Learning to Classify Seismic Images with Deep Optimum-Path Forest

Luis Afonso
Department of Computing
Federal University of São Carlos
São Carlos - SP, Brazil
sugi.luis@gmail.com

Alexandre Vidal, Michelle Kuroda
Institute of Geology
University of Campinas
Campinas - SP, Brazil
vidal@ige.unicamp.br
mckuroda@gmail.com

Alexandre Falcão
Institute of Computing
University of Campinas
Campinas - SP, Brazil
afalcao@ic.unicamp.br

João Papa
Department of Computing
São Paulo State University
Bauru - SP, Brazil
papa@fc.unesp.br

***Abstract*—Due to the lack of labeled information, clustering techniques have been paramount in the last years once more. In this paper, inspired by the deep learning phenomenon, we presented a multi-scale approach to obtain more refined cluster representations of the Optimum-Path Forest (OPF) classifier, which has obtained promising results in a number of works in the literature. Here, we propose to fill a gap in OPF-based works by using a deep-driven representation of the feature space. Additionally, we validated the work in the context of high resolution seismic images aiming at petroleum exploration, as well as in general-purpose applications. Quantitative and qualitative analysis are conducted in order to assess the robustness of the proposed approach.**

***Keywords*-Optimum-Path Forest, Image Clustering, Deep Representations, Seismic Images**

I. Introduction

Image classification plays an important role in several application domains, which range from medical image analysis to remote sensing-driven tools. However, the lack of labeled data has oriented researchers towards active learning-based techniques, which consider the user feedback to improve the classification process by labeling samples. Although promising results have been obtained in the last years, labeling images is time-consuming and it strongly depends on the human skills, which can be prone to errors as well.

The Big Data era has made available tons of digital content daily, making even more tedious the task of analyzing data by hand. In this context, a foreseeable future can be drawn: we shall not be in lockstep with the amount of data generated, thus paying the price of having important information discarded and/or meaningless. One of the first waves towards the lack of labeled data refers to the so-called deep learning, which basically ends up learning features in an unsupervised fashion [1].

Therefore, unsupervised learning has gained attention in the last years once more, but still posing a tougher challenge than supervised learning, since the notion of a cluster can somehow be doubted and personally-driven. In this scenario, a number of techniques can be highlighted, such as k-means [2], Mean-Shift [3], Self-Organizing Maps (SOM) [4] and others [5], just to name a few. The so-called k-means works surprisingly well in many situations, despite its simplicity.

However, k-means has also some well-known shortcomings: (i) first, the user is required to feed the technique with the number of clusters, and (ii) the problem itself is essentially an optimization task, in which the distance of each sample to its nearest center is minimized. For the first statement, although some scientists argue the parameter k can be seen as a meta-parameter, in fact it requires us to have some knowledge about the "unsupervised" problem, in which by definition we should not have any information so far. The second statement concerns with any non-convex optimization problem, which may get trapped from local optima. Although a number of works have focused on such drawbacks, there is still room for improvements, since there is no "exact solution" to the problem, which means approximations that may cost some computational burden can be derived and thus employed by researchers worldwide.

Graph-based clustering techniques have their appeal as well. Roughly speaking, the idea is to encode each feature vector as a graph node, and then to learn some connectivity function that can group "similar" samples and turn others far apart. Notice the notion of "similarity" also poses an interesting problem, which can be of extremely importance to the success of the technique. Some years ago, Rocha et al. [6] presented the unsupervised version of the Optimum-Path Forest (OPF) classifier, which basically models the problem of clustering data as a competition process, in which some key samples compete among themselves in order to gather others. OPF has gained considerable attention in the last years, since its supervised version has been similarly accurate as Support Vector Machines for some applications, but faster for training [7], [8].

Unsupervised OPF, hereinafter called OPF, has one parameter only (i.e., k_{max}), which requires much less knowledge than k-means with respect to the problem itself. Additionally, OPF computes the number of clusters on-the-fly, which is an interesting skill considering applications where one does not know that information, and wants to find it out. Some examples are related to data representation using bag-of-visual-words, in which the size of dictionary (i.e., the number of visual words) is of extremely importance to the success of the technique. Usually, the user does not have such information, but is eager to discover it. Recently, Afonso et al. [9] have successfully employed OPF for such task. Last but not least, OPF can obtain similar clusters' centers when compared to k-means, which is an interesting property since the latter is undoubtedly recognized to work well in several problems [10].

