Časté množiny položek, asociační pravidla

Jiří Kléma

Katedra kybernetiky,
FEL, ČVUT v Praze

http://ida.felk.cvut.cz

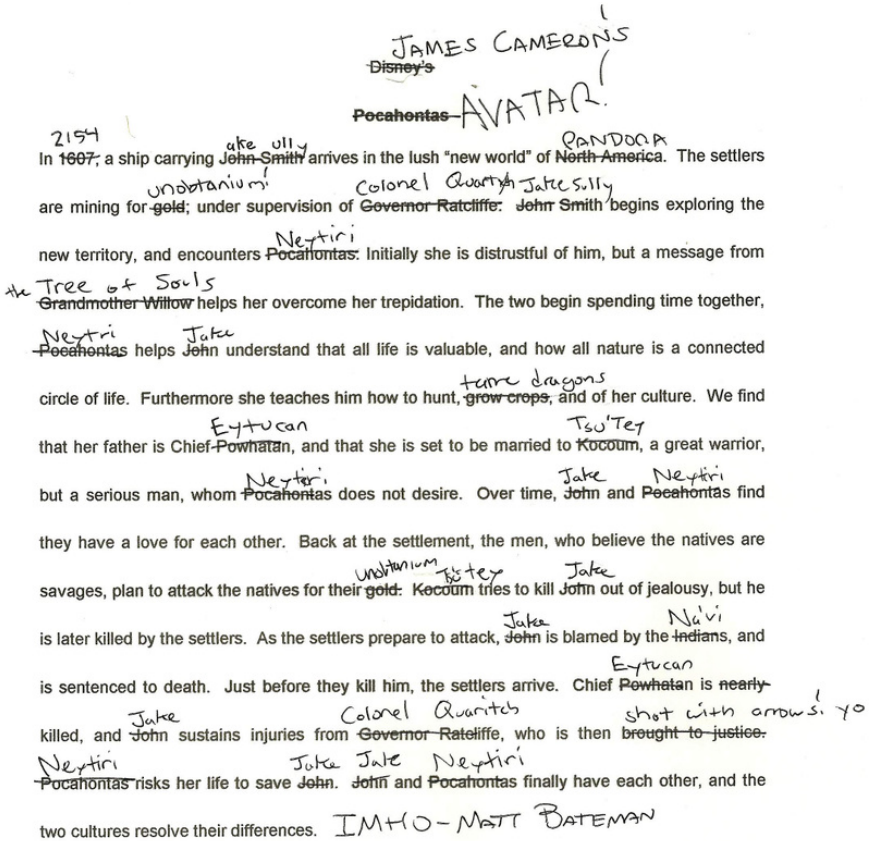
Struktura přednášky

- Motivace pro hledání spoluvýskytů událostí
  - původ – analýza nákupního košíku,
  - jiný příklad, obecné cíle,
  - lokální modely,
- formalizace asociačních pravidel
  - definice pojmů – podpora, spolehlivost,
  - (pod)problém vyhledávání velkých množin položek,
- algoritmy pro vyhledávání velkých množin položek,
  - algoritmus APRIORI,
  - algoritmy ECLAT a (FPGrowth),
- generování pravidel z velkých množin položek
  - algoritmus AR-Gen.
- ekvivalenční alternativy implikační spolehlivosti.

Obecná motivace

- data miningová legenda
  - pivo a dětské pleny.

- detekce plagiátů
  - je AVATAR uměleckou krádeží?


JAMES CAMERON'S
~~Disney's~~
~~Pocahontas~~ AVATAR

In ~~1607,~~ 2154 a ship carrying ~~John Smith~~ Jake Sully arrives in the lush "new world" of ~~North America~~ PANDORA. The settlers are mining for ~~gold~~ unobtanium; under supervision of ~~Governor Ratcliffe:~~ Colonel Quaritch ~~John Smith~~ Jake Sully begins exploring the new territory, and encounters ~~Pocahontas:~~ Neytiri. Initially she is distrustful of him, but a message from ~~Grandmother Willow~~ the Tree of Souls helps her overcome her trepidation. The two begin spending time together, ~~Pocahontas~~ Neytiri helps ~~John~~ Jake understand that all life is valuable, and how all nature is a connected circle of life. Furthermore she teaches him how to hunt, ~~grow crops,~~ tame dragons, and of her culture. We find that her father is Chief ~~Powhatan~~ Eytucan, and that she is set to be married to ~~Kocoum~~ Tsu'Tey, a great warrior, but a serious man, whom ~~Pocahontas~~ Neytiri does not desire. Over time, ~~John~~ Jake and ~~Pocahontas~~ Neytiri find they have a love for each other. Back at the settlement, the men, who believe the natives are savages, plan to attack the natives for their ~~gold:~~ unobtanium ~~Kocoum~~ Tsu'tey tries to kill ~~John~~ Jake out of jealousy, but he is later killed by the settlers. As the settlers prepare to attack, ~~John~~ Jake is blamed by the ~~Indians~~ Na'vi, and is sentenced to death. Just before they kill him, the settlers arrive. Chief ~~Powhatan~~ Eytucan is ~~nearly~~ killed, and ~~John~~ Jake sustains injuries from ~~Governor Ratcliffe~~ Colonel Quaritch, who is then ~~brought to justice.~~ shot with arrows. ~~Pocahontas~~ Neytiri risks her life to save ~~John.~~ Jake ~~John~~ Jake and ~~Pocahontas~~ Neytiri finally have each other, and the two cultures resolve their differences. IMHO - MATT BATEMAN

