Projection Methods and Visualization Techniques for High-Dimensional Model Understanding

Penny Rheingans
Marie desJardins
Department of Computer Science and Electrical Engineering
University of Maryland, Baltimore County
1000 Hilltop Circle
Baltimore, MD 21250
{rheingan—mariedj}@cs.umbc.edu

Projection Methods and Visualization Techniques for High-Dimensional Model Understanding

September 7, 2001

Abstract

Using inductive learning techniques to construct explanatory models for large, high-dimensional data sets is a useful way to discover useful information. However, these models can be difficult for users to understand. We have developed a set of visualization methods that enable a user to evaluate the quality of learned models, to compare alternative models, and identify ways in which a model might be improved.

We describe the visualization techniques we have explored, including methods for high-dimensional data space projection, variable/class correlation, instance mapping, and model sampling. We show the results of applying these techniques to a model built from a benchmark data set of census data.

1 Introduction

Discovering useful information in large, high-dimensional data spaces is a challenging problem. Using inductive learning techniques to construct explanatory models has proven to be a useful approach for solving this problem. However, these models can be difficult to interpret and evaluate. We have developed a set of visualization methods with the goal of enabling a user to evaluate the quality of learned models and to identify ways in which a model might be improved.

In this paper, we describe our exploration of a range of visualization techniques to support this goal, and show that these techniques yielded insights into the nature and quality of the models. The key problems that the work addresses are (1) projecting a high-dimensional data space into a 3D display space; (2) sampling from the data space to support this projection; and (3) visualizing model characteristics of interest in the display space.

2 Induced Models

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.

The visualization techniques we have developed are applicable to any learning methods whose output makes predictions that can be interpreted as probabilities, such as probabilistic decision trees or Bayesian networks. In the examples given in this paper, we used the ADULT data set from the UCI Machine Learning Repository [UCI 1999], which is derived from U.S. Census data, to construct classification models. We applied Tree-Augmented Naive Bayes (TAN) [Friedman and Goldszmidt 1996], a Bayesian network learning system that is tailored for classification, to construct the models. Data instances contain fourteen variables (six continuous and eight nominal) and a binary class label indicating income level ($>$ 50K (higher-income) or $<=$ 50K (lower-income)). Input variables include age, sex, race, education, occupation, hours worked per week, native country, type of employer, marital status, and household type. The data set has approximately 30,000 training instances and 15,000 test instances. Using a subset of eight of the input variables, we used TAN to construct (from the training data) a model to predict income level.

2.1 Model Characteristics

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.

We have identified a set of *model characteristics*—properties of a model that may be visualized in order to understand and analyze its behavior. 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:

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.

2.2 Model Analysis

A number of measurements have been developed by machine learning and statistics researchers to assess model performance. The most commonly used are classification accu-

racy, confusion matrices, and receiver operating characteristic (ROC) curves. By providing a visual representation of the model characteristics discussed previously, our visualization methods add depth to a user's understanding of these measurements.

In most machine learning research, model evaluation focuses on a single metric, such as classification accuracy. Classification accuracy is simply a single number that indicates the percentage of correctly classified instances in a test set. By showing the number of instances and their class labels against the background of the predicted probability distribution, the user can gain a visual understanding of the number and types of misclassifications.

A *confusion matrix* is often used to show the types of misclassifications made by a model. The confusion matrix is a two-dimensional table that indicates actual class label along one dimension and predicted class label along the other dimension. Each matrix entry has a number indicating how many instances with the corresponding actual class label were predicted by the model to have the corresponding predicted class label. Entries along the diagonal correspond to correctly classified instances. For a binary class, there are two off-diagonal entries, corresponding to false positives (negative instances with a predicted positive label) and false negatives (positive instances with a predicted negative label). These matrices are useful but often hard to understand, and do not tell the user which instances (i.e., which input attribute values) are being misclassified. The visual representation allows the user to see clusters of each type of misclassification. By querying the visualization, these clusters can be interpreted as regions in the data space.

ROC curves are used to assess the performance of the model with respect to varying misclassification costs. By changing the prediction threshold, any given model can be biased towards making more false positive predictions (lowering the threshold) or towards making more false negative predictions (raising the threshold). The ROC curves plots the false

positive rate against the false negative rate. By showing the probability distribution as a graded color background, and the instance classifications against this background, our approach gives the user a visual interpretation of the information conveyed by the ROC curve.

3 Data Space Projection

The data space for a model is in general a high-dimensional space, with one dimension for each variable. For instance, a model predicting the probability of a person earning a high income in the ADULT domain, which includes fourteen demographic and occupational characteristics, would inhabit a 14D data space. Each possible combination of variable values (i.e., each person of interest) corresponds to a point in this 14D 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 may first need to 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.

