Regularized Diffusion Process for Visual Retrieval

Song Bai,[1] Xiang Bai,[1]* Qi Tian,[2] Longin Jan Latecki[3]
[1]Huazhong University of Science and Technology
[2]University of Texas at San Antonio, [3]Temple University
{songbai, xbai}@hust.edu.cn, qitian@cs.utsa.edu, latecki@temple.edu

Abstract

Diffusion process has advanced visual retrieval greatly owing to its capacity in capturing the geometry structure of the underlying manifold. Recent studies (Donoser and Bischof 2013) have experimentally demonstrated that diffusion process on the tensor product graph yields better retrieval performances than that on the original affinity graph. However, the principle behind this kind of diffusion process remains unclear, *i.e.*, what kind of manifold structure is captured and how it is reflected. In this paper, we propose a new variant of diffusion process, which also operates on a tensor product graph. It is defined in three equivalent formulations (regularization framework, iterative framework and limit framework, respectively). Based on our study, three insightful conclusions are drawn which theoretically explain how this kind of diffusion process can better reveal the intrinsic relationship between objects. Besides, extensive experimental results on various retrieval tasks testify the validity of the proposed method.

Introduction

Given a query object, the goal of retrieval task is to find similar objects in the database according to pre-defined similarity measures. Conventionally, it is accomplished by computing pairwise dissimilarity between features in the Euclidean space. Then, similar objects are expected to be distributed with larger similarities to the query, thus they can be ranked in higher positions of the ranking list. However, it has been demonstrated (Zhou et al. 2004b) that the pairwise formulation is insufficient to reveal the intrinsic relationship between objects. Instead, similarities can be estimated more accurately along the geodesic path of the underlying data manifold, *i.e.*, in the context of other objects.

To capture the local geometry structure of the manifold, many algorithms have been developed in literature. Those algorithms share a very diverse nomenclature, including but not limited to context sensitive similarity (Jegou et al. 2010; Bai et al. 2010), affinity learning (Wang and Tu 2012; Jiang, Wang, and Tu 2011; Kontschieder, Donoser, and Bischof 2009), re-ranking (Zhang et al. 2012; Qin et al. 2011; Shen et al. 2012), ranking list comparison (Pedronette and Torres 2013; Chen et al. 2014; Pedronette, Almeida, and da Silva Torres 2014). Nevertheless, most of them model the relationships between objects on graph-based manifolds, where the vertices in the graph represent objects and the edge connecting two adjacent vertices is weighted by their similarity value. Then, similarity values are diffused on the graph in an iterative manner (*e.g.*, random walk (Page et al. 1999)). This procedure is usually called diffusion process (Yang, Koknar-Tezel, and Latecki 2009; Luo et al. 2013) in the retrieval domain.

A recent survey paper (Donoser and Bischof 2013) summarizes most common variants of diffusion process in a unified framework, and experimentally benchmarks the difference of these variants in retrieval performances. The experimental results suggest that diffusion process on tensor product graph (Yang, Koknar-Tezel, and Latecki 2009), built by computing the tensor product of the original affinity graph with itself, exhibits its superiority over other kinds of diffusion process. Tensor product graph naturally takes into account high order information, which is stated to be helpful for retrieval on manifold. However, the mechanism behind this kind of diffusion process still remains unclear. Some critical questions are: 1) what kind of manifold structure is captured and how it is reflected; 2) why high order information is useful; 3) what is the role of iteration and how many iterations are needed.

In this paper, we propose a new variant of diffusion process. The proposed method has three equivalent formulations as regularization framework, iterative framework and limit framework, respectively. Based on our study, insightful conclusions are drawn which theoretically explain why diffusion process on tensor product graph can better reveal the intrinsic relationship between objects, as:

- In the regularization framework, the proposed method can be taken as an improved version of manifold ranking with a relaxed smoothness term. Moreover, it imposes a high-order (tensor) graph Laplacian to smooth the pairwise relationship between each two nodes in the original graph.
- In the iterative framework, it can be taken as a special case of label propagation with only one category label. Then the retrieval task is converted to a classification task applied to a pair of objects, and the only category label is called "correct".

*Corresponding author

- In the limit framework, we demonstrate that the proposed method is essentially a variant of diffusion process operating on tensor product graph.

Compared with most previous works (Donoser and Bischof 2013) only focusing on iterative model, our key novelty lies in the regularization framework, which can well reveal the essence of diffusion process on tensor product graph. Therefore, we call the proposed method Regularized Diffusion Process (RDP). Moreover, RDP achieves state-of-the-art performances on various benchmark retrieval tasks, including face retrieval, shape retrieval and image retrieval.

