Paper 69-26

A Practical Introduction to the Power of Enterprise Miner™

Michael S. Lajiness, Pharmacia & Upjohn, Kalamazoo, Michigan

ABSTRACT

Enterprise Miner™ (EM) is one of the most exciting products to be produced by SAS Institute in its entire history! EM facilitates the analysis of data in a straightforward and logical way. It can be used by statistical neophytes and experts alike. This paper will attempt to give a basic introduction to this product, describe many of it's features and provide some practical hints and suggestions on how to best utilize the software. Some examples will be given to illustrate the power of EM and clearly demonstrate why Enterprise Miner is simply the best product of its kind today!

INTRODUCTION

Anyone who has not been asleep for the last couple of years has probably heard about Enterprise Miner(EM) software. This software has been highly touted by SAS Institute in a wide variety of application areas. Telecommunications, stock exchanges, insurance companies, retailers, and pharmaceuticals are all examples of areas that EM has been used. Enterprise Miner has won several awards but many traditional SAS software users know little about it. The goal of this present work is to shed a bit of light on this very interesting product, to encourage those contemplating looking at the software, and to provide a quick introduction to some of the neat features that EM has to offer.

Enterprise Miner is a very powerful product that is simply not trivial to learn nor is it easy to get started. This Beginning Tutorial paper will provide recommendations as to what documentation should be viewed and how to get the biggest "bang for the buck"! Since version 8 has now been released, it will be the version used in this paper. Where appropriate, an example dataset will be used to illustrate the features of EM and how the various analytic tools operate. It should be noted that the purpose of this paper is not to discover hidden relationships in the data so the emphasis will be given to the techniques and methodology rather than the interpretation. It should also be noted that there are SI-based training courses specifically tailored to Data Mining. These include a general "Data Mining Techniques" course and a course in the use of decision tree (recursive partitioning) analysis as well as neural network modeling.

The philosophical approach to datamining that EM embodies is based on the SEMMA paradigm, "Sample, Explore, Modify, Model, and Assess". This concept when coupled with the many tools available through EM and through SAS software provide a powerful collection of tools to understand and exploit your most complicated datasets

RECOMMENDATIONS FOR NEW (EM)USERS

Since EM has so much bundled inside, I recommend the KISS strategy. Keep It Simple to Start out! What this means is that you should start out using only the most basic tools and settings and try to get an analysis that "works". By "Works" I mean develop a model (think diagram) that is predictive to a reasonable degree and then explore the more exotic settings and options to optimize and improve the predictions. It is always a good idea that when you are evaluating or learning a new analytical tool , start out with a dataset you know and understand (for the most part). This doesn't mean that you necessarily know all the ins and outs of the data but you have a non trivial level of understanding of the data so you can, in a sense, be able to validate the methodology. To do "data mining", in general, you should have a specific question in mind that you are trying to answer. Something like "what people are most likely to buy our product".

STARTING GLOSSARY

There are a variety of terms that one needs to be familiar with to understand datamining in general and EM in particular. Different fields have different terminologies for basically the same thing. EM was first developed, it appears to me, with a more financial/business environment in mind. Thus, much of the terminology used in EM is quite a bit different to those in the pharmaceutical industry. Anyway, to get started here are some terms you should know.

Project - main organizational level, think main file folder

Diagram - A sequence of steps as defined by a visual flowchart depicting input, analyses and output. Also called **the "process flow diagram"**.

Node - A discrete set of user-definable options to perform various tasks or functions. These are represented in EM by a variety of icons that can be linked(connected) to other nodes forming a diagram. Nodes can be thought of as tools.

Data Partition – An operation that creates subsets of data to train, validate, and test any resultant model.

Tree – refers to a type of analysis. Also called decision trees, recursive partitioning, and CART (classification and regression trees).

Neural Network – a black box approach to modeling that utilizes a variety of architectures to develop predictive functions.

Process Flow - refers to the process flow diagrams, the way that the various nodes (icons) are connected in the EM workspace defining an analyses.

