) = a(x) - m_T(a)$$

**Standardyzacja:** sprowadzanie do rozkładu o średniej 0 i odchyleniu standardowym 1:

$$a'(x) = \frac{a(x) - m_T(a)}{s_T(a)}$$

# Transformacje skalujące

„Normalizacja” (1): skalowanie do przedziału  $[0, 1]$ :

$$a'(x) = \frac{a(x) - \min_{x' \in T} a(x)}{\max_{x' \in T} a(x) - \min_{x' \in T} a(x)}$$

„Normalizacja” (2): sprowadzanie do jednostkowej normy wektora wartości atrybutów:

$$a'(x) = \frac{a(x)}{\|\mathbf{a}(x)\|}$$

**Normalizacja:** sprowadzanie do rozkładu (zbliżonego do) normalnego – np. transformacja Boxa-Coxa.

# Transformacje skalujące

Transformacja Boxa-Coxa:

$$a'_i(x) = \begin{cases} \frac{a_i^\lambda(x)-1}{\lambda} & \text{jeśli } \lambda \neq 0 \\ \ln a_i(x) & \text{jeśli } \lambda = 0 \end{cases}$$

pod warunkiem, że  $a_i(x) > 0$  dla wszystkich  $x$ , w przeciwnym przypadku można to wymusić przesunięciem:

$$a'_i(x) = \begin{cases} \frac{(a_i(x)+b)^\lambda-1}{\lambda} & \text{jeśli } \lambda \neq 0 \\ \ln(a_i(x) + b) & \text{jeśli } \lambda = 0 \end{cases}$$

**Dobór  $\lambda$ :** np. maksymalizacja wiarygodności (prawdopodobieństwa przekształconych wartości atrybutu przy założeniu rozkładu normalnego).

# Transformacje redukujące wymiarowość

**Motywacja:** ograniczenie ryzyka nadmiernego dopasowania.

**Zasada działania:** liniowe transformacje algebraiczne macierzy wartości atrybutów – tworzą mniejszą liczbę nowych atrybutów liniowo zależnych od pierwotnych atrybutów zapewniających możliwość ich przybliżonej rekonstrukcji.

**Analiza składowych głównych (*principal component analysis*, PCA):** identyfikacja nowych atrybutów jako składowych głównych:

- nieskorelowane kombinacje liniowe atrybutów pierwotnych,
- każda kolejna maksymalizuje wariancję przy zachowaniu ortogonalności do poprzednich,
- wyznaczone na podstawie wektorów i wartości własnych macierzy kowariancji (lub korelacji) atrybutów (po wycentrowaniu).

## Transformacje redukujące wymiarowość

**Realizacja PCA:** dla macierzy wartości atrybutów na zbiorze trenującym  $\mathbf{a}(T)$ :

**centrowanie:** wyznaczenie macierzy wartości wycentrowanych atrybutów  $\mathbf{a}'(T)$  przez odjęcie średnich:

$$a'(x) = a(x) - m_T(a)$$

**kowariancja/korelacja:** wyznaczenie macierzy kowariancji lub korelacji (liniowej)  $R$ :

$$R[i, j] = \text{cov}_T(a'_i, a'_j)$$

lub

$$R[i, j] = \text{cor}_T(a'_i, a'_j)$$

(użycie korelacji zamiast kowariancji daje efekt równoważny sprowadzeniu atrybutów do jednakowej wariancji).



# Transformacje redukujące wymiarowość

## Realizacja PCA:

**wektory własne:** wyznaczenie wektorów i wartości własnych macierzy  $R$ ,

**składowe główne:** wybór  $k$  wektorów własnych o największych wartościach własnych  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k$ ,

**macierz projekcji:** macierz, której kolumnami są wybrane wektory własne:

$$V = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k]$$

**projekcja:** rzutowanie danych na wektory własne:

$$\mathbf{a}''(x) = V^T \mathbf{a}'(x)$$

# Transformacje redukujące wymiarowość

Rozkład według wartości osobliwej (*singular value decomposition*, SVD):  
dogodna numerycznie alternatywna metoda realizacji PCA przez faktoryzację macierzy wartości atrybutów.