Proposed Method

Diffusion processes usually model the data manifold as an weighted graph $\mathcal{G} = (X, W)$, where the vertices of the graph denote the data points $X = \{x_1, x_2, \ldots, x_N\}$. W is the graph adjacency matrix, with W_{ij} representing the pairwise similarity between x_i and x_j. Our aim is to learn a new similarity measure $A = \{A_{ij}\}_{1 \le i,j \le N}$, which varies sufficient smoothly along the graph $\mathcal{G}$.

Regularization Framework

Diffusion process on tensor product graph naturally involves high-order information, which is helpful to reveal the intrinsic similarities between objects as stated in (Donoser and Bischof 2013). However, its principle remains unclear. This section, which is the key contribution of this paper, gives this kind of diffusion process a manifold-based explanation.

We propose to obtain the new similarity measure A as the closed-form solution of the following optimization problem

$$\begin{aligned}\min_A \frac{1}{2} \sum_{i,j,k,l=1}^{N} & W_{ij} W_{kl} \left(\frac{A_{ki}}{\sqrt{D_{ii} D_{kk}}} - \frac{A_{lj}}{\sqrt{D_{jj} D_{ll}}}\right)^2 \\ & + \mu \sum_{k,i=1}^{N} (A_{ki} - Y_{ki})^2,\end{aligned} \tag{1}$$

where $\mu > 0$ is a regularization parameter. $Y \in \mathbb{R}^{N \times N}$ denotes the initial affinity values. D is a diagonal matrix with elements $D_{ii} = \sum_{j=1}^{N} W_{ij}$.

As presented in Eq. (1), the objective function of Regularized Diffusion Process (RDP) consists of two terms. The first term describes a kind of influence of the input similarity W on the learned similarity A. By analogy to Local and Global Consistency (LGC) (Zhou et al. 2004a), we will call it *smoothness term*. However, the inherent meanings of the two smoothness terms are quite different. As a semi-supervised learning algorithm, the smoothness term in LGC indicates that if x_k is similar to x_l (large W_{kl}), their probabilities of belonging to the same category should have a small difference. By contrast, the smoothness term of our method regularizes that if x_i is similar to x_j (large W_{ij}) and x_k is also similar to x_l (large W_{kl}) in the input similarity space, then the learned similarities A_{ki} and A_{lj} should be similar.

Manifold ranking (Zhou et al. 2004b) directly applies LGC to retrieval task by interpreting the probability of belonging to categories as the similarities between objects. Thus, one can find that the smoothness term in our method actually imposes a relaxed constraint against that in manifold ranking, *i.e.*, the individual object x_i is replaced by a pair of objects x_i and x_j with similarity W_{ij}. Consequently, to interrelate four tuples simultaneously, tensor product graph is a natural choice since each its node contains two points and each its edge records the relationship between four points.

In this sense, RDP can be expressed as **an improved version of manifold ranking with a relaxed smoothness term**. The second term in Eq. (1) is called *fitting term*, which has a similar meaning with that in LGC. It emphasizes that a good similarity measure should not change too much from its initial setup. Pervious works (Zhou et al. 2004b; Yang, Prasad, and Latecki 2013) take Y as identity matrix I, indicating that only the self-affinity of each node is fastened. In the experiments, we verify that this is not an optimal setup.

As demonstrated in Proposition 1, the objective function in Eq. (1) is equivalent to

$$J = vec(A)^{\mathrm{T}} \mathbb{L} vec(A) + \mu \| vec(A) - vec(Y) \|^2, \tag{2}$$

where $\mathbb{L} = I - \mathbb{S}$ and I is an identity matrix in appropriate size. $\mathbb{S} \in \mathbb{R}^{N^2 \times N^2}$ is the Kronecker product of $S = D^{-0.5} W D^{-0.5}$ with itself, that is $S \otimes S$. Operator $vec(\cdot)$ vectorizes an input matrix by stacking its columns one after the next. Its inverse operator, called reshape operator, is denoted as $vec(\cdot)^{-1}$.

In Eq. (2), $vec(A)$ can be deemed as a function, which gives each node in the tensor product graph (also a pair of nodes in the original graph) a real value to describe the pairwise relationship. $\mathbb{L}$ is the normalized graph Laplacian of the tensor product graph. So, the proposed method also aims at taking graph Laplacian as a smooth operator to preserve the local manifold structure as (Zhou et al. 2004a; 2004b). However, the key insight of our approach is **utilizing high order (tensor) graph Laplacian to smooth the pairwise relationship in the original graph**.

