



Affinity learning via a diffusion process for subspace clustering

Qilin Li*, Wanquan Liu, Ling Li

Department of computing, Curtin University, Perth, Australia

ARTICLE INFO

Article history:

Received 14 August 2017

Revised 11 May 2018

Accepted 1 July 2018

Available online 2 July 2018

Keywords:

Subspace clustering

Diffusion process

Affinity learning

ABSTRACT

Subspace clustering refers to the problem of finding low-dimensional subspaces (clusters) for high-dimensional data. Current state-of-the-art subspace clustering methods are usually based on spectral clustering, where an affinity matrix is learned by the self-expressive model, i.e., reconstructing every data point by a linear combination of all other points while regularizing the coefficients using the ℓ_1 norm. The sparsity nature of ℓ_1 norm guarantees the subspace-preserving property (i.e., no connection between clusters) of affinity matrix under certain condition, but the connectedness property (i.e., fully connected within clusters) is less considered. In this paper, we propose a novel affinity learning method by incorporating the sparse representation and diffusion process. Instead of using sparse coefficients directly as the affinity values, we apply the ℓ_1 norm as a neighborhood selection criterion, which could capture the local manifold structure. An effective diffusion process is then deployed to spread such local information along with the global geometry of data manifold. Each pairwise affinity is augmented and re-evaluated by the context of data point pair, yielding significant enhancements of within-cluster connectivity. Extensive experiments on synthetic data and real-world data have demonstrated the effectiveness of the proposed method in comparison to other state-of-the-art methods.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

Many computer vision problems, such as image compression [1], motion segmentation [2] and face clustering [3], have to deal with high-dimensional data. The high-dimensionality of data not only results in high computational cost of time and memory for related algorithms, but more importantly, could also degrade their performances due to the inevitable noise and insufficient number of samples with respect to its feature dimension, commonly referred to as the “curse of dimensionality” problem [4]. Fortunately, even though many data are high-dimensional, their intrinsic dimension is often much lower than the dimension of the ambient space [5], which has motivated many researchers to develop various techniques for finding low-dimensional subspaces for high-dimensional datasets. A natural idea is dimensionality reduction, such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), in which a common assumption is that the data lie in only one low-dimensional subspace. However, the assumption is generally not true in practice. High-dimensional data are usually drawn from several low-dimensional subspaces corresponding to multiple categories or classes to which the data belong [6]. In such scenarios, the task of clustering a high-dimensional dataset into

multiple classes becomes a task of assigning each data point to its own subspace while preserving the underlying low-dimensional data structure, a problem known in the literature as *subspace clustering* [5].

In terms of machine learning and computer vision, existing subspace clustering methods can be categorized into four groups, including algebraic methods [7,8], iterative methods [9,10], statistical methods [11,12], and spectral clustering-based methods [6,13–16]. Among them, the spectral-clustering based methods have become extremely popular due to their theoretical foundation, and empirical success. They generally include two steps: (1) constructing an affinity matrix; and (2) applying spectral clustering to the affinity matrix. In this paper, we focus on the first step as it is of essence for the success of spectral clustering.

The most commonly used affinity learning method is the Gaussian kernel, where a global bandwidth σ has to be carefully tuned. Quite a few approaches have been proposed for learning a local adaptive σ [17–19]. The ideas are somewhat similar as stated below: for each data point, use a local σ which is defined by the distances to its k -nearest neighbours. However, when dealing with high-dimensional data, these methods are susceptible to the presence of noisy and irrelevant features [20]. Instead of using Euclidean distance based Gaussian kernel, a tree-structure affinity learning method was proposed in [21]. Considering the travel path from root node to leaf node, this method derives the affinity values that are proportional to the number of nodes travelled together

* Corresponding author.

E-mail address: li.qilin@postgrad.curtin.edu.au (Q. Li).

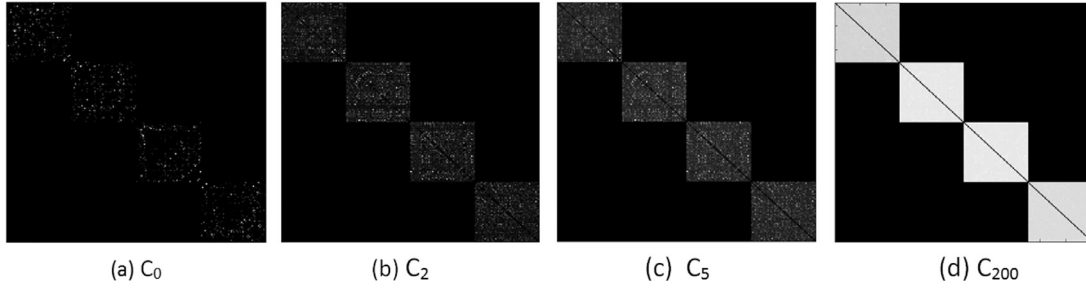


Fig. 1. Visualization of affinity matrices after different iterations of the diffusion process: (a) C_0 is the original coefficient matrix obtained by SSC(ℓ_1 norm), (b)–(d) C_t are the matrices after t steps of diffusion process applied on C_0 . Clearly, as the diffusion process goes forward, the block-diagonal structure of the affinity matrix becomes more evident.

by a pair of data points. More recently, a novel affinity learning approach based on fuzzy set was proposed in [20]. By exploiting discriminative feature subspaces, this method can capture and combine subtle similarity information distributed over feature subspaces, resulting in more accurate and robust affinity estimation.

State-of-the-art methods for constructing the affinity matrix in terms of subspace clustering are based on the *self-expressiveness property* of the data [6], i.e., each data point in a union of subspaces can be efficiently reconstructed by a linear combination of all other data points: $x_j = \sum_{i \neq j} c_{ij} x_i$, where the coefficient c_{ij} is used to define the affinity between points i and j as $w_{ij} = |c_{ij}| + |c_{ji}|$. However, this leads to an ill-posed problem with many possible solutions [5]. In order to deal with this issue, the principle of *sparsity* is employed. Specifically, every point is represented by a sparse linear combination of all other data points while regularizing its coefficient matrix. The problem can be formulated as:

$$\min_C \|C\| \quad \text{s.t.} \quad X = XC, \quad \text{diag}(C) = 0, \quad (1)$$

where $X = [x_1, \dots, x_N]$ is the data matrix, $C = [C_1, \dots, C_N]$ is the coefficient matrix, and $\|\cdot\|$ is a properly chosen norm.

Which norm is a sensible choice? In Sparse Subspace Clustering (SSC) [6], the ℓ_1 norm is used for $\|\cdot\|$ as a convex relaxation of the ℓ_0 norm to promote the sparseness of C . As pointed out by [6,22,23], even though the coefficients obtained by SSC could be subspace preserving ($c_{ij} = 0$ when x_i and x_j are not in the same subspace), its connectivity may be poor (i.e., $c_{ij} = 0$ even if x_i and x_j are in the same subspace) as the coefficient matrix is sparse. In the Low-Rank Representation (LRR) [16] and Low-Rank Subspace Clustering [24], the nuclear norm $\|\cdot\|_*$ is applied as a convex relaxation of the rank function. One benefit of the nuclear norm is that the coefficient matrix is generally dense, in which the common connectivity issue for sparse representation-based methods is alleviated. However, such representation matrix is known to be subspace preserving only when the subspaces are independent [25], which significantly limits its applicability since such condition is not satisfied generally.

In this paper, the ℓ_1 norm is still used for sparsity constraint in order to best retain the subspace preserving property, but a diffusion process is adopted for deriving the subspace of affinity graph to mitigate the connectedness problem. Specifically, the first step is to learn a sparse affinity matrix by applying the ℓ_1 norm on the optimization problem of (1). A diffusion process is then adopted to spread the affinity values through the entire graph built upon the affinity matrix. Such a process can be interpreted as a Markov random walk, where the stochastic affinity (transition) matrix determines the probabilities of walking from one node to another. Since the affinity matrix learned by ℓ_1 norm is subspace preserving, the random walk on this matrix is guaranteed to be subspace constrained. Therefore, the connectivity within subspace is significantly enhanced while the subspace preserving property remains unaltered. An illustrative example is given in Fig. 1. It clearly

demonstrates that based on the sparse affinity matrix obtained by ℓ_1 norm, the proposed method could evidently improve the affinity within subspaces while retain the sparsity between subspaces, yielding an affinity matrix with exactly block-diagonal structure.

The paper is organized as following. In Section 2, we briefly discuss the general model of subspace clustering and several state-of-the-art diffusion strategies. In Section 3, we present and justify the proposed method. The interpretation of our method from graph and random walk view is also given. In Section 4, we compare the performance of the proposed method with other state-of-the-art methods through experiments on both synthetic and real data. The conclusion is presented in Section 5.

