Dobývání znalostí

Doc. RNDr. Iveta Mrázová, CSc.

Katedra teoretické informatiky
Matematicko-fyzikální fakulta
Univerzity Karlovy v Praze

Dobývání znalostí

– Pokročilé techniky pro předzpracování dat –

Doc. RNDr. Iveta Mrázová, CSc.

Katedra teoretické informatiky
Matematicko-fyzikální fakulta
Univerzity Karlovy v Praze

Příprava (předzpracování) dat

- Klíčový význam pro úspěch aplikace
- Obtížné a časově náročné
- Výhodná spolupráce s expertem z dané oblasti

Cíl předzpracování:

- Vybrat (nebo vytvořit) z dostupných dat ty údaje, které jsou relevantní pro zvolenou úlohu dobývání znalostí
- Reprezentovat tyto údaje v podobě, která je vhodná pro zpracování zvoleným algoritmem
- **Cílový stav** ~ (jedna) datová tabulka zachycující hodnoty atributů objektů

Strukturovaná data

- **Časová data**
  - např. časové řady kurzů akcií
  - Typická úloha ~ predikce budoucí hodnoty
  - Příklad: časová řada po transformaci

VSTUPY	VÝSTUP
$y(t_0), y(t_1), y(t_2), y(t_3)$	$y(t_4)$
$y(t_1), y(t_2), y(t_3), y(t_4)$	$y(t_5)$
$y(t_2), y(t_3), y(t_4), y(t_5)$	$y(t_6)$
……	

Strukturovaná data (2)

- **Prostorová data**
  - např. geografické informační systémy
  - Implicitní relace sousednosti
    - hodnoty atributů „sousedních“ objektů se nebudou navzájem příliš lišit
  - Využití zejména při interpretaci
- **Strukturální data**
  - např. chemické sloučeniny
  - Zápis např. pomocí tzv. smile-kódů nebo jako soubor faktů v Prologu
  - např. klasifikace chemikálií do tříd podle jejich struktury

Více vzájemně propojených tabulek

- **Spojení (join) ~ z jedné nebo více relací (tabulek) se vytvoří relace (tabulka) nová**
- 1:1 ~ jedna entita první relace je svázaná s jednou entitou druhé relace
  - Příklad: Klient (ID_Klienta, Příjmení, Jméno)
    Trvalé_bydliště (ID_Bydliště, ID_Klienta, Ulice, Město)
  - nová relace bude obsahovat sloupce (atributy) z obou původních relací
  - počet řádků bude pro uvedené relace odpovídat počtu řádků v první relaci

Více vzájemně propojených tabulek

- **1:*n*** ~ jedna entita první relace je svázána s více entitami druhé relace
  - **Příklad:**

    Účet (ID_Účtu, Datum_založení, Četnost_výpisů)
    Trvalé_příkazy (ID_Příkazu, ID_Účtu, Částka, Bankovní_spojení)
  - nová tabulka bude obsahovat nové atributy pro agregované hodnoty získané z ***n*** opakování údajů (trvalých příkazů) vztahujících se k entitě na straně '**1**' (účtu) vztahu mezi relacemi

Více vzájemně propojených tabulek (2) (1:*n* pokračování)

- agregované hodnoty (atributy):
  - pro **numerické** atributy (např. částka): součet, minimum, maximum, průměr
  - pro **kategoriální** atributy (např. Typ_transakce): počet různých hodnot, výskyt konkrétní hodnoty, majoritní hodnota, …
- počet řádků v nové tabulce bude odpovídat počtu řádků v relaci na straně '**1**' (počtu účtů)

Více vzájemně propojených tabulek (3) (*n*:*m*)

- ***n*:*m*** ~ několika entitám z první relace odpovídá jedna entita z druhé relace a současně několika entitám z druhé relace odpovídá jedna entita z první relace
  - **Příklad:** relace Klient a relace Účet
    - jeden klient může mít přístup k více účtům; k jednomu účtu může mít přístup více klientů
  - vztah *m*:*n* lze vyjádřit pomocí další relace (např. Dispoziční_právo), která je s původními relacemi svázána vztahem 1:*n* (resp. 1:*m*)

Více vzájemně propojených tabulek (4) (*n*:*m* pokračování)