However, a weakness of OPF is directly related to its strength: "what if one does know in advance the number of clusters?" In fact, we have no information about any OPF-related paper that deals with that problem. Actually, if one needs to somehow control the number of clusters, we can play with k_{max} until that desired number is reached. Moreover, there is no guarantee about that. Roughly speaking, one can deal with that by experimentally trying different values for k_{max} within the range $[1, k_{max}]$ that can reach or, at least, to be close as to the desired number of clusters. Nonetheless, to try out all possible values within that range, as proposed by Rocha et al. [6], might be prohibitive. Recently, Costa et al. [11] modeled this problem as a meta-heuristic-based optimization task, being the results quite close to the ones obtained by the optimal approach proposed by Rocha et al. [6], but being faster. However, it is noteworthy to mention the aforementioned works were proposed to find suitable values for $k^* \in [1, k_{max}]$ to create the k^*-neighborhood, and not to establish a proper number of classes[1], although the design of the neighborhood size directly influences the number of clusters.

In this work, we propose to perform OPF clustering at different levels of abstractions (i.e., scales) in a deep-driven approach to be as closest as possible to the number of clusters required by that specific application. By deep we mean we are going to perform unsupervised learning using different "views" of the data until we may reach some desirable result. As aforementioned, OPF makes use of key samples, hereinafter called *prototypes*, which compete among themselves trying to offer the "best" reward (path-cost function) to the remaining samples. By taking into account the prototypes chosen at the very first (initial) step, we can thus use them as the new (and only) samples to the next clustering step. Since the number of prototypes is often much smaller than the number of dataset samples, we have a more compressed representation of the dataset at each level, and thus less clusters (in fact, each cluster is represented by one prototype). In this paper, we show it is possible to obtain the desired number of clusters (or at lest to be close to that) using few scales of representation. Such methodology is much faster than playing around with the values of k_{max}, as proposed in the works by Rocha et al. [6] and by Costa et al. [11].

A detailed look at the papers published in the last years has revealed only one work similar to ours [12], but still with a different purpose. This very recent work employed the OPF clustering for automatic video summarization, using the prototypes obtained in the first step as the shots and key frames to represent a reduced version of the video. Soon after, these key frames are clustered again to obtain more refined representations of the video summary. But clearly, we are interested into working on the problem of restricting OPF to a predefined (desired) number of clusters. Another contribution of this work is to evaluate OPF in the context of seismic-based images concerning the task of hydrocarbon accumulation detection, which are used to detect possible locations for petroleum exploration. As far as we know, OPF has never been used in this context so far. This work also explores the proposed OPF clustering in a wider context by applying it in three large labeled datasets. The remainder of this paper is organized as follows. Sections II and III present a brief theoretical background about OPF and the proposed deep-based approach to obtain different levels of representations, and thus less clusters, respectively. The methodology and experiments are discussed in Section IV, and Section V states conclusions and future works.

II. Optimum-Path Forest Clustering

Let $\mathcal{N}$ be a dataset such that for every sample $s \in \mathcal{N}$ there is a feature vector $\vec{v}(s)$. Let $d(s,t)$ be the distance between samples s and t in the feature space (e.g., $d(s,t) = \|\vec{v}(t) - \vec{v}(s)\|$). The fundamental problem in data clustering is to identify natural groups in $\mathcal{N}$.

A graph $(\mathcal{N}, \mathcal{A})$ is defined such that the arcs $(s,t) \in \mathcal{A}$ connect k-nearest neighbors in the feature space. The arcs are weighted by $d(s,t)$ and the nodes $s \in \mathcal{N}$ are weighted by a density value $\rho(s)$, given by:

$$\rho(s) = \frac{1}{\sqrt{2\pi\sigma^2}|\mathcal{A}(s)|} \sum_{\forall t \in \mathcal{A}(s)} \exp\left(\frac{-d^2(s,t)}{2\sigma^2}\right), \quad (1)$$

