# CSE521 DATA MINING TEST 2 SUMMER 2021
## (70pts )

**TEAM NUMBER:** Leader **NAME** and **ID:**

Team members Leader **NAMES** and **IDs:**

**FORMAT OF SUBMISSION**

1. The TEAM LEADER submits **ONE PDF** file for the TEAM
2. Name your PDF file as < *TEAMLEADERID* >.pdf

**Example:** 11384578.pdf

**All team members receive the same grade.**

To make your **ONE PDF** submission file you proceed as follows

YOU CAN use a sheet of PAPER and HAND-WRITE YOUR SOLUTIONS clearly indicating which problem are you solving (you have to hand write the problem statement and points assigned)

Then you can either **scan** or take **pictures** of the pages and **make ONE PDF file** with multiple pages .

**or** YOU CAN PRINT the TEST (or use the .pdf editor) and write your answers it in the spaces provided;

Then you **make ONE PDF file** with multiple pages.

**or** YOU CAN use word / latex to write your answers. In this case, you answers should CLEARLY INDICATE the problem/question (you have to type the problem statement and points assigned)

Then you can either **scan** or take **pictures** of the pages and **make ONE PDF file** with multiple pages.

**UNIVERSITY Honesty and Behavioral Expectation**

Please read carefully and sign each statement below

1. I certify that the calculations/data/answers in this exam were generated independently, using only the tools and resources defined in the course and that I did not receive any external help, coaching, or contributions during the production of this work.

**Names:** VENKATA RAVI TEJA TAKKELLA, NAGIREDDY SUSMITHA REDDY, KRISHNA SHASHANK GORANTA

2. I understand the universitys academic integrity and discipline policies and promise to uphold them.

**Names:** VENKATA RAVI TEJA TAKKELLA, NAGIREDDY SUSMITHA REDDY, KRISHNA SHASHANK GORANTA

3. I understand that the instructor may use tools to check for plagiarism and cheating.

**Names:** VENKATA RAVI TEJA TAKKELLA, NAGIREDDY SUSMITHA REDDY, KRISHNA SHASHANK GORANTA

**ACADEMIC INTEGRITY - UNIVERSITY STATEMENT**

Academic integrity is expected of all students at all times, whether in the presence or absence of members of the faculty. Understanding this, I declare that I shall not give, use, or receive unauthorized aid in this examination. I have been warned that any suspected instance of academic dishonesty will be reported to the appropriate office and that I will be subjected to the maximum possible penalty permitted under University Guidelines

**PART 1: SHORT QUESTIONS** TOTAL **15pts**

**QUESTION 1 (3pts)** Describe shortly Apriori Analysis; type of data, goals and types of applications

**Data:**

**Goals:**

**Main Applications:**

# PART-1

PROBLEM 1

Q1) Describe shortly Apriori Analysis; type of data, goals and types of applications.

Answer:

(i) Data:

Apriori Analysis is designed to operate on databases containing "transactions" i.e. "Transactional Database" is used here. Each transaction is seen as a list of items (an itemset). If a relational database is used, then it must be transformed into set of transactions.

(ii) Goals:

The goal of Apriori Analysis is as follows:

→ To find the set of "frequent items" i.e. the set of items that have minimum support. "Apriori Algorithm" is used to do this process.

→ To find the set of "strong association rules" (if required). We use the frequent itemsets to generate associate rules.

(iii) Main Applications:

The following are the applications of Aprion Analysis;

↳ It is important for effective Market Basket Analysis and it helps the customers in purchasing their items with more ease, which increases sales of the markets.

↳ Useful in finding and analysing "buying patterns".

↳ Useful is improving "shelving patterns" in big stores.

↳ Useful to improve "target marketing" by connecting clients data with their buying patterns.

↳ Useful in medical field, for example "Analysis of patients database"

↳ Used in many big companies like Google for "auto-complete feature" and by Amazon for "Recommender system".

**QUESTION 2 (2pts)** Describe the Goal and two main steps of Apriori PROCESS

**Goal**

**Steps1**

**Step 2**

PART 1:

PROBLEM-2 Describe the Goal & two main steps of Apriori Process.