To support model analysis, the display space should represent the data space in such a way that the behavior of the model can be clearly visualized, and that the properties or regions of the data space that are important for model performance are preserved. We have identified five characteristics that an ideal projection method for model analysis should exhibit: region preservation, accuracy of representation, efficiency of representation, model

comparison, and model smoothness.

Region preservation means that homogeneous regions in the data space should correspond to contiguous regions in the display space. In other words, points that are close to each other *with respect to the model* 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 treated as similar by the model should be close to each other in the display space.

Any projection in which there are fewer locations than the size of the data space will have multiple data points mapping to locations in the display space. A projection is considered to be *accurate*, with respect to a model, if the set of points that map to a given location are "close" to each other (again with respect to the model) in the data space. This criterion is similar to region preservation: in the latter, we want nearby points in the data space to correspond to nearby points in the display space. An accurate representation has nearly the inverse property: collocated points in the display space should map to nearby points in the data space.

Efficiency of representation means that the display space should be utilized to represent regions of the data space that are *important* to the behavior of the model. For example, if a large part of the data space is treated as identical by the model, this region should map to a small region of the display space. (An extreme case of this would be a classification model for predicting ectopic pregnancies, in which maleness is an automatic disqualifier. The display space in this case should allocate little if any space to males.)

The ideal projection should support *model comparison*. There is a tradeoff here: a completely model-independent projection will facilitate model comparison, since the display space will not be biased towards either model. However, a model-independent projection is not likely to exhibit other desirable characteristics, such as region preservation. On the

other hand, using model-dependent projections can make it model comparison difficult: if different display spaces are used (each being dependent on the model being displayed), the correspondence between the visualizations can be unclear. If a common display space is used, and is built from one model or the other, the resulting comparison can be biased towards that model.

Finally, *model smoothness* is an essential characteristic of a projection. This simply means that the model characteristics to be visualized (probability distribution, decision boundary, misclassifications, and meta-attributes) should be smooth and easily visible in the display space.

We will discuss three types of methods for performing dimension reduction: feature selection, principal components analysis (PCA), and similarity clustering. 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. Doing dimension reduction to 3D using feature selection and PCA is completely analagous and is, in fact, what we actually use in practice. 3D similarity clustering is theoretically straight-forward, but we have not implemented it.

3.1 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 model 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 1 shows a 2D display space created using feature selection. The selected features, education and hours worked, are shown with colored contour lines. Each location in this display space represents a high dimensional subspace where education and hours values are fixed but other values can vary over their entire range. Feature selection has the advantages that it is simple to perform and intuitive to understand.

Unfortunately, such a straightforward display frequently does not adequately capture the complex structure of the model in the high-dimensional data space, since instances with very different characteristics along other dimensions are now aggregated. This can result in a projection which provides neither accuracy, due to the collapsing of large data space hyper-regions to a single location, nor model smoothness, due to unacceptably high variability of predictions among data space locations mapping to the same display space location.

3.2 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. 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.

Locations in the data space map to display space coordinates given by their first two (or three) principal components. Figure 2 shows the display space defined by projection by

principal components of a sampled model. Colored contour lines of education and hours worked show something of the correspondence between the data and display spaces. These contours generally show increasing education level moving toward the lower right corner and increasing hours worked moving toward the lower left. If it were possible to analytically determine the data variable value for a location in the display space, these lines of constant data value would be expected to be generally straight, though not axis aligned as they were in feature selection. Unfortunatly, such an analytic determination is not possible, so we have averaged values over the model samples mapped to a location. This approach is sensitive to the distribution of samples, as well as to large unoccupied regions of the space.

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 predictive features were selected. Unfortunately, the dimensions of the display space no longer have an intuitive meaning. Additionally, PCA projections tend not to be efficient, in that important regions of the data space tend to map to the middle of the display space, leaving the corners unoccupied.

3.3 Similarity Clustering

In both feature selection and principal components projection, 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 to violate projection accuracy.

In order to achieve a display space which better represents the high-dimensional structure of the data space, we also used a set of projection techniques based on self-organizing

maps (SOM) [Kohonen 1995]. 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 data space projection using a public-domain package that implements self organizing maps [Kohonen *et al.* 1996].

Dimension reduction by similarity clustering produces display space locations with greater region preservation and representation accuracy 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.

The dimensions of a display space created through similarity clustering have a highly non-linear relationship to the dimensions of the data space. Figure 3 shows how two data space dimensions have been warped by the projection process. The blue lines show contours of constant education, with more vivid blues corresponding to higher education levels. The red and pink lines show contours of constant number of weekly hours worked, with more vivid reds corresponding to more hours. In this display space, highly educated people tend to group to the right, while people who work many hours tend to group to the top. These curving lines correspond to hyperplanes in the high-dimensional data space.

