Assessing Projection Quality for High-dimensional Information Visualization

Penny Rheingans Marie desJardins

Department of Computer Science and Electrical Engineering

University of Maryland, Baltimore County

$[rheingan \mid marie]@cs.umbc.edu$

October 22, 2002

Introduction

Many information visualization applications involve high-dimensional data which needs to be projected down to a two- or three-dimensional display space before it can be visually represented. Many methods have been developed to perform this dimension reduction, but little quantitative analysis has been conducted of the impact of the projection technique selected on the resulting visualization. We propose three key characteristics of a good data projection and define quantitative metrics to measure each. We then use these metrics to analyze the performance of three commonly used types of dimension reduction on an example application.

1 Related Work

A wide variety of techniques have been developed to perform dimension reduction of high-dimensional data. These include parallel coordinates [6], multiparameter icons [9], and a host of interactive techniques developed by dynamic statistics researchers [2]. Many of these approaches only work for discrete data instances, rather than the potentially continuous model characteristics we discuss here.

There are also other techniques that produce clusters in 2D space based on the similarity of data instances in the higher dimensions. These techniques include multi-dimensional scaling [4] and relevance maps [Assa *et al.* 1997]. Other applications of SOM techniques to information visualization include the visualization of customer characteristics [11].

Although many researchers have studied techniques for visualizing data sets, and others have developed techniques to view model structure directly, there has been relatively little effort focused on visualizing learned models in the data space. A notable exception is the MineSet data mining package [12], which includes several techniques for visualizing models, such as scatterplotting of misclassified instances. The display space is generated by manual variable selection, so the behavior of the complete model can be difficult to perceive. Visualization of classifiers in the MineSet framework was described by [1].

2 Projection Quality Characteristics

To support high-dimensional data exploration, the display space should represent the data space in such a way that the structure of the data can be clearly visualized, and that the properties or regions of the data space are preserved. We have identified three basic characteristics that an ideal projection method for high-dimensional data exploration analysis should exhibit: region preservation, specificity of representation, and feature smoothness.

Region preservation means that homogeneous regions in the data space should correspond to contiguous regions in the display space. Specifically, points that are close to each other in the data space should map to points that are close to each other in the display space. Another way of saying this is that data points that are expected to be similar should be close to each other in the display space. Especially for data spaces with data value continuity, a projection which preserves data space regions will preserve the data space structures by not aggregating heterogeneous regions.

Any projection in which a continuous space is mapped to discrete display locations will have multiple data space locations mapping to the same locations in the display space. A projection is considered to be *specific* if the set of data space locations that map to a given display space location are “close” to each other in the data space. This criterion is roughly the inverse of region preservation, in which nearby points in data space should correspond to nearby points in display space. (However, a given projection may be both specific and region-preserving.) A specific projection has an advantage over a nonspecific projection because it is less likely to break up homogenous data space regions.

Feature smoothness is the third desirable characteristic of a projection. Feature smoothness simply means that the data space characteristics to be visualized (value distribution, isolevels, discrete instances, and meta-attributes) should be smooth and easily visible in the display space.

For each of these ideal projection characteristics, we have defined a pair of metrics that indicate the degree to which a projection or individual projected locations manifest that characteristic. These metrics will be presented and discussed in later sections.

3 Continuous High-dimensional Data Spaces

3.1 Data Space Characteristics

We have identified a general set of *data space characteristics*—i.e., properties of a data set that may be visualized in order to understand and analyze its structure. Identifying projection and visualization techniques that enable the user to understand and interpret these characteristics is a key aspect of our research. Some of the characteristics we have explored are:

Value distribution: For a given location in the data space, there is a value (or potentially multiple values). **(i.e., class)** The set of these values represent a function defined over the data space. Since the data space is continuous, there are a potentially infinite number of such values. These values are likely to exhibit continuity, but such continuity is not assumed or required.

Isolevels: In many applications, isolevels of constant data value are of particular interest. These isolevels frequently divide the data space into meaningful regions, corresponding to a type of segmentation. Maintaining connectivity of isolevel curves makes regions easier to recognize.

Discrete instances: In addition to the continuous value distribution, there may be discrete instances of interest. These discrete instances are the more tradition data items of high-dimensional information visualization, made up of individual points in the data space which may have associated values. An example might be the real-world samples used to validate a simulation.

Meta-attributes: There are characteristics of the data that are not directly reflected in the data values. In particular, we are interested in understanding such things as the provenance of data values and the confidence associated with them.

3.2 Example Domain: Predictive Model Quality

Throughout this paper we will use a single example domain, that of a continuous space defined by a predictive model of gas mileage derived from discrete instances. A model is a description of how the world is expected to behave. Typically a model describes the aspects of the world that are relevant to a specific task: e.g., diagnosing a disease, predicting credit risks, or classifying documents by topic area. Here we focus on *classification* tasks, which have the form "Given an object description, classify it into one of *k* classes." Classification methods can be used for both prediction and diagnosis (e.g., "Given an applicant's characteristics, predict whether they will default on a loan," or "Given a patient's symptoms, determine what disease is affecting them"). Probabilistic classification methods give the *probability* of class membership, which is particularly useful in domains containing uncertainty, noisy data, or incomplete object descriptions.

The problem of accurately predicting class membership from available information is a key challenge of knowledge discovery. A wide variety of methods have been developed by machine learning and data mining researchers to solve this problem, ranging from decision-tree learning algorithms to nearest-neighbor techniques to Bayesian learning methods.

In classification problems, one of the variables is a distinguished *class variable*; we refer to the other variables as *input variables*. (The class variable can be thought of as the dependent variable; the input variables, as the independent variables.) The *data space* is the N-dimensional space defined by the N input variables. In a classification task, the goal is to derive the *class probabilities*, i.e., the marginal probabilities that an instance belongs to each class, given values for some (or all) of the input variables.

We have developed visualization techniques which are applicable to any learning methods whose output makes predictions that can be interpreted as probabilities, such as probabilistic decision trees or Bayesian networks [10]. The visualization techniques are designed to support model analysis by clearly displaying important characteristics of the model, highlighting potential flaws in the model, and enabling comparison of alternative models.

In the examples given in this paper, we used the AUTO-MPG data set from the UCI Machine Learning Repository [13]. We applied Tree-Augmented Naive Bayes (TAN) [5], a Bayesian network learning system that is tailored for classification, to construct the models. Data instances contain eight variables (five continuous and three discrete) and a continuous class variable indicating mileage (mpg). We have discretized this class into high mileage (positive) and low mileage (negative), using the EPA's "gas guzzler tax" threshold of 22.5 mpg. Input variables include the number of cylinders, engine displacement, weight, model year, and horsepower. We used TAN to construct (from 294 training instances) a model to predict gas mileage classification.