Asociační pravidla

- Association Rules (ARs)
- Definice
  - jednoduchá tvrzení o spoluvýskytu událostí v datech,
  - pravděpodobnostní charakter – nemusí platit vždy,
- Způsob zápisu a význam
  - if **Ant** then **Suc**,
  - alternativní zápis: Ant $\Rightarrow$ Suc,
  - antecedent (Ant) a sukcedent (Suc) definují obecné události v datech,
  - událost – jasně definovatelný jev, který buď nastává nebo nenastává,
  - z extenzivního popisu (dat) generujeme zhuštěný a přehledný popis – znalost.
- Příklady asociačních pravidel
  - doporučení pro nákup knih (Amazon):
    {Castaneda: **Učení Dona Juana**} $\Rightarrow$ {Hesse: **Stepní vlk** & Ruiz: **Čtyři dohody**}
  - vztah mezi rizikovými faktory a onemocněním v lékařství (Stulong):
    {**pivo** $\geq$ 1litr/den & **destiláty**=0} $\Rightarrow$ {not(**srdeční onemocnění**)}

Asociační pravidla – pojmy

- Položky (items): $I = \{I_1, I_2, \ldots, I_m\}$
  - binární atributy nebo výsledky aplikace relačních operátorů,
- Transakce (transactions): $D = \{t_1, t_2, \ldots, t_n\}$, $t_i \subseteq I$
  - příklady, objekty,
- Množiny položek (itemsets): $\{I_{i1}, I_{i2}, \ldots, I_{ik}\} \subseteq I$
  - analogie podmínky, současná platnost více položek,
- Podpora množiny položek: (relativní) četnost transakcí obsahujících danou množinu položek
- Frekventovaná (velká) množina položek:
  - četnost výskytů dané množiny položek je vyšší než zvolený práh,
- Asociační pravidlo (AR): implikace Ant $\Rightarrow$ Suc, kde Ant, Suc $\subseteq$ I a Ant $\cap$ Suc $= \emptyset$,
- Podpora (support) AR, s: poměr resp. počet transakcí obsahujících Ant $\cup$ Suc
  - pozn: podpora Ant $\Rightarrow$ Suc je stejná jako podpora Ant $\cup$ Suc,
- Spolehlivost (confidence) AR, α:
  - poměr mezi podporou AR (Ant $\cup$ Suc) a podporou jeho antecedentu (Ant),
  - vždy menší nebo rovna 1.

AR – problém vyhledávání

- Jsou dány:
  - množina položek $I = \{I_1, I_2, \ldots, I_m\}$,
  - databáze transakcí $D = \{t_1, t_2, \ldots, t_n\}$
    - kde $t_i = \{I_{i1}, I_{i2}, \ldots, I_{ik}\}$, a $I_{ij} \in I$,
  - minimální podpora s_{min},
  - minimální spolehlivost α_{min}.
- Problém vyhledávání asociačních pravidel:
  - nalézt všechna pravidla Ant $\Rightarrow$ Suc s podporou $s \geq s_{min}$ a spolehlivostí $\alpha \geq \alpha_{min}$.
- Realizaci lze rozdělit na 2 kroky:
  - najdi všechny frekventované (velké) (pod)množiny položek,
  - generuj z nich pravidla.

Př.: analýza nákupního košíku

- Cíl: zvýšit prodej a omezit náklady, tj. najdi položky často kupované společně,

Transakce	Položky
t_1	Chleba, Džem, Máslo
t_2	Chleba, Máslo
t_3	Chleba, Mléko, Máslo
t_4	Pivo, Chleba
t_5	Pivo, Mléko

- I = {Pivo, Chleba, Džem, Mléko, Máslo},
- Příklad pravidla: Chleba $\Rightarrow$ Máslo,
  - Ant={Chleba}$\in \{t_1, t_2, t_3, t_4\}$, s_{ant}=4/5=80%,
  - Ant $\cup$ Suc={Chleba,Máslo} $\in \{t_1, t_2, t_3\}$, podpora AR je s=3/5=60%,
  - Spolehlivost AR je $\alpha = s/s_{ant}$=75%.

