A practical method for well log data classification

Wawrzyniec Kostorz^{1,2}

Received: 27 March 2020 / Accepted: 29 September 2020 / Published online: 17 November 2020
© The Author(s) 2020

Abstract

In this work, a method for well log data classification is presented. The method relies on a coordinate transformation to restructure the data in an optimal way and a quasi-probabilistic interpolation technique capable of smoothing noisy data. The approach does not require case-specific design, is computationally efficient and provides a statistical characterization of the classification problem. Consequently, transition zones between facies can be modelled in a realistic fashion and intermediate rock types can be identified with ease. Apart from being capable of classifying unseen data with high accuracy, the technique can also be used as an informative quality and consistency assessment tool for manually classified data. The properties of the method are demonstrated on a realistic test case study.

Keywords Classification · Well log · Smoothing · Nearest neighbour

1 Introduction

Well log data classification is an important problem in the field of reservoir modelling. In recent years computer-aided classification has been gaining traction, with a particular interest in machine learning methods. Multiple authors investigated the applicability of machine learning/neural networks to the problem with varying degrees of success [1–6]. Additionally, in 2016 an open competition targeting machine learning methods was organized with over 300 entries from 40 teams according to the official report [7]. A variety of techniques were implemented by the contestants with the best performing ones including boosted trees, random forest, majority voting, multilayer perception, support vector machine, deep/convolutional neural network and nearest neighbours. Boosted trees approach dominated the top part of the leaderboard, with top eight scores all employing the method. A recent 1D-CNN model by Imamverdiyev and Sukhostat [8] deserves particular attention due to its relatively good performance.

However, classifiers relying on machine learning (ML) suffer from a number of typical issues associated with this class of methods, two examples of which are provided. First, ML typically requires the model structure to be provided by the user and the model layout is generally designed via trial and error. Furthermore, an appropriate layout or fine tuning is generally case-specific and does not generalize easily. Consequently, a new model has to be devised for each individual problem. Second, the more general issue of the lack of explainability results in detecting errors and estimating model uncertainty particularly challenging. As a result, model prediction is hoped, rather than trusted, to be accurate. This combined with the case-dependent and manually designed model layout makes alternatives with different properties a potential area of interest.

In this work a well log data classification method addressing the above issues is proposed. On top of serving as a classifier, the technique can also aid the manual classification process by identifying potential errors and inconsistencies, which were indeed found in the data sets used. The design of the method does not rely on fine-tuning parameters and is uniform, i.e. does not depend on the number of logs, samples or rock types to be assigned. Consequently, a single implementation can be applied to an arbitrary data set without any modification. Additionally, the underlying mechanism is intuitively clear and fully explicit, and the results carry a well-defined meaning.

The rest of this manuscript is organized as follows. In Section 2 the classification problem is briefly discussed and

✉ Wawrzyniec Kostorz
w.kostorz16@imperial.ac.uk

¹ Geoscience Research Centre, Total E&P UK, Westhill, Aberdeen, AB32 6JZ, UK

² Department of Earth Science and Engineering, Imperial College London, Prince Consort Road, London, SW7 2BP, UK

the methodology underlying our approach is given. Results are provided in Section 3. Discussion and conclusions follow in Sections 4 and 5.

2 Method

The semi-continuous classification problem considered in this work is of the following form. Let a feature vector $\mathbf{f} = [f_1, \dots, f_m]^T \in \mathbb{R}^m$ be associated with m continuous input features. It is assumed that there exists a (potentially implicit) classification mapping M onto a discrete set C of n distinct classes

$$C = \{c^{(1)}, \dots, c^{(n)}\} \quad (1)$$

formally written as

$$M: \mathbb{R}^m \rightarrow C. \quad (2)$$

The “semi-continuous” nature of the problem means that the feature vector is continuous (allowed to take non-discrete values) while the set of classes is finite (discrete). A typical example of such problem can be found in well log data characterization, where an array of m real-valued measurements (e.g. rock density, porosity, permeability, gamma ray data) maps onto the type of measured rock, such that an example C with $n = 3$ classes could be

$$C = \{\text{shale, coal, sandstone}\}. \quad (3)$$

It is assumed that a collection $\mathcal{S}$ of N sample-response pairs is provided *a priori*, written as

$$\mathcal{S} = \{(\mathbf{f}_i, c_i) | M(\mathbf{f}_i) = c_i\}_{i=1}^N. \quad (4)$$

With the above notation, we aim to construct a data-driven proxy for M from the sample set $\mathcal{S}$. In the following sections we discuss an initial data renormalization procedure, express the problem as multi-dimensional interpolation of a set of indicator functions and provide a geometrical interpretation supporting the discussion.

