EMERGE Theory

Introduction

EMERGE is a program that uses a combination of multiple 3D or 2D seismic attributes to predict some reservoir parameter of interest. The idea of using multiple seismic attributes to predict log properties was first proposed by Schultz, Ronen, Hattori and Corbett in a series of three articles in the Leading Edge. (Seismic-guided estimation of log properties, Parts 1, 2 and 3 by Schultz et al, The Leading Edge, May, June and July, 1994). In these articles, they point out that the traditional approach to using seismic data to derive reservoir parameters has consisted in looking for a physical relationship between the parameter to be mapped and some attribute of the seismic data, and then using that single attribute over a 2D line or 3D volume to predict the reservoir parameter. Although relationships have been inferred between these attributes and reservoir parameters, the physical basis is not always clear, and they proposed deriving statistical, rather than deterministic, relationships. This approach, which Schultz et al call a data-driven methodology, is summarized in this flow-chart:

The data-driven statistical interpretation (after Schultz et al)
Although Schultz et al proposed the basic philosophy of multi-attribute analysis, the implementation of the technique in EMERGE is substantially different from their method, and we do not propose to discuss their method further. Essentially, EMERGE consists of four steps:

1. Read in seismic data and well logs that tie the data.
2. Manually calibrate the well logs to the seismic data.
3. Train the seismic to predict the reservoir parameter of interest at the tie locations.
4. Apply the results of the training to the seismic volume.

Points (1) and (2) above are obviously fairly standard seismic methods, and the technique that is used in EMERGE for manual calibration involves the techniques of wavelet extraction and cross-correlation. The heart of EMERGE is therefore in steps (3) and (4), and a discussion of how we implement them is the basis of this document.

Mathematically, we refer to these techniques as multivariate geostatistics, which is a broad term that encompasses all methods that utilize more than one variable to predict some other variable of interest. Bivariate geostatistics is obviously the simplest subset of multivariate techniques and thus the technique of cokriging (an extension of kriging) could be termed multivariate geostatistics. (Indeed, Hampson-Russell produces a program called ISMap, which is based on kriging and cokriging methods.) However, in EMERGE we use a technique that is more readily extended to more than two variables, the generalized linear regression matrix approach, which will produce a convolutionally weighted sum of the attributes.

The detailed discussion of this technique is contained in the following pages, which is a reprint of a paper published in Geophysics in January - February, 2001.
Use of multiattribute transforms to predict log properties from seismic data

Daniel P. Hampson*, James S. Schuelke‡, and John A. Quirein**

ABSTRACT

We describe a new method for predicting well-log properties from seismic data. The analysis data consist of a series of target logs from wells which tie a 3-D seismic volume. The target logs theoretically may be of any type; however, the greatest success to date has been in predicting porosity logs. From the 3-D seismic volume a series of sample-based attributes is calculated. The objective is to derive a multiattribute transform, which is a linear or nonlinear transform between a subset of the attributes and the target log values. The selected subset is determined by a process of forward stepwise regression, which derives increasingly larger subsets of attributes. An extension of conventional crossplotting involves the use of a convolutional operator to resolve frequency differences between the target logs and the seismic data.

In the linear mode, the transform consists of a series of weights derived by least-squares minimization. In the nonlinear mode, a neural network is trained, using the selected attributes as inputs. Two types of neural networks have been evaluated: the multilayer feedforward network (MLFN) and the probabilistic neural network (PNN). Because of its mathematical simplicity, the PNN appears to be the network of choice.

To estimate the reliability of the derived multiattribute transform, crossvalidation is used. In this process, each well is systematically removed from the training set, and the transform is rederived from the remaining wells. The prediction error for the hidden well is then calculated. The validation error, which is the average error for all hidden wells, is used as a measure of the likely prediction error when the transform is applied to the seismic volume.

The method is applied to two real data sets. In each case, we see a continuous improvement in predictive power as we progress from single-attribute regression to linear multiattribute prediction to neural network prediction. This improvement is evident not only on the training data but, more importantly, on the validation data. In addition, the neural network shows a significant improvement in resolution over that from linear regression.

INTRODUCTION

The integration of well-log and seismic data has been a consistent aim of geoscientists. This has become increasingly important (and successful) in recent years because of the shift from exploration to development of existing fields, with large numbers of wells penetrating them. One type of integration is forward modeling of synthetic seismic data from the logs. A second type of integration is inverse modeling of the logs from the seismic data. This is called seismic inversion and has been described by numerous authors (e.g., Lindseth, 1979; Cooke and Schneider, 1983; Oldenburg et al., 1983; and Chi et al., 1984).

We attempt to go beyond the limits of conventional seismic inversion in several ways. First, we will directly predict log properties other than acoustic impedance, such as porosity. This differs from previous authors who have usually modeled porosity from the impedance derived from the inversion (Anderson, 1996). A second difference is that we will use attributes derived from the seismic data rather than the conventional poststack data. This allows us to include prestack information as well as nonlinear transforms of the poststack data. Third, instead of assuming a particular model relating the logs and the seismic data, a statistical relationship will be derived by analyzing a set of training data at well locations. This relationship will be either

Manuscript received by the Editor May 14, 1999; revised manuscript received April 10, 2000.

*Hampson-Russell Software Services Ltd., 510–715 Fifth Avenue SW, Calgary, Alberta T2P 2X6, Canada; E-mail: dan@hampson-russell.com.
‡Formerly Mobil Technology Company, Dallas, Texas; presently ExxonMobil Upstream Research Company, Houston, Texas. E-mail: james.s.schuelke@email.mobil.com.
**Formerly Mobil Technology Company, Dallas, Texas; presently Halliburton Energy Services, Houston, Texas. E-mail: john.quirein@halliburton.com.

© 2001 Society of Exploration Geophysicists. All rights reserved.
linear (multivariate regression) or nonlinear (neural network). Finally, we will use the concept of crossvalidation to estimate the reliability of the derived relationship.

After describing the theoretical basis of the method, two real data examples, which emphasize the enhanced resolution obtainable from the method, will be shown.

MULTIATTRIBUTE LINEAR REGRESSION

Seismic attributes

In this methodology, our aim is to find an operator, possibly nonlinear, that can predict well logs from neighboring seismic data. In fact, we choose to analyze not the seismic data itself but attributes of the seismic data. One reason why we expect this to be more beneficial than the raw seismic data is that many of these attributes will be nonlinear, thus increasing the predictive power of the technique. A second reason is that there is often benefit in breaking down the input data into component parts. This process is called preprocessing or feature extraction, and it can often greatly improve the performance of a pattern recognition system by reducing the dimensionality of the data before using it to train the system. Preprocessing can also provide a means of adding prior knowledge into the design of the pattern recognition system.

