CS 294-10

Graph Compare

Jointly laying out graphs for easy comparison

```
Initialize X;
for n from 0 to num_iter:
  initialize dX;
  for i from 0 to num_vertices:
    dX[i] = deriv(X, i);
  X = X - dX * eps;
```

The pseudocode of our layout algorithm

Abstract

Graphs are commonly used to represent structured information. There exist many techniques for graph visualization and graph layout. However, there has been much less work in the area of visualization for graph comparison. We propose a layout algorithm for simultaneous graph layout that facilitates comparison by maintaining relative distances between shared structure.

Stress-based Layout

Our layout algorithm is based on stress minimization. First, we introduce the basics for single graph layout.

Given graph G = (V, E) where the vertices have unique labels. Let X_i be the coordinates for vertex i and X be the matrix of all vertex coordinates.

Stress (X) = $\sum_{i<j} w_{ij} (||X_i - X_j|| - d_{ij})^2$

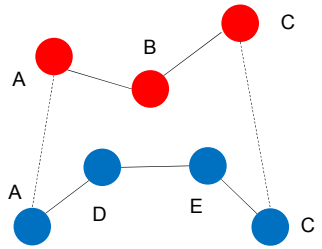
where d_{ij} is the shortest path from vertex i to vertex j and w_{ij} is d_{ij}^{-2}.

We minimize the stress using gradient descent. The partial derivative of the stress is:

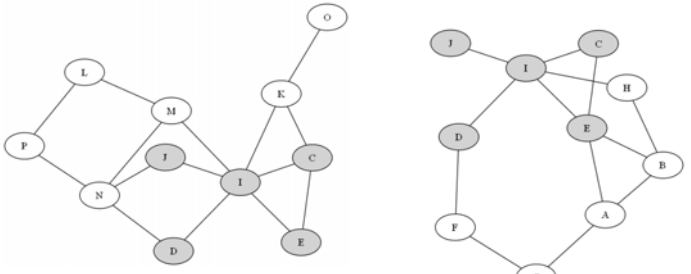
$\sum_{j \neq i} 2*w_{ij} (||X_i - X_j|| - d_{ij}) (X_i - X_j) / ||X_i - X_j||$

Simultaneous Layout

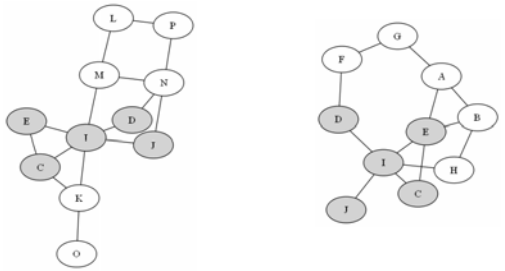
We lay out two graphs G1 and G2 simultaneously by creating a new graph G and using the stress-based layout on G. Graph G contains all vertices and edges in G1 and G2, with additional zero weight edges between vertices of G1 and G2 with the same label.



The rationale is that the side-by-side layout for the two graphs will try to preserve the relative distances between corresponding nodes, helping spot shared structure.



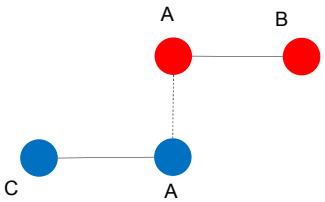
Graph layout for two graphs computed simultaneously using our algorithm.



The same two graphs as above, with the layout computed by the Graphviz neato algorithm.

Limitations

Currently, the algorithm only works for graphs whose vertices have unique labels. For graphs with non-unique labels it may be possible to compute a structured mapping between vertices using inexact graph matching (i.e. graph edit distance). In addition, while the algorithm maintains relative distances, it does not place common graph structure in the same relative positions.



Matthew Can
UC Berkeley
Computer Science