Using multi-attribute transforms to predict log properties from seismic data

Article *in* Exploration Geophysics · January 2000

Impact Factor: 0.51 · DOI: 10.1071/EG00481

Daniel P. Hampson
CGG
19 PUBLICATIONS **214** CITATIONS
SEE PROFILE

Brian Russell
The University of Calgary
69 PUBLICATIONS **293** CITATIONS
SEE PROFILE

:

Use of Multi-Attribute Transforms to Predict Log Properties from Seismic Data

Dan Hampson[1], Jim Schuelke[2], John Quirein[3]

Abstract

In this paper, we describe a new method for predicting well log properties from seismic data. The analysis data consists of a series of target logs from wells which tie a 3-D seismic volume. The target logs may be theoretically 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 *multi-attribute transform*, which is a linear or non-linear transform between a subset of the attributes and the target log values. The selected subset is determined by a process of forward step-wise regression, which derives increasingly larger subsets of attributes. An extension of conventional cross-plotting 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, which are derived by least-squares minimization. In the non-linear mode, a neural network is trained, using the selected attributes as inputs. Two types of neural networks have been evaluated: the multi-layer feedforward network (MLFN), and the probabilistic neural network (PNN). Because of its mathematical simplicity, the probabilistic neural network appears to be the network of choice.

To estimate the reliability of the derived multi-attribute transform, cross-validation is used. In this process, each well is systematically removed from the training set, and the transform is re-derived 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 multi-attribute 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.

[1] Hampson-Russell Software Services Ltd, 510-715 5th Ave, SW, Calgary, Canada; E-mail: dan@hampson-russell.com

[2] Formerly Mobil Technology Company, Dallas, Texas, USA; presently ExxonMobil Upstream Research Company, Houston, Texas, USA; E-mail: james_s_schuelke@email.mobil.com

[3] Formerly Mobil Technology Company, Dallas, Texas, USA; presently Halliburton Energy Services, Houston, Texas, USA; E-mail: john.quirein@halliburton.com

Introduction

The integration of well log data and seismic data has been a consistent aim of geoscientists. This has become increasingly important (and successful) in recent years due to the shift from exploration to development of existing fields, with large numbers of wells penetrating them. One type of integration is the forward modeling of synthetic seismic data from the logs. A second type of integration is the 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 et al (1983), Oldenburg et al (1983), Chi et al (1984)).

In this paper, 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 post-stack data itself. This allows us to include pre-stack information, as well as non-linear transforms of the post-stack data. Thirdly, instead of assuming a particular model relating the logs and the seismic, a statistical relationship will be derived by analyzing a set of training data at well locations. This relationship will be either linear (multivariate regression) or non-linear (neural network). Finally, we will use the concept of cross-validation 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.

Multi-Attribute Linear Regression

Seismic Attributes

In this methodology, our aim is to find an operator, possibly non-linear, which 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 non-linear, 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 pre-processing 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. Pre-processing 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, instantaneous frequency, etc., and complicated attributes such as seismic trace inversion and AVO attributes. The transform may or may not incorporate other data sources. Note, for example, that trace inversion assumes other data sources, such as the seismic wavelet, the initial guess model, constraints, etc. However, for this analysis, we still consider the inversion result to be an attribute of the seismic trace.

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

- Horizon-based attributes : These are average properties of the seismic trace between two boundaries, generally defined by picked horizons.

- Sample-based attributes : These are 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 this possibility in this paper.

Taner et al.(1994), has provided a long list of sample-based attributes, of which our experience has shown these to be particularly important for log prediction:

- integrated trace
- integrated absolute amplitude of the trace
- near angle stack
- AVO intercept / gradient
- frequency and absorption estimates
- seismic velocity

Conventional Cross Plotting

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

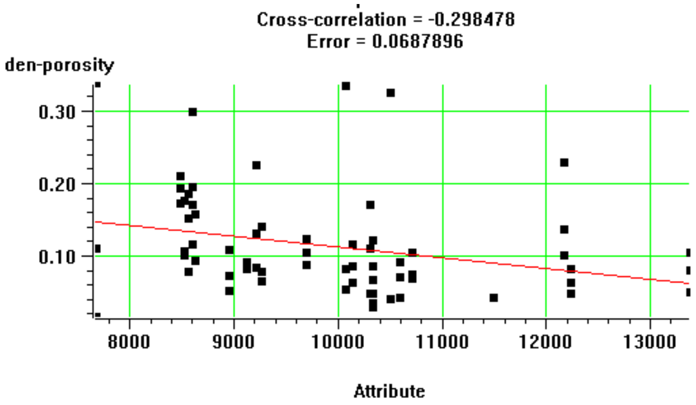


Figure 1: Conventional cross plot between the target log “den-porosity’ and the seismic attribute “Attribute”.

Figure 1 shows an example in which a target log property, in this case "den-porosity", is plotted against a seismic attribute, called "Attribute". The assumption is that the target log has been integrated to travel-time 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 cross plot 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:

$$y = a + bx \tag{1}$$

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

$$E^2 = \frac{1}{N}\sum_{i=1}^{N}(y_i - a - bx_i)^2 \tag{2}$$

where the sum is over all points in the cross plot.

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:

$$\rho = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \tag{3}$$

where:

$$\sigma_{xy} = \frac{1}{N}\sum_{i=1}^{N}(x_i - m_x)(y_i - m_y) \tag{4}$$

$$\sigma_x = \frac{1}{N}\sum_{i=1}^{N}(x_i - m_x)^2 \tag{5}$$

$$\sigma_y = \frac{1}{N}\sum_{i=1}^{N}(y_i - m_y)^2 \tag{6}$$

$$m_x = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{7}$$

$$m_y = \frac{1}{N}\sum_{i=1}^{N} y_i \tag{8}$$

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

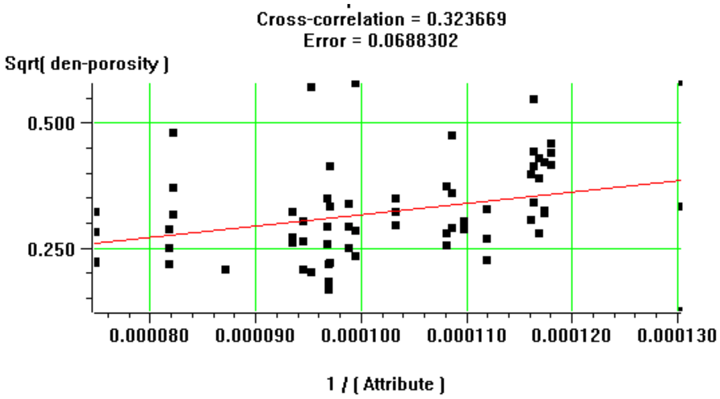


Figure 2: Applying non-linear transforms to both the target and the attribute data improves the correlation between the two.

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:

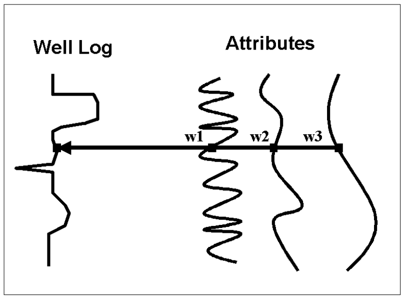


Figure 3: Assuming the case of three seismic attributes, each target log sample is modeled as a linear combination of attribute samples at the same time.

At each time sample, the target log is modeled by the linear equation:

$$L(t) = w_0+w_1A_1\ (t)+w_2\ A_2\ (t)+w_3\ A_3\ (t) \tag{9}$$

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

$$E^2 = \frac{1}{N}\sum_{i=1}^{N}(L_i - w_0 - w_1A_{1i} - w_2A_{2i} - w_3A_{3i})^2 \tag{10}$$

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

$$\begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} N & \sum A_{1i} & \sum A_{2i} & \sum A_{3i} \\ \sum A_{1i} & \sum A_{1i}^2 & \sum A_{1i}A_{2i} & \sum A_{1i}A_{3i} \\ \sum A_{2i} & \sum A_{1i}A_{2i} & \sum A_{2i}^2 & \sum A_{2i}A_{3i} \\ \sum A_{3i} & \sum A_{1i}A_{3i} & \sum A_{2i}A_{3i} & \sum A_{3i}^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum L_i \\ \sum A_{1i}L_i \\ \sum A_{2i}L_i \\ \sum A_{3i}L_i \end{bmatrix} \tag{11}$$

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 (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 multi-attribute regression assumes a single weight for each attribute. The problem with this approach is illustrated in Figure (4):

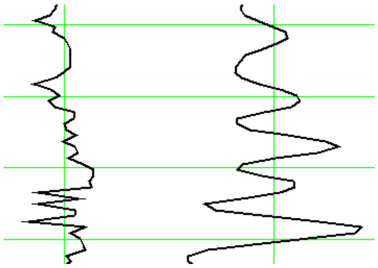


Figure 4: A comparison between the target log on the left with the seismic attribute on the right emphasizes the difference in frequency content. This observation suggests the use of a convolutional operator to resolve the difference.

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:

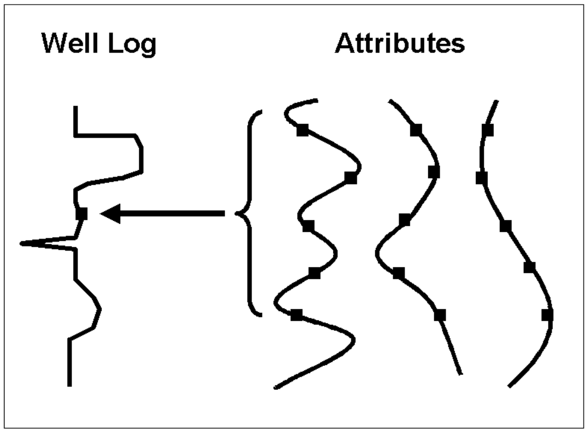


Figure 5: Using a five-point convolutional operator to relate the seismic attributes to the target log.

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:

$$L = w_0 + w_1 * A_1 + w_2 * A_2 + w_3 * A_3 \qquad (12)$$

where * represents convolution, and the 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:

$$E^2 = \frac{1}{N} \sum_{i=1}^{N} (L_i - w_0 - w_1 * A_{1i} - w_2 * A_{2i} - w_3 * A_{3i})^2 \qquad (13)$$

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 Step-Wise Regression

In the previous sections, we have derived equations which 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*. Let's assume, for example, that we wish 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 = 6,375,600 combinations of 5 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 *Step-Wise 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 re-derived. The process is illustrated in the series of steps:

Step 1: Find the single best attribute by *Exhaustive Search*. For each attribute in the list,

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. Call this *attribute*$_1$.

Step 2: Find the best pair of attributes, assuming that the first member is *attribute*$_1$. For each other attribute in the list, form all pairs,

(attribute$_1$*, Amplitude Weighted Phase),*
(attribute$_1$*, 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 *attribute$_2$*.

Step 3: Find the best triplet of attributes, assuming that the first two members are *attribute$_1$* and *attribute$_2$* . For each other attribute in the list, form all triplets,

(attribute$_1$, attribute$_2$, Amplitude Weighted Phase),
(attribute$_1$, attribute$_2$, 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 *attribute$_3$*.

