

Comparison between Various Machine Learning Classifiers to Predict Rock Facies

Ryan A. Mardani
Geo Vision Thrust

Summary

The objective of this study is to predict rock facies from well logs using machine learning concept. In this work, six different classifiers including parametric and non-parametric methods were employed to predict rock facies. Dataset has nine wells including 3600 samples of rock facies, five well logs and two geologic constraining variables. One well was excluded from training as a blind well. Then, it is divided into subset of training and testing to examine classifiers prediction accuracy. Gaussian process classifier, Random forest, Multi-layer Perceptron, K-nearest neighbor, Decision tree classifier and support vector machines are those approaches in this process. All models need to be tuned with optimized model parameters in order to show higher accuracy. These classifiers could predict facies classes with accuracy in range of 0.64 and 0.74. The best performance belongs to Multi-layer feed forward-back propagating Neural Network. Testing models with blind well shows considerable accuracy loss. This can be possibly improved if more data introduces into training dataset.

Introduction

Facies classification is one of the most important tasks that geoscientists work on development and exploration projects. Sedimentary facies reflect particular physical, chemical, and biological conditions that units experienced during sedimentation process. To study these facies, rock samples are required. In this study, it is tried to train various machine learning algorithms to predict facies from well log data. The dataset for this study, comes from Hugoton and Panoma Fields in North America which was used in class exercise in The University of Kansas (Dubois et. al, 2007). It consists of log data of nine wells. We will use these log data to train supervised classifiers in order to predict discrete facies groups. All these implementations are based on scikit-learn libraries. These are: 1) Gaussian process classification, 2) Random forest classifier, 3) Multi-layer Perceptron classifier, 4) K Neighbors Classifier, 5) Decision tree classifier and 6) Support vector machines.

Background

Let's have very short review on these classifiers.

Gaussian process classification (GPC):

This is a generic supervised learning for probabilistic classification, where test predictions take the form of class probabilities. It uses Laplace approximation for non-Gaussian posterior.

Random forest classifier (RFC):

This classifier is meta estimator that fits some decision tree classifier on different sub-samples employing averaging to avoid over-fitting as well as prediction accuracy improvement.

Multi-layer Perceptron classifier (MLPC):

MLPC model optimizes the log-loss function using various algorithms such as stochastic gradient descent. In this model you may apply favorable parameters for number of hidden layer, activation function and solvers. In fact, this is Neural Network training implementation in scikit library.

K Neighbors Classifier (KNC):

This classification tries to find a predefined number of training samples closest in distance to the new point and predict the target from these. It stores instances of training data based on K (nearest neighbors of each query point) which is an integer value specified by user to implement learning process.

Decision tree classifier (DTC):

This is non-parametric supervised learning used for both classification and regression. The model predicts the value of target data by learning simple decision rules inferred from dataset attributes.

Support vector machines (SVM):

Similarly, this supervised learning approach can be used for both classification and regression. It is effective in high dimensional spaces and uses subset training points in decision function (named as support vector), so memory efficient (Pedregosa et al. 2011).

Results & Discussion

In the first step, we read dataset into python to explore various features of training dataset. From nine wells, one is stored as blind well to further model accuracy examination. Available well logs are: gamma ray (GR), induction resistivity log (ILD_log10), photoelectric effect (PE), neutron-density porosity difference (DeltaPHI) and average neutron-density porosity and (PHIND). There is also two geologic variables: nonmarine-marine indicator (NM_M) and relative position (RELPOS) (Matt Hall, 2016).

The nine discrete rock facies are listed in the table 1. In this field succession of clastic and carbonate intervals can be seen. All facies boundaries are not distinct. Geologically they have transition with adjacent facies. Figure 1 shows facies frequency in the dataset. Dolomite has the least frequency while facies 2 (Nonmarine coarse siltstone) has highest samples in dataset. Figure 2 illustrates well logs and facies variation through the well NOLAN.

| | No. | Facies Name | Label | Neighbor Facies |
|-----------|-----|----------------------------|-------|-----------------|
| Clastic | 1 | Nonmarine sandstone | SS | 2 |
| | 2 | Nonmarine coarse siltstone | CSiS | 1,3 |
| | 3 | Nonmarine fine siltstone | FSiS | 2 |
| | 4 | Marine siltstone and shale | SiSh | 5 |
| Carbonate | 5 | Mudstone | MS | 4,6 |
| | 6 | Wackestone | WS | 5,7 |
| | 7 | Dolomite | D | 6,8 |
| | 8 | Packstone-grainstone | PS | 6,7,9 |
| | 9 | Phylloid-algal bafflestone | BS | 7,8 |

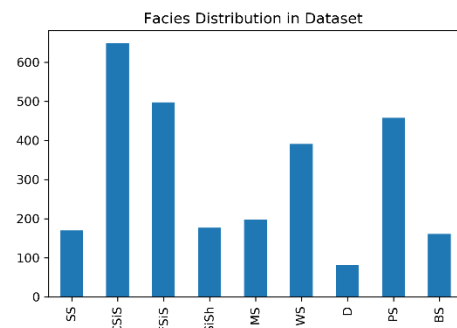


Table1: Facies types, labels and their adjacent facies

Figure 1. Facies frequency in dataset

Setting up and running models with default parameters did not give appealing results. For all models, we examined a range of sensitive model parameters and chose the optimized ones. To examine the model accuracy, there several methods such as confusion matrix, log-loss and Jaccard index. For simplicity we select the last one. Jaccard similarity coefficient is defined as size of intersection of real (y) and predicted values (y_p) divided by size of the union of those two labels.

$$J(y, y_p) = \frac{|y \cap y_p|}{|y \cup y_p|}$$

For example, for a set of size 10 samples with 8 correct prediction, this index will be 0.66.

