GEOACTA VOL. ESPECIAL 43(2):7-29 (2022) – AAGG2021 ISSN 1852-7744

MACHINE LEARNING FOR SEISMIC DATA ANALYSIS AND PROCESSING

Danilo R. Velis [1], Julián L. Gómez [1,2], Gabriel R. Gelpi [3], Germán I. Brunini [1], Daniel O. Pérez[1], Juan I. Sabbione[1]

[1]Facultad de Ciencias Astronómicas y Geofísicas, Universidad Nacional de La Plata, y CONICET, La Plata, Argentina
[2]YPF Tecnología S.A., Berisso, Argentina
[3]Facultad de Ciencias Astronómicas y Geofísicas, Universidad Nacional de La Plata, La Plata, Argentina
Contact: velis@fcaglp.unlp.edu.ar

RESUMEN

El aprendizaje automático está marcando el ritmo del avance del análisis de datos en muchos campos de la ciencia, la tecnología y la industria. En este contexto, el procesamiento y la inversión de datos sísmicos se abordan mediante estrategias que extraen la información relevante de los datos de forma casi automática. El **"dictionary learning"** y las Redes Neuronales son dos ejemplos comunes de algoritmos capaces de capturar las estructuras y patrones complejos incrustados en los datos e inferir o predecir cierta información de interés a partir de ellos. Utilizamos la técnica de **"residual dictionary denoising"** para atenuar la huella de adquisición en los datos sísmicos 3D. Además, demostramos algunos avances en el uso de una red neuronal profunda para invertir el tensor de momento sísmico en escenarios de monitorización de pozos. El aprendizaje automático también incluye técnicas de optimización global, como el recocido simulado y la evolución diferencial. Exploramos cómo estos dos algoritmos pueden automatizar procesos en la exploración sísmica, como el análisis de la velocidad y el "well-tying" que convencionalmente se hacen a mano y, por lo tanto, son susceptibles de la subjetividad y la experiencia del usuario.

PALABRAS CLAVES: EXPLORACION SISMICA; VELOCIDADES; REDES NEURONALES

ABSTRACT

Machine learning is setting the pace in the advancement of data analysis in many fields of science, technology, and industry. In this context, seismic data processing and inversion are approached by strategies that extract the relevant information from the data almost automatically. Dictionary learning and neural networks are two common examples of algorithms capable of capturing the complex structures and patterns embedded in data and inferring or predicting certain information of interest from them. We use a residual dictionary denoising technique to attenuate the acquisition footprint in 3D seismic data. Besides, we demonstrate some progress in using a deep neural network to invert the seismic moment tensor in well-monitoring scenarios. Machine learning also includes global optimization techniques, such as simulated annealing and differential evolution. We explore how these two algorithms can automate processes in seismic exploration such as velocity analysis and well-tying, which are conventionally done by hand and are thus susceptible to user subjectivity and experience.

KEY WORDS: SEISMIC EXPLORATION; VELOCITIES; NEURAL NETWORKS

INTRODUCTION

Machine learning (ML) are algorithms and strategies devised for making predictions from data without the use of explicit deterministic coding/modeling. It is difficult to pinpoint when ML first appeared, but one of the key works that put on the table the idea of a computer learning to "think" like a human was in the late 1950s (Samuel 1959). Since then, the number of developments and applications has increased dramatically. ML algorithms, such as artificial neural networks (ANN), are useful for extracting information from large datasets associated with complex systems when a deterministic approach/model is unavailable (e.g., predicting which movie a given streaming service subscriber would like to watch (Bennett, Lanning, and others 2007)).

Deep neural networks (DNN), convolutional neural networks (CNN), and supervised or unsupervised NN are ANN examples. DNN is an acronym for NN with multiple layers. In the simplest of settings, each layer of a DNN contains affine transforms that are linked together by nonlinear “activation functions". CNN, on the other hand, use convolutional filters instead of affine transforms. In supervised systems, the concept of “learn-by-examples” is crucial, which requires the examples to be labeled; while in unsupervised systems, the NN learns to cluster the input data into groups with distinct features.

Dictionary learning (DL), evolutionary algorithms such as genetic algorithms (GA) and differential evolution (DE), simulated annealing (SA), particle swarm optimization (PSO), and others are also examples of ML methods (Qadrouh et al. 2019 e.g.). DL (Tošić and Frossard 2011 e.g.), for example, draws a limited number of patches from the input data to create a dictionary, which is then used to fit the original data by learning through sparse representations. On the other hand, GA (Goldberg 1989 e.g.) mimics the process of natural selection, whereas SA (Davis 1987 e.g.) mimics the process of annealing. Contrarily, DE (Storn and Price 1997 e.g.) does not mimic any natural or biological process, but borrows concepts from GA such as crossover and mutation.

In geophysics, ML has found many successful applications (Bougher 2016; Qadrouh et al. 2019), including predicting petrophysical properties (e.g. porosity, permeability) from well-log and/or seismic data, seismic data interpolation, denoising, and reconstruction, seismic inversion for reservoir/mining applications, subsurface facies classification and structure delineation/identification (e.g. faults, salt bodies), signal detection, and horizon/velocity picking. The list of applications is constantly growing.

We describe four novel applications developed recently by the METIS group (“Métodos especializados para el procesamiento de la información sísmica”, Facultad de Ciencias Astronómicas y Geofísicas, UNLP) that use a variety of ML techniques adopted for processing seismic data. The first two are methods for uncovering hidden patterns and information in input data: (1) footprint removal using DL (Gómez and Velis 2020) and (2) moment tensor inversion using NN (Brunini, Velis, and Sabbione 2021). The other two are methods to automate processes that are normally carried out by hand: (1) automated well tying and phase estimation using DE (Gelpi, Pérez, and Velis 2020), and (2) automated velocity analysis using SA (Velis 2021).

The report is divided into two parts, each with two sections, one for each of the four applications mentioned above. Each section begins by presenting the problem, objectives, and methods, as well as a brief description of the machine learning (ML) used. The techniques are then demonstrated using synthetic and/or field data examples, which leads to draw the conclusions summarized at the end of each section. For a complete description of the methods, the reader is directed to the corresponding publications.

METHODS TO UNCOVER HIDDEN PATTERNS AND INFORMATION BY LEARNING FROM DATA

In this section we describe two ML applications to seismic data analysis: (1) footprint removal using DL (Gómez and Velis 2020) and (2) moment tensor inversion using NN (Brunini, Velis, and Sabbione 2021). Essentially, both dictionary learning (DL) and neural networks (NN) learn from the input data by uncovering hidden patterns and information, either to make predictions or to capture its most relevant components. DL assumes that the observed data is a linear combination of a few atoms. These atoms are learned from patches extracted from the input data and contain the most important blocks of information, while ignoring irrelevant details such as random noise. On the other hand, NN employ an artificial net to predict the underlying model that reproduces a set of given observations. For training the network, a large amount of simulated data can be used. The training process updates the network parameters so the resulting algorithm can model the observed data with great accuracy and, ultimately, give reasonable predictions for unseen datasets.

Footprint removal using DL

Seismic data can contain random and coherent noise. A typical example of coherent noise is the acquisition footprint, which can lead to an incorrect assessment of the geological structure of the reservoirs. Therefore, the removal of footprint noise from seismic data is of great interest for seismic interpretation and data analysis (Sahai and Soofi 2006; Alali, Machado, and Marfurt 2018). The acquisition footprint is caused by constraints in acquisition design, acquisition equipment, and the processing workflow applied to the recorded seismic data (Marfurt et al. 1998). There are various standard techniques for footprint attenuation, most of them based on wavenumber filtering (Gülünay, Martin, and Martínez 1994; Gülünay 1999; Chopra and Larsen 2000; Drummond, Budd, and Ryan 2000; Soubaras 2002; Falconer and Marfurt 2008).

In the case of random noise, dictionary learning (DL) methods borrowed from the field of sparse and redundant representation of signals (Mallat 1999; Elad 2010; Tošić and Frossard 2011), provide several interesting applications in seismic processing (Beckouche and Ma 2014; Turquais, Asgedom, and Söllner 2017a; Li, Zhang, and Mosher 2019; Zu et al. 2019). A dictionary is a collection of redundant vectors known as atoms. The DL process is an alternate optimization scheme. Given an initial dictionary, a sparse representation of the data is calculated. The sparse representation is then applied to update the initial dictionary. The updated dictionary improves the sparseness of the data representation. This two-step process iterates until convergence. For coherent noise, DL often requires complex atom classification strategies to differentiate atoms that contain structured noise from those that contain seismic signal (Turquais, Asgedom, and Söllner 2017b). Residual dictionary denoising (RDD) (Gómez and Velis 2020) overcomes this issue. It uses an augmented dictionary of filtered atoms and their residuals after edge-preserving smoothing. The atoms in the residual dictionary provide a good representation of the footprint pattern. This fact is used to enable automated source separation.

Dictionary learning

DL is a machine learning technique for finding a sparse representation of a given dataset using a relatively small set of atoms. These atoms are learned from patches of the input data:

$$\{\hat{D}, \hat{X}\} = \arg\min_{D,X} \| X \|_0 \text{ s.t.} \min_{D} \| Y - DX \|_F^2, \quad (1)$$

where Y represents a collection of 2D patches from the input data in column vector form, D the dictionary, and X the sparse representation of the data patches collected in Y (Tošić and Frossard 2011 e.g.). The operator $\| X \|_0$ counts the number of nonzero entries in X, and $\|\cdot\|_F^2$ is the square of the Frobenius matrix norm. In practice, we solve this problem using coherent DL (CDL) (Turquais, Asgedom, and Söllner 2017a).

The method of residual dictionary denoising (RDD) allows to separate signal from coherent noise by building an augmented dictionary (D_a) that comprises filtered atoms (D_f) and residual atoms (D_r) (Gómez and Velis 2020):

$$D_a = D_f \cup D_r, D_r = D - D_f. \tag{2}$$

Figure 1 shows an atom drawn from the learned dictionary $\hat{D}$ that contains both signal and acquisition footprint. After filtering, the atom will be mostly signal, while the residual atom will be mostly acquisition footprint. We can denoise each patch of the original data y by removing the footprint component given by the residual dictionary: $y_r = D_r x_r$. That is

$$\hat{y} = y - D_r x_r, \tag{3}$$

where $\hat{y}$ is the denoised patch. Assembling and averaging the denoised patches yields the filtered seismic data.

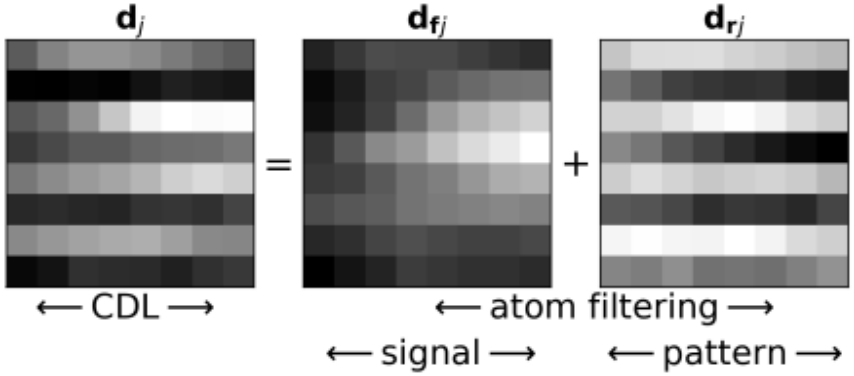

Figure 1. By filtering, a dictionary atom d_j is separated into a signal atom d_{f_j} and a residual atom d_{r_j}. The energy from the coherent noise should be mostly represented by d_{r_j}. Modified after (Gómez and Velis 2020).