2. Related work

2.1. Subspace clustering

State-of-the-art subspace clustering methods are all built on the self-expressiveness property. The choice of the regularizers for the coefficient matrix is the main difference among these methods. While different regularizers have their own advantages and drawbacks [25–27] propose to use mixed norms. The low-rank sparse subspace clustering (LRSSC) method [26] combines the ℓ_1 and nuclear norm regularizer as:

$$\|C\|_* + \lambda \|C\|_1, \quad (2)$$

where λ plays the trade-off between the two regularizers. Likewise, a mixture of ℓ_1 and ℓ_2 norms is proposed in [25,27] as:

$$\lambda \|C\|_1 + \frac{1-\lambda}{2} \|C\|_2, \quad (3)$$

where the value λ represents the trade-off between the two norms. These methods basically attempt to bridge the gap between the subspace preserving and connectedness property by controlling the trade-off between different norms.

Feng et al. [28] proposes to explicitly impose a block-diagonal constraint by fixing the rank of Laplacian matrix. One benefit of this approach is that it can be applied to all the affinity construction methods directly. The Structured Sparse Subspace Clustering (SSSC) [29] integrates the two stages, affinity learning and spectral clustering, into one unified optimization framework. Their observation is that the clustering results can help the self-expressiveness model to yield a better affinity matrix.

All these subspace clustering methods are based on spectral clustering, where the affinity matrix is of great importance. Regardless of the various norms used by existing methods, the affinity values are often learned independently. Without considering the context information when the pairwise affinity is obtained, it is less likely to capture the underlying manifold. We hence propose to adopt a diffusion process to iteratively make full use of the local neighborhood structure. Each pairwise affinity is augmented

and re-evaluated by their affinities with others, yielding an affinity matrix which can better represent the true geometry of the data manifold.

2.2. Diffusion process

The diffusion process is widely used in the field of information retrieval [30–34], where the goal is to retrieve the most similar instances with respect to certain query samples from a potentially large database. Conventional approaches are usually based on calculating pairwise affinity/distance values which are directly used to rank the most similar elements. One evident limitation of such approaches is that the structure of the underlying data manifold is entirely neglected. To tackle this, instead of considering pairwise affinity individually, a diffusion process can be adopted to derive context sensitive measures, and it has become an indispensable tool for improving retrieval performance [34].

Diffusion process has also been used for image segmentation as well [33,35], where the task is formulated as a clustering problem using the image pixels or image patches as the data points. The diffusion process could learn context-aware affinity by spreading the affinity following the data manifold. By combining the diffusion process with standard image segmentation method, e.g., Normalized Cut (NCut) [36], the combined algorithm not only significantly outperforms the NCut with the original affinities, but also outperforms many state-of-the-art image segmentation methods [35].

There is also work using the diffusion process in the scenario of semi-supervised learning [37], where the diffusion is applied on the class labels. The keynote is to let every data points iteratively spread its label information to its neighbors until a global stable state is achieved. The same idea is adopted by [38] for data denoising, where a linear relationship can be derived between class labels before and after the diffusion process, and then be used to calculate the coherence error between data before and after denoising. However, as claimed by the authors, they do not explicitly perform any diffusion, rather using a closed form solution of diffusion from [37] to reconstruct a virtual label indicator.

To obtain a $N \times N$ affinity matrix A using diffusion, an undirected graph $G = (V, E)$ is first constructed, consisting of N nodes $v_i \in V$, and edges $e_{ij} \in E$ that connect pairs of nodes. The affinity values a_{ij} are used to weight the corresponding edges. Then the diffusion process can be interpreted as a Markov random walk on G , where the edge weights e_{ij} determine the probabilities of walking from node i to node j . To illustrate this clearly, we first define the transition matrix of a random walk as:

$$P = D^{-1}A, \quad (4)$$

where D is a diagonal matrix with $D_{ii} = \sum_{k=1}^n A(i, k)$. Clearly, P is a row-stochastic transition matrix corresponding to the random walk on the graph G . With a simple update rule $A_{t+1} = A_t P$, the affinity matrix after t steps of random walks can be obtained by

$$A_t = AP^t. \quad (5)$$

This model was adopted by another well-known retrieval algorithm, the Google PageRank system [39], in which the standard random walk is modified in a way that at each time step t , there is a small probability $1 - \alpha$ to jump onto an arbitrary node, whereas the probability of the standard operation of walking to the neighbouring nodes is α . This leads to the following update strategy:

$$A_{t+1} = \alpha A_t P + (1 - \alpha)Y, \quad (6)$$

where Y defines the possible choice of random jump, which enables personalization for individual web user. A similar method was proposed in [40], namely the Ranking on Manifolds. The difference is in the slightly adapted transition matrix $P = D^{-1/2}AD^{-1/2}$.

One drawback of the diffusion process mentioned above is that the process is applied to the entire graph, in which the noisy edges could heavily impact its performance. Current state-of-the-art methods [35] restricted the diffusion process to the K nearest neighbor (KNN) graph. Given affinity matrix A , this method sets $A_{ij} = A_{ij}$ only if $x_j \in KNN(x_i)$, otherwise $A_{ij} = 0$, and results in a new update strategy as:

$$A_{t+1} = PA_t P^T. \quad (7)$$

While this approach consistently yields the state-of-the-art performance in the field of diffusion, it is observed that in practice the neighborhood number K needs to be set manually and the performance is sensitive to the choice of K [34].

Paper contributions. In this paper, we exploit the diffusion process to mitigate the connectedness problem of ℓ_1 norm in terms of subspace clustering. We show that by combining the ℓ_1 norm and diffusion process, the learned affinity matrix can be extremely effective when used in the scenario of subspace clustering. To the best of our knowledge, this is the first attempt to adopt the idea of diffusion process into this field. Since the idea of using ℓ_1 norm for subspace clustering is originally from Sparse Subspace Clustering (SSC), we name our method Diffusion-based Sparse Subspace Clustering (DSSC). Our main contributions can be summarized as:

1. For subspace clustering, instead of combining and balancing different norms for subspace preserving and connectivity property, the ℓ_1 norm is used in the proposed approach and the corresponding connectivity problem is alleviated by an efficient diffusion process. We argue that instead of using the sparse coefficients directly as the affinity values, the ℓ_1 norm serves more like a neighborhood selection in the DSSC, while the affinity learning is managed by the diffusion process.
2. From the diffusion point of view, we show that instead of choosing K nearest neighbors in the Euclidean space, the sparse property of ℓ_1 norm provides manifold-aware neighborhood construction, which is locally constrained in the corresponding subspaces. Applying the diffusion process on the affinity matrix obtained by ℓ_1 norm yields more appropriate estimation of the true geodesic distances between data points on the manifold.

It should be noted that ℓ_1 affinity matrix and KNN affinity matrix are both sparse. However, to construct the neighborhood structure, ℓ_1 is much more flexible. The reason is that the number of non-zero elements of each column of ℓ_1 affinity matrix is variational, which is controlled by the optimization method, while in KNN affinity matrix, they are all fixed to K . Clearly, it is often inappropriate to set a uniform K for all data points, as they are often not uniformly distributed. Dense areas need a larger K to cover the local structure, while sparse areas need a smaller K .

3. Diffusion based sparse subspace clustering

Before introducing the proposed approach, we first formulate the problem to be addressed in this paper.

Problem 1. Given a set of data points $\{x_j \in \mathbb{R}^D\}_{j=1}^N$ drawn from an unknown union of k subspaces $\{S_i\}_{i=1}^k$ of unknown dimensions $d_i = \dim(S_i)$, $0 < d_i < D$, $i = 1, \dots, k$, the goal is to cluster these points into their corresponding subspaces.

Sparse Subspace Clustering (SSC) attempts to solve the above problem based on the so-called self-expressiveness model, which states that each data point can be expressed as a linear combination of all other data points, i.e., $X = XC + E$, where C is the coefficient matrix and E is the matrix of error (noises or outliers). In order to guarantee the uniqueness of the solution, the sparsity constraint on C is invoked by ℓ_1 minimization, which leads to the

optimization problem:

$$\min_C \|C\|_1 + \lambda \|E\|_\ell \quad \text{s.t.} \quad X = XC + E, \quad \text{diag}(C) = 0, \quad (8)$$

where $\|\cdot\|_\ell$ represents the Frobenius norm or ℓ_1 norm to handle noise or outliers. With any ℓ_1 solver, e.g., ADMM, an optimal coefficient C can be obtained by solving the optimization problem (8), which is used to build a symmetric affinity matrix $|C| + |C^T|$ later on. The final data segmentation is obtained by applying spectral clustering using the affinity matrix.

