

**Title: Using the Component Approach to Craft Customized Data Mining Solutions**

**Summary:** One definition of data mining is the non-trivial extraction of implicit, previously unknown and potentially useful knowledge from data. Another definition is a variety of techniques used to identify nuggets of information or decision-making knowledge in bodies of data. The intimate details of how a business functions, and the nature of the data that it has at its disposal, help to shape the most meaningful data mining analyses. This paper includes examples that show how employing a component approach to automate data mining inquiries can be extremely efficient, relatively straightforward, and can consistently generate trustworthy results.

---

While some business problems lend themselves to pre-packaged applications, there are many circumstances where meaningful computation is only possible with custom solutions. This is a very common problem in business applications utilizing data mining techniques where company-specific requirements surpass the ability of off-the-shelf data mining packages in several respects.

A key problem is that IT managers often find the currently available data mining systems to be inflexible monoliths that often lack one or more critically needed functions. Additionally, there can be troublesome variations from one data mining package to another in terms of the algorithms used and their ultimate outcome.

For example, there is no cross-industry standard practice for how classification functions deal with ties in the data. Although each classifier built using a different method of dealing with ties is equally valid, its classifications may differ from the others. This is inherently confusing to end users who expect functions with the same name to give equal results on a given data set.

However, reinventing routines for data mining exercises can be prohibitively time-consuming and expensive, and homegrown code for data mining computations may not be fully trustworthy, given the stringent requirements needed for computations handling terabyte-sized data.

For these reasons, a fully-documented component approach, using data mining functions from static or shareable libraries to develop data mining applications, is growing in popularity as the most effective means to harness the power of data mining techniques.

For many years, data mining practitioners used several routines published in shareable libraries by the worldwide Numerical Algorithms Group (NAG). Last year, in response to the growth in popularity of data mining techniques, NAG introduced a fully documented shareable library of data mining algorithms entitled the NAG Data Mining Components. These components are written in ANSI C and can be easily called from other programming languages.

In this white paper, we will example how components can be used to speed the development of data mining applications that are tailored to specific business analytics situations.

### **CRM Case Study**

From a company's perspective, Customer Relationship Management (CRM) helps improve the profitability of interactions with customers. From a customer's perspective, companies that effectively use CRM are seen as having an individual touch. These points make CRM an essential tool for any modern business. However, in order to make CRM work effectively, companies need to match marketing campaigns and products to their customer's needs.

The traditional approach to customer acquisition is to combine mass marketing (e.g., magazine advertisements) with direct marketing techniques (e.g., postal offers). In both cases, the targets for a marketing campaign are determined by a set of demographics chosen by a marketing team.

For example, consider the case of a bank trying to entice its customers into subscribing to a new credit card. Suppose the bank contacts one million of its customers with the invitation to subscribe to a new credit card and gets sixty thousand replies. However, only sixteen percent of these replies are from customers with an acceptable credit rating. At a cost of \$1 per mail-shot, the cost of the marketing campaign is one million dollars. If each of the 9,600 “qualified” customers accepts the offer and, on average, the profit per cardholder is \$125, the net profit to the bank from the marketing campaign would be \$200,000.

Now suppose that the bank had conducted the same marketing program using a component approach, such as the NAG Data Mining Components model. In order to use this approach, a random subset of 50,000 of the responses and the subsequent credit check are used as historical data to build analytical CRM models. The independent variables used to predict values of the response (dependent) variable may include any relevant and admissible data that the bank holds on its customers. This data is likely to be socio-economic information and details of the customers’ dealings with the bank.

The historical data is used to design and build two predictive models. The first model indicates which of the bank’s customers are likely respondents. The second model indicates which of the respondents are likely to have an acceptable credit rating. Each customer that is deemed by the respective models to be a respondent and an acceptable credit risk is a candidate for being contacted by the bank.

Any historical data suitable for use in the new marketing campaign is likely to need pre-processing before a predictive model can be built, which may involve one or more of the following:

- Data extraction from a database;
- Data cleaning to covert character string values into a numeric representation, and to deal with missing values;
- Scaling data. For some data mining techniques, it is advisable to normalize data values on some or all variables;
- If the number of variables in the data is high, principal components analysis can be used to calculate a lower number of variables without the loss of information. This approach will often lead to more accurate predictive models.

