2016 23rd International Conference on Pattern Recognition (ICPR)
Cancún Center, Cancún, México, December 4-8, 2016

Constrained Dominant Sets for Retrieval

Eyasu Zemene
Ca'Foscari University of Venice
Email: eyasu.zemene@unive.it

Leulseged Tesfaye Alemu
Ca'Foscari University of Venice
Email: leule24@gmail.com

Marcello Pelillo
Ca'Foscari University of Venice
Email: pelillo@dais.unive.it

***Abstract*—Learning new global relations based on an initial affinity of the database objects has shown significant improvements in similarity retrievals. Locally constrained diffusion process is one of the recent effective tools in learning the intrinsic manifold structure of a given data. Existing methods, which constrain the diffusion process locally, have problems - manual choice of optimal local neighborhood size, do not allow for intrinsic relation among the neighbors, fix initialization vector to extract dense neighbor - which negatively affect the affinity propagation. We propose a new approach, which alleviate these issues, based on some properties of a family of quadratic optimization problems related to dominant sets, a well-known graph-theoretic notion of a cluster which generalizes the concept of a maximal clique to edge-weighted graphs. In particular, we show that by properly controlling a regularization parameter which determines the structure and the scale of the underlying problem, we are in a position to extract dominant set cluster which is constrained to contain user-provided query. Experimental results on standard benchmark datasets show the effectiveness of the proposed approach.**

I. Introduction

Retrieval has recently attracted considerable attention within the computer vision community, especially because of its potential applications such as database retrieval, web and mobile image search. Given a user provided query, the goal is to provide as output a ranked list of objects that best reflect the user's intent. Classical approaches perform the task based on the (dis)similarity between the query and the database objects. The main limitation of such classical retrieval approaches is that they do not allow for the intrinsic relation among the database objects.

Recently, various techniques, instead of simply using the pairwise similarity, try to learn a better similarities that consider manifold structures of the underlying data. Qin *et al.* [1] try to alleviate the asymmetry problem of the k-nearest neighbor (k-NN) using the notion of k-reciprocal nearest neighbor. In [2] the notion of shared nearest neighbor is used to build secondary similarity measure, which stabilizes the performance of the search, based on the primary distance measure. In [3] shape meta-similarity measure, which is computed as the **L1** distance between new vector representation which considers only the k-NN set of similarities fixing all others to 0, was proposed. Choosing the right size of the neighbor is important. In [4], the notion of shortest path was used to built a new affinity for retrieval.

Diffusion process is one of the recent effective tools in learning the intrinsic manifold structure of a given data [5]–[7]. Given data, a weighted graph is built where the nodes are the objects and the edge weight is a function of the affinity between the objects. The pairwise affinities are then propagated following structure of the weighted edge links in the graph. The result of the affinity propagation highly depends on the quality of the pairwise similarity [8], [9]. Inaccurate Pairwise similarity results in a graph with much noise which negatively affects the diffusion process. Constraining the diffusion process locally alleviates this issue [5], [7], [9]. Dominant neighbor (DN) and k-NN are two notions used by the recent existing methods to constrain the diffusion process locally [5]–[7]. In [5], it has been shown that affinity learning constraining relation of an object to its neighbors effectively improves the retrieval performance and was able to achieve 100 % bull's eye score in the well known MPEG datset. The author of [5] put automatically selecting local neighborhood size (K) as the main limitation of the approach and is still an open problem. The influence of selecting different K values was also studied which proved that the parameter is a serious problem of the approach. For MPEG7 dataset, the choice is insignificant while for the other two datasets YALE and ORL choosing the reasonable K is difficult which resulted in a decrease in performance for the right value of K. Moreover, it is obvious that the selection of k-NN is prone to errors in the pairwise similarities [7]. Since any k-NN decision procedure relies only on affinities of an object to all other objects, k-NN approach is handicapped in resisting errors in pairwise affinities and in capturing the structure of the underlying data manifold.

Yang *et al.* in [7], to avoid the above issues, proposed the notion of dominant neighbors (DN). Instead of the k-NN, here a compact set from the k-NN which best explains the intrinsic relation among the neighbors is considered to constrain the diffusion process. However, the approach follows heuristic based k-NN initialization scheme. To capture dominant neighbors, the approach first choose a fixed value of K, collect the K nearest neighbors and then initialize the dynamics, the dynamics which extracts dense neighbors, to the barycenter of the face of the simplex which contains the neighbors. It is obvious to see that the approach is still dependent on K. Moreover, as fixing K limits the dynamics to a specified face of the simplex, objects out of k-NN(q) which form a dominant neighbor with q will be loosed. The chosen k-NN may also be fully noisy which might not have a compact structure.