Answer:

(i) Goal:

The main goal of Apriori Process is to find "Association Rules." Given a database of transactions, where each transaction is a list of items, we find all rules that associate the presence of one set of items with that of another set of items. In Apriori process, we form the association rules ($X \Rightarrow Y$) from the frequent item sets. An association rule, $X \Rightarrow Y$ will be of the form "for a set of transactions, some value of itemset X determines the values of itemset Y under the condition in which minimum support and confidence are met."

## (ii) Steps of Apriori Process:

Two main steps are as follows –

**Step 1:** We use Apriori Algorithm to find the set of frequent itemsets defined by user fixed support count.

We follow below steps for this phase :–

* Count the occurences of items in data base D
* Fix the minimum support value (usually low)
* Calculate frequent 1-item sets, 2-item sets until k-item sets. We stop this iteration when there are no more frequent item steps. We use Apriori Algorithm to generate frequent item sets.

This is the end of "Apriori - Algorithm" phase.

**Step 2:** We fix the ~~min~~ minimum confidence parameter (usually high) and generate set of strong association rules (rules with support > min support & confidence > min confidence)

This is the end of "rules-generation" phase.

Apriori Process ends after successfully completing both the phases.

**QUESTION 3 (5pts)** Give definitions and examples of Single-dimensional and Multi-dimensional Association Rules and describe their applications

1. **Single-dimensional**

**Definition:**

**Example:**

2. **Inter-dimensional**

**Definition:**

**Example:**

3. **Hybrid-dimensional**

**Definition:**

**Example:**

Part - 1.

Question - 3.

## Single - dimensional Association rules.

Definition: The association rules in which only one predicate (or dimension) is used, are called single - dimensional Association rules.

Example: (i) In store application "buys".

buys(x, "milk") → buys (x, "bread")

## Multi - dimensional Association rules.

In Multi - dimensional association rules, two or more predicates (or dimensions) are used. These types of rules can be classified further to: Inter - dimensional association rules and Hybrid - dimensional association rules.

## Inter - dimensional Association rules.

Definiton: The association rules in which two or

more predicates or dimensions are used, and if the predicates donot repeat then these rules are called as Inter-dimensional Association rules.

Example:

(i) $occupation(x, \text{"student"}) \wedge age(x, \text{"22-28"}) \rightarrow buys(x, \text{"computer"})$.

(ii) $age(x, \text{"young"}) \wedge income(x, \text{"high"}) \rightarrow buys(x, \text{"coke"})$

## Hybrid-dimensional Association rules.

Definition: The association rules in which two or more predicates or dimensions are used, and if the predicates repeat then these rules are called as Hybrid-dimensional Association rules.

Example:

(i) $buys(x, \text{"milk"}) \wedge age(x, \text{"old"}) \rightarrow buys(x, \text{"bread"})$

(ii) $age(x, \text{"30-35"}) \wedge buys(x, \text{"ipad"}) \rightarrow buys(x, \text{"apple pencil"})$

## Applications of Association rules

Association rule mining can be used for Cross-marketing, clustering, classification, Basket data analysis, catalog design, loss-leader analysis etc.

Below are the fields in which Association rules can be effectively applied:

- Entertainment: Services like Spotify and Netflix can use association rules to fuel their content recommendation engines.

  age (x, "young") ∧ liked (x, "Science Fiction")
  → recommend (x, "Interstellar").

- Retail: Retailers can collect data about purchasing patterns. Using these past purchasing patterns a retailer can adjust marketing and sales strategy.

  age (x, "young") ∧ buys (x, "computer") → ?
  (What other products should the store stock up?)

- Medicine: Doctors can use association rules to help

diagnose patients. Many diseases share symptoms. By using association rules, doctors can determine the conditional probability of a given illness by comparing symptom relationships in the data from past cases.

