Feature Selection Using Differential Evolution for Unsupervised Image Clustering

Matheus Gutoski[1], Manassés Ribeiro[2], Nelson Marcelo Romero Aquino[1],
Leandro Takeshi Hattori[1], André Eugênio Lazzaretti[1],
and Heitor Silvério Lopes[1](✉)

[1] Graduate Program in Electrical and Computer Engineering,
Federal University of Technology – Paraná (UTFPR), Av Sete de Setembro 3165,
Curitiba, PR 80230-901, Brazil
matheusgutoski@gmail.com, hslopes@utfpr.edu.br

[2] Catarinense Federal Institute of Education, Science and Technology–(IFC),
Rod. SC 135 km 125, Videira, SC 89560-000, Brazil

Abstract. Due to the accelerated growth of unlabeled data, unsupervised classification methods have become of great importance, and clustering is one of the main approaches among these methods. However, the performance of any clustering algorithm is highly dependent on the quality of the features used for the task. This work presents a Differential Evolution algorithm for maximizing an unsupervised clustering measure. Results are evaluated using unsupervised clustering metrics, suggesting that the Differential Evolution algorithm can achieve higher scores when compared to other feature selection methods.

Keywords: Differential evolution · Feature selection · Image clustering

1 Introduction

Classification of unlabeled data is an important and challenging task. The importance is mainly due to the huge volume of unlabeled data being created every second by surveillance cameras, Internet searches, and many other sources [1]. Although supervised classification methods have been highly successful, real-world scenarios seldom make unlabeled data available. As a consequence, a great challenge arises for developing robust unsupervised feature extraction techniques that can represent the data in meaningful ways. The problem becomes even more difficult as volume and dimensionality increases, which is a trend in real-world scenarios [2].

When it comes to analyzing unlabeled data, one of the most used techniques is clustering. The goal of a clustering algorithm is to partition the data such that similar instances are grouped together, and distinct instances are well separated. Many clustering algorithms have been proposed over the years, with K-Means [3] being one of the most popular due to its simplicity and performance.

L. Rutkowski et al. (Eds.): ICAISC 2018, LNAI 10841, pp. 376–385, 2018.
https://doi.org/10.1007/978-3-319-91253-0_35

However, the performance of clustering algorithms strongly depends on the quality of the features extracted from the data. A poor feature set usually leads to poor classification results. In image processing, the usual procedure for classification is to extract meaningful features from images, creating a more informative representation than the raw array of pixels. Image feature extractors can be handcrafted, such as Histogram of Oriented Gradients (HOG) [4], or learned, such as in Convolutional Neural Networks (CNNs) [5]. Despite their great success, CNNs fall under the supervised learning category. In supervised learning, ground truth information guides the network into learning features that are discriminative between each of the classes. However, this is not the case in handcrafted feature extractors. Image features obtained in unsupervised ways may not be discriminative enough to correctly classify images in complex scenarios. Furthermore, some of the features may be irrelevant to the classification process.

Eliminating irrelevant features is necessary to improve the performance of classifiers. This role is fulfilled by feature selection methods. Moreover, selecting a subset of important features can speed up the classification process, reduce computational complexity [1] and improve the classification performance [6].

Many methods have been developed for performing the feature selection task. In [7], a Fast Clustering-Based Feature Subset Selection Algorithm (FAST) was used to select the most relevant features from data using class label information. An unsupervised feature selection method was introduced by [8], where feature saliency is introduced and estimated by an expectation-maximization algorithm. [9] also proposed another unsupervised feature selection method that combines cluster analysis and sparse structural analysis. Bio-inspired feature selection methods have also been explored in the literature. In [10], feature selection is performed using Ant Colony Optimization. In [11], a combination of Genetic Algorithms and Particle Swarm Optimization achieve great results on a supervised feature selection task. [12] performs feature selection and classification simultaneously using a Genetic Programming approach, reaching interesting results.