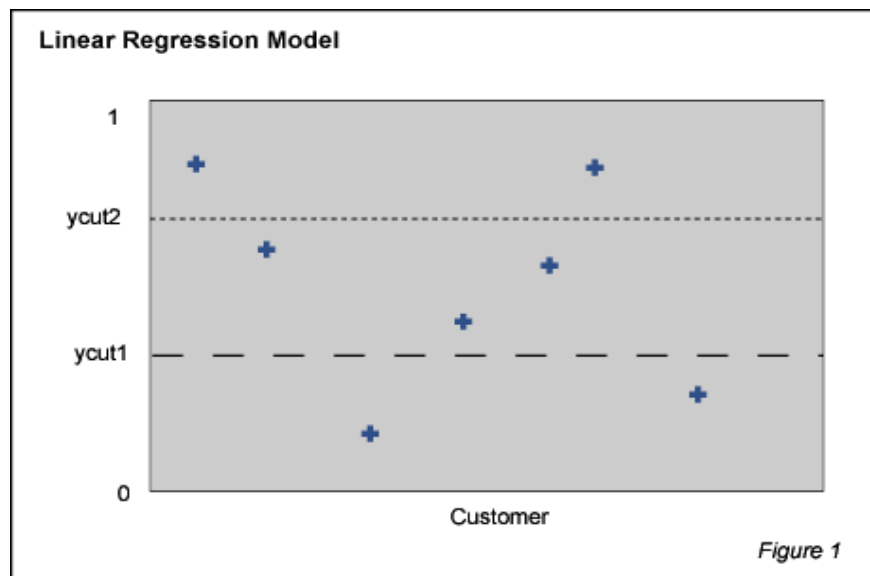
A predictive model can be calculated by using one of the NAG Data Mining Components: for logistic regression; for regression; or for decision tree analysis. The most suitable component is found by means of trial and error using the historical data and the evaluation given below.

Logistic regression requires several passes through the data to fit the model and, therefore, is the most expensive computationally of these three models.

By treating the outcome classification as a continuous dependent variable, a linear regression model can be computed. An advantage of using this predictive model is that the fitting stage requires only one pass through the data.

Information in a fitted regression model (logistic or linear) can be used to calculate predictions, give estimates on residuals and to quantify the influence each independent variable has on the dependent variable. The latter point may be used to refine the model by, for example, removing some variables deemed to have little influence on the dependent variable.

For the case of a linear regression model, the user needs to interpret the continuous output values as discrete classes. In order to do so, the value of a parameter, say **ycut**, needs to be set as described below. The role of the **ycut** parameter is to threshold a continuous-valued output such that any value less than **ycut** is set to 0 and all other values set to 1, as shown in Figure 1.



**Figure 1:** Setting a value of **ycut** that suits your data is an important modeling step. In this figure, the value *ycut1* sets makes 5 out of 7 customers likely respondents, whereas the value *ycut2* makes only 2 out of 7 customers likely respondents.

A decision tree model is computed by using a single pass through the data. Each node in a computed tree represents a decision or test on the historical data. However, because these tests involve only one independent variable, complicated problems can lead to very large decision trees. Such decision trees try to classify new data by applying a very complicated set of decisions and, therefore, may not generalize well to new data. Some decision trees use validation data to prune a computed tree, whereas others use early-stopping criteria to reduce the size of the initial decision tree.

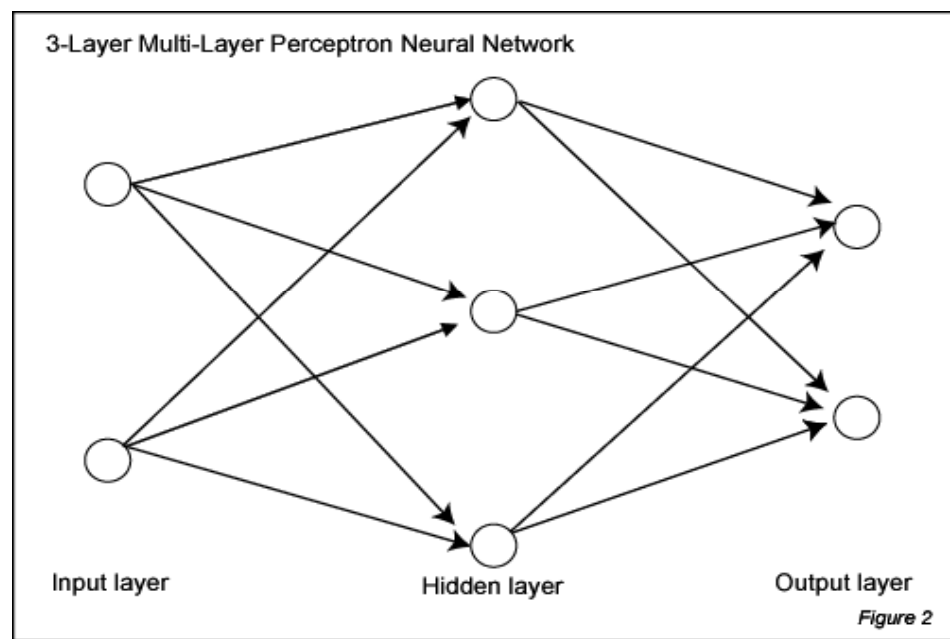
A fitted decision tree is scored to classify new data. However, unlike the regression models, the user cannot control the number of customers identified as likely respondents. Instead, the decisions the tree takes based on the historical data can be analyzed by tracing through the recursive tree structure. An analyst can then check to see if the decisions a tree takes based on the historical data are sensible.