3.4 Data Space Sampling Schemes

With variable selection, each location in the display space represents a subspace of the data space in which each of the unrepresented dimensions span their entire range. For instance, in the projection from eight dimensions to three, each location in the resulting display represents the 5D subspace in which the three represented dimensions are held to their values in the display space, while the five other dimensions are allowed to vary. Each location of the data space therefore corresponds to a high-dimensional volume sampling from the data space. In this manner, whole dimensions of the data space are projected together.

Standard PCA and Similarity clustering methods project individual data space locations (i.e., the instances in the map training set), rather than whole dimensions. In these cases, we use a set of instances to derive a projection which can then be used for any other location in the data space. Specifically, the data instances used to derive a projection are point samples from the data space. This gives rise to the issue of *which* locations in the data space should be projected—that is, how should samples be created. The selection of sample points can greatly affect the resulting display space. Once a projection has been derived using a set of sample points other points can be projected into that display space or model predictions can be computed for locations in the display space.

We have explored three basic approaches for choosing these sample points: using an instance set as the sample set, taking a sample from the model to be visualized, and using spanning vectors of the data space as the sample.

3.4.1 Training Data

In most information visualization applications, data sets (rather than models constructed from them) are displayed. The data sets are made up of discrete observations. In such applications, each observation is generally projected and displayed. Similarly, we can construct a display space for the model directly from the test or training instance set used to learn the model, applying the model to compute probabilities for the instance set before projection. For instance, the set of training instances could be used as the map training instances to build a SOM. The resulting display space would primarily contain regions of the data space which were occupied by training instances. If the training set is considered to be a representative sampling of the underlying population, such a map might be the best use of available space. Using this method, a unique map is produced for each training set and model. The effects of sample set on the derivation of PCA projection parameters are more subtle, but still influence the region of the data space traversed by the display space plane.

3.4.2 Model-Generated Sample

Alternatively, the locations used to build the display space can be sampled directly from the probability distribution specified by a model. With this method, the instance set used to derive the projection parameters will approximate the distribution of instances in the training set used to build the model, but need not have the same number of instances. As with the last case, the display space is specific to that model, facilitating understanding of the model characteristics of a particular model. However, comparing two models is difficult, since there is no particular relationship between locations in the two display spaces.

3.4.3 Data Space Spanning Vectors

A display space that is created from an instance set is naturally biased toward variable value combinations which appear in the instance set. Therefore, regions of the data space that are not represented or are sparsely represented by training instances will be underrepresented or missing from the display space. Unfortunately, these are regions in which a model can be most prone to making mistakes, so their inclusion is often desirable. One approach which addresses this problem is to generate the display space from a set of value vectors that span the data space by sampling from the data space using a uniform distribution. Such a display space depends only on the dimensions of the data space, making it easy to compare different models inhabiting the same data space. It also allows evaluation of the model over the entire data space, rather than just the portion of the data space from which the model was learned. This approach facilitates the comparison of alternative models, since display space locations are only specific to the data space, not any particular model.

4 Visualization of Model Characteristics

After projecting the data space into a 2- or 3D display space, we can visualize a variety of model characteristics in the display space. In this section, we discuss the visualization methods we have used to display the model characteristics, and how the visualizations can be used to interpret the behavior of the model.

4.1 Probability Distribution

On each projection, we can visualize the probability distribution, i.e., the probability of high income predicted by the model for each location in the display space. Because each of these locations corresponds to multiple points in the data space, it is necessary to average the predicted probability over those points. The degree of green saturation corresponds

to probability (more green means higher probability). The white contour line shows the decision boundary (50 percent probability of high income).

Figure 5 shows the probability distribution for the 2D feature selection projection. The correlations of the selected input variables with the class are easy to see by comparing Figure 1 to Figure 5: individuals with more education (towards the top) and who work more hours (towards the right) tend to make more money, with the education correlation being somewhat stronger.

In the PCA probability distribution visualization (Figure 6), we see again that more educated individuals (towards the "southeast") and individuals who work more hours (towards the "northeast") are more likely to earn a high income. The decision boundary here is rather fragmented, indicating that the PCA projection does not provide a smooth representation of this particular model characteristic.

Figure 7 shows the probability distribution for the SOM projection. When building the SOM, probability is treated as just another variable for similarity clustering. Since probability is one of the variables used to cluster points, the SOM training process can be biased to produce maps with greater coherence of predicted probabilities. Note the smoothness of the decision boundary in Figure 7 as compared with those of Figures 5 and 6.