As mentioned previously, the key difference among existing subspace clustering methods lies on the choice of the regularizer. Clearly, ℓ_1 norm of the coefficient matrix C is the core of SSC. It gives an optimal unique solution of C with a sensible property, sparsity. While the reconstruction of a single data point is forced to use as few coefficients as possible, it is believed that the data points located in the same subspace should be used as they are much more similar. Nevertheless, the sparsity property could cause some problems. In the case of clean data, SSC is guaranteed to be subspace-preserving, i.e., there are no connections between points from different subspaces. However, the within-subspace connections are usually sparse, i.e., C_{ij} could be zero even when x_i and x_j are in the same subspace. It is not a problem as long as the connections are still subspace-preserving. However in the case of noisy data, there is no theoretical guarantee that the nonzero coefficients correspond to points in the same subspace. It is then possible for SSC to make an inappropriate segmentation of the data since the within-subspace connections are weak while there are noisy connections between subspaces.

We argue that such a drawback exists in all SSC based algorithms as SSC seeks the sparse coefficients of each data point independently. The diffusion process is capable of dealing with such an issue by exploiting the contextual affinities. Given an affinity matrix $W = |C| + |C^T|$, where C is obtained by solving (8), we first normalize the row of W by $W = D^{-1}W$, where D is a diagonal matrix with $D_{ii} = \sum_{k=1}^n W(i, k)$, then classic diffusion process can be encoded into computing the power of the affinity matrix, which is

$$W_t = W^t, \quad (9)$$

where t corresponds to t steps of the diffusion process. It is obvious that such a process is sensitive to the step t . In order to make the diffusion process robust in terms of t , we can consider the accumulation of all t s. Thus, the diffusion process is reformulated as below:

$$W_t = \sum_{i=0}^t W^i, \quad (10)$$

where W is assumed to be nonnegative and the sum of each row is smaller than 1. A stochastic matrix can be used to construct the matrix that satisfies the requirements. Note that the absolute eigenvalues are bounded by the largest row sum. Therefore, the maximum eigenvalues of W is less than 1, resulting in a converged closed form given by $\lim_{t \rightarrow \infty} W_t = (I - W)^{-1}$ of (10), where I is the identity matrix.

In a graph representation, the relationship between samples are normally represented by pairwise affinities. Such naive pairwise representation is too simple to reveal the complex real world data structure, especially when the data lie on a intricate manifold, and squeezing the complex relationships into pairwise ones will certainly result in information loss. A natural way of remedying this issue is to represent the data as a higher order tensor graph, where the edge can connect more than two vertices. As demonstrated by several works [35,41,42], the higher order tensor graph is indeed helpful for revealing the intrinsic relationships among data points and leads to better performance. Given a graph $G = (V, E)$ constructed from an affinity matrix W , the tensor product graph

is defined as the Kronecker product of the original graph with itself, $\mathbb{G} = G \otimes G$. The corresponding affinity matrix is $\mathbb{W} = W \otimes W$. In particular, we have

$$\mathbb{W}(\alpha, \beta, i, j) = W(\alpha, \beta) \cdot W(i, j) = w_{\alpha, \beta} \cdot w_{i, j}. \quad (11)$$

Thus, if $W \in \mathcal{R}^{n \times n}$, $\mathbb{W} = W \otimes W \in \mathcal{R}^{nn \times nn}$. The diffusion process is then defined on the higher order tensor as

$$\mathbb{W}_t = \sum_{i=0}^t \mathbb{W}^i. \quad (12)$$

With the assumption that the maximum row sum of W is less than 1, we have

$$\sum_{\beta, j} \mathbb{W}(\alpha, \beta, i, j) = \sum_{\beta, j} w_{\alpha, \beta} w_{i, j} = \sum_{\beta} w_{\alpha, \beta} \sum_j w_{i, j} < 1. \quad (13)$$

Similar to (10), the process (12) also has a closed form solution,

$$\lim_{t \rightarrow \infty} \mathbb{W}_t = \lim_{t \rightarrow \infty} \sum_{i=0}^t \mathbb{W}^i = (I - \mathbb{W})^{-1}. \quad (14)$$

Since our goal is to learn a new affinity matrix W^* of size $n \times n$, it can be defined as

$$W^* = \text{vec}^{-1}((I - \mathbb{W})^{-1} \text{vec}(I)), \quad (15)$$

where I is the identity matrix and vec is an operation that transfers a matrix to a vector by stacking all the columns one after another. The inverse of vec is denoted as vec^{-1} .

While the tensor graph provides an adequate underlying structure of the data, it is impractical for large scale problems due to the high storage and computing cost. Therefore, we use an iterative algorithm for the diffusion process on tensor graph. Firstly, we define $A_0 = W$ and the update strategy as

$$A_{t+1} = WA_t W^T + I, \quad (16)$$

where I is the identity matrix. The diffusion process becomes the iteration of (16) until convergence. To prove the convergence of (16), we transform (16) to

$$\begin{aligned} A_{t+1} &= WA_t W^T + I = W(WA_{t-1} W^T + I)W^T + I \\ &= W^2 A_{t-1} (W^T)^2 + W I W^T + I = \dots \\ &= W^t W (W^T)^t + W^{t-1} I (W^T)^{t-1} + \dots + I \\ &= W^t W (W^T)^t + \sum_{i=0}^{t-1} W^i I (W^T)^i. \end{aligned} \quad (17)$$

Since the sum of each row is assumed to be less than 1, $W^t = \mathbf{0}$, therefore we have $\lim_{t \rightarrow \infty} W^t W (W^T)^t = \mathbf{0}$. Consequently,

$$\lim_{t \rightarrow \infty} A_{t+1} = \lim_{t \rightarrow \infty} \sum_{i=0}^{t-1} W^i I (W^T)^i. \quad (18)$$

As shown in [35], it can be proven by induction that

$$\lim_{t \rightarrow \infty} \sum_{i=0}^{t-1} W^i I (W^T)^i = \text{vec}^{-1}((I - \mathbb{W})^{-1} \text{vec}(I)). \quad (19)$$

Hence we have

$$\lim_{t \rightarrow \infty} A_{t+1} = \text{vec}^{-1}((I - \mathbb{W})^{-1} \text{vec}(I)). \quad (20)$$

The iterative algorithm (16) produces the same affinity matrix as the diffusion process on tensor graph. Once the final affinity matrix is obtained, the clustering result can be achieved by performing spectral clustering as described in Algorithm 1.

Algorithm 1 Diffusion based Sparse Subspace Clustering (DSSC).

Input: A set of data points $X^{n \times d}$, the regularization parameter λ , the number of clusters k .

- 1: Solve the ℓ_1 -minimization problem 8 to get the coefficients c_i .
- 2: Form an affinity matrix $W = |C| + |C|^T$, where $C = [c_1, c_2, \dots, c_n]$.
- 3: Normalize the row of W by $W = D^{-1}W$, where D is a diagonal matrix with $D_{ii} = \sum_{k=1}^n W(i, k)$.
- 4: Initialize the iteration $A_0 = W$, $t = 0$
- 5: **while** not converged **do**
- 6: $A_{t+1} \leftarrow WA_t W^T + I$
- 7: $t \leftarrow t + 1$
- 8: **end while**
- 9: Construct the corresponding Graph Laplacian matrix $L = I - D^{-1/2}AD^{-1/2}$, where D is a diagonal matrix with $D_{ii} = \sum_{k=1}^n A(i, k)$.
- 10: Form the matrix $V^{n \times k}$ by stacking the first k normalized eigenvectors of L corresponding to its k smallest eigenvalues.
- 11: Treat each row of V as a data point, and perform k -means to get the cluster labels.

Output: The cluster labels Y .

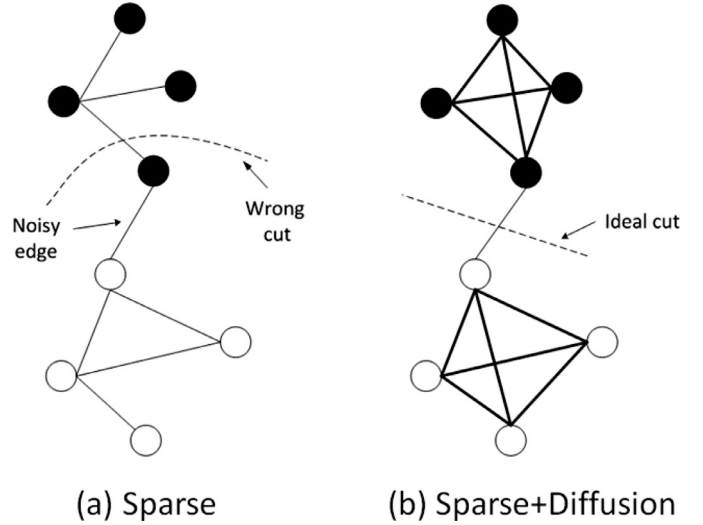


Fig. 3. An example of affinity graph for noisy data, obtained by different methods: (a) sparse, (b) sparse with diffusion. By enhancing the within subspace connections, the diffusion process could help to find the ideal cut of the affinity graph. The value of edge weights are reflected by the thickness of the edges.

