Diskretizace dat

- Metody předzpracování dat .. některé metody dokáží pracovat jen s diskrétními atributy (např. ID3)
- Též možno považovat za způsob, jak se vypořádat se šumem (binning)

- unsupervised – diskretizují atributy, aniž by počítali informaci na olabelování.
- supervised – diskretizují atributy používajíce závíslosti mezi známými labely a hodnotami atributů.

Unsupervised algoritmy diskretizace

Jsou nejjednodušší na implementaci. Potřebují jen zadat

- počet intervalů,
- (nebo) počet datapointů, který by měl být v každém intetvalu.

Následující heruristiky jsou použity k výběru intervalů:

- Počet intervalů pro každý atribut by neměl být menší, než počet tříd (pokud je znám).
- Počet intervalů n_{F_i} pro každý atribut F_i je

$$n_{F_i} = \frac{M}{3 \cdot C},$$

kde M je počet trénovacích datapointů, C je počet známých kategorií.

Equal-Width Discretization – najde minimální a maximální hodnoty pro každý atribut F_i, pak rozdělí celý interval na n_{F_i} intervalů stejné šířky.

Equal-Frequency Discretization – najde minimální a maximální hodnoty pro každý atribut F_i, hodnoty atributu F_i seřadí (eklesající usp.), a rozdělí na n_i intervalů o stejném počtu prvků.

Unsupervised algoritmy diskretizace

Algoritmy založené na teorii informace
Mnoho diskretizačních algoritmů má původ v teorii informace.
M – počet datapointů (olabelovaných)
S – počet tříd
F – spojitý atribut
Říkáme, že existuje diskretizační schéma D na F do n diskrétních intervalů, ohraničených páry čísel

$$D : \{[d_0, d_1], (d_1, d_2], \ldots, (d_{n-1}, d_n]\},$$

kde

- d_0 je minimální hodnota atributu F,
- d_n je maximální hodnota atributu F,
- $d_0 < d_1 < \cdots < d_n$

Tyto hodnoty určují množinu hranic pro diskretizaci D:

$$\{d_0, d_1, \ldots, d_{n-1}, d_n\}.$$

Vytvoříme $S \times n$ matici $\mathbf{Q}$ frekvencí pro klasifikaci a diskretizaci atributu – tzv matice kvant – *quanta matrix*.
q_{ir} je celkový počet hodnot atributu F náležející i-té třídě, které jsou uvnitř intervalu $(d_{r-1}, d_r]$.

Dále

- M_{i+} je počet hodnot, které patří k třídě i,
- M_{+r} je počet hodnot, které padnou do intervalu $(d_{r-1}, d_r]$.

Třída	Interval					celkem v třídě
	$[d_0, d_1]$	$\cdots$	$(d_r - 1, d_r]$	$\cdots$	$(d_n - 1, d_n]$	
C_1	q_{11}	$\cdots$	q_{1r}	$\cdots$	q_{1n}	M_{1+}
$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$
C_i	q_{i1}	$\cdots$	q_{ir}	$\cdots$	q_{in}	M_{i+}
$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$
celkem	M_{+1}	$\cdots$	M_{+r}	$\cdots$	M_{+n}	M

Nyní vypočítáme několik statistik nad maticí kvant.

Odhad pravděpodobnosti, že hodnoty F padne do intervalu $D_r = (d_{r-1}, d_r]$ a náleží k třídě C_i:

$$p_{ir} = p(C_i, D_r|F) = \frac{q_{ir}}{M}$$

Odhad marginální pravděpodobnosti p_{i+}, že hodnota atributu F patří k třídě C_i, je

$$p_{i+} = p(C_i) = \frac{M_{i+}}{M}$$

Odhad marginální pravděpodobnosti p_{+r}, že hodnota atributu F padne do $(d_{r-1}, d_r]$.

$$p_{+r} = p(D_r|F) = \frac{M_{+r}}{M}$$

Vzájemná informace třídy a atributu

$$I(C, D|F) = \sum_{i=1}^{S} \sum_{r=1}^{n} p_{ir} \log_2 \frac{p_{ir}}{p_{i+} p_{+r}}$$

