CS412_Assignment_5

Ramsey EL Lethy

2025-04-23

(25 points) A data set shows 100 transactions in 5 days, each being summarized as a set of items associated with the number of transactions. Let the relative min sup = 0.5 and min conf = 0.7. date items bought number of transactions

Day 1 {p, a, b, c, m} 15 Day 2 {b, e, f, p} 35 Day 3 {p, a, c, k} 15 Day 4 {a, b, e, p} 15 Day 5 {p, a, g, e} 20

(a) (5 points) List the frequent 1-itemset associated with their absolute counts.

Itme P occurs in days {1,2,3,4,5} 5

Itme a occurs in days {1,3,4,5} 4

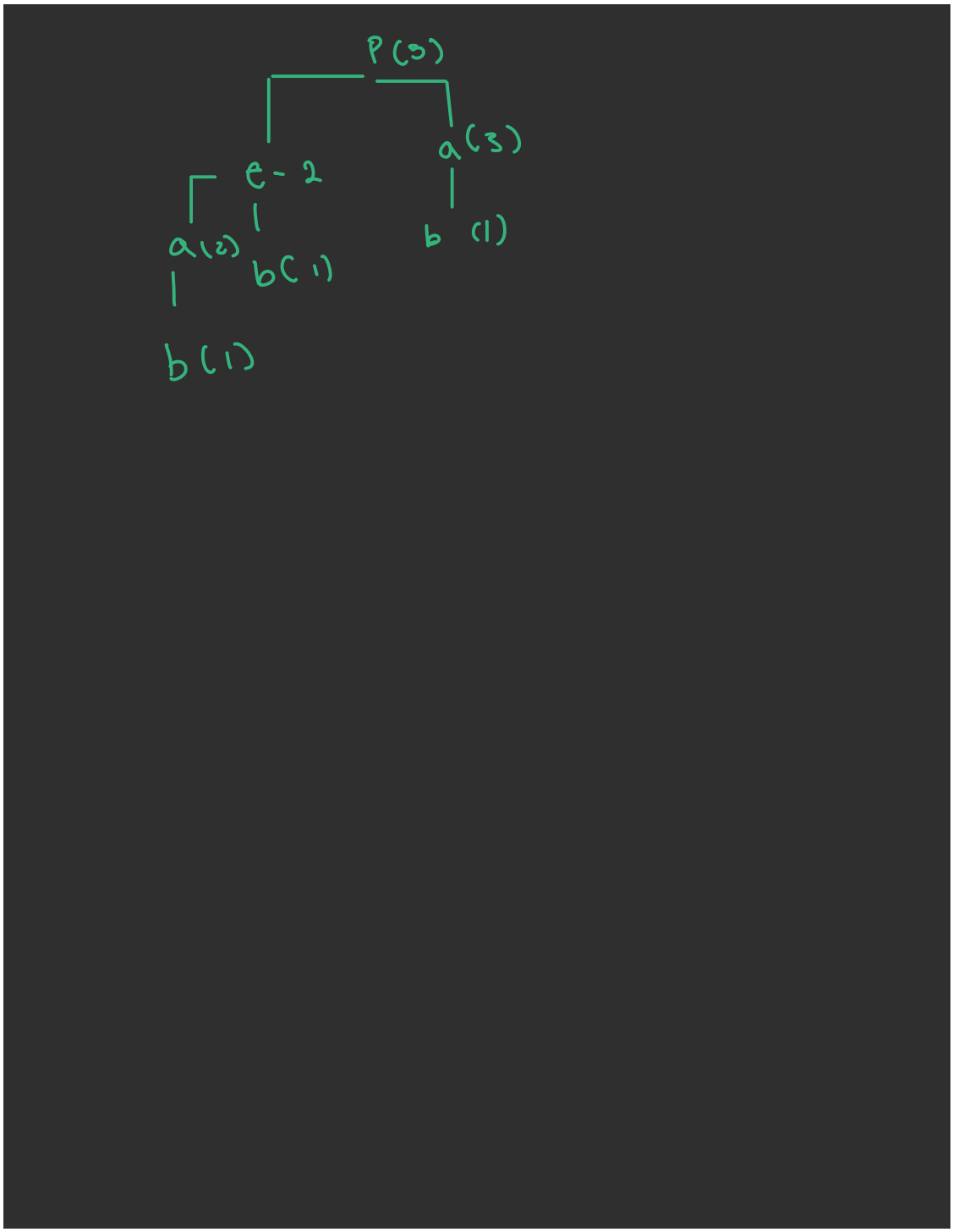
Item b occurs in days {1,2,4} 3

Item c occurs in days {1,3} 2

Item e occurs in days {2,4,5}3

The rest only occur in one day f - day 2, k - day 3, g - day 5, m - day 1 f k g and m are all equalt o 1

(b) (5 points) Draw a frequent pattern tree (FP-tree) for the dataset.



FPGrowth Tree

(c) (5 points) Present all the frequent k-itemsets for the largest k.