Pokrewne metody (poza zakresem wykładu):  
wielowymiarowa analiza odpowiedniości (*multiple correspondence analysis*, MCA): odpowiednik PCA dla atrybutów dyskretnych,  
analiza ukrytej semantyki (*latent semantic analysis*, LSA):  
ukryta alokacja Dirichleta (*latent Dirichlet allocation*, LDA):  
redukcja wymiarowości reprezentacji dokumentów tekstowych (macierzy dokument-term) przez identyfikację „ukrytych atrybutów” („tematów”).

## Dyskretyzacja atrybutów ciągłych

Utworzenie atrybutu dyskretnego zastępującego oryginalny atrybut ciągły, o wartościach dyskretnych odpowiadających przedziałom wartości ciągłych:

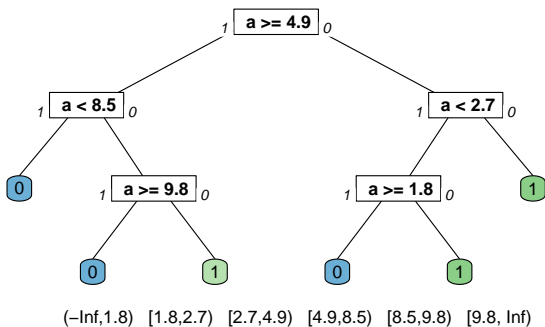
**równa szerokość:** podział na przedziały o równej szerokości,

**równa częstość:** podział na przedziały o równej częstości (liczbie przykładów),

**nadzorowana:** podział na przedziały z uwzględnieniem rozkładu klas.

# Dyskretyzacja atrybutów ciągłych

**Dyskretyzacja zstępująca:** początkowo jeden przedział, następnie iteracyjnie wybierane punkty podziału na mniejsze przedziały maksymalnie „czyste klasowo”, np. na podstawie minimalizacji entropii (równoważne tworzeniu drzewa decyzyjnego z użyciem jednego atrybutu ciągłego).



# Dyskretyzacja atrybutów ciągłych

**Dyskretyzacja wstępująca:** początkowo przedziały jednoelementowe, następnie iteracyjnie łączone pary przedziałów o maksymalnie podobnym rozkładzie klas, np. na podstawie maksymalizacji statystyki  $\chi^2$ .

**Kryteria stopu:** liczba przedziałów, nieczystość/zgodność rozkładu klas, zasada minimalnej długości kodu (por. wykład 11).

# Transformacje atrybutów dyskretnych

**Agregacja:** utworzenie atrybutu dyskretnego o mniejszej liczbie wartości zastępującego oryginalny atrybut dyskretny o większej liczbie wartości przez połączenie pewnych podzbiorów pierwotnych wartości.

**Binarne kodowanie:** atrybut dyskretny  $k$ -wartościowy  $a : X \rightarrow \{v_1, v_2, \dots, v_k\}$  zastępowany przez  $k$  atrybutów binarnych  $a'_1, a'_2, \dots, a'_k$  (tzw. *one-hot encoding*) lub  $k - 1$  atrybutów binarnych  $a'_1, a'_2, \dots, a'_{k-1}$  (tzw. *dummy variables*):

$$a'_i(x) = \begin{cases} 1 & \text{jeśli } a(x) = v_i \\ 0 & \text{w przeciwnym przypadku} \end{cases}$$

przy czym:

- wartości  $v_i$  dla  $i = 1, \dots, k - 1$  są reprezentowane przez  $a'_i(x) = 1$  i  $a'_j(x) = 0$  dla  $j \neq i$ ,
- wartość  $v_k$  jest reprezentowana przez  $a'_k(x) = 1$  i  $a'_j(x) = 0$  dla  $j \neq k$  w wariancie *one-hot encoding* oraz przez  $a'_j(x) = 0$  dla wszystkich  $j = 1, \dots, k - 1$  w wariancie *dummy variables*.

# Transformacja atrybutów jako część procesu modelowania

- Parametry transformacji ustalone na danych trenujących przed tworzeniem modelu stosowane do nowych danych przed użyciem modelu do predykcji.
- Pełna reprezentacja modelu obejmuje parametry transformacji stosowanych przed zastosowaniem właściwego modelu do predykcji.