UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

DATA MINING APPLICATIONS IN RESERVOIR MODELING

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE

By

XIAOMENG DONG
Norman, Oklahoma
2016

DATA MINING APPLICATIONS IN RESERVOIR MODELING

A THESIS APPROVED FOR THE
MEWBOURNE SCHOOL OF PETROLEUM AND GEOLOGICAL ENGINEERING

BY

_____
Dr. Deepak Devegowda, Chair

_____
Dr. Matthew Pranter

_____
Dr. Chandra.S.Rai

Acknowledgements

The completion of this thesis could not have been possible without the participation and assistance of so many people, I would like to express my deep appreciation and indebtedness particularly to the following:

Dr.Deepak Devegowda, my advisor, for his constant support and advice. He is the one who guide me into the research and show me the amazing potential and possibility of what data science can achieve.

Dr.Matthew Pranter and Dr.Chandra.S.Rai, for their advice and enthusiasm in my thesis.

To my fellow students and friends for their support and encouragement.

Table of Contents

List of Tables

List of Figures

Abstract

Data science has gained great attentions in many fields over the last decade, in this thesis, I further explored the use of data science technique in oil industry and developed three data mining applications that could be useful for reservoir modeling and exploratory data analysis. A detailed illustration of data mining algorithms such as Support Vector Machines (SVM), Probabilistic Neural Network (PNN) and Ensemble Learning algorithm is incorporated in the thesis. The performance of the proposed workflows are tested on real field data including Barnett Shale play and Mississippi Limestone.

The first two applications are for the Barnett Shale play. For the first application, I used Support Vector Machines for the prediction on lithotypes derives from core data, the prediction algorithm takes a set of well log curves as input and lithotype as output. The test results showed that we achieved 76% accuracy in the blind test well, which indicates that we can identify lithotypes in uncored wells with high accuracy. For the second application, I proposed a workflow that used Ensemble Learning and Probabilistic Neural Network to make prediction on Total Organic Content (TOC) using a different set of well log curves. The blind test results showed that the predicted TOC zones share a great similarity with the core-based TOC measurement in the lab.

The last application is for the Mississippi Lime in north-central Anadarko shelf of Oklahoma. I introduced a new porosity modeling workflow which combines Sequential Gaussian Simulation and Support Vector Machines. The results showed that my proposed workflow allows better use of exploratory data and make a more accurate estimation of the porosity in the reservoir model.

Chapter 1 Introduction

Over the last decades, along with the advancements in computing power and database technology, people are gradually starting to realize that the world we live in is immersed with vast amounts of data, and surprisingly, many answers that we were struggling to find actually lie within the data itself. Since then, people began to research on how to develop a system that could make computer analyze and extract the useful knowledge automatically from big datasets. And later on, the process of discovering patterns and correlations in large datasets is termed with "Data mining". Data mining itself intersects with multiple subjects such as artificial intelligence, machine learning, database managements and applied mathematics.

The application of data mining techniques has been tremendously successful nowadays in many fields such as business, finance, health care and education etc. For example, many retail companies use data mining techniques to conduct consumer analysis then use the results to do some smart marketing and eventually help them increase their profits. In finance, most credit card companies use data mining techniques to find the pattern between people's features (age, income, etc…) and their credit behavior based on credit history data, and then use the pattern to decide the approval of a new credit card applicant.

Compared with many other fields, data mining application in oil industry is more of an emergent field of study. Over these years, although there has been an increasing number of data mining applications being made in oil industry, the use of data mining techniques in oil industry is yet to be fully explored. Among all data mining applications in oil industry, they are most usually used in analyzing exploratory data, which is to

analyze the intrinsic connection between different types of exploratory data. The exploratory data includes seismic data, well log data and core data (Figure 1). Each type of exploratory data has its own advantages and disadvantages. For example, core data is a high-resolution and direct measurement, but the amount of core data in a field is scarce and the cost of acquiring core data is high. Well log data is dense in vertical or horizontal direction along the well bore compared to core data, however, well log is merely an indirect measurement of petrophysical property of wellbore rock, furthermore, well log measurement is subjected to different form of noises in the wellbore and sometimes it is easily affected by malfunction of logging tool sensors. Seismic data is the most abundant among all data source since it covers the entire reservoir volumes, however, due to the nature of seismic data, the data can be very obscure and irregular, which requires much preprocessing and manual interpretation. The degree of resolution and information reliability for seismic data is the lowest among all three types of data.

Reservoir model is mainly built by integrating these exploratory data together. The integration process includes interpretation of exploratory data and interpolation of sparse observations. Data mining technique can play an important role in both tasks above. First, during interpretation process, it can help us find the pattern between observed attributes and desired target with high accuracy and efficiency, which could considerably save labor and time required. Next, during the interpolation process, instead of relying on linear assumption, it allows us to discover non-linear relation between variables, which opens a new world of possibility and opportunity for us. Therefore, data mining technique has the potential to better utilize the available data and increase the data

usage efficiency and eventually provide people with more insights and more accurate information about the target formation.

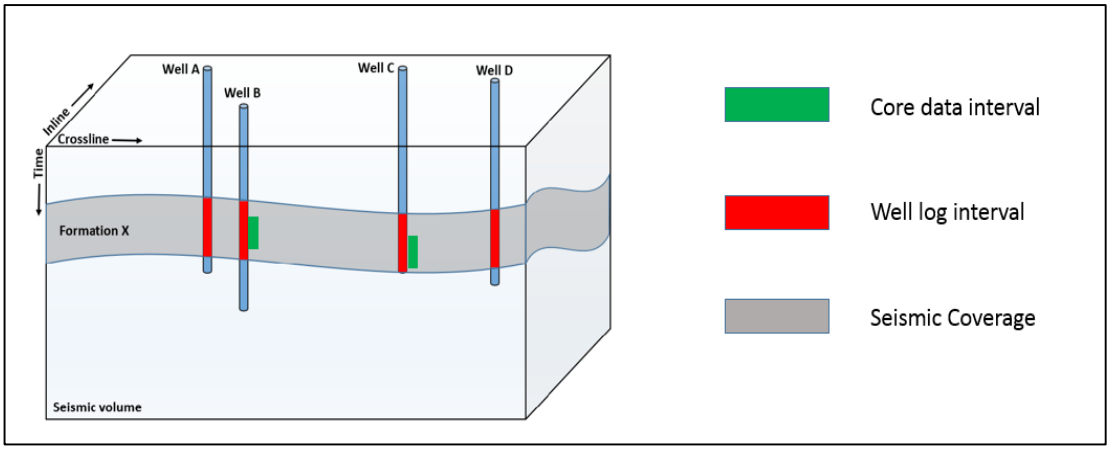

Figure 1. Illustration of Seismic, Well Log and Core Data Coverage.

In this thesis, I mainly focus on the application of two data mining algorithms, Support Vector Machines (SVM) and Probabilistic Neural Networks (PNN). SVM and PNN are both supervised learning algorithms, which can discover the internal pattern between attributes and given targets, then make prediction on target value based on a new set of attributes observed. And over the last decades, SVM and PNN have been increasingly implemented in exploratory data analysis.

Gregory in 1996 first applied PNN to pick seismic events in pre-stack migrated gathers with the attempt to automate the seismic event-picking process (Gregory et al 1996), the features they used are derived from the seismic pixels and labels are from expert's pick. Despite several limitations such as unrepresentative label existed in the study, they offered a promising direction for the later applications of PNN in that area.

David Minken and John Castagna used PNN to predict the behavior of PE well log curve based on seismic attributes, the algorithm managed to predict the basic behavior of PE log in the test well and provided us with great insights in connecting well logs and

seismic attributes (Minken and Castagna 2003). However, due to the resolution difference between seismic and well logs, the PNN did not sufficiently predict the PE value in a quantitative way.

Later, Malleswar further explored the use of PNN in the characterization of coalbed methane (Malleswar et al. 2010). They used seismic attributes to predict the acoustic impedance (AI) log. The training AI log derives from the P-wave and density well logs. They managed to predict the very general behavior of the AI log.

In 2006 , Zhao used support vector machine regression to predict water saturation based on seismic attributes (Bo Zhao et al. 2006), although their water saturation data might not be accurate enough since it was derived from density and resistivity logs, this work offered a great intuition to the later study.

Another support vector machines application is made by Nazari in 2011, in the study, support vector machines regression is used to predict core permeability based on core porosity and well log data at the same depth (Nazari et al. 2011). The study demonstrated that support vector machine can be a promising approach even in sparse datasets.

For TOC evaluation, in 1990 Passey proposed a Δ LogR workflow for evaluating organic richness based on porosity and resistivity logs (Passey et al. 1990). The evaluation is achieved by an overlay of sonic transit time curve and resistivity curve, and it achieved great consistency with the organic carbon measurements from core data. In his work he also demonstrated other evaluation methods that use gamma ray curve, density curve. However, although the Δ LogR method has gained much success in multiple fields, a successful organic richness evaluation is highly dependent on the rescaling process,

which requires a well-trained expert. Furthermore, since the workflow is evaluating TOC only based on resistivity and sonic curves, it omits the information provided in other well log curves such as gamma ray curve and density curve.

Later in 2014, Zhao used a proximal support vector machines classifier and managed to predict lithofacies (shale/limestone) based on multiple seismic attributes in Barnett shale and it provided a great insights to later research (Tao Zhao et al. 2014), it also inspired the study in this thesis.

Chapter 2 Lithotypes and Total Organic Content Prediction in Barnett Shale

In this chapter, I applied data mining techniques to correlate between core data and well log data in Barnett shale, specifically, I use SVM to correlate between lithotypes and well logs, then use PNN to correlate between total organic content and well logs.

2.1 Objectives and Data Description

The Objective for this chapter is to use data mining methods to correlate between well log data and core data (lithotypes and TOC) in cored wells, then make corresponding prediction for uncored wells. Compared with traditional workflow, data mining approach has the potential to save significant amount of time and labor and increase the prediction accuracy, the results could provide critical information for later geomodeling and decision making.

The dataset I use comes from Barnett Shale play in Newark East field (Figure 2). The Barnett Shale play in Newark East field is one of the largest natural gas producing fields in the US. The annual natural gas production of this field is around 318 billion cubic feet.). They are located in Wise and Denton Counties. The maximum thickness of Barnett shale is more than 1200ft in this area. Since shale is the source rock, accurately predicting mineralogy and targeting high TOC zones in uncored wells is the key to completion process and is crucial to a successful field development.

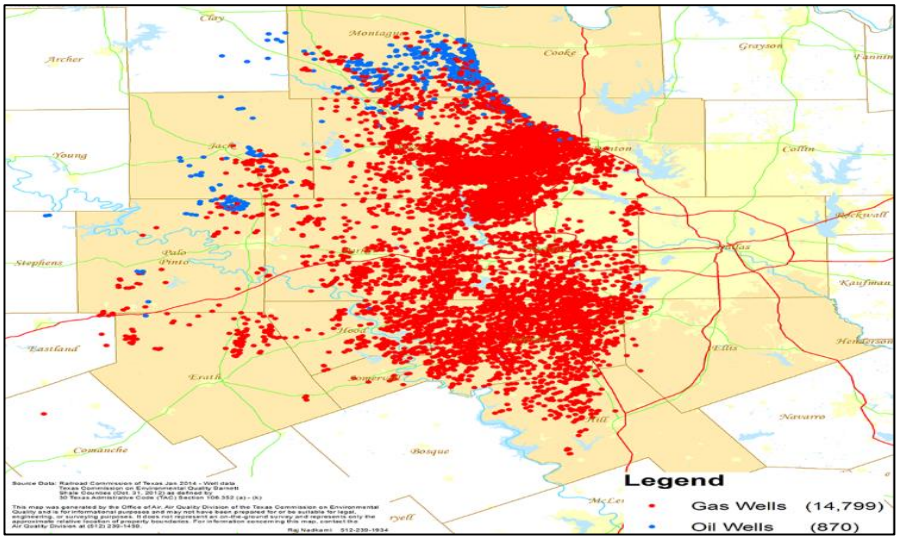

Figure 2. Barnett Shale in Newark East Field Overview (from TECQ).

In Barnett Shale datasets, I have 3 cored wells with both well log data and core data available. The Core data includes mineralogy, porosity and TOC content at core interval. The well log data includes R_s, R_d, SP, GR, DPHI, NPHI, SPHI, PE and RHOB curves. Among available log curves: R_s is shallow resistivity log, R_d is deep resistivity log, SP is spontaneous potential log, GR is gamma ray log, DPHI is density porosity log, NPHI neutron porosity log, SPHI is sonic porosity log, PE is photoelectric absorption well log and RHOB is bulk density log.

2.1.1 Lithotype and TOC Group Description from Core Data

Given the core data in Barnett Shale, for the convenience of analysis and data mining algorithm, the first step has always been breaking the core data into multiple distinct "classes". In the framework of geology and petrophysics, based on different criteria, the "class" can be referred to as "Lithofacies/Lithotypes/Petrofacies/Petrotypes". Sometimes when the criteria overlaps, the terminology can be interchangeable by people. In this study, for the sake of convenience, the term "Lithotype" will being used.

In Barnett Shale formation, Singh (2008) identified 10 different lithofacies based on sedimentary structure, gain size, texture, mineralogy, chemical features and some other biogenic features. These lithofacies are: Siliceous non-calcareous mudstones, Siliceous calcareous mudstone with low calcite, Siliceous calcareous mudstone with high calcite, Silty-Shaly deposites, Phosphatic deposits, Limy mudstone, Dolomitic mudstone, Calcareous laminae, Concretions and Fossilferous deposits. Later, Kale (2009) described the average calcite content, TOC and porosity for each lithofacies as is shown Figure 3.

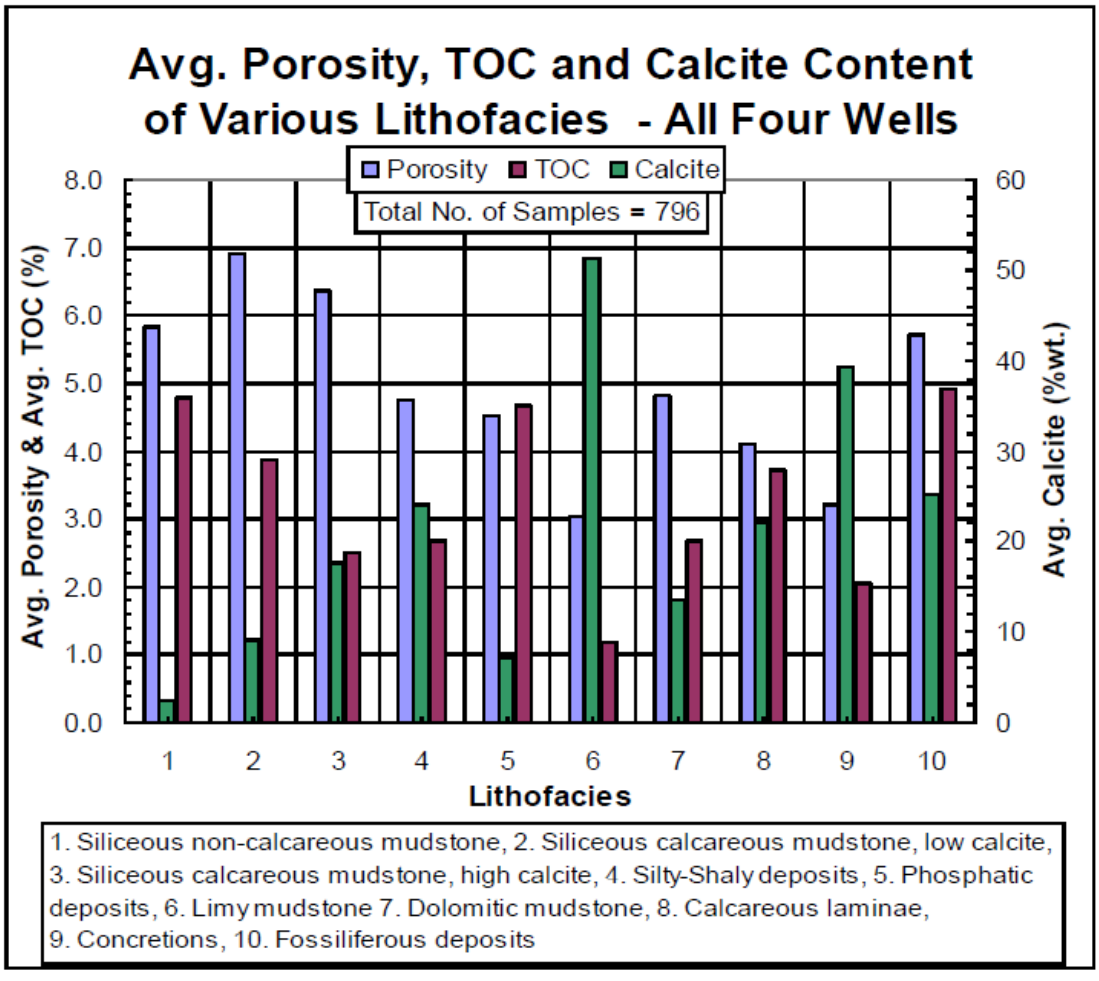

Figure 3. Average Porosity, TOC and Calcite for each lithofacies (Kale, 2009).

In Kale's study, he pointed out that core samples from different lithofacies had very similar petrophysical properties. Furthermore, some lithofacies contribute very little thickness to the entire stratigraphic column. Therefore, he lumped the 10 lithofacies into 3 broader petrofacies based on calcite content, porosity and total organic content as is shown in Figure 4.

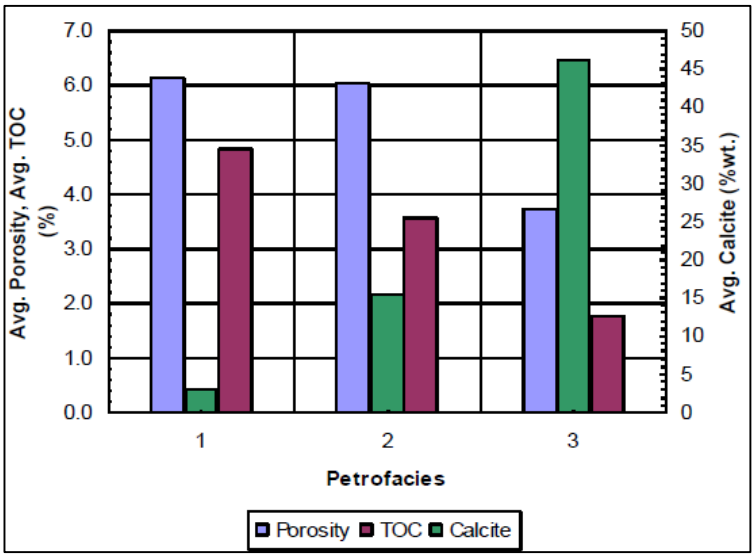

Figure 4. Porosity, TOC and Calcite Content of Three Petrofacies (Kale, 2009).

