Lecture 9. Frequent pattern mining in streams

Ricard Gavaldà

MIRI Seminar on Data Streams, Spring 2015

Contents

1. Frequent pattern mining - batch
2. Frequent pattern mining in data streams
3. IncMine: itemset mining in MOA

Frequent pattern mining - batch

Frequent pattern mining - batch

- $\mathscr{P}$: a set of *patterns*
- $\preceq$: *subpattern* relation, a *partial order*

Examples:

- sets with subset relation
- sequences with (some) subsequence relation
- trees with (some) subtree relation
- graphs with (some) subgraph relation

Frequent pattern mining - batch

- $\mathscr{D}$: a database, or multiset, of patterns
- $s(\mathscr{D}, p)$ = absolute support of p in $\mathscr{D} = |\{p' \in \mathscr{D} : p \preceq p'\}|$
- $\sigma(\mathscr{D}, p)$ = relative support = $s(\mathscr{D}, p)/|\mathscr{D}|$
- σ: a minimum support threshold

The frequent pattern mining task

Given $\mathscr{D}$, σ, find all the patterns p such that $\sigma(\mathscr{D}, p) \geq \sigma$

Frequent pattern mining - batch

Computationally costly, for two reasons:

1. Many candidate frequent patterns
   - e.g. 2^k itemsets if k distinct items
2. Many frequent patterns *actually present* in database

Frequent pattern mining - batch

For problem 1: discard many candidate patterns soon

Antimonotonicity - the apriori principle

If $p \preceq p'$, then $\sigma(p) \geq \sigma(p')$

For problem 2: compute a smaller set with same information

Closed pattern (in $\mathscr{D}$)

p is closed if every proper superpattern of p has strictly smaller support

Closed patterns

Fact

Frequent patterns and their frequencies can be generated (easily) from closed patterns and their frequencies

There are typically much fewer frequent closed patterns that there are frequent patterns

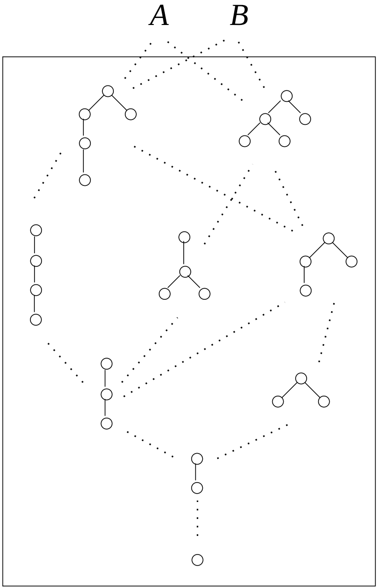
$\therefore$

savings if we only compute closed frequent patterns

Frequent closed patterns - batch

Central Concept
(and data structure):

Galois Lattice



Frequent closed patterns - batch

batch frequent closed ...

- *itemset* miners: CLOSET, CHARM, CLOSET+ ...
- *sequence* miners [Wang 04]
- *tree* miners [Balcazar-Bifet-Lozano 06-10]
- *graph* miners [Yan03]

Frequent pattern mining in data streams

Frequent patterns in data streams

Requirements:

low time per pattern, small memory, adapt to change

Taxonomy:

- Exact or Approximate
  - with false positives and/or false negatives
- Per batch or per transaction
- Incremental, sliding window, or fully adaptive
- Frequent or frequent closed

Frequent closed patterns

A general framework [Bifet-G 11] (based on [BBL06-10])

- Use a base *batch miner*
- Collect a batch of transactions from stream
- Compute *all* closed patterns and counts, C
- *Merge* C into summary of frequent closed patterns for stream

Closure Operator

Given a dataset $\mathscr{D}$ of patterns and a pattern t,

Closure of a pattern

$\Delta_{\mathscr{D}}(t)$, the closure of t, is the intersection of all patterns in $\mathscr{D}$ that contain t

Fact

t is closed in $\mathscr{D}$ if and only if it is in $\Delta_{\mathscr{D}}(t)$

Note: no mention of support!!