The second stage of this analytical CRM application is to build a model to predict the credit rating of the anticipated respondents. The credit rating model can be built by using any of the following NAG Data Mining Components: for regression; for multi-layer perceptron neural networks; for nearest neighbors classification making use of prior information; or for nearest neighbors prediction.

The most suitable component is usually found by means of trial and error using the historical data. The suitability of any single component may be evaluated by using the method described below.

A feed-forward multi-layer perceptron consists of an input layer, one hidden layer and an output layer. Each layer is made up of nodes. The number  $d$  of nodes in the input layer is the same as the number of independent variables. The number  $c$  of nodes in the output layer is the same as the number of dependent variables. In general, for classification problems with more than two classes, dummy variables should be calculated for the dependent variable.

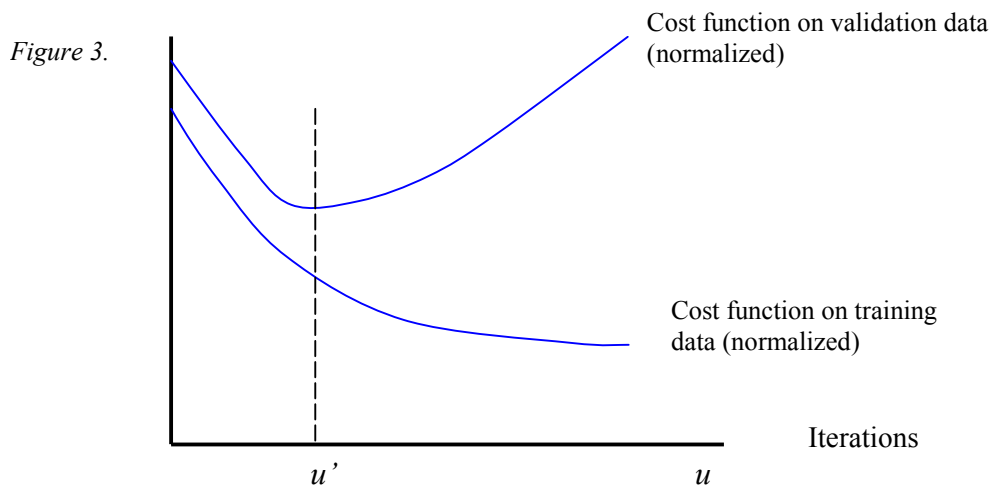
Each node in the input layer is connected to each node in the hidden layer by means of a set of first-layer weights. Each node in the hidden layer is connected to each node in the output layer by means of a set of second-layer weights. Each weight is a real-valued scalar and the determination of suitable values for these weights is known as training. Figure 2 depicts a 3-layer multi-layer perceptron (MLP) with three hidden nodes.



**Figure 2:** The design of a feed-forward, fully-connected multi-layer perceptron. The circles in the diagram represent nodes in a layer. From the left, each node is connected to the nodes in the layer to the right by a set of weights represented by arrows. The weights connecting the input layer to hidden layer are known as first-layer weights, whereas the weights connecting the hidden layer to the output layer are called second-layer weights.

The neural network is trained by using an optimization algorithm to minimize the value of a sum of squares error cost function.

In practice, it is advisable to halt training when the value of the cost function reaches a minimum on a set of validation data. Otherwise, as seen in Figure 3, after a critical number of iterations ( $u'$ ), the value of the cost function on validation data increases while the value of the cost function on training data continues to decrease. This phenomenon is known as over-fitting the data.



**Figure 3:** Although the value of the cost function on training data continues to decrease, the value of the cost function on validation reaches a minimum after  $u'$  iterations.

The number  $p$  of nodes in the hidden layer determines the number of free parameters in the model. The following guidelines are useful when determining the number of nodes in a hidden layer for a given data set of  $n$  data.

- The expression:  $10n > (d + c + 2)p$  should be true.
- Given a data set, the length of time taken per iteration during training is directly proportional to  $p$ .
- The higher the value of  $p$ , the more likely the model is to over-fit data.

