

# Machine-learning Based Automated Fault Detection in Seismic Traces \*

Chiyuan Zhang and Charlie Frogner (MIT),  
Mauricio Araya-Polo and Detlef Hohl (Shell International E & P Inc.)

June 19, 2014

## Introduction

The Initial stages of velocity model building (VMB) start off from smooth models that capture geological assumptions of the subsurface region under analysis. Acceptable velocity models result from successive iterations of human intervention (interpreter) and seismic data processing within complex workflows. The interpreters ensure that any additions or corrections made by seismic processing are compliant with geological and geophysical knowledge. The information that seismic processing adds to the model consists of structural elements, faults are one of the most relevant of those events since they can signal reservoir boundaries or hydrocarbon traps. Faults are excluded in the initial models due to their local scale. Bringing faults into the model in early stages can help to steer the VMB process.

This work introduced a tool whose purpose is to assist the interpreters during the initial stages of the VMB, when no seismic data has been migrated. Our novel method is based on machine learning techniques and can automatically identify and localize faults from not migrated seismic data. Comprehensive research has targeted the fault localization problem, but most of the results are obtained using processed seismic data or images as input (Admasu and Toennies (2004); Tingdahl et al. (2001); Cohen et al. (2006); Hale (2013), etc). Our approach suggests an additional tool that can be used to speed up the VMB process.

Fully automated VMB has not been achieved because the human factor is difficult to formalize in a way that can be systematically applied. Nonetheless, if our framework is extended to other seismic events or attributes, it might become a powerful tool to alleviate interpreters' work.

## Method and/or Theory

We pose two problems: Detection of the presence of faults in seismic traces, and localization of the fault (orientation and offset). We take a machine learning approach: We generate a set of seismic traces from velocity models containing faults with varying locality and other properties, then use these known examples to train a machine learning algorithm to identify the presence or locality of a fault in new, previously unseen traces.

The workflow is shown in Fig. 1. We first generate a set of data for training the machine learning algorithm. This training dataset contains seismic traces that are generated from a set of known velocity models, with varying properties (e.g. layer widths). To train the system, we pair each seismic gather in the training dataset with a ground truth label of the desired property (e.g. presence of a fault), taken from the velocity model – the machine learning algorithm uses these pairs of data to learn a function that identifies the desired property, given only the seismic trace. We evaluate the algorithm's performance on a test dataset, in which the ground truth labels are known but hidden from the algorithm.

---

\*Proceedings of 76th EAGE Conference and Exhibition, Amsterdam, 2014

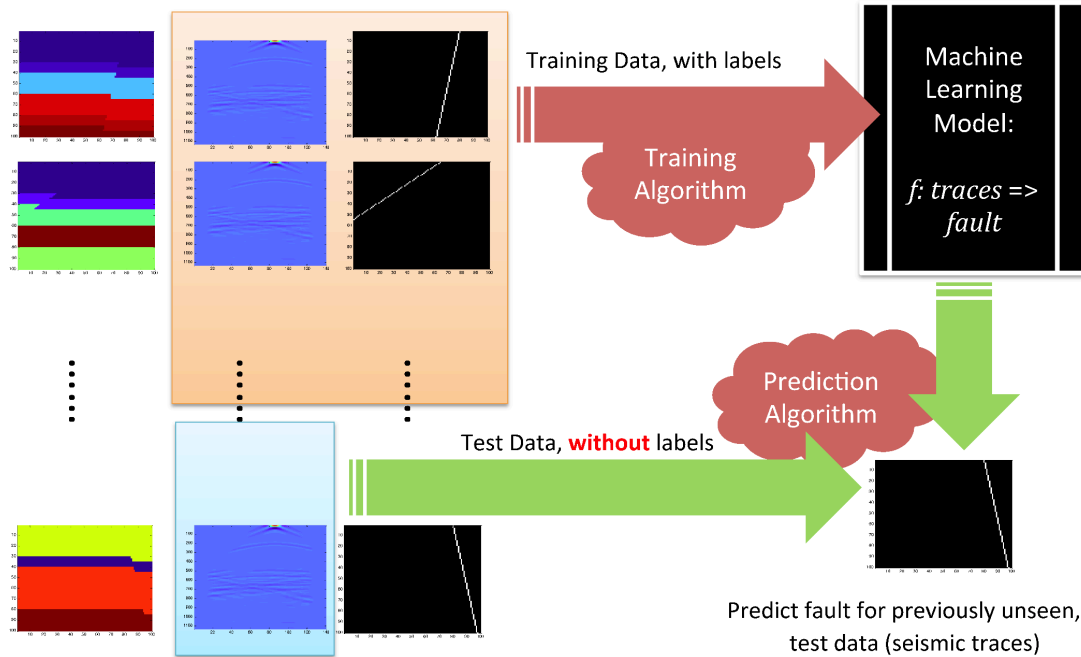


Figure 1: Workflow

We demonstrate our workflow using a well-studied machine learning algorithm: Kernel Regularized Least Squares (Kernel RLS) (Evgeniou et al. (2000); Bishop (2006)). In Kernel RLS, we treat the training data as pairs of observed data  $x$ , being seismic traces, and labels  $y$  that we would like to learn to predict –  $y$  in our case will be either a binary label for the presence of a fault in the underlying velocity model, or a two-dimensional real-valued label for the orientation and offset of the fault. Kernel RLS learns a function  $f^*$  that will map any observed data  $x$  to a prediction of the corresponding label  $y$ , by minimizing the regularized squared error over the data  $\{(x_i, y_i)\}_{i=1}^n$  in the training set:

$$f^* = \operatorname{argmin}_{f \in \mathcal{H}} \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \|f\|_{\mathcal{H}}^2 \quad (1)$$