Carry on this process as long as desired.

The first thing to note is that the computation time for this process will be much shorter than for *Exhaustive Search.* For the example above, the number of combinations to check is now 25+24+…+21 = 115, instead of 6,375,600. 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 *Step-Wise Regression* is that we cannot be sure of deriving the optimal solution. In other words, the combination of five attributes found may not, in fact, be the best five which would be found by *Exhaustive Search.* However, it can be shown that 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 *Step-Wise 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 *Step-Wise 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 *Step-Wise 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 the other attribute, A_j, is precisely zero. In summary, because we are using *Step-Wise Regression*, we may 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 *Multi-Attribute Transform*:

> A *Multi-Attribute Transform* is 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 non-linear 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 which this imposes can be understood by examining Figure 6:

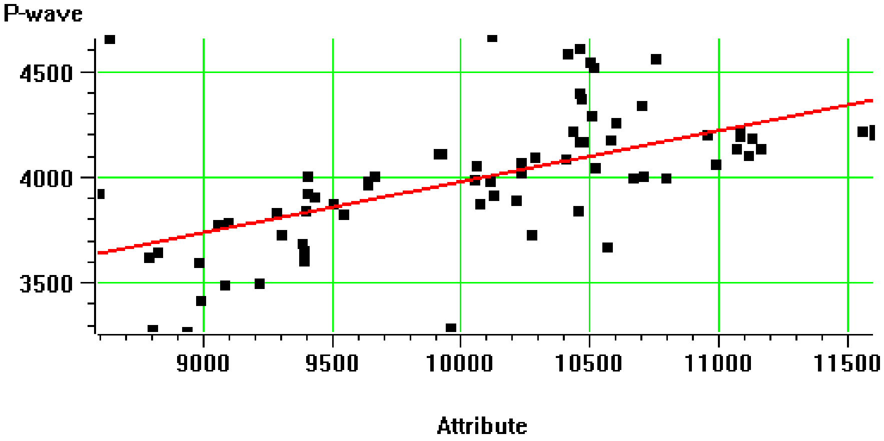


Figure 6: Cross plot of target log against seismic attribute. The regression line has been fit by minimizing Equation 2.

This figure shows a target log, called “P-wave”, cross plotted 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, which has been discussed above, is to apply a non-linear transform to either or both 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, M.D., 1991; Schultz, et al., 1994; Schuelke, Quirein, and Sarg, 1997). Recently Liu (1998) described 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:

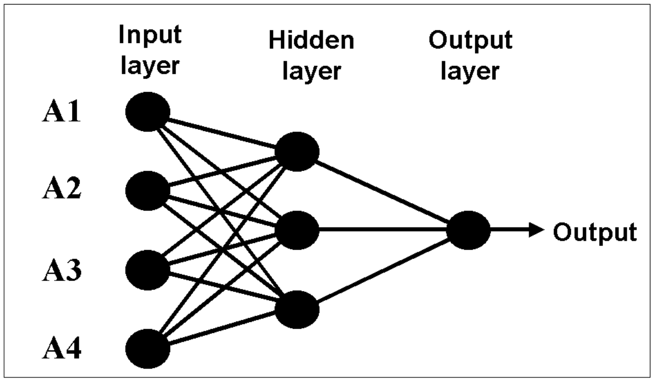


Figure 7: Multi-Layer Feedforward Neural Network architecture

The properties of the MLFN have been described in numerous textbooks (eg, 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 being used, the number of effective attributes is increased by the operator length. For example, for an operator length of 3, each attribute is repeated 3 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 non-linear 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 back-propagation, 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 MLFN, Figure 8 shows the prediction curve for the same data as Figure 6 using the MLFN with 5 nodes in the hidden layer. In this case, since there is only one attribute, there is a single node in the input layer.

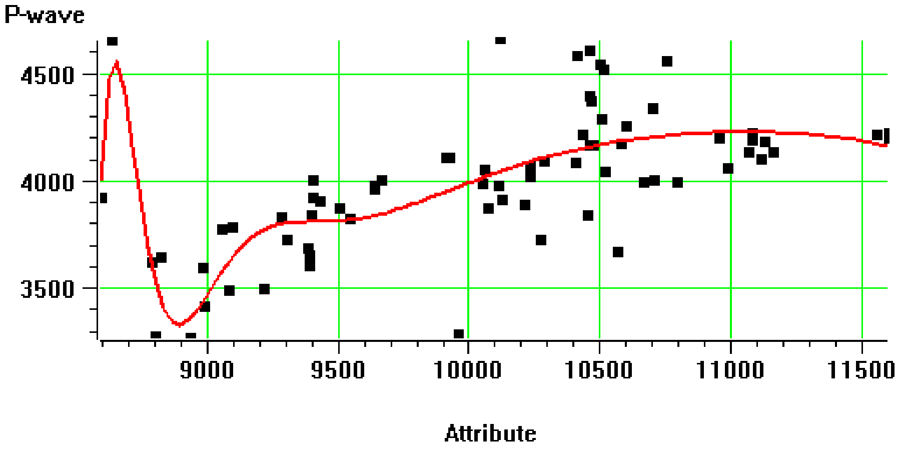


Figure 8: Prediction curve derived by Multi-Layer Feedforward Neural Network with 5 nodes in the hidden layer. The data is the same as that for Figure 6.

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 this data too closely. This is an example of a condition known as "over-training", which we will discuss in more detail below.

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 architecture for its implementation. This is a potential advantage, since by studying the mathematical formulation, we can often understand its behavior much better than the MLFN (which tends to be a black box).

The data used by the PNN is the same training data used by MLFN. It consists of a series of training "examples", one for each seismic sample in the analysis windows from all the wells.

$$\{ A_{11},\ A_{21},\ A_{31},\ L_1\}$$

$\{ A_{12},\ A_{22},\ A_{32},\ L_2 \}$
$\{ A_{13},\ A_{23},\ A_{33},\ L_3 \}$
.
.
.
$\{ A_{1n},\ A_{2n},\ A_{3n},\ L_n \}$

where there are *n* training examples and three attributes. The values L_i are the measured target log values for each of the examples.

Given the training data, the PNN assumes that each new output log value can be written as a linear combination of the log values in the training data. For a new data sample with attribute values:

$x = \{A_{1j},\ A_{2j},\ A_{3j}\}$

the new log value is estimated as:

$$\hat{L}(x) = \frac{\sum_{i=1}^{n} L_i \exp(-D(x, x_i))}{\sum_{i=1}^{n} \exp(-D(x, x_i))} \qquad (14)$$

where

$$D(x, x_i) = \sum_{j=1}^{3} \left(\frac{x_j - x_{ij}}{\sigma_j} \right)^2 \qquad (15)$$

The quantity $D(x,x_i)$ is the “distance” between the input point and each of the training points x_i. This distance is measured in the multi-dimensional space spanned by the attributes, and is scaled by the quantity σ_j, which may be different for each of the attributes.

Equations 14 and 15 describe the application of the PNN network. The training of the network consists of determining the optimal set of smoothing parameters, σ_j. The criterion for determining these parameters is that the resulting network should have the lowest validation error.

Define the validation result for the mth target sample as:

$$\hat{L}_m(x_m) = \frac{\sum_{i \neq m} L_i \exp(-D(x_m, x_i))}{\sum_{i \neq m}^{n} \exp(-D(x_m, x_i))} \tag{16}$$

This is the predicted value of the mth target sample when that sample is left out of the training data. Since we know the value of this sample, we can calculated the prediction error for that sample. Repeating this process for each of the training samples, we can define the total prediction error for the training data as:

$$E_V(\sigma_1, \sigma_2, \sigma_3) = \sum_{i=1}^{n} (L_i - \hat{L}_i)^2 \tag{17}$$

Note that the prediction error depends on the choice of the parameters, σ_j. This quantity is minimized using a non-linear 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 cross plot data is shown in Figure 9:

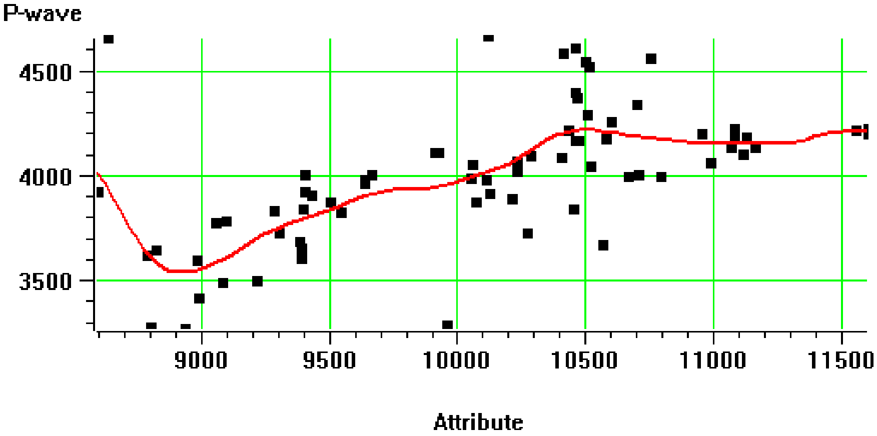


Figure 9: Prediction curve derived by the Probabilistic Neural Network. The data is the same as that for Figure 6.

From this figure, we can see that the PNN has the desirable characteristic of following the data as closely as the MLFN, but 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 multi-attribute 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:

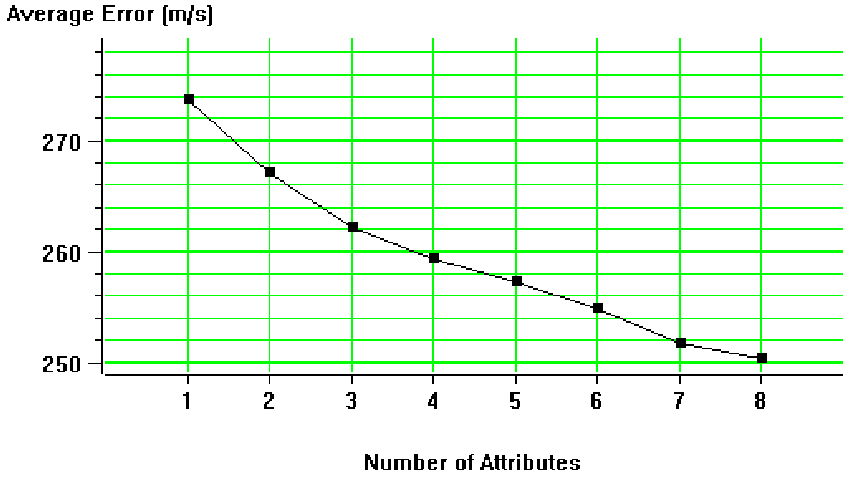


Figure 10: Plot of prediction error against number of attributes used in the transform. Mathematically, the curve must continue to decline asymptotically.

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 "over-training" and has been very well described by Kalkomey (1997). Effectively, using higher numbers of attributes is analogous to fitting a cross plot with increasingly higher order polynomials.