- Další pravidla a jejich parametry:

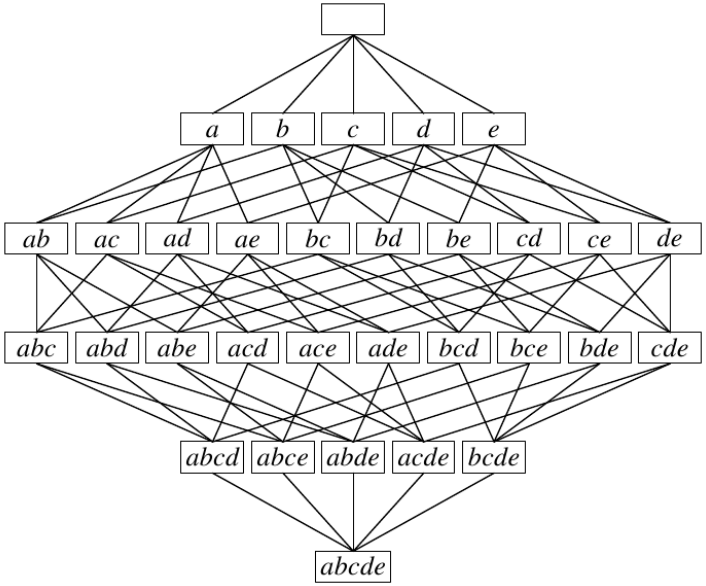
Ant $\Rightarrow$ Suc	s [%]	α [%]
Chleba $\Rightarrow$ Máslo	60	75
Máslo $\Rightarrow$ Chleba	60	100
Pivo $\Rightarrow$ Chleba	20	50
Máslo $\Rightarrow$ Džem	20	33
Džem $\Rightarrow$ Máslo	20	100
Džem $\Rightarrow$ Mléko	0	0

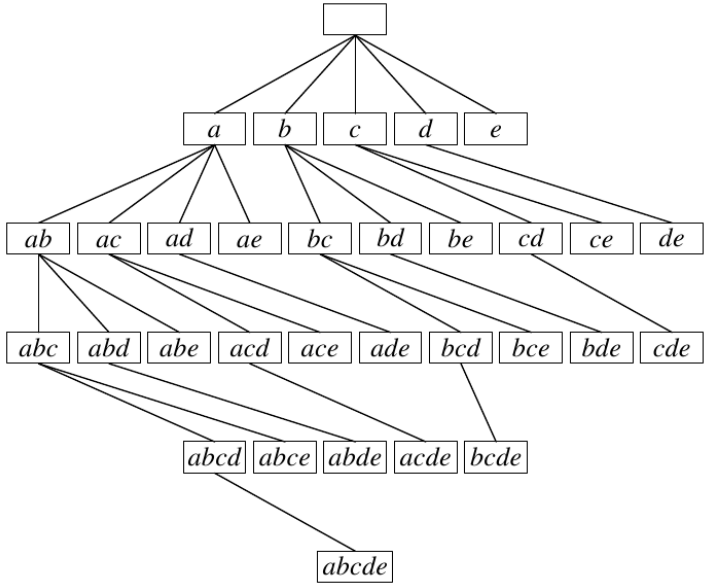
Vyhledávání velkých množin položek

- krok 1 při generování asociačních pravidel,
- často samostatná úloha
  - najdi množiny produktů, které se nakupují současně,
  - obecně **najdi jevy, které se často vyskytují spolu**,
    - analýza textů: transakce = dokumenty, položky = slova,
    - detekce plagiátů: transakce = výskyty věty, položky = dokumenty,
- úplné prohledávání prostoru množin položek
  - pro m binárních položek existuje $3^m - 1$ množin položek,
  - pro N atributů, každý z nich má K kategorií existuje $(1 + K)^N - 1$ množin položek,
  - složitost roste **exponenciálně** s počtem položek (atributů),
- u rozsáhlejších problémů předpoklady
  - řídkých dat – ne vše souvisí se vším,
  - včasného a efektivního prořezávání prohledávacího prostoru.

Vyhledávání velkých množin položek – kategorizace přístupů (1)

- množina všech množin položek je **částečně uspořádaná**
  - lze ji popsat acyklickým grafem – Hasseovým diagramem,
  - uzly jsou množiny položek, hrana $I \longrightarrow J$ pokud $I < J$ a neexistuje $K : \ I < K < J$,
  - v případě všech podmnožin množiny jde o **mřížku** (lattice),
  - tu lze z důvodu efektivity (každý uzel je třeba testovat jen jednou) redukovat na **strom**,
- metoda prohledávání mřížky/stromu podmnožin
  - do šířky – po úrovních, každá úroveň zahrnuje množiny položek dané délky,
  - do hloubky – přes množiny položek s identickým prefixem.





Vyhledávání velkých množin položek – kategorizace přístupů (2)

- reprezentace množiny/databáze transakcí
  - horizontální – po transakcích, transakce $\approx$ seznam/pole položek,
    - přirozenější způsob,
  - vertikální – po položkách, udržujeme seznam transakcí pro každou položku,
    - výhodou rychlý přístup k seznamu transakcí páru položek (rekurzivně i množiny položek),
    - jde o průnik seznamů transakcí jednotlivých položek.

