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

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.

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.

$$L = \sum_{j=1}^{K} \sum_{z \in \pi_j} \|z - \mu_j\|^2$$ (1)

where $K$ is the number of desired clusters, $Z = \{z_1, ..., z_n\}$ is the collection of elements to cluster, and $\mu_j$ is the centroid of the partition $\pi_j$. 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 $\pi_j \cap \pi_l = 0, \forall j, l = \{1, ..., K\}$ $j \neq l$ 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 more formal definition of PSBD is presented in Equations 2 and 3. Indeed, [1] uses k-nearest points to help algorithms to find symmetrical shapes. 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.

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

where

\[
z_i^{\mu_j} = 2 \cdot \mu_j - z_i \quad \text{and} \quad z_i^{\mu_j^*} = \arg \min_{z_j \in Z, z_j \neq z_i} \| z_i - z_j^{\mu_j} \|^2 
\]

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.

\[
L = \sum_{z_i \in Z} \max_{j \in \{1, \ldots, K\}} \min_{l \in \{1, \ldots, L\}} \| z_i^{\mu_l} - z_i^{\mu_j^*} \|^2, 0 \quad (4)
\]

Note that \( \mu_j \) 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 \( (\rho_{ij}) \) 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.

\[ L = KL(P || Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \]  

(5)

where

\[ q_{ij} = \frac{(1 + \|z_i^{(R)} - z_j^{(R)}\|^2 / \alpha)^{-\frac{\alpha+1}{2}}}{\sum_i (1 + \|z_i^{(R)} - z_i^{(R)}\|^2 / \alpha)^{-\frac{\alpha+1}{2}}} \]  

(6)

\(\alpha\) is the degree of freedom of the Student’s t-distribution. 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 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 scikitlearn software. 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

\(\alpha\) value is set to 1.


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 four synthetic datasets from Figure 1.

<table>
<thead>
<tr>
<th></th>
<th>Data (a)</th>
<th>Data (b)</th>
<th>Data (c)</th>
</tr>
</thead>
<tbody>
<tr>
<td>k-means</td>
<td>0.26</td>
<td>0.80</td>
<td>0.37</td>
</tr>
<tr>
<td>SBKM best [4]</td>
<td>0.75</td>
<td>0.85</td>
<td>0.65</td>
</tr>
<tr>
<td>K-SBKM best [4]</td>
<td>0.60</td>
<td>0.80</td>
<td>0.77</td>
</tr>
<tr>
<td>Hard SymDEC</td>
<td>0.53</td>
<td>0.86</td>
<td>0.82</td>
</tr>
<tr>
<td>Soft SymDEC</td>
<td>0.99</td>
<td>0.86</td>
<td>0.91</td>
</tr>
</tbody>
</table>

Table 2: Average performances for the reuters10k dataset.

<table>
<thead>
<tr>
<th></th>
<th>acc.</th>
<th>nmi</th>
<th>ari</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEC (code)</td>
<td>0.73±0.03</td>
<td>0.53±0.04</td>
<td>0.57±0.04</td>
</tr>
<tr>
<td>IDEC (code)</td>
<td>0.72±0.03</td>
<td>0.51±0.03</td>
<td>0.55±0.03</td>
</tr>
<tr>
<td>IMSAT[8]</td>
<td>0.71±0.05</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DEC [14]</td>
<td>0.72</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>IDEC [6]</td>
<td>0.76</td>
<td>0.50</td>
<td>-</td>
</tr>
<tr>
<td>Soft SymDEC</td>
<td>0.76±0.03</td>
<td>0.54±0.03</td>
<td>0.61±0.03</td>
</tr>
</tbody>
</table>

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
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].

<table>
<thead>
<tr>
<th></th>
<th>Breast Cancer</th>
<th>Ecoli</th>
<th>Glass</th>
<th>Iris</th>
<th>Leaf</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>acc.</td>
<td>nmi</td>
<td>ari</td>
<td>acc.</td>
<td>nmi</td>
</tr>
<tr>
<td>k-means</td>
<td>0.91</td>
<td>0.55</td>
<td>0.67</td>
<td>0.64</td>
<td>0.64</td>
</tr>
<tr>
<td>best of 11 [4]</td>
<td>-</td>
<td>-</td>
<td>0.69</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Soft SymDEC</td>
<td>0.83</td>
<td>0.40</td>
<td>0.44</td>
<td>0.79</td>
<td>0.76</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Seeds</th>
<th>Transfusion</th>
<th>User</th>
<th>Vertebral</th>
<th>Yeast</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>acc.</td>
<td>nmi</td>
<td>ari</td>
<td>acc.</td>
<td>nmi</td>
</tr>
<tr>
<td>k-means</td>
<td>0.92</td>
<td>0.73</td>
<td>0.77</td>
<td>0.68</td>
<td>0.01</td>
</tr>
<tr>
<td>best of 11 [4]</td>
<td>-</td>
<td>-</td>
<td>0.86</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Soft SymDEC</td>
<td>0.96</td>
<td>0.83</td>
<td>0.87</td>
<td>0.67</td>
<td>0.07</td>
</tr>
</tbody>
</table>

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