2.1 Data renormalization

In practice, the entries of a feature vector $\mathbf{f}$ can span multiple orders of magnitude, in particular if the features are independent and correspond to different physical properties of the characterized object. The classification methods proposed later in this work rely on computing distances between two feature vectors, which may introduce a severe bias without an appropriate data treatment. We illustrate this by considering two feature vectors $\mathbf{f}^{(1,2)}$ given below

$$\mathbf{f}^{(1)} = [0.0001, 1000]^T \quad (5)$$

$$\mathbf{f}^{(2)} = [0.0005, 1001]^T. \quad (6)$$

Knowing that the typical size of the first feature is 10^{-4} and of the second feature 10^3 , it is clear that the difference in the first feature is expected to be much more important. However, computing the Euclidean distance between the two vectors in a naive way makes the norm dominated by the difference in the second entry. Consequently, distance-based methods will fail to identify the relatively large relative difference that may be of greater importance.

The above problem can be mitigated easily by employing the proper orthogonal decomposition [9] (POD, also known as principal component analysis, PCA). We first compute the sample mean $\bar{\mathbf{f}}$ defined as

$$\bar{\mathbf{f}} = \frac{\sum_{i=1}^N \mathbf{f}_i}{N} \quad (7)$$

and define the mean-shifted sample array

$$\mathbf{F} = [\mathbf{f}_1 - \bar{\mathbf{f}}, \dots, \mathbf{f}_N - \bar{\mathbf{f}}] \quad (8)$$

with shifted samples stored as columns. Next, the covariance matrix $\mathbf{K}$ is constructed via

$$\mathbf{K} = \frac{\mathbf{F}\mathbf{F}^T}{N-1} \in \mathbb{R}^{m \times m} \quad (9)$$

and eigen-decomposed into

$$\begin{aligned} \mathbf{V} &= [\mathbf{v}_1, \dots, \mathbf{v}_m] \\ \boldsymbol{\sigma} &= [\sigma_1, \dots, \sigma_m]^T \end{aligned} \quad (10)$$

such that

$$\mathbf{K}\mathbf{v}_j = \sigma_j^2 \mathbf{v}_j, j = 1, \dots, m. \quad (11)$$

Note that the eigenvalues of $\mathbf{K}$ are written as σ_j^2 since $\mathbf{K}$ is positive-semidefinite, but their square roots σ_j are of interest. The arrays $\bar{\mathbf{f}}$, $\mathbf{V}$, $\boldsymbol{\sigma}$ are stored as a part of the model. They define the following transformation to be applied to any feature vector (including samples and unseen inputs)

$$\mathbf{f} \rightarrow \hat{\mathbf{f}} = [\mathbf{V}^T (\mathbf{f} - \bar{\mathbf{f}})] / \boldsymbol{\sigma} \quad (12)$$

with the division performed component-wise. The above transformation applied to the set of samples $\mathbf{f}_i$ in $\mathcal{S}$ makes the resulting set $\hat{\mathbf{f}}_i$ spherically symmetric on average, meaning that the variance of the sample set is 1 in all coordinate directions and its mean is equal to $\mathbf{0}$. Equivalently, (12) heuristically equalizes the sensitivities of the output to all inputs. An example of the transformation applied to a realistic data set is given in Fig. 1 below.

2.2 Indicator functions

We first define a collection of n indicator functions as follows

$$I^{(j)}(\hat{\mathbf{f}}) = \begin{cases} 1, & \text{if } M(\mathbf{f}) = c^{(j)} \\ 0, & \text{otherwise} \end{cases} \quad j = 1, \dots, n. \quad (13)$$

Fig. 1 Selection of feature cross-plots before and after the coordinate transformation is applied. Nine different classes are represented by different colours. The cross-plots of the first three (out of seven) original and transformed features are given below and above the

diagonal respectively. Along the diagonal the distributions of the original (dashed) and transformed (solid) features are plotted. Note how the transformation makes the cross-plots above the diagonal spherically symmetric on average, in contrast with the original data

In practical terms, $I^{(j)}$ takes the value 1 precisely in the regions of $\mathbb{R}^m$ mapping onto $c^{(j)}$. Note that full knowledge of $I^{(j)}$ for all j solves the classification problem completely; in a well-posed problem only one of $I^{(j)}$ will return 1 and all others will return 0, uniquely identifying the index j of the object class.

