Visualizing Multivariate Network Using GeoSOM and Spherical Disk Layout

Yingxin Wu*
School of Information Technologies
The University of Sydney

Masahiro Takatsuka†
ViSLAB
The University of Sydney

Richard Webber ‡
National ICT Australia Limited

Abstract

We have previously introduced a two-phase approach to visualize a multivariate network[5]. Positions of the graph nodes were determined by their attributes and binary connectivity (connected or not-connected). This paper presents the improved method: 1) graph distances are combined with the node attributes to improve the final positions of the graph nodes; 2) we use the uniform grid technique to speed up the disk layout which reduces the number of edge crossings and node/edge overlaps. We also provide an interface to generate component planes to depict how individual attribute changes across the network.

Keywords: Multivariate Network, GeoSOM, Crossing Reduction

1 Introduction

A multivariate network consists of connected data points, each of which has multidimensional attributes. Connections in the network describe relationships/activities between the data points. Many real world data sets can be represented using this type of network. One example is the world trade network in which countries are represented as nodes, each country has properties like gross domestic product (GDP), GDP growth, population and population growth, and edges represent imports/exports between the countries. Different trading activities, such as exchanging metal products or cereals, form different networks. We believe that visualizing a multivariate network can help users extract useful information from both aspects of the data.

In previous work [5], a two-phrase approach was introduced to visualize multivariate networks. The idea was to put together the graph nodes which are connected and also close in attribute space. Similarities of the graph nodes can then be observed based on the graph nodes' relative positions instead of comparing values of different attributes. In this article, we improve the original approach by introducing graph distances into Geodesic Self-Organizing Map (GeoSOM)'s training process. GeoSOM [4] determines the initial layout of the network based on data attributes and graph distances. We also use the uniform grid technique to reduce the computational complexity of the spherical disk layout. Furthermore, we add an interface to show the component planes of GeoSOM so that the user can examine the distribution of different attributes across the network.

2 Phrase One: Modify the Batch Training Process

Previously, we treat each graph node as a point in high-dimensional space and use GeoSOM to project the nodes onto the surface of a sphere [5]. Although this non-linear mapping tries to put together graph nodes who are neighbors in the graph, it has little effect on graph nodes which are two or more steps apart. We improve the projection by incorporating the graph distances.

*e-mail: chwu@it.usyd.edu.au
†e-mail:masa@vislab.net
‡e-mail:Richard.Webber@nicta.com.au

The batch mode training of self-organizing map [2] is an iterative process: At each iteration, every input $\vec{x}$ finds the best matching neuron (BMN) whose weight vector is nearest to it. Subsequently, each weight vector w_j of neuron n_j is set to be the mean over all the inputs registered with it and its neighboring neurons:

$$\vec{w}_j = \frac{\sum_{i=1}^{N} h_{b_i j}(t)\vec{x}_i}{\sum_{i=1}^{N} h_{b_i j}(t)} \tag{1}$$

where b_i is the BMN of input $\vec{x}_i$ and N is the total number of inputs mapped to the neuron n_j and its neighbors; size of the neighborhood is controlled by $h_{b_i j}$. According to equation 1, the contribution of each input to weight vector $\vec{w}_j$ is weighted by the neighborhood function $h_{b_i j}$. We add a coefficient f_{ij} to equation 1 to make the nodes who are close in graph space near each other. Here, a graph node v_i's attributes are denoted as a vector $\vec{x}_i$:

$$\vec{w}_j = \frac{\sum_{i=1}^{N}(1+f_{ij})h_{b_i j}(t)\vec{x}_i}{\sum_{i=1}^{N}(1+f_{ij})h_{b_i j}(t)} \tag{2}$$

f_{ij} is calculated from the average graph distance $d(i,j)$ of node v_i to all the nodes v_j mapped to neuron n_j:

$$f_{ij} = 1 - \frac{\sum_{v_j \in n_j} d(i,j)}{KD} \tag{3}$$

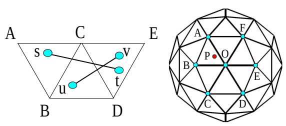
where K is the number of shortest paths connecting nodes v_i to those nodes that mapped to neuron n_j. D is the graph diameter.