For neural network models, the user needs to interpret the continuous output values as discrete classes. In order to do so, the value of a parameter, say **ycut**, needs to be set (see Figure 1).

Nearest neighbor algorithms calculate their outputs by comparing new data to historical data and searching for similar examples. The measurement of similarity is a distance function. The user must determine a suitable number  $k$  of nearest neighbors to search for, bearing in mind the following:

- In general, the computational time taken to search for nearest neighbors increases linearly with increasing values of  $k$ .
- Lower values of  $k$  (e.g.,  $k < 25$ ) lead to approximations that are more sensitive to variations in the historical data.

In order to optimize the value of  $k$ , the historical data should be partitioned into training and validation sets. Typically, this is achieved by taking a random permutation of the training data and allocating the first  $l$  data to the training set and the remaining data to the validation set. The accuracy of predictions on the validation set can be used to set a value for  $k$ .

Nearest neighbors approximations can be either continuous or discrete. In the continuous case, the user must post-process the classifications of new data to give a suitable number of likely respondents as described below.

For each predictive model, the results are evaluated by using the historical data.

Popular metrics to measure the success, or otherwise, of a customer acquisition campaign are return on investment (ROI) and the profit. (Note that often it is not helpful to consider the accuracy of a model: if a model predicts that no customers are suitable for a mail-shot it will have a high accuracy, but will be entirely useless.) For the example at hand, the profit,  $P$ , is given by:

$$P = 125t - s - c,$$

where:

- $s$  is the number of customers likely to be offered the chance to subscribe to the new service;
- $t$  is the number of customers who responded positively from the  $s$  customers identified by the predictive models. On average, each of these customers makes a profit of \$125 for the bank;
- $c$  is a scalar constant the value of which depends on the start-up costs of the project.

For components that output continuous values, further post-processing stages are required in order to obtain a value for  $s$ .

Using the historical data, the user should optimize values for a **ycut** parameter (see Figure 1) with respect to the chosen criterion, e.g., ROI or profit.

The two predictive models fitted using the historical data are used to calculate which customers of the remaining 950,000 in the bank's database are likely respondents to the credit card offer. Suppose that 600,000 of these customers are deemed likely to respond to an offer and that 8,000 of them did so. Assuming the start-up cost for the project is \$10,000 and that, on average, each new card-holder creates a profit of \$125 for the bank the profit,  $P$ , is given by:

$$\begin{aligned} P &= 125 \times 8000 - 600000 - 50000 - 10000, \\ &= \$340,000. \end{aligned}$$

By marketing their offer to one million people in their database, the bank made a profit of \$200,000. However, with a more refined analytical model (using for example, NAG Data Mining Components), the profit would have been \$340,000, representing an increase in profit of 70%. This CRM model could now be used as a key tool in campaigns to market similar new products.

Analytic CRM models can be built for other products by posting offers to a fraction of its mailing list and using the responses to this data as the historical data.

### Web-mining Analysis

Each time a customer or potential customer interacts with a website, information on the interaction is logged by the web server. Such logs are a potential gold mine of information and can be used to help design web sites that are more effective and thus improve profitability. This section describes how NAG Data Mining Components could be used for a web-mining task.

The information logged takes the form of numerous log files such as: access logs that record each "hit" on a website; agent logs that record the software used to browse the website; cookie logs that, if enabled by the user, record multiple requests of a web-page from a single user; referrer logs that record the URL of the referral page, in addition to the URL of the current page; and elf logs that are defined by the website administrator and may contain the information stored in access logs, agent logs and referrer logs.

The information in the logs recorded by a web-server is not in a suitable form for data mining. In general, for web-mining applications a considerable length of time must be spent pre-processing the log data.

Firstly, the analyst must determine the goal of the data mining exercise. Typically, this is to either personalize the website for each potential customer or to improve the ease with which potential customers can find what they want on the website. By achieving one or both of these goals, the website is likely to be more profitable in the future.

Figure 4 displays some content from a typical web-server's log file. In the table, the entries are logged according to the time at which an action was requested from a client. Consequently, the client's IP address must be used to track the actions of a user on a particular website and can be used to calculate the following data:

- The first and last entries of a user's IP address can be used to calculate the duration of time that an individual remained on the website.
- The time taken on each web page can be calculated to detect pages of particular interest (or disinterest).
- Following the path an individual takes while connected to a website, a process known as click stream analysis.