This smoothness is somewhat misleading, as it is partially an artifict of the SOM building process. As mentioned earlier, the SOM treats probability like any other attribute, constructing the map through an iterative process that converges to codebook vectors at each location that characterize a set of instances. Each attribute, including probability, has some "average" value in this codebook vector. However, using this value for the probability distribution is not quite correct, since the predictions made by the model for the input in-

stances that fall at a location may not correspond to the probability given in the codebook vector. Since multiple instances in the data space map to the same location, in order to show the model probabilities, we need to average over those instances. Figure 8 shows the probability distribution generated by sampling the data space at each point in the SOM 16 times, and averaging the resulting probabilities. The decision boundary is considerably less smooth than the SOM probabilities shown in Figure 7. The latter is a close enough approximation for the purposes of visualization, however.

4.2 Instance Mapping

Test instances may be plotted in the display space in order to compare model predictions to actual classifications. In Figure 9, each test instance is displayed on the SOM as a small sphere colored by true classification (yellow for high income; dark red for low income). This view allow us to identify false positives (red glyphs inside the decision boundary) and false negatives (yellow glyphs outside the decision boundary).

Because of the compression involved in the projection process, many instances can map to the same location in the display space. Using 5000 test instances, we found that it was not uncommon for hundreds of instances to map to the same SOM cell. This is not a major issue when all of the instances that map to the same location have the same classification, but when the collocated instances include multiple class labels, important information is lost in the repesentation of Figure 9.

To convey this missing information, we developed a representation that shows the density and class label distribution of the instances (Figure 10). Glyph size indicates the number of instances at a given point. A continuous color map is used to show the proportion of class labels in the set of collocated instances. Yellow-saturated glyphs indicate a higher proportion of positive instances; red-saturated glyphs indicate more negative in-

stances. Orange glyphs indicate points where there are roughly equal numbers of positive and negative instances. This representation gives a much more useful and informative view of the nature and distribution of the test instances with respect to the decision space defined by the model.

Visualizing the instances allows us to identify regions of misclassification. The visualization can be queried to generate a description of the instances that correspond to any given region. For example, near the upper right there is a region with two large orange glyphs (mixed positive and negative instances). This region corresponds to a group of males in private industry who work long hours (60 hours a week), have moderate education (typically some college), and work as professionals or managers. Querying further, we can get a description of the positive instances (true positives) and negative instances (false positives) in the region. Upon inspection, there are few differences between these two groups. A knowledge engineer could use this analysis process to identify groups of individuals who are not easily differentiated with respect to the class of interest. Such a conclusion might lead to further data gathering (to identify features that would differentiate the high- and low-income earners), or might simply indicate that the model was not reliable for that particular group.

An interesting artifact of the SOM building process can be observed in Figure 9, in the instance-free "trench" snaking from left to right across the map. This trench corresponds to the contour line between men and women in Figure 12, and is a result of the averaging process of the SOM: between the "mostly female" and "mostly male" regions, the codebook vectors have intermediate values for sex, which of course do not occur in the actual data. Therefore, there are no instances that fall in that intermediate region.

4.3 Attribute Contours

Figures 11 and 12 overlay attribute contours for education level, hours worked, and sex on top of the probability distribution to show how the input variables are correlated with the predicted class and with each other. Notice that the region of the space that is more likely to contain high-income earners is also more likely to have greater education levels. These pictures also show several sociologically interesting effects: first, the model predicts that there will be relatively few high-income-earning women (the red concentric semi-circles at the right of the high-income region in Figure 12). Also, many of the predicted low-income females (red) that appear just outside the decision boundary have the same education level as the high-income males just inside the decision boundary. These individuals could be interpreted as underpaid women.

4.4 Annotation Glyphs

There is a tradeoff between the feature selection approach and the other projection approaches (SOM and PCA): in feature selection, the dimensions correspond to data space dimensions, and are therefore easy to understand. However, any contribution of the other features to the shape of the data space are lost in the projection. In contrast, the SOM and PCA incorporate all of the features, but the meaning of the display space is much less intuitive. PCA can be thought of as an intermediate approach: while it uses all of the features, the fact that it uses a linear combination of feature values means that it may not be able to preserve arbitrary regions or clusters within the data space as effectively as the SOM.

Annotating the display space can help in understanding the projection, allowing more effective use of both the SOM and the PCA approaches. The attribute contours that we

introduced earlier provide one form of annotation. We have been exploring other annotation approaches as well. Figure 4 shows a display where the SOM is discretized into coarser-grained regions, and a glyph is placed at each map location. The glyph indicates average education level (color scale), sex (glyph shape), hours worked (colored percentage of glyph), and marital status (ring shape around the glyph) for individuals in that region of the map.

