Facies Characterization of a Reservoir in the North Sea Using Machine Learning Techniques

Project Final Report for CS229 (Aut 2016-17)

Peipei Li (peipei@stanford.edu); **Yuran Zhang** (yuranz4@stanford.edu)

Introduction

In oil and gas industry, oil is produced from wells drilled into the oil reservoirs. A thorough understanding and characterization of the subsurface is crucial to sustainable management of a reservoir. The subsurface, however, is largely heterogeneous and oil is not present everywhere. For our study area, the reservoir rock is categorized as three facies: brine sand, oil sand and shale, and it is only oil sand that we consider exploitable. The focus of this project is to identify oil sand based on well log data and seismic data using machine learning algorithms.

Well logging is a powerful tool for reservoir engineers to obtain information from the subsurface, where a wireline tool is lowered into the borehole, along which measurements were taken so that rock properties can be inferred from the measured data. Common well logs include gamma ray, P-wave velocity, S-wave velocity, density, etc. Gamma ray enables us to differentiate different faices (sand and shale) at the wellbore directly, and all the other logs are indirect indicators of facies. However, one limitation of well logs is that it is not able to provide information on the reservoir that is further away from the wellbore. On the other hand, seismic reflection data is generally gathered in more extensive area and can provide information for the whole 3D reservoir. While seismic data doesn't contain direct facies information, it can be used to get P-wave velocity, S-wave velocity, density, and other reservoir properties using seismic inversion techniques.

So the general idea of facies characterization is as follows. First, using well log data, a classification model is built to classify facies using reservoir properties like P-wave velocity, S-wave velocity and density. Then seismic inversion is done on seismic reflection data to invert for these properties in the whole 3D area. Once these properties are obtained from seismic data, the classification model built earlier can be applied to do prediction on the whole 3D area.

Facies characterization using seismic data and statistical rock physics has been a useful tool in reservoir exploration. However, the previous methods mostly focus on seismic inversion other than the statistical analysis [1][2][3][4][5]. In these methods, statistical rock physics is never fully exploited using machine learning methods. Instead, they are done qualitatively (scatter plot) or only two properties are used to do simple classification.

In this project, we focus on the statistical rock physics part where we will fully exploit classification of facies using well log data. Various machine learning algorithms such as GDA, logistic regression, Random Forest and SVM were explored and compared to find the best model.

Data Processing

The reservoir of interest in this project is in the North Sea. We have one well with a complete set of well logs available, which were used on training and selecting classification models.

The machine learning problem we propose is actually a multi-class classification problem. The labels are the three facies: shale, brine sand and oil sand which are created using GR logs and fluid

substitution. The features are three other well logs (V_p, V_s and $Rhob$) and six more properties calculated using available well logs.

Facies Identification

GR log data is considered as direct facies indicator. Combined with other geological observations, GR log serves as the "correct answer" of the labels of the training data. As shown in figure 1, shale and sand samples are selected using GR log and marked respectively in red and blue. The rest of reservoir (black) is considered as interbedded shale and sand (not pure shale or sand) and thus cannot be used in the modeling process. Along with the GR log used in facies identification, other well logs that will be used as features in building the statistical models are also displayed as reference, where V_p, V_s and $Rhob$ are respectively P-wave velocity, S-wave velocity and density of the reservoir at the well location.

In total, there are 394 shale samples and 492 sand samples selected available for modeling. These sand samples are considered as pure brine sand even though some of the sand is actually in oil zone. That's because mud used in the drilling and logging process is filtered into the formation while logging tools can only detect very shallow formation. Thus, it's actually measuring properties of brine sand. To retrieve oil sand properties, Gassmann fluid substitution is used to replace water with oil. By doing this, 492 oil sand samples were created as the third class.

Therefore, the total number of samples we have is 1378. As well log data is only available for one well, we further randomly sampled 80% of the data for model development, and saved 20% of the data for final testing and model comparison.

