A Machine Learning Approach to Facies Classification Using Well Logs

Paolo Bestagini, Vincenzo Lipari and Stefano Tubaro, Politecnico di Milano, Italy

SUMMARY

In this work we describe a machine learning pipeline for facies classification based on wireline logging measurements. The algorithm has been designed to work even with a relatively small training set and amount of features. The method is based on a gradient boosting classifier which demonstrated to be effective in such a circumstance. A key aspect of the algorithm is feature augmentation, which resulted in a significant boost in accuracy.
The algorithm has been tested also through participation to the SEG machine learning contest.

INTRODUCTION

Facies (i.e., lithofacies) classification consists in assigning a rock type or class to a specific sample on the basis of measured features. Classification of different lithofacies is crucial in seismic interpretation because different rocks have different permeability and fluid saturation for a given porosity.

The ideal sources for lithofacies classification are core samples of rocks extracted from wells. Nevertheless, core samples cannot always be obtained due to associated costs. Therefore, a method for classifying facies from indirect measurements (e.g., wireline logs) is necessary.

The conventional method consists in manually assigning lithofacies by human interpreters and is a very tedious and time consuming process. Therefore, several alternative approaches to the issue of facies classification from well data have been proposed. The first works were based on classical multivariate statistical methods (Wolf et al., 1982; Busch et al., 1987). Later, Wolf et al. (1982); Busch et al. (1987) proposed the use of neural networks for rock classification (Baldwin et al., 1990; Rogers et al., 1992).

Meanwhile, especially in the last few years, encouraged by the growth of the so called *big data* and by the increased computational power, there has been an increasing renewed interest in machine learning techniques. Recently these methodologies have been increasingly explored, for different applications, also by the geophysical community (Smith and Treitel, 2010; Zhang et al., 2014; Zhao et al., 2015; Kobrunov and Priezzhev, 2016).

In this context Hall (2016) proposed a Geophysical Tutorial where he showed a simple application of machine learning techniques for facies classification. In particular he used a small dataset of seven wireline logs and associated interpreted facies extracted from ten wells of the Hugoton gas field in southwest Kansas (Dubois et al., 2007), in order to predict geologic facies in two additional wells based on wireline measurements. On the same occasion, he announced a machine learning contest inviting readers to propose alternative solution for the same task. Results of the machine learning contest are resumed in a paper written by the organizers (Hall and Hall, 2017) and all submitted code is available online*.

In this paper we describe the machine learning pipeline we used for our submission to the facies classification challenge, highlighting the main characteristics that enable our method to achieve promising results. Specifically, our main contribution, as recognized by the organizers (Hall and Hall, 2017), has been a feature augmentation framework which has been subsequently used by all top scored teams.

In the following, we first formalize the facies recognition problem given for the SEG machine learning contest. Next, we describe the proposed facies classification algorithm. Finally, we discuss the results and draw our conclusions.

PROBLEM FORMULATION

Well logging is common practice whenever ground stratification characteristics must be studied (e.g., preliminary exploration of new fields). Through logging, it is possible to obtain a detailed description of rock formations at different depth levels by measuring a wide variety of rock properties. From this premises, the goal of facies classification is to detect the facies present at a given depth in the ground, from the analysis of a set of well log measurements obtained at the considered depth (Hall, 2016).

Assuming the same setup presented in the facies classification challenge (Hall, 2016) and previous studies (Bohling and Dubois, 2003; Dubois et al., 2007), we consider that at each depth d of each well indexed by w, a set of seven scalar attributes is available:

- *Gamma ray* ($f_{d,w}^{\mathrm{GR}}$) measures natural formation radioactivity;
- *Resistivity* ($f_{d,w}^{\mathrm{Res}}$) measures the subsurface ability to impede the flow of electric current;
- *Photoelectric effect* ($f_{d,w}^{\mathrm{PE}}$) measures electrons emission of a facies illuminated by light rays;
- *Neutron-density porosity difference* ($f_{d,w}^{\mathrm{Ndiff}}$) and *average neutron-density porosity* ($f_{d,w}^{\mathrm{Navg}}$) are measurements correlated to facies density;
- *Nonmarine/marine indicator* ($f_{d,w}^{\mathrm{NM}}$) is a binary flag attributed by experts to distinguish between marine and nonmarine facies upon data inspection;
- *Relative position* ($f_{d,w}^{\mathrm{RP}}$) is the integer index of each layer depth starting from 1 for the top layer, and increasing with depth.