By taking the partial derivative of J with regard to $vec(A)$, we obtain

$$\frac{\partial J}{\partial vec(A)} = 2(I - \mathbb{S}) vec(A) + 2\mu \left(vec(A) - vec(Y)\right). \tag{3}$$

By setting Eq. (3) to zero, we have

$$vec(A) = \frac{\mu}{\mu + 1} (I - \frac{1}{\mu + 1} \mathbb{S})^{-1} vec(Y). \tag{4}$$

After applying vec^{-1} to both sides of Eq. (4) and setting $\alpha = \frac{1}{\mu + 1}$, we can obtain the closed-form solution as

$$A^* = (1 - \alpha) vec^{-1} \left((I - \alpha \mathbb{S})^{-1} vec(Y)\right). \tag{5}$$

Iterative Framework

Most pervious works (Donoser and Bischof 2013) are run in an iterative manner. In RDP, iteratively similarity propagation is done as follows:

$$A^{(t+1)} = \alpha S A^{(t)} S^{\mathrm{T}} + (1 - \alpha) Y. \tag{6}$$

To facilitate the iteration, we need to initialize $A^{(1)}$. Opposed to most variants of diffusion process summarized in (Donoser and Bischof 2013), we do not consider different types of initialization $A^{(1)}$, since our algorithm is guaranteed to converge to the same solution. The only difference is that the convergence speed is not the same with different initializations as demonstrated in the experiments.

In each iteration, similarity values are propagated on the affinity graph through the contextual information around both query nodes and database nodes, which is involved by pre-multiplying $A^{(t)}$ by S and post-multiplying $A^{(t)}$ by S^{T}. The survey paper (Donoser and Bischof 2013) experimentally proves that this kind of propagating approach can consistently achieve better retrieval performances than other kinds, such as propagating affinities only through the query nodes (*e.g.*, manifold ranking). In summary, our update scheme during each iteration is to propagate similarities on the affinity graph with probability $\alpha \in (0, 1)$ and go back to the initial affinities Y with probability $(1 - \alpha)$.

Now we prove the convergence of the iterative process. According to Lemma 1, by applying $vec(\cdot)$ to both sides of $A^{(t+1)} = \alpha S A^{(t)} S^{\mathrm{T}} + (1-\alpha)Y$, we can derive that

$$vec\left(A^{(t+1)}\right) = \alpha \mathbb{S} vec\left(A^{(t)}\right) + (1-\alpha) vec(Y). \quad (7)$$

By running the iteration for t times in Eq. (7), we can easily induce that

$$vec\left(A^{(t+1)}\right) = (\alpha\mathbb{S})^t vec\left(A^{(1)}\right) + (1-\alpha)\sum_{i=0}^{t-1}(\alpha\mathbb{S})^i vec(Y). \quad (8)$$

Since the spectral radius of S is no larger than 1, according to Lemma 2, the eigenvalues of $\mathbb{S}$ are also in [-1,1]. In addition, $0 < \alpha < 1$, then,

$$\begin{aligned} &\lim_{t\to\infty} (\alpha\mathbb{S})^t vec\left(A^{(1)}\right) = 0, \\ &\lim_{t\to\infty} \sum_{i=0}^{t-1}(\alpha\mathbb{S})^i vec(Y) = (I - \alpha\mathbb{S})^{-1} vec(Y). \end{aligned} \quad (9)$$

Hence, Eq. (8) converges to

$$\lim_{t\to\infty} vec\left(A^{(t+1)}\right) = (1-\alpha)(I-\alpha\mathbb{S})^{-1} vec(Y). \quad (10)$$

After applying vec^{-1} to both sides of Eq. (10), we obtain that the iteration converges to exactly the same solution in Eq. (5) obtained by the regularization framework of RDP. This provides a different yet important explantation of diffusion process on tensor product graph which well reveals its essence, *i.e.*, before convergence, the iterative similarity propagation is always decreasing the objective value of Eq. (1). Moreover, the generated equilibrium is independent from the initialization of $A^{(1)}$, which supports our previous claim that the initial value of $A^{(1)}$ is irrelevant in our algorithm.