The data-driven classification problem is therefore equivalent to constructing a collection of proxies for $I^{(j)}$ from a discrete set of samples. Sample-response sets for each $j = 1, \dots, n$, written $\hat{S}^{(j)}$, can be generated from $\mathcal{S}$ via

$$\hat{S}^{(j)} = \{(\hat{\mathbf{f}}_i, q_i^{(j)}) | I^{(j)}(\hat{\mathbf{f}}_i) = q_i^{(j)} \in \{0, 1\}\}_{i=1}^N. \quad (14)$$

Additionally, in order to apply a wide range of interpolation methods we allow $I^{(j)}$ surrogates to map onto the full range of $\mathbb{R}$ instead of the binary set $\{0, 1\}$. Consequently, a variety of interpolation methods such as the nearest neighbour (NN, to be discussed shortly), radial basis function (RBF) [10], Gaussian process emulation (GPE) [11] or artificial neural networks (ANN) [12] can be used. Indeed, the formulation given here is the backbone of many ANN architectures for classification problems. As already pointed out, in our

approach interpolation and classification are applied after the transformation (12).

Let data-driven surrogates for $I^{(j)}$ constructed via the interpolation method of choice be written as $\tilde{I}^{(j)}$. After they are constructed, we define the proxy indicator vector

$$\tilde{\mathbf{M}}(\mathbf{f}^*) = [\tilde{I}^{(1)}(\hat{\mathbf{f}}^*), \dots, \tilde{I}^{(n)}(\hat{\mathbf{f}}^*)] = [M_1, \dots, M_n] \quad (15)$$

where $\hat{\mathbf{f}}^*$ is the transformation (12) applied to a new (unseen) feature vector $\mathbf{f}^*$. Since we are allowing for a continuous range of outputs of $\tilde{I}^{(j)}$, the entries of $\tilde{\mathbf{M}}$ will generally be non-binary. In the ideal case, only one of them would be equal to 1 while all others will be 0. In this case, one would typically choose the index of the largest entry of $\tilde{\mathbf{M}}$ to correspond to the index of the predicted class of $\mathbf{f}^*$.

We point out that a number of post-processing methods may be applied to $\tilde{\mathbf{M}}$. For example, all negative entries may be set to 0 and then the entries can be rescaled to add up to 1. Such treatment allows for interpreting them as probabilities, but we note that such interpretation has no rigorous mathematical support, unless a statistics-driven interpolation method is employed. Three examples

illustrating the influence of the interpolation method on the indicator interpolant are provided in Fig. 2 below.

2.3 Smoothed nearest neighbour classifier

The nearest neighbour (NN) interpolation method applied to the collection of samples of indicator functions can be expressed more naturally as the 1-nearest neighbour classifier, namely

$$M(\mathbf{f}) = c_k$$

$$k = \operatorname{argmin}_{i \in \{1, \dots, N\}} \|\hat{\mathbf{f}} - \hat{\mathbf{f}}_i\|, \quad (16)$$

where $\hat{\cdot}$ corresponds to the transformation (12). The 1-nearest neighbour classifier can also serve as the basis for a bit more sophisticated classification tool proposed below.

First, we describe a more involved model evaluation procedure. It is assumed that a training feature set with known classes and a set of (typically unseen) data to be classified are given. In order to classify the unseen data we repeatedly draw a fixed number N_{draw} of feature samples and their corresponding classes from the training set. For each draw, the 1-nearest neighbour classifier based on the N_{draw} random samples is constructed and evaluated, and the result is stored by updating class-specific counters associated with each unseen sample. More precisely, if a data sample is assigned a certain class, then the corresponding class counter (for this data sample) is increased by one. After a sufficient number of random draws and model evaluations all counters are divided by the number of iterations (draws) written N_{iter} . The normalized counters correspond to the likelihoods of each sample belonging to a given class. Additionally, the mean classification accuracy is estimated as the maximal

likelihood associated with a data sample averaged over all unseen samples. As an example, consider an unseen data sample and its four associated counters $M_{1,2,3,4}$, corresponding to four classes in the problem. A possible combination of counters' values after 10 iterations (random draws and the corresponding model evaluations) is $M_1 = 2$, $M_2 = 7$, $M_3 = 1$, $M_4 = 0$ with the sum of all counters necessarily equal to the number of iterations (10). Consequently, the likelihoods of the sample belonging to the three classes are estimated as 0.2, 0.7, 0.1, 0.0 respectively. The maximal value 0.7 is then averaged together with other maximal sample-specific likelihoods to approximate the global accuracy of the classification. The evaluation method proposed can also be interpreted as a smoothed variant of the nearest neighbour interpolation method applied to the set of indicator functions. A visualization of the approach and the influence of the relative sizes of N_{draw} and the total number of samples N in the training set are given in Fig. 3 below. Clearly, for $N_{\text{draw}} = N$ the method reduces to the typical 1-nearest neighbour classifier.