**QUESTION 4 (5pts)** Describe Genetic Algorithms basic encoding methods and basic operators.

**Encoding Methods**

**1.**

**2.**

PART-1

QUESTION-4

# ENCODING METHODS

Encoding is a process of representing the solution in the form of a string that is able to convey the necessary information. There are various encoding methods discussed

(1) Binary Encoding : This is one of the most common methods of encoding. Here the chromosomes are strings of 1s & 0s and each position denotes a particular characteristic of the problem.

Ex: 1001001110 01

(2) Permutation Encoding : This is useful in ordering problems. Where each object can be represented as a task to be performed. Used in Travelling Salesman Problem where each number represents a city that needs to be visited.

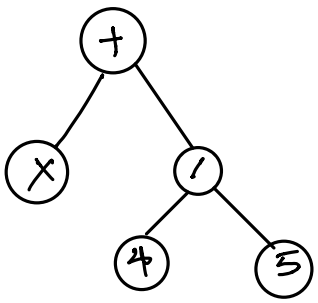
Ex: 1 4 5 3 2 9 8

(3) Value Encoding: This the method useful when more complex values are required to represent a characteristic such as real numbers.

Ex: 1.235, 2.45, 3.96

(4) Tree Encoding: In this method each chromosome is a tree of some objects, such as arithmetic operators for example.

Ex:




## OPERATORS

The basic operators that are used in a Genetic Algorithm are:

(1) Initialization: Individual solutions are generated randomly to cover the entire range of possible solutions.

(2) Selection : This is the operator where we use different techniques to select the individuals that can be copied over to the next generation like Roulette-Wheel selection, Tournament selection e.t.c.

Basically a Fitness function is used to quantify the optimality of a solution & rank it accordingly along with others.

(3) Recombination : This is the operation where we decide on which solutions need to be preserved & which need to be died out.

(4) Reproduction : This is the operation where we use operators like crossover, Mutation & Elitism. This is the method where new chromosomes are generated from the existing chromosomes.

(5) Termination : This is the operation where we terminate the genetic algorithm if a solution is found of the required criteria or fixed number of generations is reached, manual inspection. e.t.c.

## PART 2: PROBLEMS · (55 pts )

### PROBLEM 1 CLASSIFICATION BT ASSOCIATION (15pts)

**1.** 1. Use the TRAIN data to find the set of classification rules by the use of Apriori Algorithm.

Write down CAREFULLY all the steps of the PROCESS with explanations.

Do not need to compute confidence, i.e. your rules do not need to be strong. Fix Min Support= 3

Write Rules in predcate For.

**2.** Test the Rules with the TEST data

**TRAIN data**

| O | a1 | a2 | C |
|---|---|---|---|
| o1 | 1 | 1 | 1 |
| o2 | 0 | 0 | 0 |
| o3 | 0 | 1 | 0 |
| o4 | 0 | 0 | 0 |
| o5 | 1 | 1 | 1 |
| o6 | 1 | 1 | 0 |
| o7 | 0 | 0 | 0 |
| o8 | 1 | 0 | 1 |

**TEST data**

| O | a1 | a2 | C |
|---|---|---|---|
| o1 | 1 | 1 | 1 |
| o2 | 1 | 0 | 0 |
| o3 | 0 | 0 | 1 |
| o4 | 0 | 0 | 0 |

# PART-2

## PROBLEM-1: CLASSIFICATION BY ASSOCIATION:

1) Given: Fix Min support = 3.

TRAIN data:

| O | $a_1$ | $a_2$ | C |
|---|---|---|---|
| $O_1$ | 1 | 1 | 1 |
| $O_2$ | 0 | 0 | 0 |
| $O_3$ | 0 | 1 | 0 |
| $O_4$ | 0 | 0 | 0 |
| $O_5$ | 1 | 1 | 1 |
| $O_6$ | 1 | 1 | 0 |
| $O_7$ | 0 | 0 | 0 |
| $O_8$ | 1 | 0 | 1 |

Answer: Transactional Data & Supporting Calculations:

| | $I_1(a_1=0)$ | $I_2(a_1=1)$ | $I_3(a_2=0)$ | $I_4(a_2=1)$ | $I_5(C=0)$ | $I_6(C=1)$ |
|---|---|---|---|---|---|---|
| 1 | − | + | − | + | − | + |
| 2 | + | − | + | − | + | − |
| 3 | + | − | − | + | + | − |
| 4 | + | − | + | − | + | − |
| 5 | − | + | − | + | − | + |
| 6 | − | + | − | + | + | − |
| 7 | + | − | + | − | + | − |
| 8 | − | + | + | − | − | + |
| Count | 4 | 4 | 4 | 4 | 5 | 3 |

**Step 1**: Generating 1-itemset Frequent Pattern.

given min support count = 3.