In this paper, we propose a new approach to retrieval which can deal naturally with the above problems. Our approach is based on some properties of a parameterized family of quadratic optimization problems related to dominant-set clus-

978-1-5090-4846-5/16/$31.00 ©2016 IEEE

ters, a well-known generalization of the notion of maximal cliques to edge-weighted graph which have proven to be extremely effective in a variety of computer vision problems, including (automatic) image and video segmentation, group detection and tracking [10]–[14]. In particular, we show that by properly controlling a regularization parameter which determines the structure and the scale of the underlying problem, we are in a position to extract dominant-set cluster which is constrained to contain user-specified query.

The resulting algorithm has a number of interesting features which distinguishes it from existing approaches. Specifically: 1) it is able to constrain the diffusion process locally extracting dense neighbors whose local neighborhood size (K) is fixed automatically; different neighbors can have different value of K. 2) it does not have any initialization step; the dynamics, to extract the dense neighbors, can start at any point in the standard simplex 3) it turns out to be *robust* to noisy affinity matrices.

The rest of the paper is organized as follows. In the next section we will discuss the most related works to our approach. Section III will introduces in detail our constrained dominant set framework. The experimental results are given in section IV.

II. Diffusion Process

Given a set of n objects, the relation among them can be represented as an undirected edge-weighted graph $G = (V, E, w)$, where $V = \{1, ..., n\}$ is the vertex set, $E \subseteq V \times V$ is the edge set, and $w : E \to R^*_+$ is the (positive) weight function. Vertices in G correspond to data points, edges represent neighborhood relationships, and edge-weights reflect similarity between pairs of linked vertices. As customary, the graph G is represented with the corresponding weighted adjacency (or similarity) matrix, which is the $n \times n$ nonnegative, symmetric matrix $\mathtt{A} = (a_{ij})$ defined as $a_{ij} = w(i, j)$, if $(i, j) \in E$, and $a_{ij} = 0$ otherwise. A diffusion process then starts from a predefined initialization, say $\mathcal{V}$ and propagates the affinity value through the underlying manifold based on a predefined transition matrix, say $\mathcal{T}$, and diffusion scheme ($\mathcal{S}$).

Off-the-shelf diffusion processes, which basically differ based on the choice of $\mathcal{V}$, $\mathcal{T}$ and $\mathcal{S}$, the most related ones to this work are [7], [15]. In both cases, the diffusion process is locally constrained. While in [15] the notion of k-NN is used to constrain the diffusion process locally, dominant neighbor notion ($\mathcal{DN}$) is used by [7].

A. Nearest Neighbors

In the first case, the edge-weights of the k-NN are kept i.e define locally constrained affinity $\mathcal{L} = (l_{ij})$ defined as $l_{ij} = w(i, j)$, if $(i, j) \in$ k-NN(q), and $l_{ij} = 0$ otherwise. Then the diffusion process, setting $\mathcal{V}$ as the affinity $\mathtt{A}$, is performed by the following update rule.

$$\mathcal{V}_{t+1} = \mathcal{L}\mathcal{V}\mathcal{L}' \tag{1}$$

Nearest neighbors constrained diffusion process, alleviating the issue of noisy pairwise similarity, significantly increases the retrieval performance. However, the approach has two serious limitations: First, automatically selecting local neighborhood size (K) is very difficult and is still an open problem [5]. In [5] the influence of selecting different K values was studied which proved that the parameter is a serious problem of the approach. For MPEG7 dataset, the choice was insignificant while for the other two datasets, YALE and ORL, choosing the reasonable K was difficult which even resulted in a decrease in performance, for ORL from 77.30% to 73.40% and for YALE 77.08% to 73.39%, for the right value of K. Moreover, it is obvious that the selection of k-NN is prone to errors in the pairwise similarities [7].

B. Dominant Neighbors

Yang *et al.* in [7], to avoid the above issues, proposed the notion of dominant neighbors ($\mathcal{DN}$). Instead of the k-NN, here a compact set from the k-NN which best explains the intrinsic relation among the neighbors is considered to constrain the diffusion process. To do so, the author used the dominant set framework by Pavan and Pelillo [11].

A dominant neighbor ($\mathcal{DN}$) is set as a dominant set, say $\mathcal{DS}$, from the k-NN which contains the user provided query q, lets call it $\mathcal{DS}(q)$.

1) The Dominant set Framework: The dominant set framework is a well-known graph-theoretic notion of a cluster which generalizes the concept of a maximal clique to edge-weighted graphs. The approach has proven to be a fast and efficient framework for various applications [11]–[13], [16]. A generalization of its ideas to hypergraphs and multigraphs has recently been developed in [17], [18]. It has also proven to be effective in capturing the structure of the underlying data manifold [19].