In this application, the relevent data space characteristics are the properties of a model that may be visualized in order to understand and analyze its behavior. The characteristics listed below correspond to the general data space characteristics discussed earlier. Some specific characteristics of interest are:

Class probability: For a given point in the data space, we are interested in knowing the probability that it belongs to the class. For some model classes (non-probabilistic decision trees and associative rules), this probability will be given as zero or one; for probabilistic models, it will take on continuous values between zero and one.

Decision boundary: In making predictions, the model needs to be interpreted. For example, for using a Bayes net or probabilistic decision tree to solve a binary classification problem, we might use the decision rule that if the class probability is less than 25%, it is assumed not to be a member of a class; if greater than 75%, it is a member; and otherwise, the system gives a "not sure" answer. For a non-probabilistic decision tree, these boundaries will be very sharp. We would like to know where these decision boundaries fall, and in the case of probabilistic models, how varying the thresholds can affect the behavior of the classifier.

Misclassifications: In addition to knowing the overall prediction accuracy, we would like to understand *which* data points are misclassified. In addition, we would like to be able to assess the distribution of misclassification types (e.g., false positives vs. false negatives).

Meta-attributes: There are characteristics of the model that are not directly reflected in the class predictions. In particular, we are interested in understanding the distribution and density of the training data used to build the model and the confidence assigned to each estimate.

4 Data Space Projection

The data space for an information visualization application is frequently a high-dimensional space. For instance, a model predicting the probability of a car getting high mileage in the AUTO_MPG domain, which includes eight input variables, would inhabit a 8D data space. Each possible combination of variable values (i.e., each car of interest) corresponds to a point in this 8D space. Variable values, and therefore data space coordinates, may be continuous, ordinal, or nominal. Display spaces supported by most scientific visualization methods are two- or three-dimensional with continuous-valued coordinates. In order for a data space or an object inhabiting a high-dimensional data space to be visualized, it must first be transformed into a display space that is better suited for visualization. Information visualization applications, which are typically characterized by non-spatial, high-dimensional, non-continuous data spaces, generally combine some transformation of the data to a spatial display space with the application of representation techniques to transformed data points.

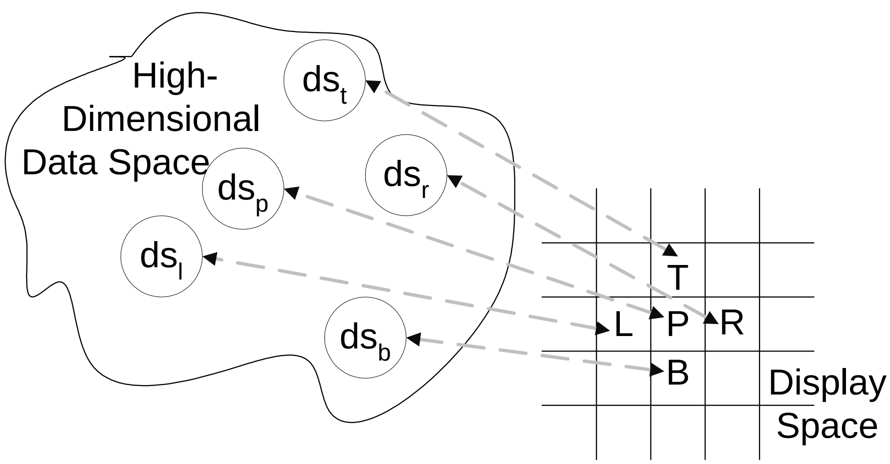


Figure 1: Forward and backward projection from data space to display space

In our discussion of data space projection we consider both *forward projection*, from data space to display space, and *backward projection*, from display space to data space. Figure 1 shows the forward and backward projection process for a 2D display grid. The grid locations L, R, T and B are the left, right, top and bottom neighbors of location P in the grid. The cluster of points

C = $\{ds_l, ds_r, ds_t, ds_b, ds_p\}$ are the locations in data space that map to the grid locations L, R, T, B and P respectively. The location ds_i in data space is a vector of length N, where N is the number of dimensions in the data space, and ds_{ij} is the value of the j^{th} coordinate. Therefore, ds_i = $\{ds_{i1}, ds_{i2}, ds_{i3}, ..., ds_{in}\}$.

We will analyze three types of methods for performing dimension reduction: feature selection, principal components analysis (PCA), and similarity clustering. For each method, we will consider both forward and backward projection. Each method is presented primarily in 2D in order to show the distribution of variable values across the space as clearly as possible in a static view. Performing dimension reduction to 3D using feature selection and PCA is completely analogous; in practice, we typically use 3D projections for the visualizations. 3D similarity clustering is theoretically straightforward, but we have not implemented it, since the building the 3D map would be computationally very intensive.

4.1 Feature Selection

Most visual data mining tools for high-dimensional data allow arbitrary data variables to be used as the coordinates in the display space. This method, called *feature selection*, can sometimes provide useful insights into the structure of the values in the data space. Using feature selection on data instances, each instance is plotted at the location determined by two (or three) variable values. For a 2D display space, the plane of the display space corresponds to an axis-aligned plane through the hD data space, with all points orthogonally projected onto the plane. Such views are most useful when the user has immediate and intuitive control over both variable selection and 3D viewpoint.

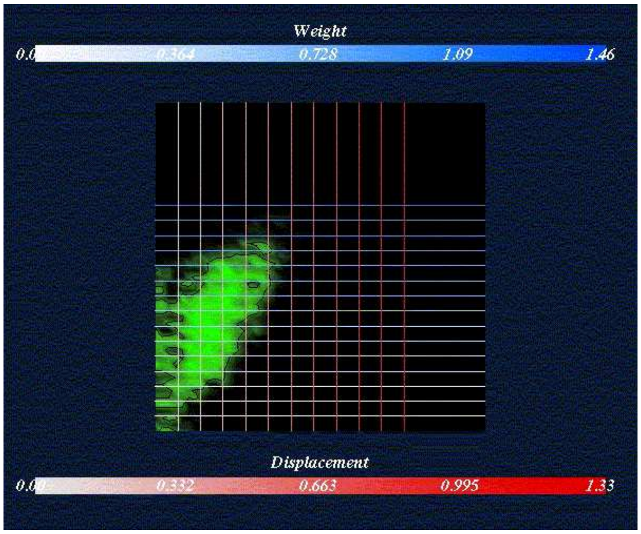
Figure 2(a) shows a 2D display space created using feature selection. The selected features, weight and displacement, are shown with colored contour lines. Each location in this display space represents a high-dimensional subspace in which weight and displacement are fixed, but other attributes can vary over their entire range.

