Point Symmetry-based deep clustering

Jose G. Moreno
University of Toulouse & IRIT, UMR 5505 CNRS
118 route de Narbonne, F-31062 Toulouse cedex 9, France
jose.moreno@irit.fr

ABSTRACT

Clustering is a central task in unsupervised learning. Recent advances that perform clustering into learned deep features (such as DEC[14], IDEC[6] or VaDe[10]) have showed improvements over classical algorithms, but most of them are based on the Euclidean distance. Moreover, symmetry-based distances have showed to be a powerful tool to distinguish symmetric shapes –such as circles, ellipses, squares, etc. This paper presents an adaptation of symmetry-based distances into deep clustering algorithms, named SymDEC. Our results show that the proposed strategy outperforms significantly the existing Euclidean-based deep clustering as well as recent symmetry-based algorithms in several of the synthetic symmetric and UCI studied datasets.

CCS CONCEPTS

• **Computing methodologies → Cluster analysis**;

KEYWORDS

Deep clustering; symmetry-based distances; unsupervised learning

ACM Reference Format:
Jose G. Moreno. 2018. Point Symmetry-based deep clustering. In *The 27th ACM International Conference on Information and Knowledge Management (CIKM '18), October 22–26, 2018, Torino, Italy.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3269206.3269328

1 INTRODUCTION

Clustering elements into meaningful groups is a challenging task in data mining and pattern recognition. Proposed solutions could be classified into hard or soft clustering [9]. In the former elements belong to a unique cluster. The latter allow elements to belong to multiple clusters. Several algorithms have been developed using the classical –but strong– k-means algorithm, including adaptations to soft clustering [3].

New techniques based on deep clustering have been recently proposed [6–8, 10, 14]. They are based on neural networks (NN) to project data into lower-dimension representations to later easily cluster them. The dimension-reduction step is performed using an autoencoder technique that consist in defining a NN architecture symmetric to the reduced (or embedded) space. The clustering step is performed for an extra clustering layer that group elements by minimizing intra-cluster similarity or maximizing extra-cluster similarity, or both. Despite the use of complex NN, most of the algorithms group near points in terms of the Euclidean distance between the element to cluster and a cluster representative.

In parallel, adaptations of the k-means algorithm using alternatives to the Euclidean distance, such as point symmetry-based distance [13] or line symmetry-based distance [12], have showed outperform the original k-means algorithm. These new distances detect points that not only are close to the centroid, but also close to a symmetric point w.r.t. the centroid. This particularity allows the recognition of symmetric shapes such as rings, circles, squares, etc. However, these methods require specific optimizations which use genetic algorithms [1] or extra parameters [4] (like kernel functions and its configuration) to achieve state-of-the-art results.

In this paper, we studied and present preliminary results on the integration of symmetry-based distances into deep clustering models. Our aim is twofold: (1) to propose and evaluate a symmetry-based clustering algorithm based on NN and composed of only one hidden layer without autoencoding; and (2) to evaluate the proposed algorithm into a deep learning architecture and its effectiveness when using symmetry-based distances in clustering.

CIKM '18, October 22–26, 2018, Torino, Italy

ACM ISBN 978-1-4503-6014-2/18/10…$15.00
https://doi.org/10.1145/3269206.3269328

2 BACKGROUND AND RELATED WORK

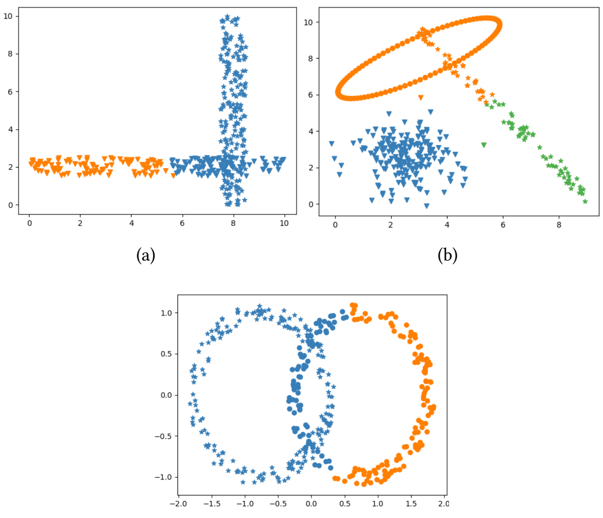
The Euclidean distance is one of the most used distances in traditional algorithms for clustering [9]. For example, k-means is a two steps algorithm that heuristically minimize the loss function in Equation 1.

