Volume 6, Issue 1, January 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Data Mining Models – Needs & Comparison

[1]Dr. Shishu Pal Singh*, [2]Manoj Kumar
[1]Assistant Professor and Head, Computer Science Department, V.S.P. Govt. PG College, Shamli, Uttar Pradesh, India
[2]Assistant Professor, Computer Science and Engineering, Department, I.A.M.R. Ghaziabad, U.P., India

***Abstract:** As data mining is about extracting hidden information from the database which could not be provided by a report. Many data mining models and visualization techniques are used for this purpose. This paper emphasizes on various models used for data mining and discusses the importance of the visualization techniques.*

***Keywords:** CART, CHAID, OLAP*

I. INTRODUCTION

The point of data visualization is to let the user understand what is going on. Since data mining usually involves extracting "hidden" information from a database, this understanding process can get somewhat complicated. In most standard database operations nearly everything the user sees is something that they knew existed in the database already. A report showing the breakdown of sales by product and region is straightforward for the user to understand because they intuitively know that this kind of information already exists in the database. If the company sells different products in different regions of the county, there is no problem translating a display of this information into a relevant understanding of the business process.

Data mining, on the other hand, extracts information from a database that the user did not already know about. Useful relationships between variables that are non-intuitive are the jewels that data mining hopes to locate. Since the user does not know beforehand what the data mining process has discovered, it is a much bigger leap to take the output of the system and translate it into an actionable solution to a business problem. Since there are usually many ways to graphically represent a model, the visualizations that are used should be chosen to maximize the value to the viewer. This requires that we understand the viewer's needs and design the visualization with that end-user in mind. If we assume that the viewer is an expert in the subject area but not data modeling, we must translate the model into a more natural representation for them. For this purpose we suggest the use of orienteering principles as a template for our visualizations.

1.1 Orienteering

Orienteering is typically accomplished by two chief approaches: maps and landmarks. Imagine yourself set down in an unknown city with instructions to find a given hotel. The usual method is to obtain a map showing the large-scale areas of the city. Once the "hotel district" is located we will then walk along looking for landmarks such as street names until we arrive at our location. If the landmarks do not match the map, we will re-consult the map and even replace one map with another. If the landmarks do not appear correct then usually one will backtrack, try a short side journey, or ask for further landmarks from people on the street. The degree to which we will follow the landmark chain or trust the map depends upon the match between the landmarks and the map. It will be reinforced by unexpected matches (happening along a unique landmark for which we were not looking), by finding the landmark by two different routes and by noting that variations are small. Additionally, our experience with cities and maps and the urgency of our journey will affect our confidence as well.

The combination of a global coordinate system (the map analogy) and the local coordinate system (the landmarks) must fit together and must instill confidence as the journey is traversed. The concept of a manifold is relevant in that the global coordinates must be realizable, as a combination of local coordinate systems is some sense. To grow trust in the user we should:

1. Show that nearby paths (small distances in the model) do not lead to widely different ends
2. Show, on demand, the effect that different perspectives (change of variables or inclusion probabilities) have on model structure
3. Make dynamic changes in coloring, shading, edge definition and viewpoint (dynamic dithering)
4. Sprinkle known relationships (landmarks) throughout the model landscape.
5. Allow interaction that provides more detail and answers queries on demand.

The advantages of this manifold approach include the ability to explore it in some optimal way (such as projection pursuit), the ability to reduce the models to a independent coordinate set, and the ability to measure model adequacy in a more natural manner.

1.2 Why a Data Mining Model is needed?

The driving forces behind the need of data mining models can be broken down into two key areas: Understanding and Trust. Understanding is undoubtedly the most fundamental motivation behind visualizing the model. Although the simplest way to deal with a data mining model is to leave the output in the form of a black box, the user will not necessarily gain an understanding of the underlying behavior in which they are interested. If they take the black box model and score a database, they can get a list of customers to target (send them a catalog, increase their credit limit, etc.). There's not much for the user to do other than sit back and watch the envelopes go out. This can be a very effective approach. Mailing costs can often be reduced by an order of magnitude without significantly reducing the response rate.

The more interesting way to use a data mining model is to get the user to actually understand what is going on so that they can take action directly. Visualizing a model should allow a user to discuss and explain the logic behind the model with colleagues, customers, and other users. Getting buy-in on the logic or rationale is part of building the users' trust in the results. For example, if the user is responsible for ordering a print advertising campaign, understanding customer demographics is critical. Decisions about where to put advertising dollars are a direct result of understanding data mining models of customer behavior. There's no automated way to do this. It's all in the marketing manager's head. Unless the output of the data mining system can be understood qualitatively, it won't be of any use. In addition, the model needs to be understood so that the actions that are taken as a result can be justified to others.

