Statistics in Data Mining : Finding Frequent Patterns

Nele Dexters
Dept. Wiskunde-Informatica, ADReM
Room G3.23, Middelheimlaan 1, 2020 Antwerpen

Data Mining = Knowledge Discovery in Databases

Given: a large amount of computerized data

Goal: finding unexpected, interesting patterns → **Frequent Itemsets**, Association Rules, Correlations, Clusters, ...

Supermarket domain: market basket data

"Discover products that are often purchased together"

Practical use:
- Product pricing
- Store layout

Notations

$I = \{i_1, i_2, \ldots, i_m\}$, set of all items in the shop

$I_1, I_2, \ldots$ are itemsets = baskets

All the baskets are stored in a database

transactions

ID	Items
1	1, 2, 3
2	4, 5, 1
...	...

Count(X) = the number of transactions that contain all the items from a set X

$$\text{Support}(X) = \frac{\text{count}(X)}{\text{total number of transactions in the db}}$$

A set X is frequent if and only if

Support(X) ≥ k

with k a certain user-defined minimal support threshold

The Frequent Itemset Mining (FIM) Problem

Given:
- a database of basket data
- a user-defined minimal support threshold k

To do: Determine which sets of items are purchased by at least k shoppers

Solution: There exists lots of algorithms to solve the problem

- Apriori
- Eclat & FP-growth
- Max-miner
- ...

Extensive literature on the experimental analysis
Few papers are devoted to theoretical analysis

Problem 1: Detailed Probabilistic Study of the Performance of FIM Algorithms for Different Data Distributions

Problem 2: Frequent Itemset Mining for Data Sets with Tall Peaks → Mining for Large Frequent Itemsets

General Shopping Model

- identical shoppers
- independent shoppers
- random shoppers

→ Correlations between items are possible!

Candidate, Success and Failure Probabilities for Set *I*

A candidate is a set that is potentially frequent → $C(I)$

A success is a candidate that turns out to be frequent → $S(I)$
A failure is a candidate that turns out to be infrequent → $F(I)$

$$C(I) = S(I) + F(I)$$

Some Important Results:

$$S(I) = \sum_{j \geq k} \binom{b}{j} [P(I)]^j [1 - P(I)]^{b-j}$$

$$C(I) = \sum_{\substack{j_0 \\ j_1 \geq k - j_0 \\ j_2 \geq k - j_0 \\ \cdots \\ j_{|I|} \geq k - j_0}} \binom{b}{j_0, j_1, \ldots, j_{|I|}, b - j_0 - j_1 \cdots - j_{|I|}} \times [P(I)]^{j_0} \left[\prod_{1 \leq i \leq |I|} Q_i(I)^{j_i} \right] \left[1 - P(I) - \sum_{1 \leq i \leq |I|} Q_i(I) \right]^{b - j_0 - \sum_{1 \leq i \leq |I|} j_i}$$

Approximation of *S*(*I*)

$S(I) = \text{Prob}(X \geq k) = 1 - \text{Prob}(X < k)$ with $X \sim \text{Binom}(b, P(I))$

Two Shopper Two Item Type Shopping Model

Shoppers	Item Types	
	n_1	n_2
s_1	p_{11}	p_{12}
s_2	p_{21}	p_{22}

- s_i shoppers of type i (i=1,2)
- n_j items of type j (j=1,2)
- probability p_{ij} that a shopper of type I buys an item of type j

By convention, p_{11} is the largest of the probabilities

Important Probability Results:

- $P_i(m_1, m_2) = p_{i1}^{m_1} p_{i2}^{m_2}$
- $s_1 P_1(m_1, m_2) + s_2 P_2(m_1, m_2) = s_1 p_{11}^{m_1} p_{12}^{m_2} + s_2 p_{21}^{m_1} p_{22}^{m_2}$

Result: We develop the level-by-level peak-jumping family:

- the Perfect Jump Algorithm
- the Perfect Jump Algorithm with Lower Bounds
- the Perfect Jump Algorithm with Connected Components
- "Max-Miner"

based on the concept of perfect jumps

References

[1] **Dexters, N. (2005),** *Approximating the Probability of an Itemset being Frequent,* in Proceedings of 22nd British National Conference on Databases, pp 1—4.

[2] **Dexters, N., Purdom, P.W. and Van Gucht, D. (2006),** *A Probability Analysis for Candidate-Based Frequent Itemset Algorithms,* in Proc. of ACM Symp. on Applied Computing, Vol. 1 of 2, pp 541—545.

[3] **Dexters, N., Purdom, P.W. and Van Gucht, D. (2006),** *Peak-Jumping Frequent Itemset Mining Algorithms,* in Proc. of PKDD Int. Conf. On Principles of Data Mining and Knowledge Discovery, Publisher: Springer.

Universiteit Antwerpen

For more information:
http://www.adrem.ua.ac.be/~ndext