Examples

Figure 2 shows a synthetic example that illustrates the filtering of a 3D seismic volume. In this case, the data (left panel) contains both random and coherent noise. Figure 3 depicts the corresponding learned, filtered, and residual dictionaries. The learned dictionary is a mixture of the seismic signal and the footprint noise. The augmented dictionary D_a consists of the filtered dictionary D_f and the residual dictionary D_r. As shown in Figure 2 (middle and right panels), RDD captures the coherence noise, which is then removed from the original data.

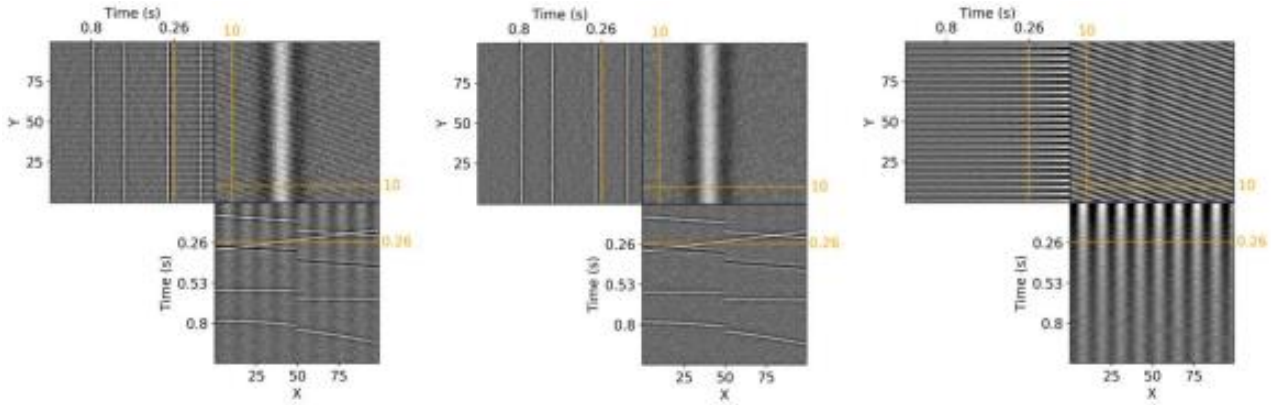

Figure 2. Synthetic example. Left: 3D cube with random and coherent noise. Middle: result after RDD. Right: residual (scaled by a factor of 5 for better visualization). Modified after (Gómez and Velis 2020).

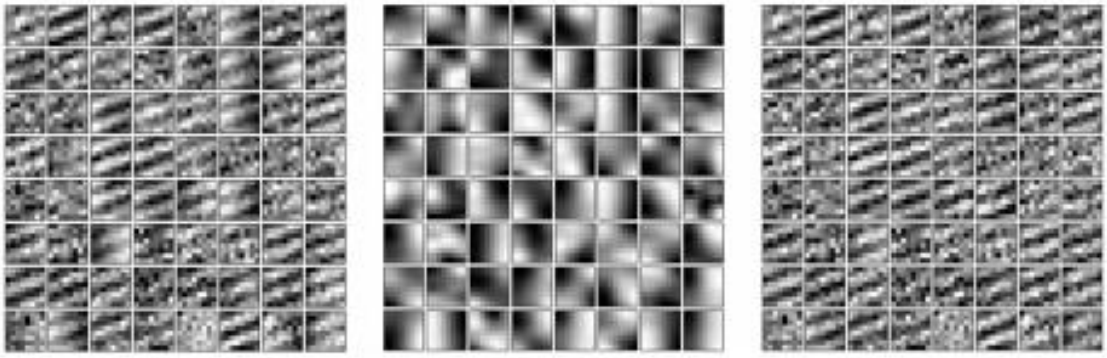

Figure 3. Synthetic example. Learned, filtered, and residual dictionaries corresponding to data shown in Figure 2 (left panel). Modified after (Gómez and Velis 2020).

We apply RDD to offshore seismic data from Nova Scotia, Canada. We compare the results of RDD, conventional DL, and wavenumber filtering using a single time slice in Figure 4. The resulting RDD image is footprint-free, sharp, and with negligible signal leakage. Figure 5 shows the corresponding dictionaries.

We note that to achieve a similar result with wavenumber filtering, a seismic processor should choose a filter and tune its parameters (e.g., the cutoff frequencies) by inspecting the given data and the filtered result in an iterative fashion. Figure 6 shows that RDD does an acceptable work in the whole seismic data: the filtered volume is free of footprint, while the residual volume clearly contains the undesired coherent noise. RDD is applied time slice by time slice in this example.

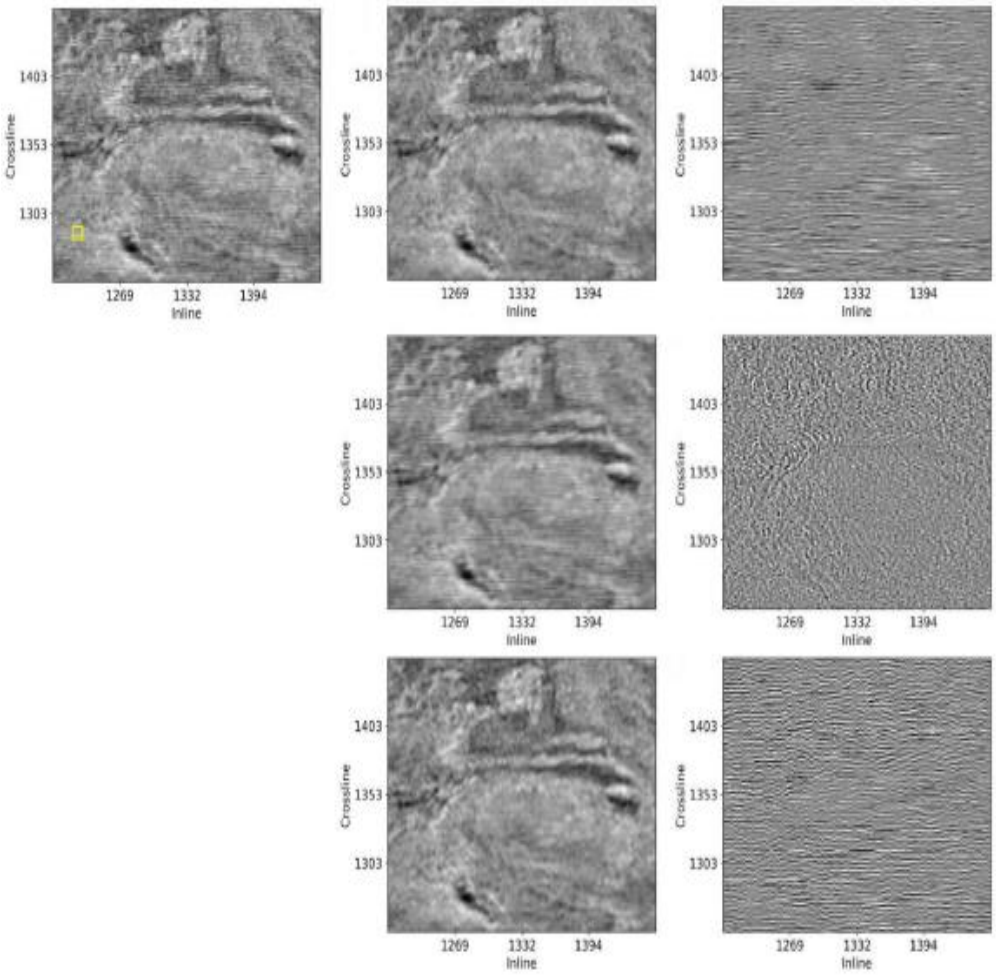

Figure 4. Field data example. Top: Time slice at $t = 0.116$ s (left), filtered by RDD (middle) and residual (right). Second row: Filtered by conventional DL and residual. Bottom: filtered by wavenumber filtering and residual. The yellow sector shows the size of a patch used for learning the dictionary. The residuals are scaled 5 times for visualization. Modified after (Gómez and Velis 2020).

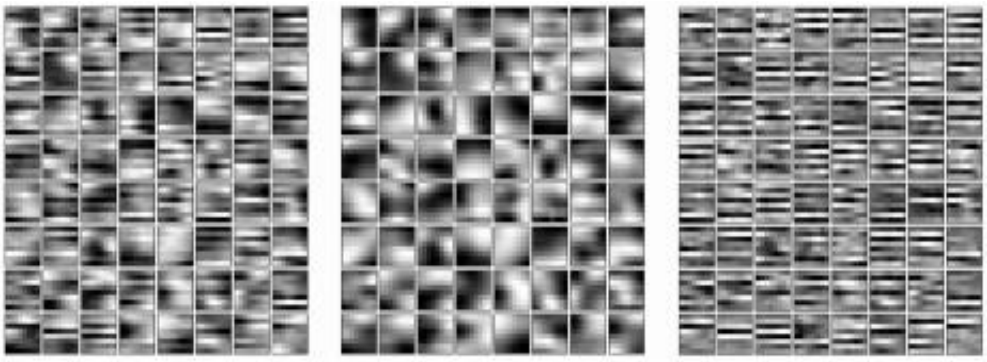

Figure 5. Field data example: learned, filtered, and residual dictionaries corresponding to data shown in Figure 4 (top left panel). Modified after (Gómez and Velis 2020).

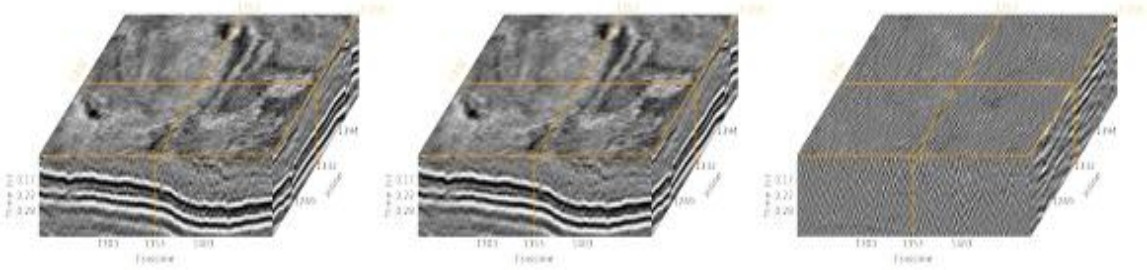

Figure 6. Field data example. Original subvolume, filtered by RDD, and residual. The residual is scaled 5 times for visualization.

One key feature of RDD is its application in transfer learning mode, where the dictionary learned from one time slice can be used to filter different time slices. This saves computational effort, since the most expensive step in RDD is learning the dictionary $\hat{D}$ from the alternate optimization scheme. Figure 7 shows an example of filtering a different time slice of the offshore dataset using the dictionary shown in Figure 5. Finally, Figure 8 shows the complete seismic data filtered using RDD with transfer learning.