In my study, different methodology and criteria are used to derive the lithotype. I applied k-means clustering algorithm to group all core data into 3 different lithotypes based on the Quartz, Clay, Carbonate Content, Porosity and TOC. Same as previous work, each lithotype has its own representative mineralogy and petrophysical properties (Figure 5 and Figure 6). Next, I used k-means clustering algorithm to group all core data into a binary TOC group based on only TOC (Figure 7). As a result, each core data interval is associated with two "classes". The first one is Lithotype, which would be the target for SVM prediction. The second "class" is TOC group, it is the target for later PNN prediction.

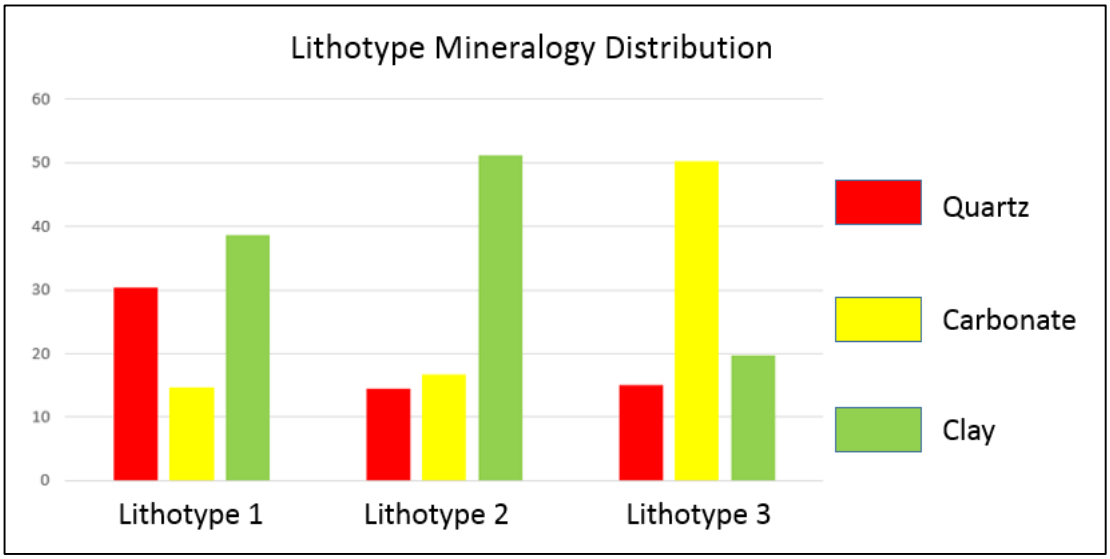

Figure 5. Mineralogy Description for Each Lithotype.

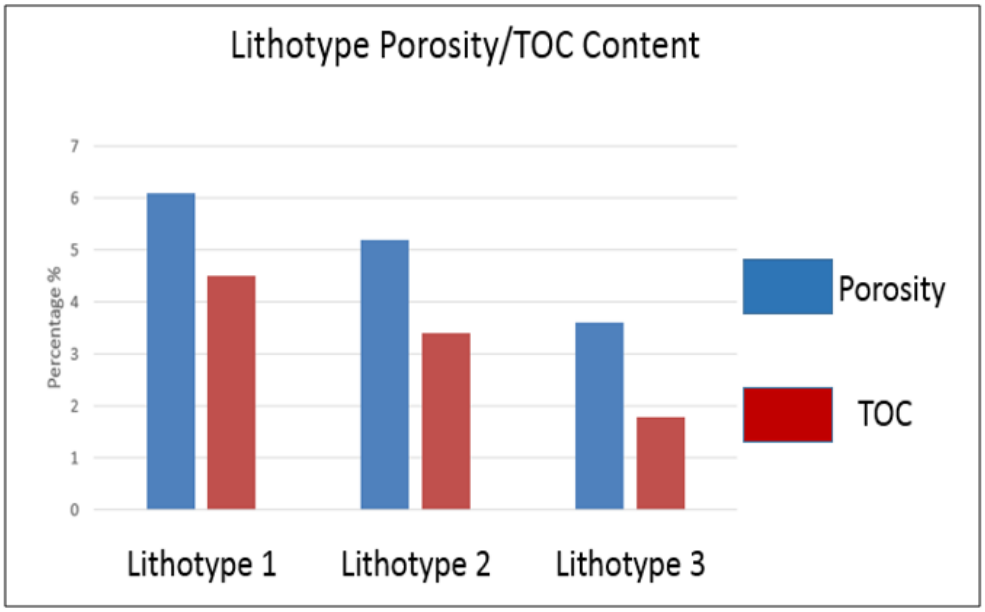

Figure 6. Porosity/TOC Content for Each Lithotype.

By comparing the results in Figure 4, Figure 5 and Figure 6, we can conclude that although the approaches are entirely different, the results from k-means clustering is

extremely consistent with the results from Kale in 2009, which further confirms the results of my clustering scheme. The description for each lithotype is shown below.

Lithotype 1 includes quartz- and clay-rich mudstones (moderately calcareous and non-calcareous) and phosphatic deposits. They exhibit high porosity (>6.1%) and the highest TOC (>4.6%) of the three lithotypes. The quartz content is also the highest of the three lithotypes, which makes it easy to fracture. Therefore, Lithotype 1 represents the best reservoir rock in the formation.

Lithotype 2 includes clay-rich, calcareous mudstones. They exhibit medium porosity (3.9%-6.1%) and TOC (1.9%-4.6%) of the three lithotypes. It is considered to be a reservoir rock with moderate productivity in the field.

Lithotype 3 includes silty and shaly lime mudstones and concretions. The porosity and TOC of this lithotype are the least among all three lithotypes. It represents petrophysically the worst reservoir rock in the formation and is therefore not expected to contribute much to the total production.

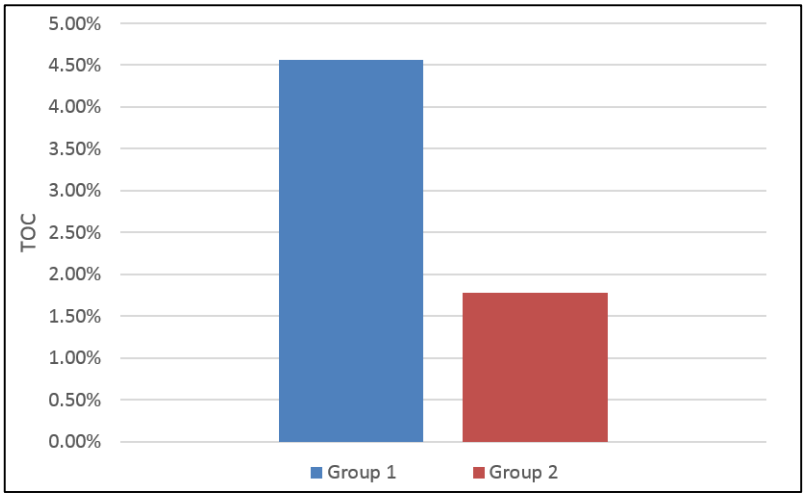

Figure 7. Average Value for Each Total Organic Content Group.

From Figure 7, we can see that there are two TOC groups, which indicates that for TOC group prediction, it is going to be a binary decision-making process. The first TOC group represents the source rock with higher average TOC percentage (TOC ~ 4.53%). The second TOC group represents a rock type with lower average TOC percentage (TOC ~ 1.78%).

2.1.2 Well Log Data Preprocessing

For well log data preprocessing, first I need to extract the log data that has same depth as core sample, then de-spike, smooth and normalization process is applied. During de-spike process, I deleted the well log response that is less than 5 percentile or more than 95 percentile. Next, for smoothing process, a local average smoothing process is being applied, with the maximum average span length equals 5ft. During the normalization process, I simply rescaled most well log curves to be within 0 and 1 with an exception of resistivity curve, where I used a natural logarithm function.

At last, for each prediction task, I need to choose the relevant well log curves as prediction attributes, because data mining approach could not observe something from nothing, and choosing what attributes to use requires knowledge of the subject itself. For lithotypes, since the target comes from grouping mineralogy, porosity and TOC together, I am using SP, GR, DPHI, NPHI, SPHI, PE and RHOB curves as attributes. As for TOC prediction, we know that traditionally, the TOC could be determined from the sonic porosity-resistivity overlay technique, if sonic porosity is missing, neutron porosity or bulk density log could also be used (Meyer 1984), also the uranium content of many source rock that contains kerogen is sometimes a function of TOC, which could affect the

gamma ray logs (Euzen at el 2015). If an irrelevant feature is selected then the irrelevant feature would provide nothing but noise in the prediction task. In this case, PHIS, PHIN, RHOB, Rs and GR curves are used.

For our lithotype prediction and TOC prediction tasks, the true correlation between lithotype/TOC and well log curves might be too complex to find, therefore, we are never sure that the selected well log curves could capture all information needed to make a perfect prediction. What we can do is just to provide as much relevant information as possible without confusing our prediction algorithm. On the other hand, as the number of relevant well log curves being used decreases, the total information about our predicted target would also decrease, which could harm the prediction accuracy.

2.1.3 Data Flow Scheme

In this study, I have in total of 312 data points, each point contains a corresponding logging and core data. It is important to correctly assign the work flow of data in machine learning in order to avoid any misuse of data mining algorithms. The data flow scheme for lithotype prediction and TOC prediction is shown in Figure 8, there are in total of 7 steps.

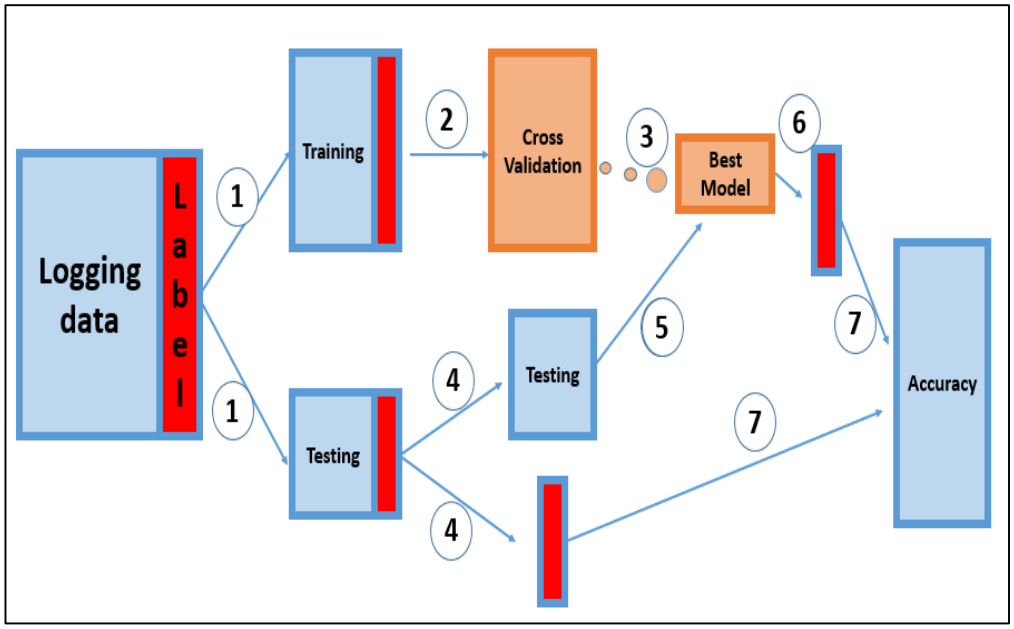

Figure 8. Data Flow Scheme in Lithotype and TOC Prediction.

In step 1, randomly chose 60% to 75% of total data as training example, and the other data as testing examples.

In step 2, only use the training data to build the model. Training Error will be generated from this step, then cross validation will evaluate the performance of each model.

In step 3, pick the model with the least training error as the best model.

In step 4, testing label (Lithotype or TOC group) and testing parameter (well log data) will be separated.

In step 5, only input the test well log data to the algorithm.

In step 6, the SVM/PNN classifier will generate the results from data in step 5.

In step 7, compare the predicted label with the real label, and calculate its prediction accuracy. The error from this step is called Testing Error, or Generalization Error.

2.1.4 K-means Clustering Algorithm

K-means clustering is a simple but very powerful grouping method which regroups the data points every iteration based on the distance from the point to the center of a cluster. The procedure of algorithm simply involved with the following 2 steps:

1. Initialize number of clusters with centers μ1, μ2….μk randomly.
2. Repeat until convergence: {
    a. For each data point, assign points to the nearest center of each group
    b. For every group, calculate the new centroids.}

The diagram of k-means clustering is shown in Figure 9. In Figure 9, first we have several random points (shown in step 1), and arbitrarily select centers of each group (step 2), next we assign each points to the closets group (step 3), then we re-calculate the new center for each group (step 4). Finally we repeat step 3 and 4 until the center converges, and we can see from the step 6 that after sufficient iterations, k-means clustering could nicely group unlabeled points.

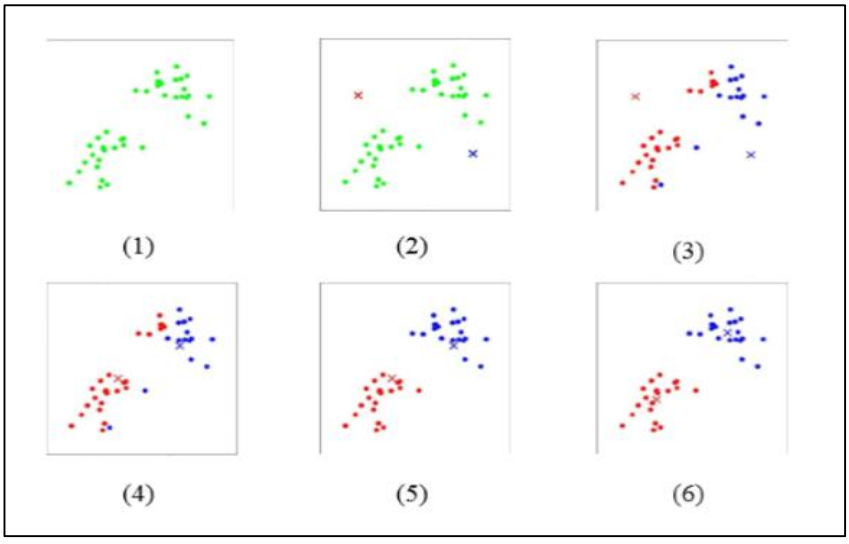

Figure 9. Illustration of Clustering Using the K-Means Algorithm (In step 1, the points are unclassified. We randomly assign cluster centroids in step 2. The data

are assigned to each cluster based on their distance from the centroid in step 3. Then centroids are recomputed in step 4. After repeating step 3 and step 4, the algorithm finally proceeds to convergence as shown in step 5 and step 6).

2.1.5 Cross Validation

Cross validation (CV) is a powerful validation algorithm. Strictly speaking, it is not necessarily any part of data mining algorithm, nevertheless, cross validation could enhance the performance by reducing the likelihood of over-fitting and increase the accuracy for generalization. However, one drawback of CV is that it could waste a certain fraction of the training data. The amount of the waste depends on the approach of CV we chose.

The main advantage of CV is that it gives us a great test accuracy estimates during the training process so that we are able to choose the best performing model among all models. In this study, due to limited number of data points, 10-fold cross validation is being applied.

The steps of 10-fold cross validation is as followed:

1. Separate the training set into 10 subsets and call these subsets S1 to S10.
2. Evaluate each model M_j (generated by SVM or PNN) as followed:
    a. For each i=1 to 10, train the model on the all the training set except Si, and get one hypothetical answer.
    b. Test the answer on Si to get test error.
    c. The total estimated blind test error is calculated as the average of test error from i=1 to 10

3. Change another model, M_{j+1}, and repeat the step 2 until all models are tested.

4. Pick the best model, and train the best model again on the entire training set to get the final model.

2.2 Lithotype Prediction Using Support Vector Machines

2.2.1 Backgrounds of Support Vector Machines

Support Vector Machines (SVM) are a powerful data mining method for classification and regression. The original concept of SVM was first proposed by Vapnik in 1963. His motivation back then was to better classify a set of binary data (Figure 10). Intuitively he generated two planes that can make the margin between two classes as wide as possible. Then the decision boundary (called hyperplane) lies at the middle of two separating planes. Some vectors in the dataset will "support" the separating plane, therefore those vectors are called support vector. At that time SVM was a hard-margin linear classifier, which means it is not suited for non-linearly-separable case. The advantages for SVM back then is that it simplifies the classification problem by only considering the support vectors instead of the whole dataset. However, it cannot handle any non-linear classification problem and the computing power back then was not able to test its performance.

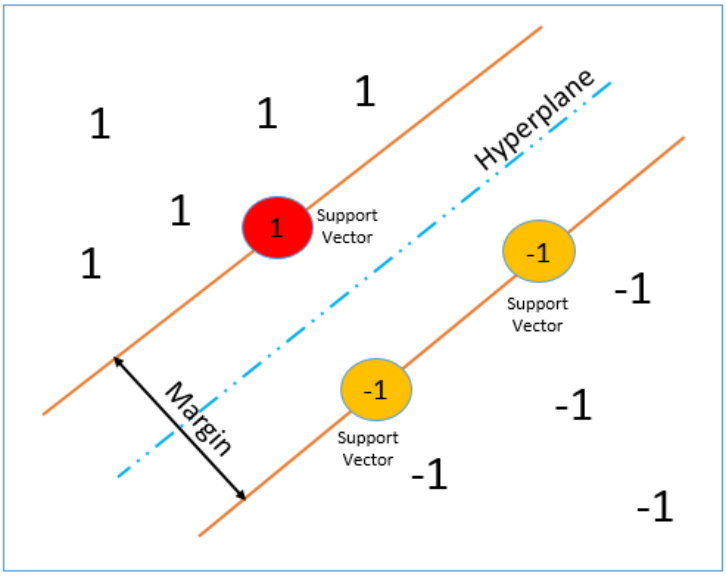

Figure 10. Support Vector Machine Hard Margin Classifier (the data is separated by the hyperplane created by support vectors).

However, in 1992, the application of Kernel method in SVM (Kernel trick) enables the linear classifier to accomplish non-linear transformation by going into a high dimensional feature space (Vapnik et al. 1992) In Figure 11, when SVM encounters a nonlinearly-separable case, it could map the input space into a higher-dimensional feature space until it become linearly separable, after separating the data using a hyper-plane, it would map back to the input space.

