The World of Music: SDP layout of high dimensional data

David Gleich* Stanford University
Matt Rasmussen† Massachutes Institute of Technology
Kevin Lang‡ Yahoo! Research Labs
Leonid Zhukov§ Yahoo! Inc.

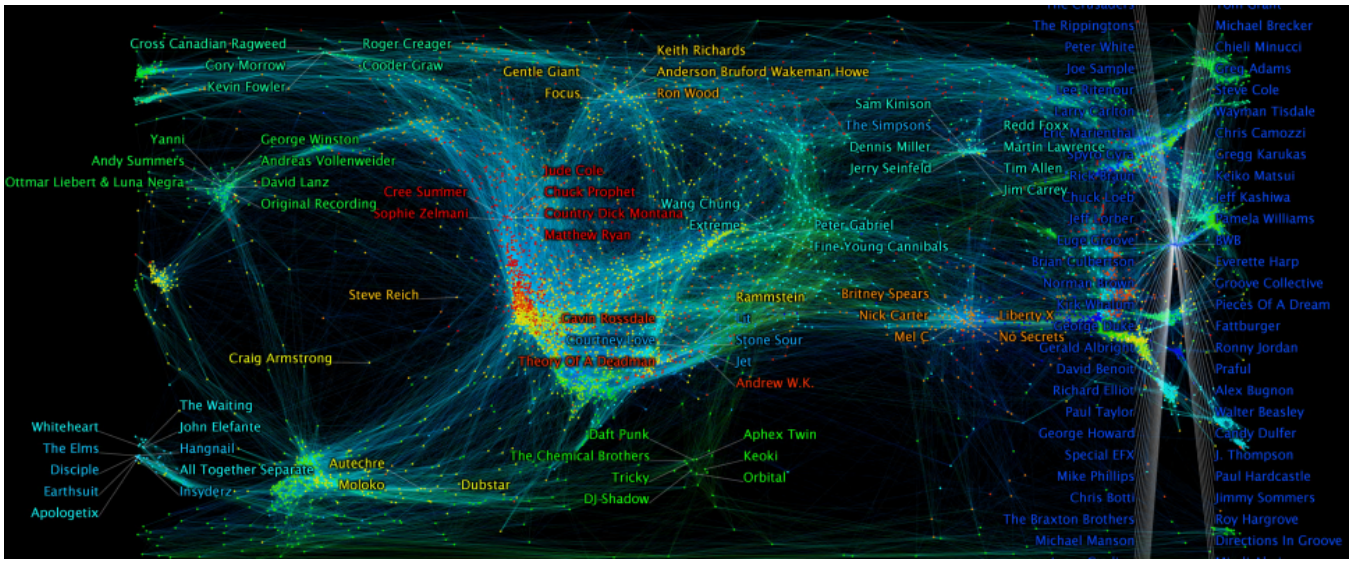

Figure 1: The World of Music: 2D layout of artist-artist similarity graph

ABSTRACT

In this paper we investigate the use of Semidefinite Programming (SDP) optimization for high dimensional data layout and graph visualization. We developed a set of interactive visualization tools and used them on music artist ratings data from Yahoo!. The computed layout preserves a natural grouping of the artists and provides visual assistance for browsing large music collections.

CR Categories: I.3.3 [**Computer Graphics**]: Picture/Image Generation - Display Algorithms I.5.3 [**Pattern Recognition**]: Clustering - similarity measures

Keywords: high dimensional data, graph layout, semi-definite programming

1 INTRODUCTION

Visual investigation of high dimensional data has always been a challenging endeavor. Usually, the data or a subset of the data is positioned in two or three dimensions for easy browsing and navigation. The most common approaches are *data slicing* – choosing several data directions to visualize – and *parallel coordinates* – simultaneously showing all data coordinates [6]. These methods, though easy to implement, have significant shortcomings, including loss of information and visual clutter. Another classical visualization approach for high dimensional data consists of finding low dimensional projections that capture the most important directions in the data. These methods are usually based on PCA or MDS. Recently, non-linear low dimensional embedding methods have attracted significant attention [5]. These methods, including LLE, Isomap and Laplacian Eigenmaps, attempt to reconstruct the actual manifold underlying the data [1]. Many interesting datasets involve relational data and can be represented as a graph, where nodes represent the data items and edges encode the relationships between them. Thus the data visualization task is closely related to the graph layout problem.

We consider visualizing a dataset from Yahoo! Music services that consists of user ratings of musical artists. This dataset obeys a power law distribution, i.e. there are few nodes with high degree and many nodes with low degree. This highly skewed distribution of degrees creates additional challenges for layout algorithms. Also, this graph possesses a significant number of clusters, or tightly connected groups of nodes; layout algorithms should reveal this structure.

In this paper we investigate an application of recently developed algorithms for Semidefinite Programming (SDP) [3] to assist in data layout. This SDP embedding method is very closely related to Laplacian Eigenmaps, but for certain "power law" graphs it makes better use of a small number of dimensions. Additional constraints force the embedding to lie on a hypersphere. The hypersphere, itself, can be used for visualization or can be further "unrolled" into lower dimensions.