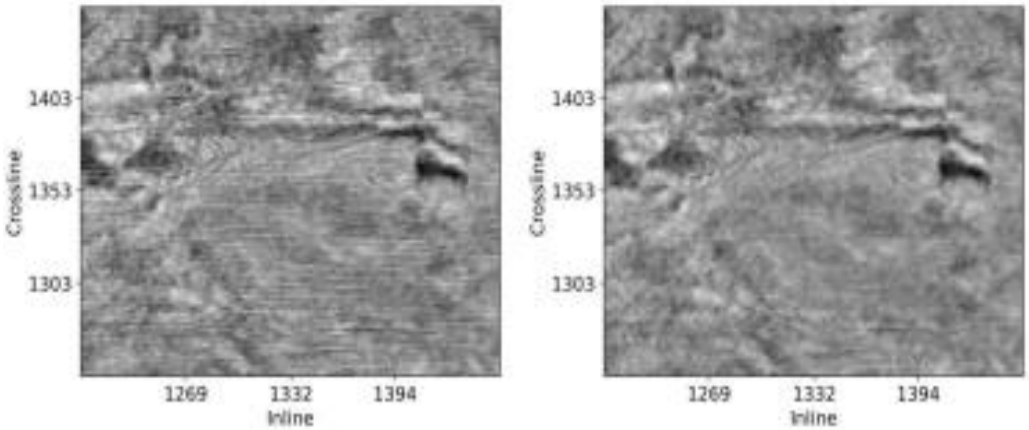

Figure 7. Field data example. Left: Time slice at $t = 0.136$ s. Right: Filtered time slice with transfer learning, where the augmented dictionary was derived from the time slice at $t = 0.116$ s (Figure 5). Modified after (Gómez and Velis 2020).

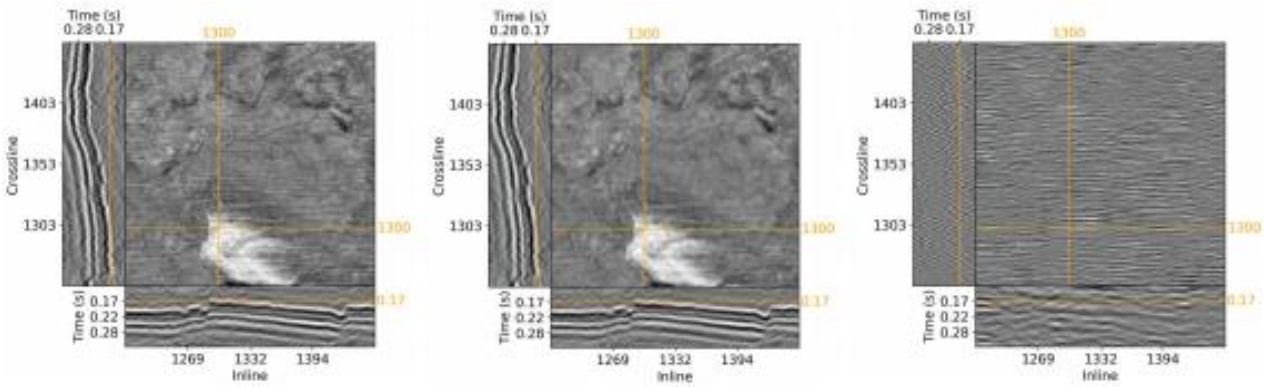

Figure 8. Field data example. Left: Whole subset. Middle: Subset filtered by RDD with transfer learning. Right: Residual (scaled by 5). The dictionary learned at $t = 0.116$ s was filtered and enhanced to be applied to the whole subset. The inlines and crosslines show that signal is preserved in the whole dataset. Modified after (Gómez and Velis 2020).

Conclusions

The proposed RDD method is a dictionary learning variant that can remove coherent noise from time slices of 3D seismic data without analyzing the data's morphology to classify the atoms by hand. The augmented dictionary joins the filtered and residual dictionaries, allowing the exclusion of the footprint patterns from the data's final sparse representation automatically. The denoising process is further automated and simplified by RDD's transfer learning capabilities.

Moment tensor inversion using NN

Solving the MTI problem is important for monitoring microseismic hydraulic fracturing processes (Baig and Urbancic 2010). The focal mechanism is typically the most valuable information extracted from a microseismic monitor survey, as it provides insights into fracture orientation and slip directions, ultimately aiding in the understanding of a reservoir's geomechanical behavior (Grechka and Heigl 2017). Figure 9 (left) depicts a typical hydraulic fracturing scenario where two vertical wells containing an array with 3C geophones record seismic data generated by several events in the nearby. Figure 9 (right) shows a sample 3C microseismic dataset as acquired by one of the well arrays. The data contains the P-wave arrival and two S-wave phases because we consider a subsurface with vertical transverse isotropy. In MTI, the fracture mechanism should be obtained from the amplitude information contained in these observed data. Several limiting factors, such as the poor angle apertures that characterize most acquisition geometries and the data's low signal-to-noise ratio, may prevent this task from being accomplished successfully and without ambiguity.

MTI methods are typically deterministic (Vavryčuk and Kühn 2012; V. I. Grechka 2015). However, only a few works have been dedicated to solving the MTI problem using ML/AI (Ovcharenko, Akram, and Peter 2018; Binder 2018; Wamriew et al. 2020). Deterministic methods necessitate a significant computational effort and do not allow for the easy solution of new MTI as new data is acquired, because the MTI solver is dependent on each specific data. NNs, on the other hand, allow for the generation of solutions for any new dataset on the fly, as long as they are well-trained beforehand. Because of the well-known theoretical limitations imposed by the single-well MTI scenario (V. Grechka 2015 e.g.), the deep neural network that we propose (Brunini, Velis, and Sabbione 2021) solves the MTI problem in a dual-vertical-borehole case.

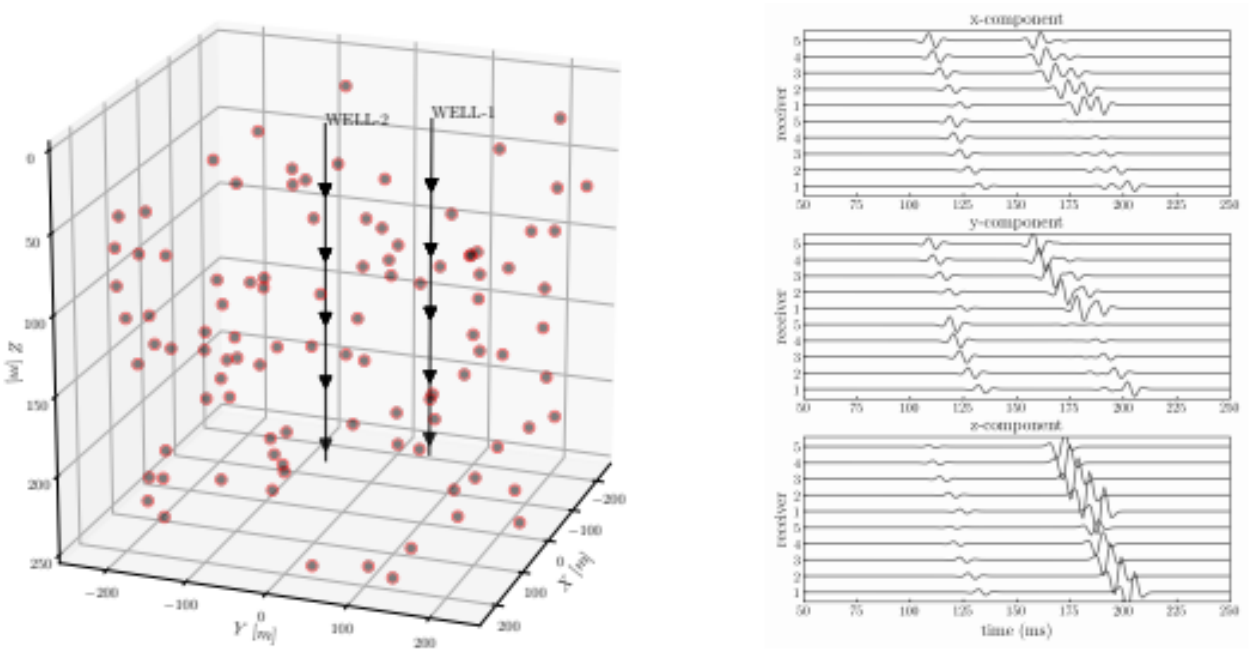

Figure 9. Left: subsurface (VTI: vertical transverse isotropy) and acquisition geometry. A large number of fractures with random focal mechanism and coordinates. Right: sample 3C microseismic signal showing P- and both slow and fast S-wave phases. Modified after (Brunini, Velis, and Sabbione 2021).

Deep neural network

A typical deep neural network (DNN) consists of several neuron layers interconnected by nonlinear activation functions. The base computation of a DNN involves the following relation for one neuron of one layer:

$$y = g(w \cdot x + b), \tag{4}$$

where x is the input as a column vector, w the weights and b the bias of the neuron, respectively, g a nonlinear activation function, y the output node, and the operator $\cdot$ the inner product (Goodfellow, Bengio, and Courville 2016). Collecting all the outputs for different neurons we get the column vector y, which is the output of one layer. The output of one layer is connected to the next layer of neurons until reaching the output layer with its desired output size or number of nodes. The weights and biases of all the layers are the model parameters (unknowns) estimated by training. The input layer is fed by the input data x (which are known as features). The output layer produces the prediction of the model.

The DNN used to solve the MTI problem contains 6 layers, where the input layer has 93 nodes, each one associated with a *feature* (10 receivers x 3 components x 3 phases + 3 coordinates), and the output layer has 6 nodes, each one associated with a *prediction* (6-independent MT elements). For the prediction, we rely on the mean squared error (MSE) loss function. For training we use 9×10^5 simulated events with known source mechanisms and coordinates, while for testing we used 9×10^4 simulated events.

Examples

Figure 10 shows the results of the MTI inversion using the proposed DNN. As shown, most predicted MT parameters are very similar to the actual values, both in the training and in the testing stages. Figure 11 shows the relative errors for the 9×10^5 predicted events during the training stage. We observe a very high accuracy in most cases, except for those events that are placed

along the vertical plane that contains both wells. In these cases, as expected, the information contained in the observed data is not enough to resolve the inverse problem unambiguously.

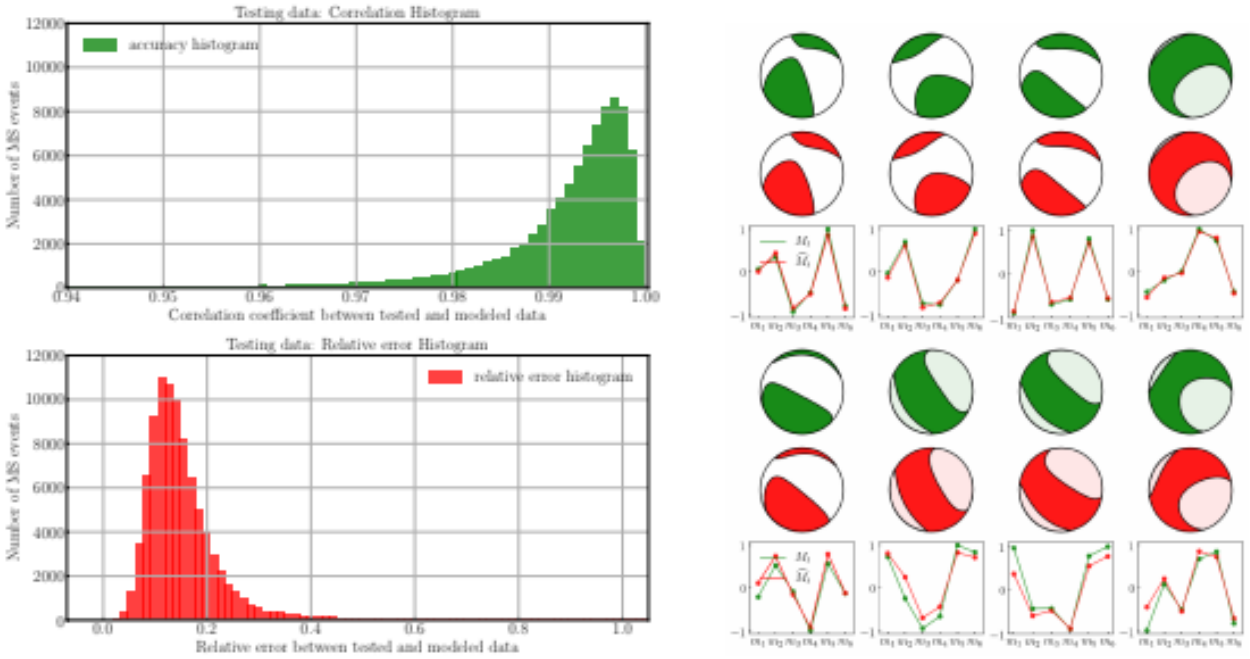

Figure 10. Left: Correlation coefficient between actual and predicted MT elements for the testing dataset. Right: Relative error between actual (green) and predicted (red) MT elements. Top: $0.99 \leq \rho \leq 1.0$ (>70%) bottom: $0.94 \leq \rho < 0.95$ (<1%). Modified after (Brunini, Velis, and Sabbione 2021).