Adding and removing pattern batches

Proposition

A pattern t is closed in $\mathscr{D}1 \cup \mathscr{D}2$ if and only if

- it is closed in $\mathscr{D}1$, or
- it is closed in $\mathscr{D}2$, or
- it is a subpattern of a closed pattern in $\mathscr{D}1$, and of a closed subpattern in $\mathscr{D}2$, and is in $\Delta_{\mathscr{D}1}(t) \cap \Delta_{\mathscr{D}2}(t)$

Incremental Algorithm

Computing the lattice of frequent patterns

Construct empty lattice L;

Repeat

  Collect batch of B patterns;

  Build closed pattern lattice for B, L';

  L = merge(L,L') (using addition rule);

  delete from L patterns with support below σ

Memory & time depend on lattice size (= number of closed patterns), not on DB size!

Batch size depends on tradeoff batch miner time / merging time

Fully adaptive algorithm

- Keep a window on recent stream batches
  - Actually, only their lattices of closed patterns
- When new batch added, drop oldest batch, and undo its effect using closure definition

Alternatively:
Use change detectors to decide which batches are stale
E.g. on number of patterns that enter or leave lattice

Further improvement: relaxed support

Consider *c-relaxed support intervals:* $[c^i, c^{i+1})$

A pattern in interval I is c-closed if the support of every superpattern is in another interval

Largely reduces lattice sizes & computation time, at the cost of c-approximate counts

IncMine: itemset mining in MOA

Closed itemset miners in data streams

- Exact: MOMENT [Chi+ 06], NEWMOMENT [Li+ 09], CLOSTREAM [Yen+ 11], ...

  High computational cost for exactness

- Approximate: IncMine [Cheng+ 08], CLAIM [Song+ 07], ...

  More efficient at the expense of false positives and/or negatives

The IncMine Algorithm [Cheng,Ke,Ng 08]

Some features:

- Keeps frequent closed itemsets in a sliding window
- Approximate algorithm, controlled by relaxation parameter
- Drops non-promising itemsets: may have false negatives

Chosen for implementation in MOA [Quadrana-Bifet-G 13&15]

Non-promising itemsets

- Assume window of last W transactions, min. support σ
- If t is σ-frequent in W, we expect σw occurrences in first w elements of window ($w < W$)
- (assuming no change)
- choose to drop it if much fewer occurrences
- more precisely, if less than $\sigma \cdot r(w)$, for $r(w) = r + (1-r)w/W$
- so that $r(0) = r$ and $r(W) = 1$

Erroneously dropped itemsets will be *false negatives*

Non-promising itemsets

- Inverted FCI index to keep updated itemsets within window
- Requires a batch method for finding FCI in new batch
- We chose CHARM [Zaki+ 02]

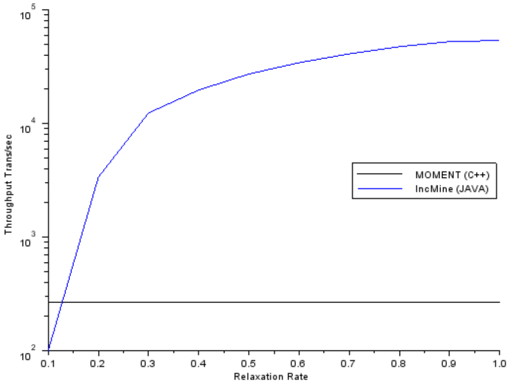
Experiments: Accuracy

Zaki's synthetic frequent itemset generator (standard in field)

100% precision (no false negatives)

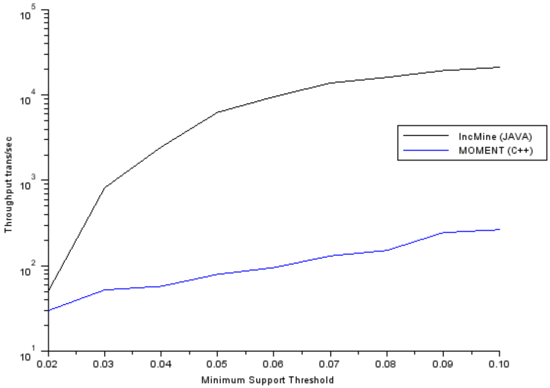
100% recall up to $r = 0.6$; down to 82% by $r = 0.8$

Experiments: Throughput



Transactions/second for different values of r ($\sigma = 0.1$). The minimum support used for MOMENT is equal to 500. Note the logarithmic scale in the y axis