where $|\mathcal{A}(s)| = k$, $\sigma = \frac{d_f}{3}$, and d_f is the maximum arc weight in $(\mathcal{N}, \mathcal{A})$. This parameter choice considers all nodes for density computation, since a Gaussian function covers most samples within $d(s,t) \in [0, 3\sigma]$. The traditional method to estimate a probability density function (pdf) is by Parzen-window. Equation (1) can provide a Parzen-window estimation based on isotropic Gaussian kernel when we define the arcs by $(s,t) \in \mathcal{A}$ if $d(s,t) \leq d_f$. This choice, however, presents problems with the differences in scale and sample concentration. Solutions for this problem lead to adaptive choices of d_f depending on the region of the feature space [3]. By taking into account the k-nearest neighbors, we are handling different concentrations and reducing the scale problem to the one of finding the best value of k within $[1, k_{\max}]$, for $1 \leq k_{\max} \leq |\mathcal{N}|$. The solution provided by Rocha et al. [6] considers the minimum graph cut provided by the clustering results for $k \in [1, k_{\max}]$, according to a measure suggested by Shi and Malik based on graph cuts [13].

Let a path π_t be a sequence of adjacent samples starting from a root $R(t)$ and ending at a sample t, being $\pi_t = \langle t \rangle$ a trivial path and $\pi_s \cdot \langle s,t \rangle$ the concatenation of π_s and arc (s,t). Among all possible paths π_t with roots on the maxima of the pdf, we wish to find a path whose the lowest density

[1] In this context, k^* stands for the neighborhood size that minimizes some fitness function (i.e., the minimum graph cut in that case).

value along it is maximum. Each maximum should then define an influence zone (cluster) by selecting the samples that are more strongly connected to it, according to this definition, than to any other maximum. More formally, we wish to maximize $f(\pi_t)$ for all $t \in \mathcal{N}$ where

$$\begin{aligned} f(\langle t \rangle) &= \begin{cases} \rho(t) & \text{if } t \in \mathcal{R} \\ \rho(t) - \delta & \text{otherwise} \end{cases} \\ f(\langle \pi_s \cdot \langle s, t \rangle \rangle) &= \min\{f(\pi_s), \rho(t)\} \end{aligned} \tag{2}$$

for $\delta = \min_{\forall (s,t) \in \mathcal{A} | \rho(t) \neq \rho(s)} |\rho(t) - \rho(s)|$ and $\mathcal{R}$ being a root set with one element for each maximum of the pdf. Higher values of delta reduce the number of maxima. We are setting $\delta = 1.0$ and scaling real numbers $\rho(t) \in [1, 1000]$ in this work. The OPF algorithm maximizes $f(\pi_t)$ such that the optimum paths form an optimum-path forest — a predecessor map P with no cycles that assigns to each sample $t \notin \mathcal{R}$ its predecessor $P(t)$ in the optimum path from $\mathcal{R}$ or a marker nil when $t \in \mathcal{R}$. In essence, each maximum of the pdf, i.e., prototype, will be the root of an optimum-path tree - OPT (cluster), and the collection of all OPTs originates the optimum-path forest that gives the name to the classifier.

III. Learning Deep Representations

As aforementioned, OPF can find the number of clusters on-the-fly, which means there is no need for such information beforehand. However, to the best of our knowledge, there is no proposed approach that can somehow "force" the number of clusters to a predefined number when one knows that information when dealing with OPF. Although we can play around with k_{max}, it can be prohibitive for large datasets, such as the one addressed in this work (high resolution seismic images).

In order to cope with this challenge, we propose to employ different representations (layers) of the dataset samples, being the first layer the original feature space to be clustered. After that, the prototypes at the first layer are then used as the new samples to compose the feature space at the second layer, which is clustered once again. The very same process is repeated until the predefined number of clusters (or at least close to) is reached. Since the OPF clustering prototypes are located in the highest density regions, they are very suitable to represent nearby samples, as argued in the works conducted by Castelo and Calderón-Ruiz [12] and Afonso et al. [9].

Let $\mathcal{S}_i$ be the set of prototypes at layer L_i, $i = 1, 2, \ldots, l$, where l stands for the number of layers. Since each root will be the maximum of a pdf (Equation 2), we have a set of samples that fall in the same optimum-path tree and are encoded by the very same prototype (root of that tree) in the next layer. In short, the higher the number of layers, the less prototypes (clusters) one shall find, i.e., $|\mathcal{S}_1| < |\mathcal{S}_2| < \ldots < |\mathcal{S}_L| < \ldots \leq 1$. Therefore, at a very coarse layer, one shall find only one cluster. Figure 1 displays the proposed OPF-based architecture for deep-driven feature space representation.