- při spojování těchto tabulek se opět vytvářejí nové atributy pro agregované hodnoty
- jednu relaci zvolíme za hlavní (primární) – např. Klient nebo Účet – a spojení budeme provádět vzhledem k ní

- **odvozené atributy** – např. agregované hodnoty při spojování relací, doménové znalosti (*rodné číslo* ⇢ *věk*)

Data s příliš mnoha objekty

- problematické zpracování v dávkovém režimu
  - ⇨ použít jen určitý vzorek (sample) vybraný ze všech dat
  - použít takový způsob uložení dat, který by umožnil přístup ke všem objektům, bez nutnosti ukládat je všechny do operační paměti
  - vytvořit více modelů na základě podmnožin objektů a modely poté zkombinovat
  - → **reprezentativnost vybraných dat**
    - vybrané objekty by měly co nejlépe vystihovat všechna data (např. podle shody rozdělení hodnot atributů ve vybraném vzorku i ve všech datech)

Data s příliš mnoha objekty (2)

- nevyvážené třídy v původních datech
  (⇒ tendence preferovat majoritní třídu)
  - různé váhy pro různý typ chybného rozhodnutí
  - vybírat příklady různých tříd s různou pravděpodobností

→ křížová validace (Cross -Validation)
  - z dat se opakovaně vybere jen určitá část pro trénování a jiná část pro testování

→ efektivnější uložení a zpracování dat (paralelizmus)

Data s příliš mnoha atributy

- redukce počtu atributů pomocí experta
- automatická redukce počtu atributů
  - **transformací** ~ z existujících atributů vytvoříme menší počet nových atributů (např. Karhunen-Loevův rozvoj, PCA, …)
    - nové atributy vzniknou jako lineární kombinace původních atributů
    - ✖ nutnost měřit hodnoty všech původních atributů
    - ✖ nové atributy nemusí mít jasnou interpretaci
  - **selekcí** ~ z existujících atributů vybereme jen ty nejdůležitější (Feature Selection)

Selekce

- **hledáme takové atributy, které nejlépe přispějí ke klasifikaci objektů do tříd**
  - **metodou filtru** ~ ke každému atributu spočítáme charakteristiku vyjadřující jeho vhodnost pro klasifikaci
  - **metodou obálky** ~ použít nějaký algoritmus strojového učení pro vytvoření modelu z podmnožiny atributů a vyhodnotit model. Použije se nejlepší z vytvořených modelů (s nalezenými atributy)
    - **„zdola nahoru“** ~ začít od modelů vytvořených pro jednotlivé atributy a atributy postupně přidávat
    - **„shora dolů“** ~ začít od modelu vytvořeného pro původní množinu atributů a atributy postupně odstraňovat

Automatická selekce metodou filtru

- **kritéria vycházející z kontingenční tabulky**

	$C(v_1)$	$C(v_2)$	…	$C(v_S)$	Σ
$A(v_1)$	a_{11}	a_{12}	…	a_{1S}	r_1
$A(v_2)$	a_{21}	a_{22}	…	a_{2S}	r_2
⋮	⋮	⋮	…	⋮	⋮
$A(v_R)$	a_{R1}	a_{R2}	…	a_{RS}	r_R
Σ	s_1	s_2	…	s_S	n

A vstupní atribut

C cílový atribut

a_{kl} četnost (frekvence) kombinace $(A(v_k) \wedge C(v_l))$

$$r_k = \sum_{l=1}^{S} a_{kl} \qquad s_l = \sum_{k=1}^{R} a_{kl} \qquad n = \sum_{k=1}^{R}\sum_{l=1}^{S} a_{kl}$$

Automatická selekce metodou filtru (2)

- **odhad pravděpodobností:**

$$P(A(v_k) \wedge C(v_l)) = \frac{a_{kl}}{n}$$

$$P(A(v_k)) = \frac{r_k}{n} \qquad P(C(v_l)) = \frac{s_l}{n}$$

- **kritéria pro výběr atributů**
  - χ^2 (čím větší hodnota, tím lépe)

$$\chi^2 = \sum_{k=1}^{R}\sum_{l=1}^{S}\frac{(a_{kl} - e_{kl})^2}{e_{kl}} = n\sum_{k=1}^{R}\sum_{l=1}^{S}\frac{\left(a_{kl} - \frac{r_k s_l}{n}\right)^2}{\frac{r_k s_l}{n}}$$