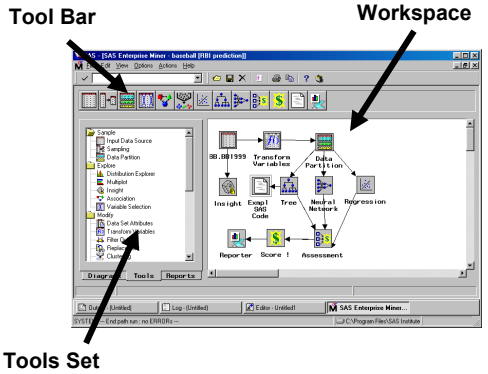
Tools – various operations/functions represented by various icons that one could select from the toolbar or the tool menu. Basically, it is the same thing as a node.

Toolbar – the EM toolbar sits on top of the EM Miner screen. This can be customized to include any tool you desire.

Workspace – The area in which you draw your diagram.

Target Variable - The dependent variable. The variable you are interested in predicting.

Inputs - The independent or descriptor variables. The variables that one will use to predict the value of the target variable.



OVERVIEW

In order to do data mining with EM one needs

- A reasonably large and representative set of data
- A question you are trying to answer
- Some knowledge of the application area
- Some knowledge of SAS software
- Enterprise Miner software!

Essentially the process for using EM to set up an analysis is simple. Anyone familiar with SAS (or any) programming knows the routine. You first read in the data, manipulate it in desired ways, analyze it, interpret the results, and generate reports/graphics that present the conclusions. The neat thing about Enterprise Miner is that it uses a visual programming interface. You can set up much of the analysis by simply including, moving and attaching icons in the workspace to define the order and manner in which the analyses will be performed! You really don't have to do any programming although it seems clear to me that to get the most out of the software a little bit of programming goes a long way. There are a lot of details that will not be delved into in this paper given the time and space limitations. We will instead focus on how one can operate EM without getting much into the details of an actual analysis. Specifically, in this tutorial we will illustrate how to:

- Specify the input dataset
- Partition the file into training, validation and test sets
- Generate models based on regression, decision trees, and neural nets
- Assess the performance of the various models
- Generate information to "score" new data.
- Add custom nodes to produce desired output
- Generate HTML reports to document the analysis

PREPARING TO RUN ENTERPRISE MINER™

Before running Enterprise Miner one needs to prepare a dataset that has all the variables needed in a format appropriate for analysis. One of the habits I have when I work on projects is to store the raw (non-SAS) data I want to analyze in specific directories, organized by project. I like to then write SAS code that reads in the raw data and creates the SAS dataset I want to analyze. I then store that dataset in a separate SAS_Datasets directory. One should be very careful about storing a dataset in any EM project folder. This is because if you ever decide to delete a particular project all files including data files contained in those directories will be deleted.

So, the first thing to do is to run one or more SAS programs to create a dataset that can be analyzed. This dataset should contain variables that may be appropriate to predicting the value of the target variable. The target variable is the variable you are interested in predicting. It can be continuous (think money!) or discrete (think "buy" or "sell").

The last thing to do before running EM is to focus on one particular problem you are trying to address. Let's say we are interested in finding out if one can adequately predict RBI (Runs batted in) given other batting statistics culled from the last 100 or so years of major league baseball[1].

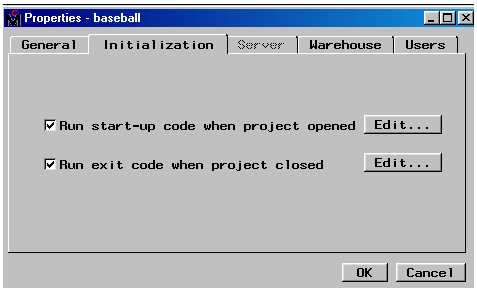
Using ENTERPRISE MINER™

In the next several sections we will go through the various steps to create, modify and run an EM project to analyze the baseball data. Due to time and space limitations not all corresponding screen shots or details will be mentioned but the main points will be hit so that the reader can understand the application.

To start EM one can enter MINER in the command line of SAS. Alternatively, one can run EM by clicking on SOLUTIONS>>ANALYSIS>>ENTERPRISE MINER. In general, one can run EM in basically two ways. You can build the diagram, one node at a time, running each node (think execute) as you build the diagram. One way to 'RUN" a node involves "right clicking" on the icon and selecting the RUN option. Or one can select and connect several nodes together and "running" the last node. Whenever RUN is selected all predecessor nodes that have not been run are executed.