Backward projection is under-defined for feature selection: since feature selection is an axis-parallel many-to-one mapping, each display space location corresponds to a hyper-region in data space. For models that support predictions about instances with missing values, the collapsed dimensions can be treated simply as missing values. When a specific location is required, we use the center of the hyper-region, given by the middle of each collapsed dimension.

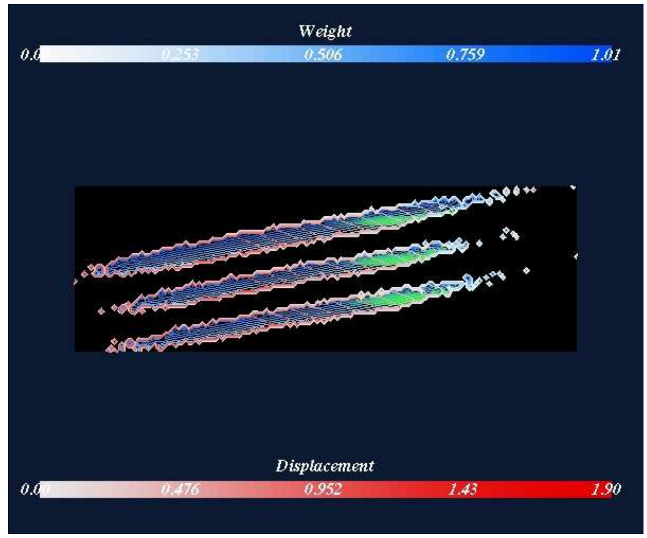
4.2 Principal Components Analysis

Principal Components Analysis (PCA) can be used to create a projection which captures more of the variability within the data space. The first principal component is a linear combination of data variable values accounting for the greatest variability. The second principal component is another linear combination of data variable values, orthogonal in the data space to the first combination and accounting for greatest amount of remaining variability. This continues for a total number of principal components equal to the original number of data variables. Locations in the data space map to display space coordinates given by their first two (or three) principal components. For a 2D display space, the plane of the display space corresponds to a plane through the hD data space, with all points orthogonally projected onto the plane. Unlike with feature selection, this plane need not be axis-aligned.

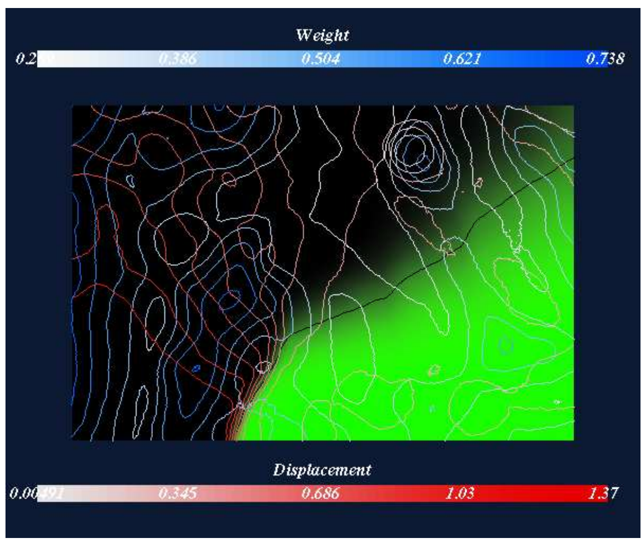
Figure 2(b) shows the display space defined by the PCA projection of a predictive model for the AUTO-MPG domain. Colored contour lines of weight and displacement show some of the attribute



(a) Feature Selection



(b) Principal Components Analysis



(c) Self-Organizing Maps

Figure 2: Mileage data space projected using three different methods. Increasing saturation of green corresponds to increasing predicted probability of high mileage. White line shows decision boundary.

correspondence between the data and display spaces. These contours generally show increasing weight towards the upper left corner and increasing displacement towards the lower left.

Backward projection is not analytically possible for PCA, since the system of equations controlling it is under-constrained. As an approximation, we have averaged the high-dimensional coordinates of the discrete instances mapping to a display space grid location. This approach is sensitive to the distribution of samples, particularly to the large unoccupied regions of the display space. We are experimenting with other methods of determining the backward projection.

4.3 Similarity Clustering

In both the feature selection and the PCA projections, the requirement that a 2D display space correspond to a plane in the hD data space (and a 3D display space correspond to a 3D subvolume) limits the degree to which the display space can span the data space. A planar display surface necessarily will pass very far from some regions of the data space. These regions will not be well represented in the display space, grouping with distant regions.

In order to achieve a display space which better represents the high-dimensional structure of the data space, we applied a set of projection techniques based on self-organizing maps (SOMs) [7]. In a SOM, neighboring locations in the display space correspond to neighboring locations in the data space, unlike feature selection, in which points that are far apart in the data space can map to the same location in the display space.

The SOM is initialized with a random *codebook vector* at each node, then the map is trained with a set of instances. For each map training instance, the map is searched for the most similar codebook vector, and the neighborhood around the matching codebook vector is altered to be more like the training instance. After the training cycle, neighboring locations in the SOM correspond to similar instances (i.e., instances that are close to each other in the data space). We are currently performing similarity clustering using a public-domain package that implements SOMs [8].

The dimensions of a display space created through similarity clustering have a highly non-linear relationship to the dimensions of the data space. Figure 2(c) shows how two data space dimensions have been warped by the projection process. The blue lines show contours of constant weigh, with more vivid blues corresponding to higher weights levels. The red and pink lines show contours of constant displacement, with more vivid reds corresponding to more displacement. In this display space, heavy cars tend to group to the left, while cars with large displacement tend to group to the lower left. These curving lines correspond to hyperplanes in the high-dimensional data space.

The backward projection for the SOM method is straightforward, since the codebook vector for each display space grid location gives the data space coordinates for the point.