Inspired by the potential of bio-inspired algorithms, we propose a Differential Evolution (DE) approach for selecting a subset of features in an unsupervised way. The objective function of the algorithm is an unsupervised clustering measure.

Clustering results are evaluated using unsupervised clustering measures. In a fully unsupervised scenario, supervised measures cannot be computed, since they require human-made annotations. Contrariwise, unsupervised measures analyze cluster results by, for instance, computing cluster density and distance between centroids. We show that the proposed DE approach outperforms other methods in regard to unsupervised measures.

The contributions of this paper are: (i) a Differential Evolution algorithm for selecting features from a large dimensional space with an unsupervised clustering measure as the objective function; (ii) a comparison between four feature selection and dimensionality reduction techniques applied to an unsupervised image classification problem; (iii) a semantic analysis of the cluster meaning after classification.

The paper is organized as follows: Sect. 2 describes the theoretical aspects of the methods employed in this work, Sect. 3 presents the method, Sect. 4 presents the experimental results and analysis, and Sect. 5 provides the final remarks along with a discussion and future works.

2 Theoretical Aspects

2.1 Feature Selection

Feature selection fulfills the role of eliminating irrelevant features from data. This process leads to a smaller dimensionality and in some cases can improve the accuracy of a classifier by eliminating noisy features that could be misleading [6].

Feature selection methods are traditionally divided into four groups: filter, wrapper, embedded, and hybrid methods [7]. Filter methods provide a feature ranking by means of mathematical analysis of the data, without the need to build a classifier [8], making them computationally efficient. Wrapper methods select feature subsets based on the results of a classifier. Such methods can be computationally expensive, as they require a classification process for each evaluation. Embedded methods combine feature selection and classification in a single process, such as Neural networks [7]. Hybrid methods combine filter and wrapper methods by doing a two-step feature selection, where the filter is applied first, and the wrapper method comes in sequence [7].

The main feature selection approach used in this work is a wrapper method. DE is a meta-heuristic from the field of Evolutionary Computation that follows the Darwinian evolution principles. The evolution process is guided by an objective function. In this case, we employ the Calinski-Harabasz (CH) coefficient, which is an unsupervised clustering measure, as the function to be maximized.

2.2 Differential Evolution

Differential Evolution (DE) is an evolutionary computation method introduced by Storn and Price [13]. The algorithm performs global optimization by minimizing complex functions such as nonlinear and non-differentiable functions [13]. The algorithm became popular due to its simplicity. It is currently widely used in diverse areas [14,15].

Similar to other evolutionary algorithms, DE evolves a population of possible solutions (vectors) using genetic operations such as crossover and mutation, along with selection techniques to choose the best solutions that will generate new solutions (the next generation). The mutation and crossover rates are parameters that have to be set in order to run the algorithm. Other parameters include population size, number of generations and selection method. The main difference between DE and other evolutionary algorithms is the way DE creates a new population. It is based on using the scaled differences of randomly selected vectors of the current population [14].

According to [16], the DE algorithm is usually described using the form DE/x/y/z, where x describes the differential mutation base, y describes the number of vector differences added to the base vector and z describes the crossover method. The most common definition of the DE algorithm is the (DE/rand/1/bin), which describes random mutation, single vectorial difference and binomial crossover [17]. Different methods for the crossover and mutation operations have been discussed in [16].

Algorithm 1 presents the pseudo-code of the DE algorithm [17]. The inputs to the algorithms are: NP= number of individuals of the population; CR= crossover probability; F= weighting factor; NV= number of variables (length of the vectors).