Automatická selekce metodou filtru (3) – Kritéria pro výběr atributů

- **kritéria pro výběr atributů**
  - **entropie** $H(A)$ (čím menší hodnota, tím lépe)

$$H(A) = \sum_{k=1}^{R} \frac{r_k}{n} H(A(v_k))$$

$$\text{kde } H(A(v_k)) = -\sum_{l=1}^{S} \frac{a_{kl}}{r_k} \log_2 \frac{a_{kl}}{r_k}$$

Automatická selekce metodou filtru (4) – Kritéria pro výběr atributů

- **informační míra závislosti ID(A,C)** (čím větší hodnota, tím lépe)

$$ID(A,C) = \frac{MI(A,C)}{H(C)} = \frac{MI(A,C)}{-\sum_{l=1}^{S} \frac{s_l}{n} \log \frac{s_l}{n}}$$

- kde **vzájemná informace MI(A,C)**:

$$MI(A,C) = \sum_{k=1}^{R}\sum_{l=1}^{S} P(A(v_k) \wedge C(v_l)) \log_2 \frac{P(A(v_k) \wedge C(v_l))}{P(A(v_k))P(C(v_l))}$$

$$= \sum_{k=1}^{R}\sum_{l=1}^{S} \frac{a_{kl}}{n} \log_2 \frac{\frac{a_{kl}}{n}}{\frac{r_k}{n}\frac{s_l}{n}} = \frac{1}{n}\sum_{k=1}^{R}\sum_{l=1}^{S} a_{kl} \log_2 \frac{n a_{kl}}{r_k s_l}$$

Automatická selekce metodou filtru (5)

- atributy lze uspořádat podle hodnoty kritéria a poté vybrat jen určitý počet těch nejlepších
  - Postup „zdola nahoru“ (~ přidávání atributů)
  - Postup „shora dolů“ (~ odstraňování atributů)
- ✖ každý atribut je posuzován zvlášť
- ✖ nezachycen současný vliv vícero atributů na správnost klasifikace
  - **počítat hodnotu kritéria pro množinu atributů** (~ pro všechny kombinace hodnot těchto atributů)
    - **výběr „zdola nahoru“** (~ přidáváním atributů – ***přímá selekce***)
    - **postup „shora dolů“** (~odstraňováním atributů – ***zpětná selekce***)

Numerické atributy

diskretizace numerických atributů (~ rozdělení na intervaly)

- **diskretizace na předem zadaný počet ekvidistantních intervalů** (~ obor hodnot numerického atributu se rozdělí na stejně dlouhé intervaly)
- **diskretizace s využitím informací o příslušnosti objektů k různým třídám**
- metody se liší:
  - strategií vytváření intervalů (rozdělováním intervalů shora dolů, popř. spojováním intervalů zdola nahoru)
  - počtem výsledných intervalů
  - typem intervalů
  - kritériem vyjadřujícím kvalitu intervalů (minimální klasifikační chyba, entropie, χ^2-test)

Numerické atributy: algoritmy diskretizace

- probereme následujíci algoritmy:
  1. **Algoritmus Fayyada a Iraniho**
  2. **Algoritmus Leeho a Shina**
  3. **Diskretizace pro KEX (Berka)**
  4. **Fuzzy diskretizace**

Algoritmus Fayyada a Iraniho

- zobecňuje binarizaci (~ rozděleni hodnot numerického atributu do dvou intervalů)
- rekurzivně (**shora dolů**) binarizuje aktuální interval; bere v úvahu jednotlivé dělící bodu a stanoví
  - zda se má interval dále rozdělit
  - pokud ano, který dělicí bod použít
- kritérium pro rozdělení:
  - **informační zisk** (~ přínos rozdělení daného intervalu (***Int***) pro klasifikaci)

Algoritmus Fayyada a Iraniho (2)

