

# Automatic web content organization based on user actions

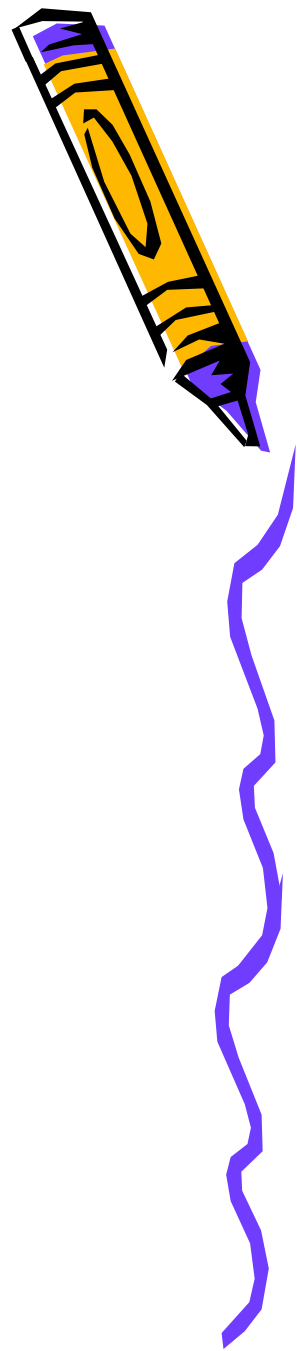
Supervisor: Morten Goodwin Olsen



Group member :  
Wenjie Li , Wenjuan Wang

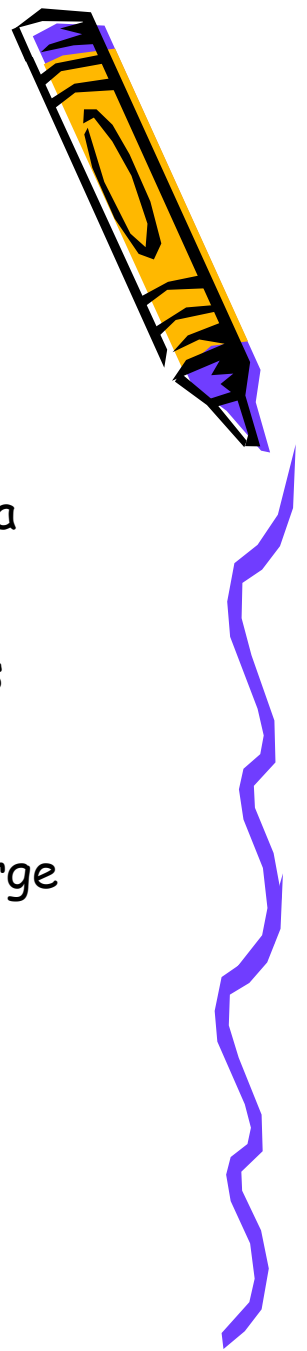
# Introduction

- Background
- Problem description
- assumption
- Solution
- Conclusion



# Background

- Data mining
  - the process of extracting information or knowledge from a data set for the purpose of decision making
  - supplies different algorithms according to different tasks
- Association rules
  - used for discovering interesting relationships hidden in large data sets
  - give an example

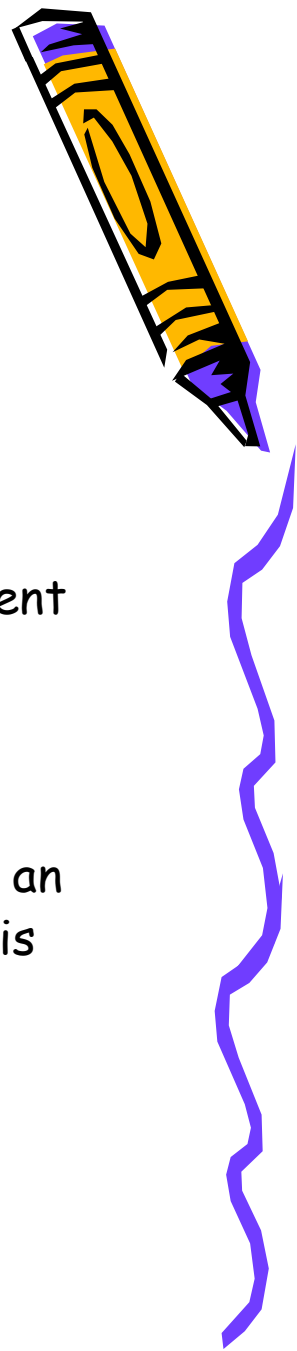


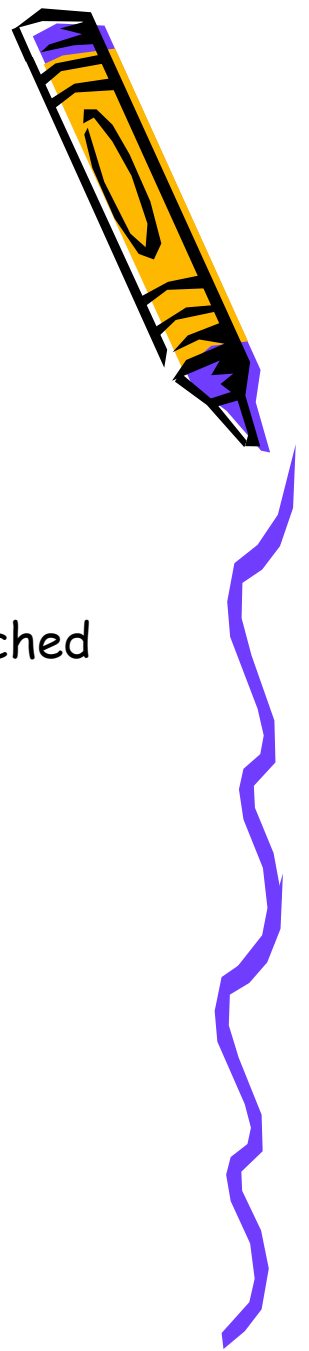
# • Apriori algorithm

- a classic algorithm
- two important characteristics
  - a level-wise algorithm
  - employs a generate-and-test strategy for finding frequent itemsets