Understanding means more than just comprehension; it also involves context. If the user can understand what has been discovered in the context of their business issues, they will trust it and put it into use. There are two parts to this problem: 1) visualization of the data mining output in a meaningful way, and 2) allowing the user to interact with the visualization so that simple questions can be answered. Creative solutions to the first part have recently been incorporated into a number of commercial data mining products (such as MineSet [1]). Graphing lift, response, and (probably most importantly) financial indicators (e.g., profit, cost, ROI) give the user a sense of context that can quickly ground the results in reality. After that, simple representations of the data mining results allow the user to see the data mining results. Graphically displaying a decision tree (CART, CHAID, and C4.5) can significantly change that way in which the data mining software is used. Some algorithms can pose more problems than others (e.g., neural networks) can but novel solutions are starting to appear.

It is the second part that has yet to be addressed fully. Interaction is, for many users, the Holy Grail of visualization in data mining. Manipulation of the data and viewing the results dynamically allows the user to get a feel for the dynamics and test whether something really counter-intuitive is going on. The interactivity helps achieve this and the easier this is to do the better. Seeing a decision tree is nice, but what they really want to do is drag-and-drop the best segments onto a map of the United States in order to see if there are sales regions that are neglected. The number of "what if" questions that can be asked is endless: How do the most likely customers break down by gender? What is the average balance for the predicted defaulters? What are the characteristics of mail order responders? The interaction will continue until the user understands what is going on with their customers. Users also often desire drill through so that they can see the actual data behind a model (or some piece of the model), although it is probably more a matter of perceptions rather than actual usefulness. Finally, integrating with other decision support tools (e.g., OLAP) will let users view the data mining results in a manner that they are already using for the purpose of understanding customer behavior. By incorporating interaction into the process, a user will be able to connect the data mining results with his or her customers.

II. TRUSTING THE MODEL

Attributing the appropriate amount of trust to data mining models is essential to using them wisely. Good quantitative measures of "trust" must ultimately reflect the probability that the model's predictions would match future test targets. However, due to the exploratory and large-scale nature of most data-mining tasks, fully articulating all of the probabilistic factors to do so would seem to be generally intractable. Thus, instead of focusing on trying to boil "trust" down to one probabilistic quantity, it is typically most useful to visualize along many dimensions some of the key factors that contribute to trust (and distrust) in ones models. Furthermore, since, as with any scientific model, one ultimately can only disprove a model, visualizing the limitations of the model is of prime importance. Indeed, one might best view the overall goal of "visualizing trust" as that of understanding the limitations of the model, as opposed to understanding the model itself.

Since data mining relies heavily on training data, it is important to understand the limitations that given data sets put on future application of the resulting model. One class of standard visualization tools involves probability density estimation and clustering over the training data. Especially interesting would be regions of state space that are uncommon in the training data yet do not violate known domain constraints. One would tend to trust a model less if it acts more confident when presented with uncommon data as future inputs. For time-series data, visualizing indicators of non-stationarity is also important.

2.1 Assessing Trust in a Model

Assessing model trustworthiness is typically much more straight-forward than the holy grail of model understanding per se — essentially because the former is largely deconstructive while the latter is constructive. For example, without a deep understanding of a given model, one can still use general domain knowledge to detect that it violates expected qualitative principles. A well-known example is that one would be concerned if ones model employed a (presumably spurious) statistic correlation between shoe size and IQ. Of course, there are still very significant challenges in declaring such knowledge as completely and consistently as possible.

Domain knowledge is also critical for outlier detection needed to clean data and avoid classic problems such as a juvenile crime committed by a 80-year-old "child". If a data mining model were build using the data in Figure 1, it is possible that outliers (most likely caused by incorrect data entry) will skew the resulting model (especially the zero-year-old children, which are more reasonable than eighty-year-old children). The common role of visualization here is mostly in terms of annotating model structures with domain knowledge that they violate.