Informace

$$\mathrm{INFO}(C, D|F) = \sum_{i=1}^{S} \sum_{r=1}^{n} p_{ir} \log_2 \frac{p_{+r}}{p_{ir}}$$

Shanonova entropie

$$H(C, D|F) = \sum_{i=1}^{S} \sum_{r=1}^{n} p_{ir} \log_2 \frac{1}{p_{ir}}$$

Redundance závisloti mezi třídou a atributem (class-attribute interdependence redunancy; CAIR)

$$R(C, D|F) = \frac{I(C, D|F)}{H(C, D|F)}$$

Nejistota závisloti mezi třídou a atributem (class-attribute interdependence uncertainty)

$$R(C, D|F) = \frac{\mathrm{INFO}(C, D|F)}{H(C, D|F)}$$

Algoritmus CAIM (Class-Attribute Interdependency Maximization)
Začíná celým intervalem $[d_0, d_n]$.
rozděluje jeden existujících intervalů do dvou nových
používá při tom kritérium, které zajišťuje dosažení optimální závislosti mezi labely a atributy po rozdělení:

$$CAIM(C, D|F) = \frac{\sum_{r=1}^{n} \frac{\max_r^2}{M_{+r}}}{n},$$

kde n je počet intervalů a $\max_r = \max(q_{ir})$
kritérium CAIM je heuristické měření, které má následující vlastnosti:

- čím větší hodnota CAIM, tím vyšší závislost mezi labely a intervaly.
- CAIM nabírá hodnot z intervalu $[0, M]$, kde M je počet hodnot atributu F.
- CAIM generuje diskretizační schéma, kde každý interval má většinu jeho hodnot seskupených v jedné třídě.

Vstup: Dataset o M datapointech, S třídách, a n spojitých atributů F_i

Pro každý atribut F_i udělej:
Krok 1

2.1 najdi maximální (d_n) a minimální (d_0) hodnotu F_i

2.2 vytvoř množinu všech různých hodnot F_i v neklesajicím pořadí, inicializuj všechny možné hranice B.

2.3 nastav iniciální diskretizační schéma na $\{[d_0, d_n]\}$, nastav $\mathrm{GlobalCAIM} = 0$

Krok 2

2.1 inicializuj $k = 1$

2.2 zkoušej přidávat vnitřní hodnoty z B, které ještě nejsou v D, vypočítej odpovídající CAIM

2.3 vyber tu, která ma nejvyšší CAIM

2.4 pokud ($\mathrm{CAIM} > \mathrm{GlobalCAIM}$ nebo $k < S$), přidej tu hranici do D a nastav $\mathrm{GlobalCAIM} = \mathrm{CAIM}$; jinak skonči.

2.5 $k = k + 1$, pokračuj krokem 2.2.

Výstup: Diskretizační schéma D.

```
> data(iris)
> vec <- iris[,1]
> sort(vec[!duplicated(vec)])
 [1] 4.3 4.4 4.5 4.6 4.7 4.8 4.9 5.0 5.1 5.2 5.3 5.4 5.5
[14] 5.6 5.7 5.8 5.9 6.0 6.1 6.2 6.3 6.4 6.5 6.6 6.7 6.8
[27] 6.9 7.0 7.1 7.2 7.3 7.4 7.6 7.7 7.9
> make.boundaries(vec)
 [1] 4.35 4.45 4.55 4.65 4.75 4.85 4.95 5.05 5.15 5.25
[11] 5.35 5.45 5.55 5.65 5.75 5.85 5.95 6.05 6.15 6.25
[21] 6.35 6.45 6.55 6.65 6.75 6.85 6.95 7.05 7.15 7.25
[31] 7.35 7.50 7.65 7.80
> d <- caim.discretize(vec,fac)
> d
[1] -Inf 5.55 6.25  Inf
> quanta.matrix(vec, d, fac)
           [,1] [,2] [,3]
setosa       47    3    0
versicolor   11   25   14
virginica     1   12   37
```