5 Related Work

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 [SGI 1999], 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 [Becker 1998].

A wide variety of techniques have been developed to perform dimension reduction of high-dimensional data. These include parallel coordinates [Inselberg and Dimsdale 1990], multiparameter icons [Pickett *et al.* 1990], and a host of interactive techniques developed by dynamic statistics researchers [Cleveland and McGill 1988]. 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 [Cox and Cox 1994] and relevance maps [Assa *et al.* 1997]. Other applications of SOM techniques to information visualization include the visualization of customer charac-

teristics [Rushmeier *et al.* 1997].

6 Conclusions

We have presented visualization techniques that support model quality assessment and model comparison. These techniques allow the user to see multiple characteristics of the model's behavior. We described our exploration of a range of techniques to support these goals, and presented results on a benchmark data set. The research described here represents a first step towards a suite of visualization methods for model evaluation, which will enable the discovery of more useful information from high-dimensional data sets.

Acknowledgments

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

References

[Assa *et al.* 1997] J. Assa, D. Cohen-Or, and T. Milo, "Displaying data in multidimensional relevance space with 2D visualization maps," IEEE Visualization '97, IEEE Computer Society Press, 1997, pp. 127-134.

[Becker 1998] B. Becker, "Research report: Visualizing decision table classifiers," Information Visualization '98, IEEE Computer Society Press, Los Alamitos CA, 1998.

[Cleveland and McGill 1988] W. Cleveland and M. McGill, *Dynamic Graphics for Statistic*, Wadsworth and Brooks/Cole, Belmont CA, 1988.

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

[Friedman and Goldszmidt 1996] N. Friedman and M. Goldszmidt, "Building classifiers using Bayesian networks," Proceedings of the Thirteenth National Conference on Artificial Intelligence, Portland, OR, pp. 1277-1284. AAAI Press, 1996.

[Inselberg and Dimsdale 1990] A. Inselberg and Bernard Dimsdale, "Parallel coordinates: A tool for visualizing multi-dimensional geometry," IEEE Visualization '90, IEEE Computer Society Press, Los Alamitos CA, 1990, pp. 361-375.

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

[Kohonen *et al.* 1996] 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, FIN-02150 Espoo, Finland, 1996.

[Mitchell 1980] T. Mitchell, "The need for biases in learning generalizations," Rutgers University Technical Report CBM-TR-117, May 1980.

[Pickett *et al.* 1990] R. Pickett, H Levkowitz, and S. Seltzer, "Iconographic displays of multiparamter and multimodality images," Proceedings of the First Conference on Visualization in Biomedical Computing, IEEE Computer Society Press, Los Alamitos CA, 1990, pp. 58-65.

[Provost and Fawcett 1997] F. Provost and T. Fawcett, "Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions," Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, Huntington Beach, CA, 1997.

[Rushmeier *et al.* 1997] H. Rushmeier, R. Lawrence, and G. Almasi, "Case Study: Visualizing Customer Segmentations Produced by Self Organizing Maps," IEEE Visualization '97, IEEE Computer Society Press, Los Alamitos CA, 1997, pp. 463-466.

[Schroeder *et al.* 1996] W. Schroeder, K. Martin, and W. Lorensen, "The design and implementation of an object-oriented toolkit for 3D graphics and visualization," IEEE Visualization '96, IEEE Computer Society Press, Los Alamitos CA, 1996, pp. 93-100.

[SGI 1999] Silicon Graphics, Inc., *MineSet User's Guide,* SGI Document 007-3214-002, 1999.

[UCI 1999] University of California, Irvine, *UCI Machine Learning Repository*, http://www.ics.uci.edu/ ~mlearn/MLRepository.html, 1999.

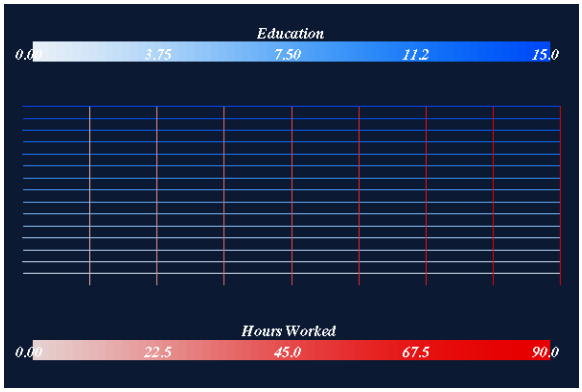

Figure 1. *Census data space projected into a 2D subspace with dimensions of hours worked (increasing to right) and age (increasing to top). Colored lines show contours of hours worked (red) and education level (blue).*

