Visualization of High-Dimensional Model Characteristics

Marie desJardins*
SRI International

Penny Rheingans†
University of Maryland, Baltimore County

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 several models built from a benchmark data set of census data.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; I.3.8 [**Computer Graphics**]: Applications

General Terms

Data Mining and Knowledge Discovery, Visualization, Multidimensional Information Spaces

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 perform three tasks: to evaluate the quality of learned models, to compare alternative 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 these tasks, 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 and (2) visualizing model characteristics of interest in the display space.

*SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025; e-mail: marie@ai.sri.com

†Department of Computer Science and Electrical Engineering, University of Maryland, 1000 Hilltop Circle, Baltimore, MD 21250; e-mail: rheingan@cs.umbc.edu

2. CONSTRUCTING MODELS

An explanatory 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. *Inductive learning* techniques take a set of training instances (object descriptions labeled with their class membership) and build a model that describes a mapping from a new object description to a class (or a probability distribution over classes). In the work we describe here, we are constructing and displaying Bayesian networks as the classification models, but the visualization methods generalize easily to other probabilistic models.

A Bayesian network (BN) is an efficient representation for capturing probabilistic relationships among variables in a domain, providing a succinct description of a joint probability distribution over these variables. A BN consists of a set of variables (nodes), a set of directed links between the nodes, and conditional probability tables (CPTs) at each node.

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 identified a set of *model characteristics* that may be visualized in order to understand and analyze a model's behavior. Some of the characteristics we have explored are *class probability* at each point in the data space; the decision boundary that delineates the region of instances that are predicted to be members of a class; *misclassifications*, which correspond to instances whose actual class label does not match the predicted class label; *misclassification types*, e.g., false positives and false negatives for binary classes; and *meta-attributes* of the model, such as the distribution and density of the training data used to build the model, and the confidence assigned to each estimate.

In the examples given in this paper, we used the ADULT data set from the UCI Machine Learning Repository [9], which is derived from U.S. Census data, to construct classification models. We applied Tree-Augmented Naive Bayes (TAN) [4], a BN 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 (gt50K (higher-income) or lt50K (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 the input variables, we used TAN to construct (from the training data) a model to predict income level.

3. VISUALIZATION TECHNIQUES

Information visualization applications are typically characterized by non-spatial, high-dimensional, non-continous data spaces. Visualizing such data generally requires a combination of some transformation of the data to a spatial display space with the application of representation techniques to the transformed data points.

3.1 Data Space Projection

A computer monitor or printed page can only display two dimensions. These two dimensions can be intuitively extended to three with the addition of computer graphics 3D shape cues such as obscuration, perspective, and view point control. While we can represent additional high-dimensional coordinate information using other display parameters, such as time, color, size, or opacity, these more abstract display parameters are not as easily or directly interpreted as spatial position. In practice, the display space is only three-dimensional; therefore, high-dimensional data must be projected down to three dimensions to be visually represented.

One obvious, and common, method for the projection of high-dimensional data into a lower-dimensional display space is the selection of a subset of available dimensions. Data observations are located in the display space according to their values in only the selected dimensions. Each location in this projected space corresponds to a subspace where all other dimensions are allowed to vary over their entire range. Unfortunately, such a straightforward display 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.

We have explored a projection technique based on self-organizing maps (SOM) [5]. In a SOM, neighboring locations in the display space correspond to neighboring locations in the data space, unlike feature selection where points far apart in the data space can map to identical locations 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 which would be located at neighboring locations in the data space. We are currently performing data space projection using a public-domain package that implements self-organizing maps [6].

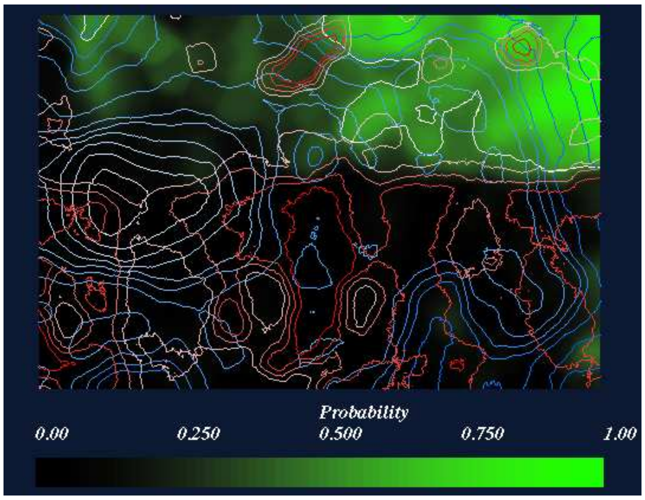

Figure 1: Projection of a model using self-organizing maps. Green saturation indicates probability of class membership; black contour line shows decision boundary. Red and blue contour lines correspond to constant values of gender and education level.