L1;

(Frequent 1 itemset)

| Itemset | Support count |
|---|---|
| $I_1$ | 4 |
| $I_2$ | 4 |
| $I_3$ | 4 |
| $I_4$ | 4 |
| $I_5$ | 5 |
| $I_6$ | 3 |

The set of frequent 1-itemsets, L1, consists of the candidate 1-itemsets satisfying minimum support count.

**Step 2**: Generating 2-itemset frequent Pattern.

→ To discover the set of frequent 2-itemsets, $L_2$, the algorithm uses "$L_1$ Join $L_1$" to generate a candidate set of 2-itemsets, $C_2$ with support count

→ 2-itemsets, $L_2$ is then determined consisting of those candidate 2-items satisfying minimum support count.

$C_2$ :
(Candidate two itemsets)

| Itemset | Support Count |
|---|---|
| 1, 2 | 0 |
| 1, 3 | 3 |
| 1, 4 | 1 |
| 1, 5 | 4 |
| 1, 6 | 0 |
| 2, 3 | 1 |
| 2, 4 | 3 |
| 2, 5 | 1 |
| 2, 6 | 3 |
| 3, 4 | 0 |
| 3, 5 | 3 |
| 3, 6 | 1 |
| 4, 5 | 2 |
| 4, 6 | 2 |
| 5, 6 | 0 |

$L_2$ :
(frequent 2 item set)

| Item set | support count |
|---|---|
| 1, 3 | 3 |
| 1, 5 | 4 |
| 2, 4 | 3 |
| 2, 6 | 3 |
| 3, 5 | 3 |

## Step 3: Generating 3-itemset Frequent Pattern.

* In order to generate $C_3$, we first compute $L_2$ Join $L_2$.

(Point 1) Here we use prune step to reduce size of $C_3$. By using property of "Apriori", we reduce the size of $C_3$.

$C_3$ : (Without Prune step) (Candidate 3 itemset)

| Item set | support count |
|---|---|
| 1, 3, 5 | 3 |
| 2, 4, 6 | 2 |

↓ (After Prune step)

| Item set | Support Count |
|---|---|
| 1, 3, 5) | 3 |

$L_3$ :- (Frequent 3 itemset)

| Item set | Support Count |
|---|---|
| 1, 3, 5 | 3 |

* 3-itemset $L_3$, is determined consisting of those candidate 3-items satisfying minimum support count.

Further explaination w.r.t point 1 :-

↳ As mentioned above, to find $C_3$, we first compute

$L_2$ Join $L_2$.

$C_3 = L_2$ Join $L_2 = \{\{1, 3, 5\}, \{2, 4, 6\}\}$.

↳ Now Join step is complete.

↳ We need to check for Prune-step and check if there is any thing that needs to be removed,

↳ "Apriori property" says that all subsets of a frequent itemset must also be frequent.

↳ Take $\{1, 3, 5\}$

* The 2-item sets of it are $\{1, 3\}$, $\{1, 5\}$, $\{3, 5\}$

All of them are members of $L_2$

So we keep $\{1, 3, 5\}$ in $C_3$

↳ Take $\{2, 4, 6\}$.

* The 2-item sets of it are $\{2, 4\}$, $\{4, 6\}$, $\{2, 6\}$

But here $\{4, 6\}$ is not the member of $L_2$ and hence it is not frequent → violating Apriori property.

Thus we remove $\{2, 4, 6\}$ from $C_3$.