We define a seismic attribute generally as any mathematical transform of the seismic trace data. This, for example, includes simple attributes such as trace envelope, instantaneous phase, and instantaneous frequency as well as complicated attributes such as seismic trace inversion and AVO. The transform may or may not incorporate other data sources. For example, trace inversion assumes other data sources, such as the seismic wavelet, the initial guess model, and constraints. However, for this analysis we still consider the inversion result to be an attribute of the seismic trace.

As pointed out by Chen and Sidney (1997), seismic attributes may be divided into two categories:

1) horizon-based attributes, the average properties of the seismic trace between two boundaries, generally defined by picked horizons, and

2) sample-based attributes, the transforms of the input trace in such a way as to produce another output trace with the same number of samples as the input.

In this paper, we consider only sample-based attributes. In theory, it should be possible to transform the horizon-based attribute into a sample-based attribute by repeating the average property as many times as there are output samples between the bounding horizons. However, we do not address that possibility in this paper.

Taner et al. (1994) provide a long list of sample-based attributes. Our experience has shown the following to be particularly important for log prediction: (1) integrated trace, (2) integrated absolute amplitude of the trace, (3) near-angle stack, (4) AVO intercept/gradient, (5) frequency and absorption estimates, and (6) seismic velocity.

Conventional crossplotting

Given a particular attribute of the seismic data, the simplest procedure for deriving the desired relationship between target data and seismic attribute is to crossplot the two.

Figure 1 shows an example in which a target log property, in this case density-porosity, is plotted against a seismic attribute. The assumption is that the target log has been integrated to traveltime at the same sample rate as the seismic attribute. Effectively, this integration reduces the target log to the same resolution as the attribute, which is usually significantly coarser than the log property. Each point in the crossplot consists of a pair of numbers corresponding to a particular time sample.

Assuming a linear relationship between the target log and the attribute, a straight line may be fit by regression:

\[ y = a + bx. \]  

(1)

The coefficients \( a \) and \( b \) in this equation may be derived by minimizing the mean-squared prediction error:

\[ E^2 = \frac{1}{N} \sum_{i=1}^{N} (y_i - a - bx_i)^2, \]

(2)

where the sum is over all points in the crossplot.

The calculated prediction error \( E \) is a measure of the goodness-of-fit for the regression line defined by equation (1). An alternative measure is the normalized correlation coefficient, defined by

\[ \rho = \frac{\sigma_{xy}}{\sigma_x \sigma_y}. \]

(3)

where

\[ \sigma_{xy} = \frac{1}{N} \sum_{i=1}^{N} (x_i - m_x)(y_i - m_y), \]

(4)

\[ \sigma_x = \frac{1}{N} \sum_{i=1}^{N} (x_i - m_x)^2, \]

(5)

\[ \sigma_y = \frac{1}{N} \sum_{i=1}^{N} (y_i - m_y)^2, \]

(6)

\[ m_x = \frac{1}{N} \sum_{i=1}^{N} x_i, \]

(7)

![FIG. 1. Conventional crossplot between the target log density-porosity and the seismic attribute.](image)

Using Attributes to Predict Logs 221
and

\[ m_y = \frac{1}{N} \sum_{i=1}^{N} y_i. \]  \( (8) \)

Note that the linear requirement may be relaxed somewhat by applying a nonlinear transform to either the target data or the attribute data or both, as shown in Figure 2.

**Extension of cross plotting to multiple attributes**

The extension of the conventional linear analysis to multiple attributes (multivariate linear regression) is straightforward. Assume, for simplicity, that we have three attributes as shown in Figure 3. At each time sample, the target log is modeled by the linear equation

\[ L(t) = w_0 + w_1 A_1(t) + w_2 A_2(t) + w_3 A_3(t). \]  \( (9) \)

The weights in this equation may be derived by minimizing the mean-squared prediction error, as extended from equation (2):

\[ E^2 = \frac{1}{N} \sum_{i=1}^{N} (L_i - w_0 - w_1 A_{1i} - w_2 A_{2i} - w_3 A_{3i})^2. \]  \( (10) \)

As shown in the Appendix, the solution for the four weights produces the standard normal equations:

\[
\begin{bmatrix}
w_0 \\
w_1 \\
w_2 \\
w_3
\end{bmatrix} = \left[ \begin{bmatrix}
N & \sum A_{1i} & \sum A_{2i} & \sum A_{3i} \\
\sum A_{1i} & \sum A_{1i}^2 & \sum A_{1i} A_{2i} & \sum A_{1i} A_{3i} \\
\sum A_{2i} & \sum A_{1i} A_{2i} & \sum A_{2i}^2 & \sum A_{2i} A_{3i} \\
\sum A_{3i} & \sum A_{1i} A_{3i} & \sum A_{2i} A_{3i} & \sum A_{3i}^2
\end{bmatrix} \right]^{-1} \times \begin{bmatrix}
\sum L_i \\
\sum A_{1i} L_i \\
\sum A_{2i} L_i \\
\sum A_{3i} L_i
\end{bmatrix}.
\]  \( (11) \)

Just as in the single-attribute case, the mean-squared error \( (10) \) calculated using the derived weights constitutes a goodness-of-fit measure for the transform, as does the normalized correlation defined in equation (3), where the \( x \)-coordinate is now the predicted log value and the \( y \)-coordinate is the real log value.

**Use of the convolutional operator**

The derivation of the multiattribute regression assumes a single weight for each attribute. The problem with this approach is illustrated in Figure 4. This figure shows that the frequency content of the target log is typically much higher than that of the seismic attribute. Consequently, correlating the log with the attributes on a sample-by-sample basis may not be optimal. The alternative is to assume that each sample of the target log is related to a group of neighboring samples on the seismic attribute, as shown in Figure 5.

The use of the convolutional operator is also suggested by the classic convolutional model in geophysics. If the well log, for example, happens to be acoustic impedance, then the five-point operator shown in Figure 5 is closely related to the seismic wavelet. In general, for any other log property we can expect...
the wavelet to smear the effects of each log sample over a range of contiguous seismic samples.

The extension of equation (9) to include the convolutional operator is

\[ L = w_0 + w_1 * A_1 + w_2 * A_2 + w_3 * A_3, \]

where * represents convolution and \( w_i \) are operators of a specified length. Note that the number of coefficients has now increased to (number of attributes times operator length) + 1. Once again, the operator coefficients may be derived by minimizing the mean-squared prediction error:

\[ E^2 = \frac{1}{N} \sum_{i=1}^{N} (L_i - w_0 + w_1 * A_1 + w_2 * A_2 + w_3 * A_3)^2. \]

As shown in the Appendix, this is equivalent to introducing a series of new attributes, which are time-shifted versions of the original attributes.

**Determining attributes by stepwise regression**

In the previous sections, we derived equations that allow us to determine optimal operators for any given set of attributes. These operators are optimal in the sense that the mean-squared prediction error between the actual target logs and the predicted target logs is minimized. The next issue to be addressed is how to select the attributes.

