Discriminative Clustering using Regularized Subspace Learning

Nikolaos Passalis, Anastasios Tefas

Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece
Email: passalis@csd.auth.gr, tefas@csd.auth.gr

Abstract

Clustering is among the most important unsupervised learning tasks with several applications in a wide range of domains. Discriminative Clustering (DC) techniques combine the unsupervised nature of clustering with the high discriminative ability of supervised subspace learning methods by simultaneously performing clustering and learning a representation that encourages the separability of the clusters. However, in contrast with classical supervised learning tasks, where the labels are usually correct, cluster assignments are inherently noisy leading to suboptimal results when used as ground truth information with highly discriminative methods. To this end, a novel similarity-based subspace learning method, that allows for learning regularized clustering-oriented representations, avoiding the pitfalls of highly discriminative methods, such as Linear Discriminant Analysis (LDA), is proposed in this paper. The ability of the proposed method to improve the quality of the obtained clustering solutions is demonstrated using extensive experiments on four datasets.

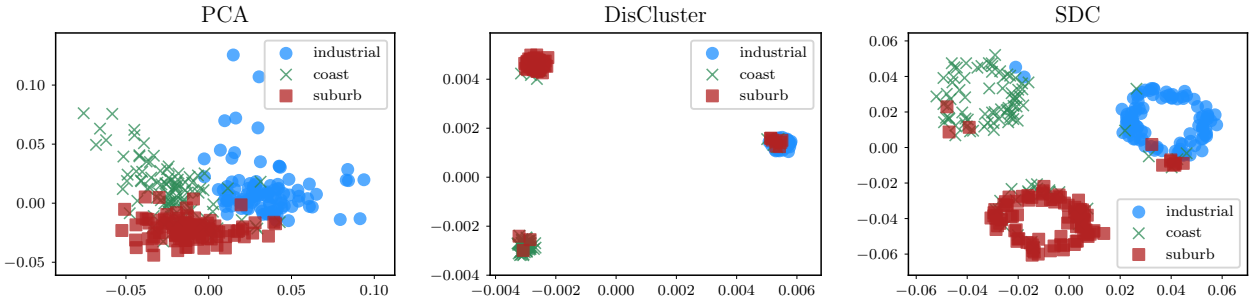
Keywords: Discriminative Clustering, Subspace Learning, Unsupervised Learning

1. Introduction

Clustering is among the most important unsupervised learning tasks with several applications in a wide range of domains, e.g., exploratory data analysis [1], machine learning [2, 3], and information retrieval [4]. The objective of clustering is to partition a set of objects into a number of groups, also known as clusters [5, 6]. Each cluster is expected to contain objects that are similar regarding some of their attributes, while objects from different clusters are expected to be different with respect to some of their properties. That way, it is possible to discover interesting (and possibly previously unknown) patterns in the data [1, 7], or perform a number of machine learning and pattern recognition tasks when no supervised information is available [2, 3]. As a result, several clustering techniques have been proposed, ranging from centroid-based clustering [5, 8], density-based clustering [9, 10], and fuzzy clustering [11, 12, 13], to spectral clustering [6, 14], subspace clustering [15, 16, 17], and discriminative clustering [18, 19, 20].

Discriminative Clustering (DC) combines the unsupervised nature of clustering with the high discrimina-

*Corresponding author
Email address: passalis@csd.auth.gr (Nikolaos Passalis)



(a) Principal Component Analysis (PCA)

(b) Discriminative Clustering (DisCluster)

(c) Proposed Similarity-based Discriminant Clustering (SDC)

Figure 1: Using different techniques to reduce the dimensionality of the data before clustering (three classes of the 15-scene dataset were used [22]). Note that highly discriminative methods, such as DisCluster, over-fits the data, while the proposed regularized SDC method can be used to avoid these issues and provide better clustering solutions.

tive ability of supervised methods by simultaneously performing clustering and learning a representation that encourages the separability of clusters. In other words, clusters are treated as the discovered categories of the data and a representation that separates these categories is learned. Inspired by the extensive literature on discriminative techniques, many DC techniques have been proposed, e.g., [18, 20, 21], improving the quality of clustering solutions. However, in contrast with classical supervised learning tasks, where the supplied labels are usually correct, in DC there is no guarantee that the *discovered* clusters will indeed correspond to the true categories of the data. Therefore, relying on highly discriminative criteria when the labels are inherently noisy can lead to suboptimal results, as it is experimentally demonstrate in Section 4.

The aforementioned problem can be addressed using *regularized* techniques that are able to both increase the discrimination ability of the learned representation, i.e., make the clusters more separable, while avoid collapsing or over-fitting the representation to the supplied noisy labels. This phenomenon is demonstrated in Fig. 1, where 500 data samples of the 15-scene dataset [22], that belong to three different classes ("industial", "coast", and "suburb"), are used. First, PCA is used to reduce the dimensionality and provide a 2-d visualization of the data in Fig. 1a. Then, in the example of Fig. 1b, the well-known DisCluster technique is used [23]. While the separability of the clusters greatly improves, DisCluster severely overfits the representation, since the clusters have almost collapsed into single points. To overcome this problem we seek to learn a regularized representation that can increase cluster separability, while following the geometry of the data. At the same time, the method should not be overly confident on the supplied labels and should account for the *uncertainty* in cluster assignments to avoid over-fitting the representation. An example of such representation, obtained using the technique proposed in this paper, is shown in Fig. 1c. The proposed method avoids over-fitting, while "organizing" the clusters into circular structures that follow the geometry of the data. This leads to more compact intra-cluster regions for data samples that belong to wrong clusters (e.g., "suburb" class in the "coast" cluster' in Fig. 1c).

To better understand the over-fitting phenomena that occur in Fig. 1b one has to consider the optimization

objective of the LDA method [24], which is used by the DisCluster method to learn a projection direction:

$$arg\,min_{\mathbf{w}} J_{LDA} = \arg\ min_{\mathbf{w}} \frac{\mathbf{w}^T \mathbf{S}_W \mathbf{w}}{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}, \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^d$ is the projection vector of LDA, d is the original dimensionality of the data, $\mathbf{S}_W = \sum_{i=1}^{N_C} \sum_{\mathbf{x} \in C_i} (\mathbf{x} - \boldsymbol{\mu}_i)(\mathbf{x} - \boldsymbol{\mu}_i)^T$ is the intra-cluster scatter matrix, N_C is the total number of clusters, C_i is the set of points that belong to the i-th cluster, and $\boldsymbol{\mu}_i$ is the mean vector of the samples that belong to the i-th cluster. Also, $\mathbf{S}_B = \sum_{i=1}^{N_C} |C_i| (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T$ is the inter-cluster scatter matrix, where $\boldsymbol{\mu}$ is the mean vector of all the data samples. Therefore, DisCluster seeks solutions that minimize the intra-cluster distances, while maximizing the inter-clusters distances. In an ideal situation, DisCluster would collapse all the data points into single-point clusters that are as far as possible from each other. Even though such LDA-based techniques usually work well when the supplied labels are noise free [24, 25, 26], the situation is vastly different when a large number of wrongly labeled data exist. In this paper, we argue that, in such cases, the used objective should appropriately account for the wrongly labeled samples and model the uncertainty in cluster assignments to avoid overfitting the representation.

