**CSE 573 SEMANTIC WEB MINING**

**Mid-Term Exam - SPRING 2022**

**400 Points (40% of course grade)**

**March 16th, 2022**

1. Take-home and open note exam.
2. This is an **individual work exam** and **you may not work in groups**.
3. The midterm is attached to these instructions.
4. Please note that this is an **individual exam** and you **may not work in groups**.

**Exam Submission Instructions**: You can (1) type your answers into this word document and convert it to PDF, or (2) write down your answers on plain papers, scan and upload them as a PDF at **Canvas > Assignments > Mid-Term Exam. DUE: March 16, 2022 11:59pm** by mid-night tonight.

**Duration**: The exam should take only about 75 minutes but I am giving you plenty extra time to work on it, convert or scan it as a single PDF document and submit it at Canvas.

**Academic Integrity**: All students in this class are subject to ASU’s Academic Integrity Policy and you should acquaint yourselves with its content and requirements, including a strict prohibition against working together or copying from each other and plagiarism. All violations will be reported to the Dean’s office, who maintain records of all offenses.

I will be online at Skype if you have any questions. My SkypeID is hdavulcu. Or, you can call me at my cell phone (602) 386.6565 if you have any questions.

**GRADING**

* The exam is worth 400 points. Best of Luck to all!

**DO NOT FORGET TO WRITE YOUR NAME and ASUID on top of each exam sheet!**

***PART I: Semantic Web Languages* [100 points]**

**1. Translate the HTML information below into an XML object. [30 pts]**



**Answer:**

The following is an example of an XML object representation of HTML content:

<?xml version="1.0" encoding="UTF-8"?>

<restaurants>

<restaurant>

<name>Culinary Dropout</name>

<phone>4802401601</phone>

<address>149 S Farmer</address>

<rating>4.0</rating>

<reviews>1621</reviews>

<pricerange>2</pricerange>

<cuisine>American (New)</cuisine>

<specialty\_icon>glasses.png</specialty\_icon>

<specialty>Most viewed <b>restaurant</b> in Tempe</specialty>

<offer\_reservation>true</offer\_reservation>

</restaurant>

<restaurant>

<name>Ghost Ranch: Modern Southwest Cuisine</name>

<phone>4804744328</phone>

<address>1006 E Warner Rd</address>

<rating>4.5</rating>

<reviews>237</reviews>

<pricerange>2</pricerange>

<cuisine>American (New)</cuisine>

<specialty\_icon>fork.png</specialty\_icon>

<specialty>Popular for its <b>Ghost Ranch</b></specialty>

<offer\_takeout>true</offer\_takeout>

</restaurant>

</restaurants>

</xml>

**Note:**

This presentation assumes that the pricerange mark (with $$) indicates the value of the restaurant. The price of 1 equals $, which means it is much cheaper and the value of 5 equals $$$$$ can mean more expensive. We also assume that a special image icon is available in the png path provided. Also, the button display is assumed to be dependent on values of ‘offers\_reservation’ and ‘offers\_takeout’ tags respectively and it need not come in XML data itself.

**2. Draw the graph representation for above XML object and its lower bound schema. [40 pts]**

**Diagram

Description automatically generated**

**Diagram, text, letter

Description automatically generated**

**3. Translate following XML-QL query with Regular Path Expressions to an English query. [30 pts]**

**SELECT R.telephone**

**FROM Yelp.restaurant R**

**WHERE R.takeout=true**

Return the **telephones numbers** of the **restaurants in Yelp** that have **takeout** option.

***PART II: Frequent Item-Sets and Association Rules* [100 points]**

**4. Apply the APRIORI algorithm assuming required minimum support at 40% (4 out of 10) to the following set of 10 transactions involving items {A, B, C, D, E, F, G}.**

**T1={A,B,C} T6={B, C, D, E}**

**T2={A,F,G} T7={E, F}**

**T3={B,C,G} T8={B, C, F, G}**

**T4={A, B, C, F} T9={B,C, F, G}**

**T5={B, C} T10={B, C, G}**

**Show how Apriori’s level-wise Item Set Generation algorithm works for the above example. Indicate what candidate itemsets will be generated in each pass, and which remain in the candidate set after pruning. [50 points]**

**Answer:**

Let's use Ci to designate the candidate itemset and Fi to signify the frequent itemset for the ith iteration. Required, Minimum Support = 40% = minSupport

Given Items Set: A, B, C, D, E, F, G

|  |  |
| --- | --- |
| **Transaction** | **Items** |
| T1 | {A, B, C} |
| T2 | {A, F, G} |
| T3 | {B, C, G} |
| T4 | {A, B, C, F} |
| T5 | {B, C} |
| T6 | {B, C, D, E} |
| T7 | {E, F} |
| T8 | {B, C, F, G} |
| T9 | {B, C, F, G} |
| T10 | {B, C, G} |

Let's look at the itemset : count pairs for various scans and iterations.

