Introduction

Data quality today remains the main problem which complicates the usage of machine learning methods. Well logs can be recorded at different times and drilling conditions by different companies and tools within beds with different geological settings. This situation makes data preprocessing step, including handling missing data, input anomalies detection and standardization of well logs a crucial and very time-consuming part of any statistical model application. In order to accelerate machine learning application for real geological tasks appropriate tool for well log data preprocessing is needed.

The main goal of this work is to develop a systematic approach to work with raw well log data. To accomplish this goal, we propose to fit a simple unsupervised generative model to the input data and automate the preprocessing step using the generative model. This approach allows to detect the anomalies in the data as the regions that the model struggles to explain (i.e., samples with extremely low likelihood), infer approximations to the missing features using the Bayes rule and incorporate additional expert knowledge in the design of the model. The contributions of this work are:

- the design of a generative model based on a hidden Markov model which is capable for well log data normalization based on particular well and geological conditions,
- the development of sub-routines for anomaly detection, data imputation and the development of heuristics to facilitate the training of the generative model,
- application of the developed model on a real problem of net pay thickness autointerpretation, where we compare the effect of different preprocessing schemes on the performance of a supervised classifier.

Geological settings and input data

As a base for the presented research, we chose one of the Western Siberia mature oilfields located at Khanty-Mansiisk autonomous district and operated by Gazpromneft. The oilfield is characterized by the complex geological setting and long life during which well log data were produced by different companies with different tools and their calibration schemes. Target formation was deposited in a shallow marine environment during a transgressive system tract and composed of lithologies from medium and fine-grained sand to siltstone and mudstone. Each of sub-beds and depositional zones has different geological, geophysical and petrophysical properties without any stable marker bed which can be used as a reference for normalization. As a result, each well needs individual attention during the data preprocessing step.

Input data included the same set of logs for each of more than 350 wells (GR, SP, neutron, ILD, LLD) recorded by a different tool in different scales but with the same physical meaning.

Method and Theory

In this work, we chose the hidden Markov model as a generative model for the raw input data. The temporal structure of HMM allows to capture the vertical continuity of the observed data within the well, and discrete latent variables can act as a proxy for lithological facies. Eidsvik et al. (2004) applied hidden Markov models to infer geological attributes from a well log and Schumann (2002) proposed a well-log classification algorithm that uses the hidden states of HMM. Lindberg and Grana (2015) showed that the vertical continuity of hidden models can lead to improved facies classification results.

Unlike the previous work, we use the hidden Markov model only as a tool for data preprocessing. Besides capturing the continuity of well logs, HMMs have other appealing characteristics. First, for a Gaussian observation model used in Eidsvik et al. (2004) one can infer hidden variables based only on a subset of logs using marginals of multivariate Gaussian distribution. Then, one can approximate the missing logs based on the observation model and the inferred latent variables. This provides us with a

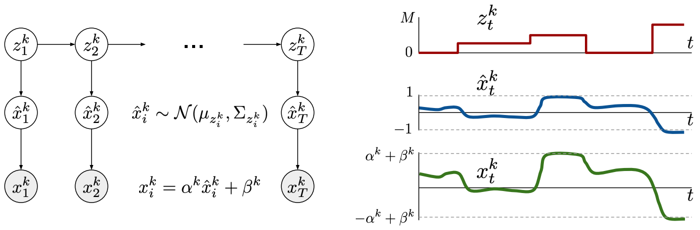

Figure 1 *Left: graphical model for the modified HMM. Grey circles represent the observed uncalibrated variables. Right: schematic representation of the generative process for data, calibrated data (blue) is produced using the hidden variables (red) and then we observe the uncalibrated data (green) obtained after a linear transformation.*

tool for data imputation that allows applying learning algorithms even on partially missing data. Second, the observation model can be further extended to incorporate additional knowledge. In our case, the logs throughout the data did not have a fixed measurement scale, and we modified the default HMM model to include log calibration as a step of the generative process (see Figure. 1).

Below we describe the proposed model and the heuristics for training in detail.