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 non-linear prediction using neural networks. For this reason, we have chosen the process of *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 over-training on the *Training Data Set* will result in a poorer fit to the *Validation Data Set*. This is illustrated in Figure 11:

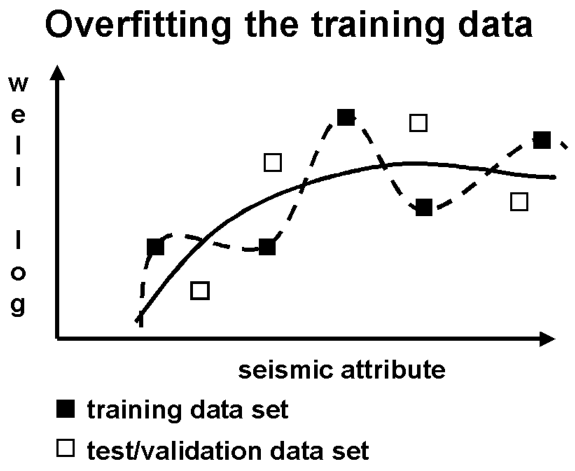


Figure 11: Illustration of cross-validation. Two curves are used to fit the data points. The solid curve is a low order polynomial. The dashed curve is a higher order polynomial. The dashed curve fits the training data set better, but shows a poorer fit when compared with the validation data set.

In our analysis, the natural sub-division 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 *Cross-Validation*, the analysis is repeated as many times as there are wells, each time leaving out a different well. The total validation error is the root mean square average of the individual errors:

$$E_V{}^2 = \frac{1}{N}\sum_{i=1}^{N} e_{Vi}{}^2 \qquad (18)$$

where: E_V is the Total Validation Error,

e_{Vi} 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 now 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 over-training 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.

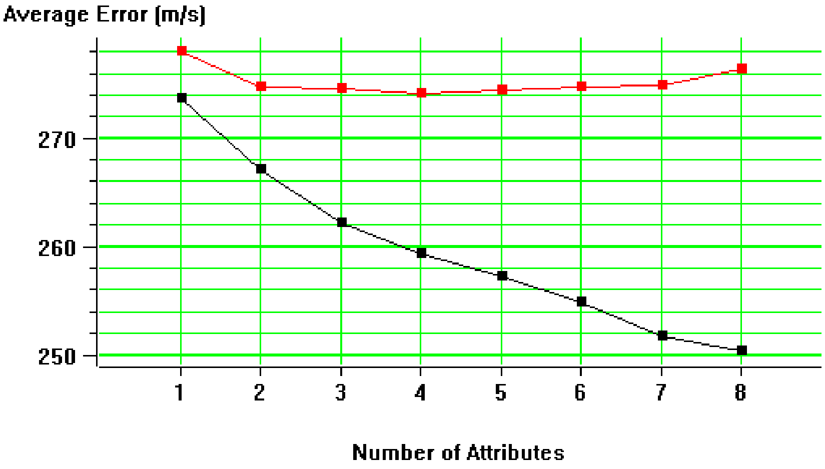


Figure 12: The same plot as Figure 10, except that the Total Validation Error is now shown as the upper curve. Note that attributes past the second contribute little improvement to the Validation Error, and, in fact, gradually cause an increase in prediction error.

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 cross-validation (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. This data has been supplied by the University of Calgary Crewes Consortium, and consists 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 1550m (1060ms), where Glauconitic sand and shale fill valleys incised into the regional Mannville stratigraphy. The objectives of the survey are to delineate the channel and distinguish between sand-fill and shale-fill. Each of the wells contains a porosity log, which is used as 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 inline from each of the seismic volumes.

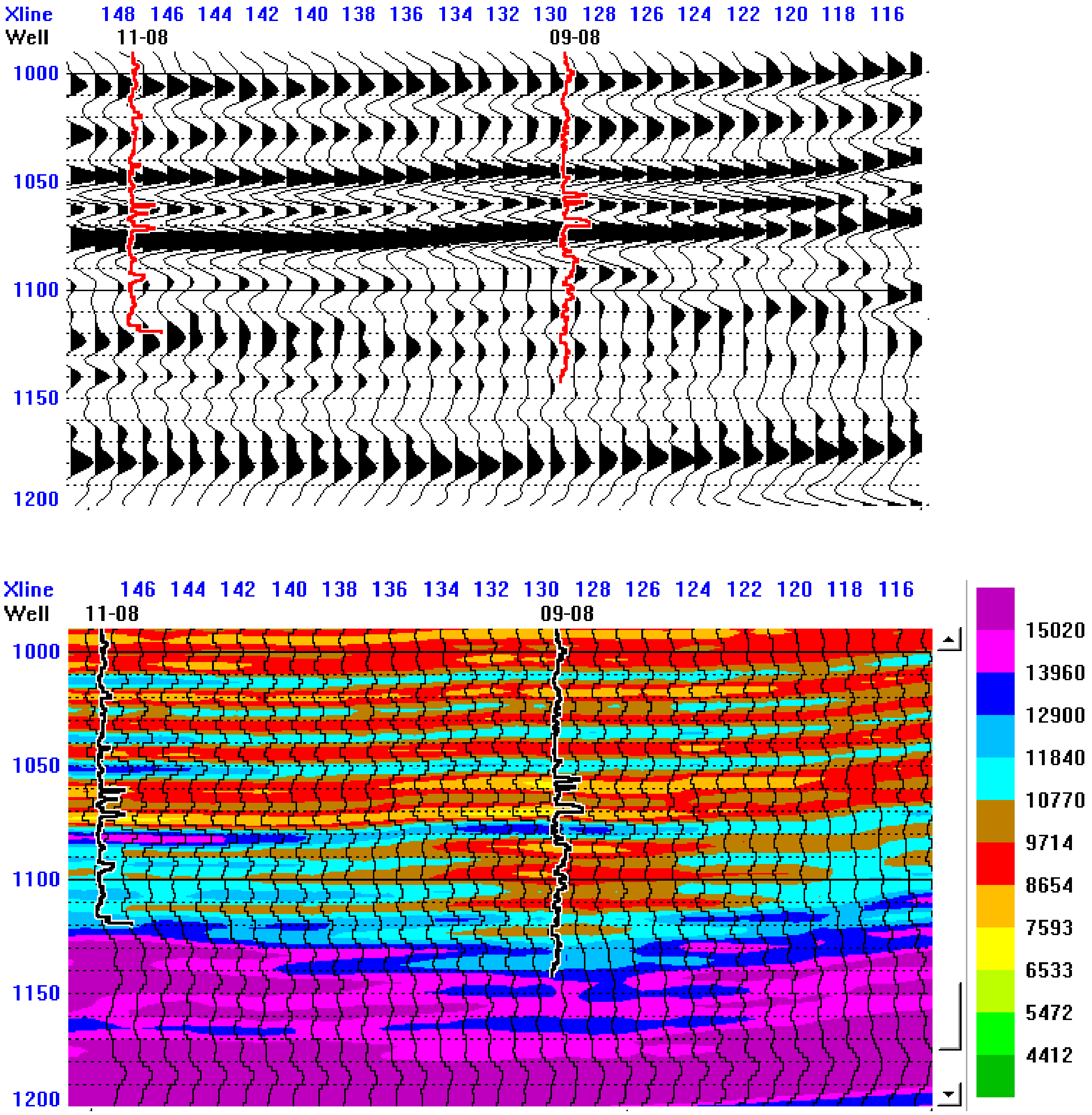


Figure 13: A single inline from the input 3-D volumes. The upper plot shows the post-stack seismic data. The lower plot shows the seismic inversion result, which is used as an attribute for this analysis. The color scale is acoustic impedance.

For each of the 13 wells, a single composite trace has been extracted from the 3-D volumes by averaging the 9 nearest traces around the borehole. The training data for one of the wells is shown in Figure 14.

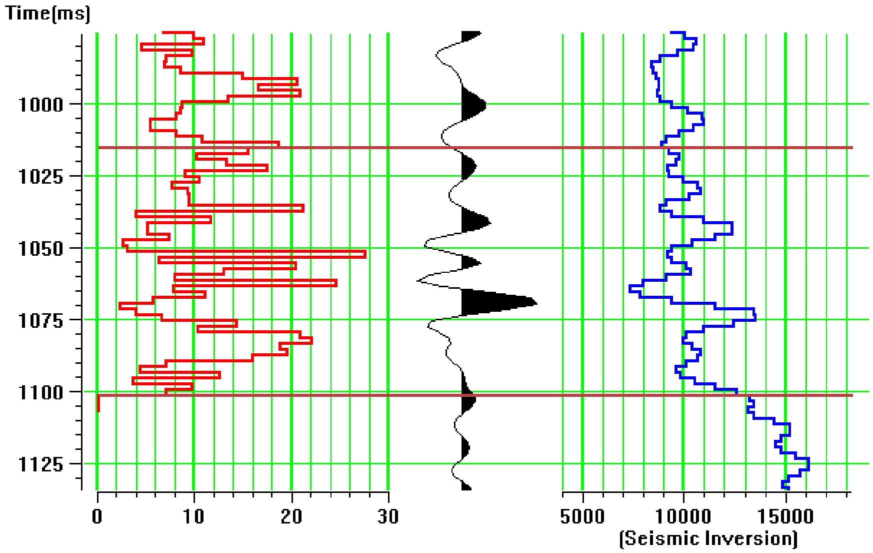


Figure 14: Training data for a single well. The curve on the left is the target porosity log from the well. The center curve is the composite seismic trace from the 3-D volume at the well location. The right curve is the composite acoustic impedance from the seismic inversion. The horizontal red lines show the analysis window.

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 less than 100 ms. Figure 15 shows a cross plot 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.

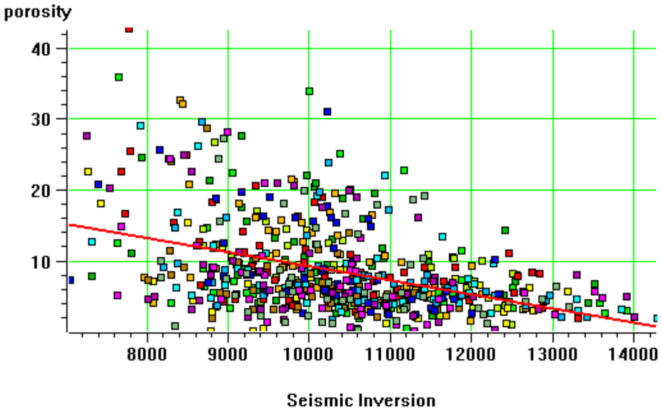


Figure 15: Cross plot of porosity against acoustic impedance from the seismic inversion, using points within the analysis windows from all 13 wells. The normalized correlation is .41.