The main contribution of this paper is the proposal of a discriminative clustering method that is able to provide regularized low-dimensional representations that are optimized toward clustering tasks. To achieve this, a similarity-based clustering-oriented discriminative subspace learning technique is proposed in this paper. The intra-cluster and the inter-cluster distances are transformed into similarities and then manipulated in an appropriate way that ensures that the representation will not collapse or over-fit the supplied labels. This also ensures that the proposed method will be less sensitive to the initial clustering solution compared to other discriminative clustering methods, as we also experimentally demonstrate in this paper. To achieve this, the proposed method learns a subspace that maintains a small dissimilarity between points that belong to the same cluster and a small similarity between points that belong to different clusters, encoding the *uncertainty* that inherently exists in the cluster assignments. On top of these, working with a non-linear similarity formulation, instead of the traditional distance-based formulation (e.g., LDA), allows for modeling the distribution of the data using higher-order statistics, accurately following the geometry of the data, and making the proposed technique more robust to outliers. Furthermore, the proposed method can readily scale to large datasets since it supports optimization in batches and it can provide out-of-sample extensions [27], overcoming the limitations of other clustering techniques, such as spectral clustering [14] or many other subspace clustering formulations [16]. The proposed method can be readily combined with most of the existing clustering algorithms, ranging from traditional k-means clustering to spectral clustering techniques, and learn discriminative subspaces that improve the clustering solutions, as it is experimentally demonstrated using extensive experiments on four datasets. A reference implementation of the proposed method will be available, as part of the PySEF library [28], at `https://github.com/passalis/sef` to allow other researchers to easily use and extend the proposed method. All the conducted experiments can be readily reproduced using the supplied code.

The rest of this paper is structured as follows. Related work is discussed in Section 2 and the proposed

method is derived and discussed in detail in Section 3. Then, the experimental evaluation is provided in Section 4. Finally, conclusions are drawn and future work is discussed in Section 5.

2. Related Work

Clustering is a well studied topic with extensive literature available on various clustering methods [9, 14, 29, 30]. Among the simplest, but widely used, clustering methods is the k-means algorithm [31], and its variants [5, 32, 33, 34]. More advanced techniques include density-based clustering [9], spectral clustering [14, 35, 36, 37], and subspace clustering [16, 38, 39, 40]. However, many of these methods, e.g., most spectral methods [14], and subspace methods [16], are unable to efficiently scale to large datasets and handle out-of-sample data [27], i.e., assign unseen data to existing clusters. On the other hand, the proposed method is able to efficiently scale to large datasets and readily provide out-of-sample extensions for the acquired solution, overcoming the limitations of the aforementioned techniques. Furthermore, note that methods that employ the self-representation principle, e.g., [38, 39, 40], also construct the similarity matrix of the data, aiming to represent every sample as a combination of other samples. Instead, the proposed method aims to appropriately manipulate the similarity matrix of the data in order to alter the way that the subspace is formed, instead of using self-representation to build the subspace. Also, quite recently, deep unsupervised clustering approaches have also been proposed [41, 42, 43]. These approaches aim to simultaneously learn deep feature extractors, as well as cluster the data to various clusters. It is worth noting that the proposed approach can be readily combined with most of the existing clustering methods, providing a structured way to learn regularized subspaces and reduce the effect of the cluster and representation collapse phenomena that often occur in many of the aforementioned methods [41].

The proposed method belongs to the family of discriminative clustering methods, e.g., [18, 19, 21]. Early approaches to discriminative clustering followed an LDA-based methodology, where they alternatively optimized a joint clustering and dimensionality reduction objective [21, 23, 44, 45]. A fast discriminative clustering approach was also proposed in [19], where an Extreme Learning Machine (ELM) [46] was used to perform discriminative clustering. These approaches use highly discriminative LDA-based optimization objectives that, as it is demonstrated in Section 4, can leads to suboptimal results. Other discriminative clustering methods also include maximin separation probability clustering [47], and regularized information maximization algorithms [18]. Also, maximum margin clustering formulations also exist [48], inspired by the corresponding supervised margin maximization methods, e.g., [49]. However, these methods are computationally intensive with several variants proposed to reduce their complexity [50, 51, 52]. In contrast to these, the proposed method supports optimization in batches allowing for efficiently scaling to large datasets. Furthermore, the proposed method aims to *learn a representation* that will make clustering easier, instead of directly solving the clustering problem and it can be combined with any clustering algorithm.

A similarity-based dimensionality reduction framework, called Similarity Embedding Framework (SEF), was proposed in [53], where it was demonstrated that similarity-based formulations are more robust to

outliers and model the distribution of the data using higher-order statistics. To the best of our knowledge, we proposed the first iterative similarity-based discriminative clustering method that is able to learn regularized discriminative subspaces optimized toward clustering tasks and significantly improve the clustering accuracy, as it was also experimentally demonstrated in Section 4. The proposed method can efficiently scale to large datasets and it can be combined with any clustering algorithm following the formulation proposed in [54], effectively overcoming the previous limitations of SEF which involved the computation of the whole similarity matrix that limited the ability of the method to scale to larger datasets. Finally, in contrast to many other discriminative clustering approaches that directly predict the cluster assignments, e.g., [19, 48], the proposed method yields a generic improved representation that can be used for other tasks, such as information retrieval tasks [4], or visualization tasks [55].

3. Proposed Method

First, the employed similarity-based subspace learning approach is described. Then, the proposed Similarity-based Discriminative Clustering (SDC) is analytically derived in Subsection 3.2.

3.1. Similarity-based Subspace Learning

Let $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ denote a set of data points. Each data point lies in a d-dimensional space, i.e., $\mathbf{x}_i \in \mathbb{R}^d$. This work aims to learn a projection function $f_{\mathbf{W}} : \mathbb{R}^d \rightarrow \mathbb{R}^m$ that projects the data of $\mathcal{X}$ to a lower dimensional space ($m < d$), where the similarity between each pair of data points is transformed appropriately to meet a predefined target and $\mathbf{W}$ denotes the parameters of the projection function that is used to form the clustering-oriented subspace. To this end, a function that measures the similarity between the data points must be used. Even though several choices exist [53], the Gaussian kernel, also known as Heat kernel, is usually used. Therefore, the similarity matrix $\mathbf{P} \in \mathbb{R}^{N \times N}$ of the projected data $\mathcal{X}$ is defined as:

$$[\mathbf{P}]_{ij} = exp(-||f_{\mathbf{W}}(\mathbf{x}_i) - f_{\mathbf{W}}(\mathbf{x}_j)||_2^2/\sigma_P), \tag{2}$$

where the notation $[\mathbf{P}]_{ij}$ is used to denote the element in the i-th row and the j-th column of the matrix $\mathbf{P}$, $||\mathbf{x}||_2$ is the l^2 norm of a vector $\mathbf{x}$ and σ_P is a scaling factor that acts similarly to the bandwidth of Kernel Density Estimation (KDE) methods [56]. Note that the non-linear behavior of (2) smoothen the effect of (possible) outliers on the learned projection.

The projection function must transform the similarities in the projected space in such way in order to be as close as possible to a predefined target. The target similarity is given by the matrix $\mathbf{T} \in \mathbb{R}^{N \times N}$. In this work, the target matrix $\mathbf{T}$ is appropriately selected to ensure that a discriminative subspace that is optimized toward clustering will be learned (more details are given later in Section 3.2). The squared loss function can be used to ensure that the projection function $f_{\mathbf{W}}(\cdot)$ will exhibit the desired behavior:

$$J_s(\mathcal{X}, \mathbf{W}) = \frac{1}{2||\mathbf{M}||_1} \sum_{i=1}^{N} \sum_{j=1}^{N} [\mathbf{M}]_{ij} ([\mathbf{P}]_{ij} - [\mathbf{T}]_{ij})^2, \tag{3}$$

where $\mathbf{M} \in \mathbb{R}^{N \times N}$ is a weighting mask that defines the importance of achieving the target similarity for each pair of points and $||\mathbf{M}||_1$ denotes the "element-wise" norm of the matrix $\mathbf{M}$, i.e., $||\mathbf{M}||_1 = \sum_{i=0}^{N} \sum_{j=0}^{N} |[\mathbf{M}]_{ij}|$. Note that the values of the weighting mask are confined to the unit interval: $0 \leq [\mathbf{M}]_{ij} \leq 1$. Also, observe that the loss function defined in (3) is minimized when each pair of the projected points achieves its target similarity.

