Detecting karst caverns by pattern recognition

Fantine Huot and Robert Clapp

ABSTRACT

Strong localized heterogeneities in the subsurface, such as karst caverns and sinkholes, cause scattering of seismic waves, thereby degrading the images obtained in conventional processing. We explore the possibility of using pattern recognition techniques for detecting these strong heterogeneities from seismic data. Through synthetic models, we generate data with significant scattering. We then perform reverse time migration (RTM) and use various pre-processing techniques to engineer features fit for supervised learning algorithms. Eventually, we use support vector machines (SVM) to classify these features and retrieve the approximate cavern locations.

INTRODUCTION

The Tengiz carbonate platform in northwestern Kazakhstan is one of the largest producing oil fields in the world. Recently, exploration has targeted karst-like zones with cavernous porosity along the margin of the platform. Lester et al. (2015) showed that these karst caverns appear as localized high-amplitude events on seismic volumes but can also resemble residual noise that may have persisted through processing and imaging. Such localized features induce positional uncertainty in the migrated velocity model and can represent drilling hazards. While various methodologies such as diffraction migration or beam migration (Fomel et al., 2007; Berkovitch et al., 2009; Lester et al., 2015) have been proposed to address the issue of imaging these strong heterogeneities, herein we investigate the potential of techniques commonly used in pattern recognition.

The first algorithm for pattern recognition was introduced 80 years ago (Fisher, 1936). With the advent of computers and the information age, statistical learning has become a highly explored field in many scientific areas as well as marketing, finance, and other business disciplines. In recent years, new and improved software packages have significantly eased the implementation burden for many statistical learning methods, providing scientists and practitioners with complete toolkits for training, testing, and deploying models with well-documented examples for all these tasks (Collobert et al., 2002; Pedregosa et al., 2011; James et al., 2013; Jia et al., 2014). With algorithms automatically tracking faces in photographs (Osuna et al., 1997), what would prevent us from training machines to detect specific seismic responses in our data?

We first generate synthetic seismic data from a cavern model. We then perform reverse time migration (RTM) and apply various pre-processing techniques, such as continuous wavelet transforms (CWT) and principal component analysis (PCA), to build appropriate input features for our pattern recognition problem. We then train a support vector machine (SVM) classifier to detect the migrated seismic signature associated with caverns and apply this classifier to different portions of the data to retrieve the approximate cavern locations.

SYNTHETIC DATA GENERATION

Cavern model

The methodology used in this study is based on the one presented by Huot and Clapp (2016). We start by generating seismic data from a synthetic cavern model.

For this purpose, we create a three-dimensional synthetic model of an underground karst channel system in a limestone bedding as illustrated in Figure 1. Caverns of different scales are inserted at random locations in the model and connected with channels partially filled with water. In the following, we use four 2D slices along the y-axis to generate data which we use to train our machine learning model, and a 2D slice along the x-axis to generate data to test our classification performance.

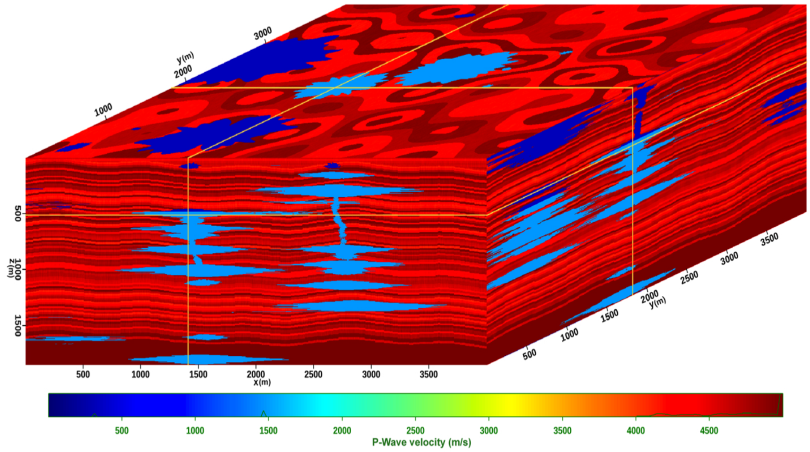

Figure 1: 3D synthetic cavern model built for this study. It features a randomly generated karst channel system in a limestone bedding. The model is color-coded by P-wave velocity. The detailed description of this model is provided in Huot and Clapp (2016). [**NR**]