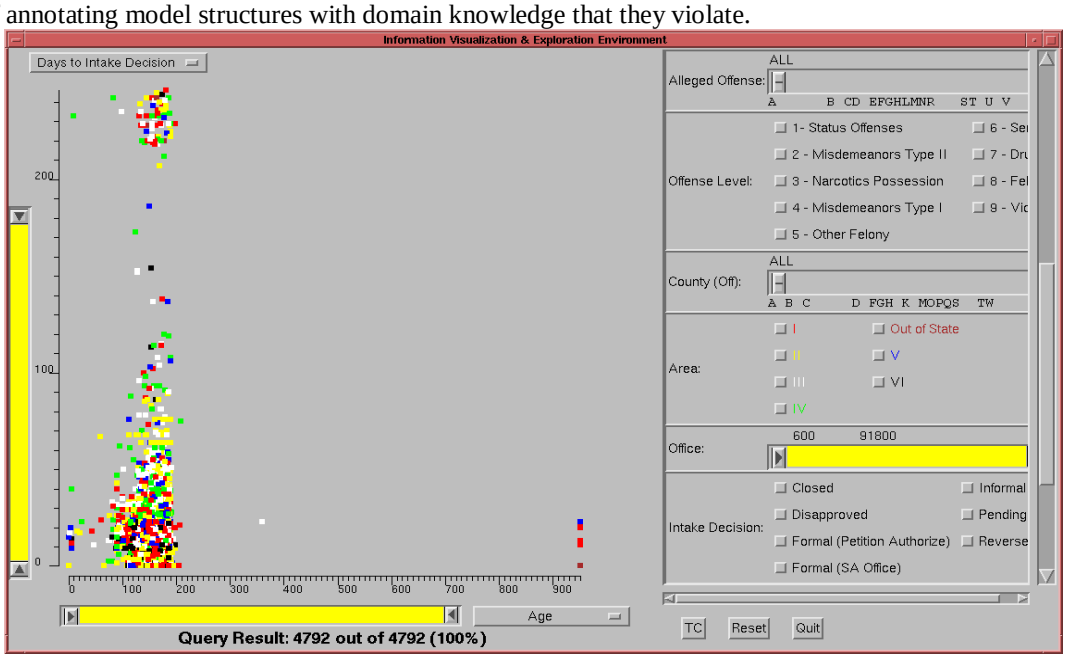


Figure 1: Age (in months) vs. Days to Intake Decision for juvenile crime offenders, Maryland Department of Juvenile Services. Note the 80-year-old children on the right side of the graph.

Not all assessments of trust are negative in nature, however. In particular, one can also increase ones trust in a model if other reasonable models seem worse. In this sense, assessing trust is also closely related to model comparison. In particular, it is very useful to understand the sensitivity of model predictions and quality to changes in parameters and/or structure of the given model. There are many ways to visualize such sensitivity, often in terms of local and global (conditional) probability densities — with special interest in determining whether multiple modes of high probability exist for some parameters and combinations. Such relative measures of trust can be considerably less demanding to formulate than attempts at more absolute measures, but do place special demands on the visualization engine, which must support quick and non-disorientating navigation through neighboring regions in model space.

Statistical summaries of all sorts are also common and useful for gathering insights for assessing model trust. Pairwise scatter-plots and low-dimensional density estimates are especially common. Summaries can be particularly useful for comparing relative trust of two models, by allowing analysis to focus on subsets of features for which their interrelationships differ most significantly between two models.

It is often useful to combine summaries with interactive ability to drill-through to the actual data. Many forms of visual summary actually display multiple scales of data along the raw to abstract continuum, making visual drill-through a natural recursive operation. For example, compressing millions of samples into a time-series strip chart that is only 1000 pixels wide allows one to quickly see the global highest and lowest points across the entire time range, as well as the local high and low points occurring within each horizontal pixel.

Most useful are models that qualify their own trustworthiness to some degree, such as in quantifying the expected variance in the error of their predictions.

In practice, such models tend to be relatively rare. Heavy emphasis on expected case rather than worst case performance is generally not all that inappropriate, since one is typically ultimately interested in concepts such as expected cumulative payoff.

There are important classes of tasks, such as novelty detection (e.g. fraud detection), for which quantified variance is essential. Standard techniques are learning confidence intervals (e.g. error bars for neural networks) and general probability density estimation. A promising recent approach [2], called bounds estimation, attempts to find a balance between the complexity of general probability density estimation and the simplicity of the mean estimation plus variance estimation approach to error bars.

Finally, it is important, though rather rare in practice to date, to consider many transformations of the data during visual exploration of model sensitivities. For example, a model that robustly predicts well the internal pressure of some engineering device should probably also be able to do well predicting related quantities, such as its derivative, its power spectrum, and other relevant quantities (such as nearby or redundant pressures). Checking for such internal consistency is perhaps ultimately one of the most important ways to judge the trustworthiness of a model, beyond standard cross validation error. Automated and interactive means of exploring and visualizing the space (and degrees) of inconsistencies a model entails seems to be a particularly important direction for future research on assessing model trustworthiness.

III. UNDERSTANDING THE MODEL

A model that can be understood is a model that can be trusted. While statistical methods build some trust in a model by assessing its accuracy, they cannot assess the model's semantic validity — its applicability to the real world.

A data mining algorithm that uses a human-understandable model can be checked easily by domain experts, providing much needed semantic validity to the model. Unfortunately, users are often forced to trade off accuracy of a model for understandability.

Advanced visualization techniques can greatly expand the range of models that can be understood by domain experts, thereby easing the accuracy/understandability trade-off. Three components are essential for understanding a model: representation, interaction, and integration. Representation refers to the visual form in which the model appears. A good representation displays the model in terms of visual components that are already familiar to the user. Interaction refers to the ability to see the model in action in real time, to let the user play with the model as if it were a machine. Integration refers to the ability to display relationships between the model and alternate views of the data on which it is based. Integration provides the user context.