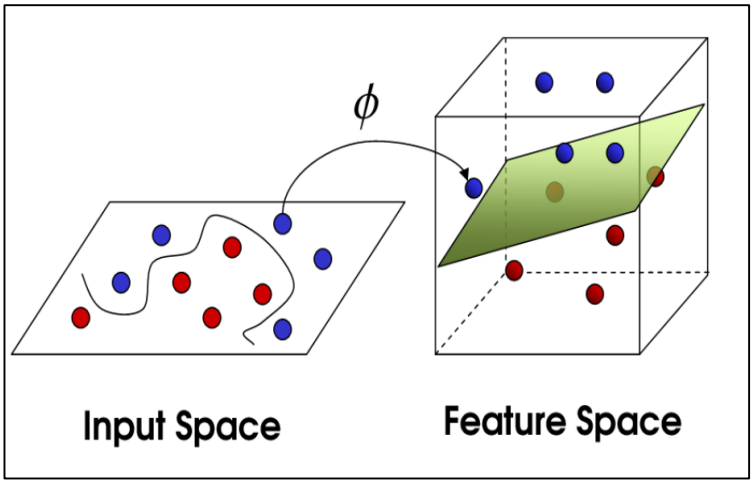

Figure 11. Non-linear Transform of Support Vector Machines.

The current standard Support Vector Machine (soft margin SVM), which allows for mislabeled instances, is developed in the middle of 90s (Cortes and Vapnik 1995). The soft margin SVM further boosts the performance of classification and reduced the likelihood of over-fitting. In 1995, it outperformed neural network and achieved a great success in handwritten character recognition using a slightly non-linear polynomial kernel function. As soon as it was shown to work in handwriting recognition, machine learning practitioner began to further explore its use in text recognition, image recognition, and medical science. It is now considered as one of the most popular and successful supervised learning methods in machine learning algorithms.

2.2.2 Modeling Scheme

The total work scheme for SVM prediction is shown in Figure 12. Following the process described in 2.1.2, I use k-means clustering algorithm to group core data into 3 different lithotypes. Then I will input the preprocessed well log data and lithotype index into Multiclass SVM classifier to learn the pattern. After training, SVM classifier will

take only well log in uncored wells as the input and make prediction on lithotype based on pattern found during the training process. Finally I will compare the predicted lithotype with real lithotype from core data and calculate the prediction accuracy then use it for evaluating the results.

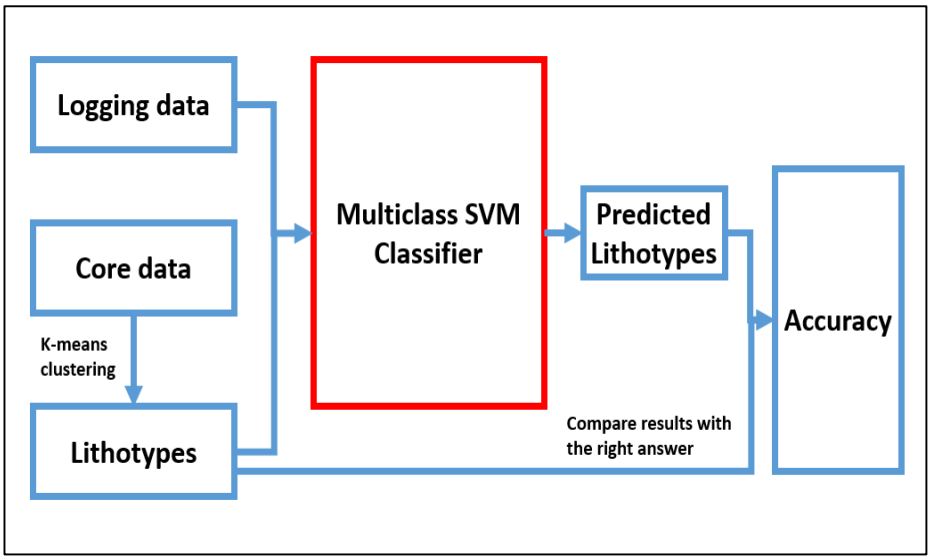

Figure 12. Modeling Scheme of Support Vector Machines Lithotype Prediction.

2.2.3 Multiclass SVM Classifier Components

This section will take a closer look at what is inside the "red box" in Figure 12.

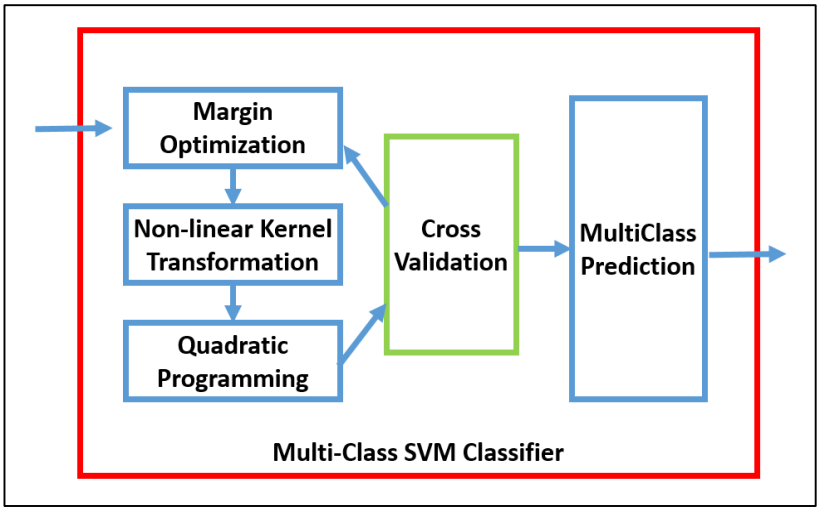

Figure 13. Detailed Workflow of Multi-class SVM.

As is shown in Figure 13, the general description of how multi-class SVM works is as followed: (1) Right after I input well logging data and lithotype, a linear margin optimization problem will be constructed. (2)Then apply the "Kernel trick" in the SVM to map the logging data into a higher dimensional feature space. (3) Next the algorithm will hand our optimization problem to quadratic programming (QP) solver, under sufficient iterations, QP can give us the number of support vector and each weight accordingly. (4) Then the weight will be handed over to cross-validation to determine the best parameter that should be used in margin-optimization and non-linear Kernel transformation. (5) After finding the best parameter, I will train the algorithm with the best-selected parameter again and make the multi-class prediction. The mathematical detail for each step will be provided later in next section.

2.2.4 Margin Optimization

Margin Optimization is a binary soft-margin linear SVM classifier. In soft-margin SVM, we define:

$$w^T x_n + b \geq 1 - \xi_n \quad (1)$$

If Xn belongs to the positive group (yn=1).

$$w^T x_n + b \leq -1 + \xi_n \quad (2)$$

If Xn belongs to the negative group (yn=-1).

Where:

w : the weight vector for x array

x_n : input parameter(in our case, logging data at one depth)

b: constant number

ξ_n : the positive tolerance for soft-margin classification

The larger the left hand side in equation (1), SVM would be more confident to assign x_n to the positive class. And vice versa in the equation (2).

And the optimization problem for SVM is

$$\text{Minimize:} \quad \frac{1}{2} w^T \mathrm{w} + \mathrm{C} \sum_{n=1}^{N} \xi_n \quad (3)$$

$$\text{Subject to:} \quad y_n(w^T x_n + b) \geq 1 - \xi_n \quad \text{for } \forall \text{ n } \in \mathrm{N} \quad (4)$$

In equation (3) and (4), C is the cost parameter in the objective function, it reflects the trade-off between maximizing the margin and making left hand side in equation (4) larger than right hand side. N is the total number of data.

Then in order to solve the minimization problem with constraints, we need to apply Lagrange multiplier to formulate the optimization without constraints to make equation easier to solve. The resulting equation becomes Minimizing the Lagrangian (L) with respect to w, b and ξ, and maximize with respect with positive α_n and β_n as shown in equation (5)

$$\mathrm{L}=\frac{1}{2}\mathrm{w}^T\mathrm{w} + \mathrm{C} \sum_{n=1}^{N} \xi_n - \sum_{n=1}^{N} \alpha_n [y_n(w^T x_n + b) - 1 + \xi_n] - \sum_{n=1}^{N} \beta_n \xi_n \quad (5)$$

Then we calculate the derivative with w, b, ξ_n and set it to 0, we get:

$$\frac{\partial \mathrm{L}}{\partial \mathrm{w}} = \mathrm{w} - \sum_{n=1}^{N} \alpha_n y_n x_n = 0 \tag{6}$$

$$\frac{\partial \mathrm{L}}{\partial \mathrm{b}} = -\sum_{n=1}^{N} \alpha_n y_n = 0 \tag{7}$$

$$\frac{\partial \mathrm{L}}{\partial_{\xi_n}} = \mathrm{C} - \alpha_n - \beta_n = 0 \tag{8}$$

After we plug equation (6) (7) (8) back to equation (5), the Lagrangian becomes:

Maximize:

$$\sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{j=1}^{N} y_n y_j \alpha_n \alpha_j x_n x_j \tag{9}$$

Under constraints:

$$0 \leq \alpha_n \leq \mathrm{C} \quad \text{For} \quad \forall \, \mathrm{n} \in \mathrm{N} \tag{10}$$

$$\sum_{n=1}^{N} \alpha_n y_n = 0 \tag{11}$$

So equation (9) is the linear soft-margin optimization problem that we are going to send to the next step and equation (10) and (11) are constrains that will follow with it.

2.2.5 Non-linear Kernel Transform

In equation (9), at the rightmost we have to compute $x_n^T x_j$, which means the only information that algorithm needs is the dot product of attributes, then imagine if we are going to a higher dimensional feature space, then we would need the dot product in the higher dimension to do the optimization, thus if we want to map the attributes into a higher dimensional space the equation (9) will become:

Maximize:

$$\sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{j=1}^{N} y_n y_j \alpha_n \alpha_j \Phi(x_n) \Phi(x_j) \tag{12}$$

Where $\Phi(x)$ represents the transformation of x in a higher dimensional space.

$$\text{We define:} \quad \mathrm{K}(\mathrm{x}_n, \mathrm{x}_j) = \Phi(\mathrm{x}_n)\Phi(\mathrm{x}_j) \tag{13}$$

Equation (9) becomes:

Maximize:

$$\sum_{n=1}^{N} \alpha_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{j=1}^{N} y_n y_j \alpha_n \alpha_j K(x_n, x_j) \tag{14}$$

In equation (14), the rightmost term "K" stands for kernel trick, it is a function of x_n and x_j and the value of the function equals the dot product of higher dimensional feature space. Therefore, it has the name "Kernel trick". Applying kernel trick allows the SVM to project to higher dimensional space without paying the price for huge amount of computation in higher dimensional feature space. Two popular Kernel function will be provided and introduced.

$$\mathrm{K}(\mathrm{x}_n, \mathrm{x}_j) = \left(\mathrm{a}\mathrm{x}_n^T \mathrm{x}_j + \mathrm{b}\right)^Q \tag{15}$$

$$\mathrm{K}(\mathrm{x}_n, \mathrm{x}_j) = \exp(-\gamma \left\| x_n - x_j \right\|^2) \tag{16}$$

Equation (15) is polynomial kernel function, its dimension depends on the value of Q and the dimension of x. The dimension of polynomial kernel function is limited.

Equation (16) is called radial basis function (RBF), this is the Kernel we used in lithotype prediction. If we expand the exponential part using Taylor Series, we would find out the dimension of RBF is actually infinite.

After the Kernel trick, regroup of equation (14) is needed in order to send it to next step, so the equation (14) becomes:

Minimize:

$$\frac{1}{2}\sum_{n=1}^{N}\sum_{j=1}^{N} y_n y_j \alpha_n \alpha_j K\left(x_n, x_j\right) - \sum_{n=1}^{N} \alpha_n \tag{15}$$

Under constraints:

$$0 \leq \alpha_n \leq \mathrm{C} \text{ for } \forall \mathrm{n} \in \mathrm{N} \tag{16}$$

$$\sum_{n=1}^{N} \alpha_n y_n = 0 \tag{17}$$

Looking at equation (15) to equation (17), we could notice that our problem is a typical constrained quadratic programming optimization problem. Such optimization is impossible to solve analytically, therefore in order to move forward, we need to apply numerical quadratic programming techniques to solve the problem.

2.2.6 Quadratic Programming

The QP problem of equation (15) to equation (17) in matrix vector form is as followed:

Minimize:

$$\frac{1}{2}\alpha^T \begin{pmatrix} y_1 y_1 K(x_1, x_1) & \cdots & y_1 y_N K(x_1, x_N) \\ \vdots & \ddots & \vdots \\ y_N y_1 K(x_N, x_1) & \cdots & y_N y_N K(x_n, x_n) \end{pmatrix} \alpha + (-1^T)\alpha \tag{18}$$

Under constraints:

$$\mathrm{y}^T \alpha = 0 \tag{19}$$

$$0 \leq \alpha \leq C \tag{20}$$

There are several popular algorithms for QP, since the optimization problem in SVM is convex optimization, the interior-point-convex method (Boyd 2004) will be applied. But for larger dataset, SMO algorithm (Platt 1998) and improved SMO algorithm (Keerthi et al. 2001) could handle this pretty well. For the scope of this study, QP optimization is not included.

After optimization, the optimal α vector will be solved by QP, and we would notice a lot of elements in α is zero, it means the corresponding training data does not contribute in the decision boundary. For those elements which α is non-zero, we would name the

corresponding training data “support vector”, which will later be passed through multi-class SVM prediction.

2.2.7 Multi-class SVM Prediction

The most popular approach for multi-class SVM prediction is to transform the multi-class problem into several binary SVM classifiers. Some prevalent methods are: one-versus-all method using winner-takes-all strategy (WTA_SVM); one-versus-one method using max-wins voting (MWV_SVM). There are some experimental results showing that WTA_SVM, MWV_SVM are competitive with each other and there is no clear superiority of one over another. (Duan and Keerthi 2005)

In lithotype estimation, WTA_SVM method is used, from k-means clustering results we have 3 groups of cores. Then we will train 3 sets of binary SVM classifier. w and b from each SVM classifier will be calculated as followed:

From equation (6), we have

$$\mathrm{w} = \sum_{n=1}^{N} \alpha_n y_n x_n \tag{21}$$

For any support vector on the margin, based on equation (4) we have:

$$y_n(w^T x_n + b) = 1 \tag{22}$$

Combine equation (21) and (22) and apply kernel nonlinear transformation we have:

$$y_n\left(\sum_{n=1}^{N} \alpha_n y_n \, \Phi(\mathrm{x}_n)\Phi(\mathrm{x}_j) + b\right) = 1 \tag{23}$$

Therefore:

$$b = -\frac{1}{y_n} - \sum_{n=1}^{N} \alpha_n y_n \, \Phi(\mathrm{x}_n)\Phi(\mathrm{x}_j) \tag{24}$$

For each binary classifier, given a testing point x_j I will calculate the "assignment confidence index" (ACI) for the group using equation (25), equation (26) and equation (27).

$$ACI_1 = \sum_{n=1}^{N} \alpha_n y_n \Phi(x_n)\Phi(x_j) + b_1 \quad (25)$$

$$ACI_2 = \sum_{n=1}^{N} \alpha_n y_n \Phi(x_n)\Phi(x_j) + b_2 \quad (26)$$

$$ACI_3 = \sum_{n=1}^{N} \alpha_n y_n \Phi(x_n)\Phi(x_j) + b_3 \quad (27)$$

Then the predicted lithotype index is the group that gives us the biggest ACI value. The mathematical expression is:

$$lithotype\ index = \arg\max_i(AC_1, AC_2, \cdots AC_i) \quad (28)$$

2.2.8 Test Results of SVM Lithotype Prediction

For test results, in order to get a comprehensive overview of testing results, I will use 3 different testing fractions of total data: 25%, 33% and 40%. And for each fraction, run 10 times of step 1-7 illustrated in Figure 11 with each time randomly choose certain fraction of total data as training data and the other as testing data. The results are shown in table 1.

Test Fraction	25%	33%	40%
#Testing points	78	104	125
Experiment #	Accuracy	Accuracy	Accuracy
1	77.2%	81.2%	79.2%
2	69.4%	74.3%	71.0%
3	72.2%	78.2%	72.7%
4	69.5%	75.0%	70.8%
5	70.1%	74.0%	74.7%
6	78.6%	76.7%	70.4%
7	77.1%	75.1%	79.3%
8	72.7%	76.1%	72.3%
9	80.0%	77.8%	71.2%
10	69.4%	73.2%	73.4%
Average	73.6%	76.2%	73.7%
Standard Deviation	4.0%	2.3%	3.0%

Table 1. Experimental Results for SVM Lithotype Prediction.

The results showed that the optimal testing fraction is 33% in the three different fractions. And I achieved an average of 76.2% accuracy with 2.3% standard deviation.

Since the total number of data is constant, if we increase the testing fraction, the amount of training data will decrease, as a result, the SVM will not be able to build a stable model from decreased number of training example.

On the other hand, if testing fraction decreases, SVM might be able to build a better model using more training data, but the testing part becomes unstable. Imagine an extreme case: if we use all the data except one as training data, then test the model on that single point, the testing error would result either 100% or 0%, which is obviously not the real case.

2.2.9 Blind Test Results and Interpretation

In reality we might have three cored wells and over a hundred uncored wells. We could not randomly select data from all wells and make prediction. So I apply another blind well test to evaluate the performance of multi-class SVM classifier. The data from well 1 and well 2 will be used to train the SVM, then SVM classifier will predict the lithotypes of blind test well at each depth interval based on only well log data of well 3. The blind well test results are shown in Figure 14.

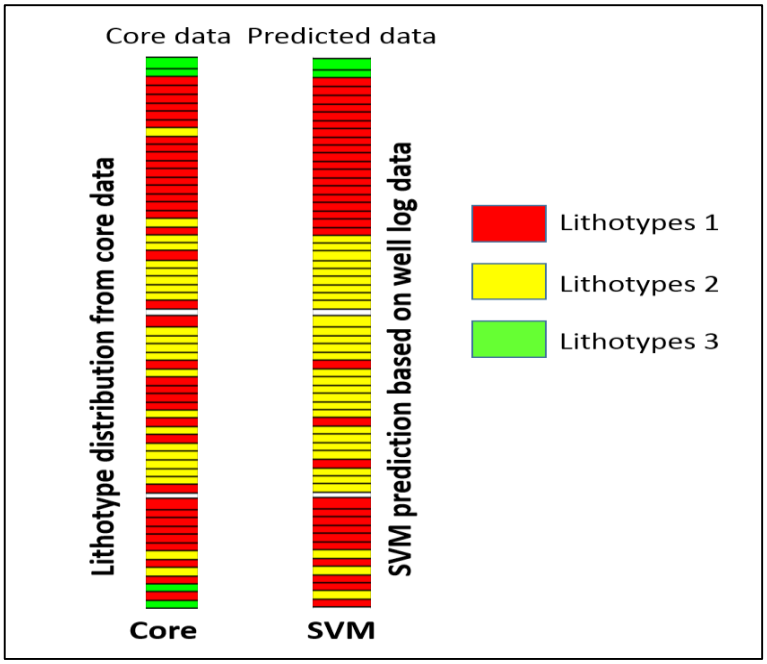

Figure 14. SVM Blind Test Results for Prediction of Lithotypes Along Wellbore (lithotype distribution acquired from core is shown in left, lithotype distribution prediction based on well log data is shown in right).