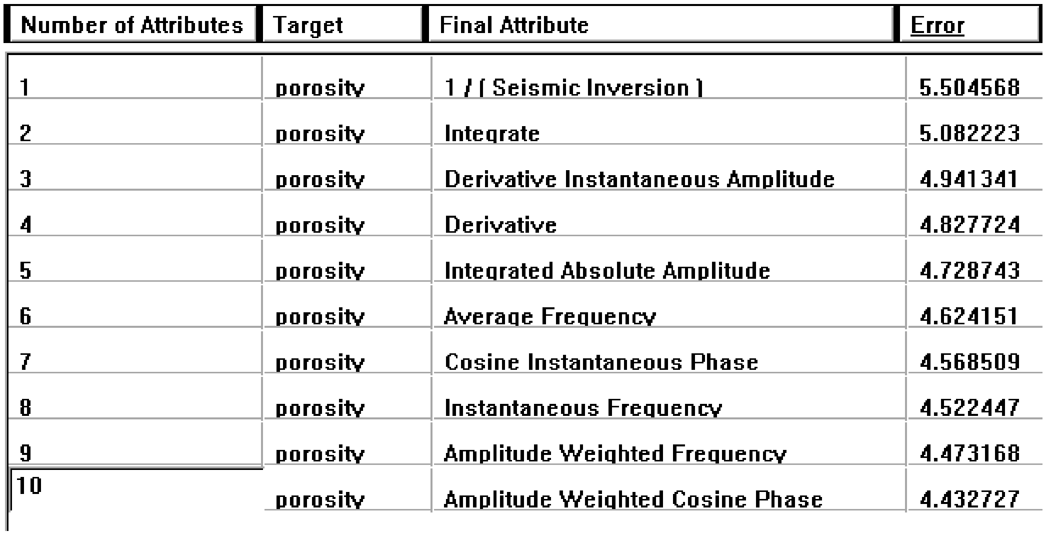
The attributes which are available for the multi-attribute analysis consist of 17 attributes derived from the seismic trace plus the external attribute *Seismic Inversion*. The complete list is:

Amplitude Envelope
Amplitude Weighted Cosine Phase
Amplitude Weighted Frequency
Amplitude Weighted Phase
Average Frequency
Apparent Polarity
Cosine Instantaneous Phase
Derivative
Derivative Instantaneous Amplitude
Dominant Frequency
Instantaneous Frequency
Instantaneous Phase
Integrate
Integrated Absolute Amplitude
Second Derivative
Time
Seismic Inversion

The list is further increased by applying the following non-linear transforms to each of the previous attributes:

Natural Log
Exponential
Square
Inverse
Square Root

The analysis consists of applying the step-wise regression described above. A seven-point convolutional operator was used. The tabular results are shown in Figure 16. Each line of this table shows a multi-attribute transform with an increasing number of attributes. The first line, 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 %). 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%. The table shows combinations up to 10 attributes.



Number of Attributes	Target	Final Attribute	Error
1	porosity	1 / [Seismic Inversion]	5.504568
2	porosity	Integrate	5.082223
3	porosity	Derivative Instantaneous Amplitude	4.941341
4	porosity	Derivative	4.827724
5	porosity	Integrated Absolute Amplitude	4.728743
6	porosity	Average Frequency	4.624151
7	porosity	Cosine Instantaneous Phase	4.568509
8	porosity	Instantaneous Frequency	4.522447
9	porosity	Amplitude Weighted Frequency	4.473168
10	porosity	Amplitude Weighted Cosine Phase	4.432727

Figure 16: The results of step-wise regression, applied to the porosity prediction problem. Each line shows a different multi-attribute transform with the number of attributes listed in the first column. The multi-attribute 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., % porosity).

The same information is displayed graphically in Figure 17, 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 causes over-training of the system.

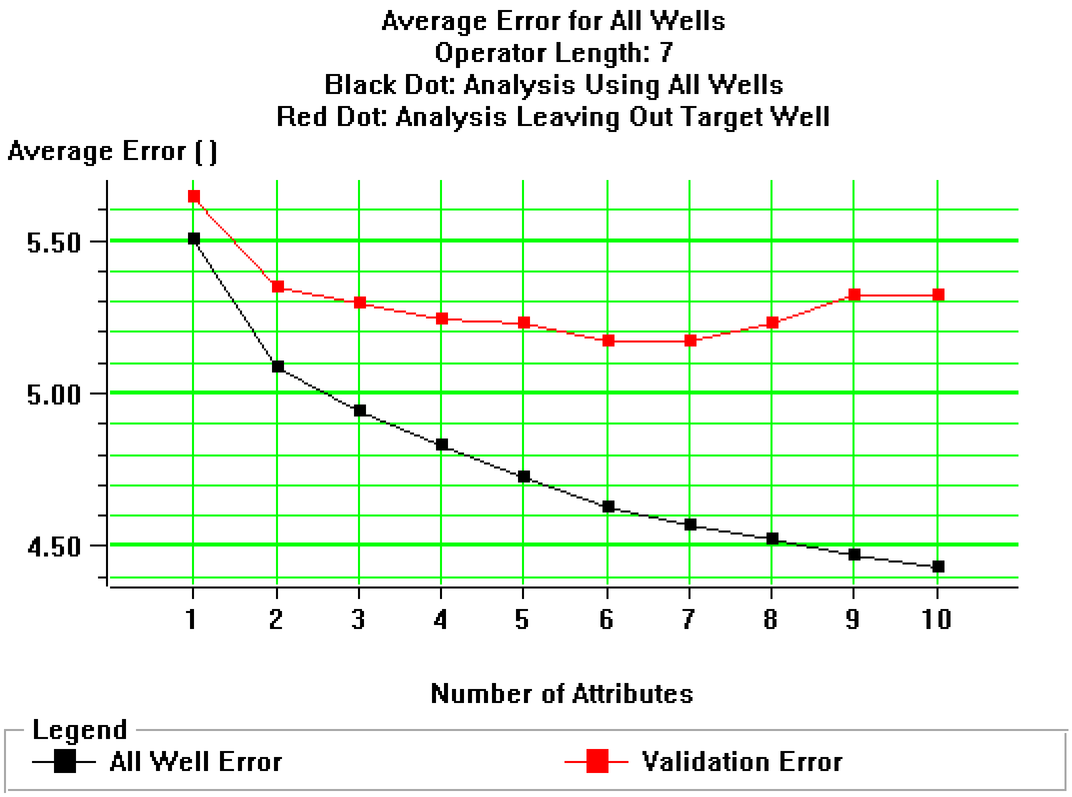


Figure 17: The results of step-wise regression in graphical form. The lower curve shows the prediction error when all wells are used in the analysis. The upper curve shows the validation error, as defined previously.

Figure 18 shows the result of applying the derived multi-attribute transform with six attributes. In this figure, the original porosity logs are shown in black with the predicted logs in red. The normalized correlation for all the wells is now 0.69, compared with the cross plot correlation of 0.41, derived from a single attribute alone.

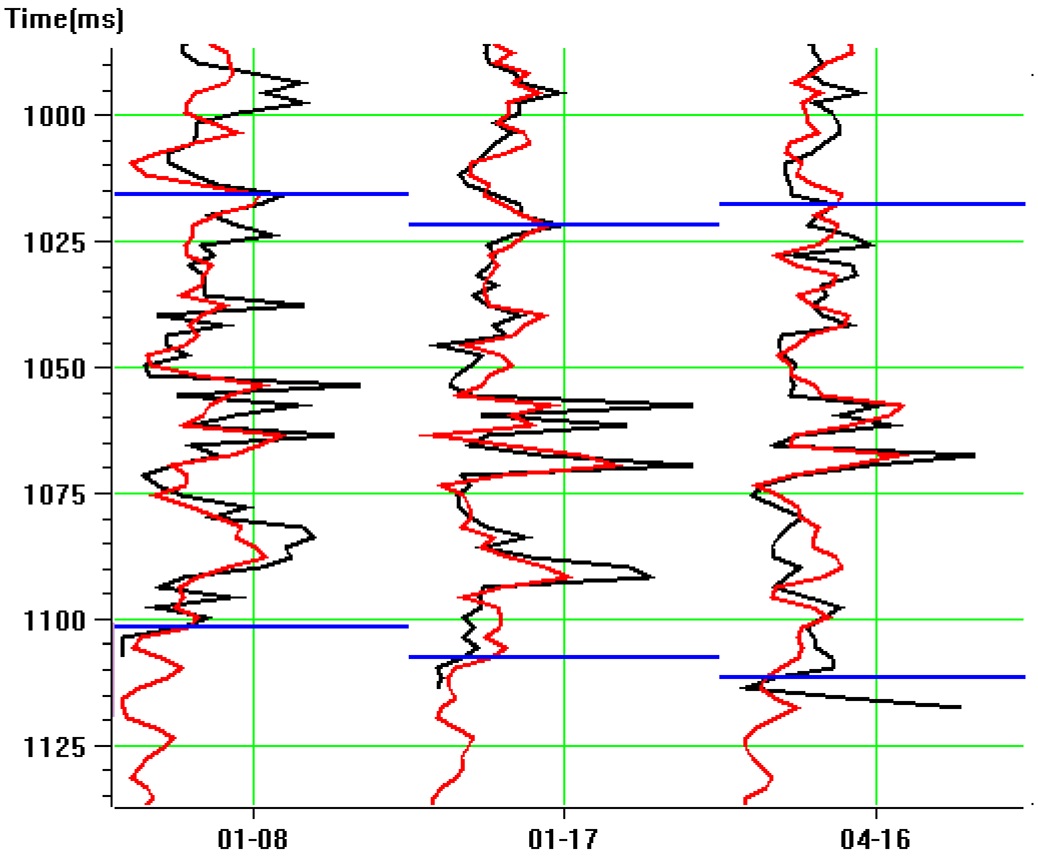


Figure 18: Applying the multi-attribute transform using six attributes and a seven-point operator. Only the first three wells are shown. The original porosity log is shown in black. The predicted log is shown in red. The normalized correlation coefficient for all the wells is 0.69.

Figure 19 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 re-calculated, using reduced training data which did not include that target log. Effectively this simulates the result of drilling that well after the analysis.