Transakce	Položky
t_1	a, d, e
t_2	b, c, d
t_3	a, c, e
t_4	a, c, d, e
t_5	a, e
t_6	a, c, d
t_7	b, c
t_8	a, c, d, e
t_9	b, c, e
t_{10}	a, d, e

a	b	c	d	e
1	2	2	1	1
3	7	3	2	3
4	9	4	4	4
5		6	6	5
6		7	8	8
8		8	10	9
10		9		10

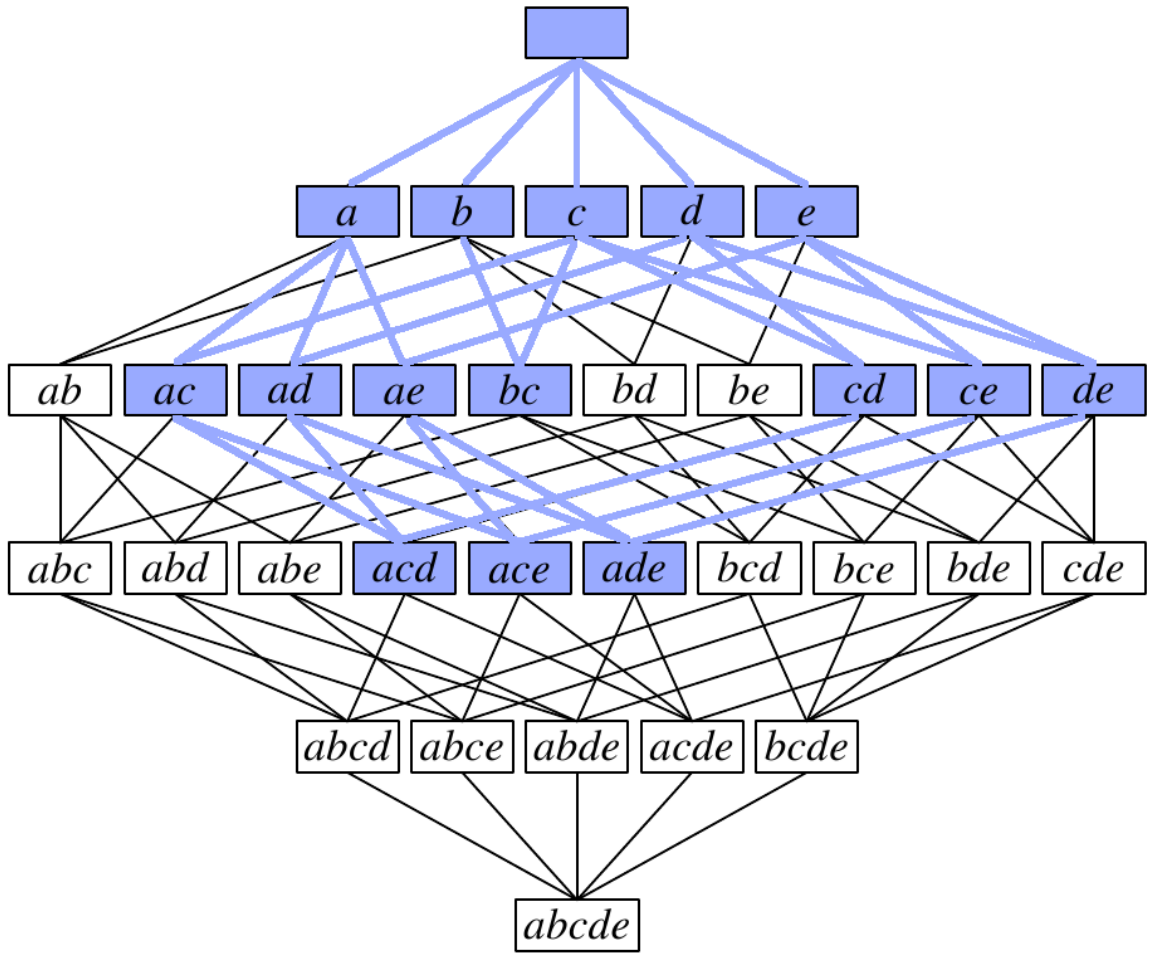
Algoritmus APRIORI – základní myšlenka

- průkopnický, nejznámější, ale ne nejefektivnější,
- využívá charakteristické vlastnosti velkých množin položek:
  Každá podmnožina velké množiny položek je velká.
- my ale postupujeme zdola nahoru – od podmnožin k nadmnožinám
  proto princip **kontrapozitivity** (přemístění) v logice
  $(p \Rightarrow q) \Leftrightarrow (\neg q \Rightarrow \neg p)$
- antimonotónní vlastnost se převádí na monotónní vlastnost, důsledek:
  Pokud množina položek není velká, žádná z jejích nadmnožin není velká.
- kandidátské množiny položek
  - potenciálně velké – o všech jejich podmnožinách je známo, že jsou velké.
- kategorizace APRIORI: prohledávání do šířky, horizontální reprezentace transakcí.

Vyhledávání velkých množin položek – ilustrace

mřížka s velkými množinami položek pro $s_{min} = 3$

Transakce	Položky
t_1	a, d, e
t_2	b, c, d
t_3	a, c, e
t_4	a, c, d, e
t_5	a, e
t_6	a, c, d
t_7	b, c
t_8	a, c, d, e
t_9	b, c, e
t_{10}	a, d, e



APRIORI algoritmus [Agrawal et al., 1996]

```
Apriori:
  C_1 = ∀ kandidátské množiny položek velikosti 1 v I;
  L_1 = ∀ velké množiny položek velikosti 1 (podpora ≥ s_min);
  i = 1;
  repeat
    i = i + 1;
    C_i = Apriori-Gen(L_{i-1});
    Počítej podporu C_i a vytvoř L_i;
  until žádná velká množina položek nenalezena (L_i = ∅);
  L = ⋃ L_i, ∀ i

Apriori-Gen(L_{i-1}):
  C_i = ∅
  pro ∀ dvojice množin položek Comb_p, Comb_q ∈ L_{i-1}:
    pokud se shodují v i-2 položkách pak přidej Comb_p ∪ Comb_q do C_i
  pro ∀ množiny položek Comb z C_i:
    pokud jakákoli podmnožina Comb o délce i-1 ∉ L_{i-1} pak odstraň Comb.
```