∴ Reduced $C_3$ (Candidate 3 item) is $\{1, 3, 5\}$.

↳ We now use this to determine $L_3$ (Candidate 3 items satisfying minimum support count.

We get $L_3 = \{1, 3, 5\}$.

We end the algorithm here as $C_4$ is a null set.

## Step 4: Generating Association Rules from Frequent Itemsets

→ When generating classification by association rules, we take association rules of the form $(p_1 \wedge p_2 \wedge \dots p_k) \rightarrow class = c$

→ In our case, class is either $I_5$ or $I_6$

The non-empty subsets needed to create association rules

$\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\},$

$\{1,3\}, \{1,5\}, \{2,4\}, \{2,6\}, \{3,5\}, \{1,3,5\}$.

↳ To create classification rules, we consider only subsets that contain class $I_5$ (or) $I_6$.

∴ Frequency set needed to form ClassificationRules is:

$$L = \{\{1,5\}, \{2,6\}, \{3,5\}, \{1,3,5\}\}$$

Short-hand Representation of Rules :-

① $1 \rightarrow 5$

② $3 \rightarrow 5$

③ $1 \nrightarrow 3 \rightarrow 5$

④ $2 \rightarrow 5$

## Predicate form Representation of Rules:-

Rule 1: $a_1(x,0) \rightarrow c(x,0)$

Rule 2: $a_2(x,0) \rightarrow c(x,0)$

Rule 3: $a_1(x,0) \wedge a_2(x,0) \rightarrow c(x,0)$

Rule 4: $a_1(x,1) \rightarrow c(x,1)$

2) Testing:

| | $a_1$ | $a_2$ | C [Test Data class] | Assigned class as per Rules | Correctly satisfied. |
|---|---|---|---|---|---|
| $O_1$ | 1 | 1 | 1 | 1 | Yes |
| $O_2$ | 1 | 0 | 0 | ? | No |
| $O_3$ | 0 | 0 | 1 | 0 | No |
| $O_4$ | 0 | 0 | 0 | 0 | Yes |

Record $O_1$: $a_1(x,1) \wedge a_2(x,1) \rightarrow c(x,1)$

from Rule 4, this is True

Record $O_2$: $a_1(x,1) \wedge a_2(x,0) \rightarrow c(x,0)$

As per Rule 4, c should be 1

As per Rule 2, c should be 0

Nothing can be said/concluded about this assigned class value. It is not satisfied since final class value is ambiguous.

Record $O_3$: $a_1(x,0)$ & $a_2(x,0) \rightarrow c(x,1)$

As per Rule 3, c should be 0.

∴ This is wrong / not satisfied.

Record $O_4$: $a_1(x,0)$ & $a_2(x,0) \rightarrow c(x,0)$

As per Rule 3, This is satisfied.

Predictive Accuracy $= \frac{2}{4} \times 100$

$= 50\%$.

**PROBLEM 2 MULTI- DIMENSIONAL ASSOCIATION RULES (20pts)**

Consider a data base **F** of a food store transaction

$$\mathbf{F} = \{ F1,\ F2,\ F3,\ F4,\ F5 \}$$

where

$F1 = \{Bread, Endive\}$, $F2 = \{Bread, Milk\}$, $F3 = \{Bread, Milk, Endive\}$, $F4 = \{Bread, Endive\}$, $F5 = \{Endive\}$.

All transactions were processed with the use of the Store Cards and we have the following stored information (simplified) about the

shoppers associated with each of them.

$F1$: livesin = SBrook, income=high

$F2$: income=high

$F3$: livesin = SBrook

$F4$: livesin = SBrook, income=high

$F5$: income=high

**1.** (5pts)

COMBINE this information with transactions from **F** into one transactional data base

$$\mathbf{T} = \{ T1,\ T2,\ T3,\ T4,\ T5 \}$$

**2.** (15pts)

Use Apriori Process to find **Single-dimensional**, **Inter-dimensional** and **Hybrid -dimensional** Association Rules.

Fix minsupport = 3

Rules do not need to be strong, but must be written in Predicate Form.

