



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií ■

Vyhledávání vzorů

František Kynych
24. 12. 2022 | MVD





TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií ■

Část I.: Úvod do problematiky



Vyhledávání vzorů

- Proces automatického rozpoznání vzorů a pravidelností v datech
- Co je to vzor?
 - Množina prvků, dílčích sekvencí nebo podstruktur, které se v datech často vyskytují společně (silně korelované)
 - Vzory často reprezentují vnitřní a důležité vlastnosti dat
- Základ pro další aplikace vytěžování dat
 - Vytěžování vzorů, analýza kauzality, analýza vzorů v multimédiích, časových řadách a datových proudech
 - Klasifikace, shlukování

Praktické aplikace

- Které produkty jsou často společně zakoupeny
- Co dalšího lidé kupují po zakoupení určitého produktu
- Analýza trhu
- Analýza kódu a logů
- Detekce frází
- Detekce silně korelovaných úseků v časových řadách (biologická data, finance, ...)
- ...

Časté vzory (Frequent patterns)

- **Itemset**
 - Množina jedné nebo více položek
- **k-itemset**: $X = \{x_1, \dots, x_k\}$
- **Absolutní support X**
 - Frekvence výskytů nějaké položky X v celé databázi
 - Pro kávu = 3
- **Relativní support, s**
 - Podíl transakcí obsahující položku X
 - Pravděpodobnost, že bude transakce obsahovat položku X
 - Pro kávu = 60 %
- Itemset je častý (frequent), pokud je **support s** vyšší než námi daný práh (**minsup**, označován také jako σ)

ID	Zakoupené položky
1	Káva, čaj, mléko
2	Káva, vejíčka, mléko
3	Káva, mléko, maso
4	Čaj, maso, pečivo
5	Čaj, vejíčka, mléko, maso, pečivo

Časté vzory (Frequent patterns)

Příklad: minsup = 50 %

- Hledáme množiny s hodnotou $s > 50 \%$

Častý 1-itemset:

- Mléko: 4/5 (80 %)
- Káva: 3/5 (60 %); čaj: 3/5 (60 %); maso: 3/5 (60 %)

Častý 2-itemset:

- {Káva, mléko}: 3 (60 %)

ID	Zakoupené položky
1	Káva, čaj, mléko
2	Káva, vejíčka, mléko
3	Káva, mléko, maso
4	Čaj, maso, pečivo
5	Čaj, vejíčka, mléko, maso, pečivo

Asociační pravidla

- Asociační pravidla: $X \rightarrow Y(s, c)$
 - Pokud uživatel koupí položku X , jaký je support (s) a jistota (confidence, c), že uživatel koupí položku Y
 - **Support** (s)
 - Pravděpodobnost, že transakce obsahuje zároveň položky X a Y ($X \cup Y$)
 - U itemsetu se značí obsažení obou položek zároveň sjednocením, nikoliv průnikem
 - **Confidence** (c)
 - Podmíněná pravděpodobnost, že transakce obsahující X obsahuje i Y
 - $$c = \frac{\text{sup}(X \cup Y)}{\text{sup}(X)}$$
 - Těžení dat pomocí asociačních pravidel
 - Snažíme se najít všechny pravidla $X \rightarrow Y$ s parametry **minsup** a **minconf**

Asociační pravidla

Příklad: minsup = 50 %

Častý 1-itemset:

- mléko: 4, **káva: 3**, čaj: 3, maso: 3

Častý 2-itemset:

- {káva, mléko}: **3 (60 %)**

minconf = 50 %

Asociační pravidla: $X \rightarrow Y(s, c)$

$$c = \frac{\text{sup}(X \cup Y)}{\text{sup}(X)}$$

$$c = \frac{\text{sup}(káva \cup mléko)}{\text{sup}(káva)} = \frac{3}{3} = 100 \%$$

$káva \rightarrow mléko(60 \%, 100 \%)$;

$mléko \rightarrow káva(60 \%, 75 \%)$

Zakoupené položky

Káva, čaj, mléko

Káva, vejíčka, mléko

Káva, mléko, maso

Čaj, maso, pečivo

Čaj, vejíčka, mléko, maso, pečivo

Reprezentace vzorů

- Kolik částých vzorů obsahuje databáze TDB_1 ?
 - $TDB_1: T_1: \{a_1, \dots, a_{50}\}; T_2: \{a_1, \dots, a_{100}\}$
 - Předpokládaný minsup = 1
 - Výsledek:
 - 1-itemsets: $\{a_1\}: 2, \dots, \{a_{50}\}: 2, \{a_{51}\}: 1, \dots, \{a_{100}\}: 1$
 - 2-itemsets: $\{a_1, a_2\}: 2, \dots, \{a_1, a_{50}\}: 2, \{a_1, a_{51}\}: 1, \dots, \{a_{99}, a_{100}\}: 1$
 - ...
 - 99-itemsets: $\{a_1, a_2, \dots, a_{99}\}: 1, \{a_2, a_3, \dots, a_{100}\}: 1$
 - 100-itemsets: $\{a_1, a_2, \dots, a_{100}\}: 1$
 - Celkem $\binom{100}{1} + \binom{100}{2} + \dots + \binom{100}{100} = 2^{100} - 1$ vzorů
- Složité na výpočet i uložení