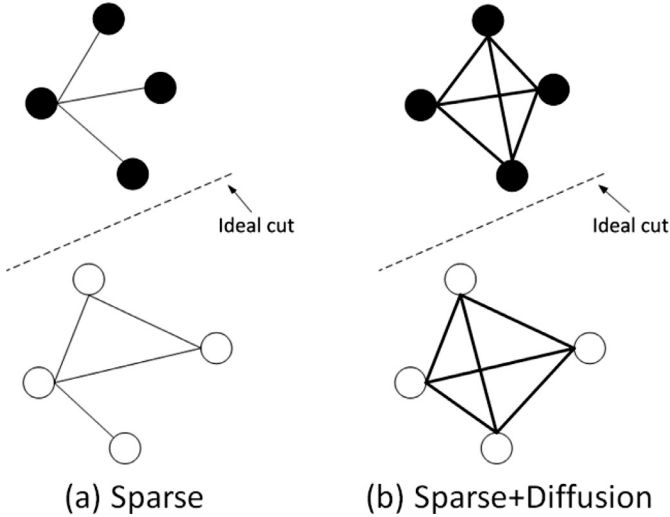


Fig. 2. An example of affinity graph for clean data, obtained by different methods: (a) sparse, (b) sparse with diffusion. While the cuts remain the same, the within subspace connections are clearly enhanced after the diffusion process. The value of edge weights are reflected by the thickness of the edges.

3.1. A graph view of the diffusion process

With ℓ_1 minimization, SSC seeks sparse representation for each data point individually. The corresponding affinity graph reveals the pairwise affinity as the “shortest path” between them, which is susceptible to noise. The proposed DSSC derives the affinity by considering the “volume of paths” through a diffusion process. One significant property is, with ℓ_1 norm, the “volumes of paths” are restricted in subspaces, resulting in the enhancement of within-subspace affinity and the preservation of between-subspace affinity.

Fig. 2 shows that in the case of clean data, SSC is guaranteed to be subspace-preserving, i.e., there are no connections between subspaces. In such situations, spectral clustering can properly cut the graph. However, due to the sparse property of ℓ_1 norm, it is very likely that two points in the same subspace are not connected. A diffusion process can accurately complete the graph

for each subspace in such cases, leading to more robust subspace graphs.

In the case of noisy data, the advantage of diffusion becomes even more significant. As shown in Fig. 3, when there are noisy edges between subspaces, spectral clustering might not be able to find the ideal cut of the graph. As the pairwise affinity is individually established in SSC, it can be difficult to distinguish “noisy edges” from “real edges”. With DSSC, due to the ability to combine contextual information, the within-subspace affinities are evidently enhanced so that the noisy edges can be properly severed by spectral clustering. Real world data are usually noisy, which explains the significant improvements of DSSC over SSC on real world data sets, shown in the experiments.

3.2. A random walk view of the diffusion process

As derived by [43], the objective function of spectral clustering can be expressed in the framework of random walk as follows. For two disjoint subsets $A, \bar{A} \subset V$, assume that we run a random walk starting with X_0 in the stationary distribution π , we have

$$P(\bar{A}|A) = P(X_1 \in \bar{A} | X_0 \in A) \quad (21)$$

as the probability of the random walk transition from set A to set $\bar{A}$. Then, the goal of spectral clustering is to minimize the objective function

$$F = P(\bar{A}|A) + P(A|\bar{A}). \quad (22)$$

As mentioned previously, the proposed diffusion process can be interpreted as Markov random walks [44], where the stochastic affinity matrix is the transition matrix that defines probabilities for walking from one node to another. One crucial observation is that for an affinity graph, there will be many connections within a subspace, and fewer connections between subspaces. Therefore, if we start a walker at one node and randomly travel to a connected node, the random walk is more likely to stay within the same subspace than travel between different subspaces. Intuitively, diffusion processes can be imagined as random walks starting at every node on the graph. As the random walks continue, the probabilities of traveling between nodes within the same subspace increase, while the probabilities of traveling between subspaces decrease, which means $P_t(\bar{A}|A) < P(\bar{A}|A)$ as well as $P_t(A|\bar{A}) < P(A|\bar{A})$, where P_t is the transition probabilities after t steps of diffusion process. Let F^* be

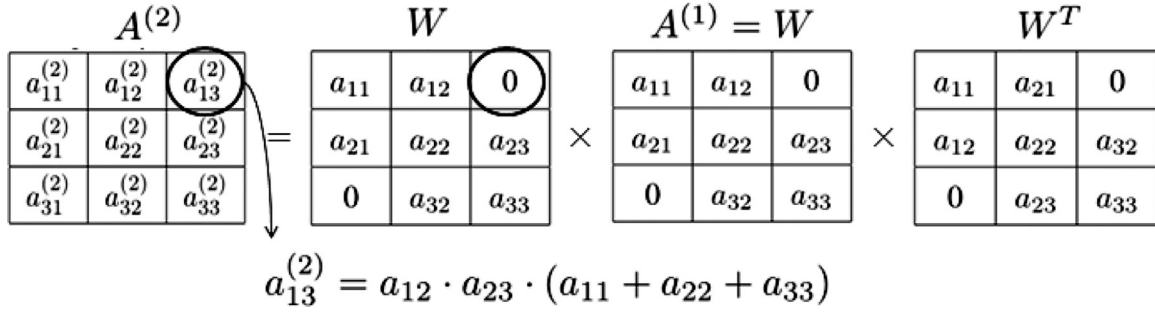


Fig. 4. Example of showing how the diffusion process could improve the affinity matrix by leveraging the context information. $a_{13}^{(1)} = 0$, after one iteration $a_{13}^{(2)} = a_{12} \cdot a_{23} \cdot (a_{11} + a_{22} + a_{33})$.

the objective of spectral clustering applied on the affinity matrix obtained by the diffusion process. In conjunction with Eq. (22), we have

$$F^* = P_t(\bar{A}|A) + P_t(A|\bar{A}) < P(\bar{A}|A) + P(A|\bar{A}) = F, \quad (23)$$

which explains why the diffusion processes can improve the performance of spectral clustering.

4. Experiments

The experiments are divided into three parts. In Section 4.1, we compare the proposed DSSC to some other state-of-the-art affinity learning methods. As the proposed DSSC method uses ideas from both the diffusion and subspace clustering fields, it is necessary to examine whether DSSC is superior to the state-of-the-art methods in both fields. In Section 4.2, we compare DSSC with a state-of-the-art diffusion method. In Section 4.3, DSSC is compared with quite a few subspace clustering methods. The clustering results are evaluated using several performance metric: clustering accuracy, clustering error and Normalized Mutual Information (NMI).

Experimental settings. For the experiments of Hopkins 155 and Extended YaleB, the regularization parameter λ is set to be the same as they are in the SSC [6] for fair comparison. For all other experiments, the λ in DSSC and all other parameters in other methods are chosen by grid search, and those given the best results are used. The iteration step t in DSSC is set to be 200 for all experiments, which is found to be large enough to converge. Note that to speed up the process, one could also set up a convergence condition, such as $\|A_{t+1} - A_t\| < \epsilon$, where $\epsilon = 1e-4$, and check this condition in every iteration. The number of classes k is set to be the true number of classes for all methods.

Toy example. We first use a toy example to show how the proposed method could improve the affinity matrix by taking the context information into account. Suppose we have an affinity matrix W_{33} that contains affinity values of three data points, as shown in Fig. 4 (a_{ij} represents the affinity value between i and j). For simplicity, we show the affinity value between 1 and 3, a_{13} , after one iteration. According to the updating strategy $A_{t+1} = WA_tW^T + I$, we have

$$a_{13}^{(2)} = a_{12} \cdot a_{23} \cdot (a_{11} + a_{22} + a_{33}). \quad (24)$$

It can be seen that initially $a_{13} = 0$, but after one step of the diffusion process, it starts to re-evaluate the affinity value by considering the context data (a_{12} and a_{23}). Intuitively, we know if A is similar to B and B is similar to C , then A should be somewhat similar to C , i.e., a_{13} should not be 0 when $a_{12}, a_{23} > 0$. From this toy example, it can be clearly seen that, the diffusion process can augment the original affinity values by spreading the context affinities. Instead of considering each pair of data points individually, the affinity values obtained by the proposed method could leverage all affinity information of neighbor data points, yielding more

precise affinity values. We stress its effectiveness when combined with ℓ_1 norm. Due to its sparsity constraint, ℓ_1 norm will produce many 0 affinities within subspace, resulting in poor clustering performance. By integrating the diffusion process, data pairs will share their neighbors and update their affinities iteratively, leading to more compact within-subspace affinities.