5 Projection Quality Metrics

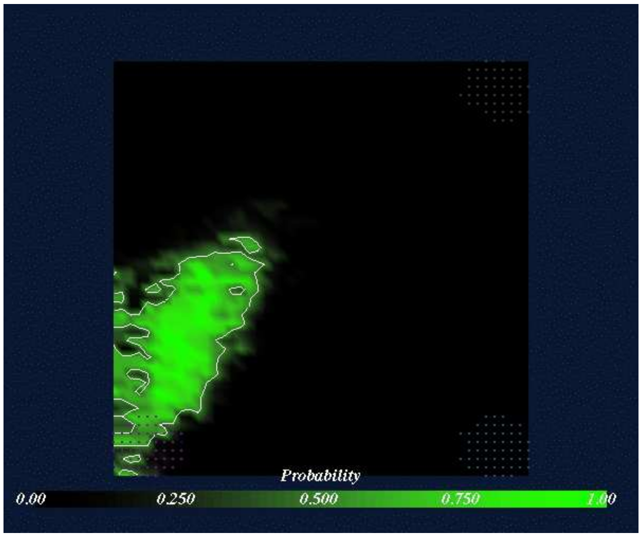
We have identified region preservation, specificity, and feature smoothness as key characteristics of a data projection technique well-suited for visualization purposes. Analyzing a specific projection technique for its attainment of these characteristics requires a qualitative assessment metric. In this section, we define such metrics for each identified projection quality characteristic.

5.1 Region Preservation

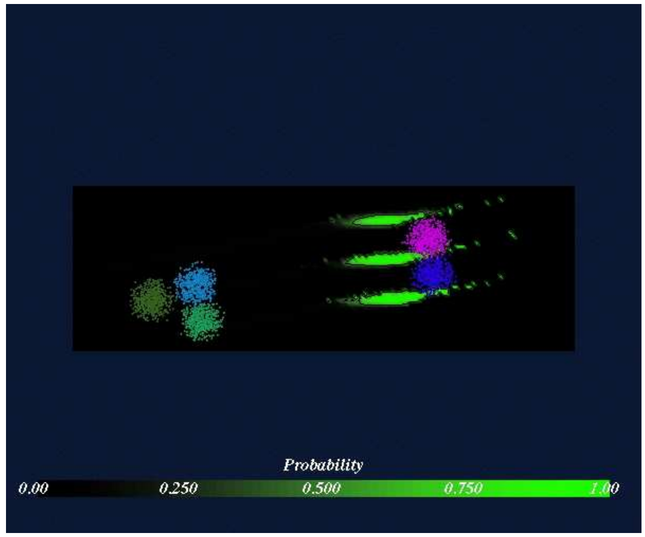
The region preservation metric addresses the need for a projection technique to preserve the regional relationship of points in the data space. To maximize region preservation, the dimension reduction technique should place points that are similar (close in data space) near each other in the display space. Formally, region preservation means that homogeneous regions in the data space should correspond to contiguous regions in the display space. We consider a simple proximity model of a region: namely, a *region* is defined to be a neighborhood of data space locations that are close to each other, measured using a high-dimensional Euclidean distance norm. A more semantic model of a region, based on values and features, is addressed by the feature smoothness metric discussed later.

Visualizing how individual data space regions projected to the display space gives an intuitive measure of the degree of region preservation of a projection technique. For each desired cluster center, a random cloud of points is generated within a specified radius in data space. Each of these clusters is then projected to the corresponding locations in the display space. The cluster can be thought of as a sampling of the high-dimensional sphere centered at the cluster center. Ideally, after projection to data space, the cluster should be compact and roughly circular.

We generated five clusters of 500 points each, projected them by the three methods, and displayed each in a unique color. Figure 3 shows the effect of each of methods on the clusters. The clusters projected by FS and PCA retain their circular shape. This can be attributed to the fact that in both methods the instances are orthogonally projected onto the plane in data space corresponding to the display place. By contrast, certain clusters projected by the SOM are warped considerably. Many of the clusters break apart during the projection process and land in completely different parts of the display space.



(a) Feature Selection



(b) Principal Component Analysis

(c) Self-Organizing Map

Figure 3: Warping of Clusters

A more quantitative measure of region preservation may be calculated by treating the data space location of each display space grid location as a cluster center. Each of these clusters is then projected into display space. The mean distance d_{mean} of each of the projected points from the display space cluster center is computed by:

$$d_{mean} = \frac{\sum_{i=1}^{m} dist(c, p_i)}{m} \quad (1)$$

where m is the number of points in the cluster, c is the center of the cluster, p_i is the i^{th} point in the cluster, and dist(c, p_i) is the distance in display space between c and p_i.

This metric shows how region preservation varies across the display space. For FS and PCA, region preservation is constant. Figure 4 shows how it varies across the SOM display space. Quite a large percentage of the image is black, indicating that for most of the display space the mean distance for each cluster is quite small. However, some parts of the display have high mean distance values, which corresponding to points within a cluster being projected to distant locations in the display space. The maximum value for the distance is 244 grid units which is greater than half the height of the grid. On careful examination we observed that the regions with large distance values mapped very closely to the variations in values of the *origin* data variable (see Figure 2(c)). We concluded that the SOM was sensitive to even slight changes of the value of the original data variable and therefore explained the high mean distance of the clusters from their centers.

Computing this quantitative region preservation metric is very expensive, with asymptotic performance of O(pqm), where p and q are the width and height of the grid and m is the number of samples in the cluster.

5.2 Specificity of Representation

Because dimension projection methods necessarily maps a volume in the high-dimensional data space to each discrete location in display space, they may aggregate heterogenous locations. An ideal projection method would not map highly dissimilar (distant) instances to the same location.