*https://github.com/seg/2016-ml-contest

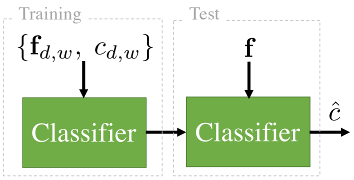


Figure 1: Pipeline of the proposed algorithm, highlighting training and test phases.

Additionally, facies characterizing the layer at depth d of the w-th well can be one of the following nine ones:

1. Nonmarine sandstone (SS);
2. Nonmarine coarse siltstone (CSiS);
3. Nonmarine fine siltstone (FSiS);
4. Marine siltstone and shale (SiSh);
5. Mudstone (MS);
6. Wackestone (WS);
7. Dolomite (D);
8. Packstone-grainstone (PS);
9. Phylloid-algal bafflestone (BS).

From the machine learning point of view, this means we can concatenate the available seven attributes and associate to each depth and well a *feature vector* defined as

$$\mathbf{f}_{d,w} = [f^{\text{GR}}_{d,w}, f^{\text{Res}}_{d,w}, f^{\text{PE}}_{d,w}, f^{\text{Ndiff}}_{d,w}, f^{\text{Navg}}_{d,w}, f^{\text{NM}}_{d,w}, f^{\text{RP}}_{d,w}]. \qquad (1)$$

Moreover, we can associate to each depth and well a *class label* $c_{d,w} \in \{\text{SS}, \text{CSiS}, \text{FSiS}, \text{SiSh}, \text{MS}, \text{WS}, \text{D}, \text{PS}, \text{BS}\}$, indicating the facies associated to the layer. We can then define our set of observations as all tuples

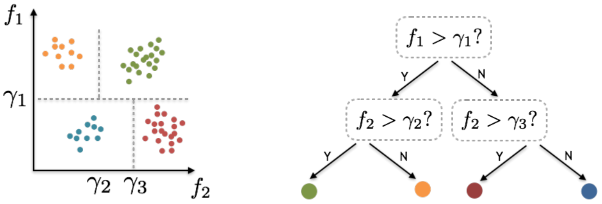
$$\{\mathbf{f}_{d,w},\ c_{d,w}\},\ d \in \mathcal{D}_w,\ w \in \mathcal{W}, \qquad (2)$$

where $\mathcal{W}$ is the set of considered wells and $\mathcal{D}_w$ is the set of considered depths for the w-th well.

With this notation in mind, the problem of facies classification turns into the problem of estimating a class label $c_{d,w}$ given a feature vector $\mathbf{f}_{d,w}$.

FACIES CLASSIFICATION ALGORITHM

The facies classification algorithm we propose in this work follows the common supervised machine learning pipeline shown in Figure 1. During training, a set of labeled observations from controlled wells $\{\mathbf{f}_{d,w},\ c_{d,w}\},\ d \in \mathcal{D}_w,\ w \in \mathcal{W}_{\text{train}}$ is used to learn a function that maps features to class labels. After training, the classifier can be used to estimate class labels $\hat{c}$ from any unlabeled feature vector $\mathbf{f}$ coming from new well logs. In the following we provide details about the proposed algorithm, introducing how the classifier works, and specifying the feature augmentation strategy we have devised. Finally, we report some additional remarks about post-processing operations used to correct some possible misclassification errors. An implementation of the proposed algorithm can be also found online[†].



(a) Geometrical interpretation

(b) Tree interpretation

Figure 2: Example of the geometrical and tree interpretation of a decision tree classifier.