- **informační zisk** $Zisk\left(A_{Int,\vartheta}\right) = H\left(A\left(Int\right)\right) - H\left(A\left(\vartheta\right)\right)$
- **entropie *H(A(Int))*** se vztahuje k intervalu před binarizací

$$H\left(A\left(Int\right)\right) = -\sum_{t=1}^{T} \frac{n_t\left(A\left(Int\right)\right)}{n\left(A\left(Int\right)\right)} \log \frac{n_t\left(A\left(Int\right)\right)}{n\left(A\left(Int\right)\right)}$$

- **entropie *H (A_ϑ)*** se vztahuje k intervalu po binarizaci

$$H\left(A_\vartheta\right) = \frac{n_t\left(A\left(<\vartheta\right)\right)}{n\left(A\left(Int\right)\right)} H\left(A\left(<\vartheta\right)\right) + \frac{n_t\left(A\left(\geq\vartheta\right)\right)}{n\left(A\left(Int\right)\right)} H\left(A\left(\geq\vartheta\right)\right)$$

  - ***n (A (Int))*** … počet příkladů s hodnotu atributu ***A*** z intervalu ***Int***
  - ***t*** … index pro příklady z třídy ***t***
  - ***n (A ($<\vartheta$))***, resp. ***n (A ($\geq\vartheta$))*** … počet příkladů, jejichž hodnota atributu ***A*** je z intervalu ***Int*** a je menší, resp. větší nebo rovna než ϑ

Algoritmus Fayyada a Iraniho (3)

- k binarizaci aktuálního intervalu ***Int*** dojde, jestliže

$$Zisk\left(A_{Int,\vartheta}\right) > \frac{\log_2\left(n\left(A\left(Int\right)\right)-1\right)}{n\left(A\left(Int\right)\right)} + \frac{\Delta_A\left(Int,\vartheta\right)}{n\left(A\left(Int\right)\right)}$$

kde

$$\Delta_A\left(Int,\vartheta\right) = \log_2\left(3^k-2\right) - k\,H\left(A\left(Int\right)\right) - k_1\,H\left(A\left(<\vartheta\right)\right) - k_2\,H\left(A\left(\geq\vartheta\right)\right)$$

k … počet různých tříd pro objekty spadající do intervalu ***Int***

k_1 … počet různých tříd pro objekty spadající do intervalu $Int_{<\vartheta}$

k_2 … počet různých tříd pro objekty spadající do intervalu $Int_{\geq\vartheta}$

Algoritmus Fayyada a Iraniho (4)

Fayyadův a Iraniho algoritmus:

1. uspořádej trénovací data vzestupně podle hodnoty diskretizovaného atributu
2. rekurzivně binarizuj aktuální interval ***Int*** tak, že
   - 2.1. najdi nejvhodnější dělicí bod ϑ a urči pro něj ***Zisk*** $(A_{int,\vartheta})$
   - 2.2. je-li

$$Zisk\left(A_{Int,\vartheta}\right) > \frac{\log_2(n-1)}{n} + \frac{\Delta_A(Int,\vartheta)}{n}$$

     - 2.2.1. rozděl interval ***Int*** na intervaly $Int_{<\vartheta}$ a $Int_{\geq\vartheta}$
     - 2.2.2. pokračuj v rekurzi

Algoritmus Leeho a Shina

- diskretizace zdola nahoru založená na postupném spojování hodnot numerického atributu do předem zadaného počtu výsledných intervalů
- při posuzování intervalů se měří rozdíl mezi množstvím informace ve všech datech a množstvím informace v intervalu

$$E(Int) = \left[\sum_t \left(\sqrt{P(Class_t)} - \sqrt{P(Class_t \mid Int)} \right)^2 \right]^{\frac{1}{2}}$$

Algoritmus Leeho a Shina (2)

$$E(Int) = \left[\sum_t \left(\sqrt{P(Class_t)} - \sqrt{P(Class_t \mid Int)} \right)^2 \right]^{\frac{1}{2}}$$

- podobně pro dělicí bod ϑ

$$E(\vartheta) = \left[\sum_t \left(\sqrt{P(Class_t \mid A(< \vartheta))} - \sqrt{P(Class_t \mid A(\geq \vartheta))} \right)^2 \right]^{\frac{1}{2}}$$

- v obou vztazích je:

$$P(Class_t) = \frac{n_t}{n} \qquad \text{a} \qquad P(Class_t \mid Int) = \frac{n_t(Int)}{n(Int)}$$