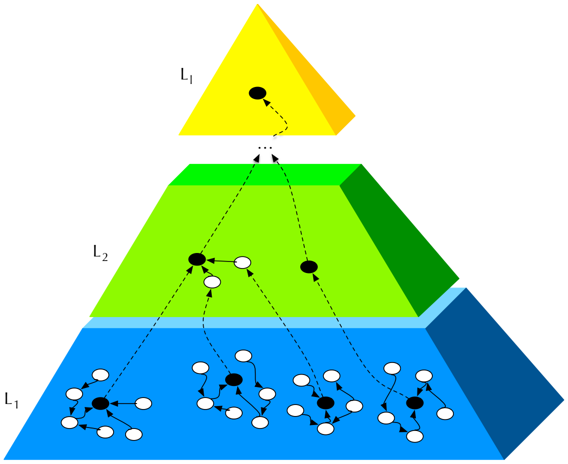

Fig. 1: Proposed approach based on coarser representations of the feature space.

At layer L_1, we can observe four clusters (optimum-path trees), where the black nodes stand for the set of prototypes at that layer, i.e., $\mathcal{S}_1$. Some of these prototypes will become new prototypes at L_2, and others not (we can observe both black and white nodes at layer 2). This process is carried out up to the last layer specified by the user. Notice at the very coarser scale, i.e., L_l, we shall find only one cluster.

IV. Methodology and Experimental Results

In order to provide both qualitative and quantitative insights of the OPF-driven approach, we divided the experiment section in two. In the first part, OPF is applied in a set of seismic images enabling to visualize the clustering result. The second part makes use of large labeled datasets so we can obtain some metrics that indicate the OPF clustering quality.

A. Seismic Images

In order to validate the proposed OPF-driven approach to obtain finer representations of the clustered space, we used seismic images from the North Sea at a specific location in the Dutch sector, the so-called "Netherlands Offshore F3 Block Complete" dataset[2]. The dataset has around 466 images (i.e., slices) that are combined together to form a volume that somehow models the geological information of the aforementioned location. In this paper, we used 5 images chosen at random for clustering purposes, say that: 924, 928, 932, 936 e 940. Note these numbers stand for the acquisition time in milliseconds of each image. Figure 2 depicts image 924, where the colors stand for different layers of rocks or their compactness in the sea floor, the green arrow points North and X1 stands for In-line.

Each dataset sample is composed of a pixel from the aforementioned images, thus resulting in 5 different datasets, where each pixel is represented by the seismic amplitude. Note the color intensities depicted in Figure 2 were used for

[2] https://opendtect.org/osr/pmwiki.php/Main

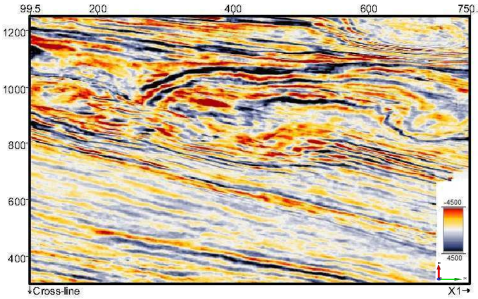

Fig. 2: Figure 924 of the F3 Block at the Dutch sector.

the sake of vizualization purposes only. After that, we then employed OPF with 4 layers against with the well-known k-means and the Self-Organizing Maps (SOM) for evaluation reasons. Aiming a fair comparison among the techniques, we used the very same number of clusters found by OPF at the last layer as the input to both k-means and SOM[3].

Notice the k_{max} value is strongly related to the number of desired clusters, i.e., the larger k_{max} value, the less clusters one shall have. The rationale behind that idea is related to the working mechanism used by OPF to find proper neighborhood sizes (k values), as explained in Section II. Since OPF performs a linear search within the range $k \in [1, k_{max}]$, if one uses larger k_{max} values we also increase the probability of finding larger values of k that minimize the graph cut criterion. Therefore, larger neighborhoods mean less clusters. As such, we employed different and decreasing values for k_{max} considering the different layers for all images. Considering the first and last layers, we used $k = 100$ and $k = 2$, respectively. In regard to the inner layers, we used 1% of the new dataset for layer 2, and 10% with respect to layer 3. Notice the dataset size decreases in the proposed approach as we move towards the upper layers. Table I presents the number of clusters found by OPF considering different layers for all images employed.

TABLE I: Number of clusters found by OPF at different layers.