One possible procedure could be exhaustive search. Assume that we want to find the best \( M \) attributes out of a total list of \( N \) attributes for a given operator length \( L \). One obvious procedure is to try all combinations of \( M \) attributes. For each combination, the optimal weights are derived using equation (11) above. That combination with the lowest prediction error is then selected.

The problem with exhaustive search is that the computation time can very quickly become excessive. Suppose, for example, that we have a total of \( N = 25 \) attributes and we wish to derive the best combination of \( M = 5 \) attributes for an operator of length \( L = 9 \). In this case, there are \( 25 * 24 * ... * 21 = 6375600 \) combinations of five attributes to be checked. Each of these combinations requires the solution of a linear system with \( 5 * 9 + 1 = 46 \) unknowns.

A much faster, although less optimal, procedure is called stepwise regression (Draper and Smith, 1966). The assumption in this procedure is that if the best combination of \( M \) attributes is already known, then the best combination of \( M + 1 \) attributes includes the previous \( M \) attributes as members. Of course, the previously calculated coefficients must be rederived. The process is illustrated in this series of steps.

First, find the single best attribute by exhaustive search. For each attribute in the list, e.g., amplitude-weighted phase, average frequency, apparent polarity, etc., solve for the optimal coefficients and calculate the prediction error. The best attribute is the one with the lowest prediction error, or attribute1.

Second, find the best pair of attributes, assuming that the first member is attribute1. For each other attribute in the list, form all pairs, e.g., (attribute1, amplitude-weighted phase), (attribute1, average frequency), etc. For each pair, solve for the optimal coefficients and calculate the prediction error. The best pair is the one with the lowest prediction error. Call this second attribute from the best pair attribute2.

Third, find the best triplet of attributes, assuming that the first two members are attribute1 and attribute2. For each other attribute in the list, form all triplets, e.g., (attribute1, attribute2, amplitude-weighted phase), (attribute1, attribute2, average frequency), etc. For each triplet, solve for the optimal coefficients and calculate the prediction error. The best triplet is the one with the lowest prediction error. Call this third attribute from the best triplet attribute3.

Carry on this process as long as desired.

The first thing to note is that the computation time for this process is much shorter than for exhaustive search. For the example above, the number of combinations to check is now \( 25 + 24 + ... + 21 = 115 \), instead of \( 6375600 \). In addition, the size of the linear system to be solved starts at \( 9 + 1 = 10 \) for the first 25 combinations and increases linearly to \( 5 * 9 + 1 + 46 \) for the last 21 combinations.

The problem with stepwise regression is that we cannot be sure of deriving the optimal solution. In other words, the combination of five attributes found may not be the best found by exhaustive search. However, each additional attribute found has a prediction error less than or equal to the previous smaller combination. This can be proven by contradiction: if the new prediction error is greater, then simply set all the weights to zero for this new attribute, and the prediction error will be equal to the previous set.

One advantage of stepwise regression is that it relieves us from the need to worry about whether the attributes in the total list are linearly independent. This is because stepwise regression automatically chooses the next attribute whose contribution in a direction orthogonal to the previous attributes is greatest. Assume, for example, that two of the attributes, say, \( A_i \) and \( A_j \), are scaled versions of each other: \( A_j = a + b * A_i \). This would represent the extreme case of linear dependence.

As the stepwise regression proceeds, one or the other of them will be chosen first, say, \( A_i \). From then on, the other attribute, \( A_j \), will never be chosen. This is because once \( A_i \) is included, the improvement by adding \( A_j \) is precisely zero. In summary, because we are using stepwise regression, we can have any arbitrary total attribute list, and the only penalty incurred by using linearly dependent attributes is computation time.

At this point, we can define the general term multiattribute transform as a set of attribute types along with rules for
transforming the attributes into the desired output log. In the analysis so far, the transformations are linear weights applied to either the attributes themselves or to nonlinear transforms of the attributes. The next section extends this analysis to include neural networks.

**NEURAL NETWORKS**

The analysis so far has been linear. The limitation this imposes can be understood by examining Figure 6.

This figure shows a target log, called P-wave, crossplotted against a single seismic attribute. As before, the regression line has been calculated by minimizing the mean-squared prediction error. Visually, we might guess that a higher order curve would fit the points better. A number of options exist for calculating this curve. One option is to apply a nonlinear transform to either or both of the variables and fit a straight line to the transformed data. A second option is to fit a higher order polynomial. In this section, we examine a third option, which is to use a neural network to derive the relationship.

**Multi-layer feedforward neural network**

Neural networks have been used for some years in geophysics (McCormack, 1991; Schultz et al., 1994; Schuelke et al., 1997). Liu and Liu (1998) describe the use of a multi-layer feedforward neural network (MLFN) to predict log properties directly from seismic data. The MLFN is the traditional network shown in Figure 7.

The properties of the MLFN are described in numerous textbooks (e.g., Masters, 1994). The network consists of an input layer, an output layer, and one or more hidden layers. Each layer consists of nodes, and the nodes are connected with weights. The weights determine the result from the output layer. In our implementation, the input layer has as many input nodes as there are attributes. If a convolutional operator is used, the number of effective attributes is increased by the operator length. For example, for an operator length of 3, each attribute is repeated three times, corresponding to a time sample shift of −1, 0, and +1. The output layer has one node, since we are predicting a single log property. We use a single hidden layer, with the number of nodes set by experimentation.

The training process consists of finding the optimum weights between the nodes. The training is performed by presenting training examples to the network. Each example consists of data for a single time sample \( \{A_1, A_2, A_3, L\} \), where \( A_i \) are the attributes and \( L \) is the measured target log value. There are as many training examples as there are cumulative seismic samples within the analysis windows from all the wells available.

The problem of estimating the weights can be considered a nonlinear optimization problem, where the objective is to minimize the mean-squared error between the actual target log values and the predicted target log values. This problem has traditionally been solved by backpropagation, which is a form of gradient descent. Modern methods now use conjugate-gradient and simulated annealing to speed convergence and avoid local minima (Masters, 1994).

As an example of the behavior of the MLFN, Figure 8 shows the prediction curve for the same data as Figure 6 using the MLFN with five nodes in the hidden layer. In this case, since there is only one attribute, there is a single node in the input layer.

Figure 8 demonstrates two aspects of the behavior of the MLFN. The positive aspect is that the data values over most of the attribute range are modeled more accurately than is the case with linear regression. The negative aspect is the instability apparent at the low attribute values as the network attempts to model the data too closely. This is an example of a condition known as overtraining, which will be discussed in more detail later.

**Probabilistic neural network**

An alternative type of neural network is the probabilistic neural network (Masters, 1995; Specht, 1990, 1991). The probabilistic neural network (PNN) is actually a mathematical interpolation scheme which happens to use a neural network...
values

We can define the total prediction error for the training data as the sum of the prediction errors for each training sample. Repeating this process for each of the training samples, we can calculate the prediction error for that sample when that sample is left out of the training data. Since we know the value of this sample, we can calculate the prediction error for that sample.