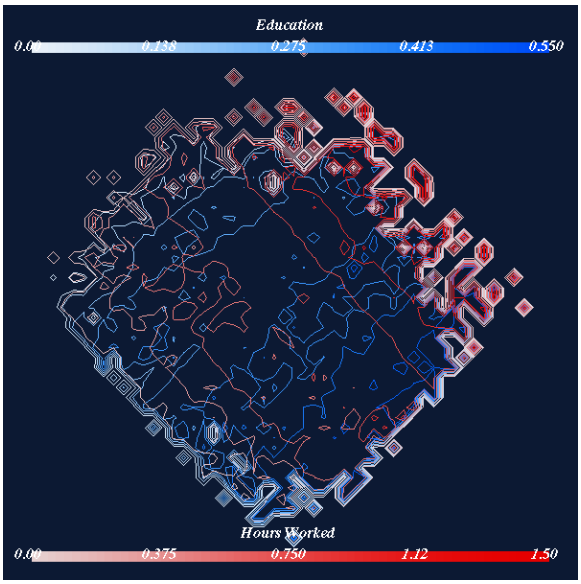

Figure 2. *Census data space projected into 2D using PCA. Contours show education (blue; roughly increasing towards lower right) and hours worked (red; roughly increasing towards upper right).*

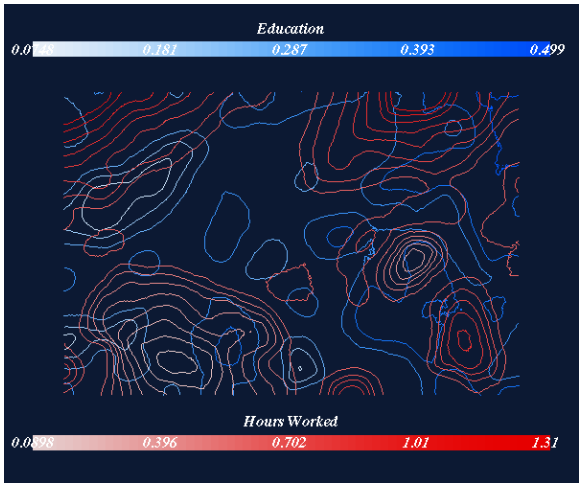

Figure 3. *Census data space projected into 2D using a SOM. Contours show education (blue) and hours worked (red).*

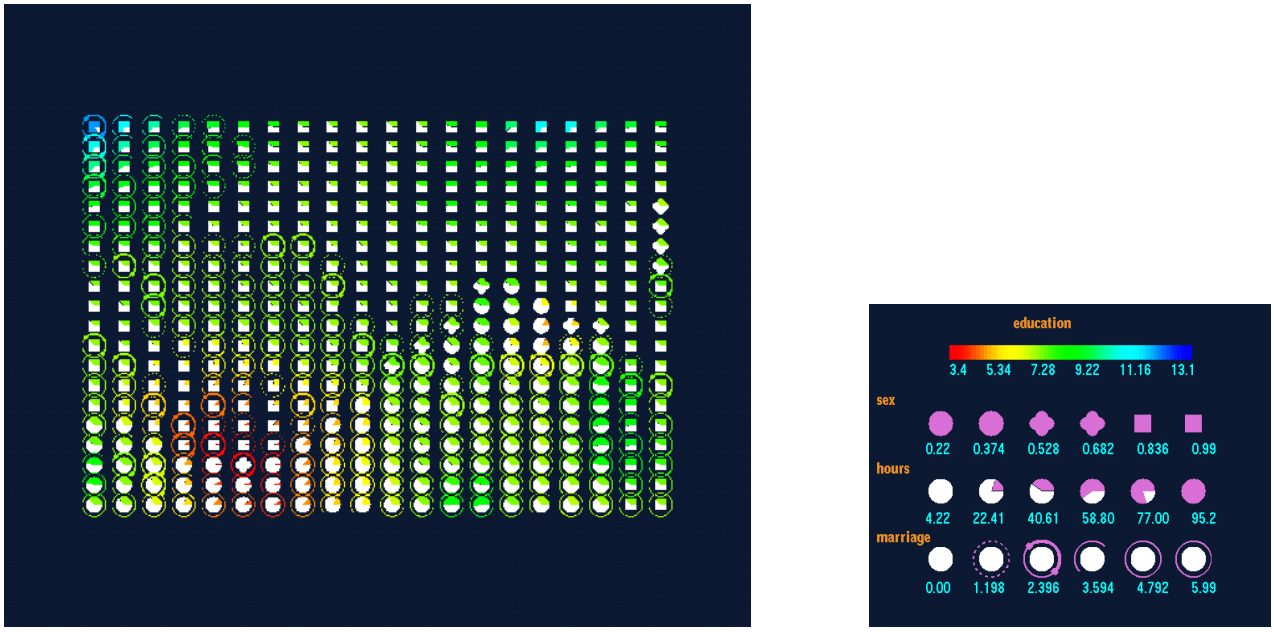

Figure 4. *2D SOM display space with annotated glyphs indicating attribute values for each map location. As shown in the legend, color scale indicates education level. Glyph shape indicates sex (circles are female, squares are male, clover shapes are mixed). Colored percentage of the glyph indicates hours worked. The ring around the glyph indicates marital status (no ring: single; dotted ring: absent spouse (separated or in armed services); ring with endpoints: divorced; half ring: widowed; circle: married).*