Recall that the update scheme of label propagation (Zhou et al. 2004a) is

$$F^{(t+1)} = \alpha S F^{(t)} + (1-\alpha)Y. \quad (11)$$

One can find that Eq. (7) has very close relationship with label propagation formulated in Eq. (11). The key difference is that the classification function F in label propagation is replaced by $vec(A) \in \mathbb{R}^{N^2 \times 1}$ in Eq. (7). In this sense, RDP **converts retrieval task to a classification task solved by label propagation with only one category**. The classification is applied to a pair of objects (x_i, x_j), and the only category label is called "correct". If points x_i and x_j belong to the same category in the original similarity space, then the initial label of the pair (x_i, x_j) corresponding to category "correct" is 1, otherwise 0. Obviously, the initial label of the pair (x_i, x_i) is 1. The classification probability is recorded in the column vector $vec(A)$, *i.e.*, $vec\left(A^{(t)}\right)_{ij}$ represents the probability of (x_i, x_j) belonging to the category "correct" at t-iteration. When applying label propagation to retrieval task (Zhou et al. 2004b; Bai et al. 2010), the assumption is that each data point in the graph is taken as one individual category. Obviously, the correlation between data points (or "category") is ignored, especially those points which lie in the same clusters or sub-manifolds. In comparison, such a correlation is captured in RDP, since it directly focuses on analyzing the pairwise similarities between objects.

Limit Framework

In this section, we show that the proposed method can be also understood as diffusion process on tensor product graph.

As is known, a simple realization of diffusion process on a graph can be done by computing powers of the adjacency matrix of the graph. In this paper, the edge weights at time t can be obtained from $(\alpha S)^t$. Many previous works find (Lafon and Lee 2006; Zhang et al. 2012) that it is crucial to stop the diffusion process at a "right" time t. However, this is usually problematic especially when no labelled data is available. To remedy this, accumulating the results at different t is suggested (Lafon and Lee 2006; Bai et al. 2012). When $t \to \infty$, the limit of the accumulation is

$$\sum_{i=1}^{\infty} (\alpha S)^i = (I - \alpha S)^{-1}. \quad (12)$$

Since the Kronecker product of the adjacency matrix of the graph with itself is the adjacency matrix of tensor product graph, diffusion process on tensor product graph can be simply achieved by replacing S in Eq (12) with $\mathbb{S} = S \otimes S$, thus yielding

$$\mathbb{S}^* = \sum_{i=1}^{\infty} (\alpha\mathbb{S})^i = (I - \alpha\mathbb{S})^{-1}. \quad (13)$$

Note that $\mathbb{S}^* \in \mathbb{R}^{N^2 \times N^2}$, and our aim is to learn a new context-sensitive similarity $A^* \in \mathbb{R}^{N \times N}$. Therefore, we need to gather a portion of elements in $\mathbb{S}^*$ to substitute A^*. In this paper, it can be achieved by

$$A^* = vec^{-1}\left((I - \alpha\mathbb{S})^{-1} vec(Y)\right), \quad (14)$$

where $Y \in \mathbb{R}^{N \times N}$ determines the entry indices of the selected elements in $\mathbb{S}^*$. Meanwhile, since Y does not need to be a binary matrix (*i.e.*, with only 0 or 1), it also specifies a degree, to which extent the elements in $\mathbb{S}^*$ should be selected.

By multiplying a constant weight $(1-\alpha)$, Eq. (14) is identical to Eq. (5), which suggests that the proposed method is essentially a variant of diffusion process operating on tensor product graph.

Experiments

In this section, we evaluate the validity of the proposed Regularized Diffusion Process (RDP) with toy problems and real retrieval tasks. Directly using the closed-form expression in Eq. (5) requires $O(N^4)$ space complexity and $O(N^6)$ time complexity, which is impractical for large graphs. Throughout our experiments, the iterative solution is used, requiring $O(N^2)$ space complexity and $O(N^3)$ time complexity.

Since the proposed algorithm is guaranteed to converge into the same solution at different affinity initializations $A^{(1)}$ after a sufficient number of iterations, we set $A^{(1)}$ randomly and the iteration number to 100. As suggested by (Yang, Koknar-Tezel, and Latecki 2009), it is crucial to constrain diffusion process locally, *i.e.*, only propagating similarities through neighborhood structures. Therefore, graph sparsification is applied by only preserving edges within k nearest neighbors. Since graph sparsification destroys its symmetry, we re-symmetrize it via $W := \frac{W+W^{\mathrm{T}}}{2}$. The regularizer μ in Eq. (1) is set to 0.18.