In the dominant set framework, the data to be clustered are represented as an undirected edge-weighted graph with no self-loops. Note that all entries on the main diagonal of $\mathtt{A}$ are zero.

For a non-empty subset $S \subseteq V$, $i \in S$, and $j \notin S$, define

$$\phi_S(i, j) = a_{ij} - \frac{1}{|S|} \sum_{k \in S} a_{ik} \ . \tag{2}$$

This quantity measures the (relative) similarity between nodes j and i, with respect to the average similarity between node i and its neighbors in S. Note that $\phi_S(i, j)$ can be either positive or negative. Next, to each vertex $i \in S$ we assign a weight defined (recursively) as follows:

$$w_S(i) = \begin{cases} 1, & \text{if} \quad |S| = 1, \\ \sum_{j \in S \setminus \{i\}} \phi_{S \setminus \{i\}}(j, i) w_{S \setminus \{i\}}(j), & \text{otherwise} \ . \end{cases} \tag{3}$$

Intuitively, $w_S(i)$ gives us a measure of the overall similarity between vertex i and the vertices of $S \setminus \{i\}$ with respect to the overall similarity among the vertices in $S \setminus \{i\}$. Therefore, a positive $w_S(i)$ indicates that adding i into its neighbors in S will increase the internal coherence of the set, whereas in the presence of a negative value we expect the overall coherence

to be decreased. Finally, the total weight of S can be simply defined as

$$W(S) = \sum_{i \in S} w_S(i) \ . \tag{4}$$

A non-empty subset of vertices $S \subseteq V$ such that $W(T) > 0$ for any non-empty $T \subseteq S$, is said to be a *dominant set* if:

1) $w_S(i) > 0$, for all $i \in S$,
2) $w_{S \cup \{i\}}(i) < 0$, for all $i \notin S$.

It is evident from the definition that a dominant set satisfies the two basic properties of a cluster: internal coherence and external incoherence. Condition 1 indicates that a dominant set is internally coherent, while condition 2 implies that this coherence will be destroyed by the addition of any vertex from outside. In other words, a dominant set is a maximally coherent data set.

Now, consider the following linearly-constrained quadratic optimization problem:

$$\begin{array}{ll} \text{maximize} & f(\mathbf{x}) = \mathbf{x}'\mathtt{A}\mathbf{x} \\ \text{subject to} & \mathbf{x} \in \Delta \end{array} \tag{5}$$

where a prime denotes transposition and

$$\Delta = \left\{ \mathbf{x} \in R^n \ : \ \sum_{i=1}^{n} x_i = 1, \text{ and } x_i \geq 0 \text{ for all } i = 1 \ldots n \right\}$$

is the standard simplex of R^n. In [11], [20] a connection is established between dominant sets and the local solutions of (5). In particular, it is shown that if S is a dominant set then its "weighted characteristics vector," which is the vector of Δ defined as,

$$x_i = \begin{cases} \frac{w_S(i)}{W(s)}, & \text{if} \quad i \in S, \\ 0, & \text{otherwise} \end{cases}$$

is a strict local solution of (5). Conversely, under mild conditions, it turns out that if $\mathbf{x}$ is a (strict) local solution of program (5) then its "support"

$$\sigma(\mathbf{x}) = \{i \in V \ : \ x_i > 0\}$$

is a dominant set. By virtue of this result, we can find a dominant set by first localizing a solution of program (5) with an appropriate continuous optimization technique, and then picking up the support set of the solution found. In this sense, we indirectly perform combinatorial optimization via continuous optimization.

A simple and effective optimization algorithm to extract a dominant set from a graph is given by the so-called *replicator dynamics*, developed and studied in evolutionary game theory, which are defined as follows:

$$x_i^{(t+1)} = x_i^{(t)} \frac{(\mathtt{A}\mathbf{x}^{(t)})_i}{(\mathbf{x}^{(t)})'\mathtt{A}(\mathbf{x}^{(t)})} \tag{6}$$

for $i = 1, \ldots, n$.

Yang *et al.* in [7], to find a dominant set in the k-NN, $\mathcal{DS}(q)$, initialized (6) with the nearest neighbor of q (k-NN(q)). They set, say the initial time is set as $t = 1$, $x_i(1) = 1/K$ if $i \in$ k-NN(q) zero otherwise. After the convergence of (6), say to $\mathbf{x}^*$, $\mathcal{DN}(q)$ is set as the support of $\mathbf{x}^*$, $i \in \mathcal{DN}(q)$ if and only if $i \in \sigma(\mathbf{x}^*)$. The edge-weights of the $\mathcal{DN}(q)$ are then kept i.e define locally constrained affinity $\mathcal{L} = (l_{ij})$ defined as $l_{ij} = w(i,j)$, if $(i,j) \in \mathcal{DN}(q)$, and $l_{ij} = 0$ otherwise. Then the diffusion process, setting $\mathcal{V}$ as the affinity $\mathtt{A}$, is performed by the same update rule as in (1).