Data modeling

For each of these 2D model slices, we generate synthetic seismic data using wave propagation code based on the one developed by Alves (2015). We use a Ricker-type explosive source with a peak frequency of 20 Hz. The recordings for a single shot clearly illustrate the scattering effect associated with the presence of the caverns (Figure 2(b)). From the zero-offset common midpoint (CMP) gathers, we observe that the caverns incoherently scatter the reflected energy (Figure 2(c)), thereby preventing accurate identification of the cavern locations.

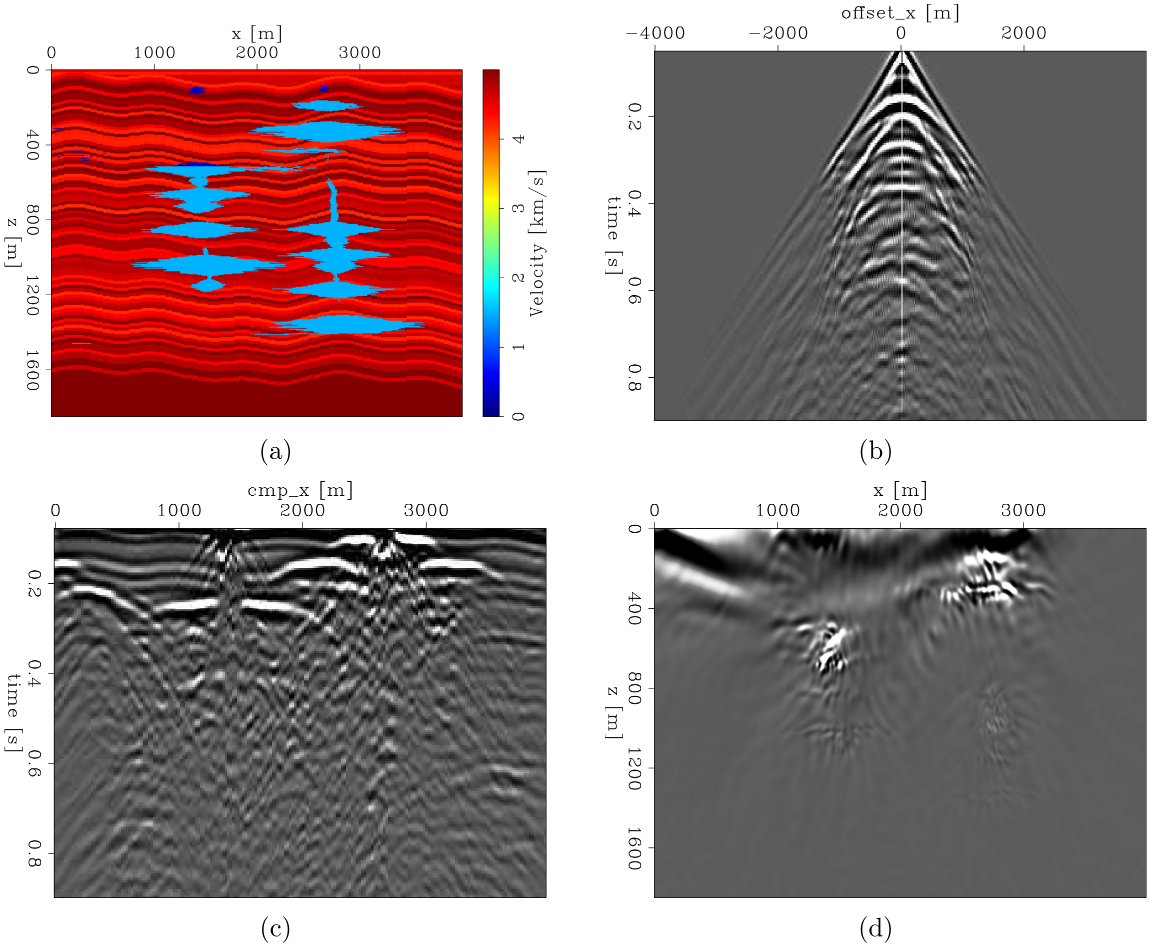

Figure 2: (a) One of the 2D model slices from which we generate data to train our classification algorithm. (b) Single shot gather obtained from this model slice with a source located at the surface at x = 2000 m. (c) Zero-offset common midpoint gather. (d) RTM image generated from this model slice. We clearly distinguish the scattering effect due to the presence of the caverns. [**CR**]