In a case when there are more than 4 rules in one category you do not need to list all of them but must explain how they are formed.

## Part-2

## Problem-2:

Multi-Dimensional Association Rules.

Data Base F = {F1, F2, F3, F4, F5}

F1 = {Bread, Endive}

F2 = {Bread, Milk}

F3 = {Bread, Milk, Endive}

F4 = {Bread, Endive}

F5 = {Endive}

Additional information:

F1: livesin = SBrook, income = high

F2: income = high

F3: livesin = SBrook

F4: livesin = SBrook, income = high

F5: income = high

1. Combine the information to one transactional database.

T = {T1, T2, T3, T4, T5}

Combined Transactional Data is represented as below

| | I1 Bread | I2 Milk | I3 Endive | I4 livesin SBrook | I5 income high |
|---|---|---|---|---|---|
| T1 | + | - | + | + | + |
| T2 | + | + | - | - | + |
| T3 | + | + | + | + | - |
| T4 | + | - | + | + | + |
| T5 | - | - | + | - | + |
| Sc | 4 | 2 | 4 | 3 | 4 |

Writing transactions using numbers gives

$T1 = \{1,3,4,5\}$

$T2 = \{1,2,5\}$

$T3 = \{1,2,3,4\}$

$T4 = \{1,3,4,5\}$

$T5 = \{3,5\}$

2. Minimum Support = 3.

∴ I2 cannot be included in Frequent 1-item set

Frequent 1-item set L1:

$L1 = \{\{1\}, \{3\}, \{4\}, \{5\}\}$

Join step:

Joining Lk-1 with itself will generate candidate set Ck.

∴ 2-item Candidate set C2 for the given transactions with support count is

$\{1,3\}(3), \{1,4\}(3), \{1,5\}(3), \{3,4\}(3), \{3,5\}(3), \{4,5\}(2)$

Prune step:

Any (k-1) itemset that is not frequent cannot be a subset of a frequent k-itemset.

All the sets in C2 are valid, so pruning is not required.

{4,5} has 2 occurences and min support is 3. so, it cannot be included in Frequent 2-itemset.

Frequent 2-itemset L2:

$L2 = \{\{1,3\}, \{1,4\}, \{1,5\}, \{3,4\}, \{3,5\}\}$

Join Step:

3-item Candidate set C3 with support count is

{1,3,4}, {1,3,5}, {1,4,5}, {3,4,5}

Prune step:

sets {1,4,5} and {3,4,5} cannot be included for Frequent 3-itemset because all their corresponding possible subsets donot exist in Frequent 2-itemset.

Now we have, {1,3,4} (3), {1,3,5} (2)

{1,3,5} has 2 occurences and min support is 3. so, it cannot be included in Frequent 3-itemset.

Frequent 3-itemset L3:

$L3 = \{\{1,3,4\}\}$

END of Apriori

∴ Frequent itemset for Association rules is

FR = {{1}, {3}, {4}, {5}, {1,3}, {1,4}, {1,5}, {3,4}, {3,5}, {1,3,4}}

Single-Dimensional rules:

The rules formed involving only 1,2,3 are single dimensional.

From frequent itemset the rules formed out of {1,3} are single dimensional.

Two single-dimensional rules can be formed:

buys (x, "bread") → buys (x, "endive")

buys (x, "endive") → buys (x, "bread")

Multi-Dimensional rules are formed involving 1,2,3 and 4,5.

From frequent itemset the rules formed out of {1,3,4} are multi dimensional.

Inter-Dimensional rules:

All the rules formed out of {1,3,4} will include 1 and 3, and we see the predicate

"buys" repeats in rules formed.

So, no inter-dimensional rules can be formed from {1,3,4}.

Considering 2-item Frequent sets {1,4}, {1,5}, {3,4}, {3,5}.