The new log value is estimated as the sum of the predicted values for each attribute, weighted by the quantity \( \sigma_j \), which may differ for each of the attributes.

Equations (14) and (15) describe the application of the PNN to new data not in the training set. This is sometimes called overtraining and is described by Kalkomey (1997). Effectively, using higher numbers of attributes is analogous to fitting a crossplot with increasingly higher order polynomials.

Note that the prediction error depends on the choice of the parameters \( \sigma_j \). This quantity is minimized using a nonlinear conjugate gradient algorithm described in Masters (1995). The resulting network has the property that the validation error is minimized.

The performance of the PNN on the simple crossplot data is shown in Figure 9. From this figure, we can see that the PNN has the desirable characteristic of following the data as closely as the MLFN, but it does not have the same instability at the limits of the attribute range. The biggest problem with the PNN is that because it carries around all its training data and compares each output sample with each training sample, the application time can be slow.

VALIDATION

In this section, we examine the question of how to determine the correct number of attributes to use. As discussed previously, we can show that a multiattribute transform with \( N + 1 \) attributes must always have a prediction error less than or equal to the transform with \( N \) attributes. As more attributes are added, we can expect an asymptotically declining prediction error, as shown in Figure 10.

Of course, while the additional attributes always improve the fit to the training data, they may be useless or worse when applied to new data not in the training set. This is sometimes called overtraining and is described by Kalkomey (1997). Effectively, using higher numbers of attributes is analogous to fitting a crossplot with increasingly higher order polynomials.

FIG. 9. Prediction curve derived by the PNN. The data are the same as for Figure 6.

FIG. 10. Plot of prediction error against number of attributes used in the transform. Mathematically, the curve must continue to decline asymptotically.
A number of statistical techniques have been derived to measure the reliability of the higher order attribute fits (e.g., Draper and Smith, 1966). Unfortunately, most of these techniques apply to linear regression and are not immediately applicable to nonlinear prediction using neural networks. For this reason, we have chosen cross-validation, which can be applied to any type of prediction.

Cross-validation consists of dividing the entire training data set into two subsets: the training data set and the validation data set. The training data set is used to derive the transform, while the validation data set is used to measure its final prediction error. The assumption is that overtraining on the training data set will result in a poorer fit to the validation data set. This is illustrated in Figure 11.

In our analysis, the natural subdivision of data is by well. In other words, the training data set consists of training samples from all wells, except some specified hidden well. The validation data set consists of samples from that hidden well. In the process of crossvalidation, the analysis is repeated as many times as there are wells, each time leaving out a different well. The total validation error is the rms average of the individual errors:

$$E_V^2 = \frac{1}{N} \sum_{i=1}^{N} e_{V_i}^2,$$

(18)

where $E_V$ is the total validation error, $e_{V_i}$ is the validation error for well $i$, and $N$ is the number of wells in the analysis.

Figure 12 shows the same plot as Figure 10, except that the total validation error has been added. As expected, the validation error for any particular number of attributes is always higher than the training error. This is because removing a well from the training set will always result in a decrease in predictive power. Also note that the validation error curve does not decline monotonically. In fact, it exhibits a broad local minimum around four attributes and then gradually increases. We interpret this to mean that all additional attributes after the fourth are overtraining the system. Generally, if a validation error curve exhibits a distinct minimum, we assume that the number of attributes at that point is optimum. If the validation error curve shows a broad minimum, such as Figure 12, or shows a series of local minima, we select the point at which the curve stops declining convincingly. This would correspond to the first two attributes in Figure 12.

Establishing the statistical significance of a prediction has been addressed by many authors in the past. Specht (1991) points out that the PNN provides a consistent estimator, asymptotically (i.e., given enough training samples) converging to the underlying joint probability density function of the attributes and prediction variables. Leonard et al. (1992) show how crossvalidation (the validation error) can be used to calculate confidence intervals for the predictions. Leonard shows for a well that the confidence interval is directly proportional to the well validation error and inversely proportional to the square root of the number of wells.

The extrapolation uncertainty must also be addressed, i.e., the behavior of the prediction between and away from the training wells. Leonard suggests an approach to determine if there is sufficient training data by estimating local training data density in the neighborhood of the input sample to be used for the prediction. We plan on implementing a variation of this approach in future research. Currently, extrapolation accuracy is assessed by visual observation of the seismic predictions between wells and comparison with the actual seismic data.

**EXAMPLE 1**

The first example comes from the Blackfoot area of western Canada. The data have been supplied by the University of Calgary Crewes Consortium and consist of a 3-D seismic volume which ties 13 wells. The primary target is the Glauconitic member of the Mannville Group. The reservoir occurs at a depth of around 1550 m (1060 ms), where Glauconitic sand and shale fill valleys incised into the regional Mannville stratigraphy. The objectives of the survey are to delineate the channel and to distinguish between sand fill and shale fill. Each well contains a porosity log, which is the target for this example. The seismic volume has been processed through a model-based inversion algorithm to produce an acoustic impedance volume.
This volume is used as an attribute in the process. Figure 13 shows a single in-line from each of the seismic volumes.

For each of the 13 wells, a single composite trace has been extracted from the 3-D volumes by averaging the nine nearest traces around the borehole. The training data for one of the wells is shown in Figure 14.

Note that the porosity log has been converted from depth to time and sampled at the same 2-ms sample rate as the seismic data. Because we will be correlating over a time window, the quality of the depth to time conversion is critical for this process. The analysis window is indicated by the horizontal red lines and is <100 ms. Figure 15 shows a crossplot of the target porosity values against the seismic inversion attribute, using points from the analysis windows of all 13 wells. The normalized correlation coefficient is 0.41, indicating a fairly poor correlation using this attribute alone.

The attributes available for the multiattribute analysis consist of 17 attributes derived from the seismic trace plus the external attribute seismic inversion. The complete list is noted in Table 1, column 1. The list is further increased by applying...
the indicated nonlinear transforms to each of the previous attributes:

The analysis consists of applying the stepwise regression described earlier. A seven-point convolutional operator was used. The tabular results are shown in Table 2. Each row shows a multiattribute transform with an increasing number of attributes. The first row, for example, shows that the single best attribute is 1/(seismic inversion). Using this attribute with a seven-point convolutional operator gives a prediction error of 5.5%. (This error is the absolute prediction error in the units of porosity, which is percent). The second line shows that the best pair of attributes is 1/(seismic inversion) and integrate. Using this pair gives a prediction error of 5.08%. Table 2 shows combinations up to 10 attributes.

The same information is displayed graphically in Figure 16, which also shows the validation results. The lower curve is the prediction error when all wells are used in the analysis. As expected, this curve decreases as attributes are added. The upper curve shows the average validation error, as defined above. We interpret this curve to mean that adding attributes after the sixth overtrains the system.