A regularization term J_p, that enforces the orthonormality of the projection vectors, is also employed to ensure that the resulting subspace will not degenerate to trivial solutions, i.e., learning multiple parallel projection directions. Combining this regularization term with the loss function of (3), leads to the final objective used to learn the projection function $f_{\mathbf{W}}(\cdot)$:

$$\arg\min_{\mathbf{W}} J(\mathcal{X}, \mathbf{W}) = (2 - \alpha_p) J_s(\mathcal{X}, \mathbf{W}) + \alpha_p J_p(\mathbf{W}), \tag{4}$$

where the hyper-parameter α_p alters the importance of the orthonormality regularizer ($0 \leq \alpha_p \leq 1$). When α_P is set to 0, then no regularization is used, while when α_P is set to 1 the orthonormality is equally important to the objective function J_s. This loss can be optimized using stochastic gradient descent, given that the projection function is continuous and differentiable:

$$\Delta \mathbf{W} = -\eta \frac{\partial J}{\partial \mathbf{W}}, \tag{5}$$

where η is the learning rate. The Adam algorithm is usually used instead of the standard stochastic gradient descent, since it provides faster and more stable convergence [57]. Also, note that instead of using the whole similarity matrix to perform the optimization, as in [53], only a random subsample of the full similarity matrix is used by selecting N_{batch} random samples and calculating the similarity between them. This allows for significantly speeding up the optimization process, without harming the quality of the learned subspace.

3.2. Similarity-based Discriminative Clustering

The motivation behind the proposed Similarity-based Discriminative Clustering (SDC) is to improve the clustering solutions by learning a projection function that improves the intra-cluster compactness and the inter-cluster separability. The first step toward this process is to obtain an initial clustering solution that we will build upon for improving the learned representation. Even though any clustering algorithm can be used for this task, in this work the standard k-means algorithm is used to ensure that the method will be computationally tractable. Therefore, the initial clustering solution $\mathcal{S}_0$ is obtained by solving the following optimization problem using k-means:

$$\mathcal{S}_0 = arg\,min_{\mathcal{S}} \sum_{i=0}^{N_C} \sum_{\mathbf{x} \in \mathcal{S}^{(i)}} ||\mathbf{x} - \boldsymbol{\mu}_i||_2^2, \tag{6}$$

where $\mathcal{S}^{(i)}$ is the set of data points that belong to the i-th cluster for a clustering solution $\mathcal{S}$, N_C is the number of clusters and $\boldsymbol{\mu}_i$ is the mean vector of the data points that belong to the i-th cluster. Alternatively other clustering algorithms can be also used, as it is demonstrated in Section 4.4.

After obtaining the initial clustering solution, then a projection function that reduces the distances between points of the same cluster, while moving the clusters away from each other must be learned. This can be achieved by defining an LDA-like similarity target:

$$[\mathbf{T}]_{ij} = \begin{cases} 1, \text{if the points } i \text{ and } j \text{ belong to the same cluster} \\ 0, \text{otherwise.} \end{cases}$$

Note that the number of data pairs that belong to different classes is significantly larger than the number of pairs that belong to the same cluster. Therefore, the optimization mask $\mathbf{M}$ must be appropriately set to account for this phenomenon and ensure that minimizing the intra-cluster scatter is equally important to maximizing the inter-cluster scatter:

$$[\mathbf{M}]_{ij} = \begin{cases} 1, \text{if the points } i \text{ and } j \text{ belong to the same cluster} \\ \frac{1}{N_C - 1}, \text{otherwise.} \end{cases} \tag{7}$$

However, using this formulation comes with the drawbacks of LDA-based approaches, since the objective (Eq. (6)) is minimized when the points that belong to the same cluster collapse into a single point and these collapsed points are as far apart as possible. To remedy these, the similarity target is redefined as follows:

$$[\mathbf{T}]_{ij} = \begin{cases} a_{intra}, \text{if the points } i \text{ and } j \text{ belong to the same cluster} \\ a_{inter}, \text{otherwise,} \end{cases} \tag{8}$$

where $a_{intra} < 1$ is a positive number that defines the desired intra-cluster similarity and $a_{inter} > 0$ is a positive number that defines the inter-cluster similarity. Therefore, two points that belong to the same cluster are no longer required to be as similar as possible to each other. Instead, a small distance has to be maintained between them, e.g., by setting $a_{intra} = 0.9$. This can have a strong regularization effect to the learned subspace avoiding overfitting the data. As a side-effect this also effectively "forces" the data to form circular structures around the center of their cluster (according to (8) the intra-cluster samples must be equidistant to each other), while following their actual geometry/manifold (as illustrated in Fig. 1c). On the other hand, the similarity between points that belong to different clusters must no longer be 0, but we account for a small similarity between them, e.g., $a_{inter} = 0.1$. In this way, more regularized solutions that avoid over-fitting the data to the noisy cluster labels can be obtained, as it is also experimentally demonstrated in Section 4.

Also, the projection function f must be defined. In this work, a simple, yet fast and effective, linear function is used:

$$f_{\mathbf{W}}(\mathbf{x}) = \mathbf{W}^T \mathbf{x}, \tag{9}$$

where $\mathbf{W} \in \mathbb{R}^{d \times m}$ is the used projection matrix. Then, the derivative of the objective function (needed for the optimization of the projection function) can be calculated as:

$$\frac{\partial J_s}{\partial [\mathbf{W}]_{kt}} = \frac{1}{||\mathbf{M}||_1} \sum_{i=1}^{N} \sum_{j=1}^{N} [\mathbf{M}]_{ij} ([\mathbf{P}]_{ij} - [\mathbf{T}]_{ij}) \frac{\partial [\mathbf{P}]_{ij}}{\partial [\mathbf{W}]_{kt}}, \tag{10}$$

where

$$\frac{\partial [\mathbf{P}]_{ij}}{\partial [\mathbf{W}]_{kt}} = -\frac{2}{\sigma_P}[\mathbf{P}]_{ij}([\mathbf{Y}]_{it} - [\mathbf{Y}]_{jt})([\mathbf{X}]_{ik} - [\mathbf{X}]_{jk}),$$

and $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N]^T \in \mathbb{R}^{N \times d}$ is the matrix that contains the data samples and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N]^T \in \mathbb{R}^{N \times m}$ is the matrix that contains the projected data samples, i.e., $\mathbf{y}_i = f_{\mathbf{W}}(\mathbf{x}_i)$. Finally, it is trivial to define an orthonomality regularizer for the linear projection function [53]:

$$J_p = \frac{1}{2m^2}||\mathbf{W}^T\mathbf{W} - \mathbf{I}_{m \times m}||_F^2,$$

where $\mathbf{I}_{m \times m}$ is the $m \times m$ identity matrix. The corresponding derivative can be calculated as:

$$\frac{\partial J_p}{\partial [\mathbf{W}]_i} = \frac{2}{m^2}\sum_{j=1}^{m}([\mathbf{W}]_i^T[\mathbf{W}]_j - \delta_{ij})[\mathbf{W}]_j, \tag{11}$$

where the notation $[\mathbf{W}]_i$ is used to refer to the i-th column vector of the matrix $\mathbf{W}$ and δ_{ij} is the Kronecker delta function.

The final solution to the optimization problem:

$$\mathbf{W} = arg\,min_{\mathbf{W}} J(\mathcal{X}, \mathbf{W}), \tag{12}$$

is obtained by performing gradient descent using the gradients calculated in (10) and (11), where $J(\mathcal{X}, \mathbf{W})$ is defined in (4). After reducing the dimensionality of the data, k-means can be used again (using the low-dimensional representation obtained by the projection function $f_{\mathbf{W}}(\cdot)$) to acquire the improved clustering solution $\mathcal{S}_1$ by solving the following updated optimization problem:

$$\mathcal{S}_1 = arg\,min_{\mathcal{S}} \sum_{i=0}^{N_C} \sum_{\mathbf{x} \in \mathcal{S}^{(i)}} ||f_{\mathbf{W}}(\mathbf{x}) - \boldsymbol{\mu}_i||_2^2, \tag{13}$$

where the mean vectors are appropriately updated. The final optimization objective:

$$min_{\mathcal{S},\mathbf{W}} \sum_{i=0}^{N_C} \sum_{\mathbf{x} \in \mathcal{S}^{(i)}} ||f_{\mathbf{W}}(\mathbf{x}) - \boldsymbol{\mu}_i||_2^2, \tag{14}$$

where $\mathbf{W} \in \mathbb{R}^{d \times m}$, is optimized using an alternating optimization scheme, i.e., by alternatively solving the two subproblems defined in (12) and in (13). Finally, note that the learned representation $f_{\mathbf{W}}(\mathbf{x})$ can be also used for a wide variety of tasks, such as information retrieval or visualization.