4.1. Comparison with state-of-the-art affinity learning methods

The proposed algorithm DSSC, at its core, is an affinity learning method, where the affinity value is initialized by ℓ_1 -norm and updated by the diffusion process. To evaluate the effectiveness of DSSC on affinity learning, we compared it to several other affinity learning methods:

- **SC.** A widely used NJW spectral clustering [45], where the affinity is obtained by the Gaussian kernel with global bandwidth sigma.
- **STSC.** Self-tuning spectral clustering [17]. The affinity is obtained by the Gaussian kernel with adaptive sigma, which is defined by the KNN distance.
- **RFSC.** Random forest based spectral clustering [21], is a tree-structure affinity inference model, where the affinity value is proportional to the number of nodes traveled together by a pair of data points, starting from the root node to the leaf node.
- **AFSSC.** Fuzzy based affinity learning method [20], in which the affinity is defined by data-driven fuzzy membership function. Instead of using all features in the ambient space, this model can capture and combine subtle proximity distributed in discriminative feature subspaces.

While the affinity matrices are learned by different algorithms, the results are obtained by applying the same NJW spectral clustering on them. The parameters in each method are chosen by grid search, as the same in [20].

Following the same settings, we redo two experiments in [20]. The first one is digits clustering on the widely used USPS dataset [46]. This dataset consists of numeric data sampled from the scanning of handwritten digits on the envelopes from the U.S. Postal Service. There are in total 9298 images, and each digit image is normalized to 16×16 grayscale, resulting in a feature vector with 256 dimensions. To fully investigate the performance with respect to the number of clusters and data points, digit subsets {0,8}, {4,9}, {0,5,8}, {3,5,8}, {1,2,3,4}, {0,2,4,6,7} and the full set {0–9} are used to test the algorithms.

Table 1 shows the clustering accuracies of different methods on the USPS datasets. It can be seen that the proposed DSSC outperforms all other competitors in all test cases. While the other methods only perform well only on the easy cases, i.e., {0,8} and {1,2,3,4}, DSSC achieves satisfactory results in all cases. Particularly, in the hard cases, significant accuracy improvements are observed, 14% and 12% in the case of {4,9} and {3,5,8}, respectively.

Table 1
Clustering accuracy on the USPS datasets.

Dataset	{0,8}	{4,9}	{0,5,8}	{3,5,8}	{1,2,3,4}	{0,2,4,6,7}	{0 ~ 9}
SC	0.966	0.756	0.829	0.713	0.937	0.713	0.538
STSC	0.843	0.776	0.724	0.604	0.921	0.622	0.624
RFSC	0.872	0.738	0.814	0.767	0.945	0.737	0.698
AFSSC	0.985	0.781	0.902	0.833	0.981	0.805	0.722
DSSC	0.991	0.922	0.975	0.950	0.987	0.979	0.850

Table 2
Clustering NMI on the USPS datasets.

Dataset	{0,8}	{4,9}	{0,5,8}	{3,5,8}	{1,2,3,4}	{0,2,4,6,7}	{0 ~ 9}
SC	0.752	0.110	0.653	0.352	0.597	0.706	0.515
STSC	0.430	0.248	0.424	0.253	0.632	0.609	0.578
RFSC	0.649	0.235	0.607	0.474	0.854	0.782	0.652
AFSSC	0.912	0.268	0.682	0.496	0.920	0.834	0.775
DSSC	0.920	0.607	0.877	0.797	0.946	0.924	0.824



Fig. 5. Example images from Yale datasets. Each row corresponds to one subject.

Table 2 shows the NMI of different methods on the USPS datasets. The results are consistent with clustering accuracy that DSSC outperforms all other methods in all test cases, and especially in the hard cases and multi-cluster cases with a large number of clusters, DSSC wins with a large margin compared to the second best.

Next, to fully evaluate the robustness of the affinity matrices, we test the clustering performances using the KNN affinity matrix, i.e., given an affinity matrix W , we set $W_{ij} = W_{ij}$ if $j \in \text{knn}(i)$, otherwise, $W_{ij} = 0$. In the scope of spectral clustering, even though there is no theoretical result showing that which type of affinity matrix is better, it has been reported in [21,47] that in practice KNN affinity matrix is more effective than the full affinity matrix, especially for noisy data. In our experiment, we use face images from Yale [48] and CMU-PIE [49]. The Yale dataset contains 165 grayscale images of 15 subjects, and each subject has 11 images of frontal face with various lighting conditions and minor expressions. CMU-PIE contains 41,368 images of 68 people, each with 13 different poses, 43 different illumination conditions, and 4 different expressions. We use all the frontal faces of the 68 people with no expression or pose, resulting in 20 images per person. Example images are shown in Figs. 5 and 6. All the images are cropped and down-sampled to 32×32 resolution. The raw pixel values are used as features.

Fig. 7 presents the clustering performances NMI with respect to different number of neighbors in the affinity matrices. It can be

seen that the proposed DSSC achieves the best results for all values of neighbourhood size k . It is worth to mention that DSSC achieves significant improvements in the CMU-PIE dataset. One possible reason is that the face images of PIE used are purely frontal face without any poses or expressions (as shown in Fig. 6), resulting in the satisfaction of our assumption that face images of a single subject lie in a linear subspace. It also can be observed that as k increases, DSSC produces more robust results against other methods. This is very important in real applications where the number of samples in each class is unknown, or the numbers of samples are different for each class. Note that we observe the best clustering results can be obtained with a very small value of k (typically $k < 10$). This observation agrees with [21,47]: in practice KNN affinity matrix is more effective than the full affinity matrix ($k = N$) for noisy data, which are face images in our case.

4.2. Comparison with the state-of-the-art diffusion method

Although the diffusion process has been used for clustering problem [33,35], it is not clear how it will perform on the subspace clustering, where the data is often of high dimension while the intrinsic dimension is much smaller. To show the effectiveness of the proposed DSSC, we compare it to:

- **Gaussian.** Gaussian affinity based on Euclidean distance. The bandwidth sigma is locally tuned as in [50], where it is determined by the neighbor distance.



Fig. 6. Example images from CMU-PIE datasets. Each row corresponds to one subject.

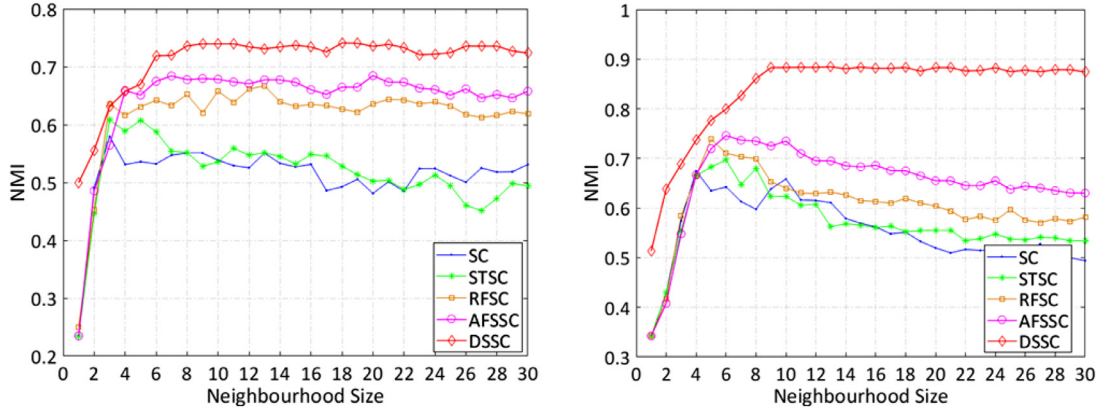


Fig. 7. Clustering performance (NMI curve) with respect to the neighbourhood size of affinity matrix. Left: Yale. Right: CMU-PIE.

- **TPG.** Affinity learned on Tensor Product Graph [35], the state-of-the-art diffusion criteria. It starts with a Gaussian affinity matrix sparsified by k-nearest neighbor.
- **SSC.** Sparse Subspace Clustering [6]. Affinity is learned with ℓ_1 -norm minimization.

The clustering results are obtained by applying spectral clustering on the affinity matrices learned by these methods on synthetic data and real data.