Figure 17 shows the result of applying the derived multiattribute transform with six attributes. In this figure, the original porosity logs are shown in black and the predicted logs are in red. The normalized correlation for all the wells is now 0.69, compared with the crossplot correlation of 0.41, derived from a single attribute alone.

Figure 18 shows a similar plot, but in this case the predicted log for each well has been calculated with a different transform. For each well, the same six attributes were used, but the weights were recalculated, using reduced training data that did not include the target log. Effectively, this simulates the result of drilling the well after analysis.

Figure 19 shows the distribution of prediction errors over the 13 wells. While the distribution is highly variable, the difference between the two curves is fairly consistent at about 0.4% in porosity units.

Using the same six attributes, the PNN was trained as outlined in the previous section. Figure 20 shows the prediction results for the PNN. Because the PNN contains a copy of all

Table 2. The results of stepwise regression, applied to the porosity prediction problem. Each line shows a different multiattribute transform with the number of attributes listed in the first column. The multiattribute transform for each line includes all attributes above it. The prediction error for that transform is shown in the last column in the units of the target log (i.e., percent porosity).

<table>
<thead>
<tr>
<th>Number of attributes</th>
<th>Target</th>
<th>Final attribute</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>porosity</td>
<td>1/(seismic inversion)</td>
<td>5.04568</td>
</tr>
<tr>
<td>2</td>
<td>porosity</td>
<td>Integrate</td>
<td>5.08223</td>
</tr>
<tr>
<td>3</td>
<td>porosity</td>
<td>Derivative instantaneous amplitude</td>
<td>4.941341</td>
</tr>
<tr>
<td>4</td>
<td>porosity</td>
<td>Derivative</td>
<td>4.82724</td>
</tr>
<tr>
<td>5</td>
<td>porosity</td>
<td>Integrated absolute amplitude</td>
<td>4.728743</td>
</tr>
<tr>
<td>6</td>
<td>porosity</td>
<td>Average frequency</td>
<td>4.62451</td>
</tr>
<tr>
<td>7</td>
<td>porosity</td>
<td>Cosine instantaneous phase</td>
<td>4.568509</td>
</tr>
<tr>
<td>8</td>
<td>porosity</td>
<td>Instantaneous frequency</td>
<td>4.522447</td>
</tr>
<tr>
<td>9</td>
<td>porosity</td>
<td>Amplitude-weighted frequency</td>
<td>4.473168</td>
</tr>
<tr>
<td>10</td>
<td>porosity</td>
<td>Amplitude-weighted cosine phase</td>
<td>4.432727</td>
</tr>
</tbody>
</table>

FIG. 15. Crossplot of porosity against acoustic impedance from the seismic inversion, using points within the analysis windows from all 13 wells. The normalized correlation is 0.41.

FIG. 16. The results of stepwise regression in graphical form. The lower curve (black) shows the prediction error when all wells are used in the analysis. The upper curve (red) shows the validation error.
the target data within its operator, prediction results are always higher than is the case with linear regression. Mathematically, this is analogous to kriging, in which derived maps always honor the input well information. As with the kriging technique (Deutsch and Journel, 1992), the real measure of performance is crossvalidation, and this is shown in Figure 21. The average normalized correlation for all of the wells is 0.62. While this is only marginally better than the validation result for multivariate linear regression (0.60), the high-frequency enhancement for thin layers can be observed at specific locations (e.g., between 1050 and 1075 ms).

Ultimately the value of multiattribute analysis must be measured in terms of its improvement over the basic crossplotting of a single attribute. Figure 22 shows the continuous improvement in prediction ability as we move from a crossplotting using the single best attribute (acoustic impedance) to multivariate linear regression with the seven-point operator and six attributes to the PNN. In each case, we are showing the validation result, which is the expected performance on this well in a blind test. Note in particular the enhancement of the thin-bed resolution.

Each of the three derived transforms was then applied to the 3-D seismic and inversion volumes. The result in each case is a volume of estimated porosity. Figure 23 shows a single in-line through each of the volumes. The anomaly at about 1090 ms is a known sand channel. Note the high-resolution result achieved with the PNN.

Finally, Figure 24 shows a data slice through the PNN volume. The data slice tracks 12 ms below the high-porosity marker at 1070 ms and averages the estimated porosity for the next 6 ms. We can see clearly the high-porosity channel trending north-south through the center of the volume.

EXAMPLE 2

The second example comes from the Pegasus field in West Texas. The Pegasus field is located 25 miles south of Midland, Texas, in the southwestern portion of what is now the Midland basin. Hydrocarbons are structurally trapped at Pegasus in a northeast-southwest-trending faulted anticline, 7 miles long (north-south) by 4 miles wide (east-west). The Devonian reservoir, at a depth of 11 500 ft, is one of six producing intervals that range in depth from 5500 to 13 000 ft. A more detailed description of the field, the geology, and the reservoir is given by Schuelke et al. (1997). The purpose of this example is to show the improvement in resolution both vertically and horizontally using the PNN neural network versus the multivariate linear regression.

The same validation procedure followed in example 1 was used to determine which and how many seismic attributes to use in this porosity prediction exercise (see Schuelke et al., 1997). Ten wells provided the calibration between the well porosity and the seismic attributes. The target region was restricted to the Devonian interval approximately 100 ms (500 ft) thick. The case study area was a small 2 × 3-mile subset of the full 3-D survey area. Figure 25 shows a seismic in-line from the 3-D survey that intersects one of the ten calibration wells. The red curve is the porosity log for the Peg 21-06 well. Deflection to the right shows increasing porosity. The top Devonian is at 1710 ms, and the base Devonian is at 1820 ms for this well location. There is one very thin high-porosity (24%) zone in this well at approximately 1765 ms. The intervals above and below this high-porosity interval in the Devonian are tight, <5% porosity. The intervals above and below the Devonian interval are shales. The porosity log shows false porosity in these zones.

Figures 26 and 27 show the porosity predictions from the seismic attributes for the same seismic line. Figure 26 is the multivariate linear regression prediction, and Figure 27 is the prediction using the PNN neural network. The color scale is in porosity units, with the higher porosity colored light green and
yellow. The tight to very low-porosity values are gray. Both predictions show the porosity development within the mid-Devonian interval. The multivariate linear regression results, however, show a smoothed, or more averaged, prediction. The PNN neural network prediction retains more of the dynamic range and high-frequency porosity content, as exhibited by the porosity logs at the well locations. The highs and lows of porosity are retained as well as the time resolution. This is to be expected because the neural network result is a nonlinear function that more closely follows the training or control data from the wells, while the linear regression approach provides only an average fit to the control data. Away from the well control the PNN results show the lateral porosity variability expected in this stratigraphic controlled reservoir. Because the network has been trained over a relatively small time window and because the attributes are calculated from band-limited seismic data, we do not expect the general trend or low-frequency component of the predicted porosity to be reliable. To some extent, this trend has been provided by the seismic inversion attribute, but the trend in that attribute is itself derived from the initial model for the inversion.