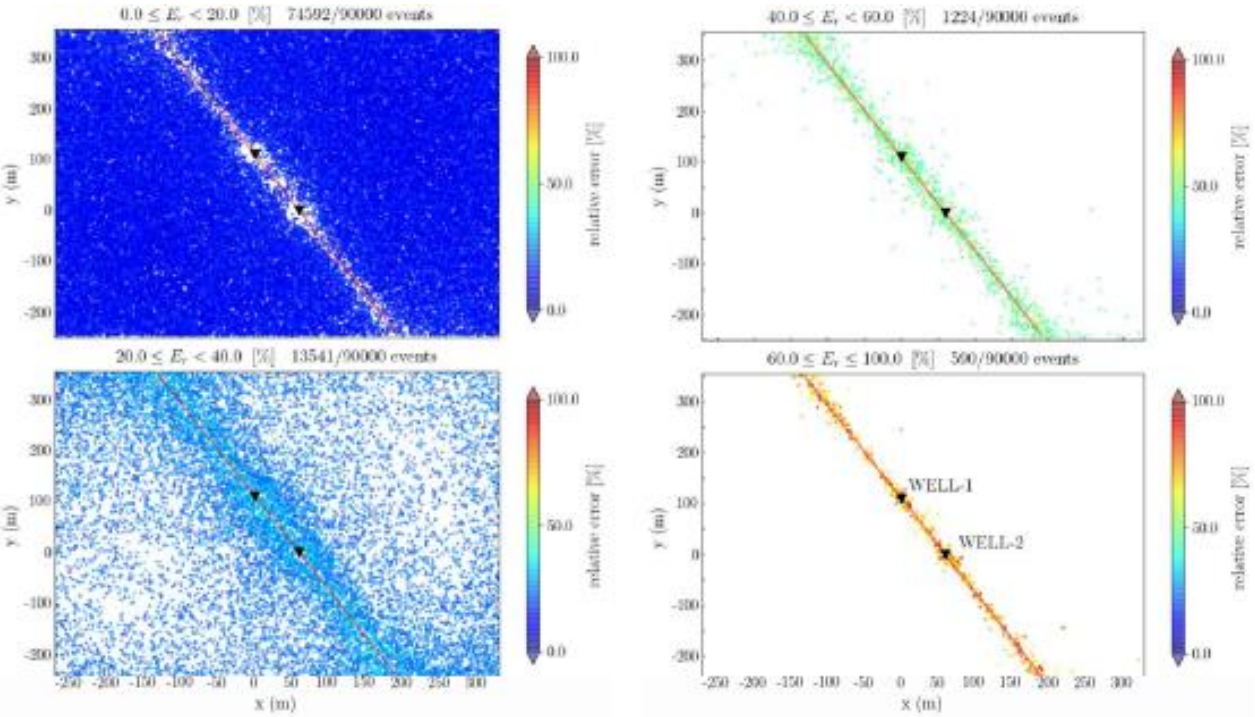

Figure 11. Relative errors of the predicted MT elements (different ranges). Largest errors are aligned along the vertical plane that contains the wells. Modified after (Brunini, Velis, and Sabbione 2021).

Conclusions

The proposed DNN architecture is capable of accurately predicting most testing data. The inversion is poor for events near the plane containing both wells, but becomes very accurate as they move away from this plane. The results show that a well-trained DNN is a good alternative to

traditional MTI methods because it can be used on the fly whenever new seismic data for the same monitoring scenario is acquired.

METHODS TO AUTOMATE PROCESSES THAT ARE NORMALLY CARRIED OUT BY HAND

We describe in this section two ML applications to automate seismic data processes that are usually carried out by hand: (1) automated well tying and phase estimation using DE (Gelpi, Pérez, and Velis 2020), and (2) automated velocity analysis using SA (Velis 2021). Both differential evolution (DE) and simulated annealing (SA) are stochastic optimization algorithms whose goal can be stated as: given a cost or objective function (any "function" or process that produces an output from a given input), find the global minimum regardless of the initial guess. Of the two methods considered, simulated annealing (SA) (Laarhoven and Aarts 1988 e.g.) mimics the process of annealing (i.e. slow cooling of metals until they reach the lowest energy state). Differential evolution (DE) (Storn and Price 1997), on the other hand, does not mimic any natural process, but borrows some concepts from genetic algorithms (GA), that do mimic a natural process (Darwin's evolution).

Both DE and SA rely on some randomization to carry out the search, which is done in a clever way so that the model space (space of feasible solutions) can be explored without evaluating all feasible solutions. Unlike most non-stochastic optimization algorithms (e.g., steepest descent), they do not require derivatives, avoid local minima, perform well for ill-behaved functions, and allow for the incorporation of constraints easily. These key characteristics enable these algorithms to solve difficult optimization problems, such as those that arise in complex scenarios. For instance, for finding a set of parameters that control a number of processes applied to input data to achieve a desired goal, regardless of the complexity of the underlying processes. It is important to note, however, that the convergence of most stochastic optimization algorithms is not guaranteed and may be too slow, especially when the number of unknowns (parameters) is relatively large ("curse of dimensionality"). In the applications that follow, we show how relatively complex manual seismic processes can be emulated using DE and SA to improve efficiency and reduce user subjectivity.

Automated well tying and phase estimation using DE

The goal of well-tying is to match the observed seismic data to the seismic trace obtained from well-log data (synthetic trace). This enables the linking of well-log information, such as geological units and formations, to seismic data, an important aspect of seismic data interpretation (Ziolkowski, Underhill, and Johnston 1998; Herron 2011; Simm and Bacon 2014 e.g.). This process is traditionally done by hand (trial-and-error) by adding bulk time shifts and phase rotations. Furthermore, some trace stretching and/or squeezing may be required at some points along the time scale to improve the match. This process, as expected, is inefficient and prone to inconsistencies because it is subject to the user's subjectivity and the arbitrary selection of the parameters (Walden and White 1984; White and Simm 2003; Newrick 2012).

These issues highlight the need for automated or semi-automated methods to mimic the human process. One of these methods is dynamic time warping (DTW), which allows to attain high correlation values between the observed and synthetic traces (Herrera, Fomel, and Baan 2014; Muñoz and Hale 2015), but frequently comes at the expense of introducing some waveform distortions. In contrast, the proposed method iteratively modifies the log data (and the wavelet phase) to significantly increase the match without any waveform distortions (Gelpi, Pérez, and Velis 2020). The velocity log is perturbed (modified) by applying a function created by a monotonic cubic splines interpolator, $p(z)$, with a fixed number of knots M:

$$v_{mod}(z_i) = v_{obs}(z_i)[1 + p(z_i)], \tag{5}$$

with $i = 1, M$. The knots coordinates are regarded as the unknowns in a global optimization problem aimed at maximizing the aforementioned correlation. We solve this problem using DE. Further, the wavelet phase is included as an additional unknown to account for wavelet uncertainties.

Differential evolution

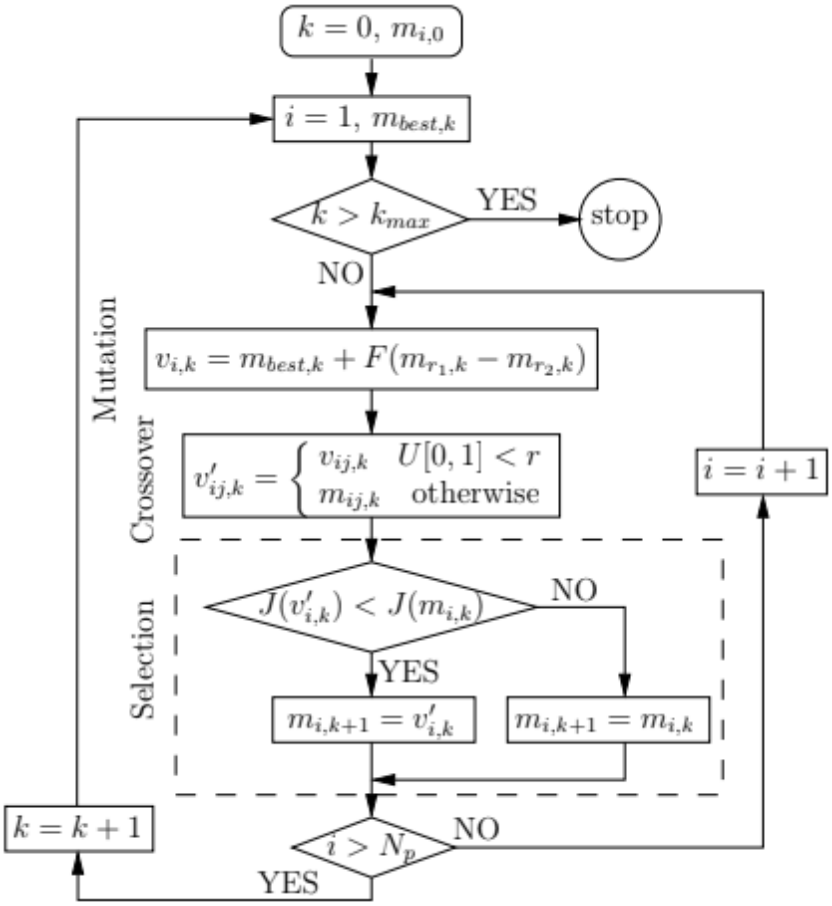

Figure 12. Flowchart of DE (used to automate the well-tying problem). m stands for model parameters (unknowns), J for cost function, and k for generation.

Figure 12 depicts a typical DE algorithm. The first step is to generate an initial population of N_p vectors $m_{i,0}$, where $i = 1, \cdots, N_P$. Then, DE involves the following steps: mutation-recombination, crossover, and selection. Mutation refers to the process of expanding the search space, whereas recombination refers to the process of reusing previously successful individuals. Here, F is a preselected factor used to scale the difference between two randomly selected vector of the same generation k. Crossover is a type of discrete recombination in which trial vectors $v'_{i,k}$ are created by randomly combining the components j of mutant and ordinary vectors (r is the crossover probability, set beforehand). Finally, selection mimics the survival of the fittest observed in the majority of natural phenomena. The number Np is normally kept constant throughout the process, but there are no theoretical constraints on this. There are numerous DE variants, with the main difference being how the mutation stage is implemented.