Creating a New Project

EM stores all diagrams and datafiles under something called a "project". Thus, one needs to create a project by selecting File>New>Project in the EM screen. I created the project BASEBALL and chose to put it in the default location. Choosing the default location makes it easy to find the project later on. One can choose to change the properties for a project (Options>Project>Properties). One, for example, can select to create and run START UP that could contain any collection of SAS code that you want to have run prior to running EM. This is where one should put any LIBNAME statements to define dataset locations and any macro or macro variables that need to be available. For example, this is where one would put the libname statement (such as LIBNAME BB 'c:\sas_datasets';) to refer to the SAS dataset containing the baseball data. One note of warning here. If you ever find that you cannot change/edit the startup code this probably indicates the project you are in is in SHARED mode. You should refer to the Troubleshooting section in this paper if this for information on how to correct this.



Under project properties one can also define the location of the Data Warehouse or server profile. We will NOT be discussing setting up Client/Server projects or using Data Warehouse. Please refer to other EM documentation referred to later in this paper. EM stores the analysis users create in something called a diagram. Once EM is opened one needs to select the DIAGRAM tab if not already there. At this point you can simply click on the default name, UNTITLED, and type in the name you want to store the customized diagram as. A project can have many different diagrams. You can create additional ones by selecting FILE>NEW>DIAGRAM. Now it's time to start defining the desired analysis and to do this we access the optional nodes available under the TOOLS tab.

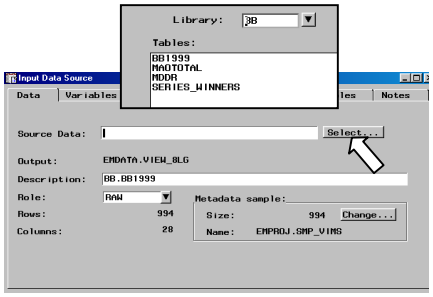
Building a Diagram

One of the neatest things about EM is how one can use visual programming icons to do almost all of the construction of the project. To starting building your diagram you just CLICK, DRAG, and MOVE nodes from the tool bar to the workspace. It's that simple!

Defining the Data to be analyzed

To define a dataset, simply click and drag the INPUT DATA SOURCE icon from the tools palette to the diagram workspace. Once you lift your finger off the mouse your icon is where you want it. Double click on the icon to open it and to define the

input dataset. It is possible to store EM input datasets virtually anywhere. However, it is probably a good idea NOT to store your datasets in the project folder. Again to reemphasize this point, if you delete a project you will also delete any datasets stored in that directory! So be careful! Once the dataset is in the proper directory one can click on SELECT and then one can click on the BASEBALL dataset as illustrated below.

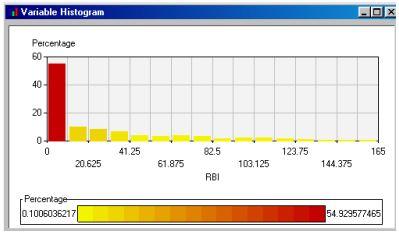


So, assuming we placed the LIBNAME BB statement in the START UP Code section we need to find the desired dataset to input. Once we have the INPUT DATA node opened we click on SELECT to find the right libname, BB. One we click on BB we find the BB1999 dataset which we select. This makes the data and variables available to browse and select. There are a few things one needs to do in this node in order to do an analysis. The most important thing to set up the target variable. Modeling requires the existence of at least one target or dependent variable. To select the appropriate target in the BASEBALL dataset one needs to click on the VARIABLES tab and scroll down to the variable RBI. Right clicking in the model role cell allows one to set the role to TARGET. The target variable can be continuous or categorical.

In this demonstration I chose to define the target variable to be RBI (runs batted in). Any variables you do not want included in the analysis needs to have their role changed to “REJECTED” so that they are not used in any modeling. Care needs to be taken when setting the event level for character/binary variables so that the order chosen reflects the proper sort sequence. One can also define a target profile for categorical target variables. Using this, one can define decision matrices and prior probabilities. This can dramatically improve prediction, especially when dealing with rare events. More information on target profiles can be found in the online guide and later on in this paper.

Exploring and Understanding your data

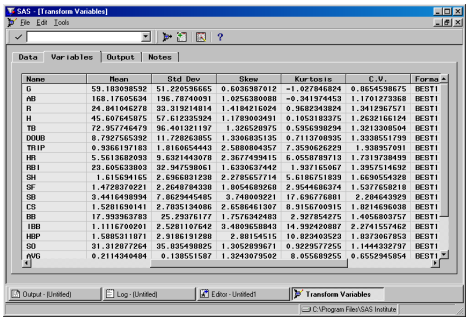
One of the most important things you need to do to understand your data and get the most out of it is to examine it. A quick and easy way to look at your data is to use the visualization option from within the INPUT DATA node. To do this find the variable you want to examine and then "right click" and select VIEW DISTRIBUTION. The distribution below illustrates this using the variable RBI.