$$L = \sum_{j=1}^{K} \sum_{z_i \in \pi_j} \|z_i - \mu_j\|^2 \tag{1}$$

where K is the number of desired clusters, $Z = \{z_1, ..., z_n\}$ is the collection of elements to cluster, and μ_j is the centroid of the partition π_j. Depending of the constraints over the partition set ($\Pi = \{\pi_1, ..., \pi_K\}$) a clustering algorithm could be considered as hard or soft clustering. In the case where $\pi_j \cap \pi_l = \emptyset, \forall j, l = \{1, ..., K\}_{j \neq l}$ the clustering algorithm is considered as hard, otherwise as soft clustering. Main advantage of soft clustering is that the same element could belong to several cluster with a membership degree. However, most traditional algorithms, including the classical k-means, are hard clustering solutions.

2.1 Symmetry-based distance

Despite k-means is widely used, it is not well adapted to recognize symmetric shapes. Consider a set of elements separated into two rings as pictured in Figure 1(c). In this case, k-means partitions based on Euclidean distances fail to correctly separate the two rings. A similar behaviour is observed with other symmetric shapes as pictured in Figure 1(a) and 1(b). To overcome this problem, several symmetric-based distances and its integration into clustering algorithms have been proposed [1, 2, 13]. Contrary to the Euclidean distance, the point symmetry-based distance (PSBD) is determined



(a) (b) (c)

Figure 1: Clustering results of k-means algorithm over synthetic symmetric datasets. Colours represent predicted class and markers true class.

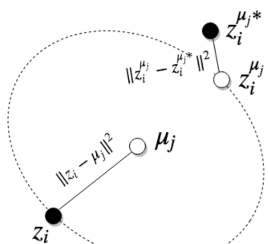


Figure 2: Euclidean distance and point symmetry-based distance between z_i and μ_j.

by the spatial distribution of other points in the collection. Consider Figure 2 where black dots denote existing elements in a collection and white dots denote calculated elements. The PSBD between z_i and μ_j is defined by the distance between the symmetric point to z_i w.r.t. μ_j, i.e. $z_i^{\mu_j}$, and its nearest point, i.e. $z_i^{\mu_j *}$. While classical clustering algorithms minimize $\|z_i - \mu_j\|^2$, PSBD algorithms minimize $\|z_i^{\mu_j} - z_i^{\mu_j *}\|^2$ to help algorithms to find symmetrical shapes. A more formal definition of PSBD is presented in Equations 2 and 3.

$$d_{psb}(z_i, \mu_j) = \|z_i^{\mu_j} - z_i^{\mu_j *}\|^2 \tag{2}$$

where

$$z_i^{\mu_j} = 2 * \mu_j - z_i \quad \textit{and} \quad z_i^{\mu_j *} = \underset{z_l \in Z, z_l \neq z_i}{\arg\min} \|z_l - z_i^{\mu_j}\|^2 \tag{3}$$

Many other PSBDs can be defined in terms of the above factors. Indeed, [1] uses k-nearest points to $z_i^{\mu_j}$ instead of just one and [13] normalize the sum between the Euclidean distance and PSBD. In this work we use the PSBD defined in Equation 2 for the sake of simplicity.

2.2 Deep clustering

The use of deep neural network techniques into clustering is known as deep embedding clustering or DEC [14]. Deep clustering algorithms simultaneously learn a reduced space and cluster elements in the reduce space. These algorithms can be divided into two steps: (1) autoencoding the input data and (2) clustering elements using the Kullback-Leibler (KL) divergence in the encoded space. Autoencoders are symmetric NNs w.r.t a central layer where the input is encoded. They are trained by using the same information as input and output of the NN. The sub-NN between the input and the central layer of an autoencoder is called the encoder and its counterpart is called the decoder. During the training of the second step of DEC, elements can be misplaced generating a downgrading of performances by the optimization algorithm that updates the weights of the NN. IDEC [6] was proposed to overcome this problem. It is an improved version of the original DEC algorithm which use the entire autoencoder instead of only the encoder component. This allow the clustering of elements in the encoded space without deformations in the encoder. IMSAT [8] is another method for discrete representation learning using deep neural networks. As DEC, it is based on the optimization of the KL divergence between a uniform distribution and the elements-clusters distribution. In [7], a more complex combination in the autoencoder is proposed in order to include convolutional features. However, most of the existing algorithms focus on the autoencoding step and mislead the importance of the characteristics in the embedded space.