Examples

First, we demonstrate the method with pseudo-synthetic data, in which we simulated the observed seismic trace using actual field well-log data and an *ad hoc* seismic wavelet. Figure 13, from left to right, shows the observed (black) and modified (red) velocity logs, the monotonic cubic splines with M knots $p(z)$ function used to modify the observed velocity log, the zero-phase (black)

and -30 ° (red) Ricker wavelets used to create the simulated data set, and the "observed" (red) and initial (black) traces. We constrain the knots to lie within a small range P around the actual values to avoid excessive well-log distortions.

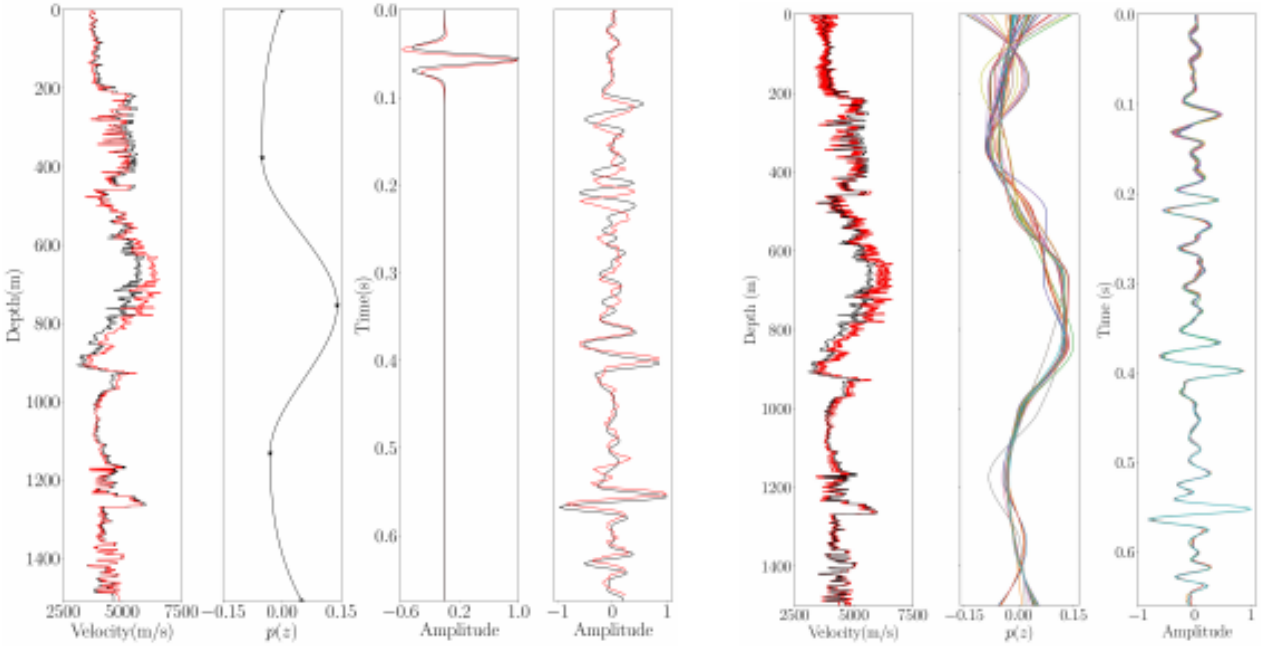

Figure 13. Left: observed (black) and modified (red) velocity logs, $p(z)$, the zero-phase (black) and -30 degrees (red) Ricker wavelets, and "observed" (red) and initial (black) traces. Right: Observed and modified velocity logs, $p(z)$ curves, and traces after 100 realizations. In all cases, $M = 10$, $P = 15\%$. Modified after (Gelpi, Pérez, and Velis 2020).

Since DE is a stochastic optimization algorithm, different solutions might be obtained for different runs. Figure 13 (right) shows the results for $M = 10$ and $P = 15\%$ after 100 realizations. Despite the fact that the individual solutions (splines) are not all that similar, the high consistency between the derived traces indicates that the DE convergence is high.

Figure 14 (left) shows the results for varying M, while keeping the well-log perturbations within $\pm 15\%$. Figure 14 (right) shows the results for varying perturbation range, while keeping the number of knots to $M = 10$. We observe a very high match in all cases. The most conservative solution would be the recommended. That is, the one that allows minimum well-log perturbations using a small number of knots, provided that the correlation is sufficiently high.

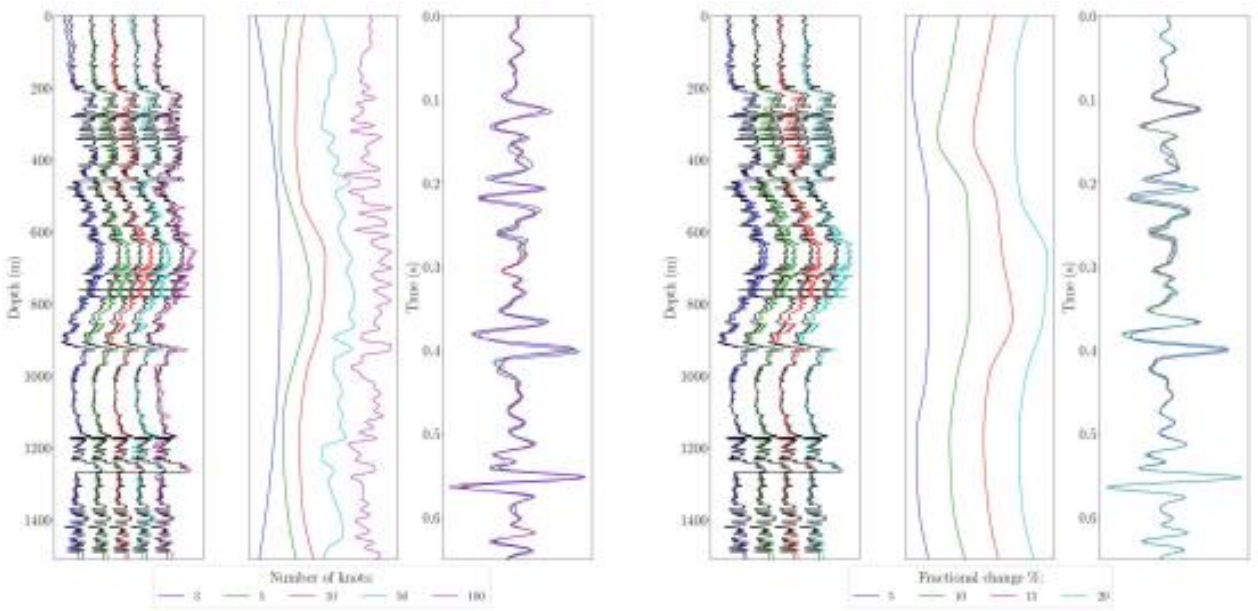

Figure 14. Left: fixed fractional change ($P = 15\%$). Right: fixed number of knots ($M = 10$). Modified after (Gelpi, Pérez, and Velis 2020).

Finally, we show two examples of field data using both field well-log and seismic data. Figure 15 (left) depicts the well-tying results for a field dataset with a moderate initial match between the observed and synthetic traces using manual tying, DE ($M = 15$, $P = 15\%$), and DTW. When we compare DE and DTW to manual tying, we see that these methods lead to a better match. In DE, there are no waveform distortions.

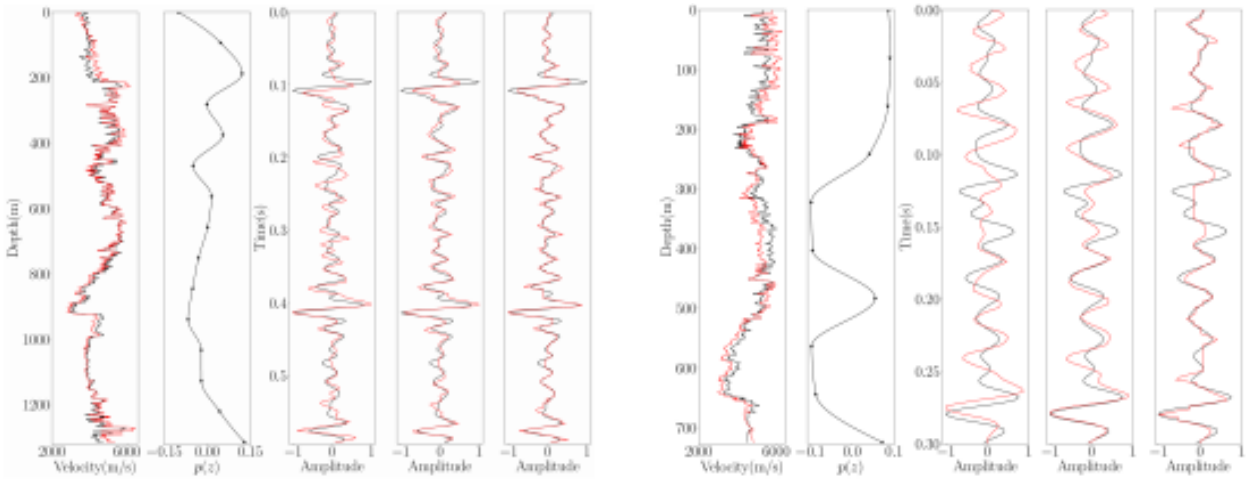

Figure 15. Field data examples. Observed (black) and modified (red) velocity logs, $p(z)$, observed (black) and synthetic (red) traces after manual tying, DE, and DTW. Left: field data example 1 ($M = 15$, $P = 15\%$). Right: field data example 1 ($M = 10$, $P = 10\%$). Modified after (Gelpi, Pérez, and Velis 2020).

A similar pattern can be seen in the second field data example (Figure 15). The original match between the observed and synthetic traces was very low in this case. Furthermore, because the time window is so short, we use a more conservative parameter selection in DE: $M = 10$, $P = 10\%$. The distortions caused by DTW are more visible. As in the previous example, DE and DTW achieve a higher correlation than manual well-tying.

For the sake of completeness, Figure 16 depicts the initial and final wavelets for both field data examples. The zero-phase initial wavelet was estimated from the input traces using a conventional statistical approach.

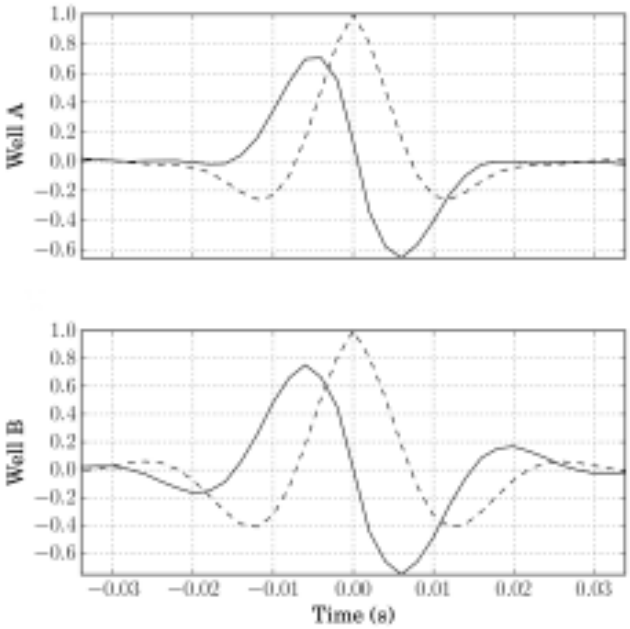

Figure 16. Zero-phase wavelet (dashed) and estimated phase rotated (solid) wavelet for the field data example 1 (top) and 2 (bottom). Modified after (Gelpi, Pérez, and Velis 2020).

