

edu.umbc.cs.maple.data_generation_MAPLE_N

--INTRO--

edu.umbc.cs.maple.data_generation_MAPLE_N is a package for automatically producing data sets with varying sizes (number of nodes), density (number of edges), and attributes (of nodes) to create randomly weighted directed graph and writes out a XML file in XGMML format(for a description of the XGMML format see <http://www.cs.rpi.edu/~puninj/XGMML/>). Output file that produced in edu.umbc.cs.maple.data_generation_MAPLE_N can be used as input file for loading data into the graph for test cases for the clustering and interactive layout algorithms.

--REQUIREMENTS--

edu.umbc.cs.maple.data_generation_MAPLE_N is written in Java 1.4 (available for free download at <http://java.sun.com/j2se/1.4.2/download.html/> and the Eclipse integrated development environment (available for free at <http://ww.eclipse.org>) to load the source files To compile, to build and run edu.umbc.cs.maple.data_generation_MAPLE_N source files, you have to use any version of Java and any development environment. The new created output file from edu.umbc.cs.maple.data_generation_MAPLE_N is loadable by prefuse, a user interface toolkit for building highly interactive visualizations of structured and unstructured data (available for free at <http://prefuse.sourceforge.net/>), but can be used as a source in any program that uses an XML file in XGMML as an input file.

--BUILDING--

If you use the prefuse package and Eclipse you must create directory edu.umbc.cs.maple.data_generation_MAPLE_N and insert it into prefuse/src. The new created output file you will find in the "Workplace". You can use the new created output file as input file in your new programs, or in any examples, that loaded input XML file in XGMML format from prefuse. These examples are available in prefuse/src/demos. To run examples with the new created file from edu.umbc.cs.maple.data_generation_MAPLE_N, you must change in the prefuse exaples the directory from witch the new created file will be loaded (for example, public static final String INPUT_FILE = "name_of_your_file.xml";)

```

/*****
 * WriteXML_N.java
 *****/

/**
 * @author nlozova
 * A class WriteXML_N writes out a graph to an XGMML-format XML file.
 *
 * FUNCTIONALITY
 * - basic console-oriented input/output,
 * - verifying prompted data,
 * - creating new output file with a given name and fill this file with prompted and
 *   manipulated data
 * - creating Vector and storing prompted name of attributes associated with each node in
 *   the Vector;
 * - getting the attributes from Vector to insert this data into the new created file;
 * - creating linked list, every the node of which contains Vector of different size,
 *   the length of linked list dependent of prompted number of attributes associated
 *   with each graph's node
 * - filled with the prompted values that each attributes is allowed to take on.
 *   the size of vectors varies for different attributes
 * - traversing linked list

```

```

* - randomly getting data from vectors in the linked list node during traversing
* - inserting this data in the created file;
* - creating squared random matrix of the size of prompted number of nodes for the graph
* - getting values from this matrix, that represent random connection between nodes in
*   the graph
* - inserting this data into the file
* - New created output XGMML-format XML file that represent graph and loaded by prefuse
* - number of nodes, edges and attributes in the represented in the output file graph
*
* For implementation this class is used higher-level functions
* 1. PromptData.java - for prompted data from console
*
* 2. ListNode.java, LinkedListItr.java, LinkedListN.java - to create and manipulate
*   linked list that allow to contain nodes with equal vectors
*   (the vectors each of which represents the values of one attribute
*   that contains graph's node can be identical)
*
* 3. MatrixN.java - to create and manipulate matrix for randomly creating edges between
*   graph's nodes;
* - in case of the directed graph - random setting direction of the
*   edges;
* - to build the bridge in the graph in case if randomly created
*   unconnected graph
*
* 4. LoadXML_N.java - to fill up output file
*
* PROCEDURE
* - prompt number of nodes
* - verify integer and if the number is greater than zero
* - prompt probability - density of an edge from node i to node j in the graph
* - verify double and interval [0, 1] for probability
* - prompt number of attributes associated with each node
* - verify integer and if the number is greater than zero
* - prompt in the loop of varying sizeID or name of ith attribute,
* - insert every attribute ID or name into the vector
* - create linked list
* - prompt in the loop of length of number of attributes values that attribute i
*   is allowed to take on.
* - (!!!) To interrupt to fill Vector-node in the linked list press "Enter"
* - store values for attribute i in the Vector-node i from the Linked List.
*   The size of every Vector-node is varied, depending of number of prompted values
*   that attribute i is allowed to take on.
* - create matrix of size n x n, n - is the number of nodes,
* - fill matrix with randomly edges for fully connected or randomly connected graph
*   (see MatrixN.java)
* - extract data from Linked List, Matrix,
* - parse the XML file
*
* OUTPUT
* - New created output XGMML-format XML file that represent graph and loaded by prefuse
* - number of nodes, edges and attributes in the represented in the output file graph
*/

```