$\{-\infty, 4.35, \infty\}$ 8.889262
$\{-\infty, 4.45, \infty\}$ ... 10.56164;
$\{-\infty, 4.55, \infty\}$ ... 11.12069;
$\{-\infty, 4.65, \infty\}$ ... 13.36525;
$\{-\infty, 4.75, \infty\}$ ... 14.49281;
$\{-\infty, 4.85, \infty\}$ ... 17.32836;
$\{-\infty, 4.95, \infty\}$ ... 18.46982;
$\{-\infty, 5.05, \infty\}$ ... 22.42373;
$\{-\infty, 5.15, \infty\}$ ... 26.81864;
$\{-\infty, 5.25, \infty\}$ ... 28.33333;
$\{-\infty, 5.35, \infty\}$ ... 28.93457;
$\{-\infty, 5.45, \infty\}$ ... 31.72115;
$\{-\infty, 5.55, \infty\}$ ... **31.91265**;
$\{-\infty, 5.65, \infty\}$ ... 30.54525;
$\{-\infty, 5.75, \infty\}$ ... 30.78936;
$\{-\infty, 5.85, \infty\}$ ... 29.45357;
$\{-\infty, 5.95, \infty\}$ ... 28.85875;

$\{-\infty, 6.05, \infty\}$ ... 27.82363;
$\{-\infty, 6.15, \infty\}$ ... 26.98517;
$\{-\infty, 6.25, \infty\}$ ... 26.04783;
$\{-\infty, 6.35, \infty\}$ ... 23.01455;
$\{-\infty, 6.45, \infty\}$ ... 20.52671;
$\{-\infty, 6.55, \infty\}$ ... 18.48333;
$\{-\infty, 6.65, \infty\}$ ... 18.88876;
$\{-\infty, 6.75, \infty\}$ ... 16.84038;
$\{-\infty, 6.85, \infty\}$ ... 16.01614;
$\{-\infty, 6.95, \infty\}$ ... 14.66255;
$\{-\infty, 7.05, \infty\}$ ... 15.05797;
$\{-\infty, 7.15, \infty\}$ ... 14.49281;
$\{-\infty, 7.25, \infty\}$ ... 12.80282;
$\{-\infty, 7.35, \infty\}$ ... 12.24126;
$\{-\infty, 7.5, \infty\}$ ... 11.68056;
$\{-\infty, 7.65, \infty\}$ ... 11.12069;
$\{-\infty, 7.8, \infty\}$ ... 8.889262;

```
> for (i in 1:4) {
+     vec <- iris[,i]
+     print(quanta.matrix(vec,
+     caim.discretize(vec,fac), fac))}

           [,1] [,2] [,3]
setosa       47    3    0
versicolor   11   25   14
virginica     1   12   37
           [,1] [,2] [,3]
setosa        2    6   42
versicolor   34    8    8
virginica    21   12   17
           [,1] [,2] [,3]
setosa       50    0    0
versicolor    0   44    6
virginica     0    1   49
           [,1] [,2] [,3]
setosa       50    0    0
versicolor    0   49    1
virginica     0    5   45
```

χ^2-**test** se používá v algoritmu CAIR/CADD, ale dá se také použít samotný pro účely diskretizace.
Každý *hraniční bod* (BP) rozděluje hodnoty atributů v rozsahu $[a, b]$ na dvě části

- $L_{\mathrm{BP}} = [a, \mathrm{BP}]$,
- $R_{\mathrm{BP}} = (\mathrm{BP}, b]$.

Použitím statistiky, můžeme měřit stupeň nezávislosti mezi rozkladem daným klasifikací a rokladem daným vybraným BP.

Pro tento účel používáme χ^2-test následovně:

$$\chi^2 = \sum_{r=1}^{2} \sum_{i=1}^{C} \frac{(q_{ir} - E_{ir})^2}{E_{ir}},$$

kde E_{ir} je očekávaná frekvence atributu F_{ir}:

$$E_{ir} = \frac{q_{+r} \cdot q_{i+}}{M}$$

Pokud $q_{+r} = 0$ nebo $q_{i+} = 0$, použije se místo toho 0.1.

Pokud jsou rozklady nezávislé, pak

$$P(q_{i+}) = P(q_{i+}|L_{\mathrm{BP}}) = P(q_{i+}|R_{\mathrm{BP}}),$$

pro všechny třídy.