Synthetic data. We construct 5 subspaces $\{S_i\}_{i=1}^5 \subset \mathbb{R}^{100}$ whose bases $\{U_i\}_{i=1}^5$ are obtained by $U_i = T_i U$, $1 \leq i \leq 5$, where U is a random matrix of dimension 100×5 and $T_i^{100 \times 100}$ is a random rotation. We sample 50 data points from each subspace by $X_i = U_i Q_i$, where the entries of $Q_i \in \mathbb{R}^{5 \times 50}$ are independent and identically distributed samples from a standard Gaussian. To further evaluate the robustness of these methods, certain percentage of data samples x are randomly chosen to be corrupted by Gaussian noise with zero mean and variance $0.2\|x\|$.

Fig. 8 demonstrates the clustering performances of these methods with respect to different ratio of corruption on the synthetic data. It can be seen that the proposed DSSC outperforms other methods on all cases. Gaussian affinity is obviously not working on such a subspace clustering problem, as it achieves only 50% accuracy even with no corruption. Clearly, based on Euclidean distance, Gaussian affinity cannot capture the geometry of the data mani-

fold, which in this case is the low-dimensional subspaces that lie in the high-dimensional ambient space. SSC performs quite well when the corruption ratio is low, which implies that the sparse representation based on ℓ_1 -norm is able to reveal the manifold structure of noise-free data. For each data point, the other data points invoked by the sparse reconstruction are indeed its “true” neighbors on the manifold. However, as the noise level increases, these subspaces are not mutually independent anymore, resulting in inadequate reconstruction coefficients (affinities), which explains the dramatic degradation of the performance.

TPG and the proposed DSSC are two methods that involve the diffusion process. Their crucial advantage is that instead of considering pairwise measurement, the diffusion process will re-evaluate the affinity using the context information. As TPG starts with Gaussian affinity, the performance gain of TPG over Gaussian clearly shows that the diffusion process is able to appropriately estimate the geodesic distance on the data manifold. However when noise percentage is high, the performance of TPG decreases significantly. One possible reason is that TPG starts with a KNN version of Gaussian affinity ($W_{ij} \neq 0$ if and only if $x_j \in \text{knn}(x_i)$). When the noise level is low, the data manifold is closer to a convex subset of the feature space, where the Euclidean distances of local neighbors are still a sensible estimation of the geodesic distances. As the noise level increases, the data manifold becomes arbitrary and of-

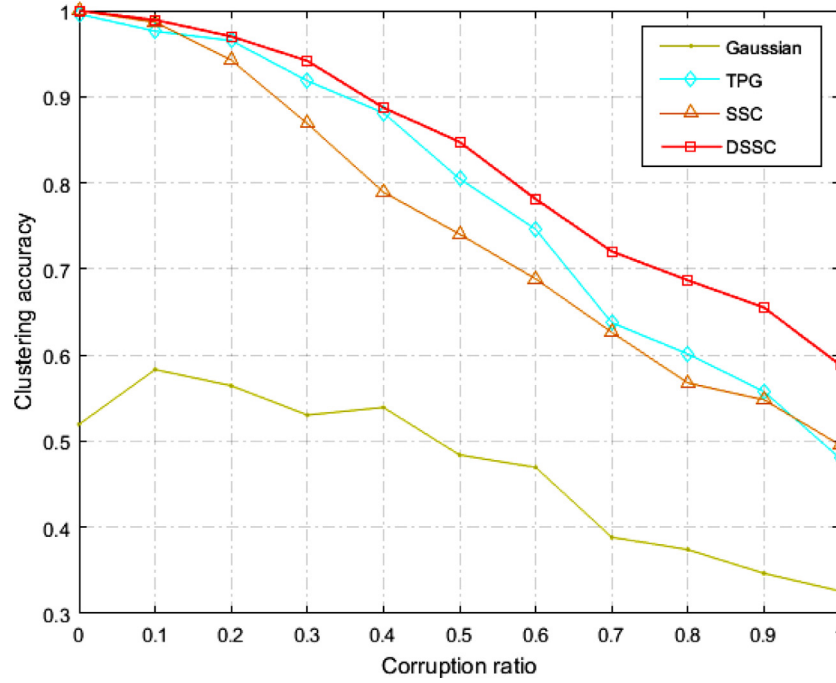


Fig. 8. Clustering performance with respect to different ratio of corruption on the synthetic data.

ten curved. In such cases, the Euclidean distances of neighbors can no longer provide good approximations of the geodesic distances.

DSSC consistently performs the best on this synthetic dataset. The success comes from two folds: firstly, DSSC starts with affinity matrix learned with ℓ_1 -norm. Such sparse representation could effectively reveal the true geometry of the data manifold, compared to the Gaussian affinity based on Euclidean distance. Secondly, all the pairwise affinities are augmented by the context of each pair, where the context is defined by the previous sparse representation. As we can see, in the proposed DSSC method, ℓ_1 -norm serves more like a neighborhood selection due to its sparse property. Local manifold is revealed by the sparse coefficients, and then the diffusion process combines such local manifolds to a complete global manifold by propagating the neighbor affinities following the geometry of the data structure. Using such a coarse-to-fine process, DSSC could appropriately estimate subspaces and achieve better clustering performance.

Face images. Given face images of different subjects obtained under various illumination conditions and a fixed pose, we consider the problem of clustering the images with respect to their subjects. Under the Lambertian assumption, it has been shown that the face images of a single subject with varying illumination and fixed pose lie in a linear subspace of dimension 9 [51]. Thus, face clustering can also be considered as a subspace clustering problem, where each subject lies in a 9D subspace.

The face image datasets used here are ORL and CMU-PIE [49]. ORL contains 400 grayscale face images of 40 subjects, each with 10 images of near frontal poses, various lighting conditions and slight facial expressions.

To demonstrate the effectiveness of DSSC, we first evaluate the affinity matrices obtained by different methods, which could qualitatively reflect the performance of affinity learning. Fig. 9 shows the visualization of these affinity matrices, obtained directly by applying MATLAB function *imagesc()*. The ground truth of affinity matrix should have block-diagonal structure, as all data samples are organized so that samples in the same cluster stay together.

Table 3
Clustering accuracy on face images.

Methods	Gaussian	TPG	SSC	DSSC
ORL	0.60	0.68	0.72	0.82
PIE	0.30	0.48	0.75	0.95

It can be seen that the affinity matrices obtained by DSSC show much more clear block-diagonal structure and more sparse off-diagonal parts (between-class affinities), compared to the other methods. Specifically, in the case of ORL, due to the existence of noise produced by face expressions and minor face poses, the main concern is the between-class affinity. Comparing the affinity matrices of SSC and DSSC, it clearly shows that DSSC could significantly reduce the between-class affinity. In the case of CMU-PIE, the within-class affinity is the concern. SSC produces highly sparse affinity matrix by ℓ_1 -norm, which causes low affinity values even within the same class. By propagating the local context affinities, DSSC successfully re-evaluates the affinity values and produces nearly perfect block-diagonal structure. The affinity matrices of Gaussian and TPG are clearly worse as they are all based on Euclidean distance. These face images lie in high-dimensional feature space (1024D) while their intrinsic dimension is very low (9D), hence the Euclidean distance cannot represent the geodesic distance.

Table 3 shows the clustering performances obtained by applying spectral clustering on these affinity matrices. It can be seen that clustering performance coincides well with the quality of the affinity matrix. Gaussian and TPG, based on Gaussian affinity, achieve unsatisfactory results on both ORL and PIE datasets. While SSC produces reasonable results on these face images, clustering performances have been boosted with large margins by the proposed DSSC, 10% and 20% on ORL and PIE, respectively. It should be noted that DSSC achieves nearly perfect clustering on PIE, which is also shown by the optimal block-diagonal structure of the corresponding affinity matrix in Fig. 9.