The rest of this section will focus on understanding classification models. Specifically, we will examine three models built using Silicon Graphic's MineSet: decision tree, simple Bayesian, and decision table classifiers [3]. Each of these tools provides a unique form of understanding based on representation, interaction, and integration.

The graphical representation should be simple enough to be easily understood, but complete enough to reveal all the information present in the model. This is a difficult balance since simplicity usually trades off against completeness. Three-dimensional visualizations have the potential to show far more information than two-dimensional visualizations while retaining their simplicity. Navigation in such a scene lets one focus on an element of interest while keeping the rest of the structure in context. It is critical, however, that the user be able to navigate through a three-dimensional visualization in real time. An image of a three-dimensional scene is merely a two-dimensional projection and is usually more difficult to understand than a scene built in two dimensions.

Even with three dimensions, many models still contain far too much information to display simply. In these cases the visualization must simplify the representation as it is displayed. The MineSet decision tree and decision table visualizers use the principle of hierarchical simplification to present a large amount of information to the user.

Decision trees are easy to understand but can become overwhelmingly large when automatically induced. The SGI MineSet Tree Visualizer uses a detail-hiding approach to simplify the visualization. In figure 2, only the first few levels of the tree are initially displayed, despite the fact that the tree is extensive. The user can gain a basic understanding of the tree by following the branches of these levels. Additional levels of detail are revealed only when the user navigates to a deeper level, providing more information only as needed.

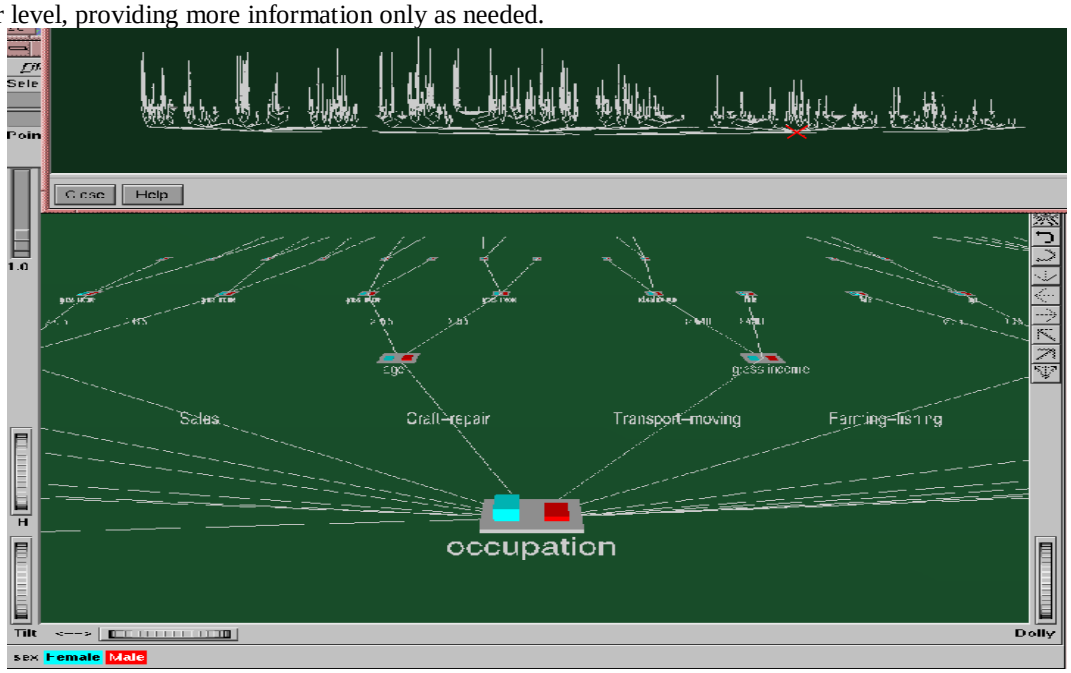


Figure 2: The MineSet Tree Visualizer shows only the portion of the model close to the viewer.

Using decision tables as a model representation generates a simple but large model. A full decision table theoretically contains the entire dataset, which may be very large. Therefore simplification is essential. The MineSet decision table arranges the model into levels based on the importance of each feature in the table. The data is automatically aggregated to provide a summary using only the most important features. When the user desires more information, he can drill down as many levels as needed to answer his question. The visualization automatically changes the aggregation of the data to display the desired level of detail. In figure 3, a decision table shows the well-known correlation between head shape and body shape in the monk dataset. It also shows that the classification is ambiguous in cases where head shape does not equal body shape. For these cases, the user can drill down to see that the attribute jacket color determines the class.

While a good representation can greatly aid the user's understanding, in many cases the model contains too much information to provide a representation that is both complete and understandable. In these cases we exploit the brain's ability to reason about cause and effect and let the user interact with the more complex model. Interaction can be thought of as "understanding by doing" as opposed to "understanding by seeing".