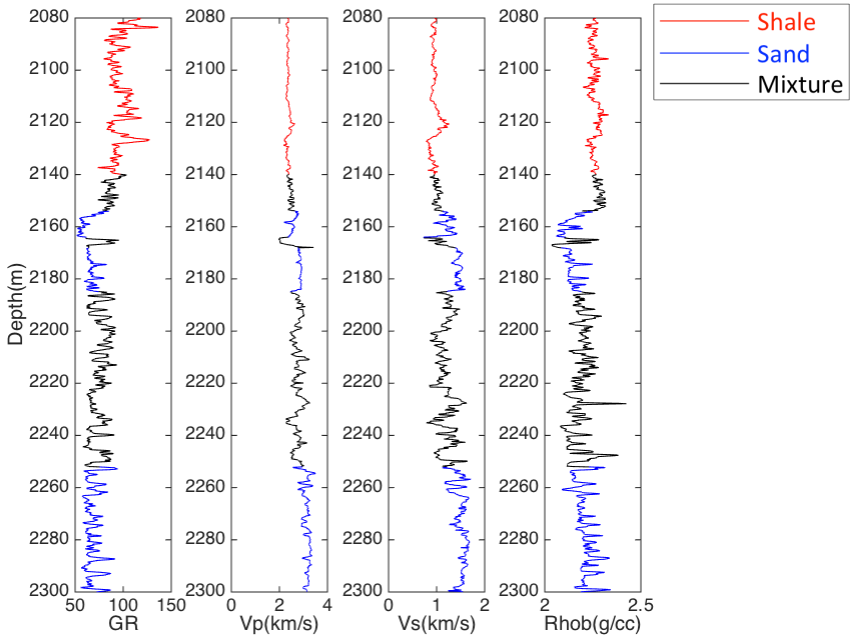

Figure 1. Gamma ray , V_p, V_s and density log data of the reservoir at the well location. Only pure shale and sand data were used for building the model.

Feature Creation

As shown in Figure 1, the available well logs V_p, V_s and $Rhob$ are indeed relevant to facies. Actually, in addition to these three, there are some other reservoir properties that could also be affected by different facies. These reservoir properties are basically non-linear functions of the above three basic reservoir properties. By applying the physical functions, six more features were created based

on the three available well logs. These extra features include: shear modulus ($\mu = Rhob \cdot V_s^2$), bulk modulus ($K = Rhob \cdot (V_p^2 - \frac{4}{3}V_s^2)$), P-wave impedance ($I_P = V_p \cdot Rhob$), S-wave impedance ($I_S = V_s \cdot Rhob$), Poisson's ratio ($\nu = \frac{V_p^2 - 2V_s^2}{2(V_p^2 - V_s^2)}$) and Lame's coefficient ($\lambda = Rhob \cdot \left(V_p^2 - \frac{4}{3}V_s^2\right) - \frac{2}{3}Rhob \cdot V_s^2$). Thus we have 9 features in total.

Implementation of Machine Learning Algorithms

Two linear classifiers (Softmax Regression and Gaussian Discriminant Analysis) and two non-linear classifiers (Random Forest and Support Vector Machine) are explored to solve the multi-classification problem. 80% of the data was used for training, such as feature selection and parameter tuning, and the rest 20% was used for final testing and model comparison.

Softmax Regression

Softmax regression is a generalized linear model that assumes the data as distributed according to a multinomial distribution. The softmax function is:

$$p(y = i|x;\theta) = \phi_i = \frac{e^{\theta_i^T x}}{\sum_{j=1}^{k} e^{\theta_j^T x}}$$

The "multinom" function from the "nnet" package in R was used to effectively conduct softmax regression on the data. In order to find out which features are most "relevant" to the learning task, forward search feature selection was applied in order to find the best feature subset. Each feature combination in the forward search method was evaluated using k-fold cross validation (with k = 10). Given the fact that there are 1000+ data samples available for modeling, k-fold cross validation not only holds out merely 10% of data in each training iteration, but is also computationally inexpensive since the sample size is not too large.

The feature set $\mathcal{F}$ obtained from forward search method is: {"$Rhob$", "μ", "V_p","I_S", "ν", "I_p", "V_s", "λ", "K"}. It was found that the cross validation error (CV error) is lowest when ~5 features were considered, as illustrated in Figure 2 (left). When the number of features further increased beyond 5, the testing error increased slightly, indicating a small degree of overfitting.

Gaussian Discriminant Analysis (GDA)

GDA is a generative learning algorithm. It tries to find a linear transformation to maximize the so-called Rayleigh coefficient, that is, the ratio of the determinant of the inter-class scatter matrix of the projected samples to the intra-class scatter matrix of the projected samples [6]:

$$J(\Phi) = \frac{\Phi^T \hat{\Sigma}_b \Phi}{\Phi^T \hat{\Sigma}_w \Phi}$$

Where $\hat{\Sigma}_b$ and $\hat{\Sigma}_w$ are respectively the inter-class and intra-class matrix.

