Relevance of Counting in Data Mining Tasks

Osmar R. Zaïane
http://www.cs.ualberta.ca/~zaiane/
zaiane@cs.ualberta.ca

The First International Conference on Advanced Data Mining and Applications
Wuhan, China, July 22-24, 2005
Database Selection Problem

Query Engine

Counting features

Query

Information Retrieval

Search Engine

Counting words

Search Query

TF-idf

\[ W_{i,j} = tf_{i,j} \times \log \left( \frac{N}{df_i} \right) \]

Measures importance of a word \( i \) within a document \( j \) over the general importance of the word in the collection.

Naïve/Full Bayesian Classifier

Bayesian Learning (Bayes Theorem)

Given a hypothesis \( H \) that some data belongs to a class \( C \) and some evidence \( E \) about the data, the posteriori probability of \( H \) given \( E \) is:

\[
P(H|E) = \frac{P(E|H)P(H)}{P(E)}
\]

Classification = prediction for discrete and nominal values

Learning = Calculating prior and posterior probabilities.

Discretization & Concept Hierarchies

- Handling continuous data
- Automatic Concept hierarchy building
-Numerosity Reduction
-Smoothing Noise

Discretization is used to reduce the number of values for a given continuous attribute, by dividing the range of the attribute into intervals.

Binning
Histogram analysis
Entropy-based
3-4-5 data segmentation

Statistics

Counting values

Counting occurrences
Genomics and Proteomics

Predicting function or location of a Protein

Investigate frequent sequences and subsequences

Counting motifs

Counting itemsets

Genomics and Proteomics

Predicting function or location of a Protein

Investigate frequent sequences and subsequences

Counting motifs

Counting itemsets

With only 5 possible items A, B, C, D, and E, there are 2^5 = 32 cases.

Given d items, there are 2^d possible candidate itemsets

Given d items, there are 2^d possible candidate itemsets

With 5 items we have 325 sub-sequences
Frequent Itemset Mining

Bound by a support threshold

Association Rules Generation

Bound by a confidence threshold

Frequent itemset generation is still computationally expensive

Apriori (Agrawal et al. 1994)

Repetitive I/O scans
Huge Computation to generate candidate items

Illustrating Apriori Principle

Latestubsets

Support=2, 13 patterns

Bottom - Up Example

Superset is candidate if ALL its subsets are frequent

18 candidates to check
Top - Down Example

Steps

TID | Items
---|---
1  | ABC
2  | ABCD
3  | ABC
4  | ACDE
5  | DE

Subset is candidate if it is marked or if one of its supersets is candidate

Support=2, \( \Rightarrow 13 \) patterns

23 candidates to check

Leap Traversal Example

Steps

TID | Items
---|---
1  | ABC
2  | ABCD
3  | ABC
4  | ACDE
5  | DE

Itemset is candidate if it is marked or if it is a subset of more than one infrequent marked superset

Support=2, \( \Rightarrow 13 \) patterns

5 frequent patterns without checking

10 candidates to check

Many Candidates – Many Patterns

Not only there are too many candidate itemsets but there are also too many frequent ones.

Frequent pattern \( \{a_1, \ldots, a_{100}\} \)
\( \Rightarrow (100^1) + (100^2) + \ldots + (100^{100}) \)
= \( 2^{100} - 1 \)
= \( 1.27 \times 10^{30} \) frequent sub-patterns!

Compressed Representation

Maximal frequent itemsets \( \subseteq \) Closed frequent itemsets \( \subseteq \) All frequent itemsets
Frequent Closed Patterns

- N. Pasquier et al. In ICDT’99

- For frequent itemset X, if there exists no item y such that every transaction containing X also contains y, then X is a frequent closed pattern.

- In other words, frequent itemset X is closed if there is no item y, not already in X, that always accompanies X in all transactions where X occurs.

- Concise representation of frequent patterns. Can generate all frequent patterns with their support from frequent closed ones.

- Reduce number of patterns and rules.

Frequent Maximal Patterns

- R. Bayardo. In SIGMOD’98

- Frequent itemset X is maximal if there is no other frequent itemset Y that is superset of X.

- In other words, there is no other frequent pattern that would include a maximal pattern.

- More concise representation of frequent patterns but the information about supports is lost.

- Can generate all frequent patterns from frequent maximal ones but without their respective support.

Maximal Versus Closed Patterns

- Closed but not maximal

- Closed and maximal

- Not supported by any transaction

- Minimum support = 2

- Infrequent

- Frequent

- Closed

- Maximal
Do We Need to Count All?

Closed frequent itemsets

Maximal frequent itemsets

All frequent itemsets

NO Support

What about Maximals?

Maximal frequent itemsets

All frequent itemsets

Support

The COFI Approach

COFI
(El-Hajj and Zaïane, 2003)

2 I/O scans
reduced candidacy generation
Small memory footprint

COFI- trees

FP-Tree

Patterns

COFI-MAX and the extra info

FP-Tree

COFI- trees

Patterns

(Zaïane and El-Hajj ACM SIGKDD 2005)
Ordered Partitioning Bases

What is this extra information?

- A data structure containing frequent pattern bases and their branch support;
- The data structure is a “free” bonus since it is used to mine for maximals;
- Frequent pattern bases are those marked sub-transactions in the leap-approach and their descendents if not frequent.
- The branch support is the number of times the frequent pattern base occurs alone (not subsumed by another pattern).

Counting the Support

Support of any pattern is the summation of the supports of its supersets of frequent-path-bases

Frequent pattern bases are marked sub-transactions in the leap-approach and their descendents if not frequent.

Some Results (synthetic data)

Some Other Selected Results

mining for MAXIMAL patterns in synthetic dataset

Mining MAXIMAL frequent patterns

pumsb (MAXIMAL with low support)
In Conclusion …

- Computers are machines that count and compute
- Many data mining tasks consist in counting
- The task of enumerating and counting is essential but not necessarily easy.
- We do not need to count all possibilities or even all patterns of direct interest
- The challenge is to reduce the enumeration without losing effectiveness (loss-less compression)
- There is no winner / no best way to count

My Students

- Maria-Luiza Antonie
- Jiyang Chen
- Mohammad El-Hajj
- Andrew Foss
- Yan Jin
- Yi Li
- Yaling Pei

Without my students this research work wouldn’t have been possible.

Osmar R. Zaïane, Ph.D.
Associate Professor
Department of Computing Science
221 Athabasca Hall
Telephone: Office +1 (780) 492 2860
Fax +1 (780) 492 1071
E-mail: zaiane@cs.ualberta.ca
http://www.cs.ualberta.ca/~zaiane/

http://www.cs.ualberta.ca/~zaiane/