Classifier

Many different and complex supervised classification techniques have been developed through the years (Bishop, 2006). However, considering that we only have seven independent features and a limited amount of training data, we decided to rely on a simpler classification technique. Our algorithm makes use of a *gradient boosting classifier* (Friedman, 2000), which is an ensemble of *decision trees* (Breiman et al., 1984). As a detailed explanation of this classifier is beyond the scope of this manuscript and has been deeply treated in the literature (Bishop, 2006), hereinafter we just provide the reader with some hints behind its way of working through examples.

Let us consider a simplified scenario in which a feature vector is composed by two scalar features (i.e., $\mathbf{f} = [f_1, f_2]$), and only four different classes (i.e., facies) exist. Figure 2a shows an example of feature vectors distribution in the (f_1, f_2) space, where different colors represent different classes. In this example, it is possible to notice that the (f_1, f_2) space can be easily partitioned into four sectors, each one containing only feature vectors belonging to a different class. During training step, a decision tree classifier learns a set of thresholds (i.e., γ_1, γ_2 and γ_3 in the example) to partition the space in order to separate features belonging to different classes. This is typically done by minimizing some cost-function conveniently defined (Rokach and Maimon, 2005) (e.g., misclassification error). To classify a new feature vector $\mathbf{f}$, it is sufficient to compare its components f_1 and f_2 against the three learned thresholds. This step can be logically represented as a tree diagram, as shown in Figure 2b. For this reason, this kind of classifiers is known as decision tree (Breiman et al., 1984).

One of the issues of decision trees, is that they tend to overfit training data. This means that during training, the classifier learns a set of thresholds that well separate training data into classes, but hardly generalize to new data (Ho, 1998). In order to avoid this effect, several methods have been developed to merger results from several smaller trees (Ho, 1995, 1998; Friedman, 2000). Among these solutions, gradient boosting classifier proposed by Friedman (2000) works by splitting training data observations into different subsets. From each subset, a limited amount of features are selected (i.e., only some ele-

[†] `https://bitbucket.org/polimi-ispl`

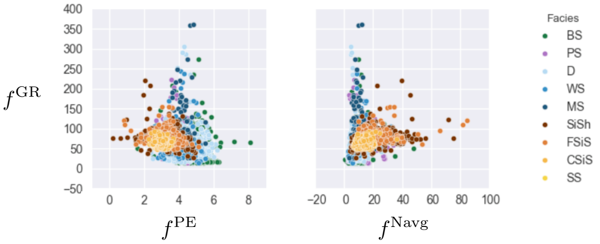


Figure 3: Example of classes distributions according to real features.

ments of **f**) to train a separate decision tree solving a gradient-based minimization problem. Once a new feature vector **f** must be classified, it is first tested against each different tree in the forest. Each tree provides a candidate label. Results from all trees are then merged into a single decision $\hat{c}$.

Feature Augmentation

Despite the simplicity of the example depicted in Figure 2, real facies are not easy to linearly separate with thresholds in the feature space. An example of facies distribution in $(f^{\mathrm{GR}}, f^{\mathrm{PE}})$ and $(f^{\mathrm{GR}}, f^{\mathrm{Navg}})$ spaces is shown in Figure 3. Notice how colors (i.e., facies) are often mixed together rather than being separated into clusters. To solve this issue and ease the classifier's work, we propose to use some feature augmentation techniques. This means, generate new features starting from the available ones, in order to better separate classes into an augmented feature space. This step was recognized also in Hall and Hall (2017) to be of great impact for the facies classification challenge.