Common forms of interaction are interactive classification, interactive model building, drill-up, drill-down, animation, searching, filtering, and level-of-detail manipulation. The fundamental techniques of searching, filtering, drill-up, and drill-down, make the task of finding information hidden within a complex model easier. However, they do not help overall understanding much. More extensive techniques (interactive classification, interactive model building) are required to help the user understand a model which is too complicated to show with a static image or table. These advanced methods aid understanding by visually showing the answer to a user query while maintaining a simplified representation of the model for context.

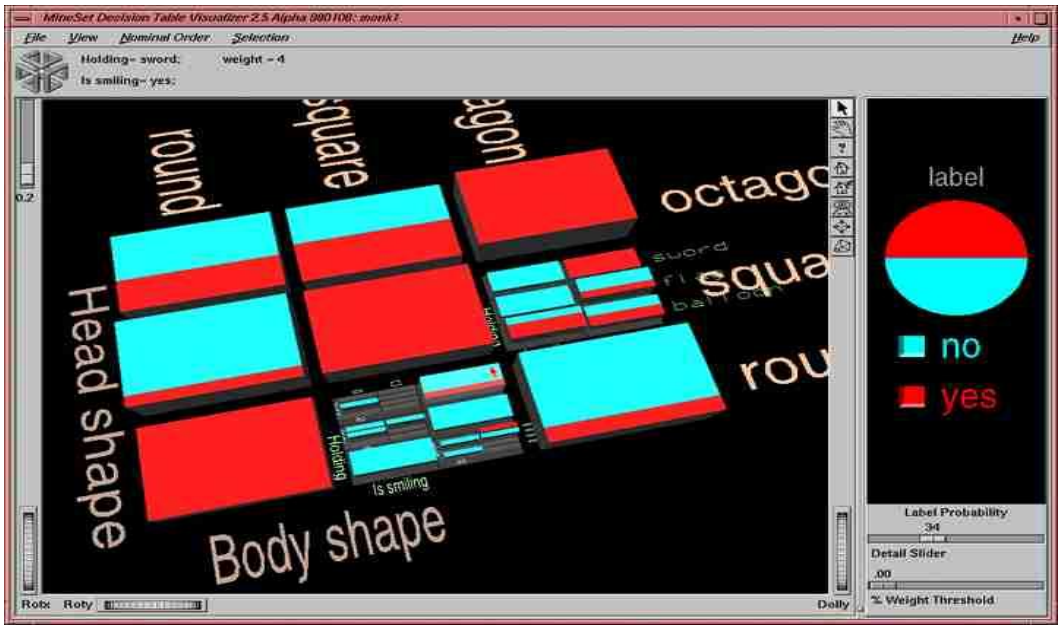


Figure 3: The MineSet Decision Table Visualizer shows additional pairs of attributes as the user drills down into the model.

The MineSet Evidence Visualizer allows the user to interact with a simple Bayesian classifier (Figure 4). Even simple Bayesian models are based on multiplying arrays of probabilities that are difficult to understand by themselves. However, by allowing the user to select values for features and see the effects, the visualization provides cause-and-effect insight into the operation of the classifier. The user can play with the model to understand exactly how much each feature affects the classification and ultimately decide to accept or reject the result. In the example in the figure, the user selects the value of "working class" to be "self-employed-incorporated," and the value of "education" to be "professional-school". The pie chart on the right displays the expected distribution of incomes for people with these characteristics.