By building a model with data from two wells and then making a prediction on the third well, under a well performed well log normalization process, 76% accuracy was achieved. In Figure 14, color column illustrates the lithotype distribution of formation rock in well 3. Red represents type 1, yellow represents 2, and green represents 3. The left color column next to the logging data in well 3 comes from the core data using k-means clustering, and the right color column is the predicted lithotype distribution from the support vector machine only based on well log data.

The SVM lithotype prediction based on well log show a similar vertical distribution to the lithotype index obtained from the core, which indicates it is possible to identify the distribution of lithotype in a blind well with great accuracy using multi-class SVM classifier. When we apply this SVM model to make lithotype prediction on uncored

wells, it could offer us great insights about the mineralogy and petrophysical property of rock along the wellbore, which would be considerably helpful in deciding completion zone and in later lithofacies geomodeling.

2.3 Total Organic Content Prediction Using Ensemble Learnings of Probabilistic Neural Networks

Probabilistic Neural Networks (PNN) is a multi-layered feed forward network (Specht 1990), it is one of the derivations in Artificial Neural Network (ANN) family and it predominantly handles classification jobs. PNN is a very effective and easy-to-implement classifier, the application of PNN in geophysics has been widely studied in the past 10 years, especially on processing seismic data.

In the following study, a new method combining ensemble learning and PNN for application is proposed and I would illustrate the study both in algorithm domain and in application perspective.

2.3.1 Total Modeling Scheme

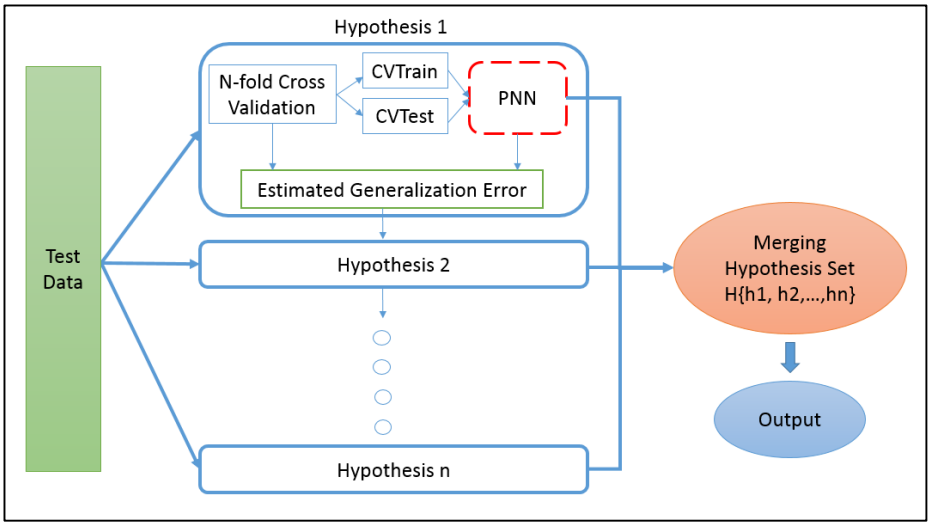

Figure 15. Total Modeling Scheme of TOC Prediction.

The total modeling process for ensemble learning of Probabilistic Neural Network is shown below in Figure 15, detailed structure of PNN will be presented later in the following part. We could notice that compared with the structure of the entire model, PNN only takes up one small step and yet this small unit is the essence of the entire application. The outer structure basically illustrates the scheme for ensemble learning, where several of hypothesis would be generated sequentially and each hypothesis is an expert on certain samples of the feature domain. After generalizing n hypothesis, we would merge the solution for all hypothesis with different weights and forms a final output.

2.3.2 Ensemble Learning

Ensemble Learning is a supervised learning method, the nature of the algorithm is pretty self-explanatory, which is to combine many learning algorithms together to build a stronger algorithm, given a set of all hypothesis, ensemble learning would learn a final

hypothesis from the hypothesis set. In this particular study, the adaptive boosting algorithm (Zhou 2012) is used, it incorporates a way of incrementally finding a final hypothesis by training each new model with emphasis on the misclassified training samples. Although in learning theory the adaptive boosting could be sensitive to outliers and noises, in practice, this method works surprisingly well at any kinds of non-extreme datasets. The scheme of this algorithm is illustrated in Figure 16 below.

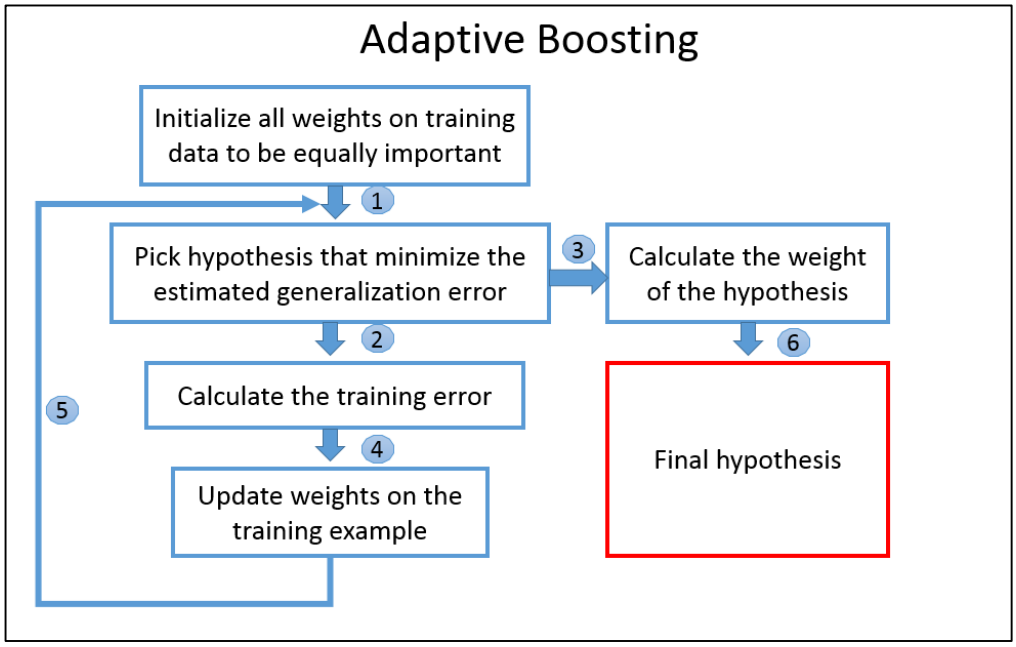

Figure 16. Illustration of Adaptive Boosting Workflow.

As is shown in Figure 16, for ensemble learning there are in total of 6 steps. We are going to illustrate the details for each part.

Step1：Weight initialization

There are two ways of doing the weight initialization, for highly balanced dataset (where the number of training data from all classes are almost equal), we could simply use:

$$w1 = \frac{1}{N_{train}} \tag{29}$$

Where:

w1 is the weight column vector for all training examples.

N_{train} is number of training data

However, for imbalanced dataset with PNN, the weight initialization above would tend to give more credit to the predominant class, in that case, our experimental results showed that applying a local initialization inside each class could effectively avoid this imbalanced class issue, I use

$$\textit{For all i in each class:} \qquad w[class == i] = \frac{1}{Ntrain^i} \tag{30}$$

Where:

w is the weight column vector for training data within class i

N_{train}^{i} is number of training data within class i

Step 2: Misclassified index

After picking the best hypothesis, the predicted class during training by cross validation and real training class will be compared and the training error is expressed as:

$$\textit{For all i in } N_{train}\textit{:} \qquad E[i] = 1 \;\; if \; h(x)[i] = y[i] \tag{31}$$

$$E[i] = -1 \;\; if \; h(x)[i] \neq y[i] \tag{32}$$

Where:

E is misclassified index column vector containing either 1 or -1

h(x) is a vector that is predicted by the hypothesis h

y is the vector that stores training label for all training data.

Step 3: Weight for current hypothesis

We could further calculate the training accuracy of the using equation (33)

$$R = \frac{\sum_{i=1}^{Ntrain} 1\{h(x)[i]==y[i]\}}{Ntrain} \tag{33}$$

$$\alpha^t = \frac{1}{2} ln \frac{R}{1-R} \quad (34)$$

Where:

R is the accuracy rate of the current hypothesis

α^t is the weight scalar for the current hypothesis at time t

Step 4: Weight update for next hypothesis

Next, the weight would be updated for the next classifier so that it could emphasize those misclassified training data.

$$w^{t+1} = w^t e^{-\alpha^t E} \quad (35)$$

Where:

w^{t+1} is the weight vector of all training data for the next hypothesis

w^t is the weight vector of all training data for the current hypothesis

Step 5: Weight normalization

In order to avoid over-fitting, there needs to be a budget of total weight of training examples, most of the time, for simplicity, we determine the sum of all weights for the next hypothesis must be 1.

$$w^{t+1} = \frac{w^{t+1}}{\sum_{i=1}^{Ntrain} w[i]} \quad (36)$$

Step 6: Merging Hypothesis

The final hypothesis could be a linear or non-linear combinations of all hypothesis, in this study, a weighted linear combination of hypothesis sets is applied.

$$H_{final} = \sum_{t=1}^{n} \alpha^t H^t \quad (37)$$

2.3.3 Probabilistic Neural Networks

Generally, there is a misconception existed among engineers and researchers that they usually tend to refer Artificial Neural Network as merely the Multi-Layer Perceptron network (MLP), without knowing the PNN, one could easily presume that PNN and MLP are almost the same. However, the only similarity that PNN and MLP share together is the structure design, which is inspired by biological connection of neurons. In fact, the PNN and MLP (or generally referred as neural network) are two totally different algorithms, the comparison between two algorithms will be presented later. PNN could provide a way to interpret the network structure in a form of probability distribution function. The scheme of PNN is shown in Figure 17 below:

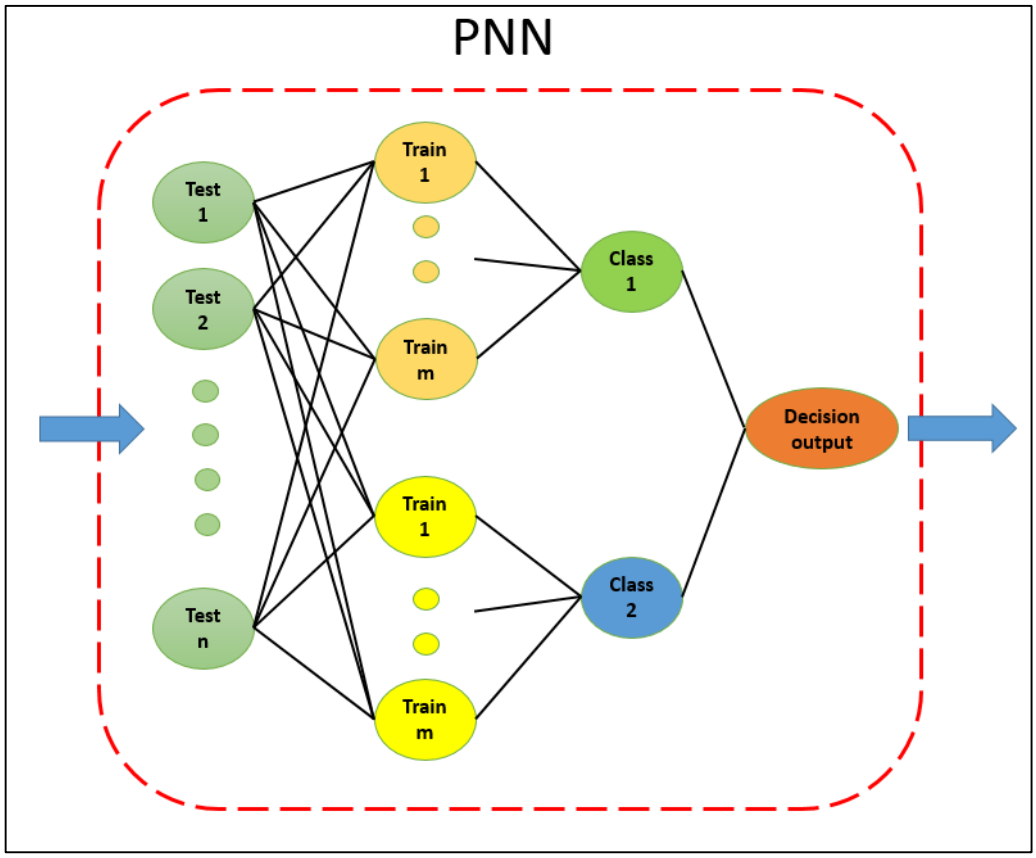

Figure 17. Modeling Scheme of Probabilistic Neural Network.

In a standard PNN, there are 4 layers: input layer, pattern layer, summation layer and output layer.

1. Input layer:

In the input layer, each test data is represented by one neuron, each input is the n-dimensional feature vector where n is the number of attributes. The vector will be later sent to all nodes in pattern layer.

$$\vec{x} = (\mathrm{x1}, \cdots, \mathrm{xn}) \tag{38}$$

2. Pattern layer:

In the pattern layer, each neuron is one training data, which is also n-dimensional vector.

$$\vec{\mu} = (\mu 1, \cdots, \mu n) \tag{39}$$

Then usually we apply Radial Basis Function (RBF) to approximate the probability distribution function $\mathrm{N}(\mu_j^k, \Sigma_j^k)$ for training data j on class k:

$$P_j^k(\vec{x}) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{-\frac{1}{2}}} \exp[-\frac{1}{2}\left(\vec{x} - \overrightarrow{\mu_j^k}\right)^T \left(\Sigma_j^k\right)^{-1} \left(\vec{x} - \overrightarrow{\mu_j^k}\right)] \tag{40}$$

3. Summation layer:

The summation layer calculates the approximation of class probability function by combining all the probability distribution in Equation (40) together. Suppose we have a total of m training examples, then the probability of x belongs to class j is:

$$P^k(\vec{x}) = \sum_{j=1}^{m} w_j^k \, P_j^k(\vec{x}) \tag{41}$$

The w is the weight vector in the Ensemble learning process. One of the main reasons why adaptive boosting is applied with PNN is because unlike many other supervised learning methods, for example, support vector machines (SVM), it is quite

mathematically convenient to achieve emphasizing certain training examples by adjusting the weight vector in PNN.

In practice, sometimes when the number of training data is small, the dataset could be very sparse, then it is very likely that the probability of one test sample for any class could be extremely small (maybe ranges from10^{-50} to 10^{-10}), in order to approximate the real prediction probability, some experimental adjustments need to be made. One possible way of solving this issue is by logarithm scaling the probability and then normalize the results for all classes using equation (42).

$$P^k(\vec{x}) = log10(P^k) \tag{42}$$

4. Output layer:

After calculating the probability of the test data belonging to class k, suppose we have total of c classes the output could be derived in several ways. The simplest way is to find the class with the greatest probability.

$$Output = argmax[P^k(\vec{x})] \quad for\ k = 1 \cdots c \tag{43}$$

The answer could also be represented by certain probability distribution from class 1 to c.

2.3.4 Comparison of Probabilistic Neural Networks with Neural Networks

Advantages:

1. The training process of PNN is much faster than MLP's back propagation.
2. PNN has an inherently parallel structure that allows parallel computing more easily.
3. PNN is less sensitive to outliers and noises than MLP.

4. PNN could generate the prediction probability for each class, which honors uncertainty in the classification task.
5. It could also be modified to do regression as well, by making the final output the expected value of different probable class.

Disadvantages:

1. The resulting model is not as general as MLP. In traditional MLP generalization would only need the structure of the network and associating weights, but in PPN, since it is an instanced based algorithm, training data is needed all the time for generalization unless a contour probability surface with high resolution is stored.
2. PNN requires large memory for both training and testing process since it needs to store either training data or probability surface.
3. Compared with other types of networks, PNN requires a more representative training dataset.

2.3.5 Test Results

During the experiment, I use 70% percent of the total data as training, the other 30% of data is used for testing, and 10-fold cross validation is applied.

In this experiment, I am trying to find out the best number of boosting hypothesis, I change the number of boosting PNN from 1 to 6 and run each scenario 10 times independently, with each time the Training and Testing data randomly selected, I got the following results in Table 2.

# of boosting	1	2	3	4	5	6
Experiment 1	80.22%	76.92%	83.52%	78.02%	78.02%	80.22%
Experiment 2	80.22%	72.53%	81.32%	81.32%	82.42%	79.12%
Experiment 3	68.13%	81.32%	74.73%	84.62%	81.32%	75.82%
Experiment 4	74.73%	80.22%	74.73%	80.22%	82.42%	71.43%
Experiment 5	80.22%	75.82%	73.63%	71.43%	83.52%	75.82%
Experiment 6	76.92%	84.62%	73.63%	81.32%	80.22%	79.12%
Experiment 7	80.22%	76.92%	82.42%	76.92%	82.42%	82.42%
Experiment 8	82.42%	85.71%	72.53%	74.73%	79.12%	76.92%
Experiment 9	80.22%	76.92%	82.42%	80.22%	83.52%	73.63%
Experiment 10	80.22%	84.62%	83.52%	81.32%	86.81%	82.42%
average	78.35%	79.56%	78.24%	79.01%	81.98%	77.69%
stdev	3.96%	4.20%	4.47%	3.63%	2.37%	3.44%

Table 2. Experimental Results of TOC Prediction Using PNN.

The statistical results is shown in the Figure 18 below, average accuracy and standard deviation are normalized and then we could easily pick the number of boosting models with highest accuracy and lowest standard deviation, which is 5.

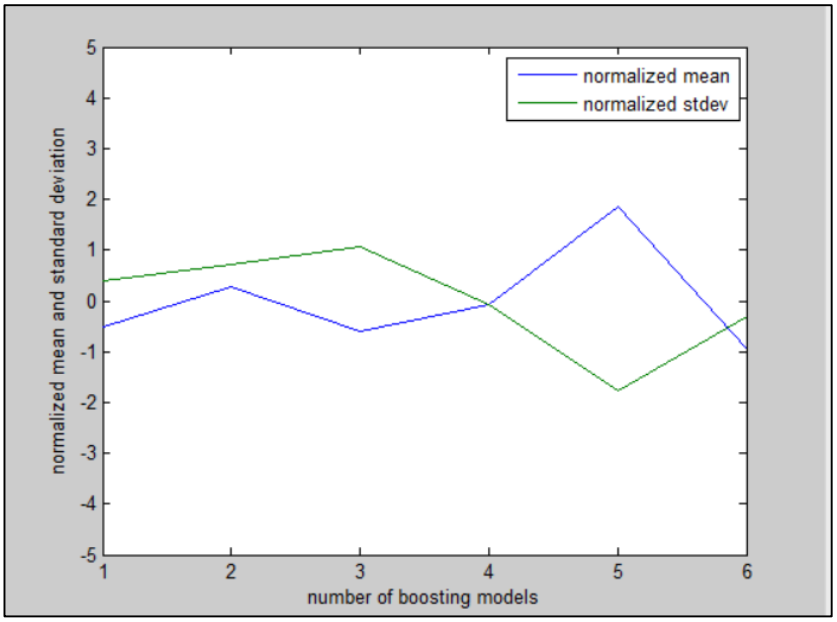

Figure 18. PNN Experiment For Choosing Best Number of Boosting Models (5 boosting models yield the highest accuracy and lowest standard deviation).