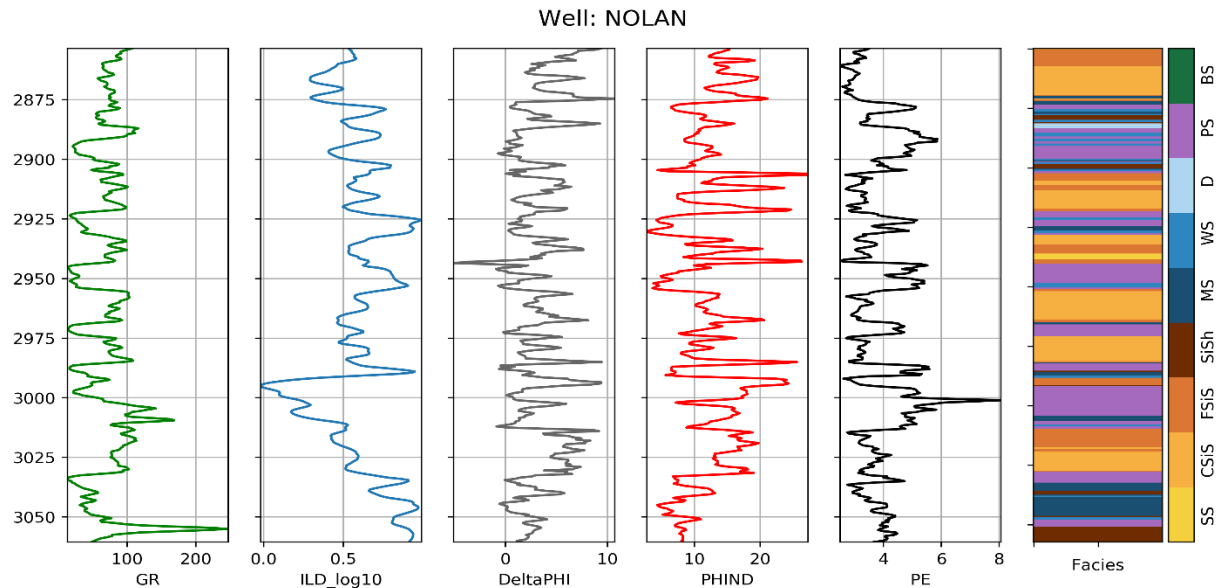


Figure 2. From left: gamma ray (GR), induction resistivity log (ILD_log10), photoelectric effect (PE), neutron-density porosity difference (DeltaPHI) and average neutron-density porosity and (PHIND) and facies distribution in one well.

Eliminating blind well from calculations, we divided dataset into training (80%) and test data (20%). The result of training various models with optimized parameters is listed in table 2. Jaccard coefficient ranges from 0.64 to 0.74 for models in training and test dataset. Feed forward-back propagating artificial neural network shows the best performance while Decision tree and Gaussian process classifier shows weaker results. MCLP performance can be interpreted as complexity in data interior structure such as non-linearity. When we use the constructed models to predict facies groups in blind well, accuracy drops considerably. The most considerable difference belongs to K-nearest neighbor method. When this happens, we may consider that models are not trained with enough data samples or data points are not evenly distributed. In fact, all characters did not imprint inside feature logs that we employed here for facies prediction. Finally, let's look at the blind well facies prediction in figure 3. Overall, classifiers could distinguish

| | Accuracy (Jaccard Index) | |
|------|--|---------------------------|
| | Performance on training & test dataset | Performance on blind well |
| GPC | 0.65 | 0.52 |
| RFC | 0.70 | 0.50 |
| MLPC | 0.74 | 0.55 |
| KNC | 0.72 | 0.44 |
| DTC | 0.64 | 0.45 |
| SVM | 0.72 | 0.42 |

Table2: Models performance

main features in the blind well. It is common for all models that thick layers are divided into thinner ones in prediction mainly with neighboring facies. This is a clue that sharp feature property changes would be beneficial in model performance while transitional variations would lead weaker predictions.