According to equation 2, the closer v_i and v_j are in terms of graph distance, the larger f_{ij} becomes. Therefore, f_{ij} amplifies the contribution of nodes v_i which are closely connected to the nodes mapped to neuron n_j. In other words, adding the term f_{ij} has the effect of pulling together the nodes which are close in graph distances.

3 Phrase Two: The Spherical Disk Layout

After using GeoSOM to produce an initial layout of a multivariate network, we adjust the positions of the graph nodes to make the final layout visually more pleasing. The algorithm should separate the graph nodes which are mapped to the same point on the sphere and reduce the number of edge crossings and node/edge overlaps. A greedy iterative process was previously used in [5] to find an appropriate location for each graph node on a circle surrounding its BMN. The algorithm tried to place each graph node at one of the equally spaced positions on the circle, such that the number of crossings and node/edge overlaps is minimized. This process is repeated until the number of crossings and overlaps no longer changes. Calculating the number of crossings and overlaps takes time $O(VE+E^2)$ in each iteration. Here we employed the uniform grid technique [1] to speed up the algorithm.

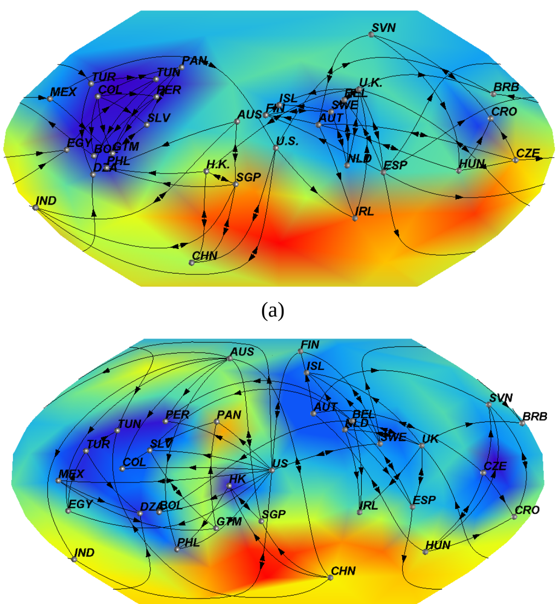
GeoSOM organizes its neurons on the grid of an icosahedron-based geodesic dome. We make use of the grid to divide the spherical surface into triangles (see Figure 1). These triangles are almost uniform in shape and size, thus can be used as a spherical "uniform" grid. Using this grid, in calculating the number of crossings, each edge only needs to compare itself with those edges who go through the same triangles. Similarly, for node/edge overlaps, each node only needs to be tested against the edges who go through the triangle where the node locates and the adjacent triangles. The time complexity of every iteration is then reduced to $O(E^2/V^{0.5}+V^{0.5}E+E)$. If the graph is sparse (E is $O(V)$), then the time complexity becomes $O(E^{1.5})$. If we the graph is dense (E is $\Theta(V^2)$), then the time complexity becomes $O(E^{1.75})$. The number of iterations for this algorithm to converge depends on the data set. In practice, it converges in a few iterations after GeoSOM's initial layout.



(a) (b)

Figure 1: (a) Edge *st* only need to compare itself with the edges going through the same triangles. (b) Each graph node only need to compare with the edges going through adjacent triangles.

4 Experiments: The World Trade Network



(a)

(b)

Figure 2: (a) The manufacture metal world trade network. (b) The cereal world trade network.