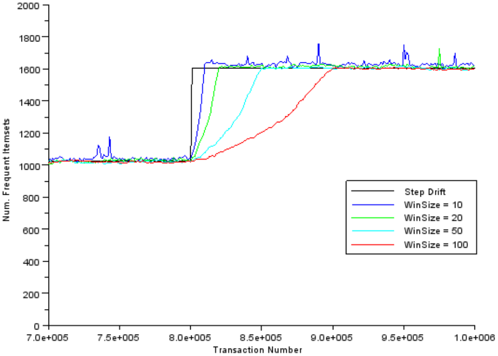
Experiments: Throughput



Transactions/second for different values of σ ($r = 0.5$). The minimum support used for MOMENT is equal to $\sigma \cdot 5000$. Note the logarithmic scale in the y axis

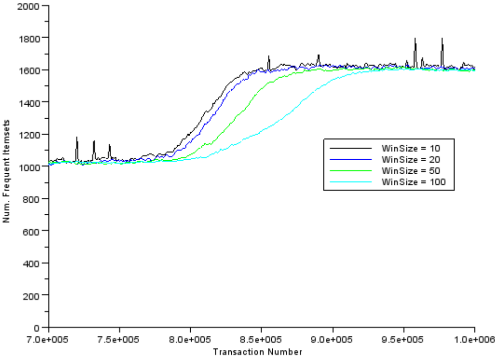
Reaction to Sudden Drift

`T40I10kD1MP6` drifts to `T50I10kD1MP6C05` dataset



Reaction time grows linearly with *window size*

Reaction to Gradual Drift



- *Fast reaction* with small windows
- *Stable response* with big windows

Analyzing MOVIELENS (I)

About 10 million ratings over 10681 movies by 71567 users

- Static data set for *movie rating* (from 29 Jan 1996 to 15 Aug 2007)
- Movies grouped by rating time (every 5 minutes)
- Transactions passed in ascending time to create a *stream*
- Stream of 620,000 transactions with average length 10.4

Results:

- Evolution of popular movies over time
- Unnoticed with static dataset analysis

Analyzing MOVIELENS (II)

date	Frequent Itemsets
Dec 2001	Lord of the Rings: The Fellowship of the Ring, The (2001); Beautiful Mind, A (2001).
	Harry Potter and the Sorcerer's Stone (2001); Lord of the Rings: The Fellowship of the Ring, The (2001).
Jul 2002	Spider-Man (2002); Star Wars: Episode II - Attack of the Clones (2002).
	Bourne Identity, The (2002); Minority Report (2002).
Dec 2002	Lord of the Rings: The Fellowship of the Ring, The (2001); Lord of the Rings: The Two Towers, The (2002).
	Minority Report (2002); Signs (2002).
Jul 2003	Lord of the Rings: The Fellowship of the Ring, The (2001); Lord of the Rings: The Two Towers, The (2002).
	Lord of the Rings: The Two Towers, The (2002); Pirates of the Caribbean: The Curse of the Black Pearl (2003).

Analysis

Model: t-th itemset draw independently from distribution D_t on set of all transactions

Theorem

Assume that $D_{t-W} = \cdots = D_{t-1} = D_t$, that is, no distribution change in the previous W time steps. Let O_t be the set of FCI output by IncMine(σ,r) at time t. Then, for every itemset X and every $\delta \in (0,1)$,

1. *if $\sigma(X, D_t) \leq (1-\varepsilon)\sigma$ then, with probability at least $1-\delta$, X is not in O_t.*
2. *if $\sigma(X, D_t) \geq (1+\varepsilon)\sigma$ then, with probability at least $1-\delta$, X is in O_t.*

provided $\varepsilon \geq f(W, B, \sigma, \delta)$ and $r \leq g(W, B, \sigma, \delta)$.

Bonus: Analysis reveals relaxation rate $r(.)$ in original paper is not optimal. Nonpromising sets can be dropped much earlier. And parameter r not needed

Conclusions

- Perfect integration with MOA
- Good accuracy and performance compared with MOMENT
- Good throughput and reasonable memory consumption
- Good adaptivity to concept drift
- Analyzable under common probabilistic assumptions
- Usable in real contexts