- daty se prochází opakovaně; při každém průchodu se spojí dvojice intervalů oddělených od sebe dělicím bodem s nejmenší hodnotou $E(\vartheta)$

Algoritmus Leeho a Shina (3)

- **algoritmus:**

Inicializace

1. uspořádej trénovací data vzestupně podle hodnoty diskretizovaného atributu
2. pro každý dělicí bod $\vartheta_i = (a_i + a_{i+1}) / 2$
   - 2.1. vytvoř interval $Int_i = [\vartheta_i , \vartheta_{i+1}]$
   - 2.2. spočítej $E(Int_i)$ a $E(\vartheta_i)$

Algoritmus Leeho a Shina (4)

- **Algoritmus:**

  Hlavní cyklus

  1. dokud není dosažen požadovaný počet intervalů

     1.1. najdi ϑ_{min} takové, že $E(\vartheta_{min}) = \min_i E(\vartheta_i)$

     1.2. vytvoř interval

     $$Int_{min} = [\vartheta_{min-1}, \vartheta_{min}] \cup [\vartheta_{min}, \vartheta_{min+1}]$$

     1.3. spočítej $E(Int_{min})$, $E(\vartheta_{min-1})$ a $E(\vartheta_{min+1})$

Diskretizace numerických atributů pro systém KEX

Cíl:

- diskretizace, která povede k vytvoření pravidel
  $A(Int) \Rightarrow Class;$
  $A(Int) = [Dmez, Hmez]$
- Snaha vytvářet takové intervaly $A(Int)$, aby se $P(Class \mid A(Int))$ signifikantně lišilo od $P(Class)$

- postup zdola nahoru spojováním

Diskretizace numerických atributů pro systém KEX (2)

- **Hlavní cyklus:**

1. vytvoř uspořádaný seznam hodnot uvažovaného atributu
2. pro každou hodnotu
   - 2.1. spočítej četnosti výskytu objektů s touto hodnotou pro jednotlivé třídy
   - 2.2. přiřaď kód třídy každé hodnotě procedurou **Assign**
3. vytvoř intervaly hodnot procedurou **Interval**

Diskretizace numerických atributů pro systém KEX (3)

Assign:

pokud pro danou hodnotu všechny objekty patří do stejné třídy

a) pak přiřaď hodnotě kód této třídy;

b) jinak pokud se pro danou hodnotu veličiny rozdělení objektů do tříd signifikantně liší (na základě χ^2-testu) od apriorního rozdělení tříd

   a) pak přiřaď hodnotě kód nejčetnější třídy

   b) jinak přiřaď hodnotě kód '?'

Diskretizace numerických atributů pro systém KEX (4)

Počáteční intervaly

1. Nechť $x_1 < \ldots < x_p$ jsou všechny hodnoty, které nabývá uvažovaný atribut; předpokládáme, že $p \geq 2$
2. vytvoř intervaly $Int_1 = \langle DMez_1, HMez_1 \rangle = \langle x_1, x_1 \rangle$,

$Int_2 = \langle DMez_2, HMez_2 \rangle = \langle x_2, x_2 \rangle$,

$\ldots$

$Int_p = \langle DMez_p, HMez_p \rangle = \langle x_p, x_p \rangle$

Spojování intervalů:

- Spojením intervalů $\langle d_1, h_1 \rangle$ a $\langle d_2, h_2 \rangle$, kde $h_1 < d_2$ vznikne interval $\langle d_1, h_2 \rangle$

Diskretizace numerických atributů pro systém KEX (5)

Interval:

3. procházej intervaly od prvního do posledního

3.1. pokud má sekvence po sobě jdoucích intervalů stejný kód třídy, pak vytvoř jeden interval spojením těchto intervalů