The "lda" function from "mass" library in R was used to conduct GDA on the data. Forward search feature selection via k-fold cross validation is applied as described in the Softmax Regression section. The feature set $\mathcal{F}$ obtained from forward search method is: {"$Rhob$", "μ", "V_p", "V_s", "I_S", "λ", "I_p", "K", "ν"}. It was found that the CV error was lowest when 7 features were considered in model training, as illustrated in Figure 2 (right).

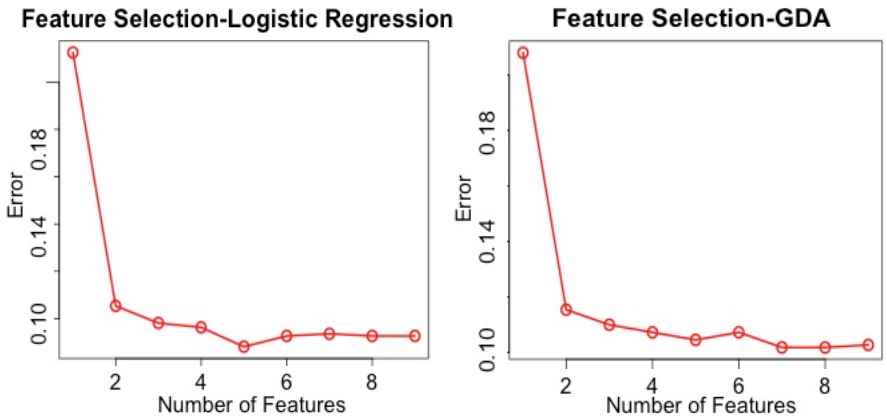

Figure 2. Error vs #features for the Softmax Regression model, indicating that 5 features yields lowest error (left); Error vs #features for the GDA model, indicating that for GDA, 7 features yields lowest error (right).

Random Forest

Random Forest is a special case of bagging methods for decision trees. "randomForest" function from "randomForest" library in R was used to perform Random Forest classification on the data. We used k-fold cross validation (k=10) to determine the best "mtry" value for classification, where "mtry" denotes the number of variables randomly sampled as candidates at each split [7]. As indicated in the figure below, mtry = 5 yields the lowest CV error. Note that all possible "mtry" values give very low error, and the lowest error is not much different from the rest.

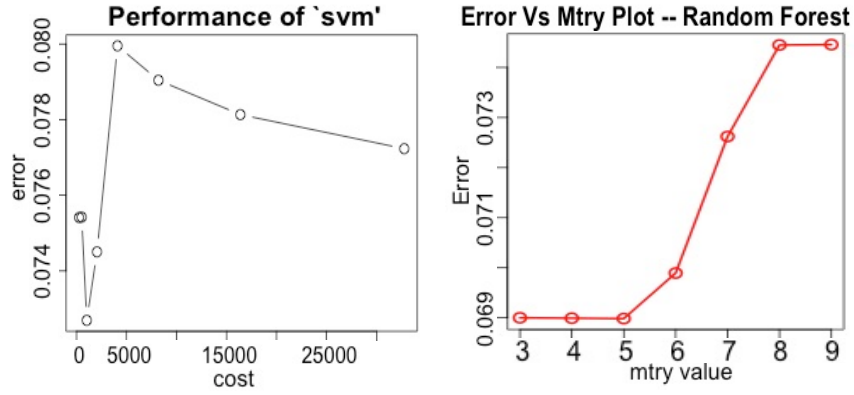

Figure 3. Parameter selection results for SVM (left) and Random Forest (right)

Support Vector Machine (SVM)

Support Vector Machine is a maximum margin classifier. It tries to solve the following optimization problem:

$$min_{\gamma,\omega,b}\ \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{m}\xi_i\,, \qquad s.t.\ y^{(i)}\left(\omega^T x^{(i)} + b\right) \geq 1 - \xi_i, \qquad \xi_i \geq 0, \qquad i = 1 \cdots m$$

"svm" function from "e1071" library in R was used to perform classification on the data [8]. The kernel function used in this project is Gaussian radial basis function. A range of the cost parameter C is evaluated, and the result is optimal when C = 1024, as illustrated in Figure 2.