For each well $k \in \{1,\ldots,K\}$ we introduce a sequence of M-valued discrete latent variables $Z^k = (z_1^k, z_2^k, \ldots, z_{T_k}^k)$ to encode the essential soil properties along the well. We assume that for each well the latent variables form a Markov chain with initial probability distribution π, $p(z_1^k = i) = \pi_i$ and transition probabilities T, $p(z_t^k = i \mid z_{t-1}^k = j) = \mathrm{T}_{ij}$. We then assume that there exists a reference measurement scale, in which the logs follow Gaussian distribution with the parameters specified by the latent variable z_t^k: if $z_t^k = m$, $\hat{x}_t^k \sim \mathcal{N}(\mu_m, \Sigma_m)$. Here the hat sign in $\hat{x}_t^k$ indicates the reference measurement scale and $\mu_m \in \mathbb{R}^d, \Sigma_m \in \mathbb{R}^{d\times d}$ are the model parameters. Even though the reference measurement scale is unknown, we assume that it is constant within a well and can be recovered as a componentwise linear transformation. In other words, there exists a set of calibration parameters $\alpha^k, \beta^k \in \mathbb{R}^d$ such, that for the observed logs $X^k = (x_1^k, x_2^k, \ldots, x_{T_k}^k)$ holds $x_t^k = \alpha^k \odot \hat{x}_t^k + \beta^k$. As a consequence, if $z_t^k = m$, the observed value x_t^k has Gaussian distribution with mean $\alpha^k \odot \mu_m + \beta^k$ and covariance matrix $\mathrm{diag}(\alpha^k)\Sigma_m \,\mathrm{diag}(\alpha^k)^T$. For the model parameters $\Theta = \left(\pi, \mathrm{T}, \{\alpha^k, \beta^k\}_{k=1}^K, \{\mu_m, \Sigma_m\}_{m=1}^M\right)$ the resulting joint of the model likelihood is

$$p(X,Z|\Theta) = \prod_{k=1}^{K} \left[\left(\pi_{z_1^k} \prod_{t=2}^{T_k} \mathrm{T}_{z_t^k, z_{t-1}^k} \right) \times \prod_{t=1}^{T_k} \mathcal{N}\left(x_t^k \mid \alpha^k \odot \mu_{z_t^k} + \beta^k, \mathrm{diag}(\alpha^k)\Sigma_{z_t^k} \,\mathrm{diag}(\alpha^k)^T \right). \right] \qquad (1)$$

To tune the model parameters we use the Baum-Welch algorithm to compute a lower bound on the marginal likelihood $\log p(X|\Theta)$ and then apply a gradient ascent optimization scheme to maximize the lower bound with respect to Θ.

By design, the optimal model parameters are not unique. Indeed, the calibration parameters (α^k, β^k) can be adjusted to different choices of the reference measure scale. For example, the change of scale for the Gaussian parameters $(\mu_m, \Sigma_m) \to (C\mu_m, C^2\Sigma_m)$ together with the calibration parameters update $(\alpha^k, \beta^k) \to (\frac{\alpha^k}{C}, \beta^k)$ has no effect of the joint likelihood and, as a result, the model outputs. We do not put any restrictions on the reference measure scale during training and adjust it after the training.

The training procedure is prone to producing sub-optimal models. Therefore, to avoid poor local optima during the training, we initialize the calibration parameters α^k and β^k with the standard deviation and mean of logs computed across the k-th well. Additionally, to initialize μ_m we apply mean-std scaler to logs and then run K-means on the standardized log values. We use cluster means as the initial values of μ_m and matrix $\sigma^2 I$ as the initial values of Σ_m for a small σ.

Results

As we mentioned in the introduction, well logs data is not clean, and the set of performed measurements can be different in different wells. As a first step, we trained the model on a set of wells for which all five well logs under consideration present. Figure 2 presents an example with results for one of them.

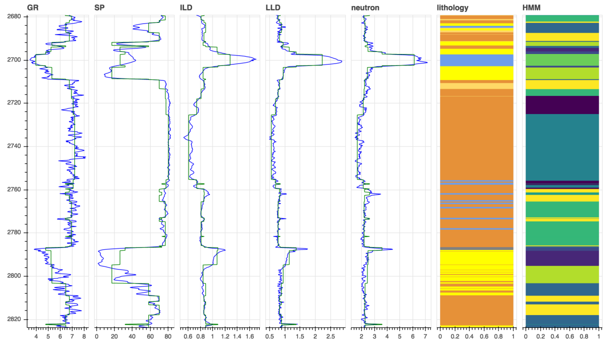

Figure 2 *Example of fitted well. Blue - input well logs, green - data generated by model, lithology - human interpretation, HMM - hidden states from HMM model.*

The number of hidden states in Hidden Markov model is a tunable hyperparameter. We tested different values in the range from 5 to 100. Choice of the number of states exhibits a trade-off: a larger number of hidden states results in a better fit of the data but leads to less interpretable states. Experiments showed that good numbers are 10 for more interpretable model and 30 for a more accurate model. Numbers of states beyond 30 did not lead to notable improvements.

