The World of Music

SDP layout of high dimensional data

David Gleich
Institute of Computation and Mathematical Engineering
Stanford University

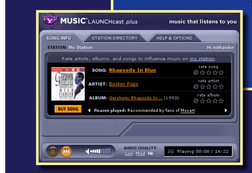
Leonid Zhukov
Yahoo! Inc.

Matthew Rasmussen
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology

Kevin Lang
Yahoo! Research

1. Dataset

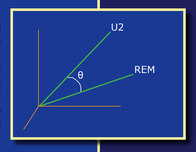
The dataset used consists of all the ratings made by users on the Yahoo! Music service during a 30 day period. Ratings are made by users in the LAUNCHCast player (lower left). The full dataset contains 250 million ratings on 100,000 artists from 4 million users. The ratings are on a scale from 1 (dislike) to 100 (like) (example at lower-right). 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.



Artist	User ID	Rat.
REM	1034047	20
U2	1034047	80
Aerosmith	1034047	60
Mozart	1034276	70
Beethoven	1034276	100
Chopin	1034276	90
U2	1034882	60
Aerosmith	1034882	50
Chopin	1034882	100
DJ Tiesto	1034882	80

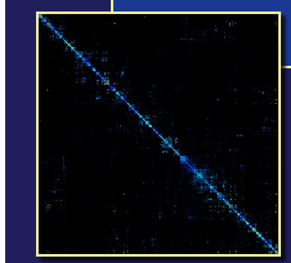
2. Similarity Graph

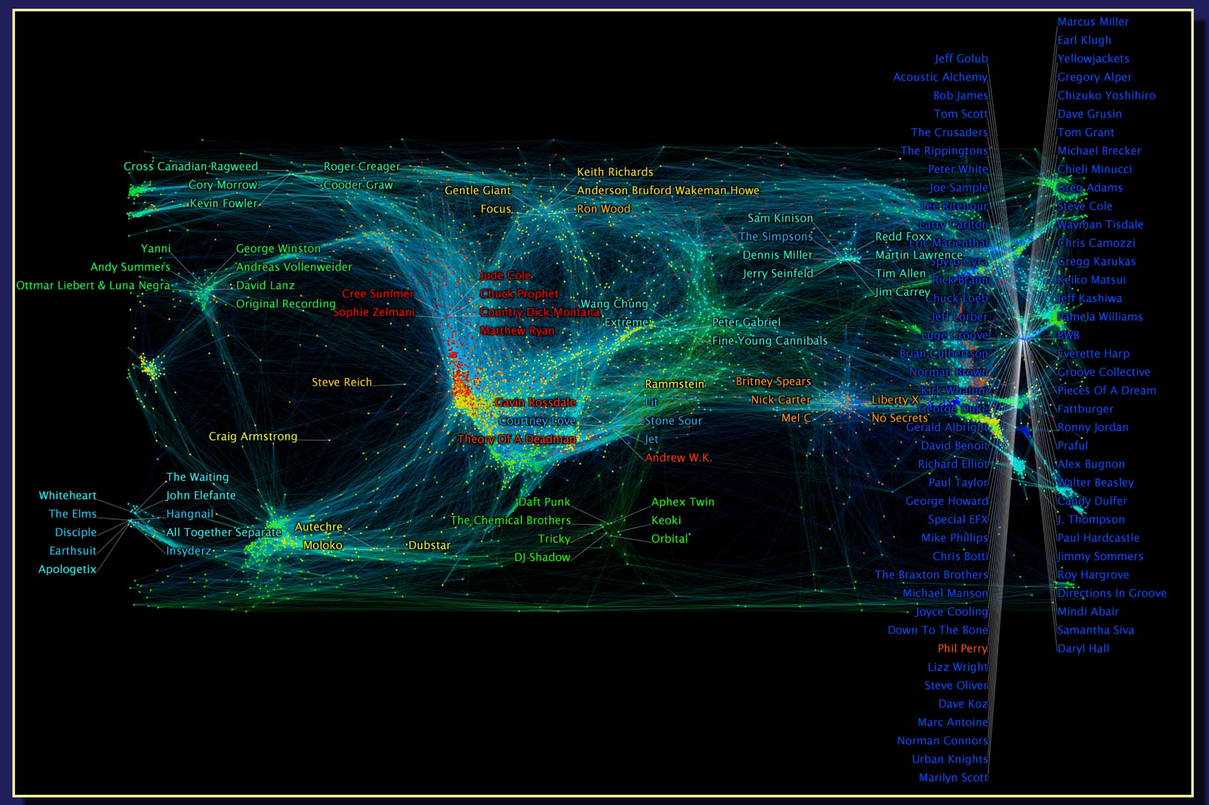
With the data, we constructed an item-item (artist-artist) similarity graph based on ratings provided by users. The artists correspond to graph nodes and edges are established by the following procedure: we connect nodes *a* and *b* in this graph if *b* was one of the top *N* similar artists to *a* or *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 represent points (vectors) in the high dimensional "user" coordinate space -- see the illustration at right. While cosine is a symmetric affinity function, the relationship "top *N* closest using