Algorithm 1. Pseudo-code of the DE algorithm.

```
function DE($NP, CR, F, NV$);
Generate randomly the initial population ($NP$ individuals);
$x \leftarrow random(NP, NV)$;
Compute the fitness for all individuals of the population;
$fitness_x \leftarrow f(x)$;
while stopping criterion=FALSE do
    for $i = 1$ to $NP$ do
        $v_i^{G+1} \leftarrow$ mutation($x_i^G, F$);
        $u_i^{G+1} \leftarrow$ crossover($x_i^G, v_i^{G+1}, CR$);
    $fitness_u \leftarrow f(u)$;
    for $i = 1$ to $NP$ do
        if $fitness_u(i) > fitness_x(i)$ then
            $x_i^{G+1} \leftarrow u_i^{G+1}$;
        else
            $x_i^{G+1} \leftarrow x_i^G$;
    Update stopping criterion;
```

2.3 Unsupervised Cluster Evaluation Metrics

Unsupervised metrics evaluate clustering based on internal information. One of those metrics is the Calinski-Harabasz score [18], which is defined by the ratio of between and within cluster dispersions. The index is defined by Eq. 1, where k is the number of clusters, B_k is the between cluster dispersion, W_k is the within cluster dispersion, and n is the number of points in the data.

$$CH(k) = \frac{B_k}{W_k} \times \frac{n-k}{k-1}. \tag{1}$$

The between cluster dispersion matrix can be calculated as shown in Eq. 2:

$$B_k = \sum_{i=1}^{k} n_i \left\| \mathbf{c_i} - \mu \right\|^2, \tag{2}$$

where k is the number of groups, n_i is the number of observations in group i, $\mathbf{c_i}$ is the center of cluster i, μ is the data mean, and $\|\mathbf{c_i} - \mu\|$ is the euclidean norm. The within cluster dispersion matrix can be calculated as shown in Eq. 3:

$$W_k = \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} \left\| \mathbf{x} - \mathbf{c_i} \right\|^2 , \tag{3}$$

where k is the number of groups, $\mathbf{x}$ is a point, C_i is the cluster i and $\mathbf{c_i}$ is the center of cluster i. The Calinski-Harabasz index produces higher values when clusters are well defined, hence maximizing it is desirable.

The Silhouette score another metric used in this work. The index calculates the *fitness* of each data point to its cluster assignment. The index ranges from -1 to $+1$, where higher values represent more compact cluster assignments. The final score is defined by the silhouette mean of all data samples. The score can be calculated for individual data points following Eq. 4:

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}, \tag{4}$$

where a_i is the mean of samples assigned to the same group, and b_i is the mean distance between the sample i and every other point in the nearest cluster.

3 Method

3.1 Overview

The proposed method is illustrated in Fig. 1. First, an input image is converted to grayscale and resized. Next, features are extracted from the images using the HOG algorithm. DE is then applied in order to select the best features according to the CH clustering measure. Next, clustering is done by using the K-Means algorithm. We evaluate the feature selection methods using two unsupervised clustering metrics: CH and Silhouette. Moreover, for the sake of comparison, we present results using two other feature selection methods: Highest Variance (VAR) and the well known Principal Component Analysis (PCA). Whilst the former is not a feature selection method, it is often used as a dimensionality reduction technique. We also present the clustering results using the Full Data, which is the original HOG feature vector. Implementation was performed using the scikit-learn and Inspyred python packages.

3.2 Preprocessing

Before the feature extraction process, the image data was reshaped to 64×128 pixels in order to allow the HOG feature extractor to operate with its default parameters. The images were also converted to grayscale.

3.3 Feature Extraction and Selection Setup

Using the default parameters, the HOG feature extractor generates 3780 features per image. In order to evaluate different feature selection methods, each algorithm was applied separately to the HOG data, selecting the top 100 features.