Aplikace APRIORI – příklad analýzy nákupního košíku

Transakce	Položky
t_1	Chleba, Džem, Máslo
t_2	Chleba, Máslo
t_3	Chleba, Mléko, Máslo
t_4	Pivo, Chleba
t_5	Pivo, Mléko

- Vstupní parametry: s_{min}=30% (α_{min}=50% – bude použito až v dalším kroku)

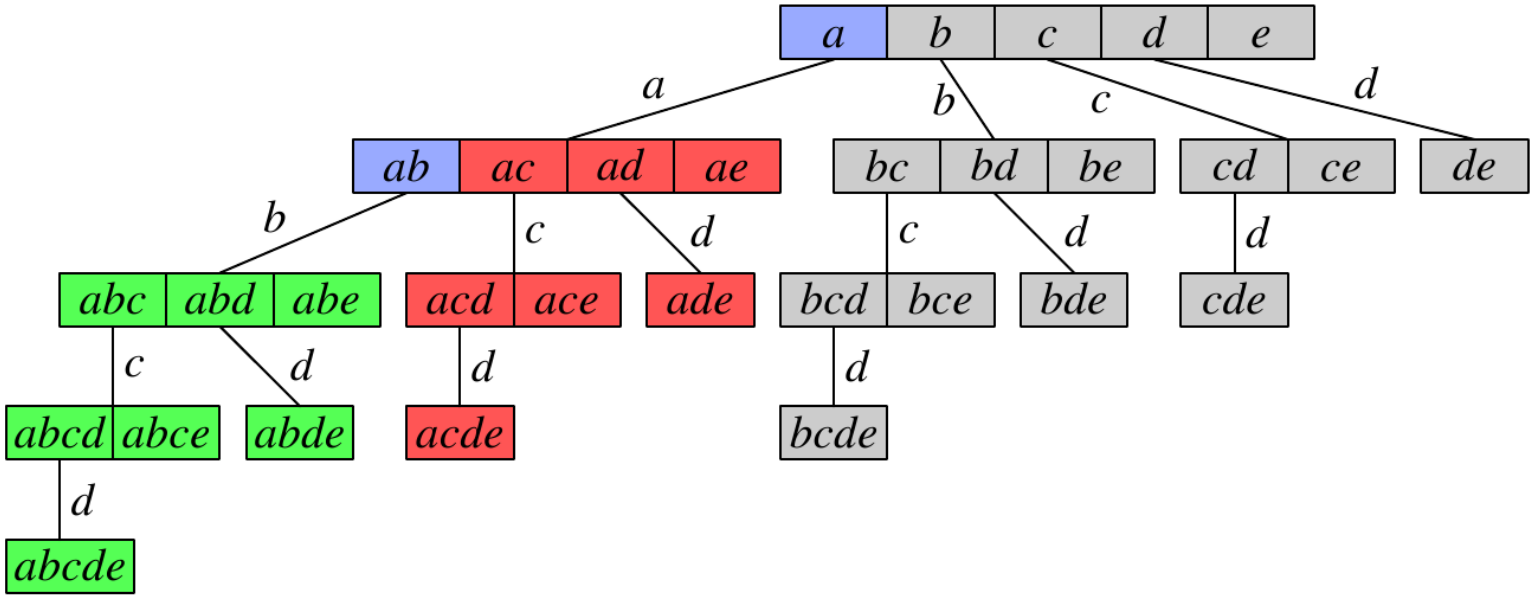
i	C_i	L_i
1	{Pivo}, {Chleba}, {Džem} {Mléko}, {Máslo}	{Pivo}, {Chleba} {Mléko}, {Máslo}
2	{Pivo, Chleba}, {Pivo, Mléko} {Pivo, Máslo}, {Chleba, Mléko} {Chleba, Máslo}, {Mléko, Máslo}	{Chleba, Máslo}

ECLAT algoritmus (Zaki et al., 1997) – základní myšlenka

- lexikograficky uspořádá položky $\longrightarrow$ kanonická reprezentace množin položek
  - $\{a, b, c\} \approx abc < bac < bca < \cdots < cba$,
  - pro množinu položek volíme lexikograficky nejmenší (největší) kódové slovo,
- strom prochází **do hloubky**
  - vzhledem ke kanonické reprezentaci jde o prefixový strom,
- používá ryze **vertikální** reprezentaci množiny transakcí,
- může generovat více kandidátských množin položek než APRIORI
  - v okamžiku rozhodnutí nemá k dispozici podpory všech podmnožin.