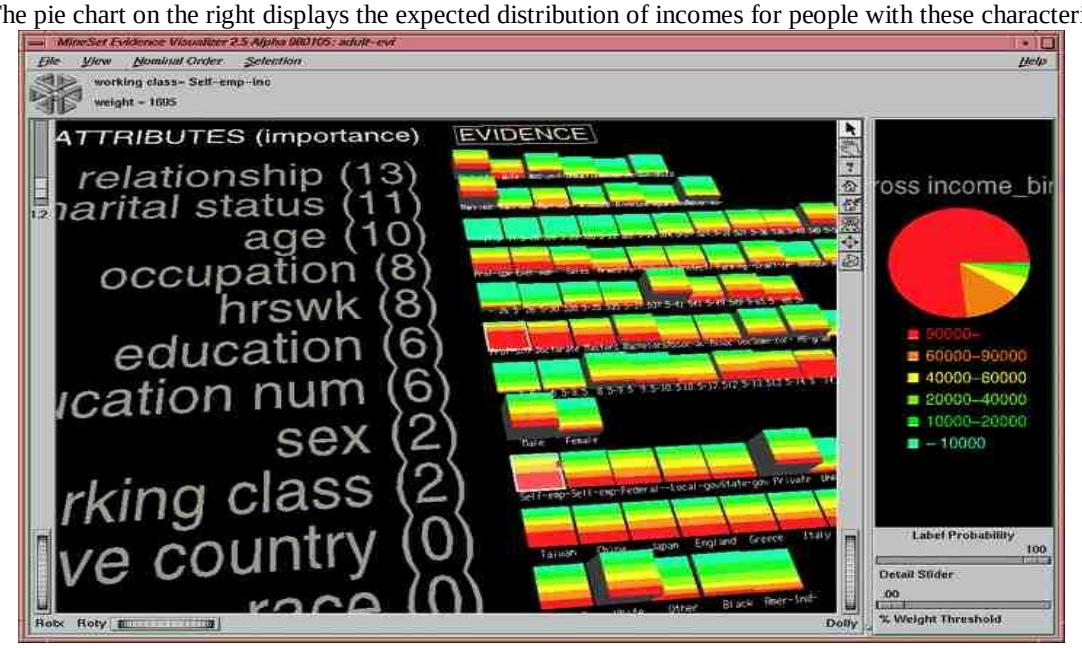
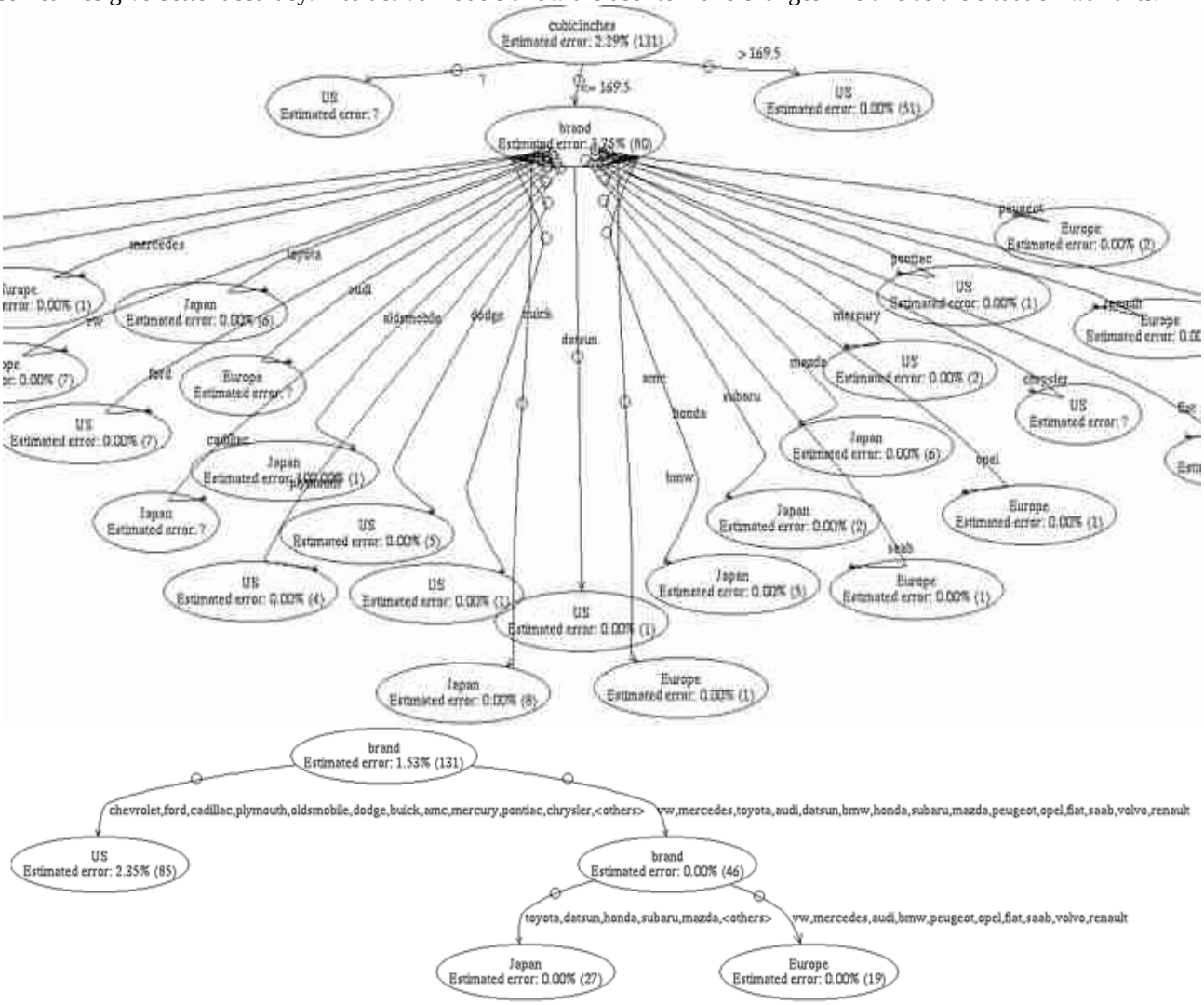


Figure 4: Specific attribute values are selected in the Evidence Visualizer in order to predict income for people with those characteristics.