BUILDING FEATURES FOR MACHINE LEARNING

We now process the generated data to build features with a suitable format for machine learning algorithms.

To set up a classification problem, we need to associate each portion of the data with a binary label indicating whether or not it corresponds to a cavern location.

Therefore, we first perform reverse time migration (RTM) in order to map the data to model space, allowing us to label the migrated image points using the synthetic model slices as reference. Figure 2(d) provides an example of one of these RTM images. The RTM image also suffers from the strong scattering due to the presence of the caverns.

Starting with the labeled RTM images, we then apply various processing steps to compute multiple data features:

- We perform both acoustic and elastic modeling and migration, creating two sets of features.
- We apply a Laplacian filter on the RTM images to attenuate the migration artifacts (Biondi, 2006).
- We apply gain power with depth to compensate for amplitude attenuation.
- We apply Gaussian smoothing and spatial frequency bandpassing to reduce high frequency jitter.
- We apply continuous wavelet transforms (CWT). CWT are commonly used in pattern recognition, as they have the ability to decompose complex patterns into elementary forms. They measure the similarity between a signal and an analyzing wavelet by comparing the input signal to shifted and compressed or stretched versions of the wavelet. An overview of CWT is provided in Huot and Clapp (2016). In this study, we use both Ricker and Morlet wavelets as the mother wavelets, producing multiple sets of features. To each image point, we associate the full panel of dyadic frequencies obtained after applying CWT, as described by Huot and Clapp (2016).
- We run a sliding window of three different sizes ($8\,\text{m} \times 8\,\text{m}$, $12\,\text{m} \times 12\,\text{m}$, $16\,\text{m} \times 16\,\text{m}$) over the image. We associate the full set of features captured by each sliding window to the image point located at its center.
- We standardize all the variables to have zero mean and standard deviation one.
- For faster computation, we perform principal component analysis (PCA). When faced with a large set of correlated variables, principal components allow us to summarize the dataset with a smaller number of representative variables that collectively explain most of the variability in the original set. A full overview of PCA is provided in Huot (2016). Herein, for each feature set, we use the minimum number of principal components that collectively explain at least 90% of the total variance.

Using different combinations of these processing steps, we build multiple sets of features and data characteristics associated with binary labels. We obtain distinct sets of training features and testing features using the data generated from the different 2D model slices.

CLASSIFICATION

Support vector machines (SVM)

In the following, we set up a classification algorithm to retrieve the cavern locations from the data features we have designed. As our problem contains highly correlated features, we decide to use support vector machines (SVM), known to be efficient for these type of problems (Hsu et al., 2003; Hastie et al., 2005).

SVM classifiers use large sets of labeled training data to build a decision function. They can then be applied on other portions of data to predict the corresponding labels. SVM classifiers are effective in high dimensional spaces, and use only a subset of training points in the decision function, making them memory efficient. They are versatile as many different kernel functions can be specified for the decision function, making it possible to define non-linear boundaries. An overview of the theory behind SVM is provided in Huot and Clapp (2016).