#### Typical content from a web-server's log file

Date	Time	Client's IP Address	Server's IP Address	Method	File
2001-11-09	09:43:09	www.xxx.yy.zz	aaa.bb.ccc	GET	/home.html
2001-11-09	09:43:12	www.xxx.yy.zz	aaa.bb.ccc	GET	/products.html
2001-11-09	09:43:13	ddd.ee.fff.gg	aaa.bb.ccc	GET	/home.html
2001-11-09	09:43:15	www.xxx.yy.zz	aaa.bb.ccc	GET	/orders/purchase.asp
2001-11-09	09:44:16	ddd.ee.fff.gg	aaa.bb.ccc	GET	/site_map.html
2001-11-09	09:44:18	hhh.iii.jj.kk	aaa.bb.ccc	GET	/home.html
2001-11-09	09:44:20	ddd.ee.fff.gg	aaa.bb.ccc	GET	/home.html

Figure 4.

Personalizing a website means finding groups of customers who, for a given task, exhibit similar behavior. When such groups are found, information that describes each group is calculated. This information is used to profile a group of customers and can be used to target new users of the website with information relevant to their most suitable profile and/or change the design of the website to appeal to users with the most profitable profiles.

For example, assume that a website consists of six key pages, namely the homepage, site map, products page, services page, "about us", and purchasing page. Such data is then taken to be the raw data for finding profiles of users that either reach or leave without visiting the purchasing page. The data mining exercise can now begin by preparing the raw data.

### Example of Raw Data from Log Files

Userid	Home	Products	Services	Site_Map	About_Us	Time	Purchase
1	N	Y	N	N	N	7.2	Y
2	Y	Y	N	Y	N	18.2	N

Figure 5.

**Figure 5:** Data gathered from log files describing the user, the pages visited by the user on the website, the time spent on the website, and if the user purchased anything during the visit.

Before a model can be built using NAG Data Mining Components the data must be in numeric form and free from missing values. If the number of variables in the data is high, principal components analysis can be used to calculate a lower number of variables without the loss of information. This method will often lead to more accurate prediction.

Once the raw data is prepared, groups of similar users must be found. The NAG Data Mining Components contain two clustering functions for this purpose.

For the  $k$  means clustering component, the analyst must set the number,  $k$ , of clusters. This component has the advantage of being suitable for high numbers of data, but the disadvantage is that the analyst is unlikely to know a suitable value for  $k$  given any one data set.

The hierarchical clustering component can be used to find a suitable number of naturally occurring groups in the data. This method is more intensive computationally and is unsuitable for a high number of data, but has the advantage that the user is not required to set the number of clusters.

Once a set of clusters has been found, the analyst needs to describe the data belonging to each cluster, and to calculate summary statistics describing each cluster.

If the profile of one or more clusters is that users tend to spend their time on the homepage and site map, this may indicate that users are looking for something in particular and not finding it. Furthermore, if a significant number of users are allocated to the same cluster, these pages should be designed again—possibly with a simpler structure.

In addition, cluster profiles can be used to find common interests of groups of users. This information can be used to design web pages specifically for groups, and therefore offer the opportunity to cross-sell products.

Thus, by finding groups representing similar types of users, the analyst is able to get an insight into how users navigate a website. Potential problems, such as users that seem unable to find what they need, can be identified as can the way in which users buy products from the website. Both of these insights can be incorporated into an improved design of the website, which, in turn, is likely to lead to a higher number of online purchases.

### Conclusion

One definition of data mining is the non-trivial extraction of implicit, previously unknown and potentially useful knowledge from data. Another definition is a variety of techniques used to identify nuggets of information or decision-making knowledge in bodies of data. The intimate details of how a business functions, and the nature of the data that it has at its disposal, help to shape the most meaningful data mining analyses. As these examples show, employing a component approach to automate data mining

inquiries can be extremely efficient, relatively straightforward, and can consistently generate trustworthy results.

By: Stephen Langdell Ph.D., a member of the Data Analysis and Visualization Group, and Rob Meyer, President of the Numerical Algorithms Group ((NAG, [www.nag.com](http://www.nag.com)), a worldwide organization dedicated to developing quality mathematical and statistical components and 3D visualization software. Inquiries can be forwarded to [Rob@nag.com](mailto:Rob@nag.com).

---

Originally published by Computer Economics, Inc. CEI is an independent research organization that specializes in providing economic and strategic analysis and data to IT and business executives. CEI's goal is to help IT management develop effective strategic and tactical plans, more efficiently manage IT costs, and maximize their return on IT investments ([www.computereconomics.com](http://www.computereconomics.com)).

**Numerical Algorithms Group**

[www.nag.com](http://www.nag.com)

[infodesk@nag.com](mailto:infodesk@nag.com)