The choice of N_{draw} is a crucial aspect of the model. We propose a method of approximating the value for which the model is the most consistent, namely the value for which the expected prediction accuracy is the closest to the actual value. Clearly, this value can be computed easily if the true classes of the unseen data are known, but such information is obviously not provided in real applications. Instead, we wish to estimate N_{draw} via cross-validations of the training set. In order to do that, two parameters that govern the process should be specified, namely the number N_{test} of training data samples which will mimic unseen data and the number of random draws (iterations) N_{iter} used for the evaluation process described previously. We point out that N_{iter} should be large enough in order

Fig. 2 Interpolants of an indicator function characterized by different interpolation methods. The indicator is sampled to be 1 on magenta points and 0 on green points. The contour of the indicator value 0.5 is plotted in red. The value of the interpolant is modelled on the gray

scale with white/black corresponding to values 1/0 respectively. The interpolation methods used were the nearest neighbour (NN), radial basis function (RBF) and Gaussian process emulation (GPE)

Fig. 3 Interpolants of an indicator function characterized by different values of $N_{\text{draw}}/N = p$. The indicator is sampled to be 1 on magenta points and 0 on green points. The contour of the indicator value 0.5 is plotted in red. Lower values of p smooth the surrogate (a) while higher values provide greater accuracy on the set of samples (c). Intermediate values of p result in partial smoothing and balance

to guarantee a converged result. Its value can either be determined automatically by monitoring convergence over multiple random draws and employing a stopping criterion, or by pre-specifying it to be “sufficiently large” a priori. In our examples the latter approach was taken due to its simplicity; it was verified that $N = O(10,000)$ training samples mapping onto $O(10)$ classes provide a convergent model with $N_{\text{iter}} = 10,000$.

A single estimate of the optimal N_{draw} , written N_{draw}^* is as follows. Given a training set, extract a number N_{test} random data samples and their corresponding classes. Treat the data just extracted as an unclassified set and the remaining data as a new training set. Given a value of N_{draw} , classify the extracted data via the evaluation procedure described earlier, i.e. average over N_{iter} random draws of size N_{draw} . For this value of N_{draw} , record the predicted and the actual classification accuracy on the removed part of the training set, calculated via comparison with the known true classification. Finally, N_{draw}^* is approximated as the value of N_{draw} for which the two accuracies are equal. A visualization of this procedure is given in Fig. 4. Clearly, a single estimate of N_{draw}^* may not be sufficient and a different set of N_{test} extracted data samples can produce a different estimate. For this reason it is better to approximate N_{draw}^* over multiple cross-validations for increased model robustness.

Here N_{test} , the size of the cross-validation sample, is not an independent model parameter and should be specified carefully. N_{test} should correspond to a relatively low fraction of the training set not to bias the training process, while being of order at least $O(100)$ for statistically meaningful results. In our example the training set consists of 3683 data samples and the value $N_{\text{test}} = 400$ was used. Observe

the two effects (b) Smoothing increases model consistency and is generally desirable if the data is expected to be noisy. This is indeed the case for well log data, since the manual classification process, from which the training set is derived, relies on subjective interpretation prone to human error

that since $N_{\text{iter}, \text{draw}}$ can be effectively fixed in advance and do not affect the model as long as their values follow the guidelines given above, the only model parameter to be extracted from the data is N_{draw}^* . The method of evaluating N_{draw}^* is summarized in Algorithm 1 below.

The procedure just described can be regarded as model “training” and has to be performed just once, similarly to many alternative classifiers. The training process also parallelizes naturally to both CPU/GPU architectures which allows for massively reducing the computational time required. The resulting classifier consists of the transformed data set (12), sample mean $\bar{\mathbf{f}}$, rotation matrix $\mathbf{V}$, scaling vector σ and the value N_{draw}^* . While the computational cost of model evaluation is much higher in comparison with many alternatives (e.g. machine learning), it is still practically low; our serial implementation runs in the order of a few seconds.