The $\mathcal{DN}$ approach has proven to be more effective than the k-NN approach [6], [7], [21]. The approach, while effective, is rather heuristic in nature and has limitations. The approach initializing (6) with the nearest neighbor of q (k-NN(q)) limits the dynamics to the face of the simplex which contains k-NN(q). Moreover, a fixed value of K should be chosen for initializing (6), the approach, as it follows k-NN based initializing scheme, is still dependent on K. However number of nearest neighbors K may be different for different objects. As fixing K limits the dynamics to a specified face of the simplex, objects out of k-NN(q) which form a dominant set with q will be loosed. The chosen k-NN may also be fully noisy which might not have a compact structure.

III. Constrained dominant sets

Let $G = (V, E, w)$ be an edge-weighted graph with n vertices and let $\mathtt{A}$ denote its (weighted) adjacency matrix. Given a subset of vertices $S \subseteq V$ and a parameter $\alpha > 0$, define the following parameterized family of quadratic programs:

$$\begin{array}{ll} \text{maximize} & f_S^{\alpha}(\mathbf{x}) = \mathbf{x}'(\mathtt{A} - \alpha \hat{I}_S)\mathbf{x} \\ \text{subject to} & \mathbf{x} \in \Delta \end{array} \tag{7}$$

where $\hat{I}_S$ is the $n \times n$ diagonal matrix whose diagonal elements are set to 1 in correspondence to the vertices contained in $V \setminus S$ and to zero otherwise, and the 0's represent null square matrices of appropriate dimensions. In other words, assuming for simplicity that S contains, say, the first k vertices of V, we have:

$$\hat{I}_S = \begin{pmatrix} 0 & 0 \\ 0 & I_{n-k} \end{pmatrix}$$

where I_{n-k} denotes the $(n-k) \times (n-k)$ principal submatrix of the $n \times n$ identity matrix I indexed by the elements of $V \setminus S$. Accordingly, the function f_S^{α} can also be written as follows:

$$f_S^{\alpha}(\mathbf{x}) = \mathbf{x}'\mathtt{A}\mathbf{x} - \alpha \mathbf{x}_S'\mathbf{x}_S$$

$\mathbf{x}_S$ being the $(n-k)$-dimensional vector obtained from $\mathbf{x}$ by dropping all the components in S. Basically, the function f_S^{α} is obtained from f by inserting in the affinity matrix $\mathtt{A}$ the value of the parameter α in the main diagonal positions corresponding to the elements of $V \setminus S$.

Notice that this differs markedly, and indeed generalizes, the formulation proposed in [20] for obtaining a hierarchical clustering in that here, only a subset of elements in the main diagonal is allowed to take the α parameter, the other ones being set to zero. We note in fact that the original (non-regularized) dominant-set formulation (5) [11] as well as its regularized counterpart described in [20] can be considered as degenerate version of ours, corresponding to the cases

$S = V$ and $S = \emptyset$, respectively. It is precisely this increased flexibility which allows us to use this idea for finding groups of "constrained" dominant-set clusters.

We now derive the Karush-Kuhn-Tucker (KKT) conditions for program (7), namely the first-order necessary conditions for local optimality (see, e.g., [22]). For a point $\mathbf{x} \in \Delta$ to be a KKT-point there should exist n nonnegative real constants $\mu_1, \ldots, \mu_n$ and an additional real number λ such that

$$[(\mathtt{A} - \alpha \hat{I}_S)\mathbf{x}]_i - \lambda + \mu_i = 0$$

for all $i = 1 \ldots n$, and

$$\sum_{i=1}^{n} x_i \mu_i = 0 \ .$$

Since both the x_i's and the μ_i's are nonnegative, the latter condition is equivalent to saying that $i \in \sigma(\mathbf{x})$ implies $\mu_i = 0$, from which we obtain:

$$[(\mathtt{A} - \alpha \hat{I}_S)\mathbf{x}]_i \begin{cases} = \lambda, & \text{if } i \in \sigma(\mathbf{x}) \\ \leq \lambda, & \text{if } i \notin \sigma(\mathbf{x}) \end{cases}$$

for some constant λ. Noting that $\lambda = \mathbf{x}'\mathtt{A}\mathbf{x} - \alpha \mathbf{x}'_S \mathbf{x}_S$ and recalling the definition of $\hat{I}_S$, the KKT conditions can be explicitly rewritten as:

$$\begin{cases} (\mathtt{A}\mathbf{x})_i - \alpha x_i &= \mathbf{x}'\mathtt{A}\mathbf{x} - \alpha \mathbf{x}'_S \mathbf{x}_S, & \text{if } i \in \sigma(\mathbf{x}) \text{ and } i \notin S \\ (\mathtt{A}\mathbf{x})_i &= \mathbf{x}'\mathtt{A}\mathbf{x} - \alpha \mathbf{x}'_S \mathbf{x}_S, & \text{if } i \in \sigma(\mathbf{x}) \text{ and } i \in S \\ (\mathtt{A}\mathbf{x})_i &\leq \mathbf{x}'\mathtt{A}\mathbf{x} - \alpha \mathbf{x}'_S \mathbf{x}_S, & \text{if } i \notin \sigma(\mathbf{x}) \end{cases} \tag{8}$$

