Interactive Poster: Visualizing high dimensional datasets using Partiview

Dinoj Surendran, Stuart Levy

A standard method of visualizing high-dimensional data (with several hundreds of dimensions) is reducing the dimensionality of the data to 2 or 3 using your favorite dimensionality-reduction algorithm, and then creating a scatterplot with data represented by labelled and/or colored dots. There are two problems with this approach that Partiview, the software exhibited in this demo, can solve.

Problem 1: Dots may not adequately represent data

For example, the image to the right shows how well a clustering algorithm (Laplacian Eigenmaps, Belkin & Niyogi 2001) works on a dataset of handwritten digits. Different digits are represented by 'points' of different colors. Points of the same color are clustered together, indicating that the clustering algorithm is doing fairly well at recognizing digits. However, what you really want to do is look at the places where the algorithm is not doing well, i.e. where differently-colored digits are near each other.

Partiview permits images to be placed at specific 3d coordinates, always facing the screen (the option to have a fixed orientation is also available). In the case where each datum is a handwritten digit, we can represent a digit by itself, i.e. an image of the original digit. Our first demo is precisely this; as evidenced from the two closeup shots below. The image above was a far-away shot of the same demo.

The below-right image has an example of how this kind of visualization can tell precisely how well this algorithm is separating the digits (or how badly people write digits). This algorithm clearly thinks 5’s with near-closed loops, are 6s, ditto poorly written 3s and 8s.

Our second demo shows precisely how well an algorithm (Locality Preserving Projections, He & Niyogi 2002) works at grouping pictures according to their semantic content. As the closeup below left shows, it achieves some success with grouping certain kinds of images, such as those of flags,
together. The rest of the interactive demo shows that other groups do not fare as well.

In cases where the data points represent something clearly visual, such as faces in face recognition applications, or documents (or part thereof) in information retrieval applications, the choice of images for glyphs is straightforward. Many applications do not have this; other methods, such as summary data (e.g. a pie chart showing the relative frequency bigrams in a ACGT string), could be used.

How many images or points can Partiview handle, and still give a user a smooth interactive experience (allowing them to zoom/rotate/translate the data)? This depends on what computer is being used, in particular how much memory it has. A recent laptop with a gigabyte of RAM can handle a few hundred thousand points, and/or a few thousand images.

Problem 2: Why reduce to just three dimensions?

If the data’s inherent dimensionality is larger than three, reducing the number of dimensions to 3 loses (and in the worst case, distorts) information. Besides, we often find in practice that the first three dimensions produced by our favorite dimensionality-reduction algorithm capture some feature of the data that is not useful for our needs.

It would be much nicer if the algorithm only had to reduce the dimensionality of the data to, say, 10, and a good way of visualizing 10-dimensional data was available. One way of doing this is to view, as XGVis does, three of the ten dimensions at a time. A more general way, that Partiview supports, is to be able to specify a 10 x 3 matrix that defines each spatial dimension as a weighted sum of the 10 dimensions. Weights can make use of human intuition in a semi-supervised form e.g. a few pairs points can be marked by a human expert as being similar or dissimilar.

Other features

Partiview supports animations; data points can be turn on and off at specified times. Further depth cues can be obtained by using its stereo capabilities, including red-blue, side-by-side (as required for GeoWalls), and chromadepth. Data points can be organized into groups, which can then be turned off and on with a press of a button. Data points can also be turned on and off based on the values of certain fields associated with them. Lines can be drawn between points, permitting 3d graphs to be drawn. Finally, Partiview runs on Linux, Windows, and MacOS X.
Author affiliations: Dinoj Surendran (dinoj@cs.uchicago.edu) is a PhD Student in the Computer Science Department at the University of Chicago. Stuart Levy (slevy@ncsa.uiuc.edu) is a Researcher at the Experimental Technologies Division of the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign.

Acknowledgements: Thanks to Misha Belkin and for providing details of his algorithm’s results on the MNIST/LeCun dataset to visualize, and Xiaofei He for access to the Corel Image Database. Thanks also to Misha Belkin, Partha Niyogi, Mike Papka, Rick Stevens, and Mark Subbarao for useful discussions.