Choosing an appropriate space of functions  $\mathcal{H}$  for learning is integral to building the machine learning system. In Kernel RLS this function space is defined by a similarity kernel between the input data points  $x_i$  (seismic gathers). Prior work has developed a variety of kernels for both continuous and discrete data (Bishop (2006)). In our case, we choose a general class of nonlinear functions, defined by the Gaussian kernel:

$$\mathbf{K}(x, x') = \exp \left\{ -\frac{1}{\sigma} \|x - x'\|_2^2 \right\}$$

where  $x$  and  $x'$  are any two observed data points, and  $\mathbf{K}(x, x')$  is the derived similarity between them. The bandwidth parameter  $\sigma$  is optimized separately from  $f^*$ , by cross-validation.

### Examples

We validate the framework on 2D synthetic data. Specifically, we first generate random layered velocity models, on each model, a *Ricker wavelet* of 20 Hz is used as perturbation and a *4th order Finite Difference* scheme to solve the isotropic acoustic wave equation. Receivers (17 per model) are evenly placed on the surface of the model. Fig. 2 shows an example of velocity model and synthetic seismic traces for a given shot.

Velocity models are generated with number of layers ranging from 2 to 5. The velocity ranges from 2000 to 4000 and increases with the depth. A diagonal interface with angle ranging from  $20^\circ$  to  $160^\circ$  is randomly introduced as fault representation.

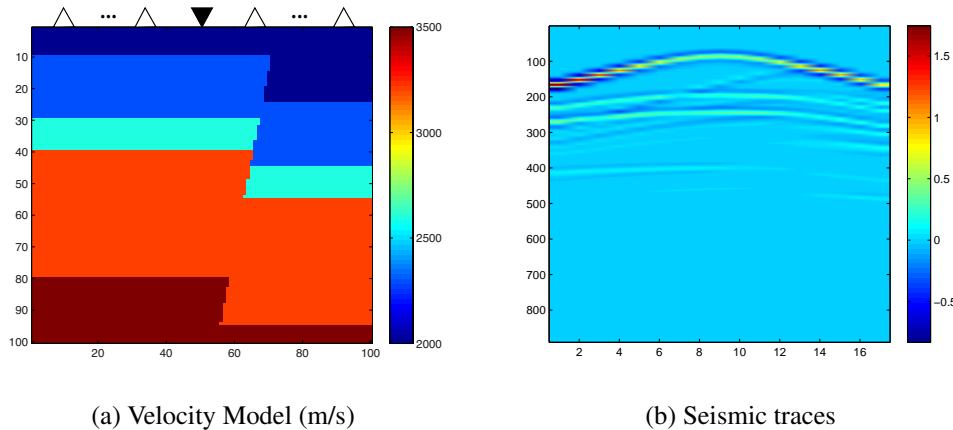


Figure 2: A random generated velocity model and its corresponding seismic trace for a shot at the middle of the surface.

In the basic scenario, each horizontal reflector is flat and evenly distributed along the depth, and the velocity in each layer linearly increases with depth. To make the velocity model more geologically realistic, we change the following elements of the basic case:

1. **Random Thickness:** horizontal reflectors are placed randomly, therefore creating layers with random thickness.
2. **Random Velocity:** the velocity in each layer is still monotonically increasing with the depth but starting from a random selected value.
3. **Swapping Neighboring Layer:** We randomly swap two neighboring layers because this pattern can be observed in real geological data.
4. **Normal / Reverse Fault:** By allowing random displacement including both positive and negative values, we simulate both normal and reverse faults in our velocity models. The displacement values are 10, 15, -10, -15 grid points.

In total, we generate approx **1000** basic models and up to **10000** more complex models, that are based on variations of the described features. For all of them seismic traces are collected.

## Results

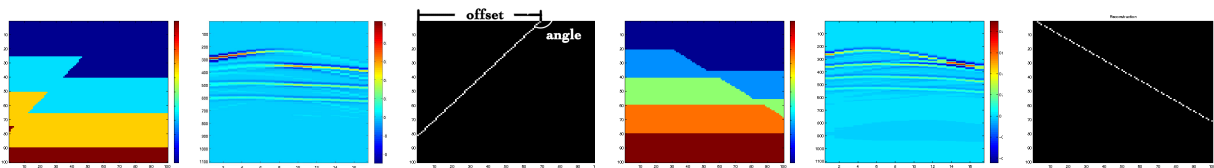
Our framework works in 2-stages. In the first stage, we formulate a binary classification problem to predict presence or absence of faults. Then a more complicated formulation is used to locate the fault (if one is presented). The results on fault or non-fault binary classification are shown in Table 1. As we can see, starting with the basic model, perfect prediction in unseen test data was possible. By adding described features the synthetic velocity models increase geological soundness, making the problem more difficult. The prediction error rates increase to 4.6%. The elapsed time for this stage is about 10 seconds in a generic X86-based computer to process both training and testing of all the 10000 models.