2.3.6 Blind Test Results and Interpretation

After determining the optimal number of boosting hypothesis, a blind well experiment is conducted using the optimal boosting hypothesis, and this time it achieved an accuracy of 82.4% for TOC classification. The comparison between TOC data from core measurements and TOC prediction based on well logs is shown in Figure 19.

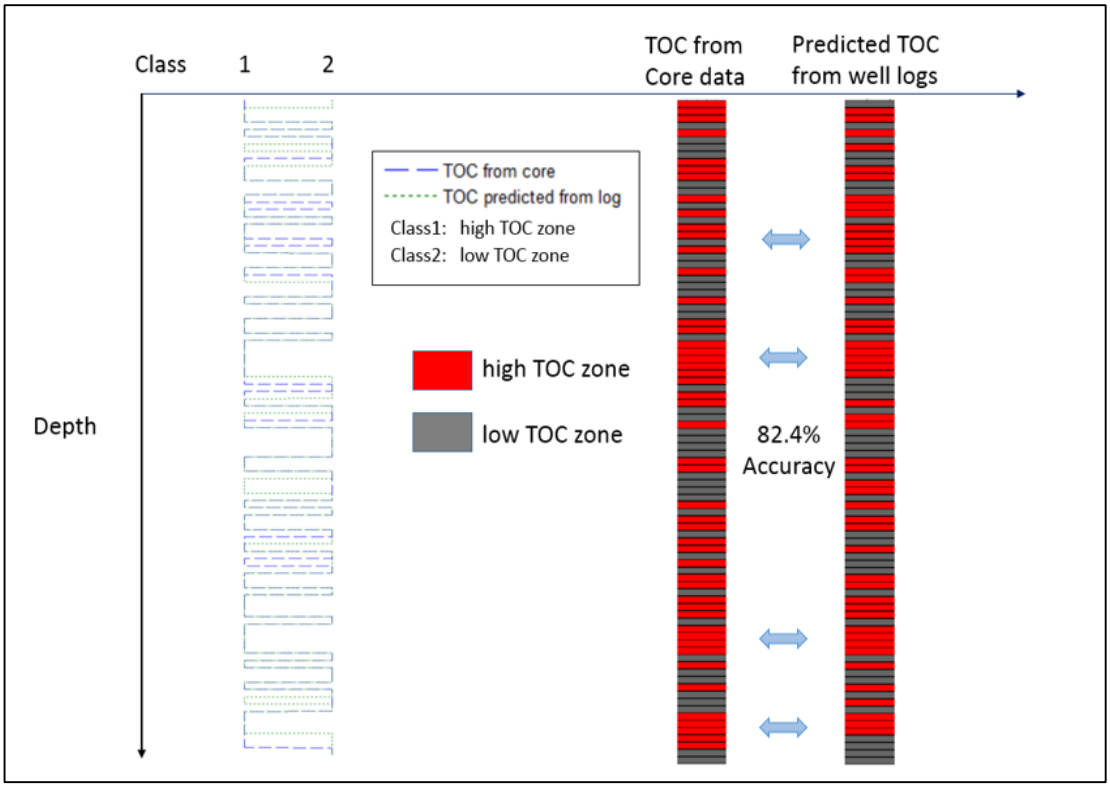

Figure 19. Blind Test Results for TOC Classification (In the left, the blue dashed line is the TOC type distribution from core measurements and the green dotted line represents the TOC prediction based on well log. In the Right, it shows the colored illustration of TOC distribution results acquired from two methods).

In Figure 19, the left side is comparison between two generated TOC Group curves, each color of dashed line represents results from different data source. On the right hand side is the comparison in blind test well in terms of TOC facies, where different color represents different TOC value of rock.

We can see from the blind test results that the predicted TOC zones based on well log using PNN shows a great similarity with core TOC measurement in lab, which indicates that it is possible to automate the process of TOC interpretation based on well log data with a good accuracy. Although there are small discrepancies exited between

predicted results and lab results, if we focus on the potential completion zone, which is represented by continuous red blocks of both results in right hand side of Figure 19 (marked by pointer), we could notice that potential completion zones from well log is almost the same as in core data. Suppose if we could accurately determine the high TOC zones for all uncored wells, it would offer great significance for completion decision and for the creation of reservoir model.

2.4 Chapter Conclusion

In this chapter, I presented two data mining applications to correlate between core and well log in Barnett Shale dataset, first I applied Support Vector Machines to classify lithotypes of rock along the wellbore based on well log data, then I used Ensemble Learnings of Probabilistic Neural Network to predict the total organic content from well log data. Overall speaking, I proved that it is entirely feasible to apply data mining techniques to automate the well log–core correlation process and achieve in a great accuracy. One great advantage for data mining over manual interpretation is that data mining allows us to explore the intrinsic correlation between datasets that human brain could not find, moreover, data mining could automate the correlation process, which could save tremendously amount of time and labor required in the correlation process. The future impact of data mining techniques in this field is limitless.

Chapter 3 Data Mining Approach for Seismic-Constrained SGS Porosity Modeling

In this chapter, I introduced a new workflow for porosity geo-modeling that combined SVM and Sequential Gaussian Simulation. The new approach enables us to discover non-linear relation between seismic attributes and porosity and therefore enhance the porosity modeling process in Mississippi Limestone.

3.1 Chapter Introduction

Traditionally in Geo-modeling process, due to limitations on exploration techniques, the amount of data is very scarce; thus, it is fairly easy to build a model that can honor all the acquired data. However, as time goes by, the advancement of exploratory technology has gradually allowed us to have access to a massive or even an overwhelming amount of data. As a result, it would be quite challenging for traditional way to build a model that could make the full use of the huge amount of data. In order to best utilize most amount of data and build a more accurate model, data mining has gained a great attention over these years. Many data mining applications in seismic processing (Keith et al. 2013; Joseph et al. 2002) have proven that with the help of modern computer's exponentially increasing computing powers, data mining is becoming a crucial step in modern geo-modeling.

One of the most widely used approaches for building a seismic-constrained porosity model is achieved through a combination of SGS and co-kriging. In this process, porosity log observation is the first variable and seismic impedance is usually the second variable. This approach is built on the assumption that the spatial correlation between the

modeling property and seismic impedance is linearly correlated, so that when first variable is scarce, the second variable would be used as a substitute variable. However, seismic impedance is affected by many other factors such as mineralogy, degree of consolidation, pore fluid, pressure and temperature. In most cases, relying on a linear assumption would omit many intrinsic correlations between two variables, sometimes the linear assumption is not even valid.

In this chapter, I aim to apply data mining technique to discover not only linear relations, but also non-linear patterns between seismic impedance and porisity. Then by making use of those patterns, we could build a more constrained model. The layout of the chapter begins with an elaborate discussion about the workflow and methodology. Then for experiment results, I would first use a synthetic datasets to demonstrate how our workflow can be effective in horizontal scale. Next I use real dataset to demonstrate the results in vertical scale. Finally I am going use a simple experiment to explain why data mining techniques have advantages over traditional geostatistics in the data assimilation process.

3.2 Chapter Objectives and Dataset Description

The main theme of reservoir modeling is to build a model that could honor as much valuable data as possible. In porosity modeling, one of the most widely used geo-statistical approach to honor both well log data and seismic data is achieved by Sequential Gaussian Simulation (SGS) and co-kriging. This approach has been proved useful in most geological settings; it is built on the assumption that the spatial correlation between first variable and second variable is linearly correlated. However, in reality, the linear behavior

does not hold true for all reservoirs, and relying on a linear assumption might omit many intrinsic correlations between two variables, which may affect the accuracy of resulting model. In this chapter, I present a new workflow that applies data mining techniques to mitigate this issue. Instead of relying on linear assumption, the new approach enables us to discover non-linear pattern between porosity and seismic attributes.

Data in this chapter comes from the Mississippian reservoir in north-central Anadarko shelf in Oklahoma, USA (Figure 20). The Mississippian carbonates in the area were deposited at the basinal edge of the Anadarko shelf margin during four showing20-upward sequences. Following the end of the Mississippian, the region was subjected to uplift and sub-aerial exposure, resulting in a widespread erosion and a production heterogeneity (Lindzey 2015). The thickness for Mississippian ranges from 350 ft to the south to as little as 100 ft to the north where the rocks have been subjected to greater erosion during the pre-Pennsylvanian unconformity (Gutschick and Sandberg 1983; Roger 2001). Data for this study include a full suite of open-hole logs from 31 vertical wells 70 mi^2 of seismic inversion data. The seismic data includes P-impedance, S-impedance, impedance ratio, $\lambda\rho$ and $\mu\rho$. In the attributes above, $\lambda\rho$ and $\mu\rho$ are elastic parameter of rocks, λ and μ are lame's first parameter and lame's second parameter.

In this work, I make one significant assumption. I assume that the water saturation levels and fracture/crack density remains the same throughout the area of study. This assumption is being made because the seismic attributes such as S-impedance, P-impedance, $\lambda\rho$ and $\mu\rho$ are impacted by water saturation and crack/fracture density variations (Omoboya et al. 2012).

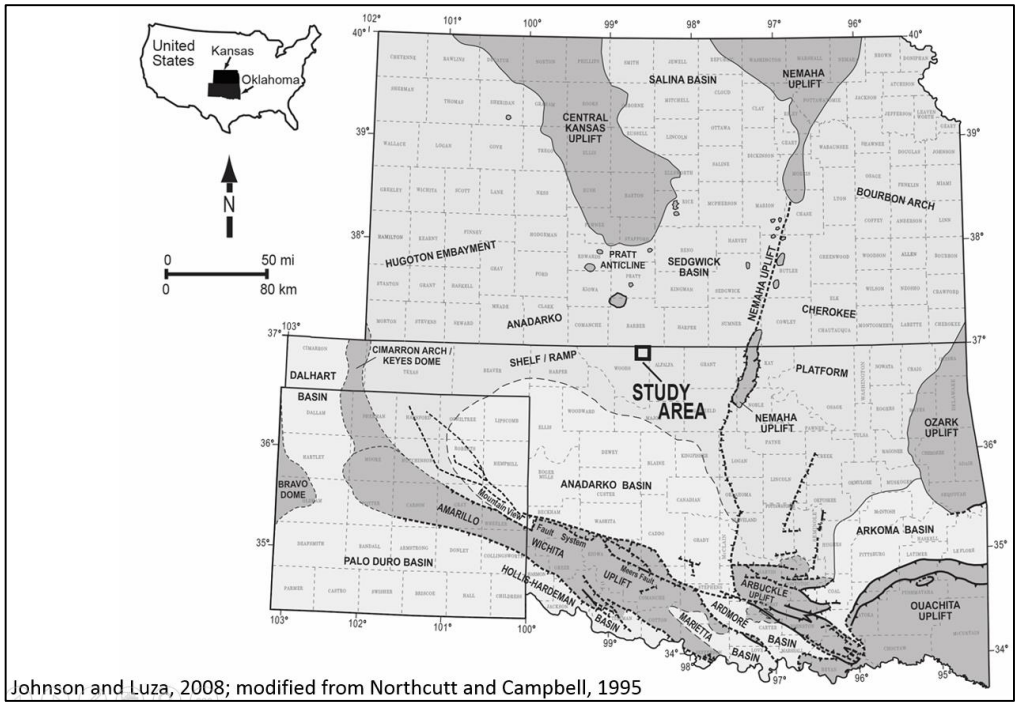

Figure 20. Regional Geology of Study Area (Johnson and Luza 2008).

3.3 Methodology

The total work flow for applying data mining techniques to build seismic-constrained porosity model is shown in Figure 21 below. In general, there are in total six major steps in the process:

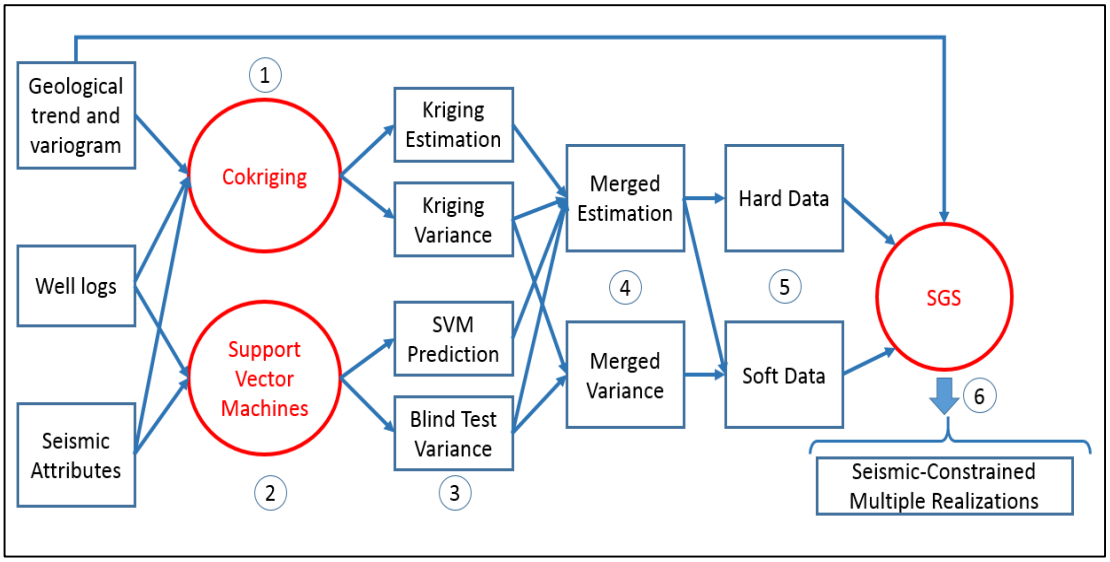

Figure 21. Total Workflow of Constructing Constrained SGS Model.

1. Use of co-kriging to estimate the porosity and variance at all location. (Doyen and Boer 1996; Xu and Tran 1992)
2. Application of Support Vector Machines (SVM) to find the pattern between seismic attributes and porosity, then predict porosity based on seismic attributes in the model.
3. Evaluation of the SVM model uncertainty.
4. Merging results from step1 and step 2.
5. From the merged map and merged variance map, create soft data in the datasets.
6. Run multiple SGS (Isaaks and Srivastava 1989) realization conditioned to both hard data and soft data to get the final constrained SGS model.

Among these steps, since both co-kriging and SGS are well-established approach in geomodeling, the basic workflow for them is not going to be incorporated in this chapter. Instead, we are going to heavily focus on step 2 to step 5.

3.3.1 Support Vector Machines Porosity Prediction Based on Seismic Attributes

Given the well log data and seismic attributes, there is one problem that needs to be solved in order to move forward: well log data and seismic data are in different resolution in both vertical scale and horizontal scale. Therefore, in order to address this issue, I further upscale the porosity log vertically to be the same as seismic vertical resolution using average method. Next, I apply tri-linear interpolation on seismic attributes to interpolate the seismic attributes at each wellbore location.

As a result, for any given porosity log observation coordinates, I am able to find its corresponding seismic attributes. These data would be used later for the data mining task. The seismic attributes that I use are: P-impedance, S-impedance, impedance ratio, $\lambda\rho$, $\mu\rho$. $\lambda\rho$ and $\mu\rho$ are elastic parameter of rocks, λ and μ are lame's first parameter and lame's second parameter. In fact, $\lambda\rho, \mu\rho$ could be represented by p-impedance and s-impedance. Including those two elastic parameters is convenient for the algorithm to explore non-linearity. The data mining target is porosity.

Intuitively, since the target (porosity) is a continuous value, it seems quite obvious to us that the best way to correlate seismic attributes and porosity is by regression. However, I tried and it did not work well. The main reason why regression failed is that the data is extremely noisy, using regression would most likely lead to over-fitting. To be more specific, the porosity data comes from upscaling nearly 15 feet of density porosity log, put aside noises from log measurements, the porosity value after averaging 15 feet of log could only provide us with a low-resolution porosity indicator. Moreover, the seismic attributes I use is even noisier than porosity; those attributes come from many times of seismic processing, no matter automatically or manually, many of the attributes

are derived under physical assumptions which might not perfectly hold. In regression, the mostly used objective function is to minimize the squared error of prediction and true target, the noise in the dataset could easily overwhelm the regression algorithm and cause over-fitting issues.

Compared with regression, classification is more robust to noisy datasets. In classification task, the target is divided into several labels and it would predict the label instead of the real value. In this way, an outlier and regular data is being treated the same as long as they are in the same label, which could effectively avoids over-reaction to noises. Having known that, we apply k-means method to make our continuous porosity log into 5 discrete labels, during the process, we are transforming regression problem into a classification task. Support Vector Machines (SVM) would be a good classification algorithm to deal with this task, and it is one of the most efficient classification algorithms. The mathematical details of the algorithm is illustrated in chapter 2.2.

3.3.2 Evaluate the Uncertainty in SVM Models

After SVM makes its prediction on porosity, cross validation is being used to evaluate the uncertainty in SVM model. The steps for cross validation in our case is described as followed:

Repeat the following two steps until all wells are being chosen as validation wells:

1) Separate the blind test wells from available datasets, and train the SVM algorithm on data from remaining wells.
2) Use the model from step 1 to predict porosity in blind test well, and compare the results, calculate the rooted mean squared error.

Following the steps above, we are generating both SVM model and test result at chosen wellbore location for each cross validation. When all wellbore location have been chosen, cross validation is finished, then we are going to have a collection of SVM models and test results at each wellbore location. Next, among the collection of SVM models, we are going to make porosity prediction based on all models. At last, the final SVM prediction is the majority prediction of all SVM models.

In order to move forward, we need to make a choice about creating a SVM variance model for the entire volume. If the variance value for all wells are fairly similar, it indicates that the pattern between seismic attributes and porosity in all places tends to be consistent throughout the reservoir, then for simplicity, it is reasonable for us to assume that the variance for the entire volume is uniformly constant, therefore it would be safe to assign the average variance to all places. However, if the variance for different blind test wells varies significantly, then we need to rely on kriging or other geostatistic algorithm to estimate the SVM variance for all places.

3.3.3 Merged Estimation and Variance Model from SVM and Kriging

So far we have created two porosity models and two variance models using two distinct approaches. The next thing we need to deal with is how to merge these two models in one model. In order to better merge our model, we need to know the advantages and disadvantages of these two models:

Model 1 is the co-kriging model conditioned to porosity logs, impedance and variograms. The advantage is:

1. The model is in fine scale.

2. The model is reliable in areas near wellbore because of the use of variogram.
3. Due to the nature of kriging, it ensures us the locally minimal variance.

