Frequent Pattern Mining

Apriori

Load required libraries

```
# Load required libraries

library(arules)

library(arulesViz)

library(ggplot2)
```

Load Data

- # Read lines

  lines <- readLines("Transactionlist2.csv")

  lines <- lines[-1] # remove header# Remove TID

  items_list <- sub("^[0-9]+,", "", lines)

- # Split by comma

  transactions_list <- strsplit(items_list, ",")

- # Trim whitespace from each item

  transactions_list <- lapply(transactions_list, trimws)

Convert to transactions object

- # Convert to transactions object
- transactions <- as(transactions_list, "transactions")
- # Inspect
- inspect(transactions)

Overview of the dataset

- # iii. Overview of the dataset
- num_transactions <- length(transactions)
- num_items <- length(itemLabels(transactions))
- itemLabels(transactions)
- sparsity <- 1 - (sum(transactions@data) / (num_transactions * num_items))
- sparse_matrix <- as(transactions, "matrix")
- co_occurrence <- crossTable(transactions, measure = "count")

Extract the diagonal counts

- # Extract the diagonal counts (support of each item)
- diag_counts <- diag(co_occurrence)
- # Sort item names by diagonal counts in descending order
- sorted_items <- names(sort(diag_counts, decreasing = TRUE))
- # Reorder the cross table by sorted items
- ct_sorted <- co_occurrence[sorted_items, sorted_items]
- # Print the sorted cross table
- print(ct_sorted)

Frequency Plot

- cat("Number of transactions:", num_transactions, "\n")
- cat("Number of items:", num_items, "\n")
- cat("Sparsity:", round(sparsity * 100, 2), "%\n")
- # iv. First 5 transactions
- inspect(transactions[1:5])
- # v. Most frequent items purchased
- itemFrequencyPlot(transactions, topN = 5, type = "absolute", main = "Top 5 Most Frequent Items")

Preprocessing

- # i. Frequency of items
- item_freq <- itemFrequency(transactions)
- head(sort(item_freq, decreasing = TRUE), 5)
- # ii. Plot items with support >= 0.3
- itemFrequencyPlot(transactions, support = 0.3, cex.names = 0.8,
- main = "Items with Support >= 0.3",col = "skyblue")

Association Rule Mining

- # i. Apply Apriori algorithm
- rules <- apriori(transactions,
- parameter = list(supp = 0.2, conf = 0.6))
- # ii. Summary of generated rules
- summary(rules)

Explanation of metrics

- # Explanation of metrics
- cat("Support: Proportion of transactions containing the items in the rule.\n")
- cat("Confidence: Likelihood that RHS is purchased when LHS is purchased.\n")
- cat("Lift: Strength of rule over random chance; >1 indicates positive association.\n")

Sort rules

- # iii. Sort rules by lift and display top 5
- rules_sorted <- sort(rules, by = "lift", decreasing = TRUE)
- inspect(rules_sorted[1:min(5, n)])

$$A \Rightarrow B$$

$$\text{Lift}(A \Rightarrow B) = \frac{\text{Confidence}(A \Rightarrow B)}{\text{Support}(B)} = \frac{P(B|A)}{P(B)}$$

d) Rule Interpretation and Visualization

- d) Rule Interpretation and Visualization
- # i. Interpret top 3 rules
- inspect(rules_sorted[1:3])
- cat("\nInterpretation:\n")
- cat("Rule 1: If customers buy X, they are highly likely to also buy Y.\n")
- cat("Rule 2: Strong association between items A and B.\n")
- cat("Rule 3: Buying C often leads to buying D.\n")

Visualize association rules

- # ii. Visualize association rules (scatter plot)
- plot(rules, method = "scatterplot", measure = c("support", "lift"),
- shading = "confidence")
- e) Business Insights and Recommendations
- cat("Expanded Insights & Recommendations:\n")

Expanded Insights & Recommendations

- cat("Expanded Insights & Recommendations:\n")
- cat("1. Product Bundling Opportunities: Combine items frequently bought together (e.g., Milk, Bread, Butter) into promotions.\n")
- cat("2. Cross-Selling Recommendations: Suggest related products at checkout (e.g., offer Bread to Egg buyers).\n")
- cat("3. Store Layout Optimization: Place high-association items close to each other.\n")
- cat("4. Time-based Promotions: Target weekend shoppers with breakfast-related discounts if rules show stronger weekend associations.\n")
- cat("5. Inventory Planning: Ensure linked products are stocked together to avoid lost sales.\n")
- cat("6. Targeted Loyalty Rewards: Give coupons for items a customer often misses in a frequent combo.\n")