The complete proposed algorithm is shown in Algorithm 1. First, the initial clustering solution is obtained by clustering the original data (lines 2-3). Then, N_{iters} alternating optimization steps are performed (lines 4-11), where a projection function is learned/updated (lines 5-8) by optimizing the proposed objective toward learning a regularized clustering-oriented representation, the data are projected into a lower dimensional subspace (line 9) and the clustering solution is updated (line 10). It is worth noting that any other clustering algorithm can be also used, instead of k-means, for providing the initial and updated clustering solutions. Also, note that the proposed method can also readily provide out-of-sample extensions, when k-means is used, since any new sample can be projected into the lower dimensional space and assigned to its nearest

Algorithm 1 Similarity-based Discriminative Clustering Algorithm

Input: A set of points $\mathcal{X}$, the batch size N_{batch}, the number of optimization steps N_{iters}, gradient descent iterations $N_{sgditers}$, and clusters N_C

Output: The clustering solution $\mathcal{S}_1$

1: **procedure** Similarity-based Discriminative Clustering
2: Calculate the initial clustering solution $\mathcal{S}_0$ using k-means (i.e., solve the problem defined in (6))
3: Set $\mathcal{S} = \mathcal{S}_0$
4: **for** i **from** 1 **to** N_{iters} **do**
5: **for** i **from** 1 **to** $N_{sgditers}$ **do**
6: Sample a batch of data $\mathbf{x}$
7: Construct the target similarity matrix for the selected samples $\mathbf{x}$ using (8) and the current solution $\mathcal{S}$.
8: Perform one optimization iteration using (5)
9: Project the data samples into the new low-dimensional space defined by $f_{\mathbf{W}}(\cdot)$
10: Calculate the updated clustering solution $\mathcal{S}_1$ using k-means on the low-dimensional representation (i.e., solve the problem defined in (12))
11: $\mathcal{S} = \mathcal{S}_1$
return the final clustering solution $\mathcal{S}$

cluster. This is in contrast with other clustering techniques, e.g., spectral clustering [14], that are unable to handle out-of-sample data, i.e., data that were not in the training set. Finally, note that for projecting one data sample in the learned subspace (line 9) $O(dm)$ time is needed, since a simple linear projection matrix is employed. Also, for performing one iteration of the optimization $O(N_{batch}^2)$ time is needed for calculating the target similarity matrix, while $O(N_{batch}^2 dm)$ time is required for calculating the required derivatives. Therefore, the total complexity of the optimization process is $O(N_{iters}(N_{sgditers}N_{batch}^2 dm + c))$, where c is the complexity of the employed clustering algorithm, e.g., $c = O(Nkmn_{kiters})$ for k-means (N is the total number of data samples, k is the number of clusters and n_{kiters} is the number of iterations needed until convergence).

4. Experiments

In this Section, the proposed method is evaluated and compared to other clustering techniques. First, the used datasets are briefly described and the evaluation setup and metrics are introduced. Next, the evaluation results, using two different clustering methods, are presented and discussed in detail. Finally, the ability of the proposed method to provide regularized solutions is demonstrated by examining the effect of the two target similarity parameters, the intra-cluster similarity a_{intra} and the inter-cluster similarity a_{inter}.

4.1. Datasets

Four datasets from a wide range of domains were used to evaluate the proposed method: one face recognition dataset, the extended Yale B dataset (abbreviated as Yale B) [58], two multi-class image recognition datasets, the 15-scene dataset [22] and the Corel dataset [59], and one multi-class video dataset, the KTH action recognition database [60]. The cropped Extended Yale Face Database B [58], contains 2,432 images from 38 individuals, taken under greatly varying lighting conditions. The size of each image is 168×192 and the raw pixel representation is used for each image. The 15-scene dataset [22], contains 15 different scene categories. The total number of images is 4,485 and each category has 200 to 400 images. HoG [61], and LBP features [62], of 8×8 non-overlapping patches were densely extracted from each image. The two feature vectors extracted from each patch were fused together to form the final feature vector. Then, the BoF model [63], was used to learn a dictionary (using the k-means algorithm) and extract a 512-dimensional histogram vector for each image. The Corel dataset [59], contains 10,800 images from 80 different concepts and a similar feature extraction process was used (HoG and LBP features of 8×8 patches were densely extracted from each image and histogram vectors were compiled for the images using the BoF model with 512 codewords). The KTH action recognition dataset [60], is composed of 2,391 video sequences that are classified into six different categories (walking, jogging, running, boxing, hand waving and hand clapping). The train+validation (1,528 videos) and the test (863 videos) splits are predefined. From each video, HoG and HoF descriptors are extracted [64]. For each type of descriptor a BoF dictionary with 512 codewords is learned. Then, each video is represented by fusing the extracted histogram vectors.

4.2. Evaluation Setup and Metrics

For all the conducted experiments, 5 alternating optimization iterations were performed ($N_{iters} = 5$), while 10 full SGD epochs were performed for each iteration. The regularizer parameter α_p was set to 1 (except for the Yale dataset for which smaller values are usually used, i.e., $\alpha_p = 10^{-5}$, due to its high dimensionality [53]). The batch size was set to $N_{batch} = 128$, while the learning rate was set to $\eta = 0.001$. The intra-cluster and inter-cluster similarities were set to $a_{intra} = 0.8$ and $a_{inter} = 0.2$ respectively for all the conducted experiments (again except for the Yale and the KTH datasets that were more prune to over-fitting and these parameters were set to $a_{intra} = 0.5$ and $a_{inter} = 0.3$). The number of clusters N_C was set to the number of classes of each dataset. The dimensionality of the low-dimensional space was set to 50 for all the conducted experiments and evaluated methods. For the DisCluster method [23], the dimensionality of the discriminative low dimensional space was set to the largest possible value, i.e., $\min(50, N_C - 1)$. Also, to ensure a fair comparison between DisCluster and the proposed approach the same number of optimization iterations ($N_{iter} = 5$) and the same initialization scheme (Algorithm 1, line 2) were used for both methods. To ensure the stability of the LDA method used in DisCluster, the shrinkage parameter was set to 0.9 for all the conducted experiments (this value led to the best performance during the conducted experiments, while having a positive regularization effect on the representations learned by DisCluster).

For the 15-scene 100 images were randomly sampled from each class to form the set used for training the method and evaluating the performance of the methods (in-sample evaluation). We also used a different split, called *out-of-sample* split, to evaluate the behavior of the proposed method on samples that were not seen during the training. For the Corel dataset the in-sample set was composed of 5,000 randomly sampled images, while for the Yale dataset 30 images were sampled for each person. Again, the rest of the images were used to evaluate the out-of-sample behavior of the proposed method. For the KTH dataset, the predefined train and test splits were used for the in-sample and out-of-sample evaluation respectively. All the experiments were repeated 5 times and the mean and standard deviation of the evaluated metrics are reported.

For evaluating the quality of the obtained clustering solutions five different external evaluation criteria were used [65]. External criteria measure how well the clusters correspond to some predefined categories (usually the classes of the dataset are used). Internal evaluation criteria, that measure how well the data are clustered without using any ground-truth information, can be also used for evaluating clustering solutions (especially when no ground-truth information is available). However, internal criteria can be easily maximized with degenerate solutions (e.g., using a kernel method, such as Kernel LDA [66], to collapse the clusters into single points). Therefore, in this work we use the following 5 external evaluation criteria: Adjusted Rand Index [67], Normalized Mutual Information [68], Homogeneity [69], Completeness [69], and Fowlkes-Mallows [70].