We use the world trade data set described in Section 1 to demonstrate our approach. It contains the trading between 32 countries in 1994 [3]. There are two relationships in this data set: trading of manufacture metal (62 edges) and trading of cereals (63 edges). The edges are directed, pointing from exporting countries to importing countries. In [5], we introduced an interface to project the spherical image of GeoSOM onto 2D plane so that a user can have a global view of the data map. Figure 2 shows the two world trade network. In both figures, U.S. are chosen to be the center. Each neuron is colored according to the average distances between its weight vector and those of neighboring neurons'. Colors change from blue (smallest variance between the weight vectors), through cyan and yellow, to orange (largest). Data mapped to the same blue region are close in attribute space and can be considered to lie in the same cluster.

Several attribute clusters can be observed from Figure 2, such as the developing countries (e.g. PER, COL) and developed countries (e.g. BEL, U.S.). CHN and IND are separated from the developing countries as they have the largest population. Different trading patterns can be observed in the two data maps. In the metal trading network, many activities happen between the developing countries such as PER, COL and BOL. In the cereal trading network, there are little trade between them — instead, all of them are importing from the developed countries, in particular U.S. and AUS. Therefore U.S. and AUS are pulled closer towards these countries.

Comparing Figure 2(a) and 2(b), we can see that the nodes' positions change because of different graph structures. However, the old and new positions are similar because they are also controlled by the node attributes. It is not difficult for a user to locate the same country in the two figures. We believe this can help users maintain their mental map while comparing various relationships of the same graph nodes.

Figure 2 gives a synthesized view of the world trade network. A user may also wants to know how individual attribute changes across the network. This can be done by the component planes: the neurons are colored according to an attribute selected by the user. Figure 3 shows the distribution of GDP growth. Color changes from white to red: white indicates the lowest GDP growth and red indicates the highest. From the figure, countries such as HUN, CRO and CZE have negative growth in GDP while CHN has the largest.

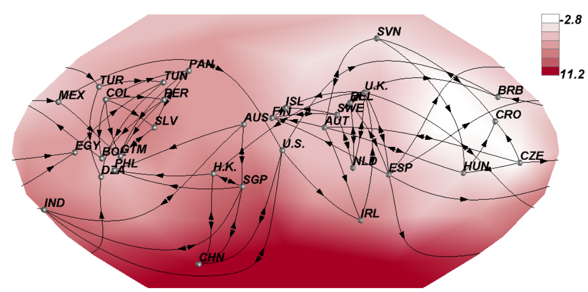


Figure 3: One of the component planes of the metal world trade network, showing the distribution of GDP growth.

5 Conclusion

We present the improved two-phrase approach to visualize the multivariate network. Initial layout of the network is produced by GeoSOM, which takes into account both of the node's attributes and graph distances. To speed up the spherical disk layout, we apply the uniform grid technique on the sphere. We also provide an interface to generate component planes so that a user can see the distribution of individual attribute across the network.

References

[1] V. Akman, W. R. Franklin, M. Kankanhalli, and C. Narayanaswami. Geometric computing and uniform grid technique. *Computer-Aided Design*, 21:410 – 420, 1989.

[2] T. Kohonen. *Self-Organizing Maps*. Springer Series in Information Sciences. Springer-Verlag, Germany, 3rd edition, 2001.

[3] W. Nooy, A. Mrvar, and V. Batagelj. *Exploratory Social Network Analysis with Pajek*. Cambridge University Press, New York, USA, 2005.

[4] Y. Wu and M. Takatsuka. Fast spherical self organizing map–use of indexed geodesic data structure. In *Proceeding of the Workshop On Self-Organizing Maps 2005 (WSOM2005)*, Paris, France, 2005.

[5] Y. Wu and M. Takatsuka. Visualizing multivariate network on the surface of a sphere. In K. Misue, K. Sugiyama, and J. Tanaka, editors, *Proceedings of Asia-Pacific Symposium on Information Visualization 2006*, pages 107–114. CRPIT, 2006.