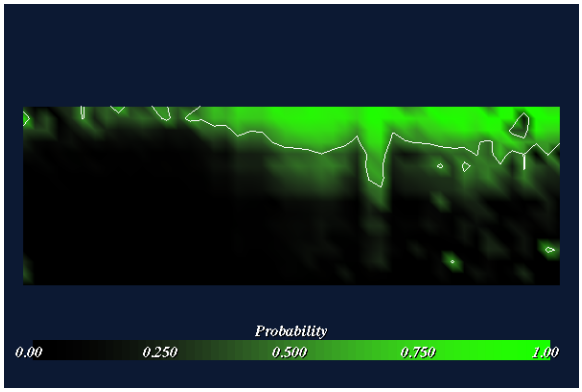

Figure 5. *Probability distribution for the 2D feature selection. Increasing saturation of green corresponds to increasing predicted probability of higher income. White line shows decision boundary.*

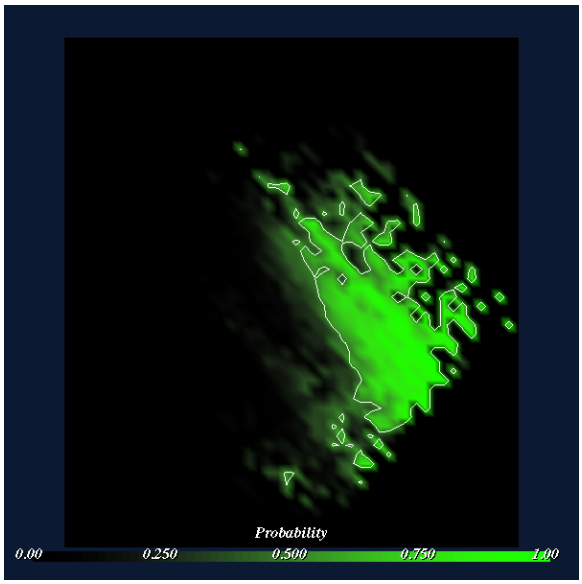

Figure 6. *Probability distribution for the PCA projection. Increasing saturation of green corresponds to increasing predicted probability of higher income. White line shows decision boundary.*

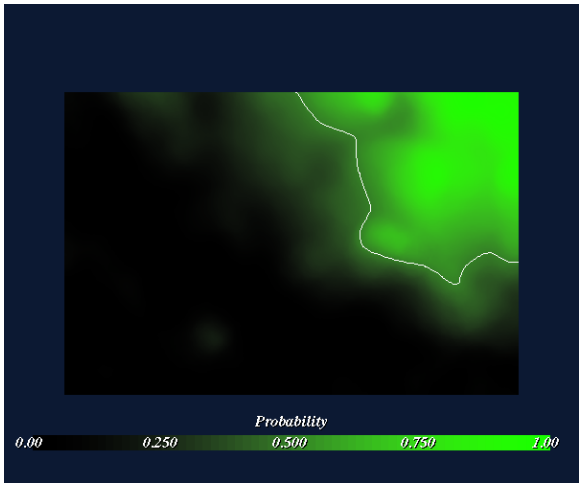

Figure 7. *Probability distribution for the SOM projection. Increasing saturation of green corresponds to increasing predicted probability of higher income (according to the codebook vector at each point). White line shows decision boundary.*

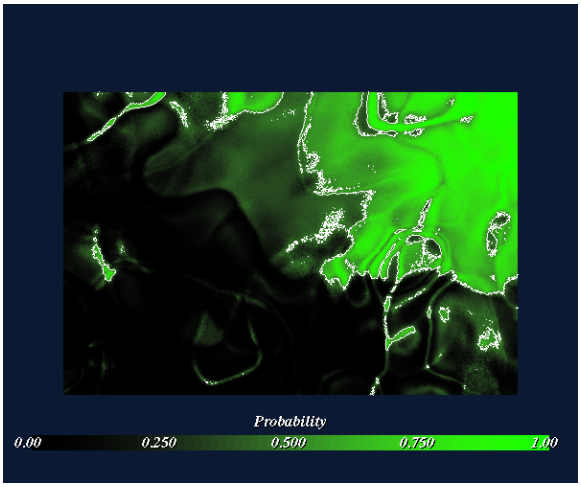

Figure 8. *SOM probability map with probabilities produced by querying the model.*