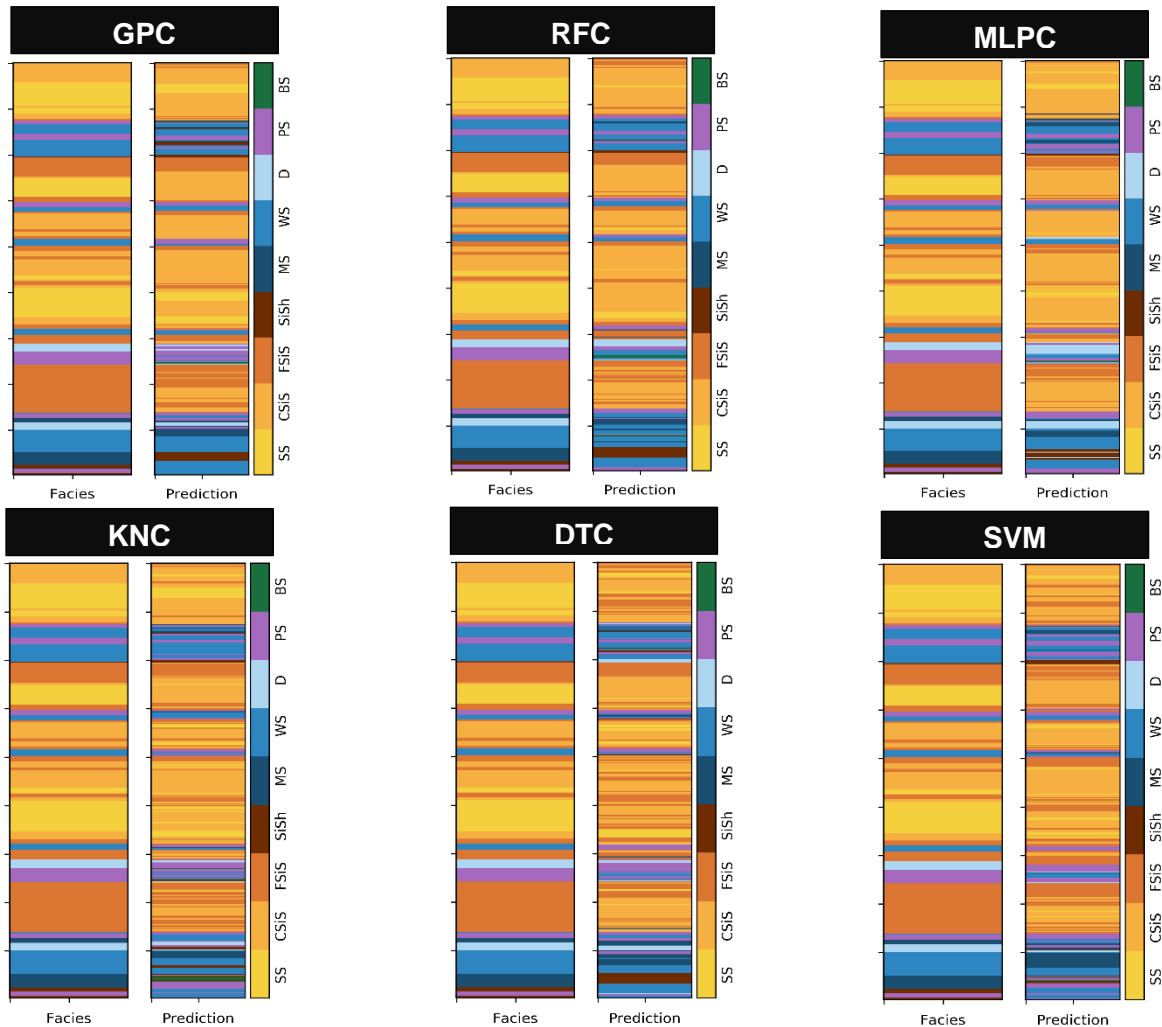


Figure 2. Each model prediction is juxtaposed with real facies in blind well. Quantitatively, MLCP has the highest accuracy in prediction. It is almost common for all models that classifier mispredicted the real facies with adjacent facies.

Conclusion

The objective of this study is to predict rock facies groups which is mainly accessible through core studies. Six various machine learning binary classifiers were employed to train with dataset including well logs and geologic constraining variable from nine wells. Excluding one well data as blind well, models are examined where dataset divided into training and testing subsets. Default model parameters generated unacceptable results, so they tuned and optimized using ranges of parameters in iterative mode. Multi-layer feed forward-back propagating Neural Network resulted the highest accuracy. This can be interpreted as non-linearity and data structure complexity. Obviously, having access for more data point, like other machine leaning examples, would increase result accuracy.

Acknowledgements

We would like to express our appreciation for Matt Hall from agile-geoscience. This research has been benefited from source codes shared by him.

References

- Bohling, G. C., and M. K. Dubois, 2003. An Integrated Application of Neural Network and Markov Chain Techniques to Prediction of Lithofacies from Well Logs, *KGS Open-File Report 2003-50*, 6 pp.
- Dubois, M. K., G. C. Bohling, and S. Chakrabarti, 2007, Comparison of four approaches to a rock facies classification problem, *Computers & Geosciences*, 33 (5), 599-617
- Matt Hall, 2016, Facies Classification, https://github.com/seg/2016-ml-contest/blob/master/Facies_classification.ipynb
- Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830,