Conclusions

When monotonic cubic splines are used to perturb the observed velocity log, controlled velocity changes occur, ensuring that the measured borehole observations are respected within the desired tolerance. The method is stable and consistent, especially when the parameters (number of knots and maximum allowable velocity change) are chosen conservatively. Unlike the manual or DTW approaches, the trace is not stretched or squeezed at any point during the tying process, allowing the wavelet shape to be preserved.

Automated velocity analysis using SA

In seismic processing, velocity analysis entails calculating the velocities required to flatten the reflections of a CDP-gather prior to stacking (Yilmaz 2001 e.g.). The most common method is to create a velocity spectrum (e.g., semblance) and then manually select the maxima associated with primary reflections. To do so, the analyst visually inspects all potential local maxima (along with their associated hyperbolic patterns) relevant to the velocity analysis at hand. The brain performs this process almost effortlessly and instantly, assessing the evidence of a certain relationship between the local maxima and the actual primary reflections using all a priori information (rules) previously built based on past experiences and learning. Prior knowledge includes the facts that normal-moveout velocity $V_{nmo}(t)$ must be within a certain range and increases with increasing two-way time t. Furthermore, only a few events (reflections) will be chosen, they will not be too close together, and maxima associated with multiples or that are negligible should be ignored.

Subjectivity can have a negative impact on the velocity analysis results, in the same way that it is important in the well-tying process. Furthermore, the picking process can be very time-consuming, taking days to process an entire dataset. There are a number of approaches to automating the velocity analysis problem and mitigating these problems (Abbad, Ursin, and Rappin 2009; Fortini et al. 2013; Garabito 2018; Abassi and Gholami 2018; Chen 2018; Sripanich et al. 2020; Park and Sacchi 2020). Most of these methods still necessitate a significant amount of user intervention, the tuning of many parameters, or are only applicable in certain scenarios (for example they assume that the primary main events are previously identified, or that the time trajectories are hyperbolic).

We devised a strategy to overcome these constraints (Velis 2021). In an attempt to mimic the analyst picking process, we propose an automatic technique that incorporates the aforementioned rules via constraints into a nonlinear optimization problem. The method is applicable to both hyperbolic and non-hyperbolic trajectories. In the first case, we minimize the following cost function using very fast simulated annealing (Ingber 1989):

$$J = 1 - \frac{1}{M}\sum_{i=1}^{M} \sigma_i \left(t_0^i, V_{nmo}^i\right), \tag{6}$$

where σ_i is the semblance (or any other normalized coherence measure) computed along the i-th hyperbolic event (we consider M events) defined by the corresponding $\left(t_0^i, V_{nmo}^i\right)$ pair. Because all of the aforementioned rules must be met, appropriate dynamic and static constraints are incorporated into the optimization problem, as described by (Velis 2021). Note that the method boils down to finding a piecewise linear curve with M nodes ($2 \times M$ unknowns), such that J is minimum and all the constraints are satisfied.

Assuming hyperbolic events leads to incomplete flattening in the nonhyperbolic case. As a result, we must consider a third parameter, η_i, which is usually associated with anisotropy or large offsets. As a result, now the optimization problem entails locating a piecewise line in a 3D space, and the number of unknowns increases to $3 \times M$. For convenience, we first assume hyperbolic events and find the picks using only near offsets. These picks are then used as an initial guess for the second step, which involves all three parameters as well as the entire CDP at the same time.

Very fast simulated annealing

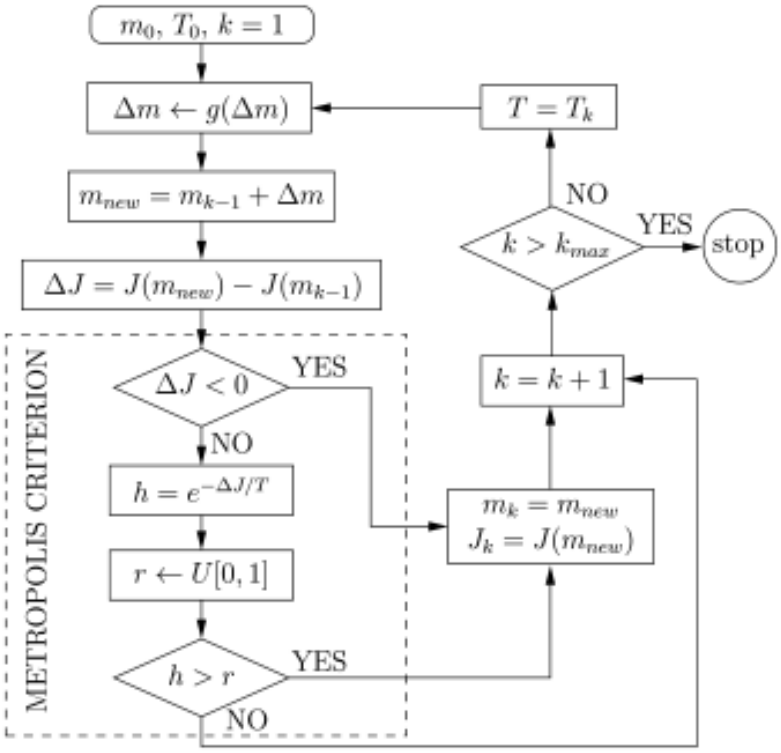

Figure 17. Flowchart of VFSA (used to automate the velocity analysis problem). m stands for model parameters (unknowns), J for cost function, and k for iteration. Modified after (Velis 2021).

Figure 17 depicts the utilized VFSA algorithm (Ingber 1989), which is a variation of the SA optimization algorithm proposed by (Kirkpatrick, Gellat, and Vecchi 1983). To generate model perturbations Δm at each iteration k, VFSA uses a long-tailed Cauchy-like distribution $g(\Delta m)$. This enables for the use of a fast cooling schedule T_k to accelerate convergence while still allowing

for a broad search of the model space at low temperatures, avoiding local minima. $T = T_0 \exp\left(ck^{Q/N_p}\right)$ is used in practice, where T_0 is the initial temperature, N_p is the number of unknowns, Q is a quenching factor ranging from 1 to N_p, and c is a user-defined parameter. VFSA's statistical convergence is guaranteed if $Q = 1$, regardless of T_0 and c. The so-called Metropolis criterion involves assigning a probability $h > 0$ to uphill moves, which is essential for avoiding local minima.

Examples

The velocity analysis problem is depicted in Figure 18. A few hyperbolic events, as well as some multiples, are included in the synthetic CDP-gather. We choose six inner nodes by hand, resulting in a velocity law that flattens the primary reflections (center panel). After solving the described constrained nonlinear optimization problem, SA yields the same result.

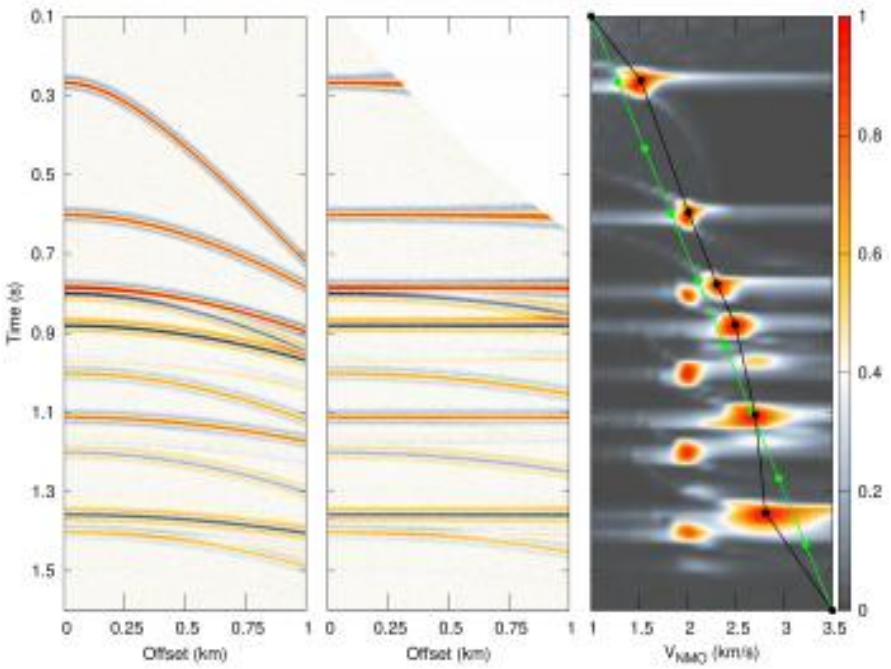

Figure 18. Synthetic gather with hyperbolic events and multiples, NMO-corrected gather, and velocity spectrum (initial picks in green manual/SA in black). Modified after (Velis 2021).

Figure 19 shows an example where the hyperbolic assumption is violated (second panel). In this case, the proposal involves two steps. First, we find the picks $\left(t_0^i, V_{nmo}^i\right)$ by assuming hyperbolic trajectories and employing only near offsets (third and fifth panels). Then, using the previous picks as an initial guess and all offsets, we obtain the final picks $\left(t_0^i, V_{nmo}^i, \eta_i\right)$ to flatten the nonyperbolic events (fourth and sixth panels).

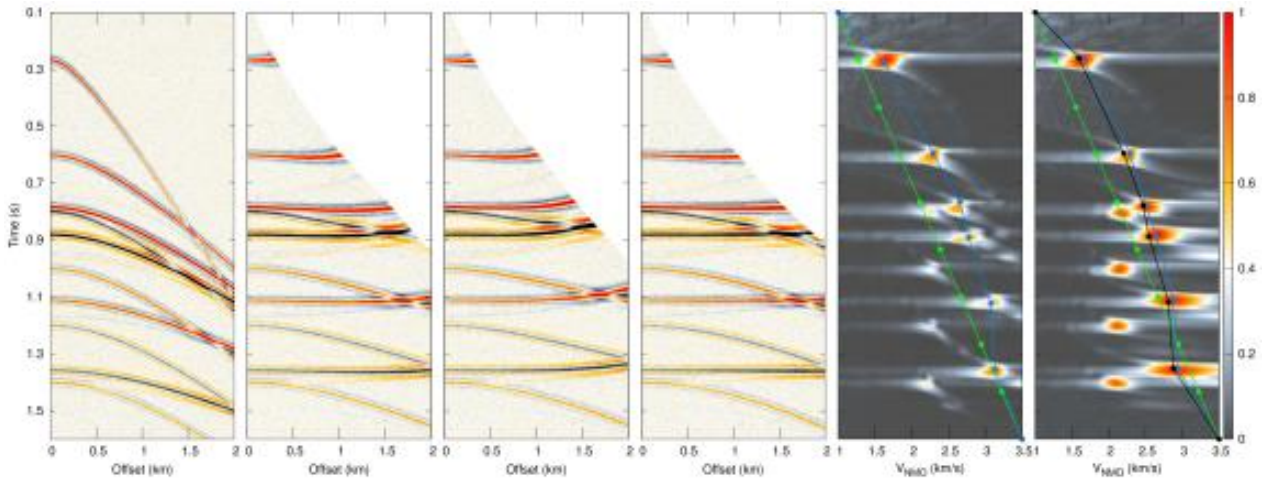

Figure 19. Synthetic gather with nonhyperbolic events and multiples, hyperbolic NMO-corrected gather using all offsets, hyperbolic NMO-corrected gather using near offsets, nonhyperbolic NMO-corrected gather, velocity spectrum (all offsets), velocity spectrum (near offsets): initial (green), SA hyperbolic (blue), SA nonhyperbolic (black). Modified after (Velis 2021).