The benefits of this improved vertical and lateral resolution are evident on a time-slice view through the two porosity volumes. Figure 28 is a time slice through the multivariate linear regression result at 1760 ms. The maximum porosity value from a 10-ms window centered at this time is displayed. The porosity color coding is the same as for the in-line displays in Figures 26 and 27. The multivariate linear regression results show the general outline of the higher porosity zone for this time slice through the reservoir interval. However, much of the lateral variability in porosity and the higher values of porosity are missing. Figure 29 is the time slice through the PNN neural network prediction at the same time interval. The PNN result shows the lateral variability of porosity better and matches the extremes of porosity, as indicated by the log data. This degree of resolution is required in estimating flow barriers or designing a horizontal drilling program. Indeed, possible porosity barriers can be seen in Figure 29 as low-porosity zones (gray) between the high-porosity zones (green).

CONCLUSIONS

We have demonstrated the use of multiple seismic attributes for predicting well-log properties. In our analysis, seismic attributes are defined as any sample-based transform of the seismic data. Two mathematical formulations have been used: multivariate linear regression and neural network prediction. For each of these cases, the selection of appropriate attributes has been determined by forward stepwise regression, which builds groups of attributes sequentially. We have introduced a modification of conventional regression which includes a

Fig. 19. The prediction errors for each of the 13 wells. The lower curve shows the prediction error when the specified well is used in the analysis. The upper curve shows the validation error when the well is not used in the analysis.

Fig. 20. Applying the PNN using six attributes and a seven-point operator. Only the first three wells are shown. The original porosity log is shown in black; the predicted log is shown in red. The normalized correlation coefficient for all the wells is 0.95.

Fig. 21. The validation result for the PNN. This is the same as Figure 20 except that the PNN for each well has been rederived with the well data removed from the analysis. The normalized correlation for all of the wells is 0.62.
convolutional operator applied to each of the attributes. This operator is assumed to be time invariant; hence, the process is applied to a targeted time window. For any derived multiattribute transform, the measure of performance has been cross-validation, which systematically removes wells from the analysis and measures the prediction error for these wells.

We have described two types of neural network as applied to this problem: the MLFN and the PNN. Each of these networks uses the same attributes derived by the multivariate linear regression analysis. In each case, we expect an increase in resolution attributable to the nonlinear behavior of the network.

We have demonstrated this methodology on two data sets. In each case, we have seen an increase in predictive power and resolution as we progress from conventional crossplotting to multivariate linear regression to neural network. This improvement is evident on the training data and is supported by the validation data as well.

In summary, the methodology can be thought of as an extension of conventional seismic inversion. Both processes deal with the same input data (seismic and logs), and both attempt to predict a log property. The main advantages of the new algorithm over conventional inversion are

1) it predicts other logs besides acoustic impedance (e.g., porosity),
2) it may use other attributes besides the conventional stack for this purpose,
3) it does not rely on any particular forward model,
4) it can achieve greatly enhanced resolution,
5) it does not require a knowledge of the seismic wavelet, and
6) it uses cross-validation as a measure of success.

However, these advantages are gained only if there is sufficient well control. This means not only a large enough number of wells but also a distribution of log data which span the range of expected conditions in the earth. Our current research is aimed at quantifying how well these conditions are satisfied in practical exploration cases.

ACKNOWLEDGMENTS

The authors are grateful for the help of research geophysicist Todor Todorov in preparing and analyzing the data in example 1, in addition to his continuous support during the research project. We also thank Brian Russell for providing insight into the mathematical operation of both multivariate analysis and the PNN. Finally, we are grateful to the members of the University of Calgary Crewes Consortium and to Mobil Exploration & Producing U.S., Midland, Texas, for allowing us to publish the results derived on their data.

REFERENCES


Masters, T., 1994, Signal and image processing with neural networks: John Wiley & Sons, Inc.


FIG. 23. Application of the derived transforms to the 3-D volumes. The upper panel shows the regression curve applied to the acoustic impedance volume. The middle panel shows the multivariate linear transform with six attributes and a seven-point operator. The lower panel shows the PNN with six attributes and a seven-point operator. The inserted log is the target porosity log at this location. The color scale is in the units of percent porosity.
FIG. 24. A data slice through the porosity volume estimated using the PNN. The color scale is in the units of percent porosity.

FIG. 25. An example seismic line from the 3-D survey through the Peg 21-06 well. The porosity log is shown as the red curve, with increasing porosity to the right.
FIG. 26. The multivariate linear regression result. Porosity is shown in color. The log porosity for the Peg 21-06 well is displayed as a black curve. The thin high-porosity zone in the middle of the Devonian is correctly identified, but the temporal resolution is less on the prediction and the absolute value is less than for the actual porosity log.

FIG. 27. The PNN neural network result. Porosity is shown in color. The log porosity for the Peg 21-06 well is displayed as a black curve. The thin high-porosity zone in the middle of the Devonian is correctly identified, and both the thickness and the absolute value of the prediction matches the log porosity. Away from the well control the predictions show the lateral variability expected in this reservoir.
APPENDIX

MULTIATTRIBUTE LINEAR REGRESSION

Multiattribute linear regression is an extension of simple linear regression to $M$ variables. That is, we will use $M$ attributes, $A_1, A_2, \ldots, A_M$, to predict the log $L$. To do this, we must determine the $M + 1$ weights, $w_0, w_1, w_2, \ldots, w_M$, which, when multiplied by the particular set of attribute values, give the closest result to the log in a least-squared sense. For simplicity, assume that $M = 3$. If we have $N$ samples in our log, we can then write the following set of equations:

$$
L_1 = w_0 + w_1 A_{11} + w_2 A_{21} + w_3 A_{31}
$$

$$
L_2 = w_0 + w_1 A_{12} + w_2 A_{22} + w_3 A_{32}
$$

$$
L_N = w_0 + w_1 A_{1N} + w_2 A_{2N} + w_3 A_{3N},
$$

where $A_{ij}$ is the $j$th sample of the $i$th attribute. Notice that equations (A-1) can be written as

$$
\begin{bmatrix}
L_1 \\
L_2 \\
L_3 \\
\vdots \\
L_N
\end{bmatrix} =
\begin{bmatrix}
1 & A_{11} & A_{21} & A_{31} \\
1 & A_{12} & A_{22} & A_{32} \\
1 & A_{1N} & A_{2N} & A_{3N}
\end{bmatrix}
\begin{bmatrix}
w_0 \\
w_1 \\
w_2 \\
w_3
\end{bmatrix}
$$

or

$$
L = AW.
$$

where $L$ is an $N \times 1$ matrix containing the known log values, $A$ is an $N \times 4$ matrix containing the attribute values, and $W$ is a $4 \times 1$ matrix with the unknown weights. This can be solved by least-squares minimization to give