One can also access visualization tools, such as Insight, which are very helpful in this regard. One can even set up a customized "SAS CODE" node to use a non-SAS tool called SPOTFIRE[2] to perform the visualization but it is beyond the scope of this tutorial to discuss this software. Outliers and skewed distributions can have a big effect on your analysis so it is a good idea to look before you leap!

Modifying attributes

To add a Transform data node one selects the appropriate tool and drags it to the workspace. One then can connect the INPUT DATA node to the TRANSFORM VARIABLES (TV) node by simply clicking on the edge of the INPUT node and dragging the “connection” to the TV node. This node enables you to view various statistics on each variable. These include the standard deviation, skewness, kurtosis, and CV (coefficient of variation).



You can also create new variables that are transformations of existing variables in your data. Transformations are useful when you want to improve the fit of a model or models to the data. For example, transformations can be used to stabilize variances, remove non-linearity, improve additivity, and correct non-normality in variables. Some of most useful functions available in this node is the MAXIMIZE NORMALITY transform, and the MAXIMIZE CORRELATION WITH TARGET option.

Quite often, in my experience, I find that I need to add new variables or create linear combinations of existing variables. Since I want these new variables in the original dataset I actually go back to the program used to create the input dataset in the first place and add new code to make the variables I want. Caution here! If you go back and change the dataset that is used in the INPUT DATA node the new variables won't be included automatically! One way to get the INPUT DATA node to recognize the new Variable is to open the node, do a right click select EDIT, and then pick REFRESH SAMPLE. The new variable should now be found under the VARIABLES tab.

Defining a Target Profile

I think of target profiles as basically a decision matrix defined on the values of the target variables that assign different weights to get better fits. These profiles are used by the various models you've included in your diagram. My recommendation is to start out NOT using target profiles and to stick with the default settings and optimization methods. Then, once you have a diagram that is working you might want to consider modifying the target profile to try and produce more optimal decisions

You can edit/change a TARGET PROFILE in a number of nodes. One place to do it is under the INPUT DATA node. Under the VARIABLES tab select the target variable and right click. Choose EDIT TARGET VARIABLE. We will not go into the intricacies of defining appropriate decision matrices here as it is beyond the scope of a beginning tutorial. Please refer to the online help files for in depth information or pp75 in the "Getting Started...v4.0" manual discussed later in this paper.

Setting the Event Level

If one is dealing with a binary target variable, you need to make sure that EM recognizes what event to model. Thus, one needs

to properly define the event level for the target variable. The convention is that the "best" value of the categorical target variable must sort first. Thus, one needs to make sure that ORDER is DESCENDING if the "best" value of the binary target variable would normally sort last. If, on the other hand, the best score sorts low, the order must be set to ASCENDING.

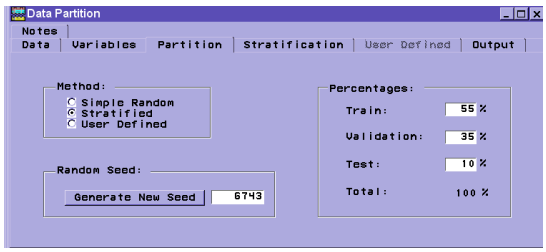
Sampling

Quite often one is interested in mining large datasets. Since some of the modeling steps can be computationally demanding it is normal practice to use a portion of the data to develop initial models before examining the full data set. The **Sampling** node serves this purpose. Since the baseball dataset we are using here is small we are not using a sampling node.

Partitioning Data to get a valid prediction

In developing models it is usual practice to split the dataset into 3 parts; a set to develop or train the model; a set to validate the model; and a set to independently test the model. The validation step is necessary to prevent overfitting. To include a partitioning step one can simply click and drag and connect the appropriate tool to the desired location on the workspace. This is exemplified below.