# • FP-growth algorithm

- encodes the data set using a compact data structure called an FP-tree and extracts frequent itemsets directly from this structure





- Distributed system

a database that is under the control of a central database management system in which storage devices are not all attached to a common CPU



# Problem description

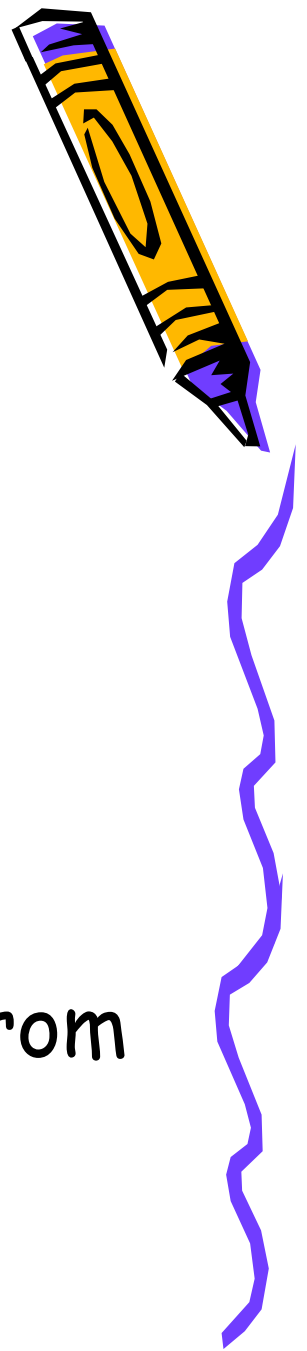
- Task

- automatically generate content based on users' actions as soon as possible
- decrease time-consuming by using large web portal servers
- search best method to generate association itemsets through the comparison of two algorithms



# Assumption

- Structure of web site  
hierarchical structure.
- Calculation style
  - offline
- Replication
  - combine input: collect records from  
all replicates



# Solutions



- Steps
- Step 1: Collect and formalize the input.
- Step 2: Pick out the frequent\_2\_itemset.
- Step 3: Rule generation.
- Step 4: Update to database.

Focus: algorithm in generation frequent itemsets.

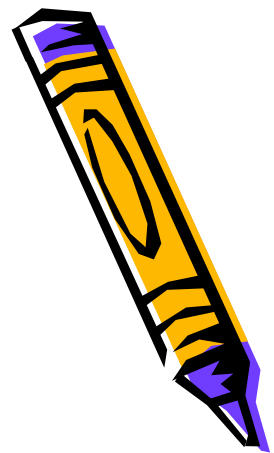


# apriori

1. Scan the data once.
2. Prune the infrequent item and generate frequent 1 item.
3. Apriori\_gen (generate candidates k Itemset)
4. Scan the data and count for candidates.
5. Redo step 3, until no frequent Itemset be found.



# myapriori



- Different Approach
  - create a bintree while scan the data.  
item and its support ,including indexset.
  - using the bintree while courting for candidates.

Candidates generation is a realy high cost.



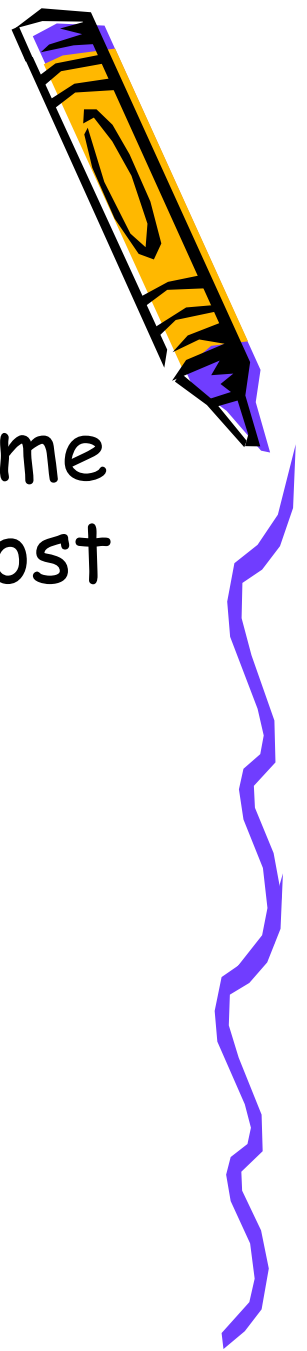
# Fp-growth

- It takes a different way to find the frequent k-itemset. It avoid the cost while generate the candidates. Instead, it encodes the input using a compact data structure called an FP-tree. Then pick out the frequent k-itemset from the structure directly.



# conclusion

- In a certain support the running time of my apriori and fp-growth is almost the same.
- The space cost of fp-growth algorithm is very big.



# Thank for you attention!

- Questions?