Beyond interactive classification, interactively guiding the model-building process provides additional control and understanding to the user. Angoss [4] provides a decision tree tool that gives the user full control over when and how the tree is built. The user may suggest splits, perform pruning, or manually construct sections of the tree. This facility can boost understanding greatly. Figure 5a shows a decision tree split on a car's brand attribute. While the default behavior of the tree is to form a separate branch on the tree for each categorical value, a better approach is often to group similar values together and produces only a few branches. The result shown in figure 5b is easier to understand and can sometimes give better accuracy. Interactive models allow the user to make changes like this as the situation warrants.



Figures 5a and 5b: A decision tree having branches for every value of the brand attribute (top), and a decision tree which groups values of brand to produce a simpler structure (bottom).

Interactive techniques and simplified representations can produce models that can be understood within their own context. However, for a user to truly understand a model, he must understand how the model relates to the data from which it was derived. For this goal, tool integration is essential.

Few tools on the market today use integration techniques. The techniques that are used come in three forms: drill-through, brushing, and coordinated visualizations. Drill-through refers to the ability to select a piece of a model and gain access to the original data upon which that piece of the model was derived. For example, the decision tree visualizer allows selection and drill-through on individual branches of the tree. This will provide access to the original data that was used to construct those branches, leaving out the data represented by other parts of the tree. Brushing refers to the ability to select pieces of a model and have the selections appear in an alternate representation. Coordinated visualizations generalize both techniques by showing multiple representations of the same model, combined with representations of the original data. Interactive actions that affect the model also affect the other visualizations. All three of these techniques help the user understand how the model relates to the original data. This provides an external context for the model and helps establish semantic validity.

As data mining becomes more extensive in industry and as the number of automated techniques employed increases, there is a natural tendency for models to become increasingly complex. In order to prevent these models from becoming mysterious oracles, whose dictates must be accepted on faith, it is essential to develop more sophisticated visualization techniques to keep pace with the increasing model complexity. Otherwise there is a danger that we will make decisions without understanding the reasoning behind them.

IV. COMPARING DIFFERENT MODELS USING VISUALIZATION

Model comparison requires the creation of an appropriate metric for the space of models under consideration. To visualize the model comparison, these metrics must be interpretable by a human observer through his or her visual system. The first step is to create a mapping from input to output of the modeling process. The second step is to map this process to the human visual space.

4.1 Different Meanings of the Word "Model"

It is important to recognize that the word "model" can have several levels of meaning. Common usage often associates the word model with the data modeling process. For example, we might talk of applying a neural network model to a particular problem. In this case, the word model refers to the generic type of model known as a neural network. Another use of the word model is associated with the end result of the modeling process. In the neural network example, the model could be the specific set of weights, topology, and node types that produces an output given a set of inputs. In still another use, the word model refers to the input-output mapping associated with a "black-box." Such a mapping necessarily places emphasis on careful identification of the input and output spaces.

4.2 Comparing Models as Input-Output Mappings

The input-output approach to model comparison simply considers the mapping from a defined input space to a defined output space. For example, we might consider a specific 1-gigabyte database with twenty-five variables (columns). The input space is simply the Cartesian product of the database's twenty-five variables. Any actions inside the model, such as creation of new variables, are hidden in the "black-box" and are not interpreted. At the end of the modeling process, an output is generated. This output could be a number, a prioritized list or even a set of rules about the system. The crucial issue is that we can define the output space in some consistent manner to derive an input to output mapping.

It is the space generated by the mappings that is of primary importance to the model comparison. For most applications the mapping space will be well defined once the input and output spaces are well defined. For example, two classifiers could be described by a set of input/output pairs, such as (obs1, class a), (obs2, class b), etc. The comparison metric could then be defined on these pairs as a count of the number differing, or GINI indices, or classification cost, etc. The resulting set of pairs could be visualized by simple plotting of points on a two-dimensional graph. The two model could be indexed by coloring or symbol codes. Or one could focus on the difference between each model directly and plot this. This approach should prove adequate so long as we restrict attention to a well-defined input-output structure.

4.3 Comparing Models as Algorithms

In the view of a model as static algorithm, again there seems to be a reasonable way to approach the model comparison problem. For example, a neural network model and an adaptive nonlinear regression model might be compared. These models would be expressed as a series of algorithmic steps. Each model's algorithm could then be analyzed by standard methods for measurement of algorithmic performance such as complexity, the finite word length and the stability of the algorithm. The investigator could also include measures on the physical implementation of the algorithm such as computation time, or computation size. Using these metrics the visualization could take the form of bar charts across the metrics. Again, different models could be encoded by color or symbol, and a graph of only difference between the two models on each metric could be provided. Each comparison would be for a static snapshot but certainly dynamic behavior could be exploited through a series of snapshots, i.e. a motion picture.