Apriori should do a good job at filtering frequent itemsets up to k = 3 if any.

```r
library(arules)

## Loading required package: Matrix

##
## Attaching package: 'arules'

## The following objects are masked from 'package:base':
##
##     abbreviate, write

transactions_list <- list(
  c("p", "a", "b"),
  c("p", "b", "e"),
  c("p", "a"),
  c("p", "a", "b", "e"),
  c("p", "a", "e")
)

transactions <- as(transactions_list, "transactions")

frequent_items <- apriori(transactions, parameter = list(support = 0.6,
target = "frequent itemsets", minlen = 1))

## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##          NA    0.1    1 none FALSE            TRUE       5     0.6      1
##  maxlen            target  ext
##      10 frequent itemsets TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 3
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[4 item(s), 5 transaction(s)] done [0.00s].
## sorting and recoding items ... [4 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 done [0.00s].
## sorting transactions ... done [0.00s].
## writing ... [7 set(s)] done [0.00s].
## creating S4 object  ... done [0.00s].

inspect(frequent_items)

##     items  support count
## [1] {b}    0.6     3
```

```
## [2] {e}    0.6     3
## [3] {a}    0.8     4
## [4] {p}    1.0     5
## [5] {b, p} 0.6     3
## [6] {e, p} 0.6     3
## [7] {a, p} 0.8     4
```

(d) (5 points) Present two strong association rules (with their support and confidence) containing the k items (for the largest k only). (Hint: An association rule will be represented by X → Y where X and Y are frequent itemsets. Here the total number of items in X and Y is k. X and Y do not share any item.)

p –> a

support = 4/5 = 0.8 confidence = 4/5 = 0.8

p –> a is a strong association rule

a –> p should be a strong association rule too because

support = 4/5 and a appears in 4/5 with confidence = 4/4 = 1.0

(e) (5 points) Given the following sequence database

Sequence ID Sequence T1 ⟨ab(ac)d(cf)⟩ T2 ⟨(ad)c(abc)(ae)⟩ T3 ⟨(ef)(ab)c(df)(cb)⟩ T4 ⟨ega(bf)cbc⟩

What is the projected database for prefix ⟨ab⟩ (absolute min sup = 2)?

- (ac)d(cf) - (c)(ae) - c(df)(cb) - f(cbc)

appears in 4 which is greater than 2

(25 points) Basics of Patterns

(a) (5 points) Given a transaction database T DB, we partition it into two parts, T DB1 and T DB2. If an itemset X is frequent in both T DB1 and T DB2 with respect to a minimum (relative) support threshold s, is it possible for X to be frequent in original T DB? Why?

suppose you had two partitions of TDB and minsupport is 0.5

TDB1 = 2 transactions only and the itemset X appears in both transactions, the frequency is 1

TDB2 = 100 transactions, and X occurs in 51 of them, the frequency = 0.51

so combined TDB has 102 transactions, 52 of then contain X the frequency should be 0.519

so TDB is frequent in the original TDB, this is because TDB is a combination of TDB1 and TDB2, the frequency should stay the same if not increase after combining partitions.

(b) (10 points) Suppose we have a T DB1 with the following transactions: T1 = {a10, a11, . . . , a20}, T2 = {a1, a2, . . . , a20}, T3 = {a1, a2, . . . , a25}

(i) (3 points) For T DB1, how many max pattern(s) do we have and what is(are) it(they) if minimum (absolute) support is 1?

T1 = {a10, a11, ..., a20} 11 items

T2 = {a1, ..., a20} 20 items

T3 = {a1, ..., a25} 25 items

(i) min supp = 1

we want any item that appears in at least 1 transaction

Every item from a1 to a25 appears in at least 1 transaction

Largest transaction = T3 = 25 items

1 max pattern

{a1, ..., a25}

(ii) (3 points) For T DB1, how many max pattern(s) do we have and what is(are) it(they) if minimum (absolute) support is 2?

a1–a9 2 times

a10–a20 3 times

a21–a25 1 time

So frequent items = a1–a20

T2: {a1–a20} T3: {a1–a25} → contains {a1–a20} but also some infrequent items T1: {a10–a20}

max pattern with only frequent items = {a1–a20}

(iii) (4 points) For T DB1, how many max pattern(s) do we have and what is(are) it(they) if minimum (absolute) support is 3?

the items have to appear in at least 3 transactions

a10–a20 is the largest itemset that appears in T1, T2, T3

1 max pattern

{a10, …, a20}

(c) (10 points) Suppose we have a T DB2 with the following transactions: T4 = {a1, a2, . . . , a10}, T5 = {a20, a21, . . . , a25}, T6 = {a1, a2, . . . , a25}

(i) (3 points) For T DB2, how many max pattern(s) do we have and what is(are) it(they) if minimum (absolute) support is 2?

Let's check frequency of items:

a1–a10 2 times

a11–a19 1 time

a20–a25 2 times

then frequent items {a1–a10, a20–a25}

max combinations

T4 {a1–a10} T5 {a20–a25} T6 all of them

so there are 2 max patterns that appear in 2 ore more transactions

{a1–a10}

{a20–a25}

(ii)(3 points) For T DB2, how many closed pattern(s) do we have and what is(are) it(they) if minimum (absolute) support is 1?

we need the frequent patterns that are not a subset of any other patterns

{a1–a10} 2

{a20–a25} 2

{a1–a25} 1

3 closed patterns

(iii)(4 points) For T DB2, how many closed pattern(s) do we have and what is(are) it(they) if minimum (absolute) support

a1–a10, a20–a25 are frequent and are the largest versions of their patterns

2 closed patterns

{a1–a10}, {a20–a25}

3. (25 points) Constraint-Based Pattern Mining.