**First Scan:**

In the join step, we generate all possible candidate itemsets of length 1

|  |  |
| --- | --- |
| Frequency 1-Item Sets | Frequency |
| A | 3 |
| B | 8 |
| C | 8 |
| D | 1 |
| E | 2 |
| F | 5 |
| G | 5 |

|  |  |
| --- | --- |
| Items | Support |
| A | 3/10 |
| B | 8/10 |
| C | 8/10 |
| D | 1/10 |
| E | 2/10 |
| F | 5/10 |
| G | 5/10 |

Now, in the Pruning stage, we eliminate all the itemsets with support less than minSupport, leaving just Frequent Itemsets with support more than minSupport. The minSupport =4

Thus, we get Frequent Itemset **F1**: {B}:8, {C}:8, {F}:5, {G}:5

Similarly we have,

**Second Scan:**

In join step, using F1 we get the following candidate set C2

**C2**: {BC}, {BF}, {BG}, {CF}, {CG}, {FG}

Accounting for the counts of different itemsets from C2, we get:

|  |  |
| --- | --- |
| Frequency 2-Item Sets | Frequency |
| BC | 8 |
| BF | 3 |
| BG | 4 |
| CF | 3 |
| CG | 4 |
| FG | 3 |

Again, After pruning, we should only be left with the itemsets that have support >= minSupport

=> **F2**: {BC}:8, {BG}:4, {CG}:4

**Third Scan:**

In join step, using F2 we get the following candidate set C3

**C3**: {BCG}

Accounting for the counts of different itemsets from C3, we get:

**C3**: {BCG}:4,

|  |  |
| --- | --- |
| Frequency 3-Item Sets | Frequency |
| BCG | 1 |

After pruning, we should only be left with the itemsets that have support >= minSupport

=> **F3**: {BCG}:4

Since, we are left with only one itemset, the Apriori’s level-wise Item Set Generation algorithm stops here.

**5. Generate association rules with support of 40% and confidence of 100%. [50 points]**

**Answer:**

Algorithm for finding association rules:

For each frequent itemset X,

For each proper nonempty subset A of X, Let B = X - A

A → B is an association rule if Confidence(A → B) ≥ minconf,

where, **support(A** → **B) = support(A**∪**B) / support(X)** and

**confidence(A** → **B) = support(A** ∪ **B) / support(A)**

Using question 4, we get 4 frequent itemsets - {BC}, {BG}, {CG}, {BCG} Using above algorithm, we get following association rules:

|  |  |  |
| --- | --- | --- |
| Rule | Support | Confidence |
| B -> C | 8/10 = 80% | 8/8 = 100% |

|  |  |  |
| --- | --- | --- |
| C -> B | 8/10 = 80% | 8/8 = 100% |
| B -> G | 4/10 = 40% | 4/8 = 50% |
| G -> B | 4/10 = 40% | 4/5 = 80% |
| C -> G | 4/10 = 40% | 4/8 = 50% |
| G -> C | 4/10 = 40% | 4/5 = 80% |
| BC -> G | 4/10 = 40% | 4/8 = 50% |
| BG -> C | 4/10 = 40% | 4/4 = 100% |
| CG -> B | 4/10 = 40% | 4/4/ = 100% |
| B -> CG | 4/10 = 40% | 4/8 = 50% |
| C -> BG | 4/10 = 40% | 4/8 = 50% |
| G -> BC | 4/10 = 40% | 4/5 = 80% |

Thus, association rules with support 40% and confidence 100% are **B -> C**; **C -> B**; **BG -> C**; **CG -> B**

***PART III: Clustering* [100 points]**

**6. You want to cluster following 7 observations into 3 clusters using the K-Means clustering algorithm. Assume that after the first iteration clusters C1, C2, C3 has following members:  
  
C1: {(2,2), (4,4), (6,6)}  
  
C2: {(0,4), (4,0)}  
  
C3: {(5,5), (9,9)}  
  
What will be the cluster centroids if you want to proceed to the second iteration? [40 pts]**

**C1: (4,4); C2:(2,2), C3:(7,7)**

**Text, letter

Description automatically generated**

**7. What will be the clusters for the second iteration? [60 points]**

**Text, letter

Description automatically generated**

**Text, letter

Description automatically generated**

**A piece of paper with writing

Description automatically generated with medium confidence**

**Text, letter

Description automatically generated**

**C1: {(4,4),(5,5)}  
  
C2: {(2,2),(0,4),(4,0)}   
  
C3: {(6,6),(9,9)}**

***PART IV: Web Ranking* [100 points]**

***8. A directed network N has vertices {A, B, C} and edges (A🡪 B), (B🡪 B), (B 🡪 C), (C 🡪 A).***

1. ***Draw the network N and write down its column-normalized hyperlink matrix M for the calculation of simplified PageRank. [30 points]***

Shape

Description automatically generated with low confidence**Answer:**

The transition matrix (adjacency matrix) for above graph looks as following:



On transposing it we get:

A picture containing text, clipart, clock

Description automatically generated

On column-normalized, the hyperlink matrix M looks like following:

A picture containing text, clock, clipart

Description automatically generated

***9. Is M an irreducible matrix? Explain why? [30 points]***

A graph is called irreducible if for any pair of distinct nodes, we can start from one of them, follow the links in the web graph and arrive at the other node, and vice versa.

As we can observe,

Start from node A, we can reach Node B and C by traversing edges (A,B) and (B,C) If we start from node B, we can reach Node C and A by traversing edges (B,C) and (C,A) If we start from node C, we can reach Node A and B by traversing edges (C,A) and (A,B).

Thus the Network N graph is irreducible.

Since M is the hyperlink matrix of an irreducible graph, hence M is an irreducible matrix.

***10. Show the evolution of the initial page-rank vector x = <xA, xB, xC> = <1, 1, 1> until the simplified pagerank algorithm converges. [40 points]***

|  |  |  |  |
| --- | --- | --- | --- |
| ***iteration*** | ***XA*** | ***XB*** | ***XC*** |
| ***0*** | ***1*** | ***1*** | ***1*** |
| ***1*** | ***1*** | ***1.5*** | ***0.5*** |
| ***2*** | ***0.5*** | ***1.75*** | ***0.75*** |
| ***3*** | ***0.75*** | ***1.375*** | ***0.875*** |

**Hint.** Replace ***x*** by the product **M *x*** until page rankings converge. Therefore, we stop after iteration 3.

Ranking for 2nd iteration and 3rd iteration remain same (***XB> XC> XA).*** Therefore, we stop after iteration 3.