3 Results

The method was tested on a public data set also employed in [1, 7, 8] where each feature vector consists of seven measurements, namely gamma ray (GR), resistivity log (RL), photoelectric effect (PE), neutron-density porosity difference (DPHI), average neutron density porosity (PHIA), nonmarine-marine indicator (NM/M) and relative position (RP). Nine classes in the data set include nonmarine sandstone (SS), nonmarine coarse silstone (CSiS), nonmarine fine sandstone (FSiS), marine sandstone and shale (SiSh), mudstone (MS), wackestone (WS), dolomite (D), packston-grainstone (PS) and phylloid-algal bafflestone (BS). The total of 3683 data samples from nine out of ten wells

Fig. 4 A method for estimating N_{draw}^* . For a random draw of N_{test} samples from the training set the predicted and actual model accuracies are computed first, for different values of N_{draw} (a). The difference between the two can be approximated via the linear least-squares fit (b). For multiple (here 20) random draws, the individual (green) and the global (magenta) linear fits are constructed (c). N_{draw}^* is

approximated as the root of the average fit (magenta), and the uncertainty of the result, corresponding to the typical accuracy prediction error, is estimated from the individual green least-squares approximations evaluated at N_{draw}^* (d). In this example the expected model accuracy is 73% and the estimate uncertainty is $\pm 3\%$ with 95% confidence

(NOLAN, SHANKLE, SHRIMPLIN, NEWBY, CHURCHMAN BIBLE, CROSS H CATTLE, KIMZEY A, LUKE G U and Recruit F9) were used for model training and evaluation. The data from the well ALEXANDER D was omitted since it was found that it was of very low quality and consistency. Well log data for the NOLAN well is given in Fig. 5 below. The value of $N_{draw}^* =$

680 (18.5% of the data) was estimated via the procedure given earlier and the value $N_{iter} = 10,000$ was used for the evaluation procedure. The accuracy measure in this work was the class-specific and micro- and macro-averaged F_1 score, which is a commonly used metric for classification problems [1] and which we briefly outline below.

Fig. 5 Well log data from the NOLAN well. The colour legend is provided at the top and the manual classification just below. Seven logs, from top to bottom, are RP, NM/M, PE, PHIA, DPHI, RL, GR. Depth along the well increases to the right in the above plots

Algorithm 1: Evaluation of the optimal value of the draw size N_{draw}^* .

Data: Classified and transformed data

$$\mathcal{S} = \left\{ (\hat{\mathbf{f}}_i, M(\hat{\mathbf{f}}_i)) \right\}_{i=1}^N$$

Result: Optimal value of the draw size N_{draw}^*

- 1 Specify the iteration count $N_{\text{iter}} \geq O(N)$ and cross-validation sample size $N_{\text{test}} \sim N/10$.
- 2 Specify the number of independent estimates of N_{draw}^* , N_e . At least 10 estimates are recommended; hundreds are desirable, especially in parallel implementations.
- 3 Specify a number of $K = O(10 - 100)$ equispaced values $1 \leq N_1 \leq N_2 \leq \dots \leq N_K \leq N$.
- 4 Set $N_{\text{draw}}^* \leftarrow 0$
- 5 **for** $m = 1, \dots, N_e$ **do**
 - 6 Split $\mathcal{S}$ into a random cross-validation sample $\mathcal{S}_0$ of size N_{test} and $\mathcal{S}' = \mathcal{S} \setminus \mathcal{S}_0$.
 - 7 **for** $k = 1, \dots, K$ **do**
 - 8 Classify all samples in $\mathcal{S}_0$ via reference to the data in $\mathcal{S}'$ with the draw size $N_{\text{draw}} = N_k$ and the value of N_{iter} fixed previously. For each value of N_k record the average classification accuracy $A_k = n^{(k)}/N_{\text{test}}$ where $n^{(k)}$ is the number of samples in $\mathcal{S}_0$ classified correctly. Similarly, record the average model confidence (predicted model accuracy) C_k , equal to the probability of the top model prediction (probability of the most likely class as predicted by the model) averaged over all samples in $\mathcal{S}_0$.
 - 9 **end**
 - 10 Given $\{(N_k, A_k, C_k)\}_{k=1}^K$, estimate N_{loc} where the difference $A_k - C_k$ crosses zero, e.g. via least-squares interpolation (Fig. 4).
 - 11 Update $N_{\text{draw}}^* \leftarrow N_{\text{draw}}^* + N_{\text{loc}}$
 - 12 **end**
 - 13 Update $N_{\text{draw}}^* \leftarrow \lfloor N_{\text{draw}}^*/N_e \rfloor$