We are now in a position to discuss the main results which motivate the algorithm presented in this paper. Note that, in the sequel, given a subset of vertices $S \subseteq V$, the face of Δ corresponding to S is given by: $\Delta_S = \{x \in \Delta : \sigma(x) \subseteq S\}$.

proposition 1: Let $S \subseteq V$, with $S \neq \emptyset$. Define

$$\gamma_S = \max_{\mathbf{x} \in \Delta_{V \setminus S}} \min_{i \in S} \frac{\mathbf{x}'\mathtt{A}\mathbf{x} - (\mathtt{A}\mathbf{x})_i}{\mathbf{x}'\mathbf{x}} \tag{9}$$

and let $\alpha > \gamma_S$. If $\mathbf{x}$ is a local maximizer of f_S^α in Δ, then $\sigma(\mathbf{x}) \cap S \neq \emptyset$.

Proof: Let $\mathbf{x}$ be a local maximizer of f_S^α in Δ, and suppose by contradiction that no element of $\sigma(\mathbf{x})$ belongs to S or, in other words, that $\mathbf{x} \in \Delta_{V \setminus S}$. By letting

$$i = \arg\min_{j \in S} \frac{\mathbf{x}'\mathtt{A}\mathbf{x} - (\mathtt{A}\mathbf{x})_j}{\mathbf{x}'\mathbf{x}}$$

and observing that $\sigma(\mathbf{x}) \subseteq V \setminus S$ implies $\mathbf{x}'\mathbf{x} = \mathbf{x}'_S \mathbf{x}_S$, we have:

$$\alpha > \gamma_S \geq \frac{\mathbf{x}'\mathtt{A}\mathbf{x} - (\mathtt{A}\mathbf{x})_i}{\mathbf{x}'\mathbf{x}} = \frac{\mathbf{x}'\mathtt{A}\mathbf{x} - (\mathtt{A}\mathbf{x})_i}{\mathbf{x}'_S \mathbf{x}_S} \ .$$

Hence, $(\mathtt{A}\mathbf{x})_i > \mathbf{x}'\mathtt{A}\mathbf{x} - \alpha \mathbf{x}'_S \mathbf{x}_S$ for $i \notin \sigma(\mathbf{x})$, but this violates the KKT conditions (8), thereby proving the proposition. ■

The following proposition provides a useful and easy-to-compute upper bound for γ_S.

proposition 2: Let $S \subseteq V$, with $S \neq \emptyset$. Then,

$$\gamma_S \leq \lambda_{\max}(\mathtt{A}_{V \setminus S}) \tag{10}$$

where $\lambda_{\max}(\mathtt{A}_{V \setminus S})$ is the largest eigenvalue of the principal submatrix of $\mathtt{A}$ indexed by the elements of $V \setminus S$.

Proof: Let $\mathbf{x}$ be a point in $\Delta_{V \setminus S}$ which attains the maximum γ_S as defined in (9). Using the Rayleigh-Ritz theorem [23] and the fact that $\sigma(\mathbf{x}) \subseteq V \setminus S$, we obtain:

$$\lambda_{\max}(\mathtt{A}_{V \setminus S}) \geq \frac{\mathbf{x}'_S \mathtt{A}_{V \setminus S} \mathbf{x}_S}{\mathbf{x}'_S \mathbf{x}_S} = \frac{\mathbf{x}'\mathtt{A}\mathbf{x}}{\mathbf{x}'\mathbf{x}} \ .$$

Now, define $\gamma_S(\mathbf{x}) = \max\{(A\mathbf{x})_i \ : \ i \in S\}$. Since $\mathtt{A}$ is nonnegative so is $\gamma_S(\mathbf{x})$, and recalling the definition of γ_S we get:

$$\frac{\mathbf{x}'\mathtt{A}\mathbf{x}}{\mathbf{x}'\mathbf{x}} \geq \frac{\mathbf{x}'\mathtt{A}\mathbf{x} - \gamma_S(\mathbf{x})}{\mathbf{x}'\mathbf{x}} = \gamma_S$$

which concludes the proof. ■

The two previous propositions provide us with a simple technique to determine dominant-set clusters containing user-selected vertices. Indeed, if S is the user provided query q, by setting

$$\alpha > \lambda_{\max}(\mathtt{A}_{V \setminus S}) \tag{11}$$

we are guaranteed that all local solutions of (7) will have a support that necessarily contains the user specified object. As customary, we can use replicator dynamics or more sophisticated algorithms to find them [24]. Note that this does not necessarily imply that the (support of the) solution found corresponds to a dominant-set cluster of the original affinity matrix $\mathtt{A}$, as adding the parameter $-\alpha$ on a portion of the main diagonal intrinsically changes the scale of the underlying problem.