*e-mail: dgleich@stanford.edu
†e-mail: rasmus@mit.edu
‡e-mail: langk@yahoo-inc.com
§e-mail: zhukovl@yahoo-inc.com

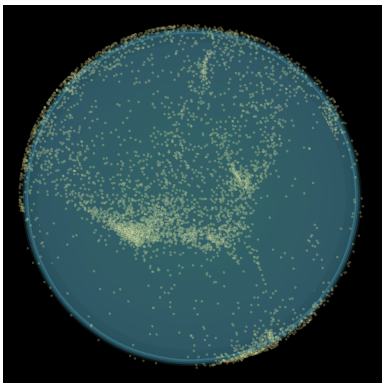

Figure 2: Data points embedded on the surface of 3D sphere

Figure 3: Zoomed in view of several clusters in layout

2 Data

The dataset used in this paper consists of all the ratings made by users on the Yahoo! Music service during a 30 day period. The full dataset contains 250 million ratings on 100,000 artists from 4 million users. The ratings are on a scale from 1 (dislike) to 100 (like).

We pre-processed the data by eliminating all ratings below 75 and considered only users and artists with at least 100 ratings. After these modifications, the new dataset contains 9,276 artists and 150,000 users with 2.5 million ratings.

3 Method: Layout generation

Our layout algorithm consists of several steps. We start by constructing an item-item (artist-artist) similarity graph based on ratings provided by users. The artists correspond to graph nodes and edges are established by the following procedure: we connect nodes *a* and *b* in this graph if *b* was one of the top *N* similar artists to *a or* *a* was one of the top *N* similar artists to *b*. To compute similarity between artists, we use the standard cosine similarity metric in a vector space model, where artists represents points (vectors) in the high dimensional "user" coordinate space. While cosine is a symmetric affinity function, the relationship "top *N* closest using cosine" is not symmetric. The above algorithm explicitly symmetrizes the graph using an "or" operation. Thus, an artist may have more than *N* connections in this similarity graph. After choosing $N = 20$, we then use the routine `CLUTO_V_GetGraph` from CLUTO [4] to construct this graph.

Next, we generate a low dimensional layout for the nodes of the similarity graph. We use an SDP embedding to accomplish this task. Our SDP is a quadratic optimization problem that tries to find a low dimensional embedding of the graph that minimizes the sum of the squared length of graph edges under additional constraints. This is a continuous relaxation of a Quadratic Integer Program encoding the Graph Bisection problem. SDP helps to get a more a uniform embedding by imposing stricter constraints compared to Laplacian Eigenmaps. These constraints prevent the algorithm from "pulling off" small pieces of the graph and leaving an unresolved lump of nodes. The constraints are equivalent to embedding the graph on a hypersphere instead of a line (or hyperplane). For an efficient numerical solution of the SDP problem we used the new low-rank method devised in [2] that can handle sparse graphs with more than a million nodes.

It suffices for our data to perform an embedding on the surface of a 3D sphere, Fig. 2. To generate a two-dimensional layout from the sphere, we *unroll* it by using the two angles of each point in spherical coordinates to determine the 2D layout. This procedure introduces significant distortion at the north and south pole of the sphere, exactly like Greenland and Antarctica are exaggerated on most maps. For clarity of visualization we prune long distance edges from the display.

To summarize, the main steps of our algorithm are:

1. Construct the nearest neighbor similarity graph.
2. Embed the graph on a sphere by solving an SDP.
3. Unroll the sphere for a 2d layout and visualize.

4 Implementation

The interactive visualization program is written in C++ using OpenGL. The nodes (points) and the edges (lines) are alpha-blended to show local density. This permits regions with many edges to show up with more intensity on the display, while regions with few edges show up as dark areas. The colors of the points were generated from an independent clustering of the artist and ratings dataset using CLUTO. Our interactive system allows for panning, zooming, searching for artists, and identifying nearest neighbors. Finally, we provide an interactive mode to choose the location of the poles and the splitting meridian used when unrolling the sphere.

5 Visualization Results

Fig. 1 shows the layout generated by the system for Yahoo! Music services. Fig. 2 demonstrates the intermediate step with data projected on the sphere. Finally, Fig. 3 is an enlarged version with some artist labels shown. The algorithm accurately separated the comedians in the dataset from the other artists.

References

[1] Mikhail Belkin and Partha Niyogi. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Comp.*, 15(6):1373–1396, 2003.

[2] Samuel Burer and Renato D.C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming (series B)*, 95(2):329–357, 2003.

[3] Michel X. Goemans and David P. Williamson. Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. *J. Assoc. Comput. Mach.*, 42:1115–1145, 1995.

[4] George Karypis. Cluto – a clustering toolkit. Technical Report 02-017, University of Minnesota, Department of Computer Science, 2002.

[5] J. C. Platt. Fast embedding of sparse music similarity graphs. In *Advances in Neural Information Processing Systems*, volume 16, pages 571–578, 2004.

[6] Spence R. *Information Visualization*. ACM Press, 2000.