Let a collection of data classified both manually (assumed to be fully correct) and by a numerical model be given. Class-specific F_1 scores are computed by first evaluating true positives (TP) and false positives (FP) and negatives (FN). Class-specific TP is the number of instances the model correctly predicts the given class. Class-specific FP is the number of times the model predicts the given class while the correct answer is different. Similarly, FN is the number of times the class should be

predicted by the model but is not. Given TP , FP and FN for each class, the class-specific F_1 score is defined as

$$F_1 = \frac{2TP}{2TP + FN + FP}. \quad (17)$$

The macro-averaged F_1 is the average of three class-specific scores weighted by the number of instances of corresponding classes in the correctly (manually) classified set. The micro-averaged F_1 score is equal to the typical average classification accuracy, i.e. the ratio of the sum of all class-specific TP divided by the total number of classified samples.

First, we the consistency of the data set was assessed via a cross-validation study. All training data was classified via reference to itself according to our model in order to assess the amount of inconsistency (noise) in the training set. A simple safety mechanism was implemented, ensuring that each data sample is never classified via reference to itself but only via reference to all other samples. In practical terms, whenever a training sample $\hat{\mathbf{f}}_j$ is to be classified, N_{iter} random subsets of size N_{draw}^* are drawn from all training data excluding $\hat{\mathbf{f}}_j$. Generally, an inconsistency is characterized by the model predicting a certain class with high likelihood meaning that many similar (neighbouring) samples were classified as this particular class, while the actual class (assigned manually) is different. In Fig. 6 an example model inconsistency is visualized, where the likelihood profiles of rock types predicted by the model (solid, thick) are plotted together with the “true” and predicted classifications. Only a small fraction of the training set is visualized for clarity of presentation. In Figs. 6 and 7 the idealized (rounded, binarized) values of the indicator functions based on model prediction are plotted as thin solid lines, the regions in which manual and data-driven classifications differ are marked in red and the expected accuracy (maximal sample-specific likelihood) is plotted in black. The second most likely prediction is visualized in the bottom part of the “Model” row with the relative sizes of the top and second likelihoods respecting the data.

The summary of the cross-validation study is given in Table 1 below. The micro-averaged F_1 score (average prediction success) increased to 91.88% if the second-best guess of the model (prediction with the second highest likelihood) is included and if its likelihood is at least 50% of the likelihood of the best prediction. More details on this approach and a comparison with an alternative are provided in the next section.

Second, the classifier was applied to unseen data from two wells, STUART and CRAWFORD. A part of the solution with another data inconsistency is provided in Fig. 7 below. Again, only a small fraction of the prediction set is visualized for illustrative purposes. The summary

Fig. 6 Example inconsistency in the training data set. The last peak in the BS class is classified with around 90% (high) confidence, while the manual classification corresponds to PS, here classified with 0% confidence. Observe that the positions of the first interface between

BS/PS and the interface between SS/CSiS are not defined precisely, since there is a transition zone between the two facies where the corresponding likelihoods vary slowly

of the prediction study is given in Table 2. The micro-averaged F_1 score (average prediction success) increased to 87.35% if the second-best guess of the model is included

and if its likelihood is at least 50% of the likelihood of the best prediction. An accuracy comparison of our method, the approach by Imamverdiyev and Sukhostat [8] (1D-CNN)

Fig. 7 A part of the model prediction vs the manually classified data. The peak in the likelihood profile of dolomite (D, light blue) is very consistent with the training data since its corresponding likelihood values are very close to unity. However, manually classified data predicts

the rock type in this region to be nonmarine fine sandstone (FSiS, dark orange) which is not at all consistent with the training set, indicating a potential problem to be examined by hand

Table 1 Class-specific and macro- and micro-averaged cross-validation results. The average confidence level (micro F_1 score) was estimated a priori by the model (Fig. 4a, d) at $69.73 \pm 3\%$ with 95% confidence

Facies									Average	
SS	CSiS	FSiS	SiSh	MS	WS	D	PS	BS	Macro	Micro
76.19%	76.65%	76.17%	72.76%	55.05%	65.41%	70.73%	69.85%	76.02%	71.82%	71.98%

and top four contestants from the 2016 competition is given in Fig. 8.