Given a query q, we scale the affinity and run the replicator (6), say the dynamics converges to $\mathbf{x}^*$. The support of $\mathbf{x}^*$, $\sigma(\mathbf{x}^*)$, is the constrained dominant set which contains the query q, let us call it $\mathcal{CDS}(q)$. The edge-weights of the $\mathcal{CDS}(q)$ are then kept i.e define locally constrained affinity $\mathcal{L} = (l_{ij})$ defined as $l_{ij} = w(i,j)$, if $(i,j) \in \mathcal{CDS}(q)$, and $l_{ij} = 0$ otherwise. The diffusion process is then performed by the same update rule as in (1). For the proof of convergence of the update rule we refer the reader to [6].

IV. Experiments

The performance of the approach is presented in this section. The approach was tested against three well known data sets in the field of retrieval: MPEG7(shape), YALE(faces) and ORL(faces). For all test data sets the number of iterations for the update rule is set to 200. A given pairwise distance $\mathcal{D}$ is transformed to similarity (edge-weight) using a standard Gaussian kernel

$$\mathtt{A}^\sigma_{ij} = \mathbb{1}_{i \neq j} exp(-\mathcal{D}/2\sigma^2)$$

where σ is the free scale parameter, and $\mathbb{1}_P = 1$ if P is true, 0 otherwise. $\mathcal{L}$ is then built, from $\mathtt{A}$, using the constrained dominant set framework. The diffusion process is then computed using the update rule (1) which resulted in the final learned affinity for ranking.

A similar experimental analysis as of [5] has been conducted. In [5], a generic framework with 72 different variant

of diffusion processes was defined which are resulted from three steps: initialization, definition of transition matrix and diffusion process. In our experiment, the update scheme is fixed to (1) which has proven to be effective. The four different types of initialization schemes are Affinity Matrix $\mathtt{A}$ (A1) [25], Identity Matrix $\mathbf{I}$ (A2), Transition Matrix $\mathbf{P}$ which is the standard random walk transition matrix (A3) [26] and Transition Matrix $\mathbf{P}_{kNN}$ which is the random walk transition matrix constrained to the k-nearest neighbors (A4) [26]. Including our transition matrix (B6), we have in total 6 different types of transition matrices: $\mathbf{P}$ (B1), Personalized PageRank Transition Matrix $\mathbf{P}_{PPR}$ (B2) [26], $\mathbf{P}_{kNN}$ (B3), Dominant Set Neighbors $\mathbf{P}_{DS}$ [7] (B4), and Affinity Matrix $\mathtt{A}$ (B5)

Metric: The Bull's eye score is used as a measure of retrieval accuracy. It measures the percentage of objects sharing the same class with a query q in the top $\mathcal{R}$ retrieved shapes. Let us say $\mathcal{C}$ is the set of objects in the same class of the query q and $\mathcal{O}$ is the set of top $\mathcal{R}$ retrieved shapes. The Bull's eye score ($\mathcal{B}$) is then computed as $\mathcal{B}=\frac{|\mathcal{O}\cap\mathcal{C}|}{|\mathcal{C}|}$

MPEG7: a well known data set for testing performance of retrieval and shape matching algorithms. It comprises 1400 silhouette shape images of 70 different categories with 20 images in each categories. Articulated Invariant Representation (AIR) [27], best performing shape matching algorithm, is used as the input pairwise distance measure. The retrieval performance is measured fixing $\mathcal{R}$ to 40.

MPEG7	B1	B2	B3	B4	B5	B6(Ours)
A1	99.91	99.93	**100**	**100**	99.88	**100**
A2	99.92	99.93	**100**	**100**	99.88	**100**
A3	99.93	99.94	**100**	**100**	99.88	**100**
A4	99.92	99.94	**100**	**100**	99.88	**100**

TABLE I: Results on MPEG7 dataset. Bull's eye score for the first 40 elements

Table I shows bull's eye score on MPEG7 dataset. Observe that we were able to achieve 100% bulle's eye score while alleviating serious problems such as the problem of selecting a reasonable local neighborhood size and initializing the dynamics to find dense neighbors.