Figure 1 shows a representation of the model where the data space has been projected to two dimensions using a SOM. At each point in the map, saturation of green indicates the likelihood of high-income predicted by the model. (In black and white, more green saturation corresponds to lighter background colors.) The black contour line shows the decision boundary (50 percent likelihoood of high income). In order to produce this map, a large sample of representative instances with predicted probability of high income were generated from the model and used as training instances for the SOM code. In this way, probability is treated as just another variable for similarity clustering.

3.2 Variable/Class Correlation

Color mapped sets of contour lines can be used to clarify the correspondence between dimensions of the original data space and dimensions of the projected display space. Contour lines of constant variable value can be drawn in the display space. These contours show how the Cartesian grid of the original space has been warped by the projection process. The colored contours in Figure 1 represent deformations of several variables from the data space. Blue contours show education level, with increasing saturation corresponding to more education. Red/pink contours show sex, with increasing saturation corresponding to greater likelihood of

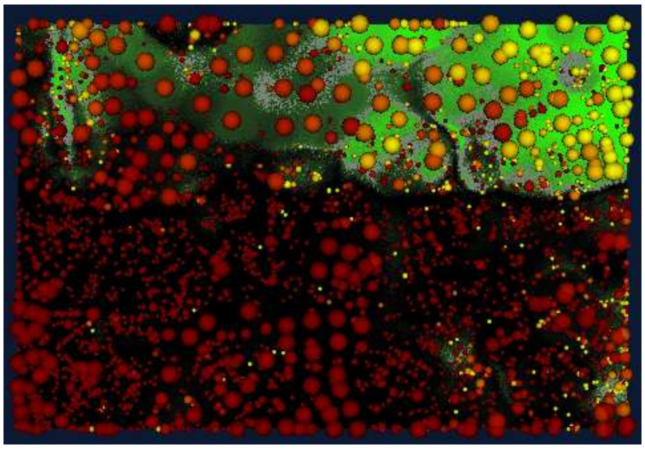

Figure 2: Test instances in projected space. Glyph size indicates number of instances at a location; yellow corresponds to positive instances; red corresponds to negative instances.

being female. (In black and white, these contours appear as white/gray lines.)

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. This picture also shows several sociologically interesting effects: first, the model predicts that there will be relatively few high-income-earning women (the red concentric circles in the center of the high-income region). 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.

3.3 Instance Mapping

Test instances may be plotted in the display space in order to compare model predictions to actual classifications. When a large number of test instances are plotted on the display space map, however, many instances can map to the same location in the SOM. Using 5000 test instances, we found that it was not uncommon for hundreds of instances to map to the same SOM cell. We developed a representation that shows the density and class label distribution of the instances (Figure 2). 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 (light) glyphs indicate a higher proportion of positive instances; red-saturated (dark) glyphs indicate more negative instances. Orange (medium gray) glyphs indicate points where there are roughly equal numbers of positive and negative instances. This representation gives a useful and informative view of the nature and distribution of the test instances with respect to the decision space defined by the model.

4. 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 [8], which includes several techniques for visualizing models, such as scatterplotting of misclassified instances. The display space is generated by manual feature selection, so the behavior of the complete model can be difficult to perceive. Visualization of classifiers in the MineSet paradigm was described by Becker [2].

Several other techniques that produce clusters in 2D space based on the similarity of data instances have been used to perform the projection of high-dimensional data to 2D. These techniques include multi-dimensional scaling [3], and relevance maps [1]. Although these methods differ substantially, the characteristics of the projections produced vary less. Other applications of SOM techniques to information visualization include the visualization of customer characteristics [7].

5. 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.

6. REFERENCES

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

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

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

[4] 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.

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

[6] 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.

[7] 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.

[8] Silicon Graphics, Inc. MineSet User's Guide, 1999.

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