4 Discussion

The method presented has two main applications. Aside from its primary function, namely predicting classes for a set of unseen data, it is capable of identifying and extracting inconsistencies from the training set. A simple demonstration (Fig. 6) was given in order to show that certain samples in the training data are incompatible with the rest. The same problem is present in the prediction set (Fig. 7), where data samples classified by the model with very high confidence were identified manually as classes with the corresponding likelihood equal to zero. Since high likelihoods indicate that a large number of similar samples was identified as a certain class in the training set, we conclude that the training and evaluation sets are not fully compatible. Such inconsistencies can be attributed to the manual classification process relying on subjective interpretation and often being performed on a locally average rather than point-by-point basis. As a result, multiple types of error are introduced, some detail may be lost and the training set becomes noisy. However, by employing the model proposed for “full” cross-validation (classifying all training data via reference to itself), severe inconsistencies can be targeted automatically and then reevaluated manually in order to improve data quality. We note that since both the training and validation data sets are noisy and erroneous, models that can match the validation data with high accuracy may in fact not be optimal as a consequence.

An collection of likelihood profiles (Figs. 6 and 7) derived from our model are generally much more informative than a simple discrete prediction. As an example, it was demonstrated that they have the capability of modelling

transition zones between different rock types in a realistic fashion. In practice, transitions from one rock type to another may be incremental over a finite depth, which is reflected by the character of the overlapping class-specific likelihood profiles (Figs. 6 and 7) and is not captured by classifiers assigning a discrete rock type to a feature vector. Sharp transitions are realistic as well and are also observed in the likelihood profiles (e.g. BS/FSiS interface in Fig. 6). Additionally, it was seen that certain facies often cannot be identified uniquely, which is particularly apparent for the three types of sandstone. On many instances all three sandstone types differing in grain size are recognized by the model with similar likelihoods (Fig. 7), suggesting that the underlying rock is of an intermediate type. A potential underlying cause of this effect is the gamma ray data due to it being a proxy for the grain size which may span a range rather than just three discrete values. Finally, examining full profiles instead of likelihoods at a single point may provide additional insight. For example, a peak in a certain profile coinciding with a dip in another one may indicate that the class associated with the peak is the right classification even if the peak value is (mildly) lower than the dip value.

Our method shows good F_1 accuracy in the prediction scenario. With the naive approach, i.e. classifying the rock according to the highest class likelihood, the accuracy reached 70.12%. This result is significantly better than multiple machine learning alternatives [7], but is bettered by the 1D-CNN approach by Imamverdiyev and Sukhostat (76.87%) [8]. A common method of evaluating well log data classifiers is to take into account adjacent facies [1, 7, 8]. Instead, we employed a simpler approach where the “second order correction” comes from the model’s second statistically best guess at any position instead of model’s best guess at neighbouring positions. With this approach, which is conceptually simpler and trivial to implement, the prediction accuracy increased to 87.35%.

Table 2 Class-specific and macro- and micro-averaged prediction results. The average confidence level (micro F_1 score) was a priori estimated by the model (Fig. 4a, d) at $66.35 \pm 3\%$ with 95% confidence

Facies									Average	
SS	CSiS	FSiS	SiSh	MS	WS	D	PS	BS	Macro	Micro
45.45%	81.41%	58.54%	67.65%	50.00%	67.55%	53.66%	80.66%	70.59%	69.95%	70.12%

Fig. 8 Prediction accuracy in comparison with 1D-CNN and the top four contestants from the 2016 competition

There are a number of factors that should be taken into account for a more practically relevant evaluation of our and any other approach. As pointed out earlier our results suggest that there is a degree of (possibly human) error in both training and prediction sets. Consequently, matching the manual prediction perfectly may actually be suboptimal. Moreover, on multiple occasions the prediction error is due to interfaces between facies being shifted or intermediate rock types; the binary nature of the evaluation process is not necessarily an informative measure in such cases. Furthermore, our method does not require fine tuning or case-specific design—for an arbitrary number of features and classes the model structure and training are unchanged. This is in contrast to CNN and many other machine learning methods, where model layout is a heuristic with a massive impact on accuracy. The optimal layout, usually designed manually, is generally problem-specific and may not be possible to specify in real applications.