The next example considers the well-known Alaska dataset (line 31-81). The line contains 535 CDPs (5280 traces) with folds in the range 9 to 12, a maximum offset distance of 1600 m, and a sampling interval of 2 ms. One of such CDPs is shown in the Figure 20, together with the NMO-corrected gather after the SA velocity analysis. Figure 21 shows the corresponding stack after manual picking and SA. In this case, we considered 9 out of 535 CDPs, only. The remaining velocity laws were obtained, as usual, through interpolation in the offset dimension. The match between the two sections is very high.

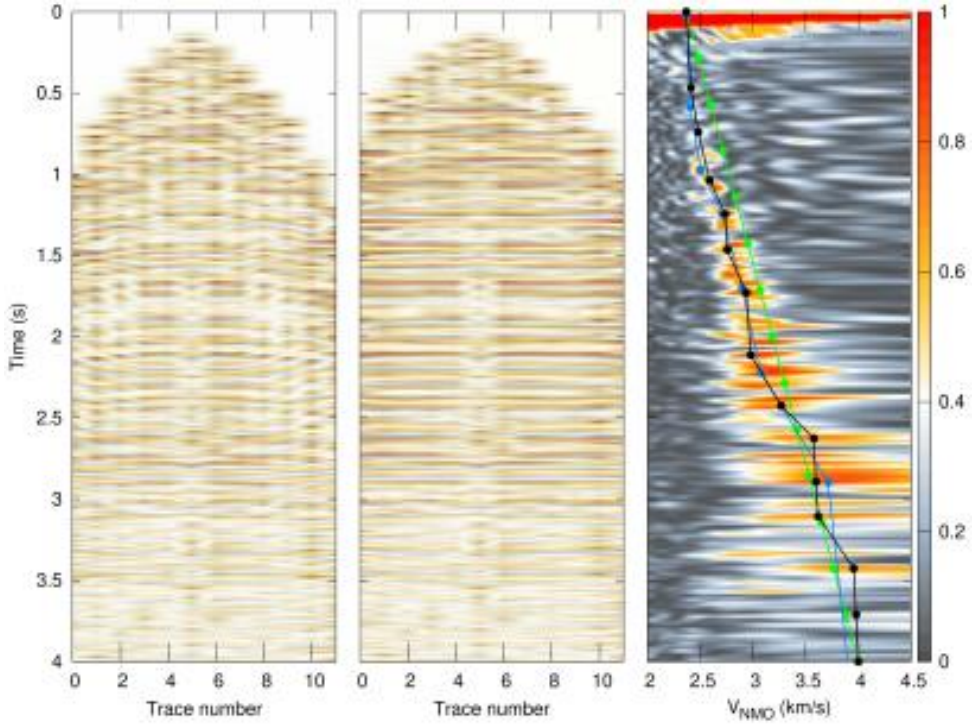

Figure 20. Alaska example. CDP #229 (line 31-81), NMO-corrected gather using SA picks, velocity spectrum: manual (blue), initial (green), SA (black). Modified after (Velis 2021).

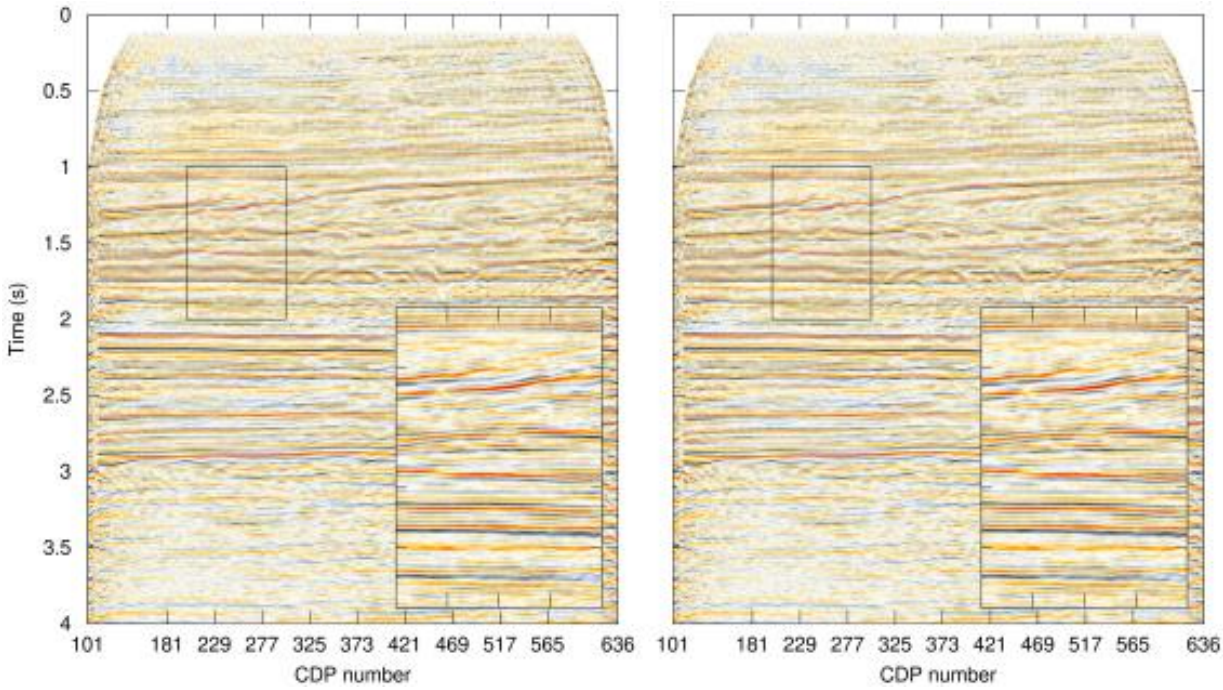

Figure 21. Alaska example. Stack using manual (left) and SA (right) picks. Modified after (Velis 2021).

The next example considers a CDP (SEG's 2007 anisotropic benchmark dataset) that contains nonhyperbolic events and large offsets. The results are shown in Figure 22. The procedure follows the same strategy as in the synthetic example in Figure 19: we start by selecting events based on the hyperbolic assumption and only using near offsets. These preliminary picks are then used as an initial guess for the second stage, which takes into account nonhyperbolic trajectories an all offsets to achieve a better flattening of primary events. Figure 23 depicts zoomed-in areas to help visualize the results.

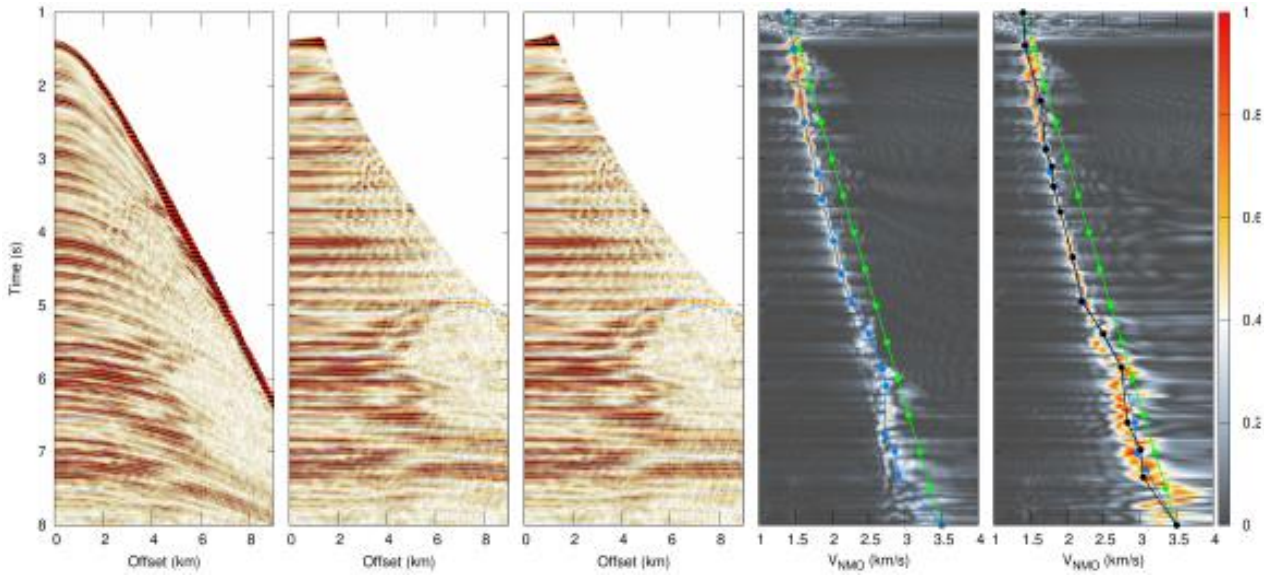

Figure 22. *Anisotropic benchmark example. CDP #8010, hyperbolic NMO-corrected gather, nonhyperbolic NMO-corrected gather, velocity spectrum (all offsets), velocity spectrum (near offsets): initial (green), SA hyperbolic (blue), SA nonhyperbolic (black). Modified after (Velis 2021).*

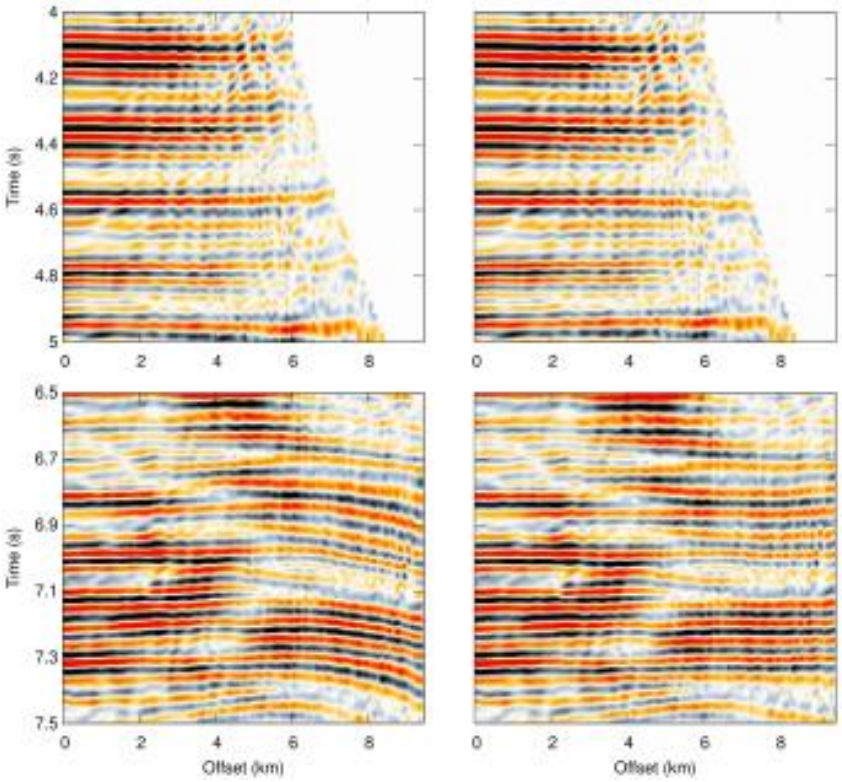

Figure 23. Anisotropic benchmark example. Details of the NMO-corrected gather: hyperbolic (top), nonhyperbolic (bottom). Modified after (Velis 2021).

Conclusions

The SA velocity analysis is carried out by searching for semblance maxima along piecewise linear curves with an unknown number of nodes representing the picks. With VFSA and appropriate constraints, the search can be completed quickly, allowing for the incorporation of a priori information to avoid meaningless picks and multiple reflections, and to guide the search towards a solution that resembles manual picking to some extent.