Implementation

In recent years, new and improved software packages have significantly eased the implementation burden for many statistical learning methods. In this study, we use the following Python packages:

- For data visualization and performing operations on data: pandas (http://pandas.pydata.org/)
- For machine learning models: scikit-learn (http://scikit-learn.org/stable/)
- For plotting data: matplotlib (http://matplotlib.org/)
- For progress monitoring: tqdm (https://pypi.python.org/pypi/tqdm)

Evaluation score

To evaluate how well our SVM classifier retrieves the cavern locations, we have to introduce an evaluation score. Common metrics used for binary classification are precision, recall and F1-score (Hastie et al., 2005). A visual representation of these metrics is provided in Figure 3. Precision is the fraction of retrieved instances that are relevant, while recall is the fraction of relevant instances that are retrieved:

$$\text{Precision} = \frac{\text{True positives}}{\text{True positives} + \text{False positives}}$$

$$\text{Recall} = \frac{\text{True positives}}{\text{True positives} + \text{False negatives}}$$

The F1-score is the weighted harmonic mean of precision and recall, and hence provides a combined measure:

$$F = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

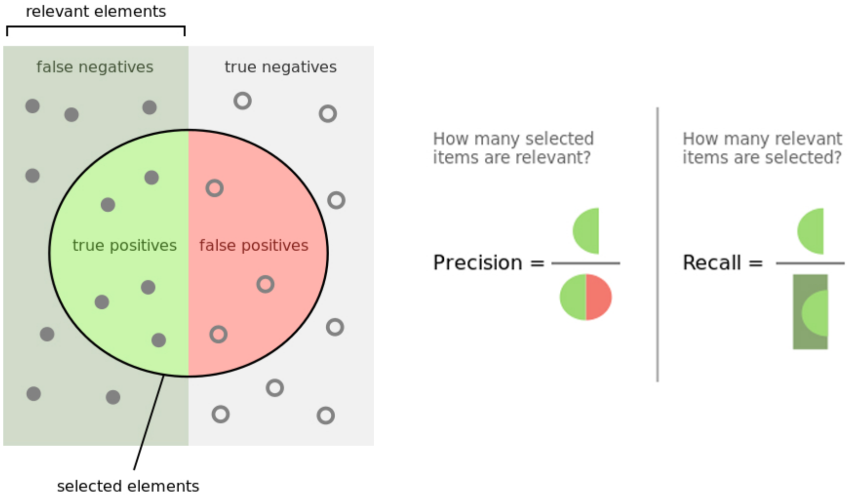

Figure 3: Visual representation of common metrics used for binary classification: precision and recall. Source: en.wikipedia.org/wiki/Precision_and_recall/media/File:Precisionrecall.svg [**NR**]

Classification results

We build an SVM classifier for each different feature set. We use a radial basis function (RBF) kernel. For each classifier, we select the SVM cost parameter C that provides best evaluation score among 7 different values : 0.001, 0.01, 0.1, 1, 10, 100, 1000. A detailed description of these parameters is provided in Huot and Clapp (2016).

The feature sets that yield best classification results are those that have CWT in the pre-processing steps. The sliding window also improves classification results significantly. However, even the best classification results we obtained return many false positives, as illustrated in Figure 4. The associated performance metrics are provided below:

	Precision	Recall	F1-score
Limestone bedding	0.99	0.95	0.97
Caverns	0.40	0.80	0.53
Avg / total	0.97	0.94	0.95

When examining the classification results, it appears that the classifier also returns a certain number of false positives on the training data, indicating that our classification model suffers from high bias. This problem can potentially be improved by adding more features. Instead of testing sets of features one-by-one to identify which features yield best evaluation score, it may be preferable to provide the classifier with the full set features that can be computed from all the pre-processing steps, and skip the PCA.

It also appears that the test error decreases with the number of 2D model slices used for the training data, suggesting that a larger training set will help.