Inter-Dimensional rules formed from these sets are:

buys (x, "bread") → livesin (x, "SBrook")

livesin (x, "SBrook") → buys (x, "bread")

buys (x, "endive") → livesin (x, "SBrook")

livesin (x, "SBrook") → buys (x, "endive")

buys (x, "bread") → income (x, "high")

income (x, "high") → buys (x, "bread")

buys (x, "endive") → income (x, "high")

income (x, "high") → buys (x, "endive")

These inter-dimensional rules are formed by taking different combinations of elements from the applicable frequent set & forming two disjoint sets in

which one set of elements implies the other.

Hybrid-Dimensional rules:

All the rules formed out of {1,3,4} are Hybrid-Dimensional rules, because they must involve 1,3 that represent the same predicate "buys".

Some of the hybrid dimensional rules are:

$lives(x, \text{"SBrook"}) \wedge buys(x, \text{"bread"}) \rightarrow buys(x, \text{"endive"})$

$lives(x, \text{"SBrook"}) \wedge buys(x, \text{"endive"}) \rightarrow buys(x, \text{"bread"})$

$buys(x, \text{"endive"}) \rightarrow lives(x, \text{"SBrook"}) \wedge buys(x, \text{"bread"})$

$buys(x, \text{"bread"}) \rightarrow lives(x, \text{"SBrook"}) \wedge buys(x, \text{"endive"})$

$lives(x, \text{"SBrook"}) \rightarrow buys(x, \text{"endive"}) \wedge buys(x, \text{"bread"})$

$buys(x, \text{"endive"}) \wedge buys(x, \text{"bread"}) \rightarrow lives(x, \text{"SBrook"})$

These hybrid dimensional rules are formed by

taking different combinations of 1, 3, 5 and forming two disjoint sets in which one set of elements implies the other.

**PROBLEM 3 GENETIC ALGORITHMS (20pts)**

Here is an initial data table **D**

| income | student | rating |
|---|---|---|
| high | no | fair |
| high | yes | fair |
| low | yes | fair |
| medium | yes | excellent |
| low | no | fair |

The goal of our GA is to transform the data table **D** into a data table **TD** representing the population with the fitness function such that **F** = 0 for all chromosomes.

**Part 1 (5pts)**

**1.** WRITE a set of Binary Encoding Chromosomes representing the Data Table D.

This is your INITIAL POPULATION **P**

**2.** EVALUATE the Fitness Function ( **Definition 2**) for all chromosomes in **P**. Put it all in one evaluation Table.

**Part 2 (15pts)**

**1.** (10pts) Create ONE generations **P1** of your INITIAL POPULATION **P** using GA operators of Selection, Reproduction, Recombination.

Use the Single Point Crossover, Single Point Mutation.

Use the random selection of parents for Crossover.

**2.** Determine the best chromosome in **P1** .

Single Point Crossover and Fitness Function **F** are defined below.

**Definition 1** CROSSOVER

Given a chromosome $CH = c1c2c3c4c5c6c7$

We use as a cross point a point after $c4$, i.e. use the following single cross point

$$c1c2c3c4|c5c6c7$$

**Definition 2** FITTNESS FUNCTION

We define the fitness function **F** as follows.

For any chromosome $CH = c1c2c3c4c5c6c7$ we put

$$\mathbf{F}(c1c2c3c4c5c6c7) = \mathbf{F1}(c1c2c3) + \mathbf{F2}(c4c5) + \mathbf{F2}(c6c7)$$

where

$\mathbf{F1}(c1, c2, c3) = 1$ when only one 1 appears in the sequence $c1c2c3$ and $\mathbf{F1}(c1c2c3) = 0$ otherwise.

$\mathbf{F2}c4c5) = 1$ when only one 1 appears in the sequence $c4c5$ and $\mathbf{F2}(c4c5) = 0$ otherwise.

$\mathbf{F2}(c6c7) = 1$ when only one 1 appears in the sequence $c6c7$ and $\mathbf{F2}(c6c7) = 0$ otherwise.

**2.** (5pts) Write data table **D1** representing the generation **P1**.

**D1** must have a format of the initial data **D**.

Observe that some chromosomes may not define values of some attributes for different reasons.