3 A POINT SYMMETRY-BASED DEEP CLUSTERING ALGORITHM

We based our deep clustering architecture on the work proposed by [14]. After training the autoencoder, we only pick the trained encoder and added an extra layer to perform clustering. Two versions of PSBD layers are proposed in this paper, the hard and soft SymDEC.

3.1 Hard SymDEC

In this case, we directly minimize the loss function defined in Equation 4 within the clustering layer.

$$L = \sum_{z_i \in Z} \max\left(\min_{j \in \{1,\ldots,K\}} \|z_i^{\mu_j} - z_i^{\mu_j *}\|^2, 0\right) \tag{4}$$

Note that μ_j is represented by the interconnection weights between clustering layer and previous layer. In this configuration, our model only have one output (the numeric calculation of the loss) and no labels are needed[1]. After optimization, cluster groups are defined using the PSBD for each element in the collection.

3.2 Soft SymDEC

Similarly that [14], we follow the Student's t-distribution strategy proposed by [11]. In this case, the output of the clustering layer (p_{ij}) is the cluster degree of pertinence to each cluster. We minimize

[1]Labels are not used in clustering, but unsupervised labels may be used to find the better fit between a real distribution and an expected distribution.

the loss function defined in Equation 5 within the clustering layer.

$$L = KL(P\|Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{5}$$

where

$$q_{ij} = \frac{(1 + \|z_i^{\mu_j} - z_i^{\mu_j *}\|^2/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_{j'}(1 + \|z_i^{\mu_{j'}} - z_i^{\mu_{j'} *}\|^2/\alpha)^{-\frac{\alpha+1}{2}}}, \tag{6}$$

α is the degree of freedom of the Student's t-distribution[2]. In this case, labels are the auxiliary target distribution and current distribution is minimized using the KL divergence. Note that no extra PSBD calculations are needed. However, cluster are defined by calculating the argmax of the clustering layer output for each point.

4 EXPERIMENTS

First experiments are conducted in order to evaluate clustering capabilities of our proposal when using symmetric synthetic datasets. In this case, inputs are directly plugged to the clustering layer without autoencoding. However, further experiments including autoencoding are presented in Section 4.4.

4.1 Technical considerations

We used a public implementation of DEC in Keras[3] and modified it to implement our proposed algorithms. Parameters were set following recommendations in [7, 14]. For our algorithms, parameters were set similarly as for the DEC algorithm in [7]. Accuracy (acc.), normalize mutual information (nmi) and adjusted rand index (ari) were used as evaluation metrics. More details about the used datasets can be found in [13] and [4]. K-means results are performed following the best practices in sckitlearn software[4]. Deep clustering initializations are performed using the clusters of the k-means algorithm. All autoencoders are pre-trained before plugging it into its respective deep clustering architecture.

4.2 Synthetic symmetric datasets

We first experiment the performance of our algorithm in terms of symmetric shape identification. For this we used three classical datasets previously proposed by [13] and pictured in Figure 1. Each dataset is composed by 400 elements and true classes are indicated by different point markers in each figure. Average ari performances are presented in Table 1. Both versions of our SymDEC algorithm perform well when compared against Euclidean-based and PSBD-based algorithms. Note that soft SymDEC outperforms the hard version nevertheless the dataset. Further experiments will be conducted only using soft SymDEC. Results of soft SymDEC for these synthetic datasets are pictured in Figure 3.

4.3 UCI datasets

Ten UCI datasets were used to evaluate the impact of PSBD into classical datasets. These datasets variate from 2 to 30 clusters, from 150 to 1484 elements, and from 4 to 30 dimensions [5]. Similar than previous experiments, the fact that the number of dimensions is low make them less suitable to use along with an autoencoder, but they are useful to grasp the SymDEC performances in real data without dimensionality reduction. Results for each dataset are presented in Table 3. Six out of ten datasets show that SymDEC outperforms the k-means algorithm, a classical but strong baseline. For three of the datasets (Iris, Seeds, and Transfusion), SymDEC outperformed the *best of 11* row, which correspond to the best result of eleven different strong algorithms and configurations tested in [4]. For other three datasets (Ecoli, Leaf, and Vertebral) our algorithm fairly approximates the best result. Note that an outstanding result is obtained for the Iris dataset. In this case, our algorithm achieves 95%-99% of performance in the three used metrics which is –up to