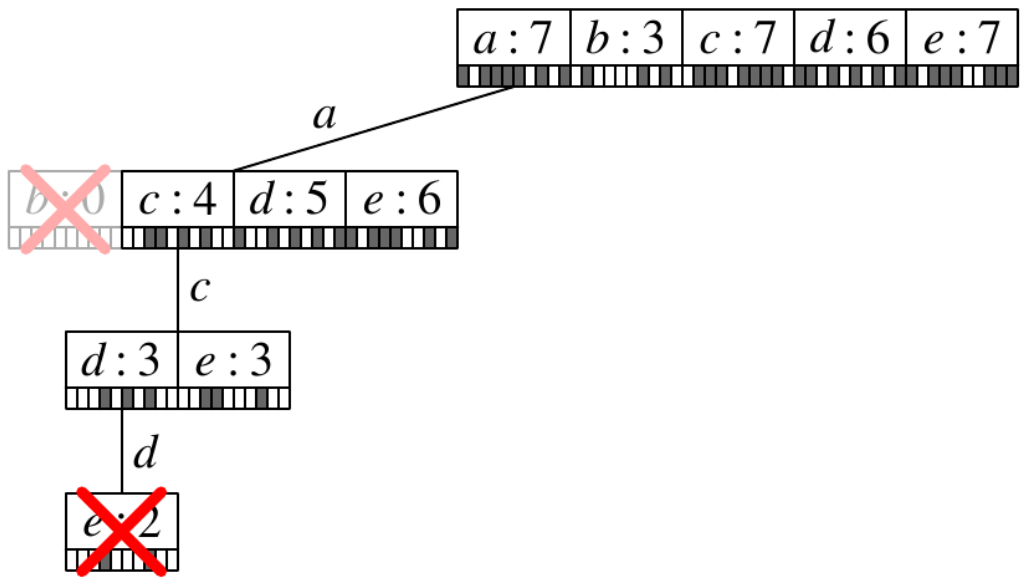
Podmíněné transakční databáze – prohledávání do hloubky

- prohledávání prefixového stromu do hloubky analogií **rozděl a panuj**
  - nejprve najdi všechny velké množiny položek s daným prefixem,
  - následně totéž bez něj,
  - vede na dělení množiny transakcí (transakce s/bez daného prefixu),
- obarvení uzlů
  - modře prefix, zeleně položky s prefixem, červeně bez prefixu,
  - postup je rekurzivní, je třeba vzít v úvahu již předchozí krok s a.



Příklad ECLAT, $s_{min} = 3$ (Borgelt: Frequent Pattern Mining)

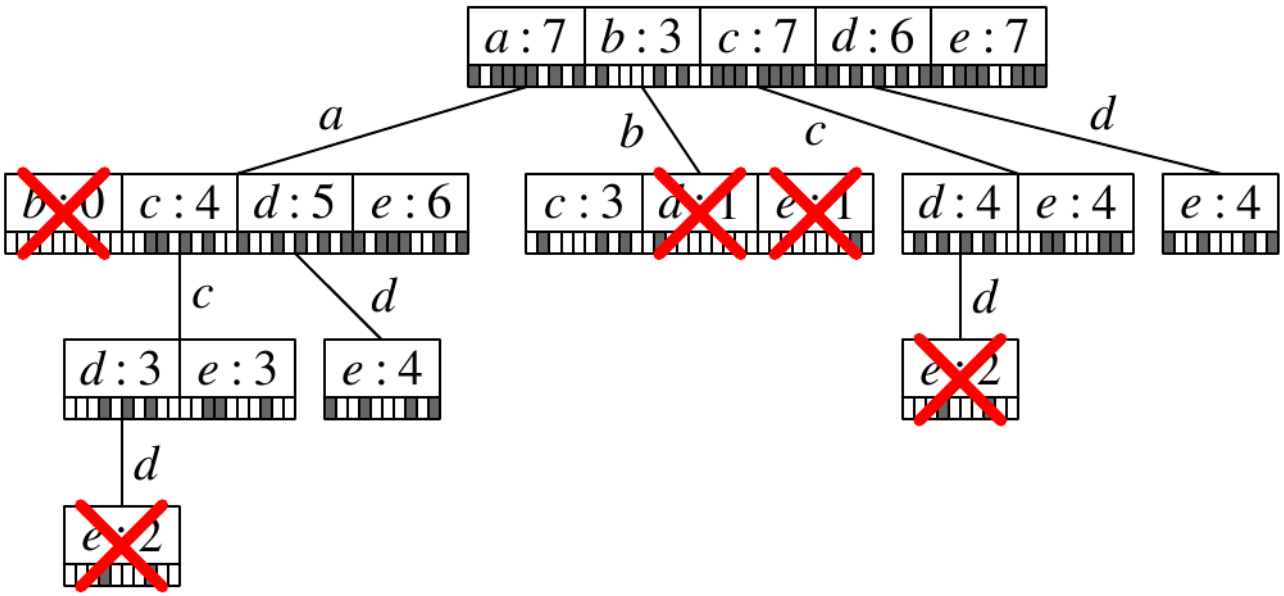
t_1	a, d, e
t_2	b, c, d
t_3	a, c, e
t_4	a, c, d, e
t_5	a, e
t_6	a, c, d
t_7	b, c
t_8	a, c, d, e
t_9	b, c, e
t_{10}	a, d, e



- předzpracování: vertikální reprezentace bitovým vektorem (šedá/bílá – položka v/mimo transakci)
  - jediný průchod transakční databází, následně pouze průniky,
- krok 1: podmíněná transakční databáze pro položku a,
- krok 2: $\{a, b\}$ není velká – prořízni,
- krok 3: podmíněná transakční databáze pro množinu $\{a, c\}$,
- krok 4: podmíněná transakční databáze pro množinu $\{a, c, d\}$ a prořízni $\{a, c, d, e\}$.