$$\cos(a_i, a_j) = \frac{a_i^T a_j}{\sqrt{||a_i|| ||a_j||}}$$

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. We use $N = 20$ and call the resulting weighted adjacency matrix W (pictured to the left).



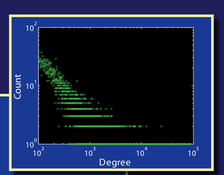


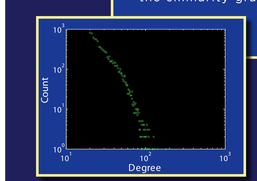
3. Dataset Statistics

Our dataset displays a power-law both in the original data and in the similarity graph. In a graph of *n* vertices, a power-law with exponent γ has c_k vertices with degree k.

$$c_k = nk^{-\gamma}$$

The upper-right figure is a histogram of the number of users rating each artist in the original dataset. The figure at the lower-left is a histogram of the degrees of each artist in the similarity graph. The maximum degree artist in the similarity graph was "Toby Slater" with 181 connections. In the original dataset, the "Red Hot Chile Peppers" were the most popular with 86,658 user ratings above 75.





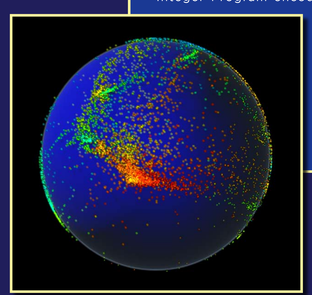
4. Layout

With our similarity graph and weighted adjacency matrix W, we compute a layout using a semi-definite embedding. This program comes from a semi-definite relaxation of the minimum bisection problem. Our SDP (below) is a

$$\mathbf{L} = \underbrace{\mathrm{Diag}(\mathbf{W}e) - \mathbf{W}}_{\text{Laplacian}}, \quad X = \begin{pmatrix} x_1, y_1, z_1 \\ \vdots \\ x_n, y_n, z_n \end{pmatrix}$$

$$\min_X \mathrm{tr}(\tfrac{X^T \mathbf{L} X}{4}) \quad s.t. \underbrace{\mathrm{diag}(XX^T) = e}_{\text{on sphere}}, \quad \underbrace{e^T(XX^T)e = 0}_{\text{center at 0}}$$

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. The 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). The layout for our similarity graph is shown on the sphere at lower-left. For an efficient numerical solution of the minimization problem, we used an existing low-rank method that can handle sparse graphs with more than a million nodes.



5. Display

To generate a two-dimensional layout from the sphere, we unroll it using the spherical coordinates of each point. This procedure introduces significant distortion at the north and south poles 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. Finally, we provide an interactive mode to choose the location of the poles and the splitting meridian used when unrolling the sphere.

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{pmatrix} = R_x R_y R_z \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

$$u = \arctan(\tilde{x}/\tilde{y})$$
$$v = \arccos(\tilde{z})$$

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 original dataset using CLUTO. Our interactive system allows for panning, zooming, searching for artists, and identifying nearest neighbors.