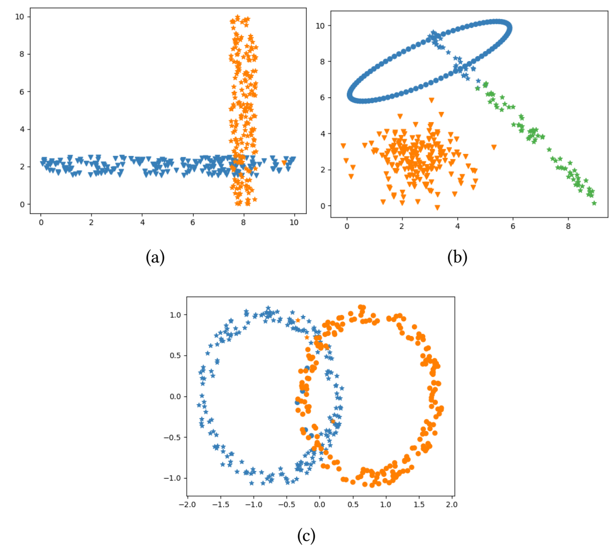


Figure 3: Clustering results of soft SymDEC algorithm over synthetic symmetric datasets. Colors represent predicted class and markers true class.

Table 1: Average ari performances for the tree synthetic datasets from Figure 1.

	Data (a)	Data (b)	Data (c)
k-means	0.26	0.80	0.37
SBKM best [4]	0.75	0.85	0.65
K-SBKM best [4]	0.60	0.80	0.77
Hard SymDEC	0.53	**0.86**	0.82
Soft SymDEC	**0.99**	**0.86**	**0.91**

Table 2: Average performances for the reuters10k dataset.

	acc.	nmi	ari
DEC (code)	0.73±0.03	0.53±0.04	0.57±0.04
IDEC (code)	0.72±0.03	0.51±0.03	0.55±0.03
IMSAT[8]	0.71±0.05	-	-
DEC [14]	0.72	-	-
IDEC [6]	**0.76**	0.50	-
Soft SymDEC	**0.76±0.03**	**0.54±0.03**	**0.61±0.03**

[2] α value is set to 1.
[3] https://github.com/XifengGuo/DEC-keras, last checked: May 22, 2018.
[4] http://scikit-learn.org), last checked: May 22, 2018.

Table 3: **Performances on terms of accuracy (acc.), normalize mutual information (nmi) and adjusted rand index (ari) for 10 UCI datasets. Note that *best of 11* row correspond to the best result of eleven strong algorithms tested by [4].**

	Breast Cancer			Ecoli			Glass			Iris			Leaf		
	acc.	nmi	ari	acc.	nmi	ari	acc.	nmi	ari	acc.	nmi	ari	acc.	nmi	ari
k-means	**0.91**	**0.55**	0.67	0.64	0.64	0.50	**0.45**	**0.31**	0.17	0.83	0.66	0.62	0.51	0.70	0.35
best of 11 [4]	-	-	**0.69**	-	-	**0.67**	-	-	**0.23**	-	-	0.66	-	-	**0.37**
Soft SymDEC	0.83	0.40	0.44	**0.79**	**0.76**	**0.66**	0.44	0.28	0.15	**0.99**	**0.95**	**0.96**	**0.54**	**0.72**	**0.36**
	Seeds			Transfusion			User			Vertebral			Yeast		
k-means	0.92	0.73	0.77	**0.68**	0.01	0.05	**0.41**	**0.14**	0.09	0.47	0.30	0.21	0.40	**0.30**	0.16
best of 11 [4]	-	-	0.86	-	-	0.06	-	-	**0.20**	-	-	**0.42**	-	-	**0.20**
Soft SymDEC	**0.96**	**0.83**	**0.87**	0.67	**0.07**	**0.11**	0.34	0.04	0.03	**0.59**	**0.53**	0.39	**0.42**	0.25	0.11

our knowledge– the best reported performance of an unsupervised algorithm on this well-known dataset.

4.4 The reuters10k dataset

The reuters10k dataset was used to evaluate the deep clustering capabilities of our proposal. This dataset is composed by 10000 elements with 2000 dimensions and grouped in 4 classes. In this case, the autoencoder is used to reduce the 2000-dimension space to a 10-dimension space. Average performances over 10 runs are presented in Table 2. Rows labelled with *(code)* correspond to results using a public implementation[5] of DEC [14] and IDEC [6] following parameters recommendations from the authors. Other baselines results were took directly from [6, 14]. Our soft SymDEC algorithm outperforms the used baselines on the three evaluated metrics.