We used two methods to measure the quality of the trained model. Firstly, it is natural to expect that hidden states learned by the model should correspond to some lithological properties. Therefore we compared the states with lithology labels from human interpretation. Contingency matrix for this comparison is presented in figure 3. There is no one-to-one correspondence because the number of lithology types and hidden states is different, but in general, the correspondence looks quite good.

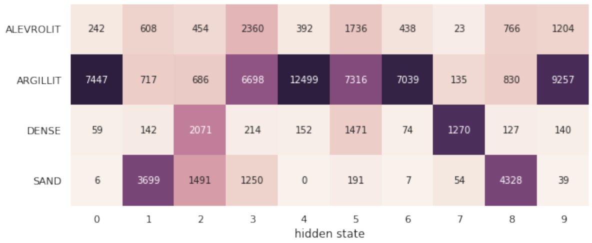

Figure 3 *Contingency matrix between human lithology labels and hidden states from model. Number in each cell represents number of data points which fall in it.*

Additionally, using the generative model, we recovered the values of the well logs from the learned hidden states. For a given hidden state m, we chose the value of well log to be equal to $\alpha^k \odot \mu_m + \beta^k$. Comparison of values generated by the model with true values is a good test of model fit. We used the coefficient of determination R^2 as a metric for this comparison. Table 1 presents calculated R^2 for the considered logs. The average coefficient of determination across all logs was $R^2 = 0.79$.

well log	GR	SP	ILD	LLD	neutron
R^2	0.68	0.87	0.91	0.78	0.72

Table 1 *Coefficient of determination for different well logs.*

One of the benefits of this approach is that the trained model can be applied to automatic detection of anomalies in the input well logs. Poor fit of the data for a given well compared to other wells strongly indicates that the well contains an anomaly. If we look at per well R^2 (figure 4), we can see that there are several wells with particularly low metric value. Manual inspection of these wells showed that these cases are real anomalies. Severe anomalies can hurt further modeling and should be discarded or handled manually.

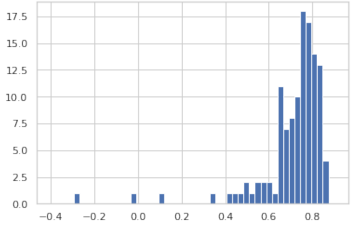

Figure 4 *Distribution of per well coefficient of determination R^2*

We further tested the model as a preprocessing step for the task of net pay intervals classification described in Belozerov et al. (2018). The task is to predict for each point in a well whether it belongs to net pay interval or not. We compare two approaches to the preprocessing. The first is per well scaling using the mean and the standard deviation of each curve. The second is obtaining calibration parameters α^k and β^k from the HMM for rescaling. Recurrent neural networks (GRU - Gated Recurrent Unit) were used for classification because they are suitable for such structured as a sequence data and show better performance than more classical non-deep learning models. Due to the imbalance of the target classes, we use F-score (harmonic mean of precision and recall) to measure the performance. In our experiments, rescaling with HMM lead to an improvement in performance. F-score of a simple scaler was 0.72, and for HMM F-score was 0.74.

Finally, we applied the model for data imputation. Some of the wells in the data did not contain ILD and LLD logs. Nevertheless, the HMM model can infer hidden states in these wells only from the observed logs and fill missing curves from using the inferred states. We tested this approach on the same task of net pay intervals classification. F-score on wells with missing logs was 0.37 for training without imputation. It improved to 0.56 after imputation step, showing the potential of such an approach.

Conclusions

In this work, we applied a generative model based on Hidden Markov model to tasks of well log data processing. The proposed algorithm showed capability for the solution of a wide range of problems related to logging data which previously were time-consuming and in many cases could not be automated. These tasks include well logs reconstruction, data normalization, anomaly detection.

Automating of pointed out tasks by the presented approach can dramatically increase the speed of research and application of machine learning models for well log data and improve its economic effectiveness.

References

Belozerov, B., Egorov, D., Bukhanov, N., Zakirov, A., Osmonalieva, O., Golitsyna, M., Reshytko, A., Semenikhin, A., Shindin, E. and Lipets, V. [2018] Automatic Well Log Analysis Across Priobskoe Field Using Machine Learning Methods. *Society of Petroleum Engineers 18RPTC*.

Eidsvik, J., Mukerji, T. and Switzer, P. [2004] Estimation of geological attributes from a well log: an application of hidden Markov chains. *Mathematical Geology*, **36**(3), 379–397.

Lindberg, D.V. and Grana, D. [2015] Petro-elastic log-facies classification using the expectation–maximization algorithm and hidden Markov models. *Mathematical Geosciences*, **47**(6), 719–752.

Schumann, A. [2002] Hidden Markov models for lithological well log classification. *Terra Nostra*, **4**, 373–378.