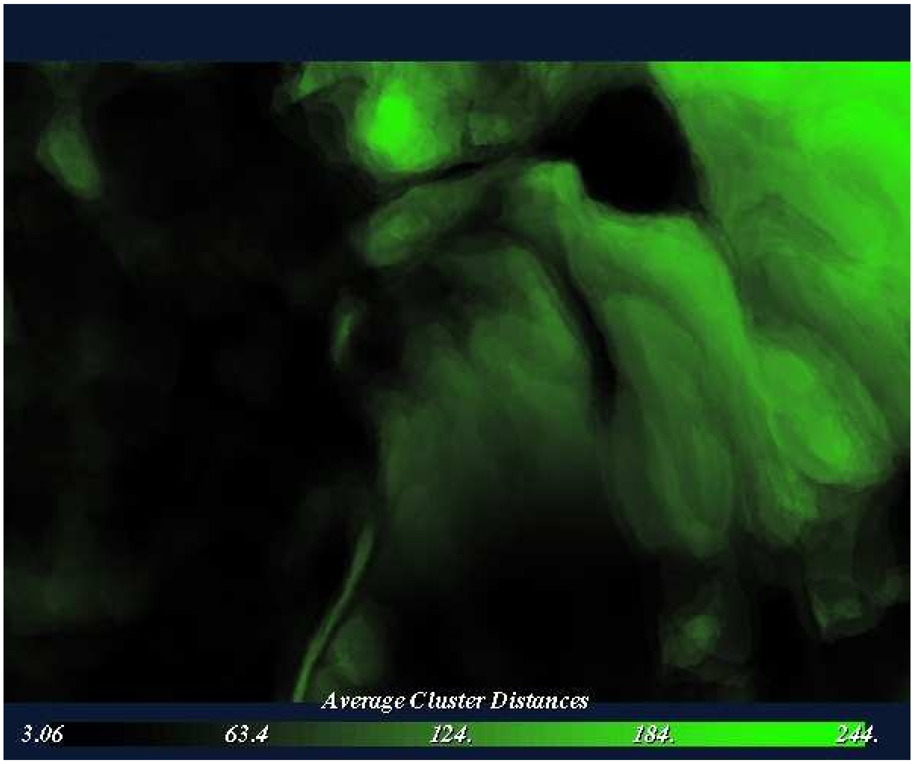


Figure 4: Mean Distance from the Cluster Center for the SOM

This leads to an intuitive definition of the specificity of representation of such techniques. A projection is considered to be *specific* with respect to a data space if the set of points that map to a display space location are close together in the data space. The specificity of a projection is conceptually the inverse of region preservation, corresponding to the region preservation of the backward projection from display space to data space. As with region preservation, we used a measure based on proxity. We measure the notion of "closeness" with two metrics: *bounding gradient* and *bounding volume*. The bounding gradient measures the closeness of the neighbors by computing the display space gradient for the high-dimensional Euclidean data space distance between neighboring points. The bounding volume is a measure of the high-dimensional axis-aligned bounding box that encloses the neighbors.

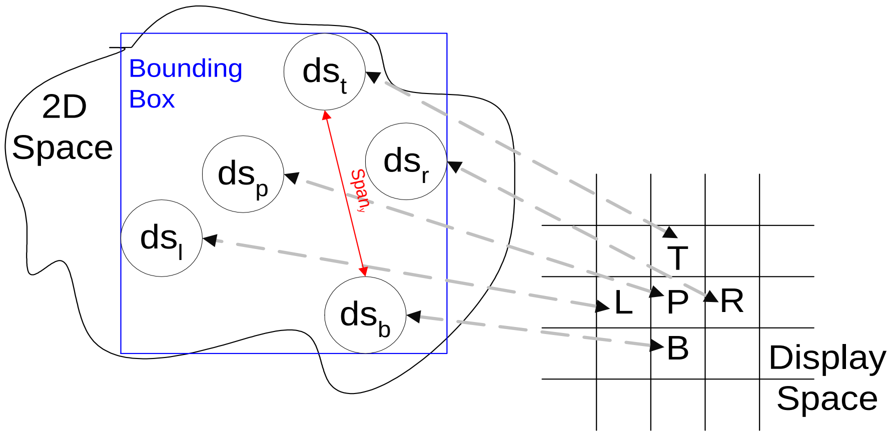


Figure 5: Conceptual view of bounding gradient and bounding volume for 2D data space

5.2.1 Bounding Gradient

The bounding gradient corresponds to the gradient magnitude of the data space distance field at a point in the display space. The distance field gradient at a point is calculated in a manner similar to that of central differences, with components along each of the axes, $Span_x$ and $Span_y$ (and $Span_z$ for a 3D display space), which are calculated as follows:

$$Span_x = |\Delta(ds_r, ds_l)| \quad \text{and} \quad Span_y = |\Delta(ds_t, ds_b)| \tag{2}$$

where the distance operator $\Delta(ds_a, ds_b)$ is:

$$\Delta(ds_a, ds_b) = \sqrt{\sum_{k=1}^{n} (ds_{ak} - ds_{bk})^2} \tag{3}$$

The 2D bounding gradient is the magnitude of this distance field gradient, specifically:

$$BGrad = \sqrt{{Span_x}^2 + {Span_y}^2} \tag{4}$$

The bounding gradient gives a measure of how rapidly data space coordinates are changing for unit increases in display space coordinates. Low gradient values indicate similarity in coordinate values, make it an appropriate measure of the specificity of representation.

We computed the bounding gradient values for each of the projection techniques applied to the automobile data space (Figure 6). The gradient is constant for both FS and PCA (considering only occupied regions of the PCA display space, with the difference in value resulting from the different inverse projection methods used (specifically, the assumption of a median value for all collapsed dimensions in FS). The bounding gradient image of the SOM display space shows a smaller maximum than either FS or PCA and large regions with a small gradient. Rapid changes in data space location are concentrated in small isolated regions.

5.2.2 Bounding Volume

The bounding volume is the high-dimensional volume in data space of the axis-aligned bounding box that contains the cluster C. It provides a spatial measure of specificity of representation whereas the bounding gradient is a one-dimensional measure (only length). The two measures distinguish between clusters on the basis of eccentricity, with bounding diameter high for those elongated along one dimension while bounding volume is highest for those elongated in several.

The volume of the bounding box is calculated as:

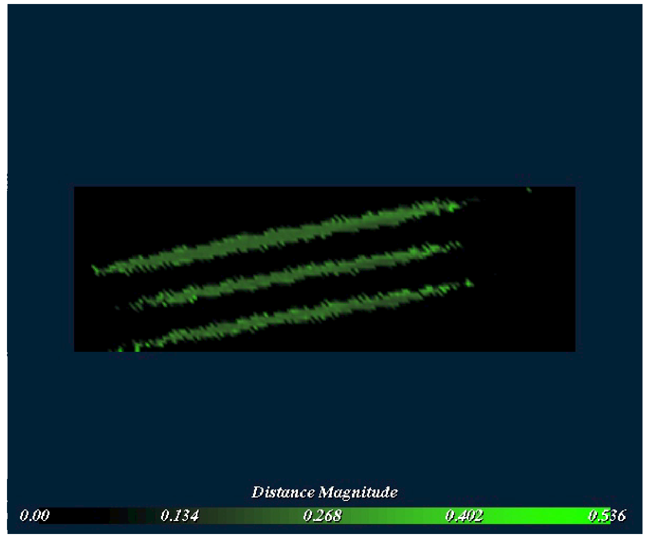
$$Volume = \prod_{k=1}^{n} (Max_k - Min_k) \quad (5)$$

where Min_k and Max_k give the minimal and maximal values of each coordinate k for the cluster C, computed as:

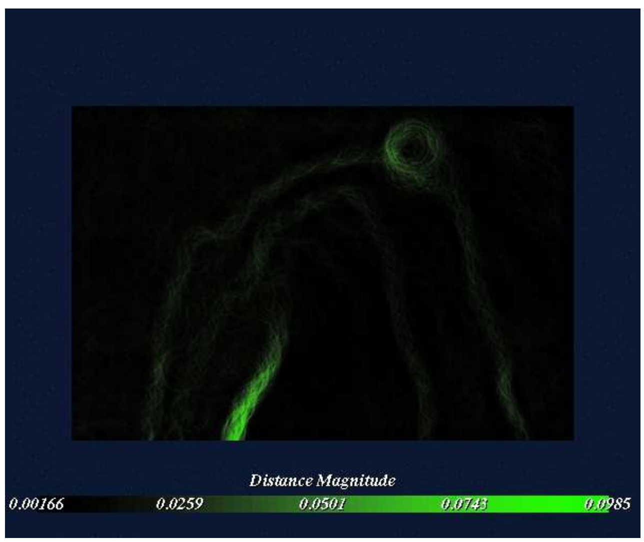
$$Min_k = min(min(ds_{rk}, ds_{lk}), min(ds_{tk}, ds_{bk})), \qquad where\ \ k = \{1, 2, ..., n\} \quad (6)$$

$$Max_k = max(max(ds_{rk}, ds_{lk}), max(ds_{tk}, ds_{bk})), \qquad where\ \ k = \{1, 2, ..., n\} \quad (7)$$

(a) 2D Feature Selection



(b) 2D Principal Components Analysis



(c) Self-Organizing Maps

Figure 6: Bounding gradient values over 2D display grid

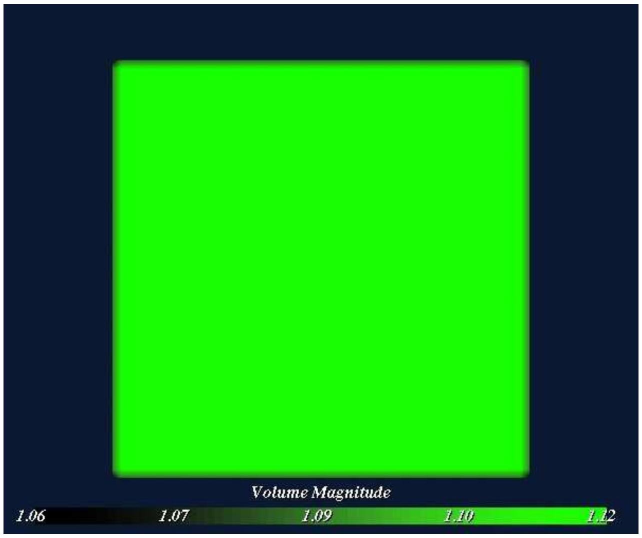
Because this measure represents the spatial spread of the points in the neighborhood, low volume values indicate tight bounding boxes and therefore greater specificity of representation. This measure was computed for the three projection techniques applied to the automobile data space. The volume of the bounding box for each of the display space locations in each projection is shown in Figure 7.

The bounding volume for each display space location projected by FS is constant, corresponding products of the ranges of the unselected dimensions. The bounding volume for locations projected by PCA is also constant, with the larger maximum reflecting that the volume projected to a location using PCA is not generally axis-aligned, resulting in an over-estimation using the bounding volume metric. By contrast, the bounding volume of a SOM display space projection varies greatly from location to location, reflecting the highly non-linear nature of the projection. The maximum bounding volume for SOM is similar to that for FS, but most display space locations show a substantially lower bounding volume, indicating much greater specificity.

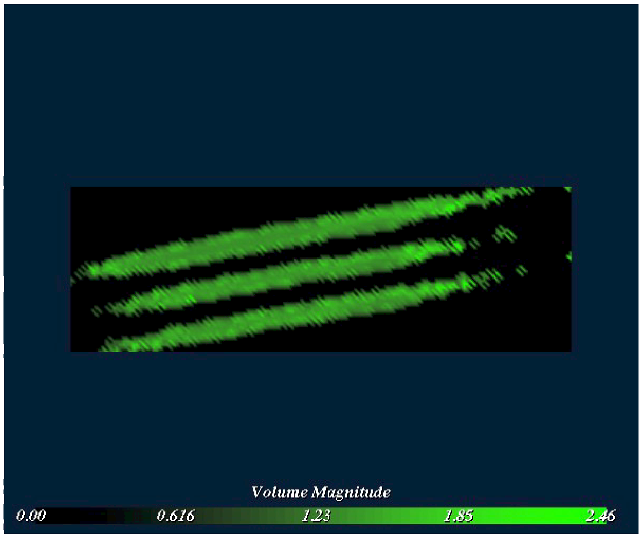
5.3 Feature Smoothness

The two previous projection metrics reflect the compression of the data space into the display space, but do not directly address the homogeneity of the compressed regions. Intuitively, compressing a region which is uniform with respect to a relevent characteristic is less worrisome than compressing one with much variability. The feature smoothness of a projection provides an indication of whether the characteristics of aggregated regions are similar and whether continuous smooth distributions are still smooth after projection to the data space.

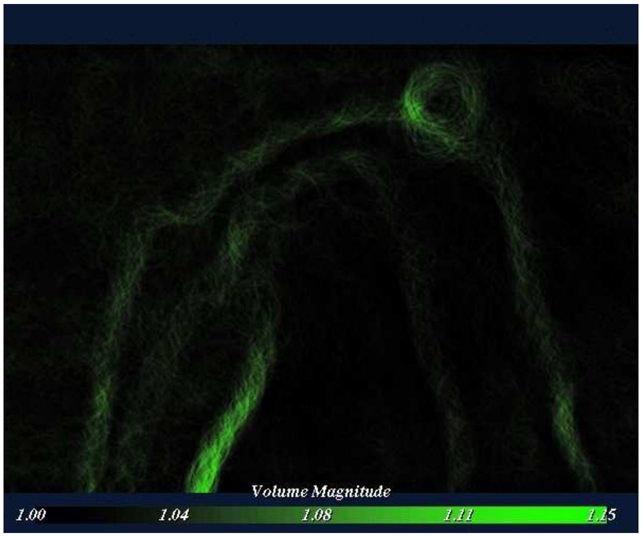
We define two metrics for feature smoothness: **standard deviation** of value and **neighborhood**



(a) Feature Selection



(b) Principal Components Analysis



(c) Self-Organizing Maps

Figure 7: Bounding volume values over the 2D display grid

autocorrelation. The standard deviation of the values displayed at a location is an indicator of the homogeneity of the region compressed to that location. The neighborhood autocorrelation measures the similarity of display space regions in terms of their characteristics, indicating the smoothness of these characteristics after projection.