$$
W = [A^T A]^{-1} A^T L.
$$

As a detailed computation, note that

$$
\begin{bmatrix}
w_0 \\
w_1 \\
w_2 \\
w_3
\end{bmatrix} =
\begin{bmatrix}
N \sum A_{1i} & \sum A_{1i} A_{2i} & \sum A_{1i} A_{3i} \\
\sum A_{2i} A_{1i} & \sum A_{2i}^2 & \sum A_{2i} A_{3i} \\
\sum A_{3i} A_{1i} & \sum A_{3i} A_{2i} & \sum A_{3i}^2
\end{bmatrix}^{-1} 
\begin{bmatrix}
N \sum L_i \\
\sum A_{1i} L_i \\
\sum A_{2i} L_i \\
\sum A_{3i} L_i
\end{bmatrix}
$$

or

$$
\begin{bmatrix}
w_0 \\
w_1 \\
w_2 \\
w_3
\end{bmatrix} =
\begin{bmatrix}
N \sum L_i \\
\sum A_{1i} L_i \\
\sum A_{2i} L_i \\
\sum A_{3i} L_i
\end{bmatrix}
\begin{bmatrix}
\sum N \\
\sum A_{1i} \\
\sum A_{2i} \\
\sum A_{3i}
\end{bmatrix}
$$
Multiattribute linear regression using convolutional weights

Next, let us generalize the preceding equations by assuming that we have a convolutional sum

\[ L = w_0 + w_1 A_1 + w_2 A_2 + \cdots + w_N A_N, \quad (A-6) \]

where \( w_0 \) is a constant and \( w_i \) is an \( l \)-point convolutional filter.

To simplify, consider equation (A-6) using only two attributes and four sample values. Also, consider the case of a 3-point operator, which we could write as

\[ w_i = [w_i(-1), w_i(0), w_i(+1)]. \]

Under these circumstances, equation (A-6) can be written in the following matrix form:

\[
\begin{bmatrix}
  L_1 \\
  L_2 \\
  L_3 \\
  L_4
\end{bmatrix} = w_0 + w_1(-1) \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} + w_1(0) \begin{bmatrix} 0 & w_1(-1) & 0 \end{bmatrix} + w_1(1) \begin{bmatrix} w_1(-1) & 0 & 0 \end{bmatrix}
\]

\[
+ w_2(-1) \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} + w_2(0) \begin{bmatrix} 0 & w_2(-1) & 0 \end{bmatrix} + w_2(1) \begin{bmatrix} w_2(-1) & 0 & 0 \end{bmatrix}
\]

Equation (A-8) shows that the effect of adding the three-point operator is exactly the same as increasing the number of attributes by a factor of three, the additional attributes being calculated by shifting the original attributes by \(-1\) and \(+1\) sample. We can now use exactly the same least-squares formulation derived in the previous section. The explicit result for this case is

\[
\begin{bmatrix} w_1(-1) \\
  w_1(0) \\
  w_1(+1)
\end{bmatrix}
\]

\[
= \left[ \begin{array}{cccc}
    A_{12} & A_{13} & A_{14} & 0 \\
    A_{11} & A_{12} & A_{13} & A_{14} \\
    0 & A_{11} & A_{12} & A_{13} \\
    0 & A_{14} & A_{13} & A_{12}
\end{array} \right]^{-1}
\times \left[ \begin{array}{c}
    A_{12} \\
    A_{13} \\
    A_{14} \\
    0
\end{array} \right]
\]

or

\[
\begin{bmatrix} w_1(-1) \\
  w_1(0) \\
  w_1(+1)
\end{bmatrix}
\]

\[
= \left[ \begin{array}{cccc}
    A_{12} & A_{13} & A_{14} & 0 \\
    A_{11} & A_{12} & A_{13} & A_{14} \\
    0 & A_{11} & A_{12} & A_{13} \\
    0 & A_{14} & A_{13} & A_{12}
\end{array} \right]^{-1}
\times \left[ \begin{array}{c}
    4 A_{i1}^2 \\
    \sum_{i=2}^{4} A_{i1} A_{i1+i} \\
    \sum_{i=2}^{4} A_{i1} A_{i1+i} \\
    \sum_{i=2}^{4} A_{i1} A_{i1+i}
\end{array} \right]
\]

(A-9)
Linear Discriminant Analysis

Linear discriminant analysis (LDA) is a statistical tool that allows us to discriminate among different groups, or clusters, of data, in multidimensional space. This technique is often called supervised clustering. The “supervised”, or training, part is done by using the multivariate means and variances within a set of known groups to develop a mathematical relationship that can then be applied to a set of measured, but unclassified, values.

To understand LDA, we will first consider the simplest case of two different sets of observations, \( X \) and \( Y \), that have been classified into two groups, \( A \) and \( B \). For example, \( X \) may represent the amplitude envelope of a seismic sample, and \( Y \) may represent the instantaneous phase of the sample, whereas \( A \) might represent data recorded in a fore-reef environment and \( B \) might represent data recorded in a back-reef environment. In the initial analysis, it is up to the interpreter to make the decision about the group into which each sample falls. Let us assume that we have \( p \) observations in total, which have been subdivided by the interpreter into two groups, \( n \) observations in group \( A \) and \( m \) observations in group \( B \). We could write these two groups as the following sets of ordered pairs of attributes:

\[
A = (x_{A1}, y_{A1}), (x_{A2}, y_{A2}), \ldots, (x_{An}, y_{An}), \quad (1)
\]

\[
B = (x_{B1}, y_{B1}), (x_{B2}, y_{B2}), \ldots, (x_{Bm}, y_{Bm}). \quad (2)
\]

Figure 1, modified from Davis (Statistics and Data Analysis in Geology), shows how the statistics of the two groups may overlap in the original parameter space, where the \( Y \)
measurements have been plotted against the \( X \) measurements. Figure 1 also shows the discriminant function line. The discriminant function is the line that creates the best separation between the two groups when the points are projected onto this line. Notice that this can be thought of as a rotation of the original parameter axes to one involving two new variables that better separate the clusters of points. With this interpretation, discriminant analysis is similar to principal component analysis. However, in discriminant analysis, we do not actually perform the rotation of the axes, but, as mentioned earlier, simply project the points onto the discriminant function line.

