IDA@SMU

SMU | Bobby B. Lyle School of Engineering

Progress Report

Research Experience for Undergraduates (REU) under Grant No. IIS-0948893

Vladimir Jovanovic

12/10/10

http://lyle.smu.edu/IDA/TRACDS/

This research is supported by the National Science Foundation under Grant No. IIS-0948893. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Fall 2010 Progress Report

Goals

For the fall semester of 2010, the primary goal for the IDA group working on EMM was development of a web based version of TRACDS and EMMSA. Visualization of the data was required as well as an easy to use interface.

Tasks

I had three main tasks for the fall semester of 2010. I was to learn about EMM by reading published articles on it and interacting with the Java implementation of EMM (JEMM). I was to get a working implementation of JEMM, and I needed to create an interactive visualization that could work both on JEMM and the web based version of TRACDS.

Results

Thanks to John Forrest the web-based interface of the EMMSA program is up and running. My job was centered on creating a Java based interactive and dynamic model for the EMMSA models. The current version is easily used in both the Java version and the applet version available online.

On top of this, JEMM (the Java implementation of EMM) was looked at by me. I fixed some bugs and added some more information in the Java version. The GUI has not been updated, but EMMSA features have been added – specifically the count generator. The interactive model is also available in this version.

Interactive Model

In-depth

I created an interactive method of visualizing the EMM generated in the EMMSA applet. This method allows the user to see in-depth information about the states in the EMM, including centroid vector values, number of transitions, and connections between other states. A help file was created for this model that goes into more depth. The different buttons and features of the visualization are described.

This code was based off of the GraphLayout applet available for free in the demo section of the Java development kit (JDK). The main component that remains unaltered from the original code is the simulation and thread section. The event handling, drawing and initialization have been changed from the original to suit the needs of the project.

JEMM Update

In-depth

JEMM is the name given to the Java implementation of the EMM code. I was told to look at this code to learn how EMM works and created. On top of this, I was told to bring the code up to date and get a working version out. Through the first initial weeks, I did get the code to functioning status; however, I did not change the GUI. This means that my current version of JEMM, while functional, does not have the visual appeal that the applet version of EMMSA has.

I added in the ‘count generator’ from a previous version of JEMM. This allows the user to create data that is needed for EMMSA modeling. On top of this, I edited the code to allow EMMSA formatted data sets to run in JEMM. I added new views of the model, including a transition and probability matrix. Finally, I added in the visualization that can be seen at the end of model creation.

While working on a class project in data mining, I had to compare two EMMs for classification. This involved calculating the probability of a certain sequence of vectors occurring in either EMM. I edited the JEMM code base to allow such a comparison of models. Again, this is a hodgepodge of features that have been clunked together that I added as needed.

Future Work

Interactive Model

The representation of each connection between states might have to be changed. Currently the length of each arrow from one state to another is determined by the visual distance between two nodes and the number of transitions. Recent talks are leaning to a constant length, but a thickness that is variable to the amount of transitions.

A source and sink method needs to be defined in the applet to allow the user to visually see how EMMSA models flow from start to end. On top of this, the vector that is printed out might have to be changed. Currently, the representation is a large mass of text whenever a node is clicked. Something that is more user-friendly would be desired.

The method of orientation for the model has to be changed as well. Right now, after a model gets to a 'relaxed' state wherein each length of arrow is only dependent on transition count, there is still some random movement of states. The proposal right now is to have the simulation run for a few seconds to have it sort out, and then keep the image static as the user clicks and drags at the nodes.

JEMM Update

The Java code needs to be documented extensively. I tried to keep up with all my changes, but even the original code does not have all of the documentation that is usually assumed with Java docs. On top of this, some sort of documentation on the GUI itself is needed to describe the functionality of all the components. As it stands, only I could be able to navigate the major features I added in, like comparison of EMMs.

The distance measure Java files need to be analyzed. Through some tests with R and Java, I have found that the Jaccard method works as hoped. I did not test the other methods, and there could be some unknown errors lying in the implementation of the different distance measure methods. The current implementation also requires searching of class files which seems cumbersome. An analysis of the code base is needed to see if there is an easier and simpler way to code that portion of the program.

The GUI needs to be redone for the program. The current one, while somewhat self-explanatory, is not up to par compared to the EMMSA applet. The display of the EMM, buttons, checkboxes and distance measures need to be rearranged so that an unfamiliar user is not confused.