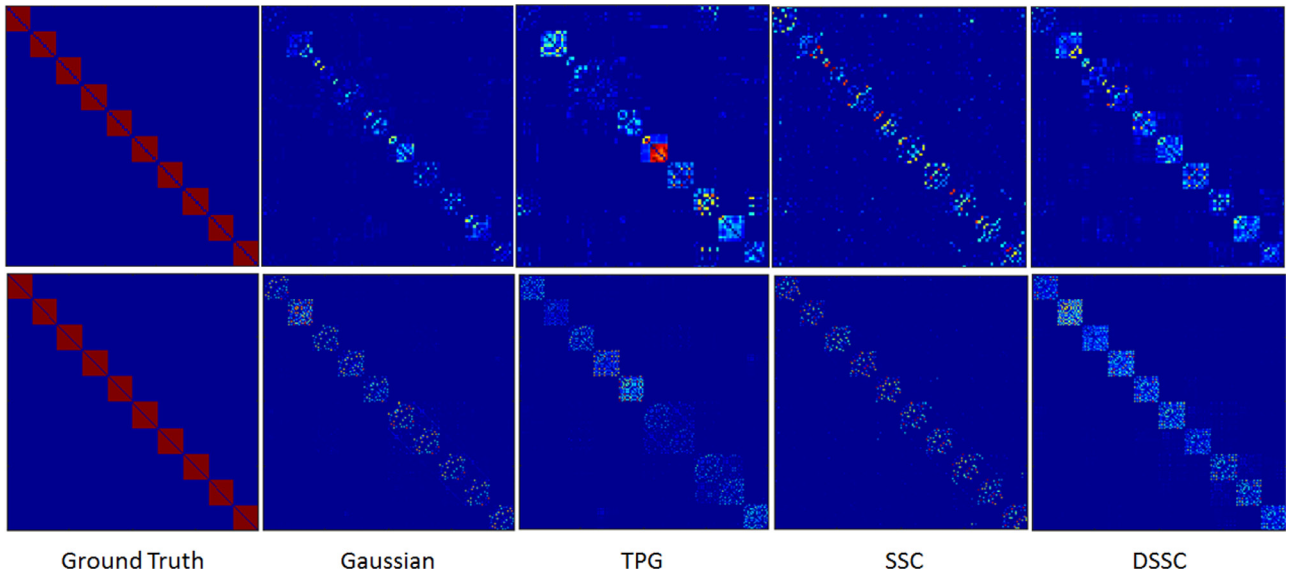


Fig. 9. Visualization of affinity matrices obtained by different methods. Top: ORL. Bottom: PIE.



Fig. 10. Example frames from videos in the motion segmentation dataset Hopkins 155 [52].

Table 4
Motion Segmentation Errors (%) on Hopkins 155 Dataset.

Methods	LRR [16]	LRSC [24]	LSR [54]	BDSSC [28]	SSC [6]	SSSC [29]	DSSC
2 motions	3.76	3.69	2.20	2.29	1.95	1.94	1.68
3 motions	9.92	7.69	7.13	4.95	4.94	4.92	4.64
All	5.15	4.59	3.31	2.89	2.63	2.61	2.35

4.3. Comparison with state-of-the-art subspace clustering methods

In this section, we evaluate the proposed DSSC algorithm on two commonly used datasets in the field of subspace clustering, Hopkins 155 [52] and Extended Yale B [53], against other state-of-the-art subspace clustering methods, i.e., SSC [6], SSSC [29], LRR [16], LSR [54], BDSSC [28], LRSC [24]. The experimental results of these methods are cited directly from [29], which provides a fair comparison as we use exactly the same experimental settings, as introduced in [6].

Hopkins 155. Motion segmentation refers to the problem of segmenting a video sequence of multiple rigidly moving objects into multiple spatiotemporal regions that correspond to different motions in the scene [6] (see Fig. 10). The objects are often represented and tracked by a set of feature points through all frames in the video. Stacking the trajectories of all feature points, we obtain a data vector that represents the motion of the corresponding object. Under the affine projection model, all feature trajectories associated with a single rigid motion lie in an affine subspace of dimension not more than 3 [6]. Therefore, motion segmentation reduces to the problem of clustering these trajectories in a union of subspaces.

We evaluate the proposed DSSC algorithm against the state-of-the-art subspace clustering methods on the Hopkins 155 motion segmentation dataset [52]. It consists of 155 video sequences, where 120 of them contain two motions and 35 contain three motions. We evaluate the average performance on three different cases: 2 motions, 3 motions, and all. It can be clearly seen from Table 4 that the proposed DSSC achieves the best performances on all cases.

Extended Yale B. In the last experiments, we evaluate the clustering performance of the proposed DSSC against the same state-of-the-art subspace clustering methods used for Hopkins 155 on the Extended Yale B dataset [53]. This dataset contains 2414 frontal face images of 38 subjects, with 64 images per subject acquired under different illumination conditions. In terms of subspace segmentation, this dataset is more challenging than the Hopkins 155 dataset due to the heavy noise, high-dimensional space, and large number of subspace in the data. In our experiments, we follow exactly the same settings as introduced in [6]: (1) each image is down-sampled to be 48×42 pixels, resulting in a 2,016-dimensional data point per image; (2) all the 38 subjects are divided into 4 groups, i.e., 1–10, 11–20, 21–30, and 31–38. For the first three groups, we conduct experiments using all possible

Table 5
Clustering errors (%) on the Extended Yale B Dataset.

Methods	LRR [16]	LRSC [24]	LSR [54]	BDSSC [28]	SSC [6]	SSSC [29]	DSSC
2 subjects	6.74	3.15	6.72	3.90	1.87	1.27	0.61
3 subjects	9.30	4.71	9.25	17.70	3.35	2.71	1.25
5 subjects	13.94	13.06	13.87	27.50	4.32	3.41	2.80
8 subjects	25.61	21.25	25.98	33.20	5.99	4.15	4.04
10 subjects	29.53	29.58	28.33	39.53	7.29	5.16	4.84

choices of $n \in 2, 3, 5, 8, 10$ subjects, and all possible choices of $n \in 2, 3, 5, 8$ for the last group.

Experimental results are presented in Table 5. It can be observed that the proposed DSSC once again achieves the best results on all cases. It is worth noting that on this more challenging data set, DSSC records more significant improvements compared to the state-of-the-art methods.

5. Conclusion

In this paper, we proposed a novel affinity learning method that combined with spectral clustering, named as Diffusion based Sparse Subspace Clustering, which successfully incorporates the ideas from both diffusion and subspace clustering. Comparing to the original diffusion method, which uses Gaussian affinity and manually enforced KNN sparsity, DSSC applies ℓ_1 norm that naturally produces sparse affinity based on the optimal reconstruction. For subspace clustering problem, where the data are often drawn from low-dimensional subspaces which lies in very high-dimensional feature space, we have qualitatively and quantitatively shown that ℓ_1 norm affinity is a more sensible choice to estimate the geometry of the data manifold than the Euclidean distance based Gaussian affinity. Comparing to Sparse Subspace Clustering, we have demonstrated that ℓ_1 norm could serve as a neighborhood selection criteria which captures local manifold structure, while a simple yet effective diffusion process can be used to reevaluate all the affinities by spreading the local structure following the global geometry of the data manifold. Extensive experiments on various types of data have shown the superiority of the proposed DSSC method over the state-of-the-art methods, both in diffusion and subspace clustering fields.

In the future, it will be interesting to study the effect of different methods for solving ℓ_1 minimization. It has been observed that some small reconstruction coefficients could degrade the clustering performance. Hence, the DSSC could be further improved when a ℓ_1 solver producing more clean coefficients is employed. Another possible work would be the combination of the proposed affinity learning method with other clustering algorithms that make use of an affinity matrix.

Acknowledgments

This work is supported by an Australian Government Research Training Program (RTP) Scholarship.