##(a) (10 points) Given a collection of constraints (based on the Product table), identify their types (monotone, anti-monotone, data anti-monotone, succinct, convertible). If multiple types coexist in the following constraints, please list them all.

Item Price Profit A 10 4 B 16 8 C 46 20 D 40 0 E 37 12 F 30 -10 G 45 -5

i. (3 points) Total price of all purchased items is less than $250.

This is anti montone because any larger sets of a set that violates this will also violate the constraint

this is also convertible anti monotone if we rank items ### ii. (3 points) Total price of all purchased items is at least $200.

monotone because any larger set will also satisfy the constraint

this is also convertible monotone for the same condition as i

iii. (4 points) The minimal price of the purchased item is less than $50.

this is data anti-monotone because adding more items will only increase the minimum

its anti monotone too because any superset wont satisfy either

this is convertible-antimonotone for the same condition as i

(b) (10 points) Constraints are important pieces of information that may speed up frequent itemset mining.

i. (3 points) What is a convertible constraint?

if we can order our items properly, then we can convert a monotonic or anti monotonic constraint into its counter part.

ii. (3 points) Give an example of a convertible constraint and explain how it can be pushed deep into the frequent itemset mining process to speed up mining.

if we say something like the average price of the itemset is at least 100 dollars

this is convertible monotone because if we sort items by increasing price and prune data with lower prices, we can say that

once the itemset hits greater than 100 on average, any addition will satisfy the constraint too

iii. (4 points) Can your above constraint be pushed deep into the mining using the Apriori algorithm or it is confined to the pattern growth process (such as FP-Growth)? Briefly present your reasoning.

if we were to add this constraint into apriori, we'd have to make the check everytime we generate candidates, but if we did it in FPGrowth, we'd need to only make one pruning round to clear out all items that would not contribute to an average of 100 or greater

so this would be better imlpemented in FP-growth

(c) (5 points) Let A be an itemset (the pattern we are considering) and V be a fixed bigger itemset. Is A ⊆ V anti-monotone? If true, please explain the reason. Otherwise, please provide counterexample.

this is anti-monotone because for A to be a subset of V, all items of A need to be in V, if this isnt the case, and we add items to A (make a superset), that previous item that violated the constraint still exists in A, Thus the constraint is still violated.

(25 points) Null Invariance

(a) (10 points) Suppose the following table shows two transaction data sets, S1 and S2, with the following count distributions after mining frequent 2-itemsets, where mc means that the transaction contains muffin but not cheese.

Data Set mc mc mc mc S1 900 100 100 900 S2 100 900 1,000 100,000

i. (5 points) For each data set S1 and S2, compute support, confidence and lift for the following rules:

all transactions: 900 + 100 + 100 + 900 = 2,000 s(m) = (900 + 100)/2000 = 0.5 s(c) = (900 + 100)/2000 = 0.5 s(m ∩ c) = 900/2000 = 0.45

m → c confidence = s(m ∩ c) / s(m) = 0.45/0.5 = 0.9

lift 0.45 / (0.5 * 0.5) = 1.8

c → m confidence 0.45 / 0.5 = 0.9 lift = 1.8

ii. (5 points) Is lift a good correlation measure here? Why or why not? (Give one-line brief reasoning)

we've got some problems specifically in S2 where there are too many null transactions, this is a key indicator that the correlation is probably distorted, so we'd need something a bit better to capture the relationships in this datasets.

(b) (5 points) The definitions of two measures on frequent patterns, lift and cosine, look rather similar as shown below,

lift(A, B) = s(A ∪ B) / s(A) × s(B) cosine(A, B) = s(A ∪ B) . sqrt(s(A) × s(B))

where s(A) is the relative support of itemset A. Explain why one of these two measures is nullinvariant but the other is not.

for lift, the denominator shrinks when there are a lot of null invariants, this will make lift look a lot bigger than the true relationship

cosine similarity isn't distorted by nulls, its bounded by 1, like a normal dot product, so it will capture the relationship regardless of nulls

(c) (5 points) We have learned at least three correlation measures: (1) lift, (2) χ 2 , and (3) Kulczynski. Give one example for Kulczynski measure that it is the most appropriate measure among the three and explain why.

kulc is a better measure thank lift or chi-squared in a dataset that contains an extremly high number of null transactions. kulc is null invariant, while lift and x^2 are not, and may become inflated.

(d) (5 points) Explain: If null transactions are predominant in large datasets, Kulczynski and Imbalance Ratio are usually used together to measure the interestingness of a pattern.

kulc does a good job at finding out the direction of relationship between two items, but not the magnitude imbalance ratio comes in to fill in the issues kulc has with finding out if the connection is trustworthy.