4.4 Comparing Models as Processes

The view of the model as a process is the most ill defined and therefore most intractable of the three views, but this should not minimize its importance. Indeed its sheer complexity might make it the most important view for the application of visualization. It is precisely in this arena that we encounter the subject area expert for whom these systems should offer the most benefit (such as confidence and trust).

The modeling process includes everything in and around the modeling activity, such as the methods, the users, the database, the support resources, and constraints such as knowledge, time and analysis implementation. Clearly this scope is too large for us to consider. Let us narrow our scope by assuming that the model comparison is being applied for one user on one database over a short time period. This implies that user differences, database differences, and knowledge difference can be neglected. We are left with analysis methods and implementation issues. For most subject area experts the implementation and the analysis are not separable, and so we will make the additional assumption that this issue can be ignored as well. With these simplifying assumptions we are essentially defining model comparison to be the comparison of modeling method and implementation simultaneously.

Imagine two models that are available in some concrete implemented form. These could be different general methods such as neural networks versus tree-based classifiers, or they could be different levels of sophistication within a class of models such as CART versus CHAID tree-structures. Remember that we are now focusing only on the modeling process, and not its input/output or algorithmic structure. It seems that reasonable metrics can be defined in this situation. For example, the running time could be such a metric, or the interpretability of instructions, or the number of tuning parameters that must be chosen by the user at run-time. The key here is that these metrics must be tailored to the user who is the target of the application. Thus, whereas the input-output view focused on these the spaces, and the algorithmic view focused on the properties of the algorithm independently of the user, now we must focus in great detail on the user's needs and perceptions.

Once a set of metrics are chosen, we appear to be in a similar situation to that described under the algorithmic comparison. We should be able to show the distances between models in each of the defined metrics in a bar chart or other standard display. Color or symbol coding can be used to show the results from each model on the same chart as well.

There will be many possible metrics for the model-building process, at least one per user. Since it is unlikely we can choose a set of "one-size-fits-all" metrics, it is more useful to establish properties of good metrics and create methods to establish them in novel situations. The metrics chosen by a academic researcher would likely be very different from those chosen business user. Some properties that good metrics for the modeling process should be:

1. That they are expressed in direct risk/benefit to user.
2. That they evaluate their sensitivity to model input and assumptions.
3. That they can be audited (open to questioning at any point).
4. That they are dynamic.
5. That they can be summarized in the sense of an overall map.
6. That they allow reference to landmarks and markers.

Some aspects of the visualization process will take on added importance. One such aspect is the sequential behavior of the modeling process. For example, it is common to plot frequently the updated fit between the data and the model predictions as a neural network learns. A human being will probably give more trust to a method which mimics his or her own learning behavior (i.e., a learning curve which starts with a few isolated details, then grows quickly to broad generalizations and then makes only incremental gains after that in the typical "S" shape). Unstable behavior or large swings should count against the modeling process.

Another aspect of importance should a visual track of the sensitivity of the modeling process to small changes in the data and modeling process parameters. For example, one might make several random starts with different random weights in a neural network model. These should be plotted versus one another showing their convergence patterns, again perhaps against a theoretical S-shaped convergence.

The model must also be auditable, meaning that inquiries may be made at any reasonable place in the modeling process. For a neural network we should be able to interrupt it and examine individual weights at any step in the modeling process. Likewise for a tree-based model we should be able to see subtrees at will. Ideally there could be several scales in which this interruption could occur.

Since most humans operate on a system of local and global coordinates it will be important to be able to supplement the visualizations with markers and a general map structure. For example, even though the direct comparison is between two neural nets with different structures, it would be good to have the same distances plotted for another method with which the user is familiar (like discriminant analysis) even if that method is inadequate. If the same model could be used on a known input, the user could establish trust with the new results. It might also be useful to have simultaneously a detailed and a summarized model displayed. For example, the full tree-based classifier might have twenty-five branches, but the summarized tree might show the broad limbs only. And if the output is a rule it might be useful to drive (through logical manipulation) other results or statements of results as a test of reasonableness.

V. CONCLUSION

In this number of methods to identify the need of data mining models have been discussed. Because data mining models typically generate results that were previously unknown to the user, it is important that any model provide the user with sufficient levels of understanding and trust.

REFERENCES

[1] C. Brunk, J. Kelly, and R. Kohavi, "MineSet: An Integrated System for Data Access, Visual Data Mining, and Analytical Data Mining," Proceedings of the Third Conference on Knowledge Discovery and Data Mining (KDD-97), Newport Beach, CA, August 1997.

[2] D. DeCoste, "Mining multivariate time-series sensor data to discover behavior envelopes," Proceedings of the Third Conference on Knowledge Discovery and Data Mining (KDD-97), Newport Beach, CA, August 1997.

[3] D. Rathjens, MineSet Users Guide, Silicon Graphics, Inc., 1997.

[4] See http://www.angoss.com.