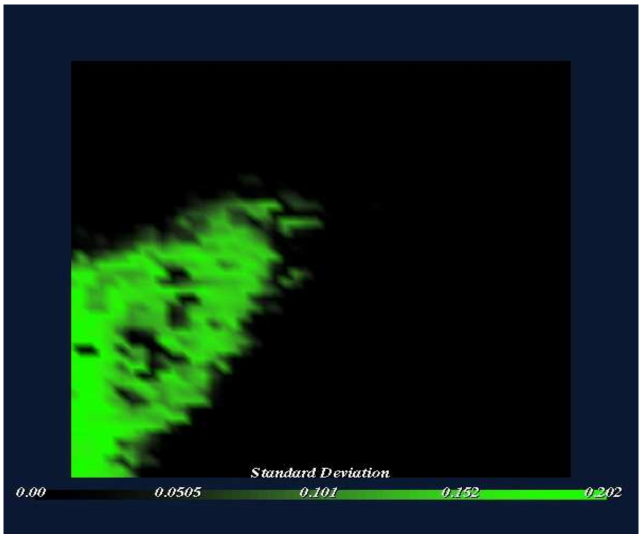
5.3.1 Standard Deviation

The standard deviation of sample values that are projected to each display space grid location provides an objective measure of the similarity of points which have been grouped together. The standard deviation distributions are shown in Figure 8. One can clearly see that the standard deviations for the PCA are higher than those of the FS and SOM, which means that points dissimilar in value have been aggregated.

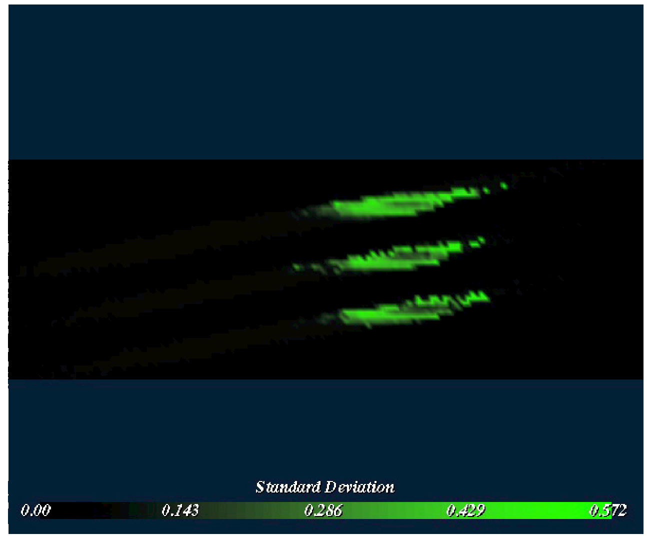
5.3.2 Neighborhood Spatial Autocorrelation

The standard deviation provides a measure of the variability of data space characteristics at each point, but not the similarity of neighboring locations in the display space. We develop our metric for feature smoothness from the concept of spatial autocorrelation used in cartography to measure the similarity of neighboring locations.

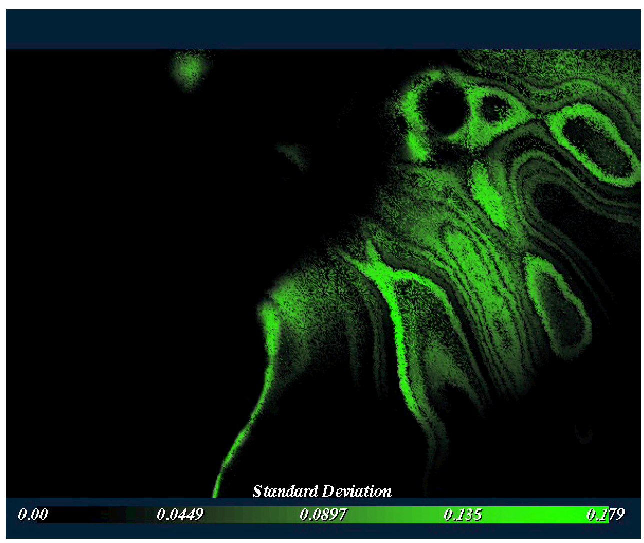
We base our metric on the spatial autocorrelation model discussed in [3]. Spatial autocorrelation, as used in cartography, computes a single value for the entire region of interest (ie. the whole display space). In order to produce an autocorrelation measure sensitive to differing similarity in different regions of the display space, we calculate local autocorrelation values for the one-neighborhood window over each location of the display grid. Each value is computed as fol-



(a) Feature Selection



(b) Principal Components Analysis



(c) Self-Organizing Maps

Figure 8: Standard Deviation values shown across the 2D display grid

lows:

$$Autocorrelation = \frac{n}{2 \times joins} \frac{\sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} \delta_{ij} v_i v_j}{\sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} v_i^2} \tag{8}$$

and the dissimilarity is computed as follows:

$$Dissimilarity = \frac{n-1}{4 \times joins} \frac{\sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} \delta_{ij} (x_i - x_j)^2}{\sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} v_i^2} \tag{9}$$

where:

n is the number of elements in the window

joins is the number of elements that are adjacent to the center of the window

x_i is the value at the i^{th} location in the window

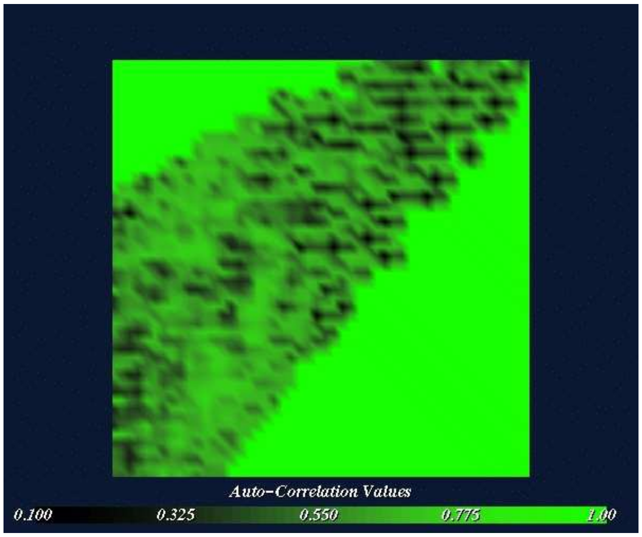
v_i is the mean-shifted value at the i^{th} location in the window

$\{\delta_{ij}\}$ is a connection matrix in which $\delta_{ij} = 1$ if the i^{th} and j^{th} elements share a common edge and $\delta_{ij} = 0$ otherwise