Reprezentace vzorů - komprese

- **Uzavřené vzory (closed patterns)**

- Vzor (itemset) X je uzavřený, pokud X je častý vzor a neexistuje žádný super-vzor (nadmnožina) $Y \supset X$ se stejnou support hodnotou jako X

Příklad:

- $TDB_1: T_1: \{a_1, \dots, a_{50}\}; T_2: \{a_1, \dots, a_{100}\}$
- Předpokládaný minsup = 1
- TDB_1 nyní obsahuje pouze dva **uzavřené** vzory
 - $P_1: \{a_1, \dots, a_{50}\}: 2$
 - $P_2: \{a_1, \dots, a_{100}\}: 1$
- Bezeztrátová komprese
 - Stále dokážeme určit např. $\{a_2, \dots, a_{40}\}: 2$

Reprezentace vzorů - komprese

- **Maximální vzory (max-patterns)**
 - Vzor X je maximální, pokud X je častý vzor a neexistuje žádný super-vzor (nadmnožina) $Y \supset X$
 - Rozdíl oproti uzavřenému vzoru:
 - Nebereme ohled na support hodnotu

Příklad:

- $TDB_1: T_1: \{a_1, \dots, a_{50}\}; T_2: \{a_1, \dots, a_{100}\}$
- Předpokládaný minsup = 1
- TDB_1 nyní obsahuje pouze jeden **maximální** vzor
 - $P: \{a_1, \dots, a_{100}\}: 1$
- Ztrátová komprese
 - Víme, že vzor $\{a_2, \dots, a_{40}\}$ je častý, ale neznáme support hodnotu

Část II.: Základní přístupy vyhledávání vzorů

Apriori algoritmus

- Využívá apriori principu
 - Každá podmnožina častého vzoru (itemsetu) musí být také častá

Algoritmus:

1. Projít databázi a najít časté 1-itemsety ($k=1$)
2. Opakovat:
 - 1) Vygenerovat všechny možné $(k+1)$ -itemsety
 - 2) Otestovat vygenerované kandidáty na databázi s ohledem na parametr *minsup* a smazat neexistující
 - 3) Inkrementovat k
 - Opakovat dokud nemohou být generováni další kandidáti
3. Navrátit nalezené itemsety

Apriori algoritmus

Databáze

ID	Položky
1	A, C, D
2	B, C, E
3	A, B, C, E
4	B, E

minsup = 2

1. kontrola DB

Itemset	Sup
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

F_1

Itemset	Sup
{A}	2
{B}	3
{C}	3
{E}	3

Generování kandidátů

Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}

2. kontrola DB

F_2

Itemset	Sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

Gen. kandidátů

Itemset
{B, C, E}

3. kontrola DB

Itemset	Sup
{B, C, E}	2

F_3

- Navrátit $F_1 \cup F_2 \cup F_3$

FPGrowth

- Frequent pattern growth (FPGrowth)
- Řeší nedostatky Apriori algoritmu
 - Apriori musí procházet opakovaně databázi pro získání support hodnoty
 - Není potřeba vytvářet list kandidátů
- Prochází databázi pouze dvakrát s využitím stromové struktury (FP-tree) pro uložení informací
- Po vytvoření stromu se využívá divide-and-conquer přístup pro vytěžení častých vzorů

Algoritmus:

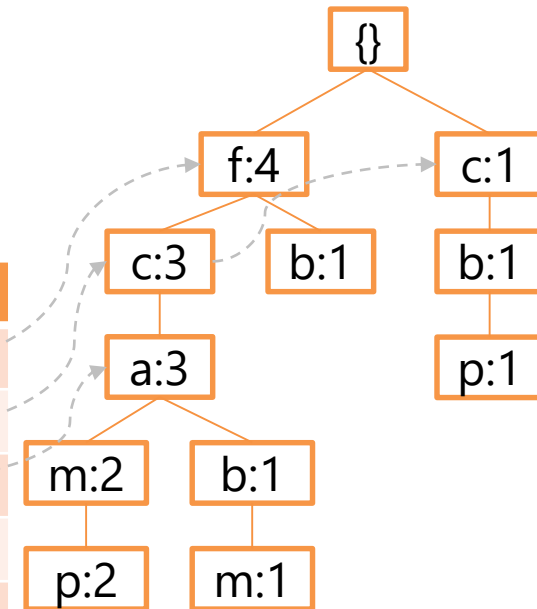
1. Seřazení položek v transakci dle support hodnot obsažených položek
2. Vytvoření FP-stromu na základě získaných hodnot v kroku 1.
3. Z FP-stromu vytvoříme FP-podmíněné stromy (FP-conditional tree) pro každou položku (nebo itemset)
4. Nalezení častých vzorů