**D1** (and data tables representing next generations) have missing values.

PART-2

PROBLEM-3

Given initial data table D is:

| income | student | rating |
|---|---|---|
| high | no | fair |
| high | yes | fair |
| low | yes | fair |
| medium | yes | excellent |
| low | no | fair. |

Goal is to transform the data table D to TD representing the population with the fitness function $F = 0$ for all chromosomes.

PART-1

We are asked to write the set of Binary Encoding chromosomes representing the data table D.

We can call this initial population as P.

The possible values for each attribute are

Income = high, low, medium.

student = yes, no.

rating = fair, excellent.

So each chromosome would be of the format

$C_1 C_2 C_3 C_4 C_5 C_6 C_7$ as there are total of 7 unique values.

And we are also asked to calculate the fitness value of each chromosome using Definition 2.

| Chromosome | In=h | In=l | In=m | Ct=Y | Ct=N | R=fair | R=Exc | F |
|---|---|---|---|---|---|---|---|---|
| Ch1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 3 |
| Ch2 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 3 |
| Ch3 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 3 |
| Ch4 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 3 |
| Ch5 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 3 |

Note: The number of chromosomes don't change from the initial data.

The fitness function is also evaluated for all the chromosomes & we got 3 for all of them. Importantly fitness value is same for all the chromosomes.

We have also kept all these values in a single table.

PART-2

(1) We need to create ONE generations P1 from our initial population P1 using our GA operators.

We are also asked to use Single Point Crossover & Single Point Mutation.

SELECTION:

If we see, the fitness values are same for all chromosomes, so we can select randomly any two chromosomes from P.

Note: We keep the population size to be constant. So after the entire process 5 chromosomes should be present.

REPRODUCTION:

We will follow Single point crossover as mentioned in Definition 1 which is after $C_4$. Since the fitness values are same, we can randomly pick 2 for crossover.

We'll pick ch1 & ch2 i.e. 1000110, 1001010 & perform the cross over after c4.

| | |
|---|---|
| ch1 | 1000 ! 110 |
| ch2 | 1001 ! 010 |
| off1 | 1000 010 |
| off2 | 1001 110 |

Similarly next we'll pick ch3 & ch4

| | |
|---|---|
| ch3 | 0101 ! 010 |
| ch4 | 0011 ! 001 |
| off3 | 0101001 |
| off4 | 0011010 |

RECOMBINATION:

Since the population size needs to be constant, we'll pick ch5 as our off5. (fitness values are same)

After this step the chromosomes obtained are 1000010, 1001110, 0101001, 0011010 & 0100110

MUTATION:

Now single point mutation has to be applied on each of the obtained offsprings.

Since it's Single point Mutation, we have randomly selected 1 bit from each offspring chromosome to obtain the final set of offsprings i.e. Since mutation pro. is not mentioned.

1000010, 1001110, 0101001, 0011010 & 0100110

Inverted $C_1$ $C_5$ $C_2$ $C_4$ $C_5$

0000010, 1001010, 0001001, 0010010 & 0100010

TERMINATE:

Since we are asked for only one generation, the ONE generation population $P_1$ looks like this using the process we followed.

| Chromosome | In=h | In=l | In=m | Ct=Y | Ct=N | R=fair | R=Exc | F |
|---|---|---|---|---|---|---|---|---|
| Ch1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| Ch2 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 3 |
| Ch3 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 2 |
| Ch4 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 2 |
| Ch5 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 2 |

(2) The best chromosome after this one generation is ch1 = 0000010 as it's fitness value is 1 which is closer to F=0 (our goal) the optimum ideal solution.

(3) Now we need to write Data Table $D_1$ using the population $P_1$.

$D_1$ =

| Income | Student | Rating |
|---|---|---|
| | | Fair |
| high | yes | Fair |
| | yes | Excellent |
| medium | | Fair |
| low | | Fair. |

If we observe there are few missing values in the data table $D_1$.

The reasons behind them are explained below.

Ch1 : This chromosome doesn't have values for Income & student attributes.

Ch3 : No value for Income attribute

Ch4 & Ch5 : No defined values for student attribute.

So finally data table $D_1$ is generated of the same format as of D. (Initial data table).