In the BASEBALL dataset, we have chosen to partition the dataset as indicated in the figure below. Note that we have reserved only 10% of the data in the TEST set to get an independent test of the final model and to get an idea about the general predictiveness of the resultant model(s). Note: The **Data Partition** node must be preceded by a node that exports at least one data table, this is usually an **Input Data Source** node or a **Sampling** node. Please also note that If one is dealing with a discrete target variable that is a rare event one might want to use stratified sampling to as to ensure there is adequate representation of each level of the target variable in all the sets.



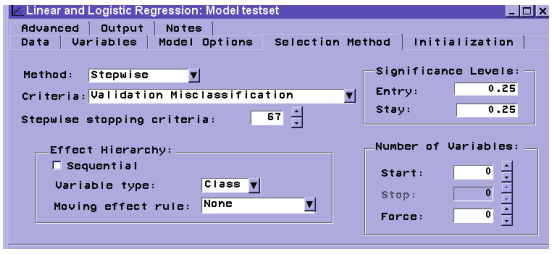
Modeling in General

Enterprise Miner provides several analytical tools for developing predictive models. In addition, you can also create your own modeling nodes using IML, for example. You may also access other statistical tools that are not built into EM, like discriminant analysis, via the SAS CODE node.

The **Regression** node, the **Tree** node, and the **Neural Network** node are all built into EM and can be effective tools for deriving useful models. Since each model is different in the way it goes about "learning" one can expect each model to be effective in different situations. It is important to start out with simple models and then making them as complex as necessary to obtain useful results. If you start out trying to develop models that are too complex you are in for some frustration! Remember, K.I.S.S.!

Building & Assessing a Regression Model

The regression node supports logistic regression and linear regression. Logistic regression works well with models that are based on ordinal or binary targets. Linear regression fits linear equations to available data when the target is an interval variable. The type of regression that will be performed will typically be automatically selected depending on the type of target involved. For either type of regression you can select from a variety of variable selection methods, FULL (no selection), FORWARD, BACKWARD, STEPWISE.

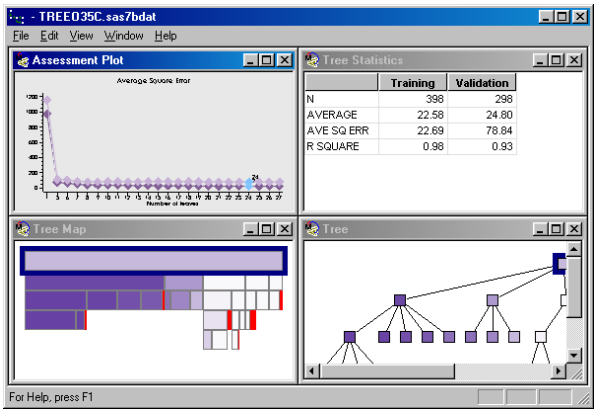


It should be noted that the more variables that are included in any model the more likely you are (in general) of overfitting your data. This will lead to poor model performance in the TEST dataset. After one links in a modeling node one also should connect an ASSESSMENT node. The ASSESSMENT node takes output from any modeling node and checks the model's accuracy against data in the Test partition.

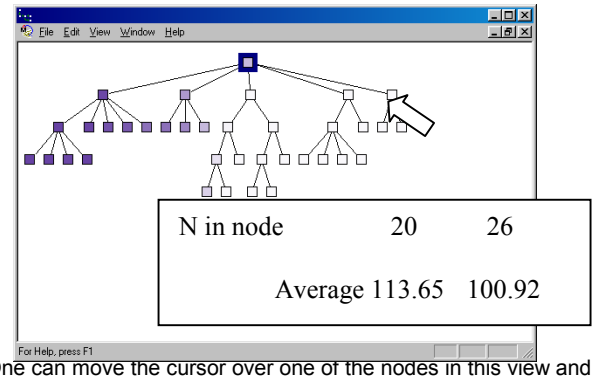
Building & Assessing a Decision Tree Model

Regression analysis can be visualized as a line or a curve that is fitted to your data. Decision trees take a different approach. They separate data into "branches" according to the values of variables in the model based on a fit of the target variable. This process continues until no more splitting can be done according to the optimization rules in force. Decision trees have a variety of advantages. They produce models that are more easily interpreted and missing values have no effect. As in the case of the regression node one also needs to assess the performance of the decision tree node.

There is a new setting for the TREE node in v4 that one can access by issuing the command: **%Let emv4tree=1**; One can do this while in an EM session by opening the editor window, inserting the command with a RUN statement and then submitting the code. This will allow the generation of tree results as shown in the following display.