To znamená, že $q_{ir} = E_{ir}$ pro všechna $r \in \{1, 2\}$ a $i \in \{1, \ldots, C\}$, a $\chi^2 = 0$.

Toto pozorování vede k následující heuristice:

Uchovávej BP *s odpovídajícími vysokými hodnotami* χ^2 *a maž ty s malými hodnotami*

Diskretizace podle maximální entropie

Nechť T je množina všech diskretizačních schémat s jejich odpovídajícími maticemi kvant.

Cílem je najít diskretizaci $t^* \in T$, t.ž.

$$H(t^*) \geq H(t) \quad \forall t \in T,$$

kde H je Shannonova entropie.

Nejdříve se navrhnou iniciální hranice. Následně se s nimi zkouší pohybovat a sleduje se změna v entropii.

Vstup: Trénovací dataset z M datapointů a C tříd.

Pro každý atribut proveď:
Iniciální výběr hranic intervalů.

- Vypočti počet inervalů $n = M/3 \cdot C$
- Nastav hranice tak, aby součty $M_{M_{i_+}}$ byly co nejrovnoměrnější – aby se maximalizovala marginální entropie.

Lokální zlepšení intervalových hodnot

- Úpravy hranic jsou dělány postupně dosazováním hodnot do dolní hranice a horní hranice každého intervalu.
- Přijmi novou hranici, pokud se celková entropie zvyšuje.
- Opakuj, dokud nenastane, že se nedosáhne žádného vylepšení.

Výstup: hranice intervalů pro každý atribut.

Algoritmus diskretizace CAIR (Class-Attribute Interdependence Redundancy Discretization)

Vstup: Trénovací dataset z M datapointů a C tříd.
(A) Inicializace intervalů

- Setriď do neklesajícího pořadí spojité atributy
- Vypočítej počet intervalů $M/3 \cdot C$
- Proveď ME diskretizaci na setřízených hodnotách $\rightarrow$ iniciální intervaly.
- Vytvoř matici kvant odpovídající těmto iniciálním intervalům

(B) Vnitřní úpravy

- Posupně zkoušej eliminovat každou vnitřní hranici a vypočítej odpovídající CAIR.
- Poté, co byly vyzkoušeny všechny eliminace, přijmi tu, která má nejvyšší hodnotu CAIR.
- Uprav hranice, a opakuj tento krok, dokud se CAIR neprestane zvyššovat.

(C) Eliminace intervalů
V tomto kroku se slijí statisticky nedůležité intervaly. K tomu se používá χ^2 test následovně:

- Proveď test:

$$R(C : F_j) \geq \frac{\chi^2}{2 \cdot L \cdot H(C : F_j)},$$

  kde χ^2 je χ^2 hodnota na určité úrovni signifikance určené uživatelem, L je počet hodnot ve dvou sousedních intervalech, H je entropie těchto intervalů.
- Pokud je tesi signifikantní na určené úrovni, proveď stejný test pro následující dvojici intervalů.
- Pokud ne, jeden z těch intervalů je redundantní, a proto jsou slity do jednoho odstraněním oddělující hranice.

Výstup: hranice intervalů pro každý atribut.

Ostatní algoritmy pro supervised diskretizaci

Diskretizace pomocí jedno-úrovňového rozhodovacího stromu

(proberem až u rozhodovacích stromů, YYY)