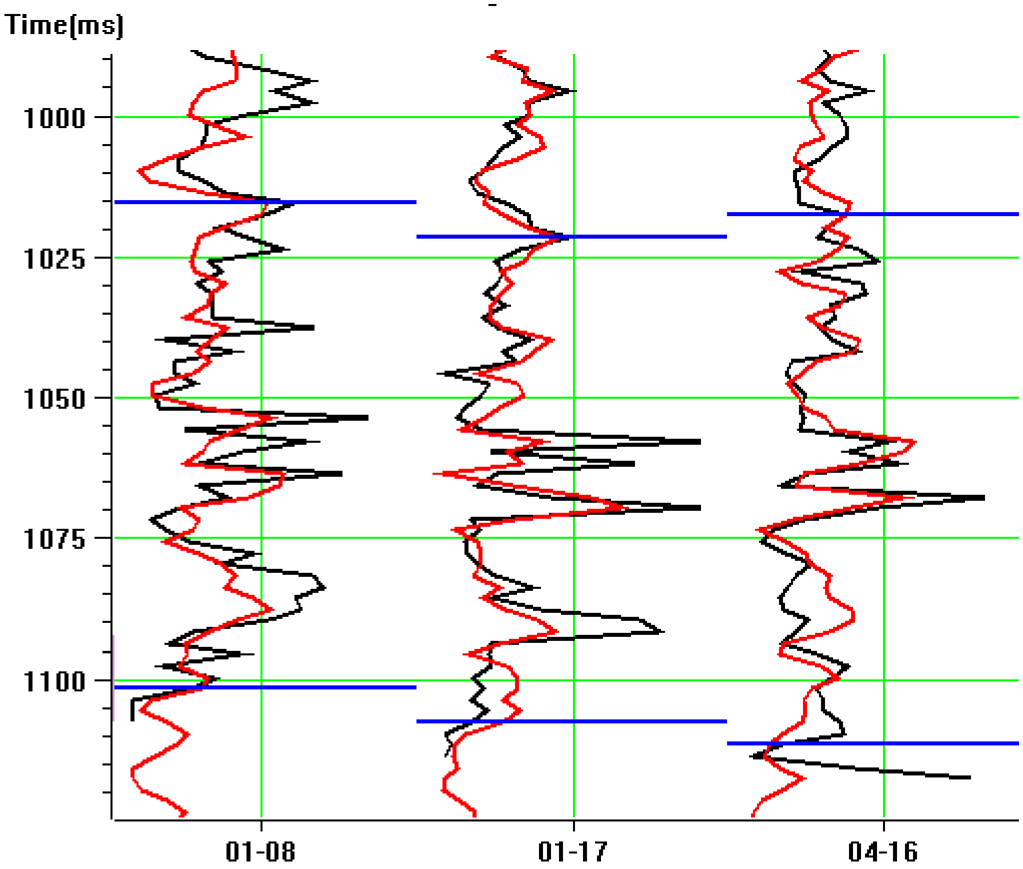


Figure 19: The validation result for the multi-attribute analysis. This is the same as Figure 18 except that the multi-attribute transform for each well has been re-derived with that well data removed from the analysis. This simulates the effect of drilling the well after the prediction. The normalized correlation for all the wells is 0.60.

Figure 20 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.

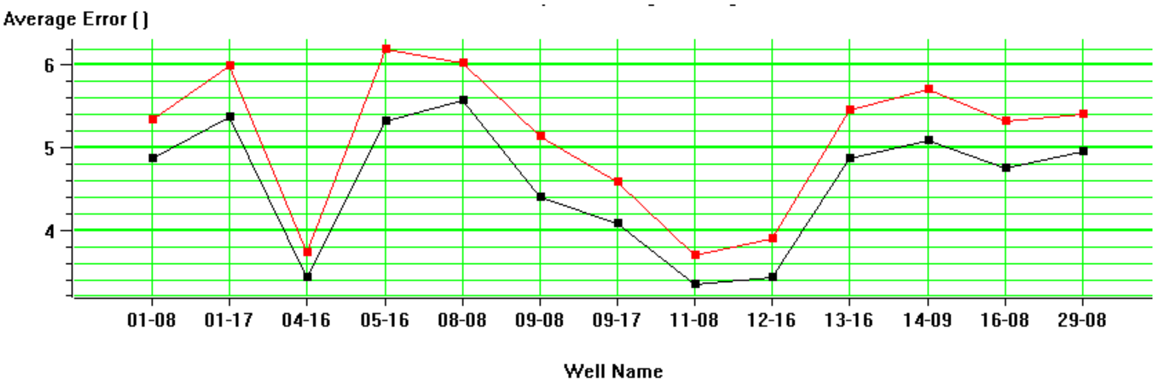


Figure 20: The prediction errors for each of the 13 wells. The lower curve shows the prediction error when the specified well is used in the analysis. The upper curve shows the validation error, when the well is not used in the analysis.

Using the same six attributes, the Probabilistic Neural Network was trained, as outlined in the previous section. Figure 21 shows the prediction results for the PNN. Because the PNN contains a copy of all 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 will always honor the input well information. As with the kriging technique (Deutsch and Journel, 1992), the real measure of performance is cross-validation, and this is shown in Figure 22. The average normalized correlation for all 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 (eg, between 1050ms and 1075ms).

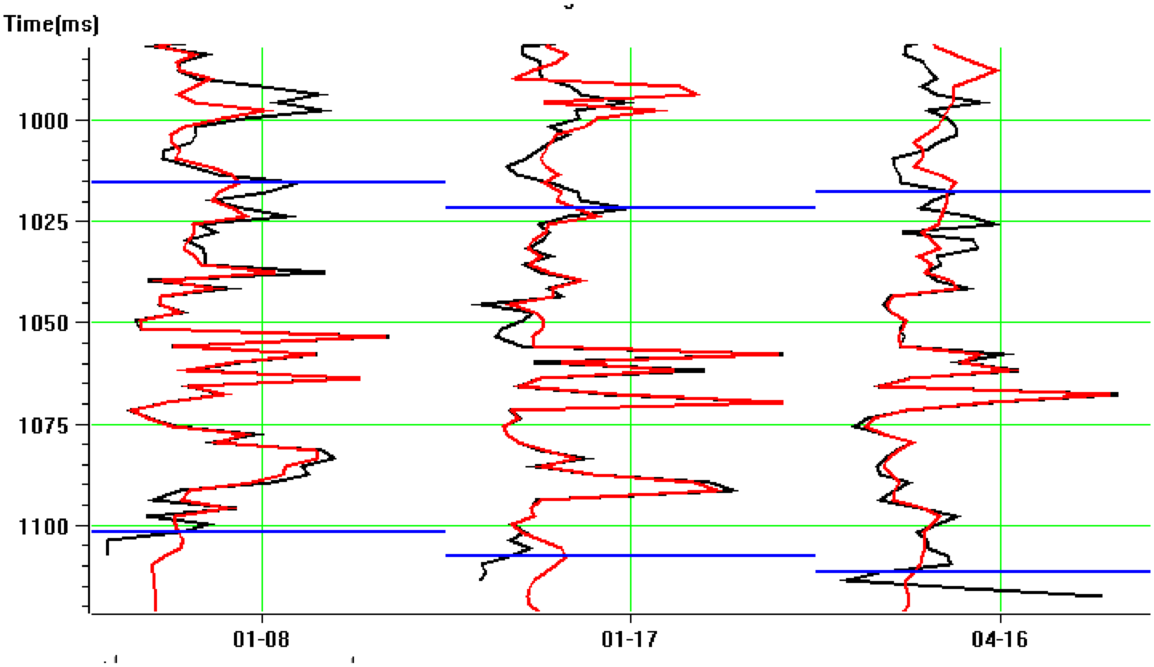


Figure 21: Applying the Probabilistic Neural Network using six attributes and a seven-point operator. Only the first three wells are shown. The original porosity log is shown in black. The predicted log is shown in red. The normalized correlation coefficient for all the wells is 0.95.

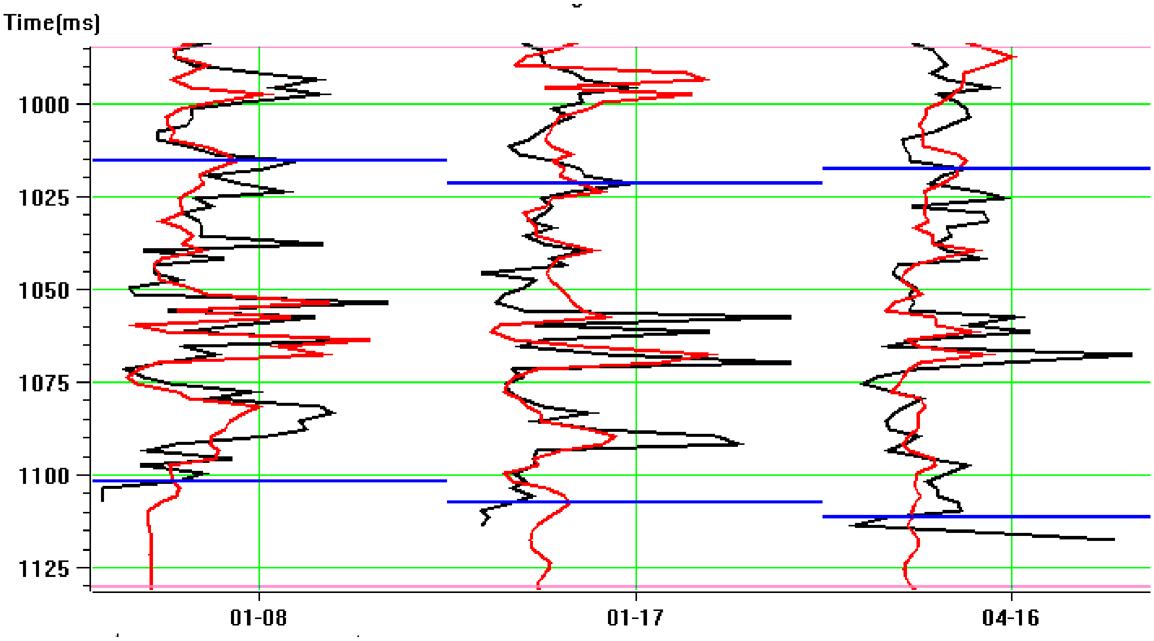


Figure 22: The validation result for the Probabilistic Neural Network. This is the same as Figure 21 except that PNN for each well has been re-derived with that well data removed from the analysis. The normalized correlation for all the wells is 0.62.

Ultimately the value of multi-attribute analysis has to be measured in terms of its improvement over the basic cross plotting of a single attribute. Figure 23 shows the continuous improvement in prediction ability as we move from a cross plot 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.

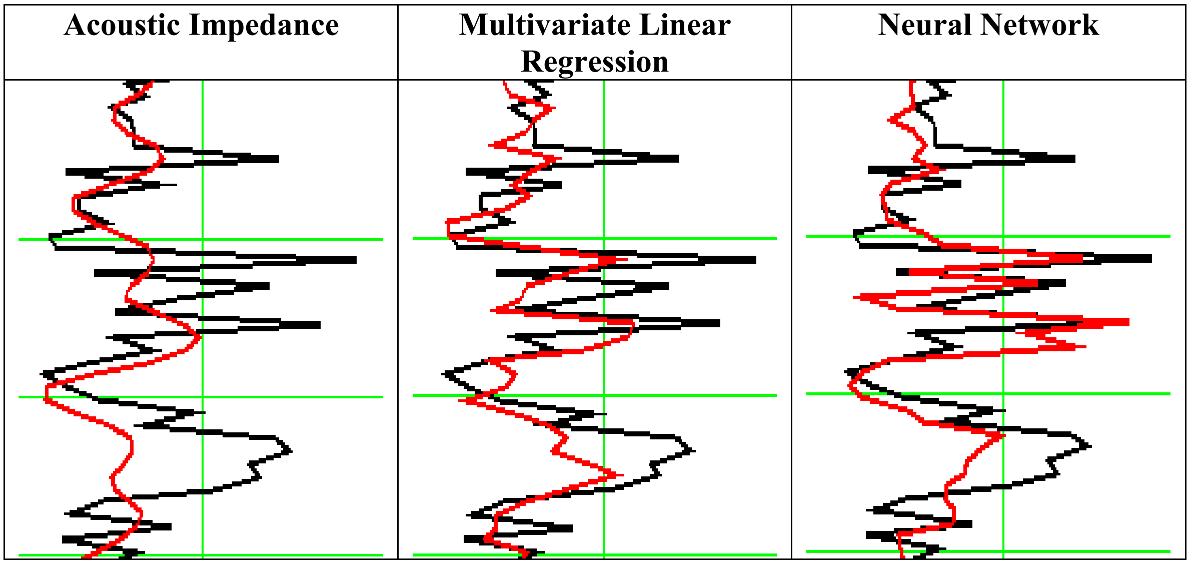


Figure 23: Comparison of validation results for a single well using three prediction methods. Each panel is a blind test, i.e., this well was not used in the operator derivation. The left panel shows linear regression applied to the Acoustic Impedance volume. The middle panel shows multivariate linear regression using six attributes and a seven-point operator. The right panel shows the result of using Probabilistic Neural Network with the same six attributes.

