**User Manual of GAIA**

*This is the user manual of an implementation of "GAIA: graph classification using evolutionary computation" in Proceedings of the ACM SIGMOD International Conference on management of Data, pages 879-890, 2010.*

*authors: Ning Jin, Calvin Young, Wei Wang*

*affiliation: University of North Carolina at Chapel Hill, U.S.A.*

*implemented by: Ning Jin*

*contact: njin@cs.unc.edu*

**CONTENTS:**

**1. WHAT'S INCLUDED**

**2. HOW TO COMPILE**

**3. HOW TO USE THE BINARY**

**1. WHAT'S INCLUDED**

**1.1 sample input files**

edge\_file.txt: a sample edge file with 11 positive graphs and 52 negative graphs (please see 3.2 for file formats)

node\_file.txt: a sample node file associated with the edge file (please see 3.2 for file formats)

**1.2 source code files**

candidate\_list.h: declaration of class candidate\_list, which corresponds to the candidate list in the paper

candidate\_list.cpp: implementation of class candidate\_list

common.h: macro definitions

EVO.h: declaration of class EVO, which corresponds to the evolutionary mining algorithm in the paper

EVO.cpp: implementation of class EVO

feature.h: declaration of class feature, which corresponds to the representative feature in the paper

feature.cpp: implementation of class feature

graph.h: declaration of class graph, which corresponds to the input graph in the paper

graph.cpp: implementation of class graph; including reading the input graphs

main.cpp: main function and some other auxiliary functions

pattern\_index.h: declaration of class pattern\_index, which is used to keep track of the codes of subgraph patterns that have been generated

pattern\_index.cpp: implementation of class pattern\_index

pattern.h: declaration of class pattern, which corresponds to the subgraph pattern in the paper

pattern.cpp: implementation of class pattern; including pattern encoding and pattern extension

**1.3 user manual file**

user\_manual\_of\_GAIA.pdf: the file you are reading, including a brief introduction to how to use the source code

**1.4 developer manual file**

developer\_manual\_of\_GAIA.pdf: description of classes, non-trivial members and methods of each class, relationship between classes and the execution order of methods to run GAIA

**1.4 configuration file**

GAIA\_config: it specifies argument values

**2. HOW TO COMPILE**

g++ -O2 -o gaia candidate\_list.cpp EVO.cpp feature.cpp graph.cpp main.cpp pattern.cpp pattern\_index.cpp

**3. HOW TO USE THE BINARY**

**3.1 arguments and example**

All arguments can be set in file GAIA\_config and the file should be in the same directory as this binary. If GAIA\_config is absent or cannot be parsed successfully, default values (described as below) will be used. Below is an example of argument setting GAIA\_config (as in the file GAIA\_config):

node\_file\_name = node\_file.txt

edge\_file\_name = edge\_file.txt

number\_of\_positive\_graphs = 50

candidate\_list\_size = 100

number\_of\_iterations = 10

Or you can set arguments in command line with the following options:

-v: node\_file\_name

-e: edge\_file\_name

-p: number\_of\_positive\_graphs

-n: number\_of\_iterations

-s: candidate\_list\_size

Note: Command line settings override config file settings.

**3.2 file formats**

(a) input file formats

One input graph dataset is composed of two files: a node file and an edge file. The two files are supposed to share the same prefix indicating the name and/or property of the dataset and differ only in their suffices. The node file ends with "\_node\_file.txt" and the edge file ends with "\_edge\_file.txt". Each row of a node file stores the information of one node and each row of an edge file stores the information of one edge.

The node files have the following format:

1st column 2nd column 3rd column 4th column

extra information of the node graph ID node ID node label

(not used for pattern mining)

ATTENTION: NODE LABELS CANNOT BE ZERO.

The edge files have the following format:

1st column 2nd column 3rd column 4th column 5th column

extra information graph ID ID of node1 ID of node2 edge label

(not used for pattern mining)

In the same row, ID of node 1 is assumed to be smaller than ID of node 2.

ATTENTION: EDGE LABELS CANNOT BE ZERO.

(b) output file formats

i) "pattern.txt" file contains the adjacency matrices of the resulting subgraph patterns and their frequencies; the first line shows the number of resulting patterns N; the next N lines list the pattern IDs of the resulting patterns and their corresponding frequencies; at the end of file is the adjacency matrices of the resulting patterns

ii) "feature.txt" file contains the code, discrimination score and IDs of the supporting graphs of each resulting pattern for each pattern, the first line shows the number of nodes in the pattern; the next two lines show the code of the pattern; the 4th line is the score of the pattern; the remaining lines list the IDs of the supporting graphs

iii) "svm.txt" file the input file for LIBSVM based on the resulting subgraph patterns each line shows the feature vector for one input graph; if the input graph is positive, then the line begins with "+1", otherwise it begins with "-1"; given a feature with ID=K, if the graph has this feature, then the corresponding line includes "K:1.0", otherwise feature K is absent from that line