We conclude this section by proposing a practical approach to well log data classification. The workflow can be summarized in the following steps:

1. Construct a training set manually (team of experts), construct the coordinate transformation (Section 2.1), transform the training data and evaluate N_{draw}^* (Section 2.3).
2. Apply the smoothed nearest neighbour classifier in a “full” cross-validation study. Employ a safety mechanism so that each training sample is never classified via reference to itself, i.e. the random subset of N_{draw}^* samples never includes the sample to be classified. Identify, analyze and manually fix potential inconsistencies in the initial classification
3. Recompute N_{draw}^* for the corrected set of training data
4. Apply the resulting smoothed nearest neighbour classifier to any (transformed) unseen data and generate class-specific likelihood profiles
5. Examine the class-specific profiles manually and generate a detailed well profile including rock types, transition zones, intermediate facies and quality of prediction. Optionally, automatically generate the “first order” approximation, i.e. assign discrete classes corresponding to the highest likelihood

5 Conclusions

A practical method for well log data classification was proposed and evaluated. The method is characterized by a uniform design applicable to an arbitrary number of logs (features) and rock types (classes). Instead of providing just a discrete estimate for each class, it also characterizes data via likelihood profiles. The latter were shown to have the potential to identify non-trivial features such as transition zones and intermediate rock types, and can be proposed as a practical and more informative alternative to the typically used discrete solution to the classification problem. However, a more detailed study is required to systematically assess the method’s capability to recognize and model non-trivial features accurately.

The method demonstrated good performance on a fairly complex realistic test case. The micro-averaged F_1 score of 70.12% is comparable with top-performing machine learning alternatives. However, the simple measure used was demonstrated not to be necessarily representative due to inconsistencies in the training and prediction sets, identified with the aid of our technique. As an example, the 1D-CNN model managed to better our accuracy but relies on erroneous training and prediction sets and consequently the comparison with our method is not fully conclusive. Its architecture (the number of layers etc.) also cannot be argued to be optimal for any other well log data classification problem with a different number of features, classes and data samples.

It was demonstrated that the method can also serve as a consistency/quality check and indicate which data samples should potentially be reevaluated. Due to its universal design, fully explicit and intuitive model nature, a more sophisticated nature of the output and multi-purpose character our approach has the potential to be a practical tool. This is in strong contrast with many machine learning alternatives, which typically provide a manually fine-tuned, black-box-like and case-specific optimal solution that does not generalize in a straightforward way.

The code developed for this work was implemented in MATLAB and executed in serial. For the data set used the training (evaluation of N_{draw}^*) took approximately 40

minutes and model cross-validation and prediction were evaluated in the order of seconds. Both times can be reduced by employing parallelization which applies in a straightforward fashion.

Acknowledgements The author would like to thank Total E&P UK for permission to publish this work. Additional thanks go to Yan Chen, Fahmi Bahrini and Nick Hayward for providing valuable input, feedback and discussions.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Hall, B.: Facies classification using machine learning. *Lead. Edge* **35**, 906–909 (2016)
- Dubois, M.K., Bohling, G.C., Chakrabarti, S.: Comparison of four approaches to a rock facies classification problem. *Comput. Geosci.* **33**, 599–617 (2007)
- Qi, L., Carr, T.: Neural network prediction of carbonate lithofacies from well logs, Big Bow and Sand Arroyo Creek fields, Southwest Kansas. *Comput. Geosci.* **32**, 947–964 (2006)
- Saljooghi, B., Hezarkhani, A.: Comparison of wavenet and ann for predicting the porosity obtained from well log data. *J. Pet. Sci. Eng.* **123**, 11 (2014)
- Sfidari, E., Kadhodaie, A., Rahimpour-Bonab, H., Soltani, B.: A hybrid approach for litho-facies characterization in the framework of sequence stratigraphy: a case study from the south pars gas field, the persian gulf basin. *Journal of Petroleum Science and Engineering*, 121 (2014)
- Chehrizi, A., Rezaee, R.: A systematic method for permeability prediction, a Petro-Facies approach. *J. Petroleum Sci. Eng.* **82–83**, 1–16 (2012)
- Hall, M., prediction, B.Hall.: Distributed collaborative: results of the machine learning contest. *Lead. Edge* **36**, 267–269 (2017)
- Imamverdiyev, Y., Sukhostat, L.: Lithological facies classification using deep convolutional neural network. *Journal of Petroleum Science and Engineering*, 174 (2018)
- Awrejcewicz, J., Kryśko, V.A., Vakakis, A.F.: *Order Reduction by Proper Orthogonal Decomposition (POD) Analysis*, pp. 177–238. Springer, Berlin (2004)
- Hardy, R.: Multiquadric equations of topography and other irregular surfaces. *J. Geophys. Res.* (1896-1977) **76**(8), 1905–1915 (1971)
- Jones, D., Schonlau, M., Welch, W.: Efficient global optimization of expensive black-box functions. *J. Glob. Optim.* **13**, 455–492, 12 (1998)
- Llanas, B., Sainz, F.J.: Constructive approximate interpolation by neural networks. *J. Comput. Appl. Math.* **188**(2), 283–308 (2006)

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.