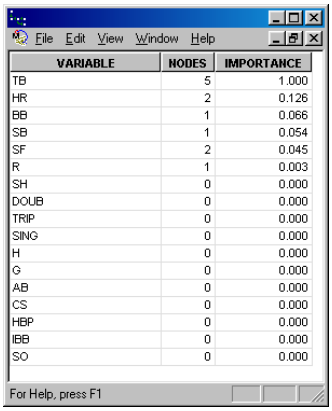


This is a pretty neat new feature. The tree node is more compact and understandable and can be expanded to fill the entire screen as shown below.



One can move the cursor over one of the nodes in this view and see a summarization of the "fit" at that node as shown above. The box that is displayed indicates that at this particular node two other nodes can be "split". One has 20 obs. and the other 26, with average RBI's of 113.65 and 100.92 respectively.

Another useful thing you can get out of this new view is a list of the most important descriptors in terms of predicting the target variable.



VARIABLE	NODES	IMPORTANCE
TB	5	1.000
HR	2	0.126
BB	1	0.066
SB	1	0.054
SF	2	0.045
R	1	0.003
SH	0	0.000
DOUB	0	0.000
TRIP	0	0.000
SING	0	0.000
H	0	0.000
G	0	0.000
AB	0	0.000
CS	0	0.000
HBP	0	0.000
IBB	0	0.000
SO	0	0.000

Building & Assessing a Neural Net Model

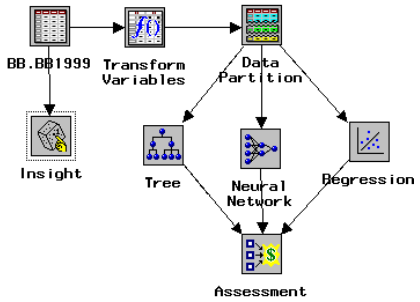
There is an excellent description and background information on Neural Networks(NN) in the online documentation of EMv4.0. One small part of it will be stated here

"The **Neural Network** node provides a variety of feedforward networks that are commonly called **backpropagation** or **backprop** networks. This terminology causes much confusion. Strictly speaking, **backpropagation** refers to the method for computing the error gradient for a feedforward network, a straightforward application of the chain rule of elementary calculus. By extension, **backprop** refers to various training methods that use backpropagation to compute the gradient. By further extension, a **backprop network** is a feedforward network trained by any of various gradient-descent techniques. **Standard backprop** is a euphemism for the **generalized delta rule,** the training technique that was popularized by Rumelhart, Hinton, and Williams in 1986 and which remains the most widely used supervised training method for feed forward neural nets. Standard backprop is also one of the most difficult to use, tedious, and unreliable training methods."

That all said, in our experience at PNU, Neural Networks can be applied to many problems and are a worthwhile addition to the any toolkit. That said, my own personal bias is that neural networks are extremely complex as there are many different architectures that one can build and many ways to influence convergence and the predictive results. They offer a nice "black box" approach if you are lucky enough to get something predictive that appears to work but the resultant parameters and weights will probably not give you much insight into the problem you are studying.

Anyway, after dragging and connecting the Neural Network Node to the Data Partition node one can and should also connect it to the assessment node. Note that at this point our diagram looks like that shown in the figure below.

After opening the neural network node, under the GENERAL tab



we could choose to set the option to select an ADVANCED INTERFACE. This allows, among other things, to define the precise architecture we desire. In the present example, we are just accepting the basic NN model. Please note that one can quite often significantly improve NN performance by changing the architecture and other settings available under the ADVANCED tab.