References

- [1] W. Hong, J. Wright, K. Huang, Y. Ma, Multiscale hybrid linear models for lossy image representation, *Image Process. IEEE Trans.* 15 (12) (2006) 3655–3671.
- [2] S. Rao, R. Tron, R. Vidal, Y. Ma, Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories, *Pattern Anal. Mach. Intelligence, IEEE Trans.* 32 (10) (2010) 1832–1845.
- [3] J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, D. Kriegman, Clustering appearances of objects under varying illumination conditions, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2003, pp. 1–11.
- [4] R.E. Bellman, *Dynamic Programming*, Princeton Univ. Press, 1957.
- [5] R. Vidal, A tutorial on subspace clustering, *IEEE Signal Process. Mag.* 28 (2) (2010) 52–68.
- [6] E. Elhamifar, R. Vidal, Sparse subspace clustering: algorithm, theory, and applications, *Pattern Anal. Mach. Intell. IEEE Trans.* 35 (11) (2013) 2765–2781.
- [7] J.P. Costeira, T. Kanade, A multibody factorization method for independently moving objects, *Int. J. Comput. Vis.* 29 (3) (1998) 159–179.
- [8] R. Vidal, Y. Ma, S. Sastry, Generalized principal component analysis (GPCA), *Pattern Anal. Mach. Intell. IEEE Trans.* 27 (12) (2005) 1945–1959.
- [9] P. Tseng, Nearest q-flat to m points, *J. Optim. Theory Appl.* 105 (1) (2000) 249–252.
- [10] L. Lu, R. Vidal, Combined central and subspace clustering for computer vision applications, in: *International Conference on Machine Learning*, 2006, pp. 593–600.
- [11] Y. Ma, H. Derksen, W. Hong, J. Wright, Segmentation of multivariate mixed data via lossy data coding and compression, *Pattern Anal. Mach. Intell. IEEE Trans.* 29 (9) (2007) 1546–1562.
- [12] S.R. Rao, R. Tron, R. Vidal, Y. Ma, Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.
- [13] T. Zhang, A. Szlam, Y. Wang, G. Lerman, Hybrid linear modeling via local best-fit flats, *Int. J. Comput. Vis.* 100 (3) (2012) 217–240.
- [14] A. Goh, R. Vidal, Segmenting motions of different types by unsupervised manifold clustering, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–6.
- [15] G. Chen, G. Lerman, Spectral curvature clustering (SCC), *Int. J. Comput. Vis.* 81 (3) (2009) 317–330.
- [16] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, Y. Ma, Robust recovery of subspace structures by low-rank representation, *IEEE Trans. Pattern Anal. Mach. Intell.* 35 (1) (2013) 171–184.
- [17] L. Zelnik-Manor, P. Perona, Self-tuning spectral clustering, in: *Advances in Neural Information Processing Systems*, 2004, pp. 1601–1608.
- [18] K. Taşdemir, Vector quantization based approximate spectral clustering of large datasets, *Pattern Recognit.* 45 (8) (2012) 3034–3044.
- [19] K. Taşdemir, B. Yalçın, I. Yildirim, Approximate spectral clustering with utilized similarity information using geodesic based hybrid distance measures, *Pattern Recognit.* 48 (4) (2015) 1465–1477.
- [20] Q. Li, Y. Ren, L. Li, W. Liu, Fuzzy based affinity learning for spectral clustering, *Pattern Recognit.* 60 (2016) 531–542.
- [21] X. Zhu, C.C. Loy, S. Gong, Constructing robust affinity graphs for spectral clustering, in: *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2014, pp. 1450–1457.
- [22] C. You, R. Vidal, Geometric conditions for subspace-sparse recovery, in: *International Conference on Machine Learning*, 2015, pp. 1585–1593.
- [23] B. Nasihatkon, R. Hartley, Graph connectivity in sparse subspace clustering, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 2137–2144.
- [24] P. Favaro, R. Vidal, A. Ravichandran, A closed form solution to robust subspace estimation and clustering, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1801–1807.
- [25] C. You, C.-G. Li, D.P. Robinson, R. Vidal, Oracle based active set algorithm for scalable elastic net subspace clustering, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3928–3937.
- [26] Y.-X. Wang, H. Xu, C. Leng, Provable subspace clustering: When lrr meets ssc, in: *Advances in Neural Information Processing Systems*, 2013, pp. 64–72.
- [27] Y. Panagakis, C. Kotropoulos, Elastic net subspace clustering applied to pop/rock music structure analysis, *Pattern Recognit. Lett.* 38 (2014) 46–53.
- [28] J. Feng, Z. Lin, H. Xu, S. Yan, Robust subspace segmentation with block-diagonal prior, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3818–3825.
- [29] C.-G. Li, C. You, R. Vidal, Structured sparse subspace clustering: a joint affinity learning and subspace clustering framework, *IEEE Trans. Image Process.* 26 (6) (2017) 2988–3001.
- [30] X. Yang, S. Koknar-Tezel, L.J. Latecki, Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 357–364.
- [31] X. Bai, X. Yang, L.J. Latecki, W. Liu, Z. Tu, Learning context-sensitive shape similarity by graph transduction, *Pattern Anal. Mach. Intell. IEEE Trans.* 32 (5) (2010) 861–874.
- [32] A. Egozi, Y. Keller, H. Guterman, Improving shape retrieval by spectral matching and meta similarity, *Image Process. IEEE Trans.* 19 (5) (2010) 1319–1327.
- [33] B. Wang, Z. Tu, Affinity learning via self-diffusion for image segmentation and clustering, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2312–2319.
- [34] M. Donoser, H. Bischof, Diffusion processes for retrieval revisited, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1320–1327.

- [35] X. Yang, L. Prasad, L.J. Latecki, Affinity learning with diffusion on tensor product graph, *Pattern Anal. Mach. Intell. IEEE Trans.* 35 (1) (2013) 28–38.
- [36] J. Shi, J. Malik, Normalized cuts and image segmentation, *Pattern Anal. Mach. Intell. IEEE Trans.* 22 (8) (2000) 888–905.
- [37] D. Zhou, O. Bousquet, T.N. Lal, J. Weston, B. Schölkopf, Learning with local and global consistency, in: *Advances in neural information processing systems*, 2004, pp. 321–328.
- [38] B. Wang, Z. Tu, Sparse subspace denoising for image manifolds, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 468–475.
- [39] L. Page, S. Brin, R. Motwani, T. Winograd, The PageRank citation ranking: bringing order to the web. (1999).
- [40] D. Zhou, J. Weston, A. Gretton, O. Bousquet, B. Schölkopf, Ranking on data manifolds, in: *Advances in Neural Information Processing Systems*, 2004, pp. 169–176.
- [41] D. Zhou, J. Huang, B. Schölkopf, Learning with hypergraphs: Clustering, classification, and embedding, in: *Advances in Neural Information Processing Systems*, 2007, pp. 1601–1608.
- [42] Y. Huang, Q. Liu, D. Metaxas, Video object segmentation by hypergraph cut, in: *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2009, pp. 1738–1745.
- [43] M. Meila, J. Shi, A random walks view of spectral segmentation(2001).
- [44] M. Szummer, T. Jaakkola, Partially labeled classification with markov random walks, in: *Advances in Neural Information Processing Systems*, 2002, pp. 945–952.
- [45] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, in: *Advances in Neural Information Processing Systems*, 2002, pp. 849–856.
- [46] J.J. Hull, A database for handwritten text recognition research, *Pattern Anal. Mach. Intell. IEEE Trans.* 16 (5) (1994) 550–554.
- [47] V. Premachandran, R. Kakarala, Consensus of k-nns for robust neighborhood selection on graph-based manifolds, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1594–1601.
- [48] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, Eigenfaces vs. fisherfaces: recognition using class specific linear projection, *Pattern Anal. Mach. Intell. IEEE Trans.* 19 (7) (1997) 711–720.
- [49] T. Sim, S. Baker, M. Bsat, The cmu pose, illumination, and expression (pie) database, in: *IEEE Conference on Automatic Face and Gesture Recognition*, 2002, pp. 53–58.
- [50] J. Wang, S.-F. Chang, X. Zhou, S.T. Wong, Active microscopic cellular image annotation by superposable graph transduction with imbalanced labels, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.
- [51] R. Basri, D.W. Jacobs, Lambertian reflectance and linear subspaces, *Pattern Anal. Mach. Intelligence, IEEE Trans.* 25 (2) (2003) 218–233.
- [52] R. Tron, R. Vidal, A benchmark for the comparison of 3-d motion segmentation algorithms, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.
- [53] A.S. Georghiades, P.N. Belhumeur, D.J. Kriegman, From few to many: illumination cone models for face recognition under variable lighting and pose, *Pattern Anal. Mach. Intell. IEEE Trans.* 23 (6) (2001) 643–660.
- [54] C.-Y. Lu, H. Min, Z.-Q. Zhao, L. Zhu, D.-S. Huang, S. Yan, Robust and efficient subspace segmentation via least squares regression, in: *European Conference on Computer Vision*, Springer, 2012, pp. 347–360.

Qilin Li received the BSc degree in Computer Science from Sun Yat-Sen University, PR China, in 2013, and the MPhil degree from Curtin University, Australia, in 2016. Currently, he is doing his PhD at Curtin University with interests in pattern recognition, computer vision and deep learning.

Wanquan Liu received the BSc degree in Applied Mathematics from Qufu Normal University, PR China, in 1985, the MSc degree in Control Theory and Operation Research from Chinese Academy of Science in 1988, and the PhD degree in Electrical Engineering from Shanghai Jiaotong University, in 1993. He once held the ARC Fellowship, U2000 Fellowship and JSPS Fellowship and attracted research funds from different resources over 2 million dollars. He is currently an Associate Professor in the Department of Computing at Curtin University and is in editorial board for seven international journals. His current research interests include large-scale pattern recognition, signal processing, machine learning, and control systems.

Ling Li obtained her Bachelor of Computer Science from Sichuan University, China, Master of Electrical Engineering from China Academy of Post and Telecommunication, and PhD of Computer Engineering from Nanyang Technological University (NTU), Singapore. She worked as an Assistant Professor and subsequently an Associate Professor in the School of Computer Engineering in NTU. She is now an Associate Professor in the Department of Computing at Curtin University, Australia. Her research interest is mainly in computer graphics and vision, and artificially intelligent beings. She has given a number of keynotes addresses in international conferences and published over 100 referred research papers in international journals and conferences.