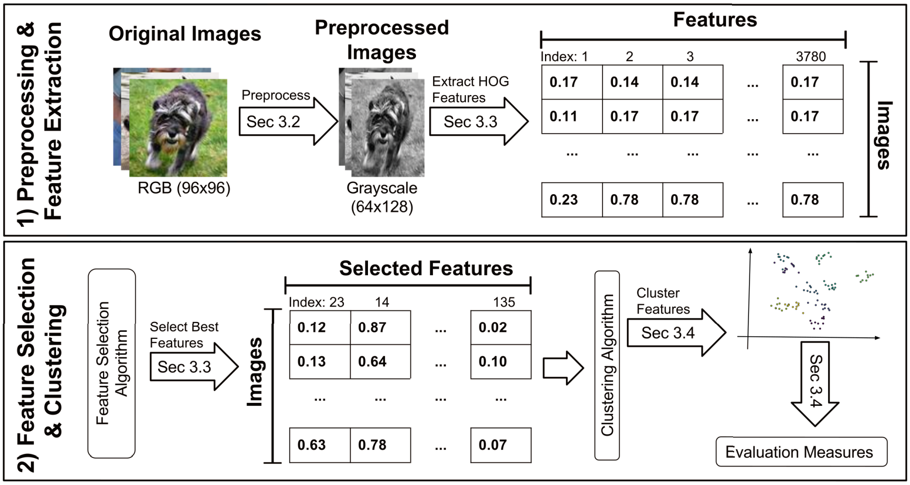

Fig. 1. Overview of the proposed method.

The only parameters required by the PCA and Variance filters are the n number of components to select.

The individuals of the DE algorithm are initialized with a random vector of length 100. Floating point values range from 1 to 3780, matching the length of the HOG feature vector. The lower and upper limits are ensured by the Bounding Function 5.

$$x = \begin{cases} 1 & \text{if } x < 1 \\ 3780 & \text{if } x > 3780 \\ x & \text{otherwise} \end{cases} \tag{5}$$

The decoding process takes the integer part of each value, where each element selects the index of a feature in the original HOG vector. Individuals are penalized in case of re-occurrence of any index by reducing their fitness by 10% per repetition.

The objective function maximizes the CH index (see Eq. 1). The DE (DE/rand/1/bin) parameters were set as follows: population size: 30; heuristic crossover rate: 0.75; Gaussian mutation rate: 0.01; generations: 2,000; Elitism: 0; selection by tournament with 2 candidates and 1 winner. The result reported is the best of 5 runs with different random seeds.

3.4 Clustering and Evaluation

Clustering is done with K-Means algorithm using the K-Means++ initialization method. The algorithm runs 10 times with different random seeds. The best run is then selected based on the least sum of squared distances from each point to its centroid. Evaluation is done using the Silhouette and CH coefficients.

4 Experimental Results and Analysis

4.1 Dataset Description

The STL-10[1] dataset contains 96×96 pixels color image data divided into 10 classes: airplane, bird, car, cat, deer, dog, horse, monkey, ship, truck. The dataset was developed for unsupervised and weakly supervised learning algorithms, and is divided as follows: unsupervised train: 100, 000 unlabeled instances containing data from all 10 classes and some variations; supervised train: contains 500 labeled images per class; test: 800 images per class. In this work, we employ only the Test set, since our method does not include a training phase.

4.2 Evaluation Measures

We perform the evaluation using the measures discussed in Sect. 2.3. Since the clustering algorithms require the number of groups parameters to be set, we propose two experiments. First, we cluster the data into 10 groups, since it is the real number of classes in the STL-10 dataset. Next, we propose a two groups experiment to approach the problem without using any ground truth information.

Experiment #1: 10 Groups. This experiment aims at evaluating the feature selection and clustering methods in a scenario where the number of clusters is set according to a prior ground truth information. The absolute classification performance is not the concern of this experiment. Instead, the relative performance of the different methods was evaluated.

In this experiment, we evolve the DE algorithm and cluster the data in 10 groups, i.e $K = 10$. Table 1 shows the results for this experiment.

Analyzing the results, a few conclusions can be drawn. Regarding the unsupervised measures, DE shows superior results compared to the other methods. The Silhouette score is also slightly higher, however, it shows that clusters are completely overlapped in the feature space.