SAS CODE node

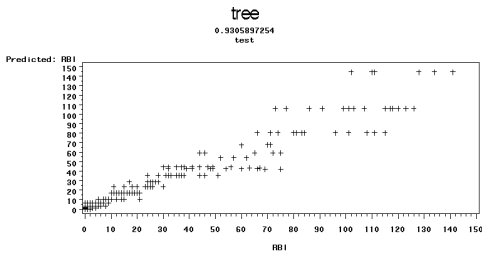
In the author's experience, EM does not (yet) provide all the functionality and reports necessary to efficiently assess and interpret the performance of the modeling. V4 is MUCH better than earlier versions but we find it still necessary to write code to generate reports and displays suited to our tastes. For example, in this BASEBALL example, we have included a SAS CODE node to examine the predicted vs observed plots for the TRAINING, VALIDATION, and TEST sets. In each we included code like that shown below to read in the appropriate datasets and get a simple plot, displaying the R^2.

```
title1 'tree';

%macro rsq(ds);
proc corr noprint data=&ds outs=outs;
var p_rbi rbi;
data a(keep=rsq); set outs;
if _type_='CORR' and rbi^=1;
rsq=rbi**2;
call symput('rs',rsq);
%mend;

%rsq(&_MAC_1);
proc gplot data=&_mac_1;
 plot p_rbi*rbi; title2 "&rs";
title3 "training";
%rsq(&_MAC_2);
proc gplot data=&_mac_2;
plot p_rbi*rbi; title2 "&rs";
title3 "valid";
%rsq(&_MAC_3);
proc gplot data=&_mac_3;
plot p_rbi*rbi; title2 "&rs";
title3 "test";
run;
```

The names of the datasets and variables used in the above code can be found under the VARIABLES and MACROS tabs in the SAS CODE node. A plot resulting from this code is shown below showing that the tree node produced a good fit in the test data set as reflected in the R^2 of over .93.



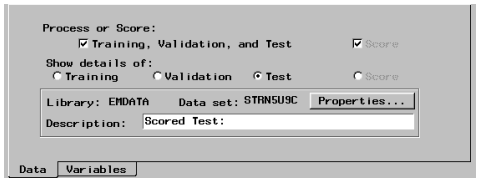
Final ASSESSMENT

After all the models have been defined, the analyses performed, and the results examined it is time to decide which model is the best. Once that decision has been made one needs to highlight the best model in the models tab of the ASSESSMENT node before generating scoring code. This is to tell the SCORE node which code to use for scoring new datasets.

Creating scoring code

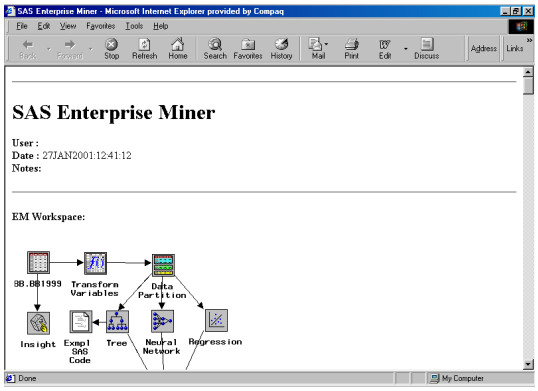
The purpose of most of the predictive modeling the author has been involved with also included the application of the "analyses" to new data. To apply the completed analyses to predict the performance of new data, the SCORE node should be connected to the ASSESSMENT node. The Score node generates and manages scoring code from the models that were run. First, though, one has to tell EM what dataset to score. This is done through another INPUT DATA node. This node is connected to the SCORE node as before to identify the dataset that is to be scored/input. We also need to define the variable(s) to be predicted. The model role for these variables must be set to PREDICT. In addition, under the SETTINGS tab of the SCORE node one needs to set the action to "Apply Training Data Score code….". This will use the previously generated scoring code to generate predictions for the appropriate variables in the input data set. For example, if we explicitly selected the neural network model to be output the corresponding scoring code is accessed.

If you are interested in developing scoring code to be used to score other datasets one has to be sure to set the appropriate box in each of the analysis nodes. For example, in the TREE node one can "click" the Process or Score box to cause the appropriate scoring datasets to be created and accessible for future nodes.



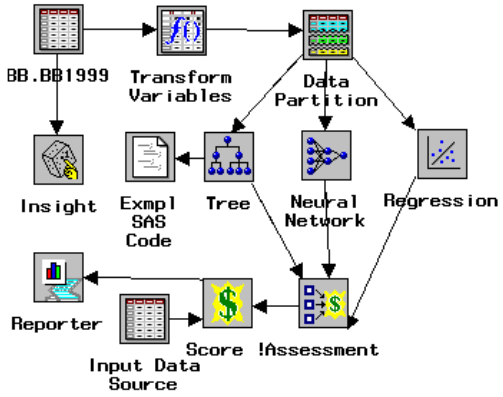
Generating an HTML report

Another neat thing about EM is that it can automatically document your mining results in HTML form! The **Reporter** node is the Enterprise Miner™ tool for drawing a detailed map of the analysis and processing performed. As you go through the browser display, one finds links to additional pages that contain tables and charts of data output, training and score code, and statistics. The report follows the flow of your data throughout the entire data mining process. Much of the information will be new to you because the **Reporter** node compiles tables and charts that describe node actions from a broad perspective. Some nodes run silently after you have set a few options; the report tells you what they've been doing. An example of the beginning display appears below.