	Layer			
Image	**1**	**2**	**3**	**4**
924	4,102	41	8	3
928	4,135	41	6	2
932	4,074	38	6	2
936	4,144	41	10	2
940	4,193	44	8	2

As aforementioned, one can observe the number of clusters decreases as we move to the upper layers. Clearly, one can obtain much less clusters by just using 2 layers. Additionally, it seems, at least for the images employed in this work, that 4 layers are enough to cope with the problem of seismic-driven image classification, since we achieved the minimum number of clusters we can work with. Figures 3, 4 and 5 illustrate the classified images at layer 4 with respect to OPF, k-means and SOM techniques, respectively, where the different colors represent the different labels. Note these figures refer to the image 924 (Figure 2).

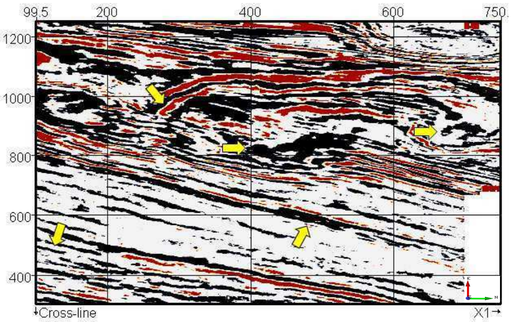

Fig. 3: Classified image 924 using OPF at layer 4.

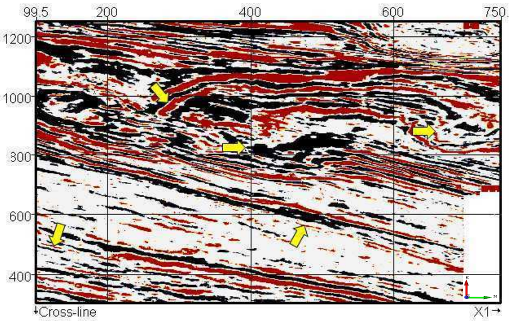

Fig. 4: Classified image 924 using k-means with 3 clusters.

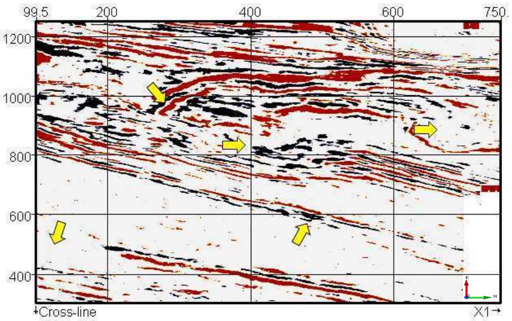

Fig. 5: Classified image 924 using SOM with 3 clusters.

In order to provide a qualitative comparison among the techniques, we asked for a geologist to provide insightful comments about the results. In regard of OPF and SOM results, one can observe the border of the reservoir body (the leftmost arrow) is not visible in the SOM results, which means a negative impact for exploration purposes. In addition, SOM was also unable to provide lateral continuity of reflections, i.e., some structures appear as a dashed-like lines instead of a continuous-like line. On the other hand, OPF classification overcame such issues by providing some details of the interest region and other main structures, besides lateral continuity.

[3] Recall that we employed k-means to label the SOM map after their learning.

Lateral continuity is important because it allows to identify the limits of a seismic body and any faults that may seal or conduct fluids off a reservoir. Finally, k-means was able to provide suitable levels of details, thus obtaining very good results as well.

B. General-purpose Images

This section aims to provide some quantitative insight concerning the proposed deep-driven OPF[4] by evaluating it against k-means[5], Mean-Shift[6] and SOM[7] over three large well-known labeled datasets:

- CIFAR-10[8]: it consists of 60,000 32 × 32 images distributed in 10 classes, being $6,000$ images per class. The training set has $50,000$ images and the remaining $10,000$ images are used to compose the testing set. The experiments used all $60,000$ images as a single set.
- CIFAR-100[8]: This dataset is similar to CIFAR-10, and it contains $60,000$ images distributed in 100 classes, being 600 images per each. The 100 classes represent a "finer" label, and are grouped into 20 superclasses as a "coarser" label. The experiments used the $60,000$ images and the coarse label for evaluating the clustering techniques.
- MNIST[9]: this dataset has a total of $70,000$ images of handwritten digits divided in 10 classes (one class for each digit), in which $60,000$ belong to the training set and the remaining images belong to the testing set. The digits are size-normalized and centered in a fixed-size image. All $70,000$ were used for the experiments as single set.