Příklad ECLAT, $s_{min} = 3$ (Borgelt: Frequent Pattern Mining)

t_1	a, d, e
t_2	b, c, d
t_3	a, c, e
t_4	a, c, d, e
t_5	a, e
t_6	a, c, d
t_7	b, c
t_8	a, c, d, e
t_9	b, c, e
t_{10}	a, d, e



- ilustrace celého stromu, výsledek (samozřejmě) shodný s APRIORI
- u APRIORI mohla být $\{a, c, d, e\}$ proříznuta bez výpočtu podpory,
  - víme, že $s_{\{c,d,e\}} = 2 \leq s_{min} = 3$,
- u APRIORI naopak více přístupů do transakční databáze.

Generování pravidel z velkých množin položek – krok 2

```
Vstupy:
  I, D, L, α_min;

Výstup:
  R; % pravidla splňující s_min a α_min

AR-Gen:
  R = ∅;
  pro ∀ l ∈ L proveď:
    pro ∀ x ⊂ l taková, že x ≠ ∅ a x ≠ l proveď:
      jestliže s(l)/s(x) ≥ α_min, pak R = R ∪ {x ⇒ (l-x)}
```

- Příklad analýzy nákupního košíku
  - Vstupní parametry: L={Chleba, Máslo} (generováno pro s_{min}=30%), α_{min}=50%
  - Výstup: R={Chleba ⇒ Máslo: s=60%, α=75%, Máslo ⇒ Chleba: s=60%, α=100%}

Př.: průchod studiem

- Cíl: zjistit, zda skutečný průchod studiem odpovídá předpokladům a doporučením
- Předměty: RZN (Reprezentace znalostí), PAH (Plánování a hry), VI (Výpočetní inteligence), MAS (Multi-agentní systémy), SAD (Strojové učení a analýza dat), AU (Automatické uvažování)

Transakce	Položky
t_1	RZN
t_2	VI, SAD, AU
t_3	PAH, AU
t_4	PAH, VI, AU
t_5	PAH, MAS
t_6	VI, AU
t_7	PAH, SAD
t_8	PAH, VI, MAS, AU
t_9	PAH
t_{10}	PAH, VI, AU

Transakce	Položky
t_{11}	AU
t_{12}	RZN, PAH, VI, SAD, AU
t_{13}	PAH, VI, MAS, AU
t_{14}	VI, SAD, AU
t_{15}	PAH, AU
t_{16}	SAD, AU
t_{17}	RZN, PAH, SAD
t_{18}	PAH, VI, MAS, AU
t_{19}	PAH
t_{20}	PAH, VI, MAS, AU

APRIORI krok – s_{min}=20%, resp. 4

i	C_i	L_i
1	{RZN}, {PAH}, {VI} {MAS}, {SAD}, {AU}	{PAH}, {VI}, {MAS} {SAD}, {AU}
2	{PAH, VI}, {PAH, MAS}, {PAH, SAD} {PAH, AU}, {VI, MAS}, {VI, SAD} {VI, AU}, {MAS, SAD}, {MAS, AU} {SAD, AU}	{PAH, VI}, {PAH, MAS} {PAH, AU}, {VI, MAS} {VI, AU}, {MAS, AU} {SAD, AU}
3	{PAH, VI, MAS}, {PAH, VI, AU} {PAH, MAS, AU}, {PAH, SAD, AU} {VI, MAS, AU}, {VI, SAD, AU} {MAS, SAD, AU}	{PAH, VI, MAS} {PAH, VI, AU} {PAH, MAS, AU} {VI, MAS, AU}
4	{PAH, VI, MAS, AU}	{PAH, VI, MAS, AU}
5	$\emptyset$	$\emptyset$

AR-Gen krok – α_{min}=80%

PAH, VI: PAH ⇒ VI α=50%, VI ⇒ PAH α=70%
(PAH & VI současně 7krát, PAH 14 krát, VI 10 krát)
PAH, MAS: PAH ⇒ MAS 36%, **MAS ⇒ PAH** 100%
PAH, AU: PAH ⇒ AU 57%, AU ⇒ PAH 57%
VI, MAS: VI ⇒ MAS 40%, **MAS ⇒ VI** 80%
VI,AU: **VI ⇒ AU** 100%, AU ⇒ VI 71%
MAS, AU: **MAS ⇒ AU** 80%, AU ⇒ MAS 29%
SAD, AU: SAD ⇒ AU 66%, AU ⇒ SAD 29%

PAH, VI, MAS: PAH & VI ⇒ MAS 57%, **PAH & MAS ⇒ VI** 80%,
VI & MAS ⇒ PAH 100%
PAH, VI, AU: **PAH & VI ⇒ AU** 100%, **PAH & AU ⇒ VI** 88%,
VI & AU ⇒ PAH 70%
PAH, MAS, AU: **PAH & MAS ⇒ AU** 80%, PAH & AU ⇒ MAS 50%,
MAS & AU ⇒ PAH 100%
VI, MAS, AU: **VI & MAS ⇒ AU** 100%, **MAS & AU ⇒ VI** 100%,
VI & AU ⇒ MAS 40%

PAH, VI, MAS, AU: **PAH & VI & MAS ⇒ AU** 100%, PAH & VI & AU ⇒ MAS 57%,
VI & MAS & AU ⇒ PAH 100%, PAH & VI ⇒ MAS & AU 57%, atd.