The retrieval performance has also been tested by varying the first $\mathcal{R}$ returned objects, the set in which instances of the same category are checked in. For the purpose of this experiment we used the best diffusion variants (B3 and B4 initialized with A2). The performance of the algorithms is shown in Table II. As can be observed, in this case our algorithm, besides giving flexibility, shows a small increment in the results.

MPEG7 has been used, most frequently, for testing retrieval algorithms. Table III shows the comparison against different state-of-the-art approaches. The proposed approach and [5] achieve 100% bulle's eye score. However, [5] needs to set an optimal neighborhood size whereas in our approach the number of neighbors to individual items arises intuitively.

$\mathcal{R}$	20	25	30	35	40
B3	94.321	97.871	**98.614**	99.357	**100**
B4	94.296	97.846	**98.614**	99.357	**100**
Ours	**94.354**	**97.896**	**98.614**	**99.360**	**100**

TABLE II: Results on MPEG7 dataset varying the first $\mathcal{R}$ returned objects

Methods	[28]	[29]	[27]	[30]	[7]	[5]	Ours
$\mathcal{B}$	85.40	91.61	93.67	95.96	99.99	**100**	**100**

TABLE III: Retrieval performance comparison on MPEG7 dataset. **Up:** methods, **Down:** Bull's eye score for the first 40 elements

YALE: [31] a popular benchmark for face clustering which consists of 15 unique people with 11 pictures for each under different conditions: normal, sad, sleepy, center light, right light, etc that include variations of pose, illumination and expression. Similar procedures of [5], [32] were followed to build the distance matrix. Down sample the image, normalize to 0-mean and 1-variance, and compute the Euclidean distance between the vectorized representation. The retrieval performance, measured fixing $\mathcal{R}$ to 15, is demonstrated in table IV. Our approach shows a small improvement in the retrieval performance except in one where the affinity itself initializes the diffusion process.

YALE	B1	B2	B3	B4	B5	B6(Ours)
A1	71.74	71.24	**75.59**	75.31	70.25	75.15
A2	71.96	70.69	77.30	76.20	69.92	**77.41**
A3	72.07	70.57	74.93	76.14	70.30	**75.37**
A4	72.23	70.74	77.08	76.10	70.25	**77.36**

TABLE IV: Results on YALE dataset. Bull's eye score for the first 15 elements

Results of the algorithm on YALE data set varying $\mathcal{R}$ is shown in Table V.

$\mathcal{R}$	11	13	15	17	20
B3	71.240	**74.105**	77.303	79.559	80.826
B4	70.854	72.176	76.198	77.741	79.063
Ours	**71.350**	74.050	**77.411**	**80.000**	**81.653**

TABLE V: Results on YALE dataset varying the first $\mathcal{R}$ returned objects

ORL: face data set of 40 different persons with 10 grayscale images per person with slight variations of pose, illumination, and expression. Similar procedure as of YALE data set was followed and The retrieval performance is measured fixing $\mathcal{R}$ to 15.

Results of the algorithm on ORL data set varying $\mathcal{R}$ is shown in Table VII.

As can be observed from Tables VI and VII, the proposed approach and [5] perform equally on the ORL face dataset.

ORL	B1	B2	B3	B4	B5	B6(Ours)
A1	72.75	73.48	**74.25**	73.90	70.58	**74.25**
A2	72.75	73.75	**77.42**	74.82	70.15	**77.42**
A3	73.12	73.75	**75.52**	75.35	71.05	**75.52**
A4	73.12	73.75	**77.32**	75.50	71.40	**77.32**

TABLE VI: Results on ORL dataset. Bull's eye score for the first 15 elements

$\mathcal{R}$	10	13	15	17	20
B3	**70.950**	**75.250**	**77.425**	**79.275**	**80.550**
B4	68.850	72.900	74.825	76.775	77.700
Ours	**70.950**	**75.250**	**77.425**	**79.275**	**80.550**

TABLE VII: Results on ORL dataset varying the first $\mathcal{R}$ returned objects

V. Conclusion

In this work, we have developed a locally constrained diffusion process which, as of existing methods, has no problems such as choosing optimal local neighbor size and initializing the dynamics to extract dense neighbor which constrain the diffusion process. The framework alleviates the issues while improving the performance. Experimental results on three well known data sets in the field of retrieval demonstrate that the approach compares favorably with state-of-the-art algorithms. Future work will focus on applying the framework on other computer vision problems such as action retrieval and video object segmentation and co-segmentation.

References

[1] D. Qin, S. Gammeter, L. Bossard, T. Quack, and L. J. V. Gool, "Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors," in *IEEE, CVPR*, 2011, pp. 777–784.

[2] M. E. Houle, H. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Can shared-neighbor distances defeat the curse of dimensionality?" in *SSDBM*, 2010, pp. 482–500.