Figures 6, 7 and 8 depict some examples from CIFAR-10, CIFAR-100 and MNIST datasets, respectively.

In order to describe the images from all aforementioned datasets, we employed the Border/Interior Pixel Classification (BIC) [16] technique, which is a 64-dimensional descriptor. The reason for using such descriptor relies on its compactness and low dimensionality, since we are dealing with thousands of images to be clustered. As in Section IV-A, OPF is evaluated using a four-layer design, as well as following the same rules for setting the value of parameter k for each layer. Notice k-means and SOM used parameter k equals to the number of clusters found by the OPF on its last layer. The reason for using the very same value is to allow a fair comparison among the techniques[10].

Since Mean-Shift has no parameter k, its number of clusters is different from the other techniques in all datasets, but its clustering metrics are computed though. Mean-Shift found 4, 4 and 3 clusters in CIFAR-10, CIFAR-100 and MNIST datasets, respectively. Table II presents the number of clusters found by each layer of OPF considering each dataset. The very same number of clusters were computed by k-means and SOM. We considered that Mean-Shift did not obtain interesting results, since it has found less clusters than desired. Since CIFAR-100 contains 100 classes, it is expected to find out 100 clusters at least. If one considers the results in Table II, OPF has the flexibility to play around with different number of layers until a desired number of clusters has been found.

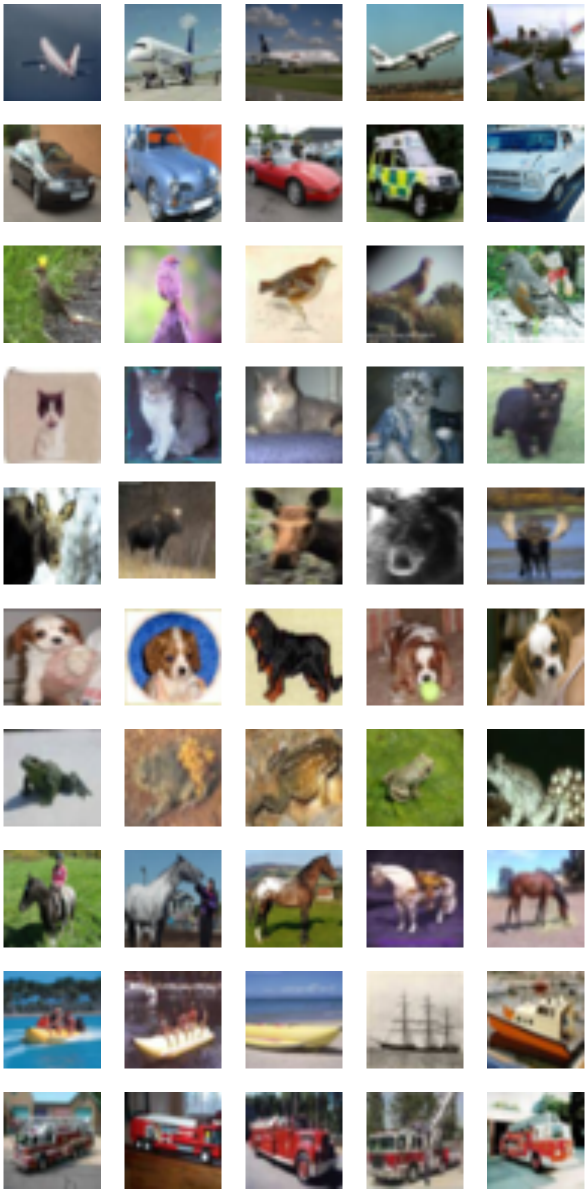

Fig. 6: Some samples from CIFAR-10 dataset.

TABLE II: Number of clusters found by OPF at different layers considering each dataset.

	Layer			
Dataset	**1**	**2**	**3**	**4**
CIFAR 10	137	121	17	8
CIFAR 100	216	163	24	15
MNIST	221	145	5	2

[4] In regard to OPF implementation, we used LibOPF [14].
[5] We used our own implementation.
[6] We employed an implementation provided by scikit-learn [15].
[7] http://somoclu.readthedocs.io/en/stable
[8] https://www.cs.toronto.edu/~kriz/cifar.html
[9] http://yann.lecun.com/exdb/mnist/
[10] Notice parameter k has distinct meaning for OPF, k-means and SOM. We decided to keep the same notation for the sake of simplicity.

