APRIORI ALGORITHM

BY

International School of Engineering

We Are Applied Engineering

OVERVIEW

- DEFNITION OF APRIORI ALGORITHM
- KEY CONCEPTS
- STEPS TO PERFORM APRIORI ALGORITHM
- APRIORI ALGORITHM EXAMPLE
- MARKET BASKET ANALYSIS
- THE APRIORI ALGORITHM : PSEUDO CODE
- LIMITATIONS
- METHODS TO IMPROVE APRIORI'S EFFICIENCY
- APRIORI ADVANTAGES/DISADVANTAGES
- VIDEO OF APRIORI ALGORITHM

DEFINITION OF APRIORI ALGORITHM

- The Apriori Algorithm is an influential algorithm for mining frequent itemsets for boolean association rules.
- Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as *candidate generation*, and groups of candidates are tested against the data.
- Apriori is designed to operate on database containing transactions (for example, collections of items bought by customers, or details of a website frequentation).

KEY CONCEPTS

- Frequent Itemsets: All the sets which contain the item with the minimum support (denoted by L_i for i^{th} itemset).
- Apriori Property: Any subset of frequent itemset must be frequent.
- Join Operation: To find L_k, a set of candidate k-itemsets is generated by joining L_{k-1} with itself.

STEPS TO PERFORM APRIORI ALGORITHM

STEP 1
Scan the transaction data base to get the support of S each 1-itemset, compare S with min_sup, and get a support of 1-itemsets, L1

STEP 2
Use L_{k-1} join L_{k-1} to generate a set of candidate k-itemsets. And use Apriori property to prune the unfrequented k-itemsets from this set.

STEP 3
Scan the transaction database to get the support S of each candidate k-itemset in the find set, compare S with min_sup, and get a set of frequent k-itemsets L_k

STEP 4
The candidate set = Null

NO

YES

STEP 5
For each frequent itemset 1, generate all nonempty subsets of 1

STEP 6
For every nonempty subset s of 1, output the rule "s=>(1-s)" if confidence C of the rule "s=>(1-s)" (=support s of 1/support S of s)' min_conf

APRIORI ALGORITHM EXAMPLE

Market basket

MARKET BASKET ANALYSIS

- Provides insight into which products tend to be purchased together and which are most amenable to promotion.
- Actionable rules
- Trivial rules
  - People who buy chalk-piece also buy duster
- Inexplicable
  - People who buy mobile also buy bag

APRIORI ALGORITHM EXAMPLE

Database D

Minsup = 0.5

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

Scan D

C_2

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

L_2

itemset	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

C_3

itemset
{2 3 5}

Scan D

L_3

itemset	sup
{2 3 5}	2

The Apriori Algorithm : Pseudo Code

- Join Step: C_k is generated by joining L_{k-1} with itself
- Prune Step: Any (k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset
- Pseudo-code :C_k: Candidate itemset of size k

  L_k: frequent itemset of size k

L_1 = {frequent items};
for (k = 1; L_k !=∅; k++) **do begin**
  C_{k+1} = candidates generated from L_k;
  for each transaction t in database do
    increment the count of all candidates in C_{k+1}
    that are contained in t
  L_{k+1} = candidates in C_{k+1} with min_support
  end
return $\cup_k L_k$;

LIMITATIONS

- Apriori algorithm can be very slow and the bottleneck is candidate generation.

- For example, if the transaction DB has 10^4 frequent 1-itemsets, they will generate 10^7 candidate 2-itemsets even after employing the downward closure.

- To compute those with sup more than min sup, the database need to be scanned at every level. It needs (n +1) scans, where n is the length of the longest pattern.

METHODS TO IMPROVE APRIORI'S EFFICIENCY

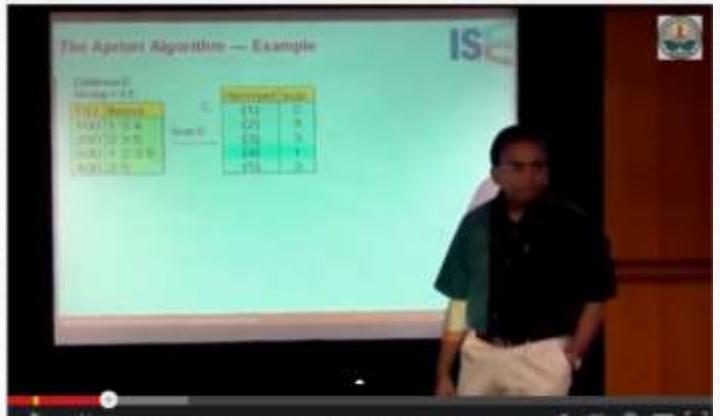
- Hash-based itemset counting: A k-itemset whose corresponding hashing bucket count is below the threshold cannot be frequent
- Transaction reduction: A transaction that does not contain any frequent k-itemset is useless in subsequent scans
- Partitioning: Any itemset that is potentially frequent in DB must be frequent in at least one of the partitions of DB.
- Sampling: mining on a subset of given data, lower support threshold + a method to determine the completeness
- Dynamic itemset counting: add new candidate itemsets only when all of their subsets are estimated to be frequent

APRIORI ADVANTAGES/DISADVANTAGES

- Advantages
  - Uses large itemset property
  - Easily parallelized
  - Easy to implement
- Disadvantages
  - Assumes transaction database is memory resident.
  - Requires many database scans

For Detailed Description of APRIORI ALGORITHM

Check out our video on



International School of Engineering

Plot no 63/A, 1st Floor, Road No 13, Film Nagar, Jubilee Hills, Hyderabad-500033

For Individuals (+91) 9502334561/62
For Corporates (+91) 9618 483 483

Facebook: www.facebook.com/insofe

Slide share: www.slideshare.net/INSOFE