Mohammed J. Zaki

Dmcourse /

Assign6

Graph Mining: Due Date: 6th Dec (extended), before Midnight

In this assignment you will implement the frequent graph mining algorithm given in **Algorithm 11.1** in the textbook on page 288. This algorithm implements the gSpan algorithm, which is a depth-first method. It uses the notion of rightmost path extensions to generate the possible edge-extensions from an input frequent subgraph (in Algorithm 11.2). It uses the notion of minimal DFScode to determine whether a candidate subgraph extension is canonical or not (Algorithm 11.4), and it enumerates the subgraph isomorphisms to determine whether it is frequent (Algorithm 11.3). Graphs are represented in the DFScode format (a list of extended tuples). All the details are described in the text.

The input graph database comprises multiple graphs in the following format:

```
 t # GRAPHID
 v ID LABEL (there are multiple such lines, one per vertex in the graph, giving its ID and LABEL)
 e ID1 ID2 LABEL (there are multiple such lines, one per edge, with ID1 and ID2 as the two end-points, and LABEL  the edge label)
```

finally, there are multiple such graphs with dif ferent GRAPHIDs

The example dataset comprising two patterns as shown in Figure 11.7 is therefore specified as follows:

```
t # 1
v 10 a
v 20 b
v 30 a
v 40 b
e 10 20 _
e 10 30 _
e 20 30 _
e 30 40 _
t # 2
v 50 b
v 60 a
v 70 b
v 80 a
e 50 60 _
e 50 70 _
e 60 70 _
e 60 80 _
e 70 80 _
```

You can download this in the following text file: Attach:exampleG.txt

The correct output for this small example for minsup=2 is as follows:

```
pattern 1
()

pattern 2
(0, 1, 'a', 'a', '_')

pattern 3
(0, 1, 'a', 'a', '_')
(1, 2, 'a', 'b', '_')

pattern 4
(0, 1, 'a', 'a', '_')
(1, 2, 'a', 'b', '_')
(2, 0, 'b', 'a', '_')

pattern 5
(0, 1, 'a', 'a', '_')
(1, 2, 'a', 'b', '_')
(2, 0, 'b', 'a', '_')
(1, 3, 'a', 'b', '_')

pattern 6
(0, 1, 'a', 'a', '_')
(1, 2, 'a', 'b', '_')
(1, 3, 'a', 'b', '_')

pattern 7
(0, 1, 'a', 'a', '_')
(1, 2, 'a', 'b', '_')
(0, 3, 'a', 'b', '_')

pattern 8
(0, 1, 'a', 'b', '_')

pattern 9
(0, 1, 'a', 'b', '_')
(1, 2, 'b', 'a', '_')

pattern 10
(0, 1, 'a', 'b', '_')
(1, 2, 'b', 'a', '_')
(2, 3, 'a', 'b', '_')
```

```
pattern 11
(0, 1, 'a', 'b', '_')
(0, 2, 'a', 'b', '_')
```

You should print your output in this same format.

Here is another test dataset Attach:protein.txt that has 26 graphs. Here is the output you should get for minsup=10, and here is the output for minsup=15. You can use these to test your program.

Run your code on the Attach:Compound_422.txt and print the output using minsup=100.

What to turn in

- Write a python script called **assign6-LAST-FIRST.py**, and submit a text file that contains the output of the script.
- Your script should read the filename and minsup from the command line as parameters, i.e., it will be run as "assign6.py FILE minsup", where minsup is an integer.
- Submit the assignment via email to: datamining.rpi@gmail.com. The subject of your email should be **assign6-LAST-FIRST**.

Page last modified on December 05, 2016, at 11:24 PM