The disadvantage for model 1 is:

1. It tends to smooth out geological feature.
2. The information used in the data is limited due to its linear assumption between porosity and seismic attributes. The number of attributes that is used is also limited.
3. In areas far away from wellbore where the porosiy is absent, it provides us only a conservative estimation.

Model 2 is the SVM predicted porosity facies based on P-impedance, S-impedance, relative impedance, $\lambda\rho$, $\mu\rho$. The advantage is:

1. It can make more use of data by using multiple features.
2. It is able to find non-linear pattern between feature and target.
3. It is free of geo-statistical assumption.
4. It could provide us with a reliable porosity prediction for places far away from any well log observation, because porosity prediction totally comes from investigating the pattern between seismic attributes and porosity as long as the quality for seismic attributes are consistent, the prediction performance could hold.

The disadvantages for model 2 are:

1. The resulting model is a coarse scale model, the resolution is the same as seismic attributes samples.

2. The variance model of SVM might rely on unrealistic assumptions.
3. The prediction results is less reliable than SGS for areas near wellbore since it is only making use of seismic data.

Therefore, intuitively, we want our final merged model to combine the advantages of both models; for areas close to wellbore, the final model should mainly trust the model that is conditioned to well log observations (as is provided by model 1). On the other hand, for areas that are distant from any well log observations, the final model should mainly trust the model that is based on seismic attributes (as is provided by model 2).

Not surprisingly, the information needed to achieve the above goal is stored in two variance models. Since we have the variance model for both kriging estimation and SVM prediction, the merged step is fairly straightforward: the merged variance for a given location is going to be the smaller one between two variance model, and the merged estimation for that location would come from the model that gives us a smaller variance for that location.

Compared with Kriging alone, the merged estimate of the porosity field utilizes information contained in seismic and well log data. However, the merged model lacks geologic realism. In this thesis, I have not utilized indicator kriging to model facies variations and then to model porosity variations. It is likely that the use of a prior step of facies modeling will result in more realistic models; however, this is beyond the scope of the work in this thesis.

3.3.4 Creating Soft Data for SGS Modeling

One thing to be noted is that the merged model itself is not a real geomodel, it is more of a data source that contains both direct observation on wellbore location and our best prediction between wellbores.

Therefore, in order to create a qualified geological model that could make use of information in merged model and honor the spatial trend/variogram and histogram as well as account for uncertainties, I turn to SGS again. But this time, I made some adjustments in the SGS workflow to enable SGS to condition to some soft data as well.

The soft data is the porosity estimation in the merged model that is between wellbore locations, it is associated with some uncertainty. Here, we are assuming the soft data for any given location is follows a Gaussian probability distribution function with a mean from merged estimation model and a certain standard deviation from merged variance model.

The procedure is as followed:

1. Randomly select N locations that are reasonably away from any hard data.
2. For each N observation:
    a) Get the corresponding mean and standard deviation from merged model and merged porosity model.
    b) Draw at random a value from the normal distribution defined in step a)
    c) Assign that value to the observation
3. Add N observation data along with hard data in the SGS conditioned dataset.
4. Run one realization of SGS conditioned to the data in step3.
5. Repeat step 1-4 until the number of realization is enough

Another thing to be noted is that in order for our constraints to be effective, the number of soft data N should not be less than the amount of hard data, one rule of thumb is to make N equals the number of hard data. Next, in order avoid the bias, the number of realization needs to be sufficient enough. In our case study, I am arbitrarily setting the number of realization is 100. However, the number of soft data and number of realization required to ensure statistical stability in our workflow still needs to be further investigated in the future.

3.4 Experimental Results and Analysis

In order to illustrate how our workflow for modeling porosity could effectively constrain SGS model and help make a better estimation of porosity, in this section I am going to show the results from both horizontal and vertical perspectives. For Horizontal illustration, since horizontal data in reservoir is very scarce, I am going to use synthetic data for illustration without loss of generalization. For vertical illustration, real dataset from the study area would be used. At last, I am going to use a small experiment to demonstrate why data mining techniques have advantages over geostatistics in the process of data assimilation.

3.4.1 Horizontal Illustration Using Synthetic Dataset

First I generate a 2-dimentional grid with 180 by 180 resolution, each cell has length of 400 ft. Then in the grid I create a certain distribution of porosity that follows a Gaussian variogram with minimum range,median range and maximum range to be equally 8000 ft. (Figure 22), at last I randomly generate 31 well locations onto our grid,

they are represented by black circles. We would treat this porosity distribution as the "true distribution" that we are trying to approximate.

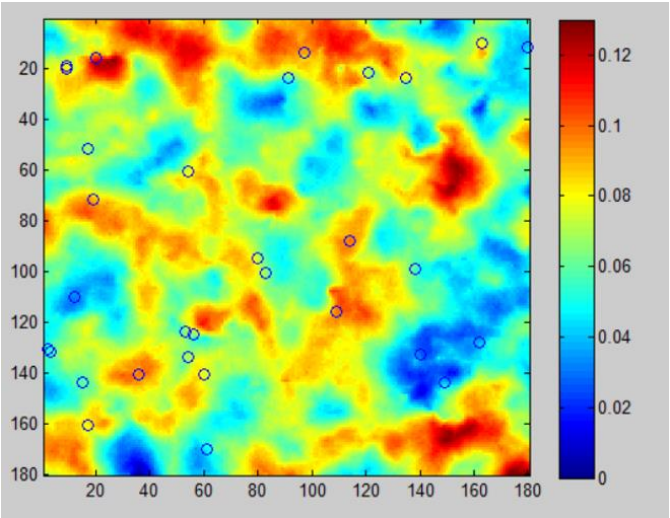

Figure 22. Synthetic True Porosity Distribution.

In order to generate seismic attributes, we need a coarse scale porosity map, we assume the seismic attributes has a horizontal resolution of 3 by 3 grid. The coarse scale porosity map is shown in Figure 23.

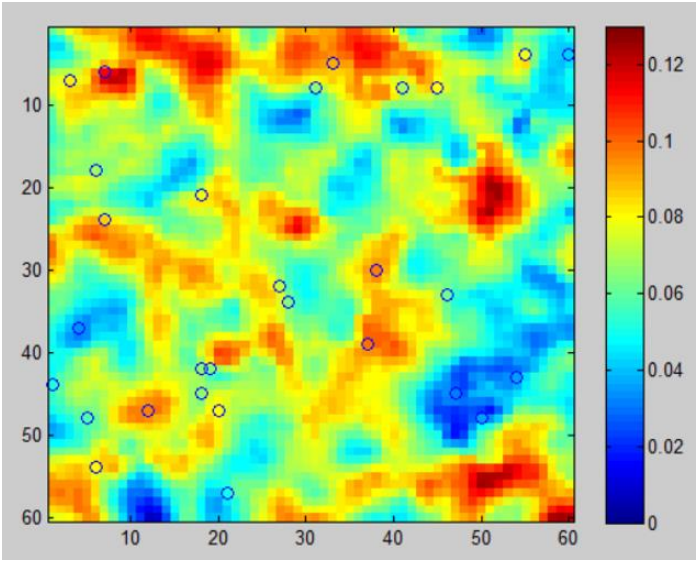

Figure 23. Coarse Scale Porosity Distribution.

Next, we are going to generate seismic attributes from coarse scale porosity map using following procedure:

1. From the real dataset at wellbore location, I extract seismic attributes and its corresponding porosity value at different depth and for different wells. Each set of attributes and its corresponding porosity is called “sample”.
2. Collect all samples together, and sort it according to porosity value.
3. For any given location, find the coarse scale porosity value, then match the same porosity value in real dataset, next randomly select one set of corresponding seismic attributes for that location
4. Repeat step 3 until all locations has a corresponding seismic attributes set.

Therefore, our available dataset includes: porosity observation at wellbore location, also seismic attributes map that corresponds to the same porosity value from real dataset, the seismic map covers the entire area. The cross plot where seismic attributes intersect with porosity observation at wellbore is shown in Figure 24 to Figure 28.

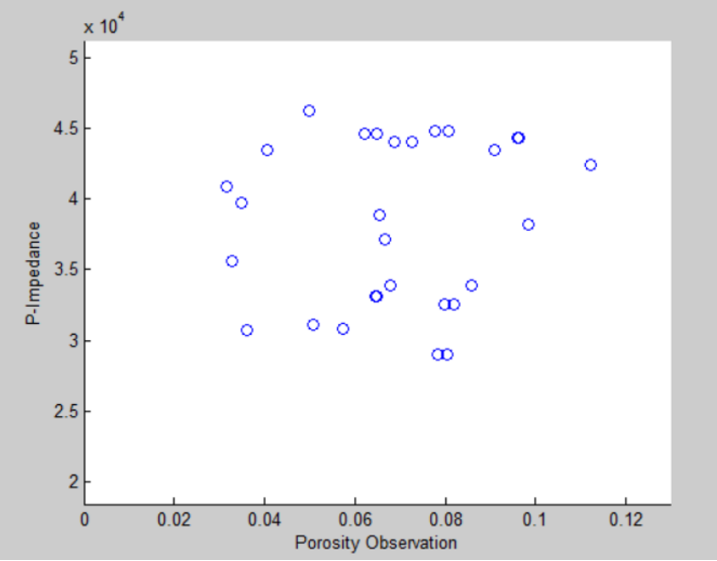

Figure 24. Cross Plot of Porosity vs. P-impedance at Wellbore.

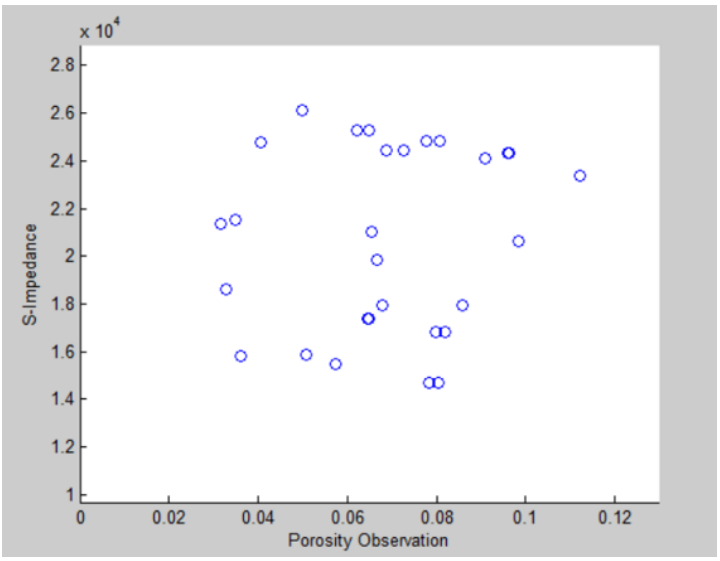

Figure 25. Cross Plot of Porosity vs. S-impedance at Wellbore.

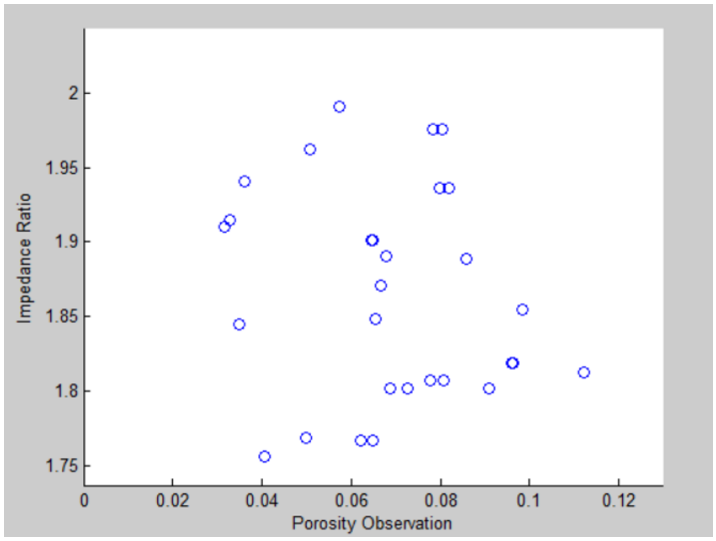

Figure 26. Cross Plot of Porosity vs. Impedance Ratio at Wellbore.

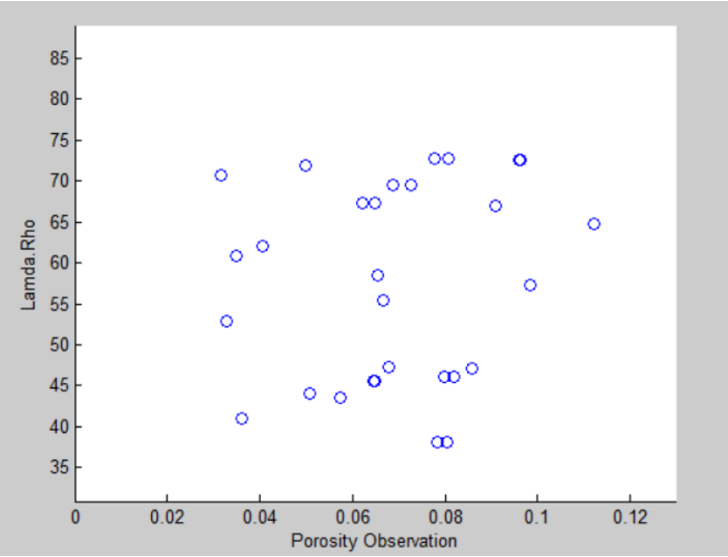

Figure 27. Cross Plot of Porosity vs. Lamda*rho at Wellbore.

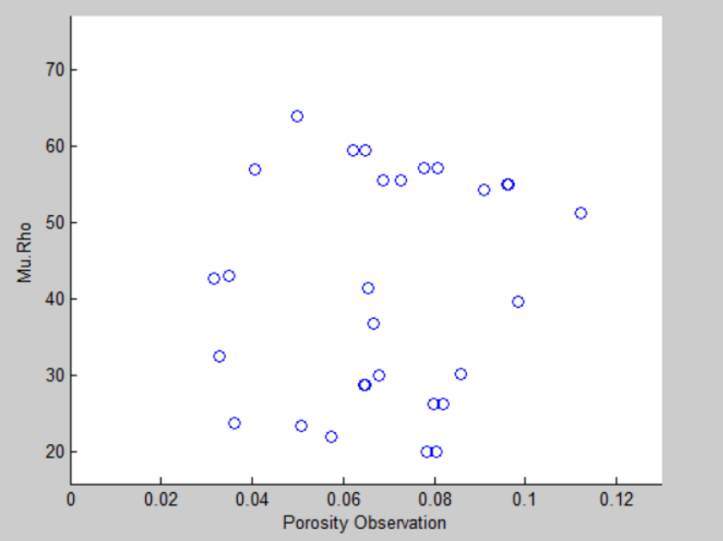

Figure 28. Cross Plot of Porosity vs. Mu.rho at Wellbore.

Having look at the relation between any seismic attributes and porosity in observation data, it is probably not good idea to use co-kriging since the linear assumption does not hold at all. Therefore, for step 1, given observations at wellbore and same variogram inputs, I use kriging to create the porosity estimation (Figure 29) and standard deviation estimation (Figure 30).

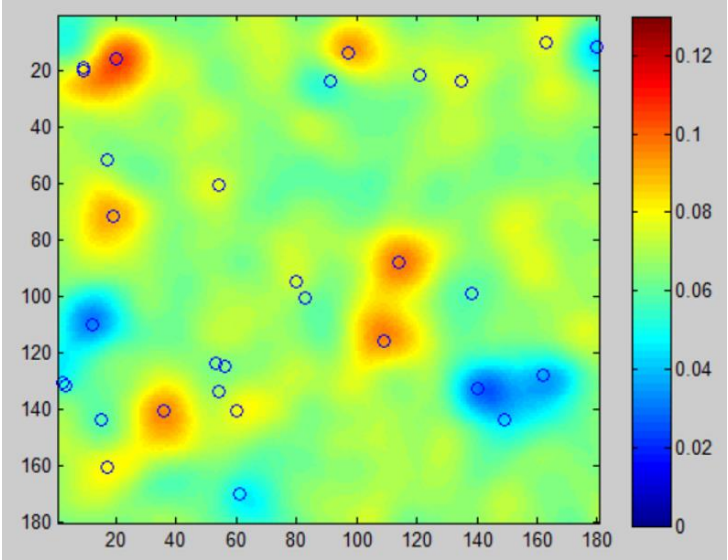

Figure 29. Kriging Estimation Conditioned to Wellbore Observation.

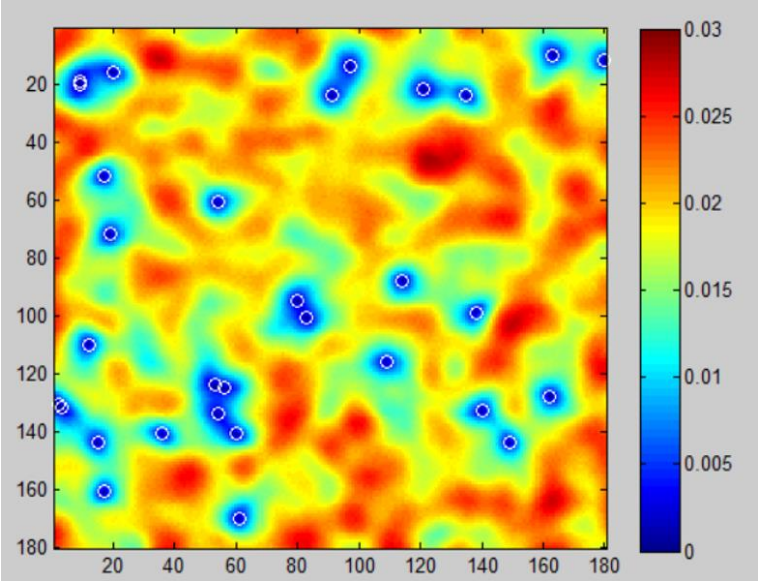

Figure 30. Kriging Standard Deviation.

Then following the methodology in step 2 and step 3, I extract the seismic attributes and porosity value at observation location, I run SVM algorithm 10 times independently, and for each time I randomly select 70% of data for training and 30% as testing data, inside the training process, leave-one-out cross validation is applied.

Next, I make prediction on porosity based on seismic attributes in entire map for all models. Then combining all models together by making a vote and take the majority of prediction. The final SVM prediction model is shown in Figure 31. The SVM porosity standard deviation ranges from 0.0169 to 0.0224 with an average of 0.0189. Since it is not varied greatly for all test, also for the case of simplicity, we are assuming the standard deviation for SVM prediction throughout the reservoir is constant.