Velocity Model	Train (%)	Test (%)
Basic	0.0	0.0
+ Random Thickness	0.0	0.4
+ Random Velocity	0.0	2.5
+ Swapping Neighboring Layer	0.0	4.5
+ Normal / Reverse Fault	<b>0.1</b>	<b>4.6</b>

Table 1: Fault binary classification error rate. For increasingly complex synthetic velocity model

In the next step, the more complex problem of localization of the faults is tackled. Figure 3 shows ve-

locity model examples and the corresponding prediction of the fault location by our machine learning model. The machine learning algorithm faithfully find the location of the faults and produce a representation of them (white line on black background).



(a) Velocity Model (b) Seismic trace (c) Fault Prediction (d) Velocity Model (e) Seismic trace (f) Fault Prediction

Figure 3: Examples of random velocity models in the test data and the corresponding predicted fault location.

We quantify and summarize the discrepancy between the predicted fault location and the ground-truth in Fig. 4. Specifically, the straight fault in our model is described by two parameters: angle and offset (distance of the fault projection crossing-point with axis  $x$ ). Absolute errors for the two parameters are computed and compared. As can be seen in Fig. 4, most of the probability density is concentrated on very small absolute errors for both parameters, and the mean absolute test errors are around 2. The computation for training and localization is more costly than the binary case, the elapsed time is about 20 minutes for a generic X86-based computer to process all the models.

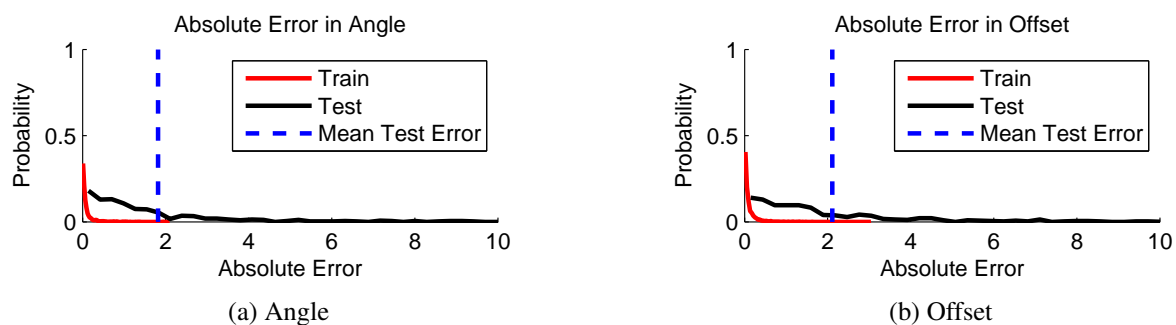


Figure 4: Distribution of prediction errors on the fault location.

## Conclusions

The proposed approach is accurate in both classification (4.6% error for the most complex synthetic model) and localization (mean test error of 2) of faults from not migrated seismic data. We believe that the number of tests make these results meaningful.

Our framework is general, and in principle could be extend to handle detection of any kind of seismic attribute that is well defined and represented. Our long term goal is to add this technique to VMB toolsets. Towards that goal, we will first adapt the approach to the complexities of real both 2D and 3D data.

Our preliminary results on synthetic data could attract more attention from both machine learning and Geo-informatics to further investigate data-driven analysis of not migrated seismic data to enhanced the VMB process.

## Acknowledgements

We thank Fernando Gonzalez Del Cueto, Ludwig Schmidt, Mohamed El-Toukhy, Jonathan Kane and Mark Verschuren for useful discussions.

## References

- Admasu, F. and Toennies, K. [2004] A model-based approach to automatic 3d seismic horizons correlations across faults. *In: Simulation und Visualisierung*, 239–250.
- Bishop, C.M. [2006] *Pattern recognition and machine learning*. Springer, New York, ISBN 0387310738; 9780387310732.
- Cohen, I., Coult, N. and Vassiliou, A.A. [2006] Detection and extraction of fault surfaces in 3d seismic data. *GEOPHYSICS*, **71**(4), 21–27.
- Evgeniou, T., Pontil, M. and Poggio, T. [2000] Regularization networks and support vector machines. *Advances in Computational Mathematics*, **13**(1), 1–50.
- Hale, D. [2013] Methods to compute fault images, extract fault surfaces, and estimate fault throws from 3d seismic images. *GEOPHYSICS*, **78**(2), 33–43.
- Tingdahl, K.M., Steen, O., Meldahl, P. and Ligtenberg, J.H. [2001] Semi-automatic detection of faults in 3-d seismic signals. *SEG Annual Meeting*.