3.2. jinak, pokud interval Int_i patří do třídy **'?'**

pak pokud jeho sousední intervaly Int_{i-1} a Int_{i+1} patří do téže třídy

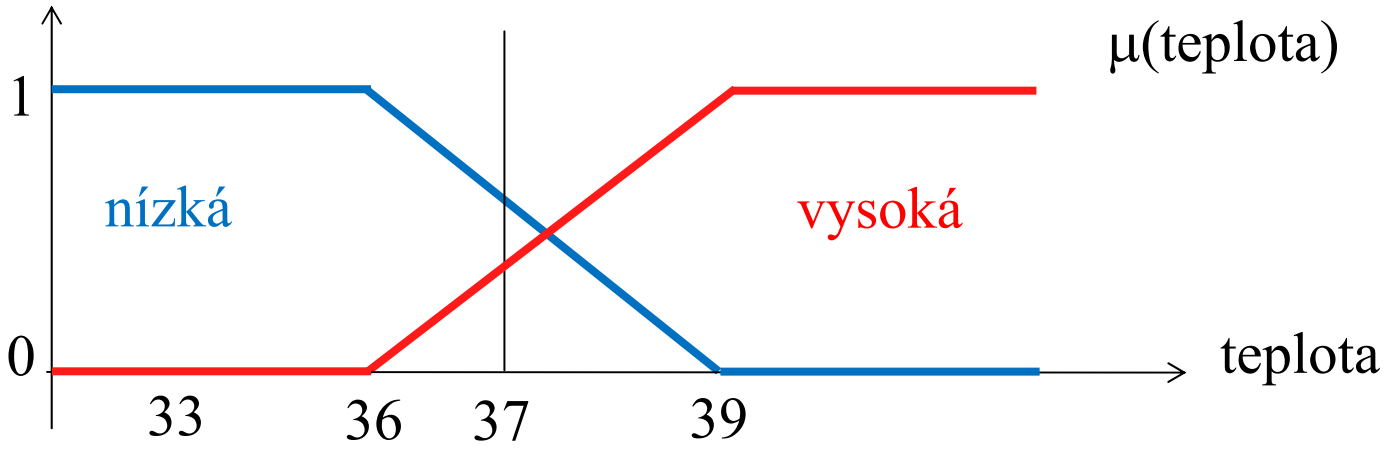
3.2.1. vytvoř interval $\langle DMez_{i-1}, HMez_{i+1} \rangle$ spojením $Int_{i-1} \cup Int_i \cup Int_{i+1}$

3.2.2. jinak vytvoř interval buď spojením $Int_{i-1} \cup Int_i$ nebo spojením $Int_i \cup Int_{i+1}$ podle výsledku χ^2-testu

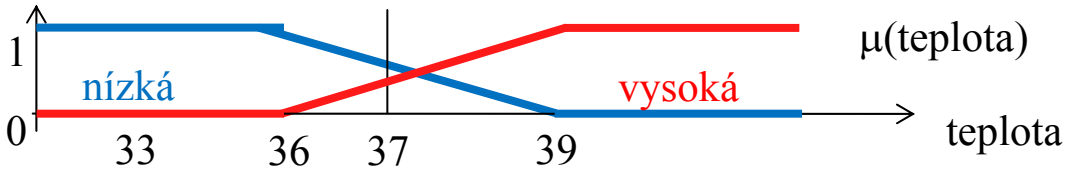
3.3. pokud dolní mez některého z vytvořených intervalů není x_1, tak jeho dolní mez nastav na horní mez předchozího vytvořeného intervalu, interval bude zleva otevřený $\Rightarrow$ dostaneme spojité pokrytí definičního oboru atributu

Fuzzy diskretizace

- hranice fuzzy-intervalů jsou ‘neostré’ ⇒ nová procedura **‘Interval’** pro vytváření intervalů spojením sekvencí hodnot téže (fuzzy-)třídy
- **fuzzy-intervaly**



Fuzzy diskretizace (2)



- z jedné numerické hodnoty můžeme získat dvě diskretizované hodnoty tak, že součet odpovídajících charakteristických funkcí bude roven 1
  - vznikají ‘fuzzy’-objekty s vahou < 1
  - váha ‘fuzzy’-objektu je dána součinem hodnot charakteristických funkcí všech atributů:

$$w(obj) = \prod_j \mu(x_i)$$

  - možná ztráta informací
  - **kontradikce** ~ objekty se stejnými hodnotami vstupních (diskretizovaných) atributů patří do různých tříd

Fuzzy diskretizace (3): procedura 'Interval'