To expand the number of available features, a common technique is to add some non-linearities. To do so, we compute the augmented feature vector $\mathbf{f}'_{d,w}$ applying quadratic expansion to $\mathbf{f}_{d,w}$ as

$$\mathbf{f}'_{d,w} = \mathbf{f}^2_{d,w} = [(f^{\mathrm{GR}}_{d,w})^2,\ (f^{\mathrm{Res}}_{d,w})^2, \ldots,\ (f^{\mathrm{RP}}_{d,w})^2], \tag{3}$$

where all operations are applied elementwise. Additionally, we compute the augmented feature vector $\mathbf{f}''_{d,w}$ by considering all second order interaction terms

$$\mathbf{f}''_{d,w} = [f^{\mathrm{GR}}_{d,w} \cdot f^{\mathrm{Res}}_{d,w},\quad f^{\mathrm{GR}}_{d,w} \cdot f^{\mathrm{PE}}_{d,w},\quad \ldots, f^{\mathrm{NM}}_{d,w} \cdot f^{\mathrm{RP}}_{d,w}]. \tag{4}$$

Given the physical nature of the problem, it is also possible to exploit some apriori information to further obtain additional derived features. As a matter of fact, we can consider facies at neighboring layers to be strongly correlated. In other words, it is unlikely that facies are randomly distributed underground. From this reasonable assumptions, we believe that features at layers $d+1$ and $d-1$ can help in classifying feature vector at layer d. From this intuition, we define the augmented gradient feature vector as

$$\mathbf{f}'''_{d,w} = \frac{\mathbf{f}_{d-1,w} - \mathbf{f}_{d,w}}{\Delta\text{depth}}, \tag{5}$$

where all operations are applied elementwise, and Δdepth is the depth difference in meters between layer d and $d-1$.

Once the three augmented feature vectors are computed, we concatenate all of them, and feed the classifier with the final feature vector

$$\mathbf{f}_{d,w} = [\mathbf{f}'_{d,w},\quad \mathbf{f}''_{d,w},\quad \mathbf{f}'''_{d,w}]. \tag{6}$$

Notice that, even if some of the augmented features might have no physical meaning (e.g., gradient of nonmarine/marine indicator), this is not a problem. As a matter of fact, gradient boosting classifier will take care of non-informative features by avoiding selecting them for tree creation at training time.

Post-Processing Refinement

Considering that facies at neighboring layers are correlated, we propose a further classification refinement step based on a simple post-processing operation. Specifically, let us assume a set of layers at neighboring depths. If all facies are of one kind (e.g., SS), and only a single different facies is detected among these layers (e.g., WS), we can reasonably believe that this single outlier is due to a misclassification error. To compensate for this kind of errors, we make use of a median filter applied to obtained class estimates $\hat{c}_{d,w}$ for each well in the d dimension. This removes spurious isolated values and substitute them with the most present facies in their neighborhood.

EXPERIMENTAL RESULTS

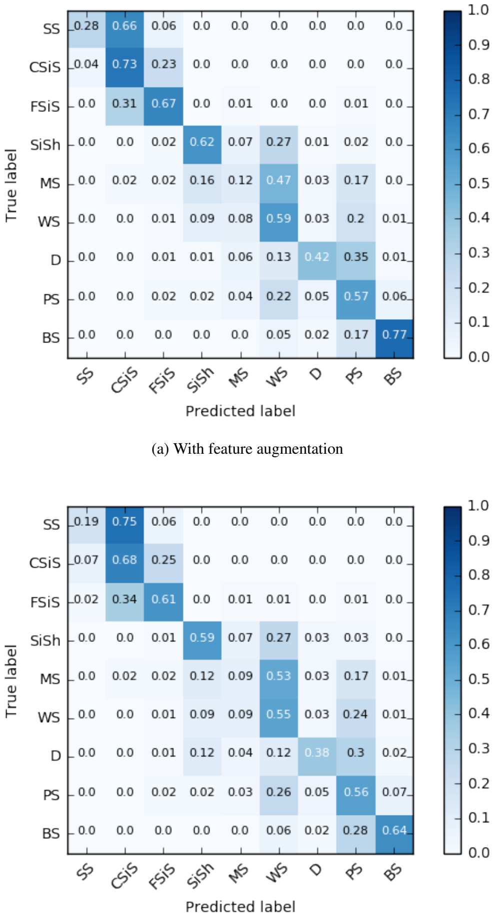
Our experimental campaign has been carried out on the dataset released for the facies classification challenge described by Hall (2016). This dataset is composed by more than 4 000 observations of labeled well log measurements coming from 10 different wells. For many machine learning problems this can be considered a small training set.