Final Diagram

The final diagram built illustrating the analysis of the baseball dataset is shown below.



CHANGES and ENHANCEMENTS in v4.0 (SAS <8.2)

There are a variety of changes and enhancements to EM in version 4. Some of the more interesting ones are

- A new TREE view display that you must invoke via a **%Let emv4tree=1;** command. You can place it in the START UP CODE window or simply execute it in a program editor window.
- You can see the current macro variable names in the SASCODE node rather than searching for the names in the log files.
- The regression node automatically changes variables not included in the model to REJECTED.
- The Ensemble node now computes and displays fit statistics
- A Beta (experimental) version of a C*Score node to convert6 EM scoring code to C-functions.

CHANGES and ENHANCEMENTS in v4.1 (SAS 8.2)

This release contains many enhancements and new experimental tools. The following contains a short list of some of these new features and enhancements

- The C-score node is now production
- Two new production nodes: Two-stage model and Princomp/DMNeural.
  1. The Two-Stage Model does a categorical prediction and an interval-based prediction and then creates score code as a composite of both models.

2. The Princomp/DMNeural Node fits an additive nonlinear model that uses bucketed principal components as inputs to predict a binary or interval target variable. It can also perform a "stand alone" PCA and passes the scored PCs to successor nodes.

- The regression node can now handle nominal variables.
- New Experimental nodes
  1. Memory-based reasoning (based on a k-Nearest neighbor algorithm)
  2. Text Mining nodes (allows one to combine free-form text and quantitative variables to derive information)
  3. Expectation-maximization (~fuzzy) clustering
  4. Time series Node
  5. Link Analysis node (uses MDS to get useful plots and facilitate analysis)

EM DOCUMENTATION

There is actually a lot of good information on how to get the most out of Enterprise Miner™, especially introductory material. My favorites are

1. The “Getting Started with Enterprise Miner Software, Version 4" booklet. It is a very well written and easy-to-follow introduction to EM.
2. www.sas.com/software/tutorialsv8/em/mainmenu.htm. An excellent tutorial that is very short and clear and is available on the SAS.COM web site. Specifically, it provides straightforward information on how to do data mining with EM.
3. The "Predictive Modeling" section of the online Enterprise Miner™ Reference Help facility. This is a must read for anyone new to data mining that wants to produce useful and accurate predictions. In addition to the predictive modeling section,

There is a really quite a large body of help information on virtually every aspect of EM

TROUBLESHOOTING

There are a variety of situations from which you may need to extricate yourself. In preparing for this talk I encountered one of these and was rescued by the troubleshooting tips located in the “Getting started with Enterprise Miner v4.0” page 32. When I was locked out of a particular diagram the trick that worked most often for me was to delete any *.lck file I could. Also you might want to delete any users in the corresponding USERS folder. Inappropriate entries here would put the project in SHARED mode so that you couldn't edit the start-up code for example. For other situations, please refer to page 32 in that excellent introduction for the details.

CONCLUSION

Enterprise Miner™ is an awesome product containing a variety of flexible tools to support data mining and knowledge discovery. Several papers have now been written describing applications of EM in a variety of industrial areas[3]. While it is a very powerful product, it is also a bit intimidating to use. Hopefully this beginning tutorial paper has helped you gain a better understanding of this product and has made any future utilization of the software easier.

ACKNOWLWDGEMENTS

I'd like to thank Dr. Wayne Thompson of SAS Institute for his assistance and support.

REFERENCES

1. This baseball dataset is available from the author and is based on historical batting statistics for major league baseball players.
2. Please refer to www.spotfire.com for information on this visualization software.
3. “Using Enterprise Miner™ to Explore and Exploit Drug Discovery Data”. M.S.Lajiness SUGI 25 Proceedings. Paper 266-25.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Mic Lajiness
7247/267/124
Pharmacia Corporation
301 Henrietta St
Kalamazoo, Michigan 49008
616-833-1794(w):
616-833-9183(fax)
michael.s.lajiness@am.pnu.com