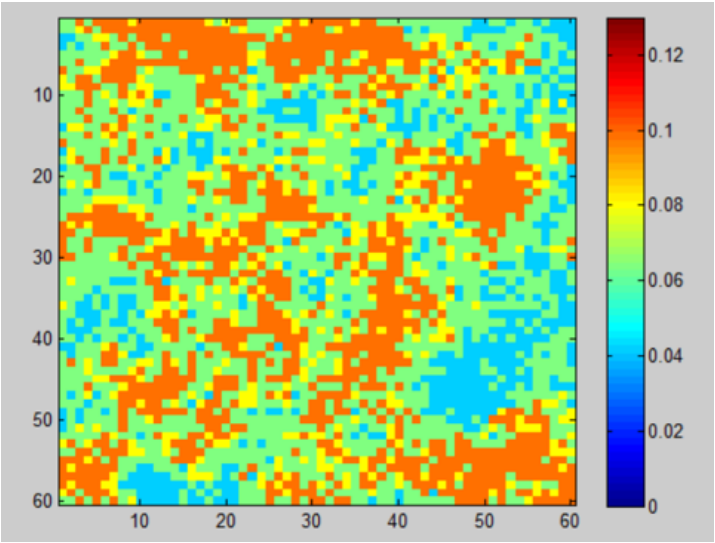

Figure 31. SVM Porosity Prediction Based on Seismic Attributes.

Next, following step 4, I merged two estimation and standard deviation map together, the merged map are shown in Figure 32 and the merged standard deviation map is in Figure 33.

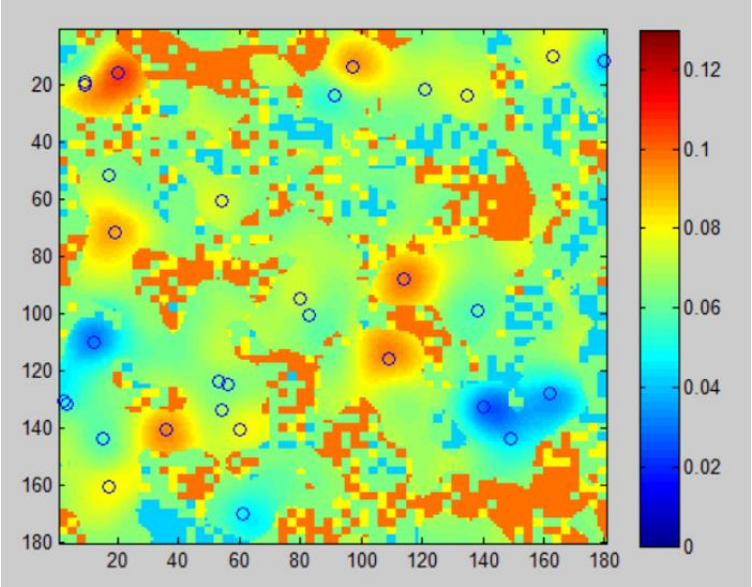

Figure 32. Merged Porosity Estimation Map.

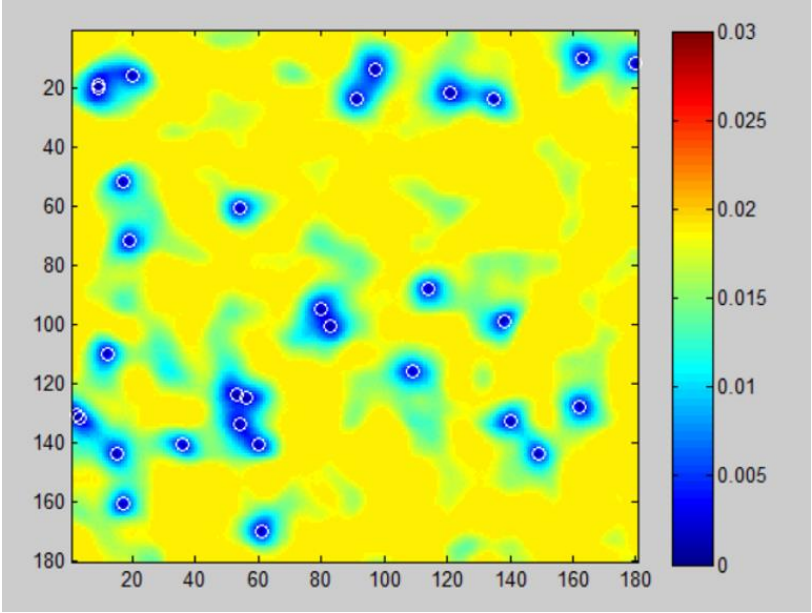

Figure 33. Merged Standard Deviation Map.

Next, following the scheme in step 5, I run the modified SGS algorithm and generate 100 realizations. For the purpose of comparison, we are going to present some realizations without seismic constraints first in Figure 34. Results for modified SGS using our workflow is shown in Figure 35.

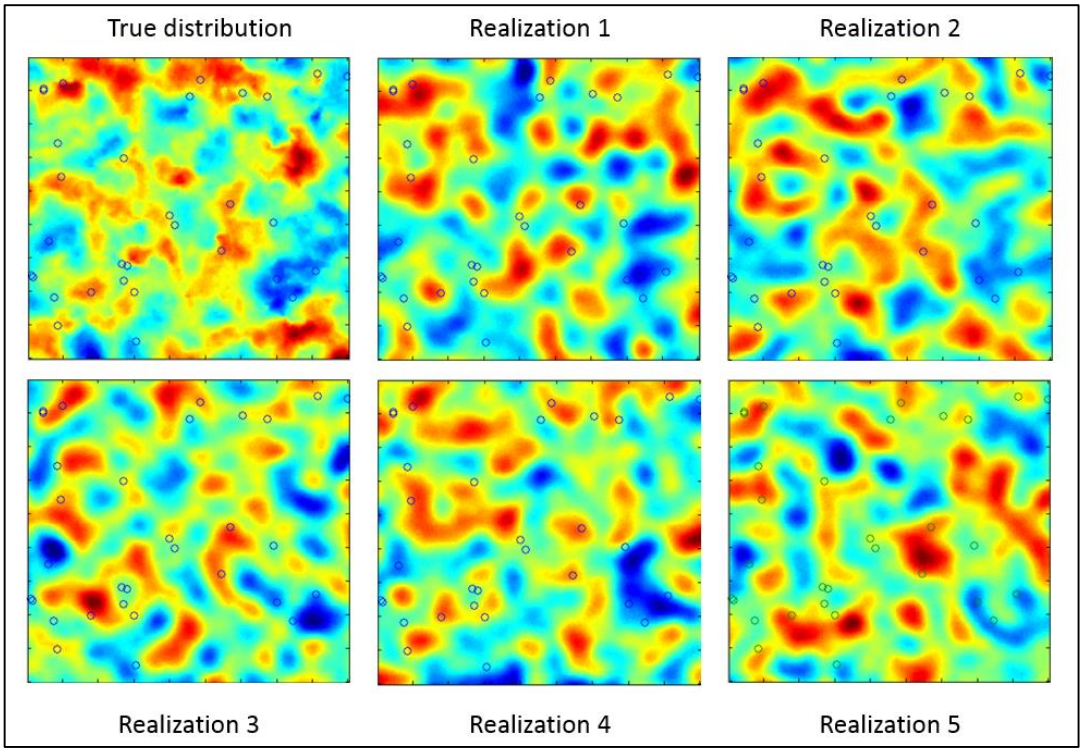

Figure 34. Regular SGS Modeling Realization.

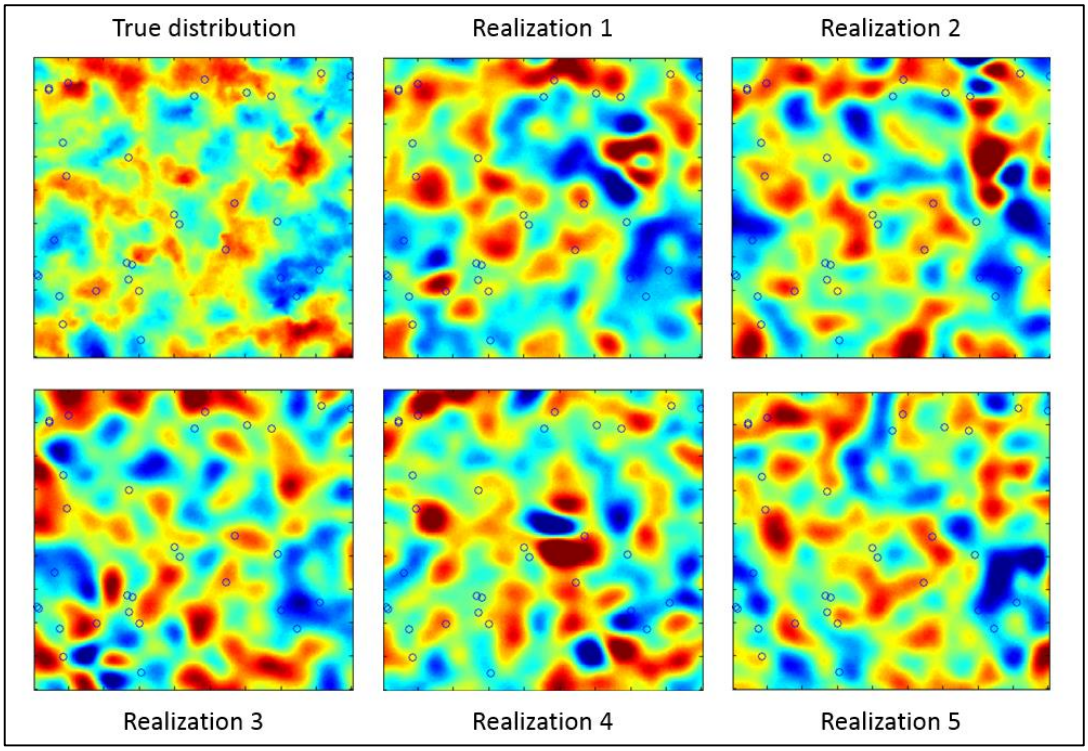

Figure 35. Modified SGS Modeling Using New Workflow.

From the comparison between unstrained SGS model and unconstrained SGS model, we could see that because of making use of intrinsic pattern between seismic attributes and porosity, constrained SGS model is more likely to be able to predict the high porosity area for places where no nearby well log data is available, and the model using our workflow is more likely to reproduce high porosity area in the true map.

Next, we are going to compare the e-type map of regular SGS and modified SGS for all realizations. As is shown in Figure 36, we could clearly see from the cross plot that the average model for from our workflow not only yields a smaller rooted mean squared error, but also show a more detailed and more accurate reproduction of porosity spatial distribution. These "extra information" is made possible by the data mining techniques, which indicates that applying data mining could effectively increase the efficiency for data utilization and help us build a more accurate distribution model.

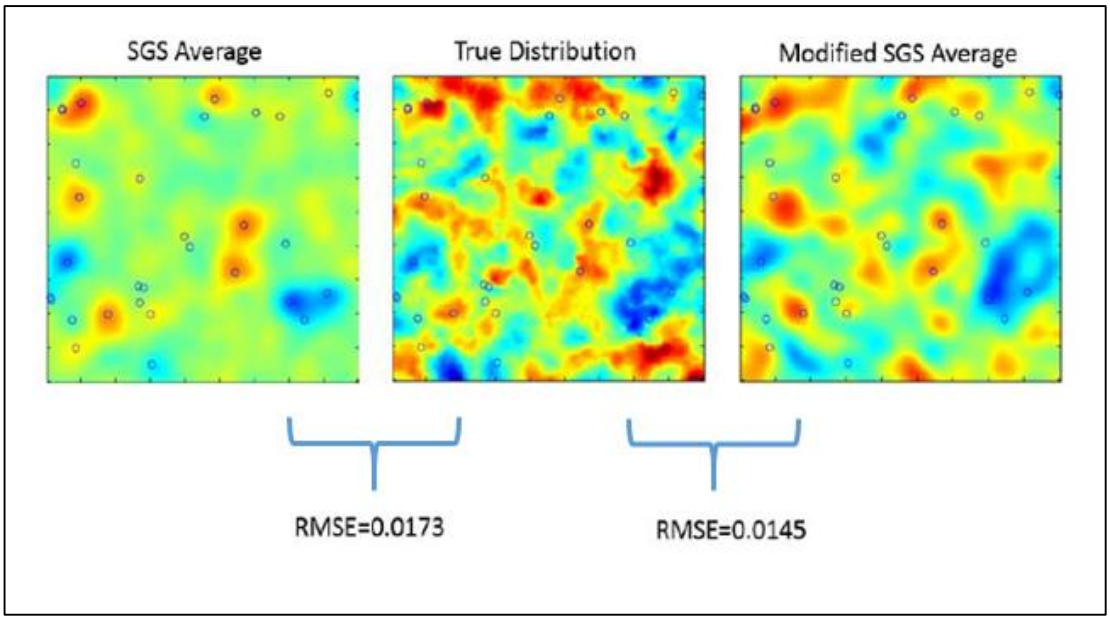

Figure 36. Comparison of E-type Map for Both SGS and Modified SGS.

3.4.2 Vertical Illustration Using Real Dataset

This section shows results from different vertical scales of reservoir (Figure 37). In Figure 37, test result 1 is the evaluation of SVM algorithm predictions, it includes a comparison between predicted porosity based on seismic attributes data and porosity from well log data. Then after building the uncertainty map and the final constrained model, test result 2 would be used to show the comparison between modified SGS model, regular SGS model and how close both models are from well logs, which could illustrate the significance of our model.

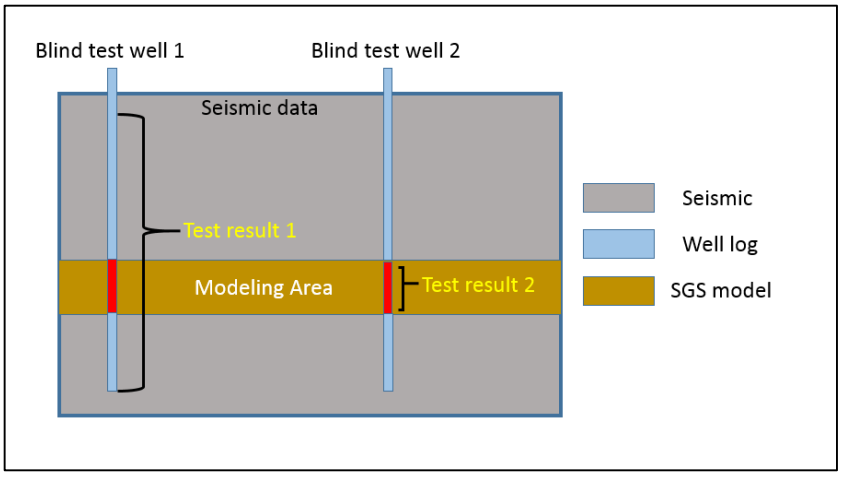

Figure 37. Illustration of Scales of Two Test Results.

3.4.2.1 Results from Support Vector Machines Porosity Prediction (Test Result 1)

Given the porosity log for all wells. I am using k-means to discretizing porosity log into 5 different porosity facies. And each facies is represented by an average porosity value Next, I match each porosity facies with corresponding seismic attributes using trilinear interpolation method. Finally, I apply Support Vector Machines to handle the classification task. In our experiment, a total amount of 4000 observation data is

available, I take data from 3 blind wells out from our observation data, then use the others to build our algorithm. The blind test result for 3 wells are shown in Figure 38.

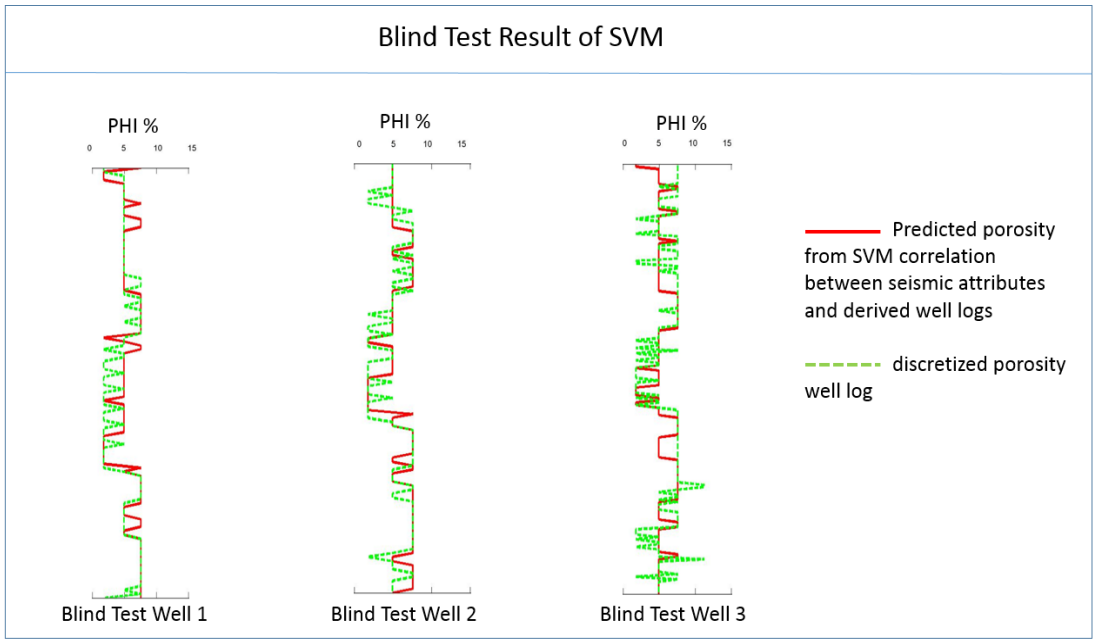

Figure 38. Test Results for Support Vector Machines Prediction.

From the test results, we could see that SVM is able to broadly predict the porosity behavior on vertical scale. Since the seismic data itself is noisy and the pattern between seismic attributes and porosity is not universal, there are still many discrepancies between SVM prediction and down-hole porosity measurements. However, comparing with regular kriging method, these rough prediction is good enough to be the source of "extra information".

3.4.2.2 SGS Model Validation Results (Test Result 2)

In order to illustrate how SVM could be useful in building a better constrained porosity model on a vertical scale, I applied three independent leave-one-out tests. For each test, I follow the procedure below:

1) Randomly select one well and delete its well log data from available dataset.
2) Build a porosity model using SGS conditioned to available porosity log and p-impedance.
3) Following the workflow in this chapter, build another porosity model using SVM and modified SGS.
4) For the selected well, calculate the root mean square error between porosity observations and two SGS models in step 2 and 3.

The test results for three different wells are in Figure 39, in this test, density porosity log is being used as porosity observation.