4.3. Clustering Evaluation

First, the proposed method was evaluated using the well-known Yale B dataset. The evaluation results are reported in Table 1. The proposed method was also compared to several other techniques: a) performing k-means in the original space (abbreviated as "k-means"), b) performing dimensionality reduction using the PCA method and then clustering the data points using k-means (abbreviated as "PCA"), and c) using the DisCluster method [23].

Several conclusions can be drawn from the results of Table 1. First, reducing the dimensionality of the data using a generic unsupervised technique, i.e., PCA, does not significantly affect the quality of the obtained clustering solutions. The DisCluster method improves mainly the out-of-sample evaluation criteria by maintaining the information contained in data used for training. On the other hand, the proposed method leads to spectacular improvements in the clustering accuracy (the Rand index increases more than 500%, NMI more than 80% and FMI more than 200%), demonstrating the importance of using well-regularized clustering-oriented representations for performing clustering tasks. This behavior can be also attributed to the well-defined geometry of the Yale B dataset. The strong regularization effect of maintaining a small intra-cluster similarity, used in the proposed method, forces the data samples that depicts the same face in different poses to unfold into a circular structure (similar to those shown in Fig. 1c). This way, the proposed method is capable of effectively discovering and unfolding the manifolds that exist in the data, leading to this spectacular improvements in clustering performance.

The 15-scene dataset was used for the evaluation. Again, using PCA does not alter the clustering metrics, while the DisCluster method slightly improves all the clustering metrics. As before, the proposed method significantly improves all the evaluated metrics over the rest of the evaluated techniques for both the in-sample and out-of-sample evaluation. This behavior is also confirmed using the Corel and KTH datasets.

4.4. Spectral Clustering Evaluation

Next, the proposed method was combined with spectral clustering [14]. Therefore, instead of using k-means for providing the initial and the updated clustering solutions (lines 2 and 10 of Algorithm 1), spectral clustering was used. The spectral clustering implementation provided by the scikit-learn library was used [71], combined with a k-nearest neighbor affinity matrix (k was set to 200). The evaluation results are reported in Table 2. Several interesting conclusions can be drawn. First, the proposed method significantly improves the evaluation metrics over the original space or using the PCA/DisCluster methods. Furthermore, using spectral clustering improve the original clustering results for all the evaluated datasets. Combining the proposed method with spectral clustering also outperforms the standard k-means clustering approach for the 15-scene dataset (e.g., the Rand index increases from 0.192 to 0.241) and the Corel dataset (e.g., the NMI increases from 0.411 to 452). For the other two evaluated datasets (Yale and KTH) the proposed SDC method yields better results when combined with the standard k-means clustering approach. Nonetheless, the proposed method always outperforms all the other evaluated techniques, regardless the used clustering algorithm, improving the acquired clustering solution in any evaluation scenario. The proposed method runs in comparable time with the rest of the evaluated methods. For example, the time required to run the evaluated methods on the Yale B dataset using a six-core CPU workstation with 32GB of RAM equipped with a mid-range graphics card are as follows: 8.1s (seconds) for the plain k-means algorithm, 0.7s for PCA + k-means, 54.9s for DisCluster and 8.8 sec for the proposed method. Even though the implementation used for the proposed method was not fully optimized, its ability to run the optimization in batches, exploiting the available GPU accelerator, provided a significant performance benefit over DisCluster, which is severely impacted by the high dimensionality of the data, while achieving better clustering accuracy.

To further demonstrate the ability of the proposed method to work with existing clustering approaches, we also combined and evaluated SDC with another spectral clustering method, the Robust Spectral Clustering (RSC) [36]. The implementation provided by the authors of the method was used, while the number of neighbors was set to 30 for all the conducted experiments, while θ was set to 20 and at least 90% of neighbors were kept for each node (please refer to [36] for more details regarding the RSC method). The experimental evaluation is provided in Table 3. Similarly with the other two clustering approaches, combining the proposed method with RSC leads to the best results compared to the other evaluated subspace learning approaches. It is worth noting that even though RSC performs slightly worse on the YALE dataset (Rand index of 0.021 instead of 0.025 for the plain spectral clustering), it outperforms all the other evaluated clustering methods when combined with the proposed subspace learning method. For example, Rand index increases from 0.108 for the plain spectral clustering + SDC to 0.155 for RSC+SDC. This behavior highlights the ability of the

Table 1: Clustering evaluation using k-means. "in" refers to "in-sample" evaluation, while "out" refers to "out-of-sample" evaluation.

Method	Dataset	**Type**	**Rand**	**NMI**	**Homogeneity**	**Completeness**	**FMI**
Original	Yale B	in	0.020 ± 0.004	0.229 ± 0.009	0.226 ± 0.009	0.233 ± 0.009	0.048 ± 0.004
PCA	Yale B	in	0.018 ± 0.003	0.227 ± 0.010	0.224 ± 0.010	0.230 ± 0.010	0.046 ± 0.003
DisCluster	Yale B	in	0.020 ± 0.005	0.229 ± 0.014	0.225 ± 0.013	0.233 ± 0.014	0.049 ± 0.022
SDC	Yale B	in	$\mathbf{0.124 \pm 0.012}$	$\mathbf{0.418 \pm 0.016}$	$\mathbf{0.414 \pm 0.015}$	$\mathbf{0.423 \pm 0.018}$	$\mathbf{0.149 \pm 0.005}$
Original	Yale B	out	0.011 ± 0.002	0.193 ± 0.005	0.189 ± 0.005	0.197 ± 0.004	0.041 ± 0.004
PCA	Yale B	out	0.012 ± 0.003	0.195 ± 0.009	0.191 ± 0.009	0.199 ± 0.008	0.041 ± 0.004
DisCluster	Yale B	out	0.025 ± 0.003	0.226 ± 0.006	0.220 ± 0.006	0.231 ± 0.007	0.055 ± 0.018
SDC	Yale B	out	$\mathbf{0.111 \pm 0.006}$	$\mathbf{0.389 \pm 0.010}$	$\mathbf{0.383 \pm 0.009}$	$\mathbf{0.395 \pm 0.010}$	$\mathbf{0.137 \pm 0.003}$
Original	15-scene	in	0.168 ± 0.011	0.360 ± 0.008	0.344 ± 0.007	0.376 ± 0.010	0.239 ± 0.005
PCA	15-scene	in	0.168 ± 0.014	0.359 ± 0.012	0.338 ± 0.013	0.382 ± 0.012	0.243 ± 0.006
DisCluster	15-scene	in	0.174 ± 0.008	0.377 ± 0.009	0.354 ± 0.009	0.401 ± 0.010	0.250 ± 0.009
SDC	15-scene	in	$\mathbf{0.200 \pm 0.004}$	$\mathbf{0.397 \pm 0.007}$	$\mathbf{0.389 \pm 0.008}$	$\mathbf{0.405 \pm 0.006}$	$\mathbf{0.259 \pm 0.003}$
Original	15-scene	out	0.148 ± 0.006	0.333 ± 0.005	0.322 ± 0.005	0.345 ± 0.006	0.221 ± 0.003
PCA	15-scene	out	0.143 ± 0.012	0.330 ± 0.013	0.314 ± 0.010	0.346 ± 0.016	0.221 ± 0.003
DisCluster	15-scene	out	0.161 ± 0.009	0.354 ± 0.007	0.335 ± 0.009	0.375 ± 0.006	0.241 ± 0.008
SDC	15-scene	out	$\mathbf{0.192 \pm 0.007}$	$\mathbf{0.378 \pm 0.008}$	$\mathbf{0.371 \pm 0.009}$	$\mathbf{0.385 \pm 0.008}$	$\mathbf{0.256 \pm 0.004}$
Original	Corel	in	0.102 ± 0.002	0.399 ± 0.003	0.391 ± 0.004	0.408 ± 0.002	0.120 ± 0.002
PCA	Corel	in	0.104 ± 0.006	0.398 ± 0.004	0.390 ± 0.005	0.407 ± 0.003	0.122 ± 0.001
DisCluster	Corel	in	0.091 ± 0.003	0.391 ± 0.003	0.378 ± 0.003	0.404 ± 0.004	0.111 ± 0.004
SDC	Corel	in	$\mathbf{0.112 \pm 0.004}$	$\mathbf{0.425 \pm 0.003}$	$\mathbf{0.428 \pm 0.003}$	$\mathbf{0.422 \pm 0.003}$	$\mathbf{0.126 \pm 0.002}$
Original	Corel	out	0.103 ± 0.005	0.388 ± 0.003	0.377 ± 0.004	0.399 ± 0.002	0.122 ± 0.002
PCA	Corel	out	0.103 ± 0.001	0.386 ± 0.003	0.375 ± 0.004	0.396 ± 0.003	0.121 ± 0.002
DisCluster	Corel	out	0.092 ± 0.005	0.381 ± 0.002	0.367 ± 0.002	0.396 ± 0.003	0.113 ± 0.003
SDC	Corel	out	$\mathbf{0.109 \pm 0.005}$	$\mathbf{0.411 \pm 0.003}$	$\mathbf{0.413 \pm 0.003}$	$\mathbf{0.410 \pm 0.003}$	$\mathbf{0.124 \pm 0.003}$
Original	KTH	in	0.361 ± 0.046	0.496 ± 0.041	0.482 ± 0.046	0.510 ± 0.036	0.480 ± 0.007
PCA	KTH	in	0.396 ± 0.010	0.535 ± 0.025	0.520 ± 0.020	0.550 ± 0.031	0.507 ± 0.005
DisCluster	KTH	in	0.409 ± 0.042	0.547 ± 0.051	0.531 ± 0.052	0.564 ± 0.051	0.519 ± 0.028
SDC	KTH	in	$\mathbf{0.437 \pm 0.041}$	$\mathbf{0.568 \pm 0.046}$	$\mathbf{0.560 \pm 0.047}$	$\mathbf{0.576 \pm 0.046}$	$\mathbf{0.536 \pm 0.005}$
Original	KTH	out	0.386 ± 0.043	0.522 ± 0.025	0.499 ± 0.037	0.546 ± 0.013	0.506 ± 0.007
PCA	KTH	out	0.425 ± 0.014	0.555 ± 0.026	0.535 ± 0.025	0.576 ± 0.030	0.534 ± 0.018
DisCluster	KTH	out	0.438 ± 0.055	0.568 ± 0.052	0.545 ± 0.056	0.592 ± 0.051	0.547 ± 0.020
SDC	KTH	out	$\mathbf{0.478 \pm 0.039}$	$\mathbf{0.594 \pm 0.039}$	$\mathbf{0.584 \pm 0.042}$	$\mathbf{0.603 \pm 0.035}$	$\mathbf{0.570 \pm 0.009}$