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 24 shows a single inline through each of the volumes. The anomaly at about 1090 ms is a known sand channel. Note the high resolution result achieved with the Probabilistic Neural Network.

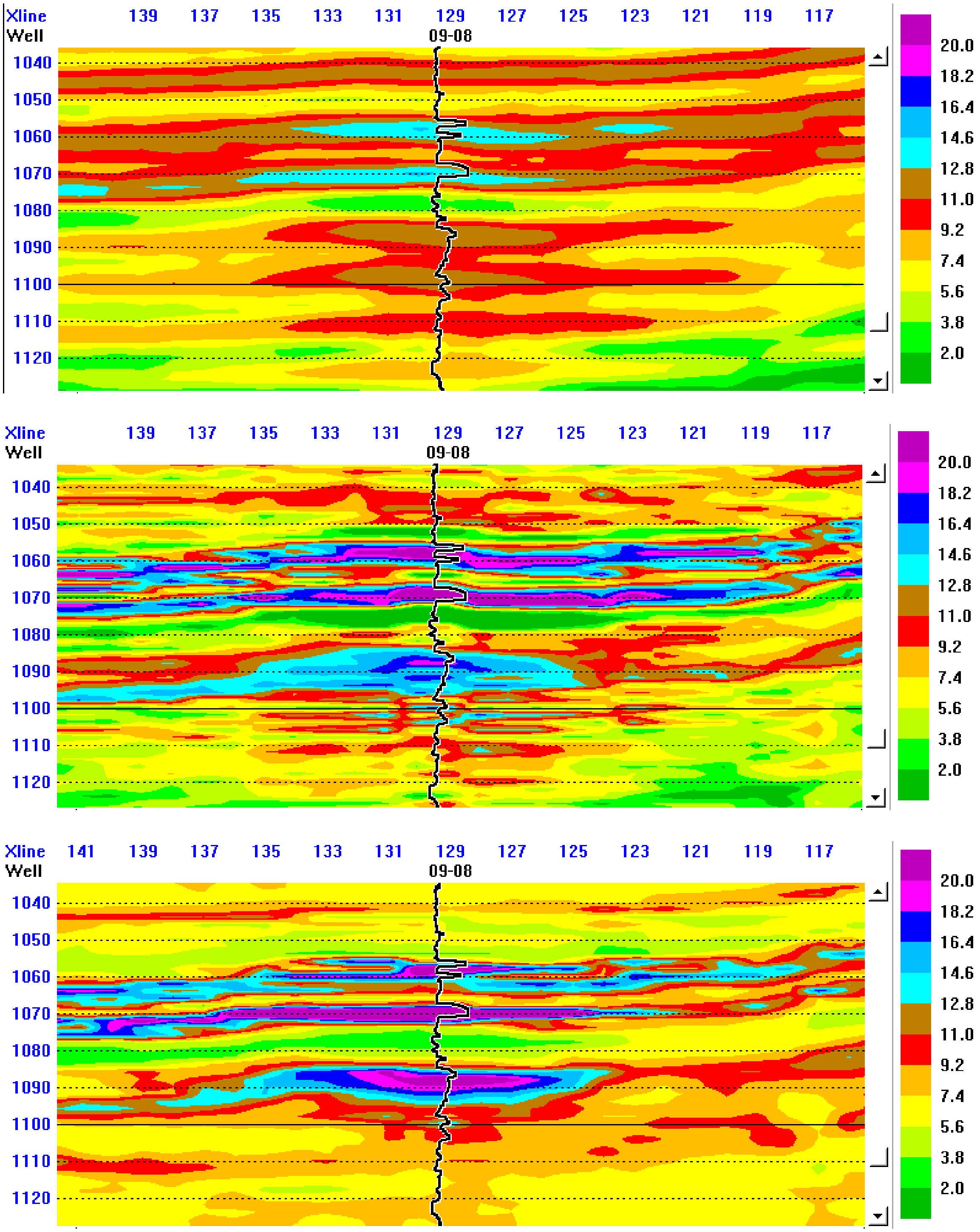


Figure 24: Application of the derived transforms to the 3-D volumes. The upper panel shows the regression curve applied to the acoustic impedance volume. The middle panel shows the multivariate linear transform with six attributes and a seven-point operator. The lower panel shows the Probabilistic Neural Network 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 % porosity.

Finally, Figure 25 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.

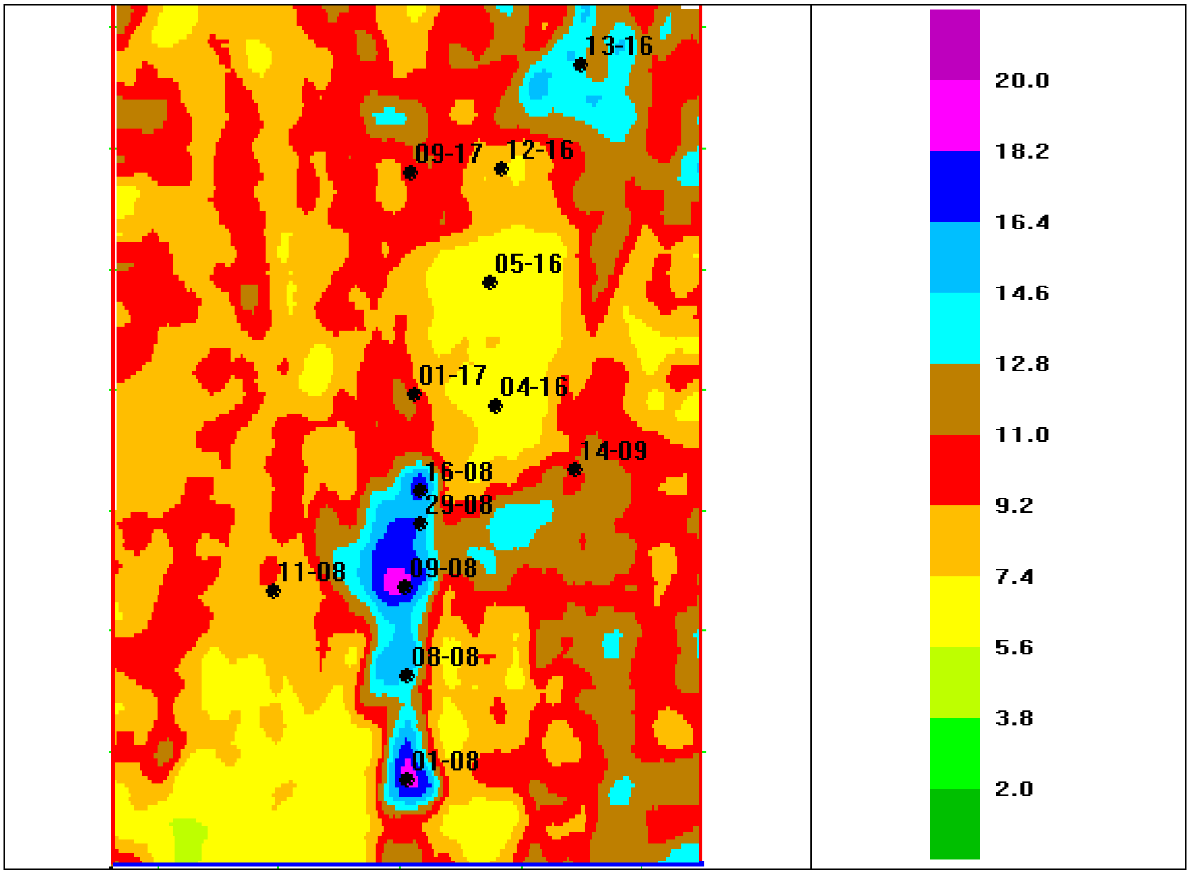


Figure 25: A data slice through the porosity volume estimated using the Probabilistic Neural Network. The color scale is in the units of % porosity.

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 to southwest trending faulted anticline, 7 miles long (N-S) by 4 miles wide (E-W). The Devonian reservoir, at a depth of 11,500 feet, is one of six producing intervals that range in depth from 5,500 to 13,000 feet. A more detailed description of the field, the geology, and the reservoir is given by Schuelke, Quirein and Sarg (1997). The purpose of this example is to show the improvement in resolution both vertically and horizontally using the PNN neural network vs. 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, Quirein and Sarg, 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, or 500 feet thick. The case study area was a small 2 by 3 mile subset of the full 3-D survey area. Figure 26 shows a seismic inline 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, less than 5% porosity. The intervals above and below the Devonian interval are shales. The porosity log shows false porosity in these zones.

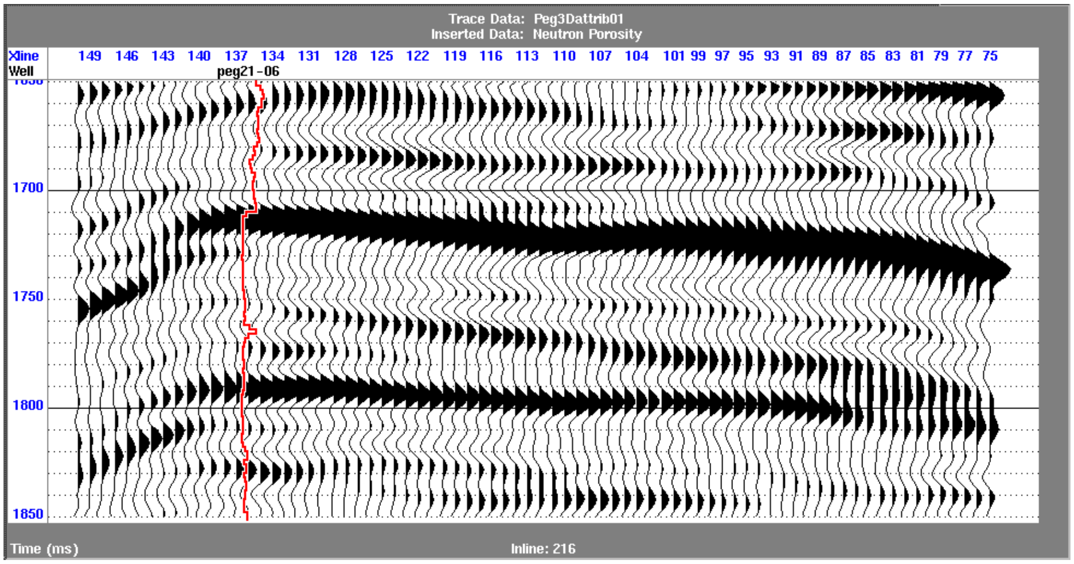


Figure 26. An example seismic line from the 3D survey through the Peg 21-06 well. The porosity log is show as the red curve with increasing porosity to the right.