5 DISCUSSION

Experimental results show that hard and soft algorithms based on PSBD are useful for both classical and synthetic symmetric datasets. Moreover, soft SymDEC outperformed its hard counterpart in most of the evaluated datasets. However, more experiments are needed to test other symmetric-based distances that have showed to improve the basic PSBD used in this paper [1, 12].

During our experiments, we noted that the update interval to calculate the target distribution of DEC [14] and IDEC [6] has a significant impact on the results. We have used the default parameter proposed by DEC[6]. More experiments may be conducted to better understand this parameter which could deal with an improvement in both, existing methods and SymDEC. Other disadvantage for the SymDEC algorithm in our evaluation setup is the use of k-means for initialization. It seems fairer to use a PSBD k-means initializer as [2] to initialize SymDEC weights in the clustering layer.

Despite it is clear that datasets from Figure 1 are symmetric, it is hard to imagine symmetry in spaces with more than 3 dimensions. It seems appropriate to explore the evaluation of symmetric properties in datasets before applying PSDB algorithms. However, it is promising that our SymDEC algorithm is capable to exploit symmetry in several datasets with more than 3 dimensions. Results on UCI datasets suggest some symmetric properties in nature related datasets (Iris, Seeds, and Transfusion) encouraging us to pursue further experiments with this kind of data.

[5]https://github.com/XifengGuo/DEC-keras, last checked: May 22, 2018.
[6]Update interval equal to 30 iterations.

6 CONCLUSION

In this paper, we present a deep clustering algorithm based on a point symmetry-based distance. Two loss functions, one for hard clustering and other for soft clustering, are defined and optimized using a deep learning framework. Several experiments are conducted using 3 symmetric synthetic, 10 UCI, and the reuters10k datasets. Our results show the usefulness of PSBD into clustering, symmetric clustering, and deep clustering. Results using soft SymDEC over the Iris dataset not only significantly outperform unsupervised solutions but fairly approximate supervised algorithms. Our findings also suggest that some of the UCI datasets and the reuters10k are symmetric friendly datasets.

REFERENCES

[1] Sanghamitra Bandyopadhyay and Sriparna Saha. 2007. GAPS: A Clustering Method Using a New Point Symmetry-based Distance Measure. *Pattern Recogn.* 40, 12 (Dec. 2007), 3430–3451.

[2] Kuo-Liang Chung and Jhin-Sian Lin. 2007. Faster and More Robust Point Symmetry-based K-means Algorithm. *Pattern Recogn.* 40, 2 (Feb. 2007), 410–422.

[3] G. Cleuziou. 2008. An extended version of the k-means method for overlapping clustering. In *2008 19th International Conference on Pattern Recognition*. 1–4.

[4] Guillaume Cleuziou and Jose G. Moreno. 2015. Kernel Methods for Point Symmetry-based Clustering. *Pattern Recogn.* 48, 9 (Sept. 2015), 2812–2830.

[5] Dua Dheeru and Efi Karra Taniskidou. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

[6] Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin. 2017. Improved Deep Embedded Clustering with Local Structure Preservation. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*. 1753–1759.

[7] Xifeng Guo, Xinwang Liu, En Zhu, and Jianping Yin. 2017. Deep Clustering with Convolutional Autoencoders. In *Neural Information Processing*. 373–382.

[8] Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama. 2017. Learning Discrete Representations via Information Maximizing Self-Augmented Training. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, Vol. 70. 1558–1567.

[9] Anil K. Jain and Richard C. Dubes. 1988. *Algorithms for Clustering Data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

[10] Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. 2017. Variational Deep Embedding: An Unsupervised and Generative Approach to Clustering. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*. 1965–1972.

[11] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[12] Sriparna Saha and Sanghamitra Bandyopadhyay. 2007. A genetic clustering technique using a new line symmetry based distance measure. In *Advanced Computing and Communications, 2007. ADCOM 2007. International Conference on.* IEEE, 365–370.

[13] Mu-Chun Su and Chien-Hsing Chou. 2001. A modified version of the K-means algorithm with a distance based on cluster symmetry. *IEEE Transactions on pattern analysis and machine intelligence* 23, 6 (2001), 674–680.

[14] Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised Deep Embedding for Clustering Analysis. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48 (ICML'16)*. 478–487.