FP-tree

ID	Položky v transakcích	Seřazené časté položky
1	{f, a, c, d, g, i, m, p}	{f, c, a, m, p}
2	{a, b, c, f, l, m, o}	{f, c, a, b, m}
3	{b, f, h, j, o, w}	{f, b}
4	{b, c, k, s, p}	{c, b, p}
5	{a, f, c, e, l, p, m, n}	{f, c, a, m, p}

- Projití DB a nalezení častých položek
minsup = 3
 Výsledek **f: 4, a: 3, c: 4, b: 3, m: 3, p: 3**
- Seřazení položek dle jejich frekvence
F-list = f-c-a-b-m-p
- Procházíme znovu databázi
 a vytváříme FP-strom

Položka	Sup	Pointer
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	

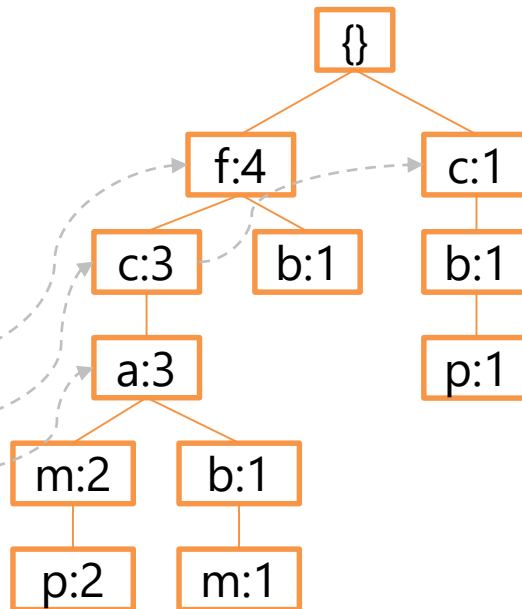


- Pointer ukazuje na první prvek v linked listu

FP-conditional tree

minsup = 3

Položka	Sup	Pointer
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	



Položka	Conditional pattern base
c	f:3
a	fc:3
b	fca:1, f:1, c:1
m	fca:2, fcab:1
p	fcam:2, cb:1

- Obsahuje prefixové cesty položek

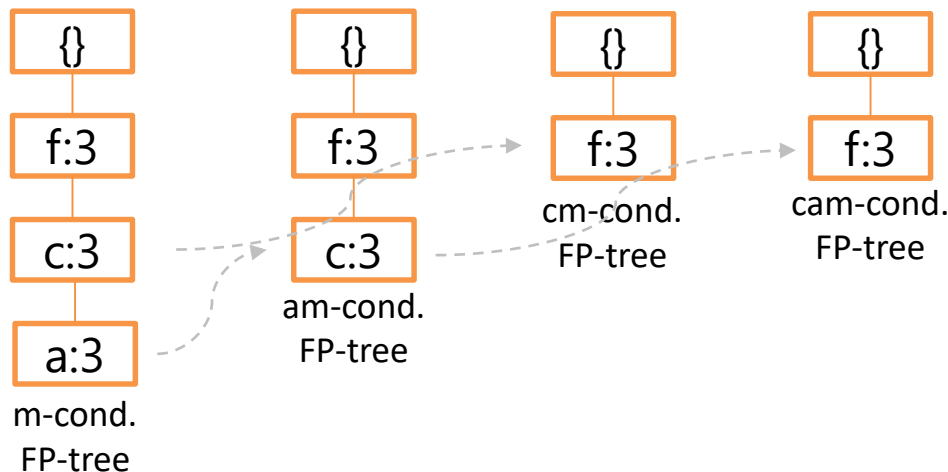
FP-conditional tree

Položka	Conditional pattern base
c	f:3
a	fc:3
b	fca:1, f:1, c:1
m	fca:2, fcab:1
p	fcam:2, cb:1