APRIORI – výhody a nevýhody

- Výhody
  - efektivně využívá monotónní vlastnosti velkých množin položek,
  - obecně stále exponenciální složitost, ale zvládnutelný výpočet při:
    - vhodné volbě s_{min} a α_{min},
    - řídkých datech (v praxi spíše platí).
  - snadná implementace včetně paralelizace,
  - pro kolerovaná data s velkým počtem velkých množin položek může být dále vylepšen
    - použití zahuštěné reprezentace.
- Nevýhody
  - předpokládá residentní umístění databáze transakcí v paměti,
  - vyžaduje až m (počet položek) průchodů databází,
    - přístup lze urychlit použitím hashovacích stromů,
    - počet průchodů lze také snížit sloučením dvou následných velikostí do jednoho kroku,
    - za to zaplatíme větším počtem kandidátských množin, ale ...

Zápis vztahu mezi Ant a Suc čtyřpolní tabulkou

- Čtyřpolní tabulka = 4-fold table (4FT),
  - a, b, c, d $\longrightarrow$ počty transakcí splňujících podmínky.

4FT	Suc	¬Suc	$\sum$
Ant	a	b	r=a+b
¬Ant	c	d	s=c+d
$\sum$	k=a+c	l=b+d	n=a+b+c+d

- Ne vždy je spolehlivost smysluplným kvantifikátorem
  - u často platných sukcedentů je **implikační** charakter spolehlivosti zavádějící,
  - i nezávislé množiny položek vykazují vysokou spolehlivost,
  - příklad čtyřpolní tabulky (s=45%,α=90%, Ant a Suc nezávislé):

450	50	500
450	50	500
900	100	1000

Příklady alternativních kvantifikátorů

- Spolehlivost lze v kroku AR-Gen nahradit libovolnou funkcí nad hodnotami čtyřpolní tabulky:
  - **Zdvih** (lift, above-average) je míra zlepšení přesnosti defaultní predikce pravé strany (spolehlivost dělená obecným podílem příkladů pokrytých sukcedentem)
    - lift=an/rk
  - **Páka** (leverage) je podíl příkladů, které jsou pravidlem (tedy Ant i Suc) pokryté dodatečně nad rámec počtu příkladů pokrytých za předpokladu nezávislosti Ant a Suc.
    - leverage=1/n(a-rk/n)
  - **Přesvědčivost** (conviction) je podobná zdvihu, ale uvažuje příklady nepokryté Suc pravidla, tím pádem musí pracovat s převráceným poměrem uvažovaných četností.
    - conviction=rl/bn

450	50	500
450	50	500
900	100	1000

s=0.45, α=0.9,
zdvih=1, páka=0,
přesvědčivost=1

10	1	11
90	899	989
100	900	1000

s=0.01, α=0.91,
zdvih=9.09, **páka=0.01**,
přesvědčivost=9.9

450	50	500
50	450	500
500	500	1000

s=0.45, α=0.9,
zdvih=1.8, páka=0.2,
přesvědčivost=5

Asociační pravidla – shrnutí

- Základní nástroj deskriptivního dolování dat
  - obecně identifikace jakéhokoli častého spoluvýskytu událostí v datech,
  - identifikace podskupin, odhalování skrytých závislostí, extrakce znalostí.
- Praktické využití
  - nejenom analýza nákupního košíku!!!
  - obecně jakákoli data atributového typu
    - lékařství, průmyslová měření, časo-prostorová data, . . . ,
  - nutností je pouze transakční převod dat, tj. přechod na binární atributy
    - nejčastěji dichotomizací (postupným tříděním do 2 skupin),
    - pro numerické veličiny navíc diskretizací,
    - v úvahu připadá i kódování (minimalizuje počet položek, ale snižuje přehlednost výstupu),
    - př. atribut teplota ve st. Celsia
      - diskretizace: $\{(-\infty,0\rangle \equiv$ nízká, $(0,15\rangle \equiv$ střední, $(15, \infty) \equiv$ vysoká$\}$,
      - dichotomizace: $\{i_1 \equiv$ t=nízká, $i_2 \equiv$ t=střední, $i_3 \equiv$ t=vysoká$\}$,

Doporučené doplňky – zdroje přednášky

:: Četba

- Agrawal, Srikant: **Fast Algorithms for Mining Association Rules.**
  - článek, který definoval problém a navrhl APRIORI algoritmus,
  - http://rakesh.agrawal-family.com/papers/vldb94apriori.pdf,
- Borgelt: **Frequent Pattern Mining.**
  - slajdy, velmi detailní kurz, včetně formálního popisu,
  - http://www.borgelt.net/teach/fpm/slides.html,