PCA and *Full Data* presented similar results, as PCA captures most of the variance in 100 features. It also shows that a much smaller subset of features can produce the same results, with fewer computational endeavor. The variance filter performs worse than the other feature selection methods with respect to the CH coefficient, but has slightly better performance than PCA and Full Data regarding the Silhouette coefficient.

Experiment #2: 2 Groups. In this experiment, we also seek to explore the relative performances of the different feature selection methods. However, unlike the previous experiment, we assume no prior knowledge regarding the class distribution.

[1] http://cs.stanford.edu/~acoates/stl10/.

Without using any a priori information, we perform this experiment using $K = 2$ as a parameter for K-Means. Results are displayed in Table 1.

Results show that the algorithms form more compact and distant clusters, as shown by the CH and Silhouette scores. As before, the DE algorithm outperforms the other methods. Full Data and PCA presented very similar results, and the VAR filter was outperformed by all methods.

Despite the ground truth containing 10 classes, in a scenario where the user must define the number of groups without any previous knowledge, the 2 groups solution would be preferable over the 10 groups solution. Addressing this, we analyze the contents of each cluster in the two-groups solution searching for semantic patterns in each group. The analysis is found in the following Section.

Table 1. Evaluation of data clustering using K-Means with K $= 10, 2$

	K = 10				K = 2			
Metric	Full data	DE	Variance	PCA	Full data	DE	Variance	PCA
Calinski-Harabasz	188.30	483.59	143.35	188.08	992.21	2,164.37	724.65	992.11
Silhouette	0.007	0.04	0.01	0.006	0.13	0.23	0.12	0.13

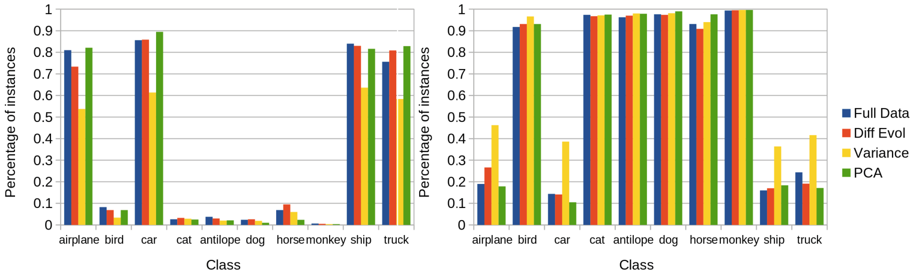

Fig. 2. Histogram of class occurrence per cluster. Cluster 1 (left), Cluster 2 (right)

4.3 Cluster Analysis

In order to better explore the 2 clusters solution, we present a histogram of class occurrences per cluster. Each cluster histogram counts every data point belonging to a class in ground truth level. Figure 2 shows the histograms of clusters 1 and 2. The x-axis shows each class at ground-truth level. The y-axis shows the percentage of instances belonging to a given class in the current cluster.

The chart shows a clear indication that HOG features form clusters with semantic meaning. Cluster 1 (left) shows that vehicles such as airplanes, cars, ships, and trucks were clustered together, whilst in cluster 2 (right) there is a clear predominance of animal classes such as birds, cats, antelopes, dogs, horses and monkeys.

Regarding the feature selection methods, there is no noticeable difference between DE, PCA and Full Data. The VAR filter, however, appears not to be as consistent regarding the animal/vehicle separation observed in other methods.

5 Conclusion

This paper presented a DE approach for performing feature selection in an unsupervised clustering task. The effectiveness of the DE algorithm was compared to other two simple feature selection methods. The evaluation was done by using two unsupervised clustering metrics. A qualitative analysis showing the contents of each cluster was also presented.

Results have shown that the DE feature selection outperformed other methods in terms of unsupervised metrics. Moreover, we have shown that HOG features are sufficiently discriminative between animal and vehicle classes on the STL-10 dataset.

Whilst clustering can be used as a classification method and evaluated as such, it is also true that being an unsupervised method, there is no single correct solution to the classification problem. For a clustering result to match the desired classification, it is necessary that the features are discriminative enough to differ between the target classes. Hence, future works should aim at developing methods for extracting semantic features from images.