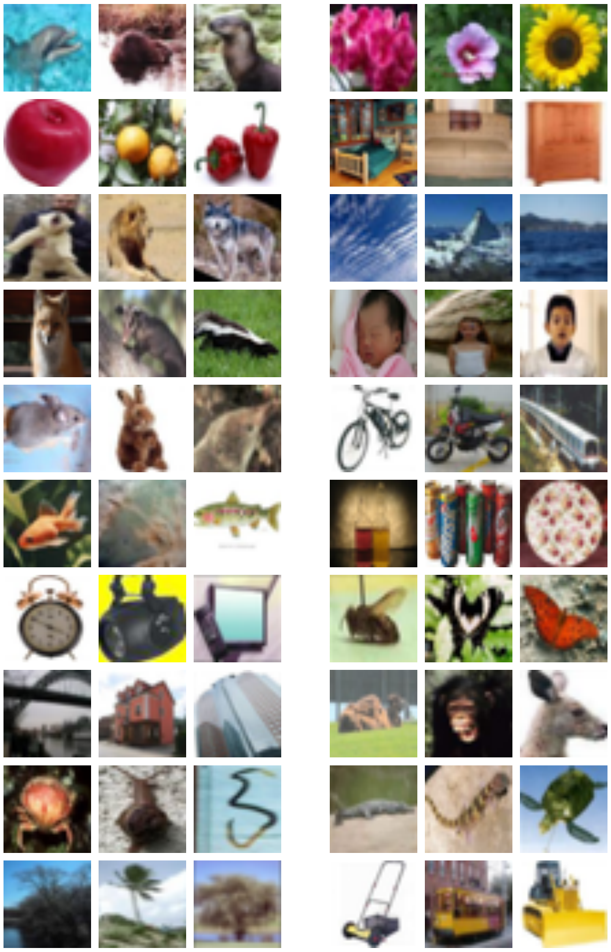

Fig. 7: Some samples from CIFAR-100 dataset.

Fig. 8: Some samples from MNIST dataset.

Since we have the true labels for each dataset, the overall performance is assessed by five metrics[11] which use the true and predicted labels for computation purposes, as follows:

- Homogeneity (H): this metric regards to how pure clusters are, in other words, clusters have maximum value of homogeneity if they contain only samples that belong to the same class. Notice $H \in [0, 1]$, where $H = 1$ denotes the best result.
- Completeness (C): a clustering result satisfies completeness if all samples that are members of a given class are elements of the same cluster. Notice $C \in [0, 1]$, where $C = 1$ denotes the best result.
- V-measure (V): this metric is the harmonic mean between homogeneity and completeness, as follows:

$$V = 2 * \frac{(H * C)}{(H + C)}. \tag{3}$$

Notice $V \in [0, 1]$, where $V = 1$ denotes the best result.

The results achieved by each algorithm are shown in Tables III, IV and V. The best results are in bold. Notice these results consider the number of clusters found by OPF at the layer.

TABLE III: Results for CIFAR10 dataset.

	Technique			
Metric	**OPF**	**k-means**	**Mean-Shift**	**SOM**
H	0.000	**0.054**	0.001	0.049
C	**0.153**	0.060	0.039	0.056
V	0.000	**0.057**	0.001	0.052

TABLE IV: Results for CIFAR100 dataset.

	Technique			
Metric	**OPF**	**k-means**	**Mean-Shift**	**SOM**
H	0.010	**0.033**	0.001	0.030
C	0.069	0.038	**0.077**	0.034
V	0.017	**0.035**	0.003	0.032

TABLE V: Results for MNIST dataset.

	Technique			
Metric	**OPF**	**k-means**	**Mean-Shift**	**SOM**
H	0.000	0.007	0.000	**0.073**
C	**1.000**	0.024	0.005	0.376
V	0.000	0.011	0.001	**0.122**

Since the number of clusters found in all datasets does not match the real number of classes, it is expected the clustering homogeneity of all techniques never reaches the maximum value. The k-means was able to find the most homogeneous clusters considering CIFAR-10 and CIFAR-100 datasets, and SOM obtained the best result concerning MNIST dataset. However, keep in mind all homogeneity values are pretty close to the minimum in all cases. Regarding the clustering completeness, OPF outperformed the other techniques in CIFAR-10 and MNIST datasets, achieving the maximum value in the latter one, thus meaning all samples for every class were grouped in the very same cluster. Since V-measure tries to balance both homogeneity and completeness, k-means

[11] http://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation

obtained the best V values concerning CIFAR-10 and CIFAR-100, and SOM outperformed all techniques in MNIST dataset. With respect to the efficiency, k-means was the fastest one, followed by OPF, SOM and Mean-Shift.