Concerning the used gradient boosting classifier, we considered a forest of 100 trees, each one fed with no more than 10 features, and adopted a one-versus-one strategy (Bishop, 2006) to deal with the multi-class problem (i.e., 9 facies).

As validation protocol, we selected a 10-fold cross-validation strategy. This means, we select observations from a set $\mathcal{W}_{\mathrm{train}}$ of 9 wells to train our classifier, then test it on the 10-th remaining well. The training and testing procedure is repeated 10 times shuffling the wells in order to build different sets $\mathcal{W}_{\mathrm{train}}$ and test each well once. Results are then averaged over the 10 repeated experiments.

Figure 4a shows the confusion matrix obtained with the proposed method. Each entry of the confusion matrix indicates the percentage of observations of a given class (true label) identified as any class (predicted label). The accuracy value of the detector can be obtained by averaging diagonal values of the confusion matrix. A perfect detector would produce a diagonal confusion matrix.

From these results it is possible to notice that some facies are actually easier to classify than others. As an example, it is possible to correctly detect the 77% of bafflestone (BS) and the 73% of nonmarine coarse siltstone (CSiS). Conversely, mudstone (MS) is more often identified as wackstone (WS).



(a) With feature augmentation

(b) Without feature augmentation

Figure 4: Average confusion matrix obtained through 10-folds cross validation using (a) or not (b) the proposed feature augmentation strategy.

Nonetheless, marine facies are misclassified for other marine facies. Accordingly, nonmarine facies are misclassified for other nonmarine facies. Only rarely it occurs that a marine facies is confused for a nonmarine one, and viceversa.

For the sake of comparison, we also report in Figure 4b the confusion matrix obtained using gradient boosting without the proposed feature augmentation strategies (i.e., using only the original 7 feature values provided by the challenge). Notice that all classes show a drop in performance, as all elements on the diagonal are lower in this situation than using the whole proposed pipeline. This confirms the importance of feature augmentation.

In order to explain why some facies are better recognized than others, we report in Figure 5 some statistics about how many observations of each facies are present per well. It is immediately possible to notice that some facies are represented many more times than others, thus making the classification problem strongly unbalanced. In particular, mudstone (MS) is the facies with less samples in the dataset, thus it is not surprising that the classifier did not learned very well how to characterize it. Conversely, many samples of nonmarine coarse siltstone (CSiS) are available. Indeed, this is one of the best detected facies.

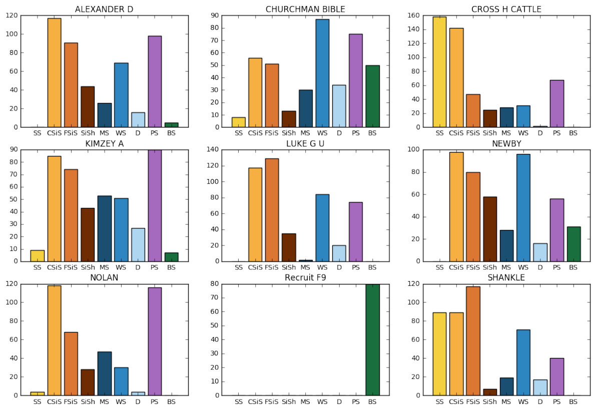


Figure 5: Distribution of facies for 9 wells. Well names are reported over each plot.

Finally, as suggested by Hall (2016), we also evaluated our algorithm in terms of f-score. Hall and Hall (2017) declare that random guess score is around 0.16 due to class unbalancing. Our algorithm achieves an f-score of 0.61 averaged over the available 10 folds. This is perfectly in line with the f-score of about 0.62 we obtained on completely unknown wells according to Hall and Hall (2017), thus confirming the good generalization capability of the proposed strategy. Moreover, we computed also the f-score achieved without using feature augmentation. This value is 0.55, thus 6 percentage lower than using feature augmentation, which again proves paramount.