[3] A. Egozi, Y. Keller, and H. Guterman, "Improving shape retrieval by spectral matching and meta similarity," *IEEE Trans. Image Processing*, vol. 19, no. 5, pp. 1319–1327, 2010.

[4] P. Kontschieder, M. Donoser, and H. Bischof, "Beyond pairwise shape similarity analysis," in *ACCV*, 2009, pp. 655–666.

[5] M. Donoser and H. Bischof, "Diffusion processes for retrieval revisited," in *IEEE, CVPR*, 2013, pp. 1320–1327.

[6] X. Yang and L. J. Latecki, "Affinity learning on a tensor product graph with applications to shape and image retrieval," in *IEEE, CVPR*, 2011, pp. 2369–2376.

[7] X. Yang, L. Prasad, and L. J. Latecki, "Affinity learning with diffusion on tensor product graph," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 28–38, 2013.

[8] S. Lafon and A. B. Lee, "Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 9, pp. 1393–1403, 2006.

[9] M. Szummer and T. S. Jaakkola, "Partially labeled classification with markov random walks," in *NIPS*, 2001, pp. 945–952.

[10] M. Pavan and M. Pelillo, "Graph-theoretic approach to clustering and segmentation," in *(CVPR*, 2003, pp. 145–152.

[11] ——, "Dominant sets and pairwise clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 167–172, 2007.

[12] S. Vascon, E. Z. Mequanint, M. Cristani, H. Hung, M. Pelillo, and V. Murino, "Detecting conversational groups in images and sequences: A robust game-theoretic approach," *CVIU*, vol. 143, pp. 11–24, 2016.

[13] ——, "A game-theoretic probabilistic approach for detecting conversational groups," in *ACCV*, 2014, pp. 658–675.

[14] Y. T. Tesfaye, E. Zemene, M. Pelillo, and A. Prati, "Multi-object tracking using dominant sets," *IET computer vision*, vol. 10, p. 289298, 2016.

[15] X. Yang, S. Köknar-Tezel, and L. J. Latecki, "Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval," in *IEEE, CVPR*, 2009, pp. 357–364.

[16] M. Pavan and M. Pelillo, "Dominant sets and hierarchical clustering," in *IEEE, ICCV*, 2003, pp. 362–369.

[17] S. Rota Bulò and M. Pelillo, "A game-theoretic approach to hypergraph clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1312–1327, 2013.

[18] E. Z. Mequanint, S. Rota Bulò, and M. Pelillo, "Dominant-set clustering using multiple affinity matrices," in *SIMBAD*, 2015, pp. 186–198.

[19] E. Zemene and M. Pelillo, "Path-based dominant-set clustering," in *ICIAP*, 2015, pp. 150–160.

[20] M. Pavan and M. Pelillo, "Graph-theoretic approach to clustering and segmentation," in *IEEE, CVPR*, 2003, pp. 145–152.

[21] H. Liu, X. Yang, L. J. Latecki, and S. Yan, "Dense neighborhoods on affinity graph," *IJCV*, vol. 98, no. 1, pp. 65–82, 2012.

[22] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*, 2008, vol. 3.

[23] R. Horn and C. Johnson, *Matrix Analysis*. Cambridge University Press, 1990.

[24] S. Rota Bulò and I. M. Bomze, "Infection and immunization: A new class of evolutionary game dynamics," *Games and Economic Behavior*, vol. 71, no. 1, pp. 193–211, 2011.

[25] B. Wang and Z. Tu, "Affinity learning via self-diffusion for image segmentation and clustering," in *IEEE CVPR*, 2012, pp. 2312–2319.

[26] C. Minsu and M. Kyoung, "Authority-shift clustering: Hierarchical clustering by authority seeking on graphs," in *IEEE CVPR*, 2010, pp. 3193–3200.

[27] R. Gopalan, P. K. Turaga, and R. Chellappa, "Articulation-invariant representation of non-planar shapes," in *ECCV*, 2010, pp. 286–299.

[28] H. Ling and D. W. Jacobs, "Shape classification using the inner-distance," *IEEE TPAMI*, vol. 29, no. 2, pp. 286–299.

[29] X. Bai, X. Yang, L. J. Latecki, W. Liu, and Z. Tu, "Learning context-sensitive shape similarity by graph transduction," *IEEE TPAMI*, vol. 32, no. 5, pp. 861–874, 2010.

[30] H. Ling, X. Yang, and L. J. Latecki, "Balancing deformability and discriminability for shape matching," in *ECCV*, 2010, pp. 411–424.

[31] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, 1997.

[32] J. Jiang, B. Wang, and Z. Tu, "Unsupervised metric learning by self-smoothing operator," in *IEEE, ICCV*, 2011, pp. 794–801.