For each pair of values, \( x_i \) and \( y_i \), the discriminant function is computed by:

\[
R_i = w_1 x_i + w_2 y_i \quad (3)
\]

where \( w_1 \) and \( w_2 \) are weights to be computed.

\[G]\text{Figure 1 – Linear discriminant analysis for two attributes, X and Y, and two clusters, A and B. (modified from Davis)}\]

The weights are determined by using the grouped data. To facilitate the description of how this is done, consider breaking the two clusters into the following four sets of observations:

\[
X_A = (x_{A1}, x_{A2}, \ldots, x_{An}) = \text{the } n \text{ observations of attribute } X \text{ in group } A, \quad (4)
\]

\[
Y_A = (y_{A1}, y_{A2}, \ldots, y_{An}) = \text{the } n \text{ observations of attribute } Y \text{ in group } A, \quad (6)
\]

\[
X_B = (x_{B1}, x_{B2}, \ldots, x_{Bm}) = \text{the } m \text{ observations of attribute } X \text{ in group } B, \quad (5)
\]

\[
Y_B = (y_{B1}, y_{B2}, \ldots, y_{Bm}) = \text{the } m \text{ observations of attribute } Y \text{ in group } B. \quad (7)
\]
The first step in discriminant analysis is to compute the means of the four different cases, in the following way:

\[ m_{AX} = \frac{1}{n} \sum_{i=1}^{n} x_{Ai} \]  
(8)

\[ m_{AY} = \frac{1}{n} \sum_{i=1}^{n} y_{Ai} \]  
(9)

\[ m_{BX} = \frac{1}{m} \sum_{i=1}^{m} x_{Bi} \]  
(10)

\[ m_{BY} = \frac{1}{m} \sum_{i=1}^{m} y_{Bi} \]  
(11)

We then take the difference between the means of the two groups with respect to each variable, which gives us the mean separation between the two groups along each of the variable axes, as well as the average of the means:

\[ d_x = m_{AX} - m_{BX} \]  
(12)

\[ d_y = m_{AY} - m_{BY} \]  
(13)

\[ m_x = \frac{m_{AX} + m_{BX}}{2} \]  
(14)

\[ m_y = \frac{m_{AY} + m_{BY}}{2} \]  
(15)

All of these definitions are shown in Figure 1.

To compute the weights, we next construct a matrix consisting of the pooled variances and covariances of the two groups. Since the number of observations in the two groups is different, this involves first computing the variances and covariances of the separate groups, and then combining, or pooling, the two results. We can write the unpooled matrices as:

\[ C_A = \begin{bmatrix} C_{AXX} & C_{AXY} \\ C_{AYX} & C_{AYY} \end{bmatrix}, \text{ where:} \]

\[ C_{AXX} = \sum_{i=1}^{n} x_{Ai}^2 - \frac{\left( \sum_{i=1}^{n} x_{Ai} \right)^2}{n} \]

\[ C_{AXY} = C_{AYX} = \sum_{i=1}^{n} x_{Ai} y_{Ai} - \frac{\sum_{i=1}^{n} x_{Ai} \sum_{i=1}^{n} y_{Ai}}{n} \]
\[ C_{AYY} = \sum_{i=1}^{n} y_{Ai}^2 - \left( \sum_{i=1}^{n} y_{Ai} \right)^2 / n \]

and:

\[ C_B = \begin{bmatrix} C_{BXX} & C_{BXY} \\ C_{BYX} & C_{BYY} \end{bmatrix}, \quad (17) \]

Where the terms are created in the same way as in \( C_A \), but B replaces A, and m replaces n. The pooled matrix is simply the normalized sum of \( C_A \) and \( C_B \):

\[ C = \frac{C_A + C_B}{n + m - 2} \quad (18) \]

Notice that the three matrices \( C_A \), \( C_B \) and \( C \) contain information about the width of the clusters (the variances) and their inter-relationships (the covariances), although this is more difficult to interpret visually on Figure 1 than was the difference in the means.

We can now find the weights of discriminant function by a straightforward matrix inversion. The forward equation is:

\[ CW = D \quad (19) \]

Where:

\[ W = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \]

And:

\[ D = \begin{bmatrix} d_x \\ d_y \end{bmatrix} \]

And the solution for \( W \) is:

\[ W = C^{-1}D \quad (20) \]

Recall that for each measured pair of values, or attributes, \( x_i \) and \( y_i \), we can then compute the discriminant function by the formula:

\[ R_i = w_1 x_i + w_2 y_i \quad (21) \]

The problem with the value of \( R_i \) is that we still don’t know whether an attribute pair falls into cluster \( A \) or \( B \). To find out, we simply compute the central value:

\[ R_0 = w_1 m_X + w_2 m_Y \quad (22) \]

Where \( m_1 \) and \( m_2 \) are the average means defined earlier. If \( R_i - R_0 \) is positive, the value falls into cluster \( B \) and, if negative, into cluster \( A \).
From the above mathematical discussion, and a look at Figure 1, it should be obvious what the limitations of this method are. If the clusters are not linearly separable, the method will fail. To show this more clearly, observe the situation shown in Figure 2. In this case, LDA will fail to separate the two clusters, and we will need to use some type of nonlinear training method, such as neural networks.

![Figure 2 - A cross-plot between two attributes showing nonlinear separation of the clusters.](image)

**Using More Attributes and Clusters**

We’ve spent a lot of time looking at the discrimination of two clusters, A and B, using two attributes, X and Y. But what happens as we increase both the number of clusters and the number of attributes? If we continue to use the difference between the means, we soon produce a bewildering array of differences! Thus, it would seem more logical to focus on the means themselves. Indeed, this is the formulation given by Tabachnik and Fidell, in their book entitled *Using Multivariate Statistics*.

In this approach, the authors consider the general case of k clusters ($A_1, A_2, \ldots, A_k$) and p attributes ($X_1, X_2, \ldots, X_p$). For each cluster, the authors develop $p+1$ coefficients so that, for the $j^{th}$ group, a classification score can be determined by summing the weighting coefficients multiplied by the sample attribute values (a zero$^{th}$ coefficient is also derived, hence the $p+1$ terms). The equation for the weights in the $j^{th}$ cluster looks like this:

$$S_j = w_{j0} + w_{j1} X_1 + w_{j2} X_2 + \ldots + w_{jp} X_p,$$  \hspace{1cm} (23)
where: $j = 1, 2, \ldots, k$.
and: $X_i$ = samples from $X_i$ vectors.

To assign a particular value into a cluster, we simply select the one with the largest classification score. The derivation of the weighting terms (other than the first term) is very similar to the derivation of the weighting terms in the previous section except that we use the means themselves rather than the differences of the means. That is:

$$W_j = C^{-1}M_j$$  \hspace{1cm} (24)

where:

$$W_j = \begin{bmatrix} w_{j1} \\ w_{j2} \\ \vdots \\ w_{jp} \end{bmatrix}$$

$$C = \begin{bmatrix} C_{11} & C_{12} & \cdots & C_{1p} \\ C_{21} & C_{22} & \cdots & C_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ C_{p1} & C_{p2} & \cdots & C_{pp} \end{bmatrix} = \text{pooled covariance matrix}$$

$$M_j = \begin{bmatrix} m_{j1} \\ m_{j2} \\ \vdots \\ m_{jp} \end{bmatrix} = j^{th} \text{ cluster means for the } p \text{ attributes}$$

The constant value for each cluster $j$ is then found by:

$$w_{jp} = \left(-\frac{1}{2}\right)W_j M_j = w_{j1}m_{j1} + w_{j2}m_{j2} + \ldots + w_{jp}m_{jp}$$  \hspace{1cm} (25)