Table 2: Spectral Clustering Evaluation

Method	Dataset	Rand	NMI	Homogeneity	Completeness	FMI
Original	Yale	0.025 ± 0.003	0.247 ± 0.007	0.245 ± 0.008	0.248 ± 0.007	0.051 ± 0.002
PCA	Yale	0.027 ± 0.003	0.251 ± 0.006	0.249 ± 0.006	0.253 ± 0.006	0.053 ± 0.007
DisCluster	Yale	0.039 ± 0.005	0.284 ± 0.009	0.281 ± 0.010	0.288 ± 0.008	0.067 ± 0.016
SDC	Yale	$\mathbf{0.108 \pm 0.005}$	$\mathbf{0.411 \pm 0.006}$	$\mathbf{0.408 \pm 0.006}$	$\mathbf{0.414 \pm 0.006}$	$\mathbf{0.133 \pm 0.005}$
Original	15-scene	0.189 ± 0.016	0.353 ± 0.018	0.351 ± 0.018	0.354 ± 0.018	0.244 ± 0.009
PCA	15-scene	0.193 ± 0.014	0.355 ± 0.017	0.354 ± 0.017	0.357 ± 0.017	0.247 ± 0.007
DisCluster	15-scene	0.212 ± 0.013	0.381 ± 0.013	0.380 ± 0.013	0.382 ± 0.013	0.265 ± 0.007
SDC	15-scene	$\mathbf{0.241 \pm 0.016}$	$\mathbf{0.420 \pm 0.017}$	$\mathbf{0.418 \pm 0.017}$	$\mathbf{0.422 \pm 0.017}$	$\mathbf{0.292 \pm 0.001}$
Original	Corel	0.096 ± 0.005	0.431 ± 0.003	0.435 ± 0.004	0.426 ± 0.003	0.110 ± 0.003
PCA	Corel	0.099 ± 0.004	0.436 ± 0.002	0.441 ± 0.002	0.430 ± 0.003	0.113 ± 0.003
DisCluster	Corel	0.103 ± 0.002	0.439 ± 0.003	0.444 ± 0.003	0.434 ± 0.003	0.117 ± 0.006
SDC	Corel	$\mathbf{0.116 \pm 0.005}$	$\mathbf{0.452 \pm 0.002}$	$\mathbf{0.456 \pm 0.003}$	$\mathbf{0.448 \pm 0.002}$	$\mathbf{0.129 \pm 0.002}$
Original	KTH	0.403 ± 0.000	0.532 ± 0.000	0.524 ± 0.000	0.540 ± 0.000	0.507 ± 0.000
PCA	KTH	0.400 ± 0.002	0.528 ± 0.002	0.521 ± 0.002	0.535 ± 0.002	0.504 ± 0.000
DisCluster	KTH	0.400 ± 0.001	0.541 ± 0.001	$\mathbf{0.537 \pm 0.001}$	0.545 ± 0.001	0.502 ± 0.000
SDC	KTH	$\mathbf{0.411 \pm 0.003}$	$\mathbf{0.543 \pm 0.003}$	0.536 ± 0.003	$\mathbf{0.551 \pm 0.003}$	$\mathbf{0.513 \pm 0.001}$

proposed method to produce regularized subspaces that can be then exploited by more powerful clustering approaches, such as RSC. On the other hand, more discriminative methods, such DisCluster, tend to be more sensitive to the used initialization, e.g., DisCluster leads to a Rand index of 0.039 for spectral clustering, but its performance is severely limited when the initialization was not good enough (Rand index of 0.027 when combined with RSC).

4.5. Regularization Evaluation

To demonstrate the effect of the two regularization parameters, the intra-cluster similarity a_{intra} and inter-cluster similarity a_{inter}, some additional experiments were performed. The effect of varying the value of

Table 3: Robust Spectral Clustering Evaluation

Method	Dataset	Rand	NMI	Homogeneity	Completeness	FMI
Original	Yale	0.021 ± 0.002	0.228 ± 0.005	0.228 ± 0.004	0.229 ± 0.005	0.047 ± 0.001
PCA	Yale	0.021 ± 0.003	0.230 ± 0.008	0.229 ± 0.008	0.230 ± 0.008	0.047 ± 0.004
DisCluster	Yale	0.027 ± 0.004	0.245 ± 0.009	0.245 ± 0.009	0.246 ± 0.009	0.053 ± 0.008
SDC	Yale	$\mathbf{0.155 \pm 0.012}$	$\mathbf{0.450 \pm 0.007}$	$\mathbf{0.449 \pm 0.006}$	$\mathbf{0.451 \pm 0.007}$	$\mathbf{0.177 \pm 0.005}$
Original	15scene	0.189 ± 0.007	0.369 ± 0.010	0.364 ± 0.010	0.375 ± 0.010	0.248 ± 0.004
PCA	15scene	0.197 ± 0.005	0.370 ± 0.010	0.366 ± 0.008	0.374 ± 0.011	0.253 ± 0.002
DisCluster	15scene	0.200 ± 0.010	0.387 ± 0.018	0.382 ± 0.018	0.392 ± 0.018	0.257 ± 0.005
SDC	15scene	$\mathbf{0.245 \pm 0.003}$	$\mathbf{0.425 \pm 0.005}$	$\mathbf{0.421 \pm 0.004}$	$\mathbf{0.429 \pm 0.006}$	$\mathbf{0.298 \pm 0.002}$