FINAL REMARKS

Seismic data analysis involves several complex processes and problems that can be solved using different techniques. The ever-increasing size of seismic data sets, combined with the industry's demands for shorter turnarounds, makes machine learning techniques very appealing. As a result, many traditional methods are being superseded by machine learning approaches that frequently yield results with fewer model assumptions and are less reliant on user supervision. Our ultimate goal is to make seismic data interpretation easier, faster, and more accurate. The algorithms we reviewed follow these guidelines.

REFERENCES

Abassi, Mostafa, and Ali Gholami. 2018. "Automatic Nonhyperbolic Velocity Analyis by Polynomial Chaos Expansion." *Geophysics* 86 (6): no. 6, U79–U88.

Abbad, Brahim, Bjørn Ursin, and Didier Rappin. 2009. "Automatic Nonhyperbolic Velocity Analysis." *Geophysics* 74 (2): no. 2, U1–U12.

Alali, Abdulmohsen, Gabriel Machado, and Kurt J. Marfurt. 2018. "Attribute-Assisted Footprint Suppression Using a 2D Continuous Wavelet Transform." *Interpretation* 6 (2): T457–T470. https://doi.org/10.1190/INT-2017-0175.1.

Baig, Adam, and Ted Urbancic. 2010. “Microseismic moment tensors: A path to understanding frac growth.” *The Leading Edge* 29 (3): 320–24.

Beckouche, Simon, and Jianwei Ma. 2014. “Simultaneous Dictionary Learning and Denoising for Seismic Data.” *Geophysics* 79 (3): no. 3, A27–A31. https://doi.org/10.1190/geo2013-0382.1.

Bennett, James, Stan Lanning, and others. 2007. “The Netflix Prize.” In *Proceedings of Kdd Cup and Workshop*, 2007:35. New York, NY, USA.

Binder, Gary. 2018. “Neural Networks for Moment-Tensor Inversion of Surface Microseismic Data.” In *SEG Technical Program Expanded Abstracts 2018*, 2917–21. Society of Exploration Geophysicists.

Bougher, Benjamin Bryan. 2016. “Machine Learning Applications to Geophysical Data Analysis.” PhD thesis, University of British Columbia.

Brunini, Germán I., Danilo R. Velis, and Juan I. Sabbione. 2021. “Seismic Moment Tensor Inversion in Anisotropy Media Using Deep Neural Networks.” In *Anales*, 5 pages. San Juan, Argentina.

Chen, Yangkang. 2018. “Automatic Velocity Analysis Using High-Resolution Hyperbolic Radon Transform.” *Geophysics* 83 (4): no. 4, A53–A57.

Chopra, Satinder, and Glen Larsen. 2000. “Acquisition Footprint–Its Detection and Removal.” *CSEG Recorder* 25 (8): 16–20.

Davis, L., ed. 1987. *Genetic Algorithms and Simulated Annealing*. Los Altos, CA: Morgan Kaufmann Publishers.

Drummond, J. M., Arthur J. L. Budd, and James W. Ryan. 2000. “Adapting to Noisy 3D Data - Attenuating the Acquisition Footprint.” *SEG Technical Program Expanded Abstracts*, 9–12. https://doi.org/10.1190/1.1816247.

Elad, Michael. 2010. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer Science & Business Media.

Falconer, Scott, and Kurt J. Marfurt. 2008. “Attribute-Driven Footprint Suppression.” *SEG Technical Program Expanded Abstracts*, 2667–71. https://doi.org/10.1190/1.3063897.

Fortini, Carlo, Davide Maggi, Vincenzo Lipari, and Maurizio Ferla. 2013. “Particle Swarm Optimization for Seismic Velocity Analysis.” In *Expanded Abstracts*, 4864–8. SEG.

Garabito, German. 2018. “Global Optimization Strategies for Implementing 3D Common-Reflection-Surface Stack Using the Very Fast Simulated Annealing Algorithm: Application to Real Land Data.” *Geophysics* 83 (4): no. 4, V253–V261.

Gelpi, Gabriel, Daniel O. Pérez, and Danilo R. Velis. 2020. “Automatic Well Tying and Wavelet Phases Estimation with No Waveform Stretching or Squeezing.” *Geophysics* 85 (3): D83–D91.

Goldberg, D. E., ed. 1989. *Genetics Algorithms in Search, Optimization, and Machine Learning*. Addisson-Wesley Publishing Company, Inc.

Gómez, Julián L., and Danilo R. Velis. 2020. “Footprint Removal from Seismic Data with Residual Dictionary Learning.” *Geophysics* 85 (4): V355–V365. https://doi.org/10.1190/geo2019-0482.1.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT press.

Grechka, Vladimir. 2015. “On the Feasibility of Inversion of Single-Well Microseismic Data for Full Moment Tensor.” *Geophysics* 80 (4): KS41–KS49.

Grechka, Vladimir, and Werner M. Heigl. 2017. *Microseismic Monitoring*. Society of Exploration Geophysicists, Tulsa, OK. https://doi.org/10.1190/1.9781560803485.

Grechka, Vladimir I. 2015. “Moment Tensor Inversion of Single-Well Microseismic Data: Is It Feasible?” In *SEG Technical Program Expanded Abstracts 2015*, 2506–11. Society of Exploration Geophysicists.

Gülünay, Necati. 1999. “Acquisition Geometry Footprints Removal.” *SEG Technical Program Expanded Abstracts*, 637–40. https://doi.org/10.1190/1.1821103.

Gülünay, Necati, F. Martin, and R. Martinez. 1994. “3D Data Acquisition Artifacts Removal by Spot Editing in the Spatial-Temporal Frequency Domain.” *56th Annual International Conference and Exhibition, EAGE, Extended Abstracts, H049*.

Herrera, H., S. Fomel, and M. van der Baan. 2014. “Automatic Approaches for Seismic to Well Tying.” *Interpretation* 2 (2): SD101–SD109.

Herron, D. 2011. *First Steps in Seismic Interpretation*. Society of Exploration Geophysicists. https://doi.org/10.1190/1.9781560802938.

Ingber, L. 1989. “Very Fast Simulated Re-Annealing.” *Journal of Mathematical Computation and Modelling* 12: 967–73.

Kirkpatrick, S., C. D. Jr. Gellat, and M. P. Vecchi. 1983. “Optimization by Simulated Annealing.” *Science* 220: 671–80.

Laarhoven, P. I. M. van, and E. H. L. Aarts. 1988. *Simulated Annealing: Theory and Applications*. Dordrecht: D. Riedel.

Li, Chengbo, Yu Zhang, and Charles C. Mosher. 2019. “A Hybrid Learning-Based Framework for Seismic Denoising.” *The Leading Edge* 38 (7): 542–49. https://doi.org/10.1190/tle38070542.1.

Mallat, Stephane. 1999. *A Wavelet Tour of Signal Processing: The Sparse Way*. 3rd ed. Academic Press.

Marfurt, Kurt J., Ronald M. Scheet, John A. Sharp, and Mark G. Harper. 1998. “Suppression of the Acquisition Footprint for Seismic Sequence Attribute Mapping.” *Geophysics* 63 (3): 1024–35. https://doi.org/10.1190/1.1444380.

Muñoz, A., and D. Hale. 2015. “Automatic Simultaneous Multiple Well Ties.” *Geophysics* 80 (8): IM45–IM51.

Newrick, R. 2012. *Well Tie Perfection - 52 Things You Should Know About Geophysics*. Agile Libre.

Ovcharenko, Oleg, Jubran Akram, and Daniel Peter. 2018. “Feasibility of Moment Tensor Inversion from a Single Borehole Data Using Artificial Neural Networks.” *Search and Discovery*.

Park, Min Jun, and Mauricio D. Sacchi. 2020. “Automatic Velocity Analysis Using Convolutional Neural Network and Transfer Learning.” *Geophysics* 85 (1): no. 1, V33–V43. https://doi.org/10.1190/geo2018-0870.1.

Qadrouh, AN, JM Carcione, M Alajmi, and MM Alyousif. 2019. “A Tutorial on Machine Learning with Geophysical Applications.” *Bollettino Di Geofisica Teorica Ed Applicata* 60 (3).

Sahai, Surinder K, and Khalid A Soofi. 2006. “Use of Simple 2-D Filters to Reduce Footprint Noise in Seismic Data.” *Geohorizons* 7: 14–17.

Samuel, Arthur L. 1959. “Some Studies in Machine Learning Using the Game of Checkers.” *IBM Journal of Research and Development* 3 (3): 210–29.

Simm, R., and M. Bacon. 2014. “Seismic Amplitude: An Interpreter’s Handbook.” In. Cambridge University Press.

Soubaras, Robert. 2002. “Attenuation of Acquisition Footprint for Non-Orthogonal 3D Geometries.” *SEG Technical Program Expanded Abstracts*, 2142–5. https://doi.org/10.1190/1.1817129.

Sripanich, Yanadet, Sergey Fomel, Jeannot Trampert, William Burnett, and Thomas Hess. 2020. “Probabilistic Moveout Analysis by Time Warping.” *Geophysics* 85 (1): no. 1, U1–U20.

Storn, R., and K. Price. 1997. “Differential Evolution – a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces.” *Journal of Global Optimization* 11: 341–59.

Tošić, Ivana, and Pascal Frossard. 2011. “Dictionary Learning: What Is the Right Representation for My Signal?” *IEEE Signal Processing Magazine* 28 (2): 27–38.

Turquais, Pierre, Endrias G. Asgedom, and Walter Söllner. 2017a. “A Method of Combining Coherence-Constrained Sparse Coding and Dictionary Learning for Denoising.” *Geophysics* 82 (3): V137–V148. https://doi.org/10.1190/geo2016-0164.1.

———. 2017b. “Coherent Noise Suppression by Learning and Analyzing the Morphology of the Data.” *Geophysics* 82 (6): V397–V411. https://doi.org/10.1190/geo2017-0092.1.

Vavryčuk, Václav, and Daniela Kühn. 2012. “Moment Tensor Inversion of Waveforms: A Two-Step Time-Frequency Approach.” *Geophysical Journal International* 190 (3): 1761–76.

Velis, Danilo R. 2021. “Simulated Annealing Velocity Analysis: Automating the Picking Process.” *Geophysics* 86 (6).

Walden, A. T., and R. E. White. 1984. “On Errors of Fit and Accuracy in Matching Synthetic Seismograms and Seismic Traces.” *Geophysical Prospecting* 32: 871–91.

Wamriew, Daniel Stephen, Marwan Charara, Evgenii Maltsev, and others. 2020. “Deep Neural Network for Real-Time Location and Moment Tensor Inversion of Borehole Microseismic Events Induced by Hydraulic Fracturing.” In *SPE Russian Petroleum Technology Conference*. Society of Petroleum Engineers.

White, R., and R. Simm. 2003. “Tutorial: Good Practice in Well Ties.” *EAGE First Break* 21: 75–83.

Yilmaz, Özdoân. 2001. *Seismic Data Analysis: Processing, Inversion, and Interpretation of Seismic Data*. Investigations in Geophysics. SEG.

Ziolkowski, A., J. R. Underhill, and R. G. K. Johnston. 1998. “Wavelets, Well Ties, and the Search for Subtle Stratigraphic Traps.” *Geophysics* 63 (1): P297–P313.

Zu, Shaohuan, Hui Zhou, Rushan Wu, Maocai Jiang, and Yangkang Chen. 2019. “Dictionary Learning Based on Dip Patch Selection Training for Random Noise Attenuation.” *Geophysics* 84 (3): V1–V15. https://doi.org/10.1190/geo2018-0596.1.