Unlabeled image clustering is a difficult task. Separating data into groups with meaningful semantic information in an unsupervised way has proven to be one of the biggest challenges in the fields of Machine Learning, Deep Learning, and Computer Vision.

Acknowledgements. Author M. Gutoski and L.T. Hattori would like to thank CAPES for the scholarship; Author M. Ribeiro would like to thank the Catarinense Federal Institute of Education, Science and Technology and IFC/CAPES/Prodoutoral for the scholarship; Author N. Aquino would like to thank the Organization of the American States, the Coimbra Group of Brazilian Universities and the Pan American Health Organization; author H. S. Lopes would like to thank to CNPq for the research grant number 440977/2015-0.

References

1. Jain, A.K.: Data clustering: 50 years beyond K-means. Pattern Recogn. Lett. **31**(8), 651–666 (2010)
2. Gantz, J.F.: The diverse and exploding digital universe: an updated forecast of worldwide information growth through 2011. Technical report, IDC (2008)
3. Lloyd, S.P.: Least squares quantization in PCM. IEEE Trans. Inf. Theory **28**(2), 129–137 (1982)
4. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Piscataway, NJ, vol. 1, pp. 886–893. IEEE Press (2005)

5. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems, (USA), vol. 1, pp. 1097–1105. Curran Associates Inc. (2012)
6. Chandrashekar, G., Sahin, F.: A survey on feature selection methods. Comput. Electr. Eng. **40**(1), 16–28 (2014)
7. Song, Q., Ni, J., Wang, G.: A fast clustering-based feature subset selection algorithm for high-dimensional data. IEEE Trans. Knowl. Data Eng. **25**(1), 1–14 (2013)
8. Law, M.H., Figueiredo, M.A., Jain, A.K.: Simultaneous feature selection and clustering using mixture models. IEEE Trans. Pattern Anal. Mach. Intell. **26**(9), 1154–1166 (2004)
9. Li, Z., Liu, J., Yang, Y., Zhou, X., Lu, H.: Clustering-guided sparse structural learning for unsupervised feature selection. IEEE Trans. Knowl. Data Eng. **26**(9), 2138–2150 (2014)
10. Tabakhi, S., Moradi, P., Akhlaghian, F.: An unsupervised feature selection algorithm based on ant colony optimization. Eng. Appl. Artif. Intell. **32**(1), 112–123 (2014)
11. Ghamisi, P., Benediktsson, J.A.: Feature selection based on hybridization of genetic algorithm and particle swarm optimization. IEEE Geosci. Remote Sens. Lett. **12**(2), 309–313 (2015)
12. Nag, K., Pal, N.R.: A multi-objective genetic programming-based ensemble for simultaneous feature selection and classification. IEEE Trans. Cybern. **46**(2), 499–510 (2016)
13. Storn, R., Price, K.: Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. J. Glob. Optim. **11**(4), 341–359 (1997)
14. Das, S., Suganthan, P.N.: Differential evolution: a survey of the state-of-the-art. IEEE Trans. Evol. Comput. **15**(1), 4–31 (2011)
15. Hattori, L.T., Lopes, H.S., Lopes, F.M.: Evolutionary computation and swarm intelligence for the inference of gene regulatory networks. Int. J. Innov. Comput. Appl. **7**(4), 225–235 (2016)
16. Lin, C., Qing, A., Feng, Q.: A comparative study of crossover in differential evolution. J. Heuristics **17**(6), 675–703 (2011)
17. Krause, J., Lopes, H.S.: A comparison of differential evolution algorithm with binary and continuous encoding for the MKP. In: Proceedings of the BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence, pp. 381–387 (2013)
18. Caliński, T., Harabasz, J.: A dendrite method for cluster analysis. Commun. Stat.-Theory Methods **3**(1), 1–27 (1974)