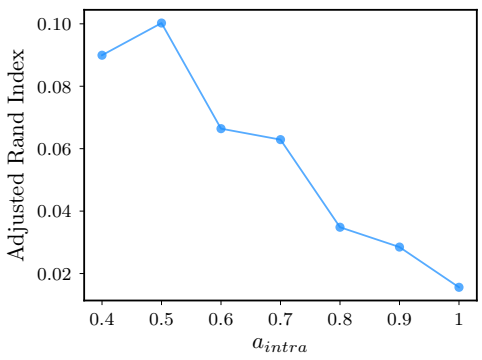


Figure 2: Evaluating the effect of varying the target intra-cluster similarity on the adjusted Rand index

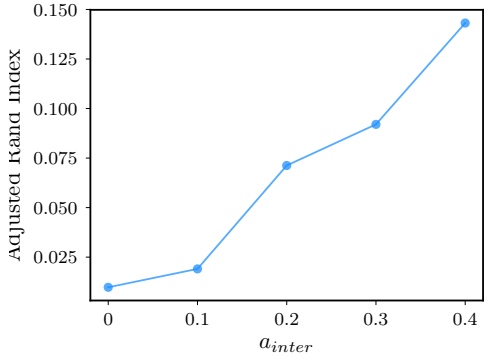


Figure 3: Evaluating the effect of varying the target inter-cluster similarity on the adjusted Rand index

a_{intra} on the Rand index is shown in Figure 2 using the Yale dataset (a_{inter} was set to 0.3 for the conducted experiments). Recall that using a value of $a_{intra} = 1$ yields the similarity-based equivalent of the LDA method. Indeed, as the value of a_{intra} *decreases* the clustering metrics improve highlighting the importance of avoiding collapsing the samples of the same class into small regions of the space. Instead, using more relaxed constraints, i.e., allowing the in-cluster samples to maintain a small distance between them, improves the quality of the clustering solutions. For the specific dataset, the best results were obtained for $a_{intra} = 0.5$, even though any value smaller than 1 seems to improve the evaluated metrics.

The effect of varying the value of the inter-cluster similarity a_{inter} was evaluated in Figure 3 (a_{intra} was set to 0.5 for the conducted experiments). Similarly, setting $a_{inter} = 0$ yields the similarity-based equivalent of the LDA method. Again, it is demonstrated that relaxing this requirement, i.e., move the clusters as far apart as possible, improves the clustering metrics. The best results for this dataset were obtained for $a_{inter} = 0.4$, even though any other value larger than 0 also improves the obtained solutions. It should be noted that these values were not finetuned for the conducted clustering evaluation experiments. Instead, two generic sets of values were used to demonstrate the effectiveness of the proposed method, without finetuning these two regularization parameters to each dataset. For example, for the Yale dataset the clustering performance can be further increased, from 0.124 to 0.143 (as measured using the Adjusted Rand Index), by setting $a_{inter} = 0.4$ and $a_{intra} = 0.5$.

5. Conclusions and Future Work

In this work we proposed a discriminative clustering method that is able to provide regularized low-dimensional representations that are optimized toward clustering tasks. The intra-cluster and the inter-cluster distances are transformed into the similarities and then manipulated in an appropriate way that ensures that a robust clustering-oriented representation will be learned. The proposed method is capable of readily scaling to large datasets and maintains its expressive power for both in-sample and out-of-sample data.

Furthermore, the ability of the proposed method to improve the clustering solutions over other clustering techniques was demonstrated using extensive experiments on four datasets from a diverse range of domains. The proposed similarity-based formulation of Discriminative Clustering will allow for deriving a wide range of other clustering techniques on top of (or inspired by) the proposed method. The code of the proposed method will be available as part of the PySEF library [28] at `https://github.com/passalis/sef`.

There are several interesting future research directions, especially given the impressive results that were obtained for some of the datasets, e.g., the Yale dataset. The ability of the proposed method to provide useful representations for other unsupervised tasks, e.g., information retrieval [4, 72], or data visualization [55], can be examined. Also, the optimization objective used in this paper can be also combined with other deep clustering architectures, e.g., [73, 74], to allow for learning deep regularized models for clustering tasks. Finally, the regularization parameters a_{intra} and α_{inter} can be adaptively chosen based on the confidence/probability for each sample to belong to each cluster, possibly further improving the clustering performance.

References

[1] T. Dasu, T. Johnson, Exploratory data mining and data cleaning, Vol. 479, John Wiley & Sons, 2003.

[2] C. M. Bishop, Pattern recognition and machine learning, Springer, 2006.

[3] R. S. Michalski, J. G. Carbonell, T. M. Mitchell, Machine learning: An artificial intelligence approach, Springer Science & Business Media, 2013.

[4] R. Baeza-Yates, B. Ribeiro-Neto, et al., Modern information retrieval, Vol. 463, ACM Press, New York, 1999.

[5] A. Likas, N. Vlassis, J. J. Verbeek, The global k-means clustering algorithm, Pattern recognition 36 (2) (2003) 451–461.

[6] M. Filippone, F. Camastra, F. Masulli, S. Rovetta, A survey of kernel and spectral methods for clustering, Pattern recognition 41 (1) (2008) 176–190.

[7] G. J. Myatt, Making sense of data: a practical guide to exploratory data analysis and data mining, John Wiley & Sons, 2007.

[8] N. Tsapanos, A. Tefas, N. Nikolaidis, I. Pitas, A distributed framework for trimmed kernel k-means clustering, Pattern recognition 48 (8) (2015) 2685–2698.

[9] H.-P. Kriegel, P. Kröger, J. Sander, A. Zimek, Density-based clustering, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 1 (3) (2011) 231–240.

[10] L. Bai, X. Cheng, J. Liang, H. Shen, Y. Guo, Fast density clustering strategies based on the k-means algorithm, Pattern Recognition 71 (2017) 375–386.

[11] M. Zarinbal, M. F. Zarandi, I. Turksen, Relative entropy collaborative fuzzy clustering method, Pattern Recognition 48 (3) (2015) 933–940.

[12] M.-S. Yang, Y. Nataliani, Robust-learning fuzzy c-means clustering algorithm with unknown number of clusters, Pattern Recognition 71 (2017) 45–59.

[13] S. K. Choy, S. Y. Lam, K. W. Yu, W. Y. Lee, K. T. Leung, Fuzzy model-based clustering and its application in image segmentation, Pattern Recognition 68 (2017) 141–157.

[14] A. Y. Ng, M. I. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, in: Proceedings of the Advances in Neural Information Processing Systems, 2002, pp. 849–856.

[15] Z. Deng, K.-S. Choi, F.-L. Chung, S. Wang, Enhanced soft subspace clustering integrating within-cluster and between-cluster information, Pattern Recognition 43 (3) (2010) 767–781.

[16] C. You, D. Robinson, R. Vidal, Scalable sparse subspace clustering by orthogonal matching pursuit, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3918–3927.

[17] X. Chen, Y. Ye, X. Xu, J. Z. Huang, A feature group weighting method for subspace clustering of high-dimensional data, Pattern Recognition 45 (1) (2012) 434–446.

[18] A. Krause, P. Perona, R. G. Gomes, Discriminative clustering by regularized information maximization, in: Proceedings of the Advances in Neural Information Processing Systems, 2010, pp. 775–783.

[19] G. Huang, T. Liu, Y. Yang, Z. Lin, S. Song, C. Wu, Discriminative clustering via extreme learning machine, Neural Networks 70 (2015) 1–8.

[20] R. Shang, Z. Zhang, L. Jiao, W. Wang, S. Yang, Global discriminative-based nonnegative spectral clustering, Pattern Recognition 55 (2016) 172–182.

[21] J. Ye, Z. Zhao, M. Wu, Discriminative k-means for clustering, in: Proceedings of the Advances in Neural Information Processing Systems, 2008, pp. 1649–1656.

[22] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2, 2006, pp. 2169–2178.