Figures 27 and 28 show the porosity predictions from the seismic attributes for the same seismic line. Figure 27 is the multivariate linear regression prediction, and Figure 28 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 colored gray. Both predictions show the porosity development within the mid Devonian interval. The multivariate linear regression results, however, show a smoothed, or more averaged, result. 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, as the neural network result is a non-linear 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 bandlimited 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.

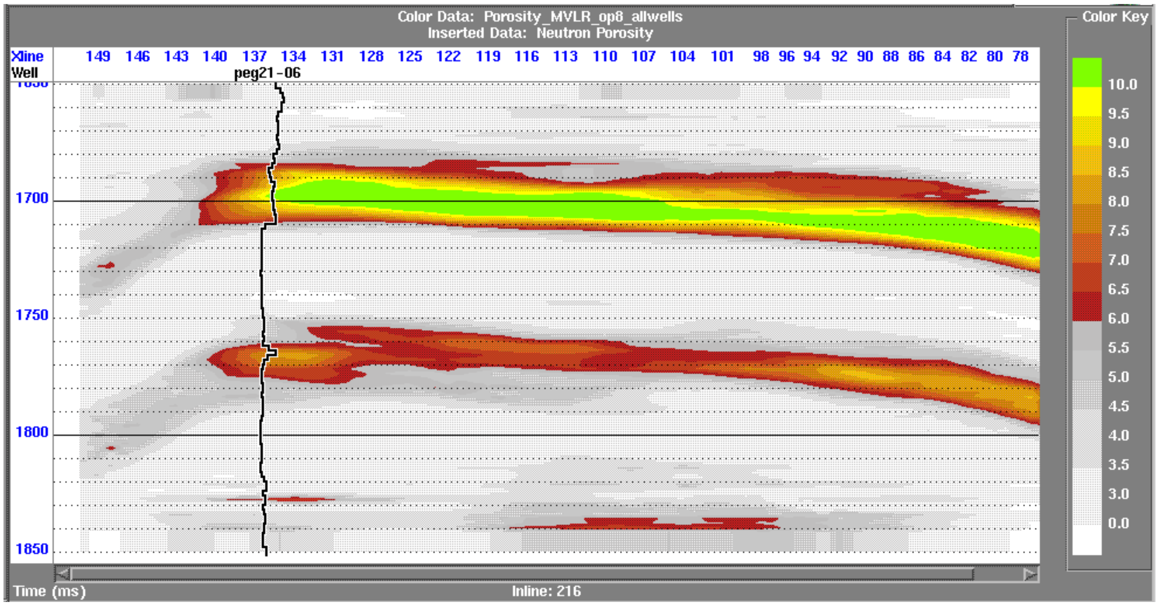


Figure 27. The multivariate linear regression result. Porosity is shown in color. The log porosity for Well Peg 21-06 is displayed as a black curve. The thin high porosity zone in the middle of the Devonian is correctly identified, but the temporal resolution is less on the prediction and the absolute value is less than for the actual porosity log.

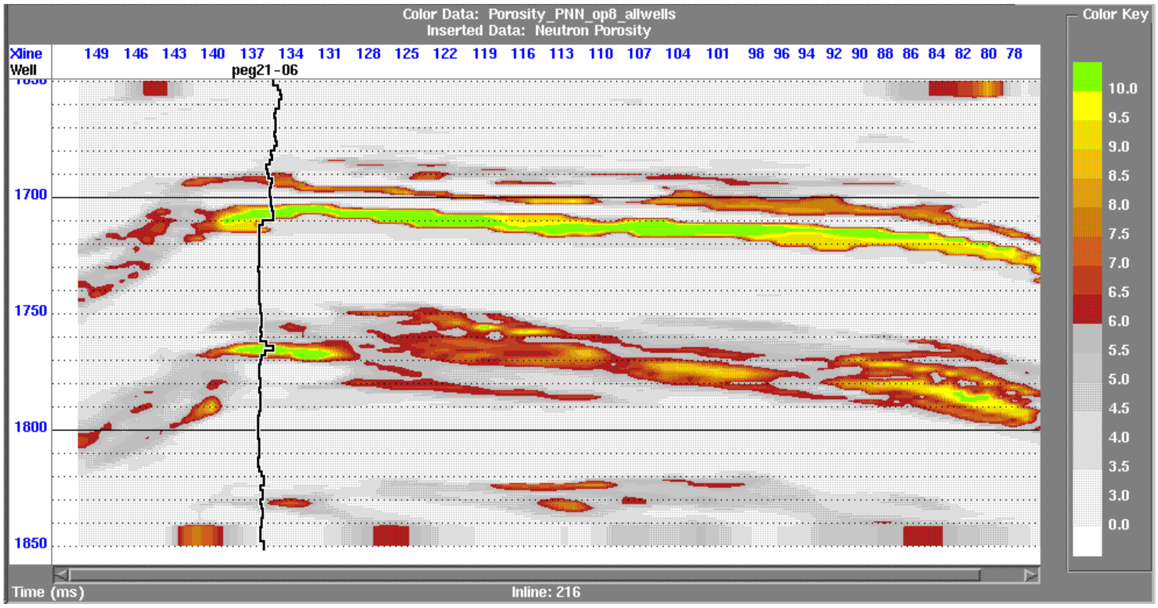


Figure 28. The PNN neural network result. Porosity is shown in color. The log porosity for Well Peg 21-06 is displayed as a black curve. The thin high porosity zone in the middle of the Devonian is correctly identified, and both the thickness and the absolute value of the prediction matches the log porosity. Away from the well control the predictions show the lateral variability expected in this reservoir.

The benefits of this improved vertical and lateral resolution are evident on a time slice view through the two porosity volumes. Figure 29 is a time slice through the multivariate linear regression result at a time of 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 inline displays in Figures 27 and 28. 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 30 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 30 as low porosity zones (gray colors) between the high porosity zones (green color).

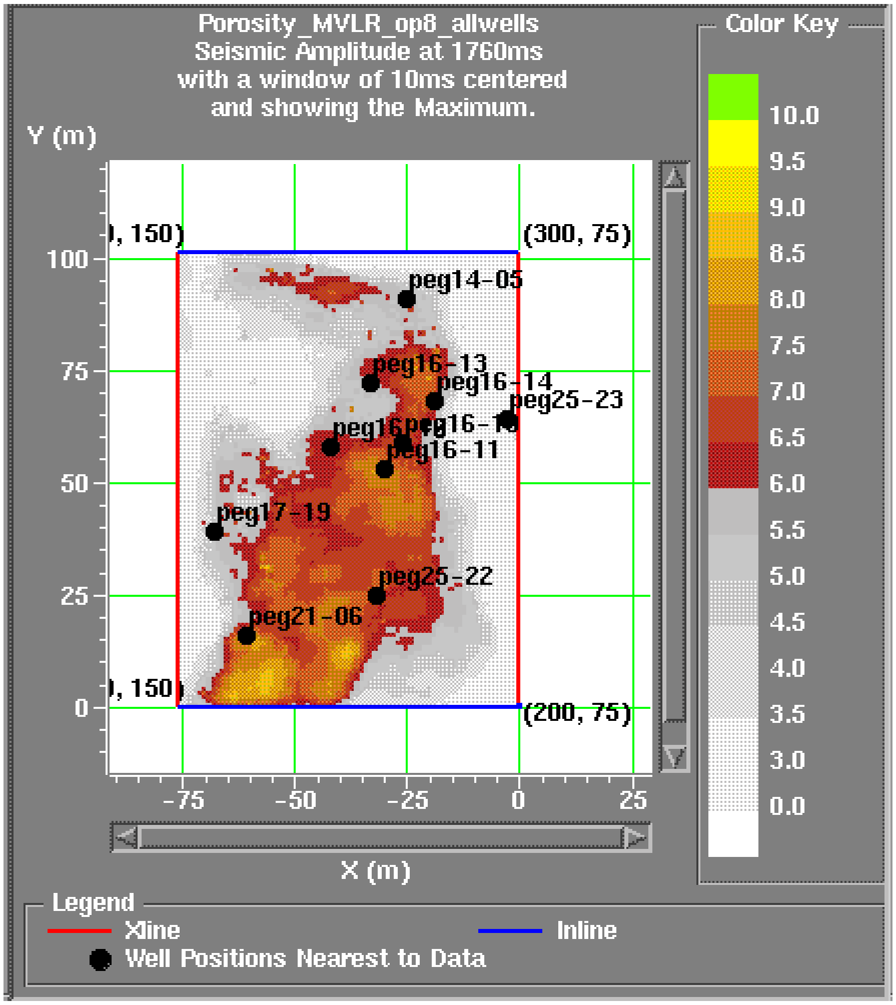


Figure 29. A time slice through the multivariate linear regression results. The predicted porosity is shown in color. Producible porosity zones are in the bright colors grading from red to yellow, with the highest porosity shown in green. Tight zones are indicated with the gray colors.

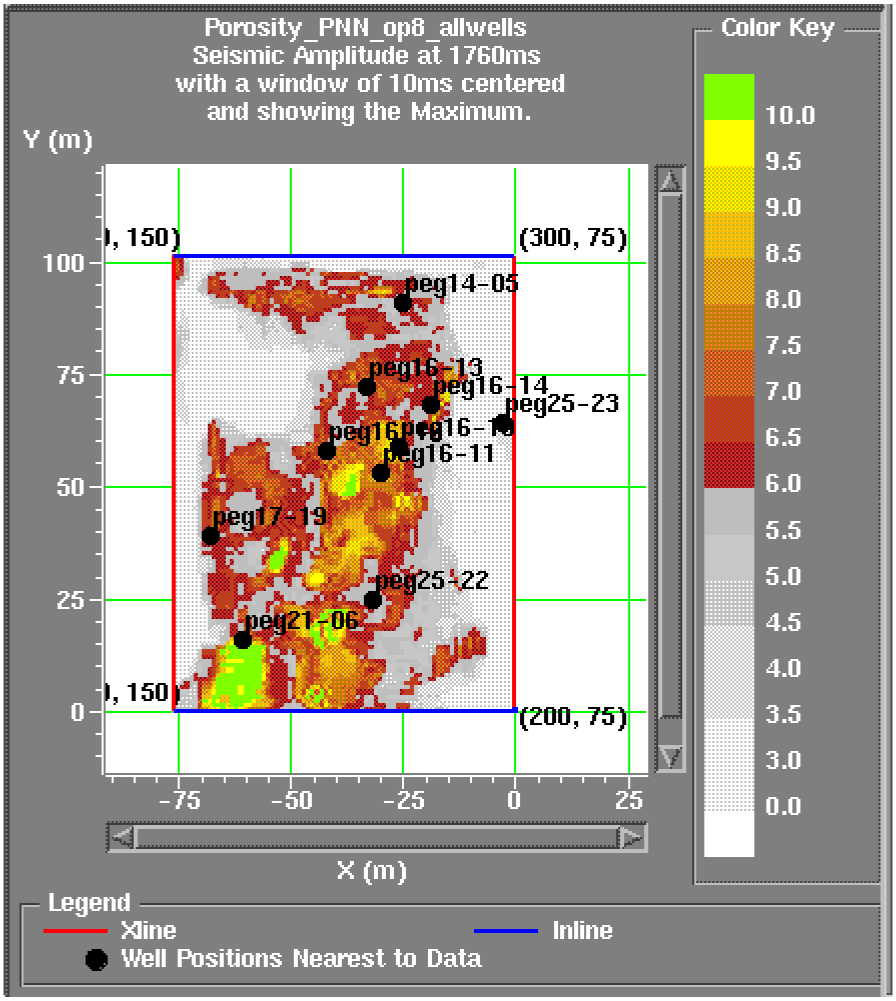


Figure 30. A time slice through the PNN neural network results. The predicted porosity is shown in color. Producible porosity zones are in the bright colors grading from red to yellow, with the highest porosity shown in green. Tight zones are indicated with the gray colors.