Model Comparison & Results & Discussion

Now we rebuild each model using selected features or parameters in the previous section on the training data and compare the models using the testing data. The training error and testing error are summarized in Table 1.

Table 1. Model comparison with MSE

Methods	Training Error	Testing Error
Softmax Regression	0.086	0.12
GDA	0.107	0.13
SVM	0.051	0.101
Random Forest	0.065	0.109

As shown in the table, non-linear models generally outperform linear models, as indicated by their lower training and testing error. For each model, testing errors are slightly higher than training errors, indicating proper model complexity.

To get more insights on model performance, we generated the confusion matrix for each model, as illustrated in Figure 4. It can be seen that the results are consistent with model comparison using MSE. All models performed well in discriminating sand from shale, although differentiating oil sand from brine sand turned out to be slightly less optimal.

Among all four algorithms, SVM achieves best classification results, especially in terms of the low number of false positive predictions of oil sand.

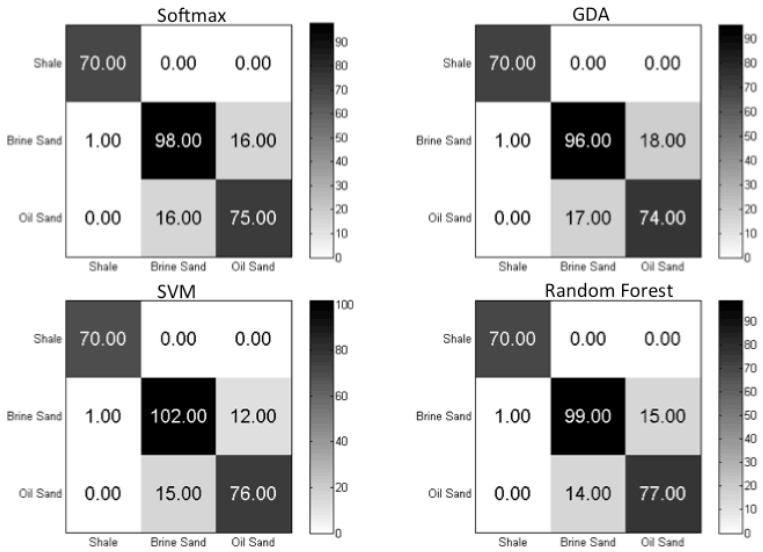

Figure 4. Model comparison with confusion matrix

Future Work

With the conclusion above, the next step is to apply the SVM to 3D dataset for facies classification. However, we only have P-wave impedance and S-wave impedance inverted from seismic data available for use. As a reference, we built a softmax regression model using these two features only and it showed an error of 0.23. So we think getting more seismic attributes could improve the classification significantly.

References

[1] Avseth, P., et al. "Seismic reservoir mapping from 3-D AVO in a North Sea turbidite system." Geophysics 66.4 (2001): 1157-1176.

[2] Mukerji, T., et al. "Mapping lithofacies and pore-fluid probabilities in a North Sea reservoir: Seismic inversions and statistical rock physics." Geophysics66.4 (2001): 988-1001.

[3] Mukerji, Tapan, et al. "Statistical rock physics: Combining rock physics, information theory, and geostatistics to reduce uncertainty in seismic reservoir characterization." The Leading Edge 20.3 (2001): 313-319.

[4] Avseth, Per Åge. Combining rock physics and sedimentology for seismic reservoir characterization of North Sea turbidite systems. Diss. Stanford University, 2000.

[5] Bosch, Miguel, Tapan Mukerji, and Ezequiel F. Gonzalez. "Seismic inversion for reservoir properties combining statistical rock physics and geostatistics: A review." Geophysics 75.5 (2010): 75A165-75A176.

[6] Li, Tao, Shenghuo Zhu, and Mitsunori Ogihara. "Using discriminant analysis for multi-class classification: an experimental investigation." Knowledge and information systems 10.4 (2006): 453-472.

[7] Liaw, Andy, and Matthew Wiener. "Classification and regression by randomForest." R news 2.3 (2002): 18-22.

[8] Meyer, David, and FH Technikum Wien. "Support vector machines." The Interface to libsvm in package e1071 (2015).