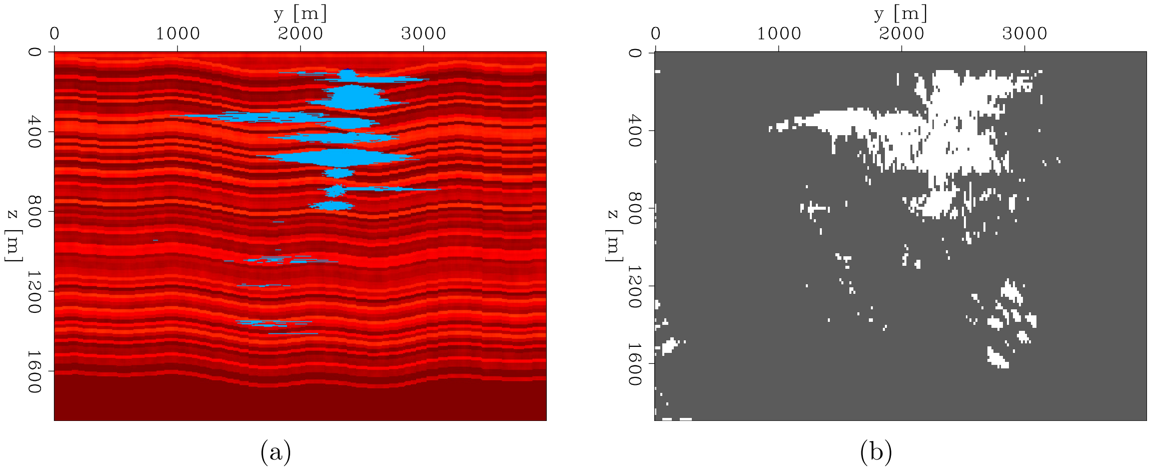

Figure 4: (a) 2D model slice on which we test our classification algorithm. (b) Classification results. While we seem to retrieve the approximate cavern location, the classifier returns many false positives. [**CR**]

DISCUSSION

The classification results indicate that it is possible to retrieve approximate karst cavern locations from seismic recordings using pattern recognition algorithms. Migration is an essential step for accurate labeling of caverns. However, the classification tests returned many false positives. These results seem to indicate that the classifier would benefit from using more input features and larger training data. Another idea would be to change classification algorithm for an edge detection method.

The next step would be to explore whether a classifier could be trained on synthetic data, which can be conveniently labeled, and be used to predict cavern locations on field data. We were provided a with a three-dimensional mapping of the cavern locations of the Lechuguilla channel system from the Carlsbad Caverns National Park. It features a fine and intricate channel system with caverns of various scales. This would allow us to generate data on a configuration that is closer to reality than our simplistic synthetic model.

ACKNOWLEDGMENTS

We would like to thank Gustavo Catao Alves for his help with his elastic modeling code.

REFERENCES

Alves, G., 2015, Adjoint formulation for the elastic wave equation: SEP-Report, **158**, 133–150.

Berkovitch, A., I. Belfer, Y. Hassin, and E. Landa, 2009, Diffraction imaging by multifocusing: Geophysics, **74**, WCA75–WCA81.

Biondi, B., 2006, 3d seismic imaging: Society of Exploration Geophysicists.

Collobert, R., S. Bengio, and J. Mariéthoz, 2002, Torch: a modular machine learning software library: Technical report, IDIAP.

Fisher, R. A., 1936, The use of multiple measurements in taxonomic problems: Annals of eugenics, **7**, 179–188.

Fomel, S., E. Landa, and M. T. Taner, 2007, Poststack velocity analysis by separation and imaging of seismic diffractions: Geophysics, **72**, U89–U94.

Hastie, T., R. Tibshirani, J. Friedman, and J. Franklin, 2005, The elements of statistical learning: Data mining, inference and prediction: 534–552.

Hsu, C.-W., C.-C. Chang, C.-J. Lin, et al., 2003, A practical guide to support vector classification.

Huot, F., 2016, Exploratory data analysis of the sola land dataset: SEP–Report, **165**.

Huot, F. and R. Clapp, 2016, Detecting karst caverns by pattern recognition : SEP-Report, **163**, 297–307.

James, G., D. Witten, T. Hastie, and R. Tibshirani, 2013, An introduction to statistical learning: Springer.

Jia, Y., E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, 2014, Caffe: Convolutional architecture for fast feature embedding: Proceedings of the ACM International Conference on Multimedia, 675–678.

Lester, R., E. Liebes, R. Hill, S. Jenkins, A. Makedonov, et al., 2015, Investigating mega-amplitudes in tengiz carbonates through interactive model-building and gaussian beam migration.

Osuna, E., R. Freund, and F. Girosi, 1997, Training support vector machines: an application to face detection: Computer vision and pattern recognition, 130–136.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., 2011, Scikit-learn: Machine learning in python: The Journal of Machine Learning Research, **12**, 2825–2830.