CONCLUSIONS

In this paper we proposed a machine learning approach to facies classification problem. The proposed algorithm is based on a random forest classifier fed with a set of augmented features. Results on a set of ten wells validate the proposed approach, highlighting the positive impact of the developed feature augmentation strategy. Moreover, results obtained on a blind test performed by Hall and Hall (2017) also confirm a good capability of this algorithm to generalize to new data.

Considering the achieved promising results, future work will be devoted to the analysis of deep learning strategies for feature learning and classification (e.g., convolutional neural networks). Moreover, we will focus on the possibility of adding geological constraints to drive classification with the help of apriori information about rock formation.

ACKNOWLEDGMENTS

The authors would like to thank the organizers of the SEG machine learning constest for the challenging opportunity and Nicola Bienati for the suggestion to investigate this issue.

REFERENCES

Baldwin, J. L., R. M. Bateman, C. L. Wheatley, et al., 1990, Application of a neural network to the problem of mineral identification from well logs: The Log Analyst, **31**.

Bishop, C. M., 2006, Pattern recognition and machine learning (information science and statistics): Springer-Verlag New York, Inc.

Bohling, G. C., and M. Dubois, 2003, An integrated application of neural network and markov chain techniques to prediction of lithofacies from well logs: Technical Report 50, Kansas Geological Survey Open File Report.

Breiman, L., J. Friedman, R. Olshen, and C. Stone, 1984, Classification and Regression Trees: Wadsworth and Brooks.

Busch, J., W. Fortney, L. Berry, et al., 1987, Determination of lithology from well logs by statistical analysis: SPE formation evaluation, **2**, 412–418.

Dubois, M. K., G. C. Bohling, and S. Chakrabarti, 2007, Comparison of four approaches to a rock facies classification problem: Computers & Geosciences, **33**, 599–617.

Friedman, J. H., 2000, Greedy function approximation: A gradient boosting machine: Annals of Statistics, **29**, 1189–1232.

Hall, B., 2016, Facies classification using machine learning: The Leading Edge, **35**, 906–909.

Hall, M., and B. Hall, 2017, Distributed collaborative prediction: Results of the machine learning contest: The Leading Edge, **36**, 267–269.

Ho, T. K., 1995, Random decision forests: Presented at the International Conference on Document Analysis and Recognition (ICDAR).

———, 1998, The random subspace method for constructing decision forests: IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), **20**, 832–844.

Kobrunov, A., and I. Priezzhev, 2016, Hybrid combination genetic algorithm and controlled gradient method to train a neural network: Geophysics, **81**, IM35–IM43.

Rogers, S. J., J. Fang, C. Karr, and D. Stanley, 1992, Determination of lithology from well logs using a neural network (1): AAPG bulletin, **76**, 731–739.

Rokach, L., and O. Maimon, 2005, Top-down induction of decision trees classifiers - a survey: IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) (TSMCC), **35**, 476–487.

Smith, T., and S. Treitel, 2010, Self-organizing artificial neural nets for automatic anomaly identification, *in* SEG Technical Program Expanded Abstracts 2010: Society of Exploration Geophysicists, 1403–1407.

Wolf, M., J. Pelissier-Combescure, et al., 1982, Faciolog-automatic electrofacies determination: Presented at the SPWLA 23rd Annual Logging Symposium, Society of Petrophysicists and Well-Log Analysts.

Zhang, C., C. Frogner, M. Araya-Polo, and D. Hohl, 2014, Machine-learning based automated fault detection in seismic traces: Presented at the 76th EAGE Conference and Exhibition 2014.

Zhao, T., V. Jayaram, A. Roy, and K. J. Marfurt, 2015, A comparison of classification techniques for seismic facies recognition: Interpretation, **3**, SAE29–SAE58.