Hi Developers,

Name:
Abhinav Shankaranarayanan Venkataraman (Abhinav S V)

Contact Information:
Gmail Id : abhinavsv3@gmail.com
IRC : abhinav or abhinavsv3

Title: Visualization of high dimensional data using graph and matrix algorithms

1. Synopsis:
A point or a node in a high-dimensional space, such as R^1000 or more cannot be easily shown on the screen. On a screen we can only show points in R^2, or perhaps R^3 or R^4 somehow if we add colors, sizes or shapes etc. The project that I propose looks into the best way of visualizing or plotting such data on screen such that complex landscapes are easily seen. Graphs are made up of vertices and edges and would be aptly suitable for this problem and thus the project would be using graph and matrix(Graph can be represented in the form of matrix) based algorithm for its backend processing.

2. Benefits to community:
This can be used in places where there is difficulty in visualization of a very complex landscape such as medical domain. In Medical domain a patient can be a vector of diseases and visualization of such patients(patients graph – which shows relations of how two patients are similar, a graph in which patient-patient edge weight is the similarity value) would be useful for analyzing and predicting the disease landscape of a region and in turn multiple regions. City road planners, Bus traffic control system, Public transport system etc can be made more visualizable through this kind of a representation of a complex landscape.

3. Challenge to be tackled in the project:
There are many obstacles being faced such as projecting high dimensional data into low dimensional data, representing the data in the form of blobs etc.

4. Stages or Components to be present in the project:
Stage1 : Developing the graph algorithm modules in python separately. The visualization tool in JavaScript separately. The data for testing separately.
Stage2: Integrating the algorithms to represent the graphs dynamically
Stage3: Testing with the help of SNAP graphs. SNAP provides multiple graphs to test the algorithms and to compare the speeds of it.
Stage4: Revamping and final patching
Stage5: Documentation

5. Deliverables:

5.A Investigation(23 April to 22 May – Community Bonding Phase):

This involves knowledge acquisition about the problem. Revising the papers on louvain algorithm. This also involves trying many algorithms,comparing running time and choosing the one that best fits the need. This deliverable would deliver various snippets of program from which best one is chosen for further usage in the project.

5.B Coding Phase 1 (23 May to 4 June):

Developing Graph Algorithm Modules in python which can separate communities such that representation would be feasible. Louvain modularity index would be useful in doing such splits. In this phase the code will be written in python. The input is taken from a matrix of numbers stored in a file or multiple files depending on the complexity of the landscape of data. Since, the main motivation is visualization, JavaScript can produce a web based visualization. Thus, the output of the program should be a JSON object which would be useful for further processing in phase 2. For testing purpose in Testing Phase 1 we can also produce a text file output containing a matrix of the rearranged graph locations.

5.C Testing Phase 1 (7 June to 16 June):

Testing the various algorithms that have been coded in Coding Phase 1 using SNAP data of graphs. SNAP - Stanford Network Analysis Project has many data to test the algorithm and comparison of speed can be done and errors must be checked. Debugging can be done using simple gnu debugger or using nemiver debugger. Testing the input and out formats is also essential in this stage as the next phase depends on the output of the current phase.

5.D Documentation Phase 1 (17 June to 20 June) :

Documentation of the above code in English and uploading the code into repository with documentation would be necessary for future purpose. Final wrap-up of phase 1 can be made. A buffer of three days is given to compensate any delays if there is one.

5.E Mid-Term Evaluation (23 June Submission):

A part of the project is ready for mid term evaluation by this time. The part that will be available would be the graph and matrix algorithm that takes inputs from a text file of inputs.It evaluates and produces a JSON object file.

5.F Coding Phase 2 (24 June to July 2)

Phase 2 of the project mainly deals with the idea of visualization of the data.
Using the JSON Object created in Phase 1 we would now write code in JavaScript to visualize the graph using visualization software such as Alchemy.js or D3 (which would have been chosen in investigation phase). JSON object is used as input for the web based JavaScript visualization. The JavaScript code must also take the robustness as a factor into consideration and generate the program the same way.

5.G Testing Phase 2 (July 3 to July 14):

Test hosting of the visualization can be done using the github pages facility. This phase must test the robustness of representation, color,shape and the depth of visualization. Proper labeling must be tested. SNAP data generated JSON should be used and the output must be tested accordingly.

5.H Documentation Phase 2(July 15 to July 23):
Documentation of the visualization of the software is done in this phase of the project. All functionalities must be stated in order and future works and development comments must be inline in the code. Final wrap-up of phase 2 can be made. A buffer of three days is given to compensate any delays if there is one.

5.I Final Testing (July 24 to August 10):
Final testing involves running the full program right from the python backend to the client side JavaScript code. Errors pertaining to the integration between the two phases and runtime errors should be rectified using debugging softwares and human intervention.

5.J Final Documentation (15 August)
Review on final documentation of both phases must be done. Verification of the presence of all the stages of the project must be done in this phase. Final documentation should also include future-works and improvements that can be done and the method in which the work has to be done.

5.K Final Evaluation and Outcome:
The outcome of the project will be a software that would perform a backend processing in python and the foreground or client side processing in javascript and thus would be fast and an effective method of visualizing the complex landscape of data.

6. Related Works
Louvain Algorithm : https://perso.uclouvain.be/vincent.blondel/research/louvain.html would be useful.

7. Biographical Information
I am currently pursuing my final year B.Tech Computer Science and Engineeing at SASTRA University, India. I was an intern at the Indian Institute of Information Technology (IIIT), Kanchepuram in graph algorithms and have also done my summer intern at Institute of Mathematical Sciences(IMSc), Chennai,India. I have worked in software concerning graph and matrix at Tokyo city university. My final year intern is at Universitat Politecnica de Caltalunya (UPC) Barcelona.I have written minor patches for GNOME in their gnome-calculator project which is in vala. I am an open source programmer and a researcher.

8. Links:
- LinkedIn Public Profile : https://in.linkedin.com/in/abhinav-shankaranarayanan-venkataraman-81b2a626
- github : https://github.com/abhinavsv3
- github Page: http://abhinavsv3.github.io/

9. Commitments:
I have completed all my courses and I have no more course to further take or complete during the period of May to August. I have completed all my current semester courses through an accelerated program at my university during my previous semesters and thus I am free to do the project. This is the only project that I will be working on and thus will have no distraction.

10. Future Works:

Trajectory tracking for finding the most used paths can also be analysed. Analysis of frequent trajectory can be used to identify the most chosen direction by the program. Integrating such components can make this project a potential graph algorithm based project which can serve as a new method of analysis of data, both in an efficient and in a visual way .

11. Conclusion

Such a visualization software in view would be a great contribution to the open source community.