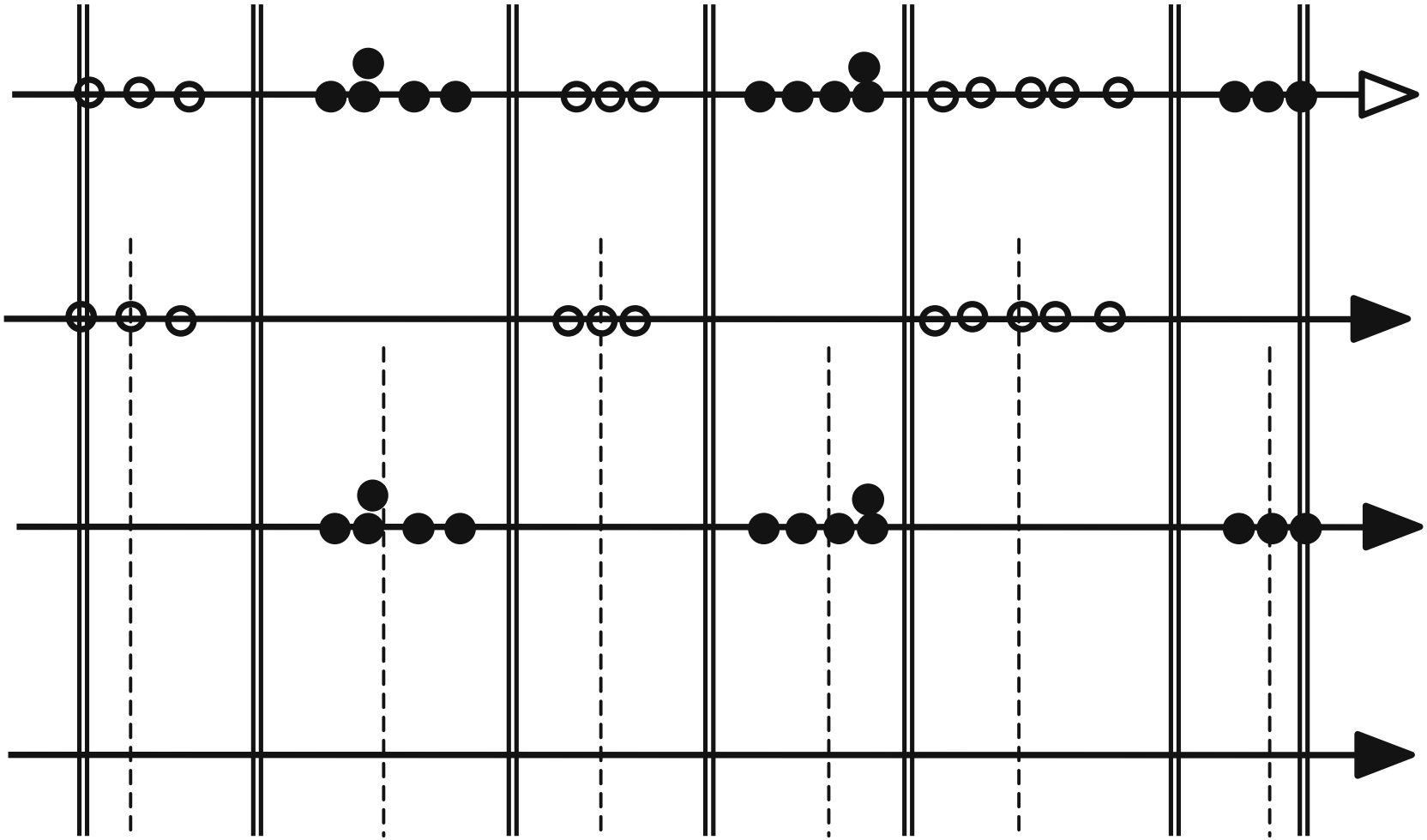
***K*-means shlukování pro diskretizaci**

Vstup: Trénovací dataset z M datapointů a C; uživatelem definovaný počet intervalů n_{F_i} pro každý atribut F_i

- 1. Pro třídu c_j (j=1,dots,C) dělej následující:
- 2. Vyber $K = n_{F_i}$ jako iniciální počet středů shluků. Nejdříve budou centra prvních K hodnot.
- 3. Distribuuj hodnoty atributů mezi K shluků na základě kritéria minimální vzdálenosti.
- 4. Vypočítej nové středy shluků tak, aby byl minimalizován součet kvadrátů vzdáleností ze všech bodů ve stejném shluku k novému středu.
- 5. Pokud jsou nové středy na stejném místě jako ty původní, pokračuj krokem 1. jinak pokračuj krokem 3.

Výstup: hranice pro jeden atribut, v podobě: minimální hodnota, maximální hodnota, středy mezi dvěma sousedními středy shluků pro všechny třídy

Ideální případ:



Algoritmus je silně ovlivněn počtem shluků, které musí určit uživatel, výběrem iniciálních shluků, a geometrií dat.
Problém výběru počtu shluků je společný pro (téměř) všechny shlukovací algoritmy. (budem se tím zabývat 7.3)