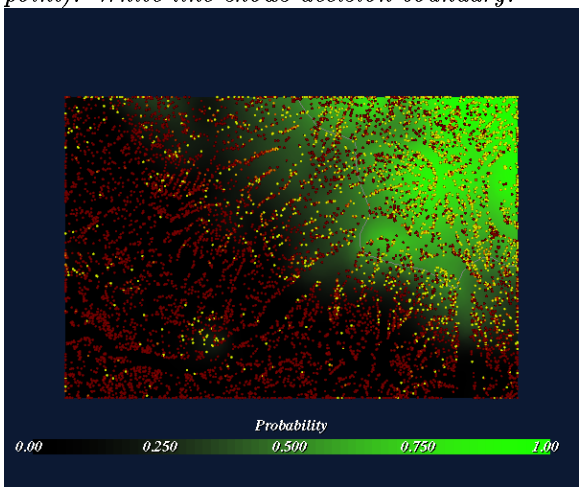

Figure 9. *SOM probability map with test instances overlaid. Yellow-saturated glyphs indicates instances that are higher-income earners; red-saturated glyphs correspond to lower-income earners. Orange-saturated glyphs correspond to points where both higher- and lower-income individuals are mapped.*

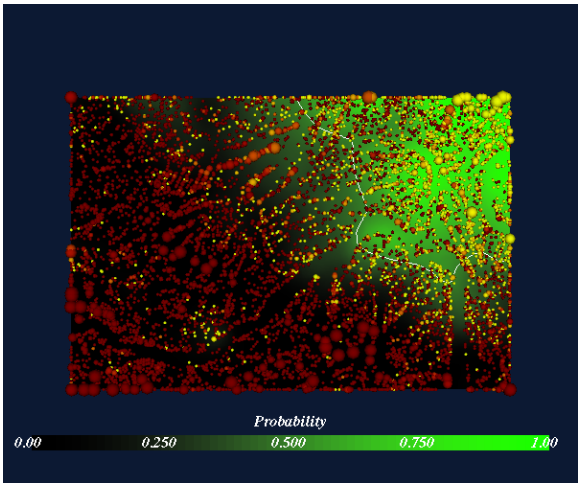

Figure 10. *SOM probability map with scaled test instances overlaid. Size of glyph at a point indicates number of instances that map to that point.*

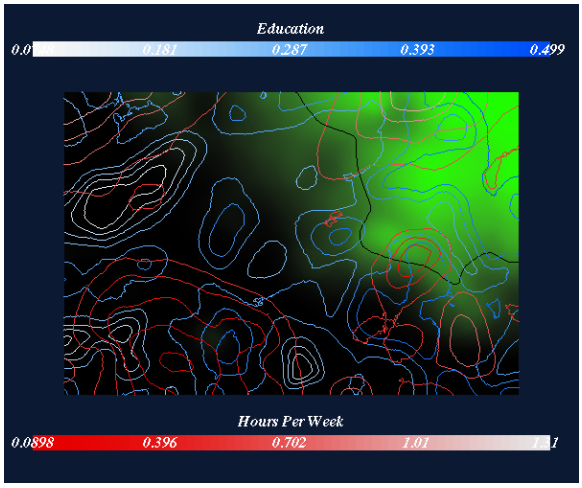

Figure 11. *SOM probability map with contours for education (blue) and hours worked (red) overlaid.*

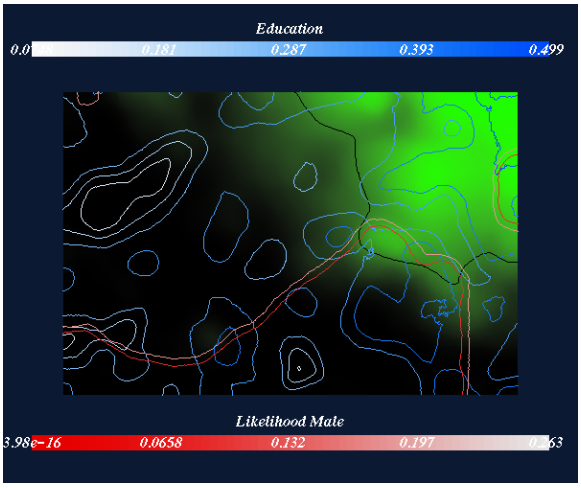

Figure 12. *SOM probability map with contours for education (blue) and sex (red) overlaid.*