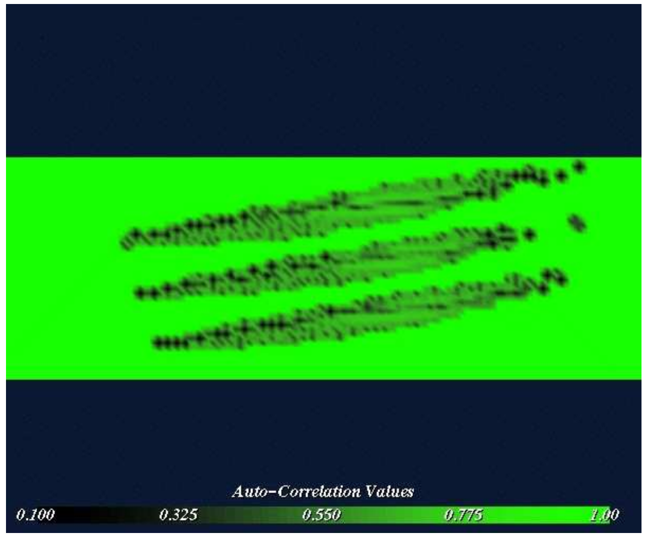
The autocorrelation (Equation 8) is the product of a scaling factor (the fraction on the left) and another fractional value to the right. This fraction is conceptually equivalent to $\sum_{i=1}^{n} \frac{z_i}{z_j}$, where z_i is the mean shifted value at the i^{th} location in the window and z_j is the center of the window. Hence it can be thought of as a ratio, so values close to one indicate similarity and values away from one indicate dissimilarity. The smoothness of the display space is directly proportional to its autocorrelation values. Large autocorrelation values mean that neighbors have similar values and therefore the distribution of the values is continuous and smooth.

The neighborhood spatial autocorrelations were computed for each of the projection techniques. The results are shown in Figure 9. A large part of the display spaces for the PCA and

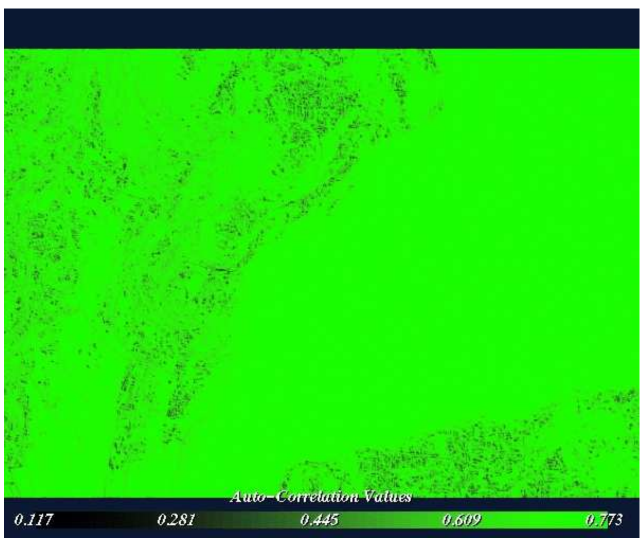
FS are empty and are shown to have high autocorrelation. However, the areas in display space where instances from data space have been projected have a low level of autocorrelation. The SOM image on the the other hand has high autocorrelation values throughout the display space. There are points of low autocorrelation but they are much fewer in number than those in the PCA and FS.



(a) Feature Selection



(b) Principal Components Analysis



(c) Self-Organizing Maps

Figure 9: Neighborhood spatial autocorrelation across the 2D display space

6 Discussion

Feature selection has the advantages of being simple to perform and intuitive to understand. Unfortunately, such a straightforward display frequently does not adequately capture the complex structure of the high-dimensional data space, since locations with very different characteristics along other dimensions are now aggregated. This can result in a projection which provides neither specificity, due to the collapsing of large data space hyper-regions to a single location, nor feature smoothness, due to unacceptably high variability of values among data space locations mapping to the same display space location.

For most data sets, the PCA projection has the advantage of showing more of the variability of the data space than does feature selection, even when the most important dimensions were selected. Unfortunately, the dimensions of the display space no longer have an intuitive meaning.

Dimension reduction by similarity clustering produces display space locations with greater region preservation and representation specificity than the maps produced by variable selection. Nearby locations in the data space map to nearby locations in the display space, and each data space location corresponds to a contiguous region of the data space.

Acknowledgments

This work was partially supported by NSF CAREER Grant 9996043 (Dr. Rheingans) and DARPA's HPKB program (Dr. desJardins).

References

[1] B. Becker. Research report: Visualizing decision table classifiers. In *Information Visualization '98*. IEEE Computer Society Press, 1998.

[2] W. Cleveland and M. McGill. *Dynamic Graphics for Statistics*. Wadsworth and Brooks/Cole, 1988.

[3] A. D. Cliff and J. K. Ord. *Spatial Autocorrelation*. Pion Limited, 1973.

[4] T. Cox and M. Cox. *Multidimensional Scaling*. Monographs on Statistics and Applied Probability. Chapman & Hall, 1994.

[5] N. Friedman and M. Goldszmidt. Building classifiers using Bayesian networks. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 1277–1284. AAAI Press, 1996.

[6] A. Inselberg and Bernard Dimsdale. Parallel coordinates: A tool for visualizing multi-dimensional geometry. In *IEEE Visualization '90*, pages 361–375. IEEE Computer Society Press, 1990.

[7] T. Kohonen. *Self-Organizing Maps*. Springer, 1995.

[8] T. Kohonen, J. Hynninen, J. Kangas, and J. Laaksonen. SOM_PAK: The self-organizing map program package. Technical Report A31, Helsinki University of Technology, Laboratory of Computer and Information Science, 1996.

[9] R. Pickett, H. Levkowitz, and S. Seltzer. Iconographic displays of multiparameter and multi-modality images. In *Proc. of Vis. in Biomedical Comp.*, pages 58–65. IEEE Computer Society Press, 1990.

[10] Penny Rheingans and Marie desJardins. Visualizing high-dimensional predictive model quality. In *IEEE Visualization '00*, 2000.

[11] H. Rushmeier, R. Lawrence, and G. Almasi. Case study: Visualizing customer segmentations produced by self organizing maps. In *IEEE Visualization '97*, pages 463–466. IEEE Computer Society Press, 1997.

[12] Silicon Graphics, Inc. MineSet user's guide. Technical Report 007-3214-002, SGI, 1999.

[13] University of California, Irvine. UCI machine learning repository. http://www.ics.uci.edu/˜mlearn/MLRepository, 1999.