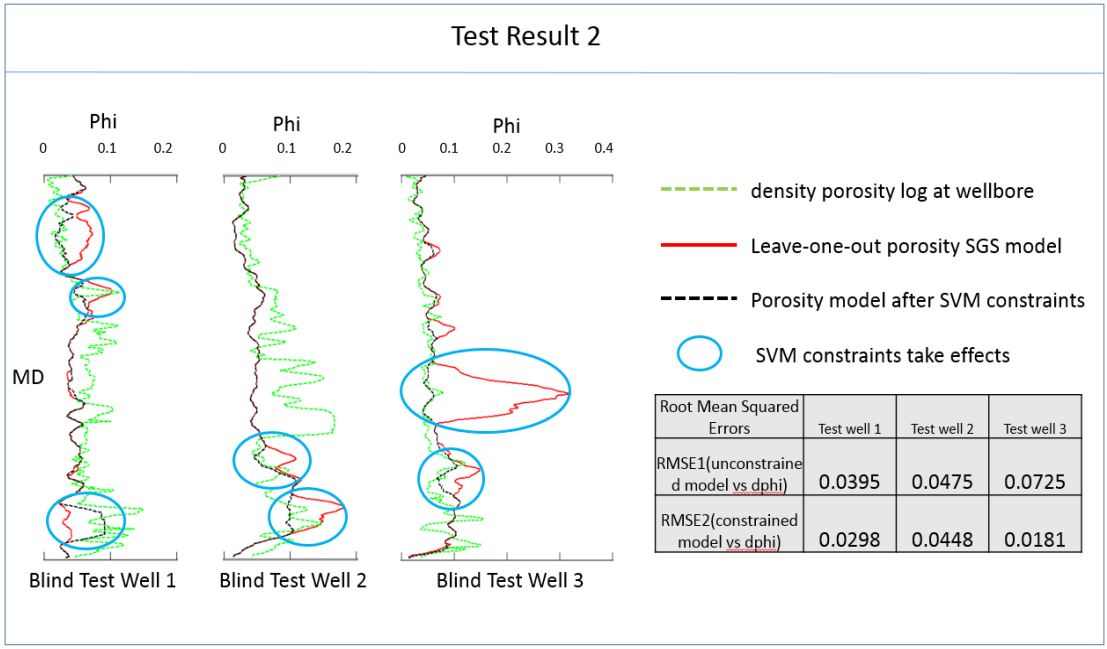

Figure 39. Illustration of Test Results 2.

In Figure 39, we could notice that for different test wells, the all models has different performance. Although there are many major discrepancies existed between constrained model and real porosity log for all blind test wells, applying SVM constraints to SGS model could effectively tune the SGS model towards real porosity down-hole

measurements and decrease the error. This is because SVM constraints could enable us to discover the intrinsic non-linear pattern between porosity and seismic attributes and making use of the pattern and give more guidance for SGS modeling.

Now we might wonder, how could SVM algorithm could more efficiently dig out extra in a seemingly chaotic dataset than geostatistics? In order to explain its advantage, I am going to conduct a simple experiment in the next section.

3.4.3 Simple Experiment on Advantages of SVM

In order to approximate the real case, we assume the porosity is a continuous function of 3 factors. For the purpose of illustration, we arbitrarily create one porosity function without consideration of physical meanings:

$$\Phi = x1^2 + x2^2 + \ln(x3) \tag{44}$$

In equation (44), Φ is porosity, x1 could be overburden pressure, x2 could be some mineralogy measurements, x3 could be degree of consolidation. Let us assume the equation (44) is the true equation that we do not know.

Fortunately, we have some attributes that contains some degree of information about these factors. These attributes are also a continuous function. In our case, we arbitrarily create three attributes without physical consideration:

$$\text{Pimpedance} = \text{x1} + e^{x2} - x3^2 \tag{45}$$

$$\text{Simpedance} = \ln(\text{x1}) - x2^2 + x3x1 \tag{46}$$

$$\text{Elastic Modulus} = x1^2 + x2 - x2x3 \tag{47}$$

Then I generate 100 sets of x1, x2 and x3 from uniform random distribution scaled from 0 to 1. Next, I use the equation (44) to equation (37) to generate Φ , P-impedance,

S-impedance, Elastic Modulus. Now, I erase equation (44) to equation (47) from our mind, what is available to us is only the data, and let us pretend we only know two things:

1. We know that porosity is affected by some factor x1, x2, x3 to some degree.
2. We know that the attributes we use also contains some degree of information about x1, x2 and x3 as well.

If we cross plot every attributes vs. porosity, as shown in Figure 40, Figure 41 and Figure 42, we could see that our dataset is extremely noisy and even lack of any pattern. This situation is very much alike when we are trying to correlate between seismic attributes and porosity in real case.

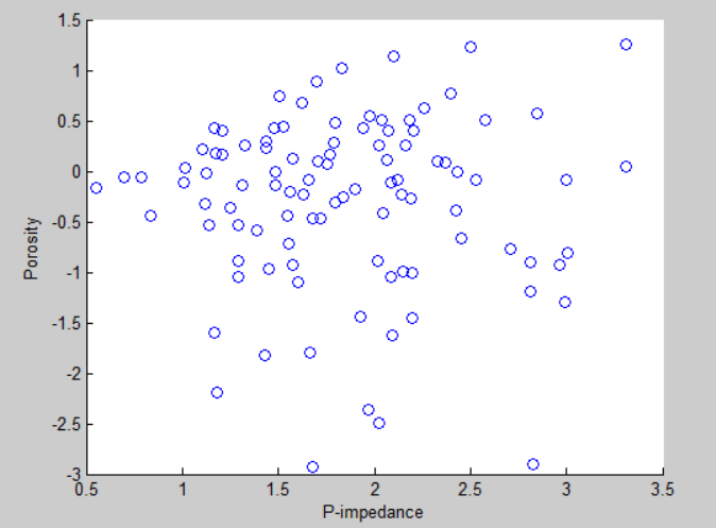

Figure 40. Cross Plot of P-impedance vs. Porosity

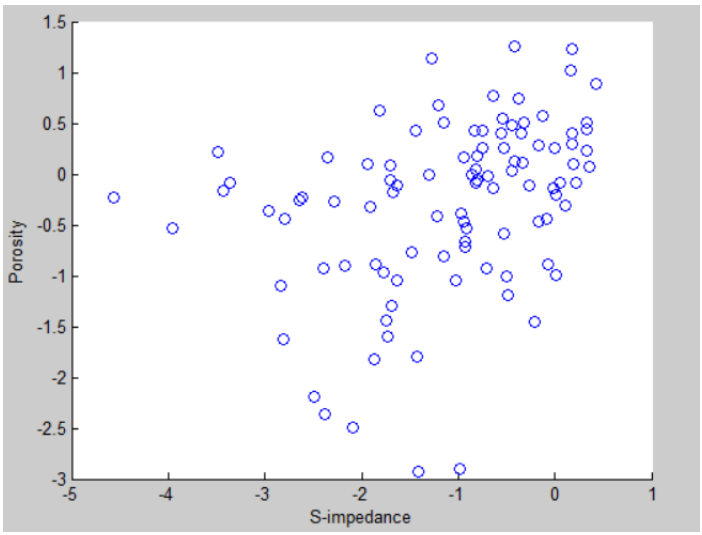

Figure 41. Cross Plot of S-impedance vs. Porosity

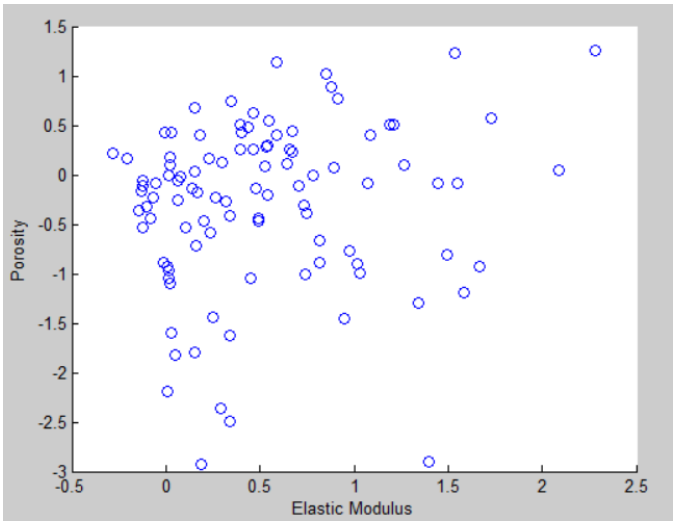

Figure 42. Cross Plot of Elastic Modulus vs. Porosity

Given these data, I randomly select 70% of data as observation, and applied a least squared linear regression, which has the same principle as co-kriging, then I test the correlation on the other 30% test dataset, the result is shown in Figure 43, the rooted

mean squared error between prediction and true value is 0.7924.

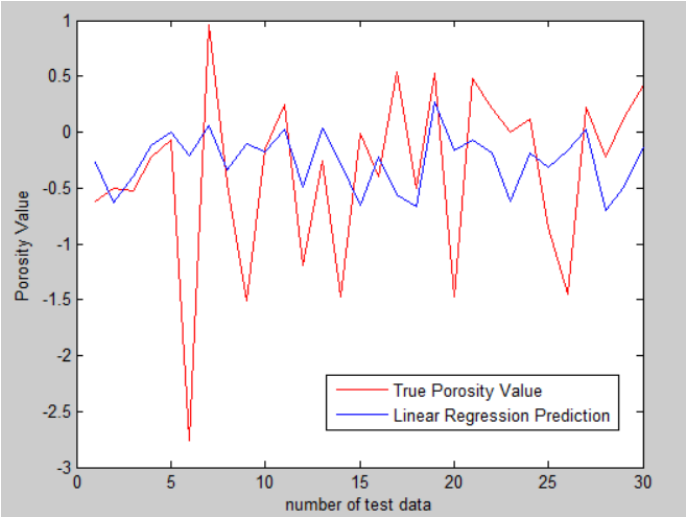

Figure 43. Linear Regression Prediction based on Attributes

Then, given the same attributes, I use SVM algorithm to train on the same 70% of data and predict the other 30%, and follow the same workflow described in this paper, the results in shown in Figure 44, the rooted mean squared error is 0.4201.

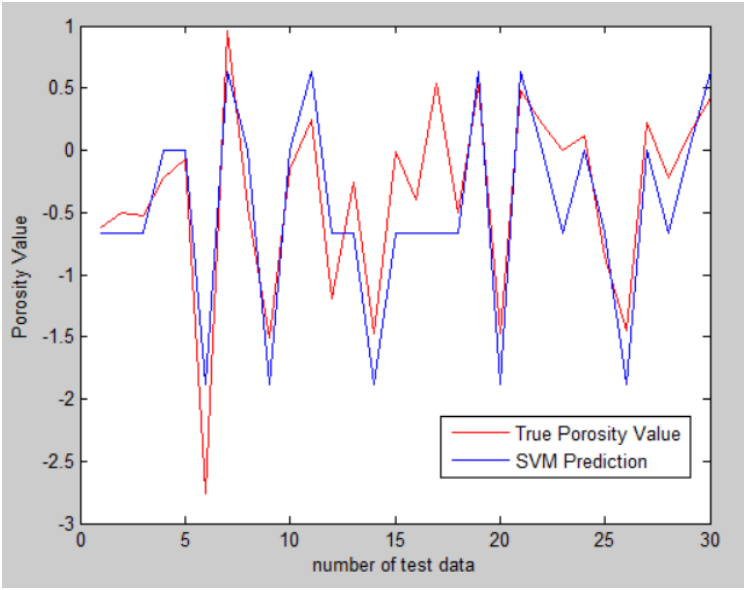

Figure 44. SVM Prediction Based on Attributes

Therefore, SVM is capable of make better use of data and dig out "extra information" from the same chaotic dataset. Conventionally, if we want to improve our

prediction accuracy, we would study the correlation between target and some factor in lab, then make adjustment in the model for the new correlation. However, this process is extremely time-consuming and experience demanding. Instead, one fascinating fact about data mining techniques is that it allows us to discover the pattern between targets and attributes implicitly, which means we could be able to predict porosity based on seismic attributes and accounting for the correlation for other factors without knowing what exactly the pattern is or how each attribute is related with each other. All we need to make sure is the attributes we use contains some degree of information about the factors that could determine the target. Once we input those attributes that into data mining algorithms, then algorithm could make the data speak for itself.

3.5 Chapter Conclusion

In this chapter, I am presenting a new data mining scheme for modeling porosity, compared with regular SGS modeling, combining data mining workflow with SGS model could make a better use of available data and provide extra information and help SGS build a more accurate model. These advantages are achieved by utilizing non-linear pattern between seismic attributes and porosity. The results of this workflow can be crucial to a successful field development.

Chapter 4 Conclusion

In this thesis, I developed three different applications that use data mining techniques for reservoir modeling purposes. The data mining algorithms includes Support Vector Machines, K-means algorithm, Probabilistic Neural Network and the Ensemble Learning algorithm. I also created a new geomodeling algorithm that is capable of handling non-linear correlations between different datasets by combining Support Vector Machines and Sequential Gaussian Simulation. The following conclusions can be drawn from this work:

1. Machine learning algorithms are a promising approach to integrate multiple data types that span different length scales for the purposes of reservoir characterization.
2. For the Barnett shale play, I used K-Means and Support Vector Machines for the prediction on lithotypes derived from core-data using a set of well log curves (SP, GR, DPHI, NPHI, SPHI, PE and RHOB). The results showed that we are able to achieve 76% accuracy in blind test prediction and can therefore identify lithotypes for un-cored wells with a high degree of accuracy.
3. Again for the Barnett shale play, I used the Ensemble Learning algorithm and Probabilistic Neural Network (PNN) to predict total organic content (TOC) based on a different set of well log curves (PHIS, PHIN, RHOB, Rd, GR) in cored wells. This application demonstrates that the TOC can be predicted accurately in un-cored wells. The blind test results also show that the predicted TOC zones using PNN share a great similarity with the core-based TOC measurement in the lab.
4. Lastly, I developed a new porosity modeling work flow that applied data mining techniques for an application in the Mississippi Limestone. The result shows that

applying new approach allows better use of exploratory data for more accurate estimation of the distribution of porosity in the reservoir.

In summary, data mining techniques have a huge potential for several applications in the petroleum industry. The advantages of data mining techniques are that they can handle vast amounts of data and are easily customizable for automated handling of diverse types of data. However, as with any data analysis technique, careful user interpretation of the results continues to be of the utmost importance.

References

B, L. Meyer. 1984. Identification of Source Rocks on Wireline Logs by Density-Resistivity and Sonic-Resistivity Crossplots. AAPG Bulletin, V. 68. pp. 121-129

Boser, E. B. Guyon, I. M. Vapnik, V. N. 1992. A training algorithm for optimal margin classifiers. Proceedings of the fifth annual workshop on Computational learning theory - COLT '92. pp. 144.

Boyd, Stephen. and Vandenberghe, Lieven. 2004. Convex Optimization. Cambridge University Press. pp. 143.

Clark, G. A., Glinsky, M. E., Devi, K. R. S., Robinson, J. H., Cheng, P. K. Z., & Ford, G. E. (1996, January 1). Automatic Event Picking In Pre-stack Migrated Gathers Using a Probabilistic Neural Network. Society of Exploration Geophysicists.

Corrtes, C. and Vapnik, V. 1995. Support-vector networks. Machine Learning 20 (3): pp. 273.

Doyen, P. M., den Boer, L. D., & Pillet, W. R. (1996, January 1). Seismic Porosity Mapping in the Ekofisk Field Using a New Form of Collocated Cokriging. Society of Petroleum Engineers. doi:10.2118/36498-MS

Duan, K. B. and Keerthi, S. S. 2005. Which Is the Best Multiclass SVM Method? An Empirical Study. Multiple Classifier Systems LNCS 3541. pp. 278–285.

Gutschick, R.C. and C, A. Sandberg. 1983. Mississippian continental margins of the conterminous United States, in D.J. Stanley and G.T. Moore. The Shelfbreak: critical interface on continental margins. SEPM Special Publication 33, pp. 79-96.

Holdaway, K. R. (2013, October 28). Data Mining Methodologies enhance Probabilistic Well Forecasting. Society of Petroleum Engineers. doi:10.2118/167428-MS

Isaaks,E.H., and Srivastava,R.M. (1989). An Introduction to Applied Geostatistics. 561 pp. Newyork: Oxford University Press

Kale S., 2009, Petrophysical Characterization of Barnett Shale Play, MS. Thesis, Oklahoma U., Norman, Oklahoma

Keerthi, S. Shevade, S. K. Bhattacharyya, C. and Murthy, K. R. K. 2001. Improvements to Platt's SMO algorithm for SVM classifier design. Neural Computation. v13. pp. 637–649.

Lindzey K., 2015, Geologically Constrained Seismic Characterization and 3-D Reservoir Modeling of Mississippian Reservoirs, North-Central Anadarko Shelf, Oklahoma. MS. Thesis, University of Oklahoma.

Minken, D. A., & Castagna, J. P. (2003, January 1). Gas-bearing Hydrothermal Dolomite Prediction Using Probabilistic Neural Networks In the Trenton-Black River Interval, NE Ohio. Society of Exploration Geophysicists.

Nazari, S., Kuzma, H. A., & Rector, J. W. (2011, January 1). Predicting Permeability From Well Log Data And Core Measurements Using Support Vector Machines. Society of Exploration Geophysicists.

Omoboya, B., Pacal, E., Dyaur, N., Stewart, R. R., & de Figueiredo, J. J. S. (2012, November 4). Fluid Filled Cracks and Seismic Anisotropy: An Experimental Study. Society of Exploration Geophysicists.

Passey, Q.R., Creaney, S., Kulla, J.B., Moretti, F.J., and Stroud, J.D. 1990. A practical model for organic richness from porosity and resistivity logs. AAPG Bulletin 74 (12): 1777–1794.

Platt, John. 1998. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Microsoft Research, Washington. Tech. pp. 98-14.

Reilly, J. M. (2002, January 1). 3D Prestack Data Mining to Meet Emerging Challenges. Society of Exploration Geophysicists.

Rogers, S.M. 2001. Deposition and diagenesis of Mississippian chat reservoirs, north-central Oklahoma: AAPG Bulletin, 85, pp. 115-129.

Singh P., 2008, Lithofacies and Sequence Stratigraphic Framework of the Barnett Shale, NorthEast Texas, Phd. Dissertation, Oklahoma U., Norman, Oklahoma

Specht, D. F. 1990. Probabilistic neural networks. Neural Networks 3. pp. 109–118.

Xu, W., Tran, T. T., Srivastava, R. M., & Journel, A. G. (1992, January 1). Integrating Seismic Data in Reservoir Modeling: The Collocated Cokriging Alternative. Society of Petroleum Engineers. doi:10.2118/24742-MS

Yenugu, M., Fisk, J. C., & Marfurt, K. J. (2010, January 1). Probabilistic Neural Network Inversion For Characterization of Coalbed Methane. Society of Exploration Geophysicists.

Zhao, B., Zhou, H., Li, X., & Han, D. (2006, January 1). Water Saturation Estimation Using Support Vector Machine. Society of Exploration Geophysicists.

Zhao, T., Jayaram, V., Marfurt, K. J., & Zhou, H. (2014, October 29). Lithofacies Classification in Barnett Shale Using Proximal Support Vector Machines. Society of Exploration Geophysicists.

Zhou, Z.H. 2012. Ensemble Methods: Foundations and Algorithms. Chapman & Hall/CRC. pp. 23-60.