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 step-wise regression, which builds up groups of attributes sequentially. We have introduced a modification of conventional regression which includes a 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 multi-attribute 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 multi-layer feed forward network (MLFN), and the Probabilistic Neural Network (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 due to the non-linear 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 cross-plotting to multivariate linear regression to Neural Network. This improvement is evident not only on the training data, but 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:

- It predicts other logs besides acoustic impedance (eg., porosity)
- It may use other attributes besides the conventional stack for this purpose.
- It does not rely on any particular forward model.
- It can achieve greatly enhanced resolution.
- It does not require a knowledge of the seismic wavelet.
- It uses cross-validation as a measure of success.

All of these advantages are gained, however, 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 spans 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.

Acknowledgements

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 insights into the mathematical operation of both the multivariate analysis and the PNN network. Finally, we are grateful to the members of the University of Calgary CREWES Consortium and to Mobil Exploration & Producing U.S., Midland for allowing us to publish the results derived on their data.

References

Anderson, J.K., 1996, Limitations of seismic inversion for porosity and pore fluid: Lessons from chalk reservoir characterization exploration: 66th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 309-312.

Chen, Q. and Sidney, S., 1997, Seismic attribute technology for reservoir forecasting and monitoring: The Leading Edge, Vol. 16, No. 5, May, 1997, 445-456.

Chi, C. Y., Mendel, J. M. and Hampson, D., 1984, A computationally fast approach to maximum-likelihood deconvolution : Geophysics, 49, no. 05, 550-565.

Cooke, D. A. and Schneider, W.A., 1983, Generalized lienar inversion of reflection seismic data: Geophysics, 48, no. 06, 665-676.

Deutsch, C. V. and Journel, A. G., 1992, GSLIB Geostatistical Software Library and User's Guide: Oxford University Press.

Draper, N.R. and Smith, H., 1966, Applied regression analysis: John Wiley & Sons, Inc.

Kalkomey, C.T., 1997, Potential risks when using seismic attributes as predictors of reservoir properties: The Leading Edge, Vol. 16, No. 9, 247-251.

Leonard, J.A., Kramer, M.A., and Ungar, L.H., 1992, Using radial basis functions to approximate a function and its error bounds: IEEE Transactions on Neural Networks, Vol. 3, No. 4.

Lindseth, Roy O., 1979, Synthetic sonic logs – A process for stratigraphic interpretation: Geophysics, 44, no. 01, 3-26.

Liu, Z., and Liu, J., 1998, Seismic-controlled nonlinear extrapolation of well parameters using neural networks: Geophysics, Vol. 63, No. 6, 2035-2041.

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

Masters, T., 1995, Advanced algorithms for neural networks: John Wiley & Sons, Inc.

McCormack, M.D., 1991, Neural computing in geophysics: The Leading Edge, Vol. 10, No. 1, 11-15.

Oldenburg, D. W., Scheuer, T. and Levy, S., 1983, Recovery of the acoustic impedance from reflection seismograms: Geophysics, 48, no. 10, 1318-1337.

Schuelke, J.S., Quirein, J.A., and Sarg, J.F., 1997, Reservoir architecture and porosity distribution, Pegasus Field, West Texas -- an integrated sequence stratigraphy - seismic attribute study using neural networks: 67th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 668-671.

Schultz, P.S., Ronen, S., Hattori, M., and Corbett, C., 1994, Seismic guided estimation of log propties, Parts 1, 2, and 3: The Leading Edge, Vol.13, No. 5,6,&7, May, 1994, 305-310, June, 1994, 674-678, July, 1994, 770-776.

Specht, Donald, 1990, Probabilistic neural networks: Neural Networks, 3, 109-118.

Specht, Donald, 1991, A general regression neural network, IEEE Transactions on Neural Networks, 2(6), 568-576.

Taner, M.T., Schuelke, J.S., O'Doherty, R., Baysal, E., 1994, Seismic attributes revisited: 64th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 94, 1104-1106.

Appendix

Multi-Attribute Linear Regression

Multi-Attribute Linear Regression is an extension of simple linear regression to M variables. That is, we will use M attributes, A_1, A_2, … , A_M, to predict the log, L. To do this, we must determine the M+1 weights, w_0, w_1, w_2, …., 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:

$$\begin{aligned} L_1 &= w_0 + w_1A_{11} + w_2A_{21} + w_3A_{31} \\ L_2 &= w_0 + w_1A_{12} + w_2A_{22} + w_3A_{32} \\ &\;\;\vdots \\ L_N &= w_0 + w_1A_{1N} + w_2A_{2N} + w_3A_{3N} \end{aligned} \qquad \text{(A-1)}$$

where A_{ij} is the j^{th} sample of the i^{th} attribute.

Notice that equations (A-1) can be written as:

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

or:

$$\mathbf{L} = \mathbf{AW} \qquad \text{(A-3)}$$

where **L** is an (N x 1) matrix containing the known log values, **A** is an (N x 4) matrix containing the attribute values, and **W** is a (4 x 1) matrix with the unknown weights.

This can be solved by least squares minimization to give:

$$\mathbf{W = [A^TA]^{-1}A^TL} \qquad \text{(A-4)}$$

As a detailed computation, note that:

$$\begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} N & \sum A_{1i} & \sum A_{2i} & \sum A_{3i} \\ \sum A_{1i} & \sum A_{1i}^2 & \sum A_{1i}A_{2i} & \sum A_{1i}A_{3i} \\ \sum A_{2i} & \sum A_{1i}A_{2i} & \sum A_{2i}^2 & \sum A_{2i}A_{3i} \\ \sum A_{3i} & \sum A_{1i}A_{3i} & \sum A_{2i}A_{3i} & \sum A_{3i}^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum L_i \\ \sum A_{1i}L_i \\ \sum A_{2i}L_i \\ \sum A_{3i}L_i \end{bmatrix} \qquad \text{(A-5)}$$

Multi-Attribute Linear Regression using Convolutional Weights

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

$$L = w_0 + w_1{*}A_1 + w_2{*}A_2 + \ldots. + w_N{*}A_N \qquad \text{(A-6)}$$

where w_0 = a constant,
and w_i = l-point convolutional filters.

To simplify, consider the above equation, 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:

$$\begin{bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \end{bmatrix} = w_0 + \begin{bmatrix} w_1(0) & w_1(-1) & 0 & 0 \\ w_1(+1) & w_1(0) & w_1(-1) & 0 \\ 0 & w_1(+1) & w_1(0) & w_1(-1) \\ 0 & 0 & w_1(+1) & w_1(0) \end{bmatrix} \begin{bmatrix} A_{11} \\ A_{12} \\ A_{13} \\ A_{14} \end{bmatrix} + \begin{bmatrix} w_2(0) & w_2(-1) & 0 & 0 \\ w_2(+1) & w_2(0) & w_2(-1) & 0 \\ 0 & w_2(+1) & w_2(0) & w_2(-1) \\ 0 & 0 & w_2(+1) & w_2(0) \end{bmatrix} \begin{bmatrix} A_{21} \\ A_{22} \\ A_{23} \\ A_{24} \end{bmatrix}$$

(A-7)

This can then be re-arranged as:

$$\begin{bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \end{bmatrix} = w_0 + w_1(-1) \begin{bmatrix} A_{12} \\ A_{13} \\ A_{14} \\ 0 \end{bmatrix} + w_1(0) \begin{bmatrix} A_{11} \\ A_{12} \\ A_{13} \\ A_{14} \end{bmatrix} + w_1(+1) \begin{bmatrix} 0 \\ A_{11} \\ A_{12} \\ A_{13} \end{bmatrix} + w_2(-1) \begin{bmatrix} A_{22} \\ A_{23} \\ A_{24} \\ 0 \end{bmatrix} + w_2(0) \begin{bmatrix} A_{21} \\ A_{22} \\ A_{23} \\ A_{24} \end{bmatrix} + w_2(+1) \begin{bmatrix} 0 \\ A_{21} \\ A_{22} \\ A_{23} \end{bmatrix}$$

(A-8)

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:

$$\begin{bmatrix} w_1(-1) \\ w_1(0) \\ w_1(+1) \end{bmatrix} = \left\{ \begin{bmatrix} A_{12} & A_{13} & A_{14} & 0 \\ A_{11} & A_{12} & A_{13} & A_{14} \\ 0 & A_{11} & A_{12} & A_{13} \end{bmatrix} \begin{bmatrix} A_{12} & A_{11} & 0 \\ A_{13} & A_{12} & A_{11} \\ A_{14} & A_{13} & A_{12} \\ 0 & A_{14} & A_{13} \end{bmatrix} \right\}^{-1} \begin{bmatrix} A_{12} & A_{13} & A_{14} & 0 \\ A_{11} & A_{12} & A_{13} & A_{14} \\ 0 & A_{11} & A_{12} & A_{13} \end{bmatrix} \begin{bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \end{bmatrix} \tag{A-9}$$

Or:

$$\begin{bmatrix} w_1(-1) \\ w_1(0) \\ w_1(+1) \end{bmatrix} = \begin{bmatrix} \sum_{i=2}^{4} A_{1i}^2 & \sum_{i=2}^{4} A_{1i} A_{1i-1} & \sum_{i=2}^{3} A_{1i} A_{1i-2} \\ \sum_{i=2}^{3} A_{1i} A_{1i+1} & \sum_{i=1}^{4} A_{1i}^2 & \sum_{i=2}^{4} A_{1i} A_{1i-1} \\ \sum_{i=1}^{2} A_{1i} A_{1i+1} & \sum_{i=1}^{3} A_{1i} A_{1i+1} & \sum_{i=1}^{3} A_{1i}^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=2}^{4} A_{1i} L_{i-1} \\ \sum_{i=2}^{4} A_{1i} L_{i} \\ \sum_{i=2}^{4} A_{1i} L_{i+1} \end{bmatrix} \tag{A-10}$$