3.1. pokud má sekvence hodnot stejný kód třídy, pak vytvoř interval z těchto hodnot (s charakteristickou funkcí rovnou 1 v celém intervalu)

3.2. pro každý interval Int_i

pokud interval Int_i patří do třídy '?'

pak pokud jeho sousední intervaly Int_{i-1} a Int_{i+1} patří do téže třídy

3.2.1. vytvoř interval spojením $Int_{i-1} \cup Int_i \cup Int_{i+1}$ (s charakteristickou funkcí rovnou 1 v celém intervalu)

3.2.2. jinak vytvoř dva fuzzy-intervaly spojením $Int_{i-1} \cup Int_i$ a spojením $Int_i \cup Int_{i+1}$ viz následující slide

Fuzzy diskretizace (4): procedura 'Interval'

3.2.2. jinak vytvoř jeden interval spojením $\boldsymbol{Int_{i-1} \cup Int_i}$ tak, že charakteristická funkce je rovna 1 v intervalu $\boldsymbol{[Dmez_{i-1}, Hmez_{i-1}]}$ a rovna

$$\frac{DMez_{i+1} - x}{DMez_{i+1} - HMez_{i-1}} \quad \text{pro} \quad x \in [HMez_{i-1}, DMez_{i+1}]$$

a druhý interval spojením $\boldsymbol{Int_i \cup Int_{i+1}}$ tak, že charakteristická funkce je rovna 1 v intervalu $\boldsymbol{[Dmez_{i+1}, Hmez_{i+1}]}$ a rovna

$$\frac{x - HMez_{i-1}}{DMez_{i+1} - HMez_{i-1}} \quad \text{pro} \quad x \in [HMez_{i-1}, DMez_{i+1}]$$

3.3. vytvoř spojité pokrytí definičního intervalu veličiny shodně s krokem 3.2.2. (mezery mezi intervaly jsou chápány jako interval třídy '**?**')

Kategoriální atributy

Algoritmus pro seskupování hodnot (KEX):

- pokud nabývá kategoriální atribut příliš velkého počtu hodnot, je vhodné jejich **seskupování** (např. na základě charakteristik spočítaných na trénovacích datech)

⇒ **Cíl:** vytvořit skupiny ***A*(*Grp*)** takové, aby se ***P*(*Class* | *A*(*Grp*))** signifikantně lišilo od ***P*(*Class*)**

\+ vytvoří se tolik skupin, kolik je různých tříd plus jedna navíc (´**?**´)

Kategoriální atributy (2)

Algoritmus pro seskupování hodnot (KEX):

Hlavní cyklus:

1. vytvoř uspořádaný seznam hodnot uvažovaného atributu
2. pro každou hodnotu
   2.1. spočítej četnosti výskytu objektů s touto hodnotou pro jednotlivé třídy
   2.2. přiřaď kód třídy každé hodnotě procedurou **Assign**
3. vytvoř intervaly hodnot procedurou **Group**

- *až na volání* **Group** *v kroku 3 stejný jako pro diskretizaci numerických atributů pro systém KEX*

Kategoriální atributy (3)

Algoritmus pro seskupování hodnot (KEX):

Assign:

Pokud pro danou hodnotu všechny objekty patří do stejné třídy

1. pak přiřaď hodnotě kód této třídy;
2. jinak – pokud se pro danou hodnotu veličiny rozdělení objektů do tříd signifikantně liší (na základě χ^2-testu) od apriorního rozdělení tříd
   a) Pak přiřaď hodnotě kód nejčetnější třídy
   b) Jinak přiřaď hodnotě kód **'?'**

Stejný jako pro diskretizaci numerických atributů pro systém KEX

Kategoriální atributy (4)

Algoritmus pro seskupování hodnot (KEX):

Group:

vytvoř skupinu z těch hodnot, které mají přiřazen kód stejné třídy

Chybějící hodnoty

- ignorovat objekt s nějakou chybějící hodnotou
- nahradit chybějící hodnotu novou hodnotou '***nevím***'
- nahradit chybějící hodnotu některou z existujících hodnot atributu
  - nejčetnější hodnotou
  - proporcionálním podílem všech hodnot
  - libovolnou hodnotou
- doplnění chybějící hodnoty na základě (použitého) modelu