[23] C. Ding, T. Li, Adaptive dimension reduction using discriminant analysis and k-means clustering, in: Proceedings of the International Conference on Machine learning, 2007, pp. 521–528.

[24] G. McLachlan, Discriminant analysis and statistical pattern recognition, Vol. 544, John Wiley & Sons, 2004.

[25] K. Torkkola, Linear discriminant analysis in document classification, in: IEEE ICDM Workshop on Text Mining, 2001, pp. 800–806.

[26] A. Iosifidis, A. Tefas, I. Pitas, Discriminant bag of words based representation for human action recognition, Pattern Recognition Letters 49 (2014) 185–192.

[27] Y. Bengio, J.-f. Paiement, P. Vincent, O. Delalleau, N. L. Roux, M. Ouimet, Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering, in: Proceedings of the Advances in Neural Information Processing Systems, 2004, pp. 177–184.

[28] N. Passalis, A. Tefas, Pysef: A python library for similarity-based dimensionality reduction, Knowledge-Based Systems.

[29] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, A. Y. Wu, An efficient k-means clustering algorithm: Analysis and implementation, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (7) (2002) 881–892.

[30] C. K. L. Lekamalage, T. Liu, Y. Yang, Z. Lin, G.-B. Huang, Extreme learning machine for clustering, in: Proceedings of ELM-2014, 2015, pp. 435–444.

[31] S. P. Lloyd, Least squares quantization in pcm, IEEE Transactions on Information Theory 28 (2) (1982) 129–137.

[32] Z. Huang, Extensions to the k-means algorithm for clustering large data sets with categorical values, Data Mining and Knowledge Discovery 2 (3) (1998) 283–304.

[33] D. Arthur, S. Vassilvitskii, k-means++: The advantages of careful seeding, in: Proceedings of the ACM-SIAM symposium on Discrete algorithms, 2007, pp. 1027–1035.

[34] N. Tsapanos, A. Tefas, N. Nikolaidis, I. Pitas, Efficient mapreduce kernel k-means for big data clustering, in: Proceedings of the Hellenic Conference on Artificial Intelligence, 2016, p. 28.

[35] N. Passalis, A. Tefas, Spectral clustering using optimized bag-of-features, in: Proceedings of the Hellenic Conference on Artificial Intelligence, 2016, p. 19.

[36] A. Bojchevski, Y. Matkovic, S. Günnemann, Robust spectral clustering for noisy data: Modeling sparse corruptions improves latent embeddings, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, pp. 737–746.

[37] Y. Wang, L. Wu, X. Lin, J. Gao, Multiview spectral clustering via structured low-rank matrix factorization, IEEE Transactions on Neural Networks and Learning Systems, 2018.

[38] H. Hu, Z. Lin, J. Feng, J. Zhou, Smooth representation clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3834–3841.

[39] C.-G. Li, R. Vidal, Structured sparse subspace clustering: A unified optimization framework, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 277–286.

[40] H. Chen, W. Wang, X. Feng, Structured sparse subspace clustering with within-cluster grouping, Pattern Recognition 83 (2018) 107–118.

[41] M. Caron, P. Bojanowski, A. Joulin, M. Douze, Deep clustering for unsupervised learning of visual features, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 132–149.

[42] K. Ghasedi Dizaji, A. Herandi, C. Deng, W. Cai, H. Huang, Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 5736–5745.

[43] E. Tzinis, S. Venkataramani, P. Smaragdis, Unsupervised deep clustering for source separation: Direct learning from mixtures using spatial information, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019, pp. 81–85.

[44] F. De la Torre, T. Kanade, Discriminative cluster analysis, in: Proceedings of the International Conference on Machine Learning, 2006, pp. 241–248.

[45] J. Ye, Z. Zhao, H. Liu, Adaptive distance metric learning for clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–7.

[46] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1-3) (2006) 489–501.

[47] G. Huang, J. Zhang, S. Song, Z. Chen, Maximin separation probability clustering, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2015.

[48] L. Xu, J. Neufeld, B. Larson, D. Schuurmans, Maximum margin clustering, in: Proceedings of the Advances in Neural Information Processing Systems, 2005, pp. 1537–1544.

[49] I. Steinwart, A. Christmann, Support vector machines, Springer Science & Business Media, 2008.

[50] L. Xu, D. Schuurmans, Unsupervised and semi-supervised multi-class support vector machines, in: Proceedings of the AAAI Conference on Artificial Intelligence, Citeseer, 2005.

[51] K. Zhang, I. W. Tsang, J. T. Kwok, Maximum margin clustering made practical, IEEE Transactions on Neural Networks 20 (4) (2009) 583–596.

[52] B. Zhao, F. Wang, C. Zhang, Efficient multiclass maximum margin clustering, in: Proceedings of the International Conference on Machine learning, 2008, pp. 1248–1255.

[53] N. Passalis, A. Tefas, Dimensionality reduction using similarity-induced embeddings, IEEE Transactions on Neural Networks and Learning Systems 29 (8) (2018) 3429–3441.

[54] N. Passalis, A. Tefas, Learning discriminative representations for big data clustering using similarity-based dimensionality reduction, in: Proceedings of the IEEE Image, Video, and Multidimensional Signal Processing Workshop, 2018, pp. 1–5.

[55] A. N. Gorban, B. Kégl, D. C. Wunsch, A. Y. Zinovyev, et al., Principal manifolds for data visualization and dimension reduction, Vol. 58 of Lecture Notes in Computational Science and Engineering, Springer, 2008.

[56] M. Rosenblatt, et al., Remarks on some nonparametric estimates of a density function, The Annals of Mathematical Statistics 27 (3) (1956) 832–837.

[57] D. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv:1412.6980.

[58] K. Lee, J. Ho, D. Kriegman, Acquiring linear subspaces for face recognition under variable lighting, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (5) (2005) 684–698.

[59] W. Bian, D. Tao, The COREL database for content based image retrieval, `https://sites.google.com/site/dctresearch/Home/content-based-image-retrieval` (2009).

[60] C. Schüldt, I. Laptev, B. Caputo, Recognizing human actions: a local SVM approach, in: Proceedings of the 17th International Conference on Pattern Recognition, Vol. 3, 2004, pp. 32–36.

[61] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 1, 2005, pp. 886–893.

[62] T. Ojala, M. Pietikäinen, T. Mäenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (7) (2002) 971–987.

[63] J. Sivic, A. Zisserman, Video google: A text retrieval approach to object matching in videos, in: Proceedings of the IEEE International Conference on Computer Vision, 2003, pp. 1470–1477.

[64] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[65] H. Schütze, Introduction to information retrieval, in: Proceedings of the International Communication of Association for Computing Machinery Conference, 2008.

[66] J. Lu, K. N. Plataniotis, A. N. Venetsanopoulos, Face recognition using kernel direct discriminant analysis algorithms, IEEE Transactions on Neural Networks 14 (1) (2003) 117–126.

[67] W. M. Rand, Objective criteria for the evaluation of clustering methods, Journal of the American Statistical association 66 (336) (1971) 846–850.

[68] A. Strehl, J. Ghosh, Cluster ensembles—a knowledge reuse framework for combining multiple partitions, Journal of Machine Learning Research 3 (Dec) (2002) 583–617.

[69] A. Rosenberg, J. Hirschberg, V-measure: A conditional entropy-based external cluster evaluation measure., in: EMNLP-CoNLL, Vol. 7, 2007, pp. 410–420.

[70] E. B. Fowlkes, C. L. Mallows, A method for comparing two hierarchical clusterings, Journal of the American statistical association 78 (383) (1983) 553–569.

[71] 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, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[72] N. Passalis, A. Tefas, Entropy optimized feature-based bag-of-words representation for information retrieval, IEEE Transactions on Knowledge and Data Engineering 28 (7) (2016) 1664–1677.

[73] J. R. Hershey, Z. Chen, J. Le Roux, S. Watanabe, Deep clustering: Discriminative embeddings for segmentation and separation, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, 2016, pp. 31–35.

[74] P. Nousi, A. Tefas, Deep learning algorithms for discriminant autoencoding, Neurocomputing.