V. Conclusions

This paper presented a deep-driven approach that allows OPF to obtain coarser clustered images, being the problem of unsupervised learning decomposed in different layers. The proposed approach was evaluated in the context of seismic image classification, being its results comparable to the ones obtained by k-means and SOM techniques. In this specific case, OPF was able to provide visual details that are important to identify certain structures, where k-means and SOM were unable to highlight.

Considering general-purpose datasets, we can highlight OPF was able to found a number of clusters close to the real number of classes in CIFAR-10 (8 clusters found out of 10 classes) and CIFAR-100 (15 clusters found out of 20 classes) datasets. Although the numbers of clusters found in MNIST dataset is not close to the number of classes, OPF was able to cluster the whole dataset on either cluster 1 or cluster 2, thus achieving a completeness equals to 1. Regarding other techniques, k-means and SOM were able to find more homogeneous clusters in all situations.

The experimental section showed us the proposed deep-driven OPF allows the user a more flexible tool when working with unsupervised clustering-oriented applications where we know the desired number of clusters. Also, experiments over general-purpose datasets shed light over all techniques might be complementary to each other, since they obtained different results over distinct datasets and measures.

Acknowledgment

The authors would like to thank FAPESP grant #2014/16250-9, and CNPq grants #306166/2014-3, #479070/2013-0 and #302970/2014-2.

References

[1] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[2] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*. Berkeley, California: University of California Press, 1967, pp. 281–297.

[3] D. Comaniciu, "An algorithm for data-driven bandwidth selection," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 281–288, 2003.

[4] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, no. 1, pp. 59–69, 1982.

[5] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, September 1999.

[6] L. M. Rocha, F. A. M. Cappabianco, and A. X. Falcão, "Data clustering as an optimum-path forest problem with applications in image analysis," *International Journal of Imaging Systems and Technology*, vol. 19, no. 2, pp. 50–68, 2009.

[7] J. P. Papa, A. X. Falcão, and C. T. N. Suzuki, "Supervised pattern classification based on optimum-path forest," *International Journal of Imaging Systems and Technology*, vol. 19, no. 2, pp. 120–131, 2009.

[8] J. P. Papa, A. X. Falcão, V. H. C. Albuquerque, and J. M. R. S. Tavares, "Efficient supervised optimum-path forest classification for large datasets," *Pattern Recognition*, vol. 45, no. 1, pp. 512–520, 2012.

[9] L. C. Afonso, J. P. Papa, L. P. Papa, A. N. Marana, and A. R. Rocha, "Automatic visual dictionary generation through optimum-path forest clustering," in *19th IEEE International Conference on Image Processing*, 2012, pp. 1897–1900.

[10] G. H. Rosa, K. A. P. Costa, L. A. Passos Junior, J. P. Papa, A. X. Falcão, and J. M. R. S. Tavares, "On the training of artificial neural networks with radial basis function using optimum-path forest clustering," in *22nd International Conference on Pattern Recognition*, 2014, pp. 1472–1477.

[11] K. A. P. Costa, L. A. M. Pereira, R. Y. M. Nakamura, C. R. Pereira, J. P. Papa, and A. X. Falcão, "A nature-inspired approach to speed up optimum-path forest clustering and its application to intrusion detection in computer networks," *Information Sciences*, pp. 95–108, 2015, innovative Applications of Artificial Neural Networks in Engineering.

[12] C. Castelo-Fernández and G. Calderón-Ruiz, "Automatic video summarization using the optimum-path forest unsupervised classifier," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, ser. Lecture Notes in Computer Science, A. Pardo and J. Kittler, Eds. Springer International Publishing, 2015, vol. 9423, pp. 760–767.

[13] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, Aug 2000.

[14] J. P. Papa, C. T. N. S., and A. X. Falcão, *LibOPF: A library for the design of optimum-path forest classifiers*, 2009, software version 2.0 available at http://www.ic.unicamp.br/∼afalcao/LibOPF.

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[16] R. O. Stehling, M. A. Nascimento, and A. X. Falcão, "A compact and efficient image retrieval approach based on border/interior pixel classification," in *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, ser. CIKM '02. New York, NY, USA: ACM, 2002, pp. 102–109.