minsup = 3

Zbývá vytěžit data z conditional pattern base

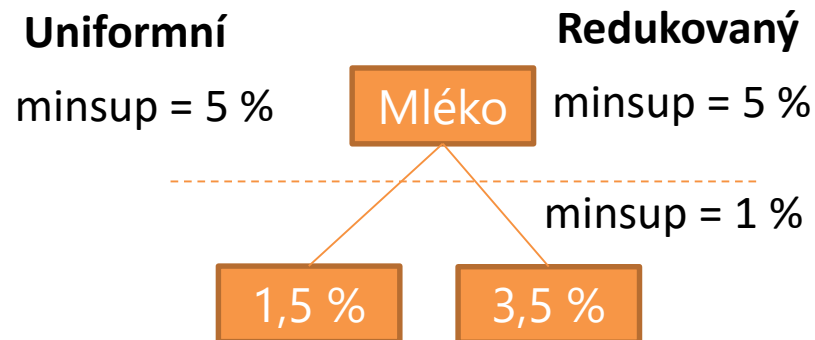
- p-conditional PB: ***fcam: 2, cb: 1*** → ***c: 3***
- m-conditional PB: ***fca: 2, fcab: 1*** → ***fca: 3***
- b-conditional PB: ***fca: 1, f: 1, c: 1*** → **\emptyset**



m: 3
fm: 3, cm: 3, am: 3
fcm: 3, fam: 3, cam: 3
fcam: 3

Těžení víceúrovňových častých vzorů

- Produkty často tvoří hierarchie
 - Např. hlavní kategorie mléko
 - Dále se dělí na jednotlivé značky a druhy
- Jak nastavit minsup parametr:
 - Uniformní minsup přes všechny úrovně
 - Redukovaný support v závislosti na úrovni



- V praxi je potřeba nastavit různou hodnotu minsup pro odlišné kategorie
 - Drahé produkty se neprodávají tak často, ale mohou mít velký přínos k příjmu

Těžení sekvenčních častých vzorů

- Sekvenční databáze obsahuje seřazené položky v závislosti na pořadí jejich zakoupení
- Aplikace:
 - Nákupy zákazníků (telefon, obal -> kabel -> sklo)
 - Biologické signály, přírodní katastrofy (zemětřesení)
 - DNA sekvence
 - Akcie a trhy

SID	Sekvence
1	<a(abc)(ac)d(cf)>
2	<(ad)c(bc)(ae)>
3	<(ef)(ab)(df)cb>
4	<eg(af)cbc>

a, (abc), (ac), ... jsou jednotlivé transakce
<a(bc)dc> je subsekvence <**a(abc)(ac)d(cf)**>

minsup = 2
<(ab)c> je sekvenční vzor

SPADE algoritmus

- SPADE (Sequential Pattern Discovery using Equivalent class)
- Založeno na Apriori principu a generování kandidátů
- Využívá vertikální transformace databáze
- Databáze sekvencí je transformována do velké množiny položek <SID, EID> (<sequence_id, element_id>)

SID	Sekvence
1	<a(abc)(ac)d(cf)>
2	<(ad)c(bc)(ae)>
3	<(ef)(ab)(df)cb>
4	<eg(af)cbc>



SID	EID	Items
1	1	a
1	2	abc
1	3	ac
1	4	d
1	5	cf
2	1	ad
2	2	c
2	3	bc
2	4	ae
3	1	ef
3	2	ab
3	3	df
3	4	c
3	5	b
4	1	e
4	2	g
4	3	af
4	4	c
4	5	b
4	6	c

<https://faculty.cc.gatech.edu/~hic/CS7616/pdf/lecture13.pdf>

SPADE algoritmus

minsup = 2

SID	EID	Items
1	1	a
1	2	abc
1	3	ac
1	4	d
1	5	cf
2	1	ad
2	2	c
2	3	bc
2	4	ae
3	1	ef
3	2	ab
3	3	df
3	4	c
3	5	b
4	1	e
4	2	g
4	3	af
4	4	c
4	5	b
4	6	c



a		b		...
SID	EID	SID	EID	...
1	1	1	2	
1	2	2	3	
1	3	3	2	
2	1	3	5	
2	4	4	5	
3	2			
4	3			



ab			ba			...
SID	EID (a)	EID(b)	SID	EID (b)	EID(a)	...
1	1	2	1	2	3	
2	1	3	2	3	4	
3	2	5				
4	3	5				



aba				...
SID	EID (a)	EID(b)	EID(a)	...
1	1	2	3	
2	1	3	4	

- Pro nalezení sekvencí **ab** je potřeba, aby **EID(a)** bylo v dané sekvenci **SID** menší než **EID(b)** = prvek **a** byl před **b**



Užitečná literatura / kurzy

- [Coursera Data Mining specializace](#)
- [Těžení sekvenčních vzorů](#)
- [Sequential Pattern Mining: Approaches and Algorithms](#)