Toy Problems

In this section, we present several toy examples to illustrate that RDP can well capture the geometry of manifold structures. The original data distribution is a two-spiral pattern with 200 data points, with each spiral having 100 points and one query point in cross shape. An ideal retrieval result is that points in a certain spiral have larger similarities values with the query points in this spiral than the query in the other spiral.

In Fig. 1(d) and Fig. 1(h), we present the retrieval results of manifold ranking (Zhou et al. 2004b) and RDP after convergence (100 iterations). The parameter setup of manifold ranking is the same as RDP, and Y is set to identity matrix I. We can find that the retrieval performance of RDP is significantly better than manifold ranking. Manifold ranking fails to reflect the intrinsic structures of two spirals probably because the two spirals are very close.

The retrieval results of manifold ranking and RDP at different iterations are also given in Fig. 1. Since kNN graph is used, there exist points that do not receive any similarity values at a small amount of iterations, which are marked in gray color. By comparing Fig. 1(a) with Fig 1(e), Fig. 1(b) with Fig 1(f), Fig. 1(c) with Fig 1(g) respectively, we can observe that RDP exhibits a much faster diffusion speed than manifold ranking due to the usage of high order information.

Face and Shape Retrieval

Following the survey paper (Donoser and Bischof 2013), we assess the effectiveness of the proposed RDP on ORL face dataset, YALE face dataset B (Georghiades, Belhumeur, and Kriegman 2001) and MPEG-7 shape dataset (Latecki, Lakämper, and Eckhardt 2000).

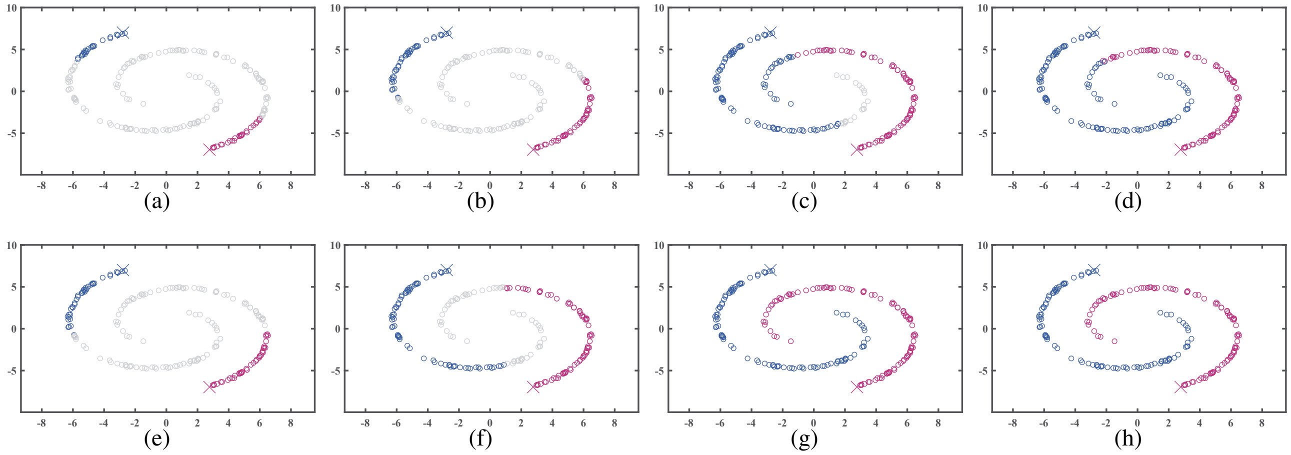
ORL dataset has 400 face images in total, divided into 40 categories with 10 images per category. As for YALE face dataset B, the same subset as (Jiang, Wang, and Tu 2011) is used. It contains 15 subjects under 11 different conditions. The same baselines and parameter setup ($k = 5$) as (Donoser and Bischof 2013) are used. In more detail, each image is down-sampled and normalized to 0-mean and 1-variance. Then, Euclidean distance between the vectorized representations is utilized to measure the pairwise dissimilarity. MPEG-7 dataset consists of $1,400$ silhouette images divided into 70 categories, where each category has 20 shapes. We do not use AIR descriptor (Gopalan, Turaga, and Chellappa 2010), since its performance on MPEG-7 dataset is already saturated. Instead, we turn to a more frequently-used shape descriptor, Inner Distance Shape Context (IDSC) (Ling and Jacobs 2007). k is set to 10.

The retrieval task is defined as follows: each image is used as query in turn and the rest images serve as the database. We use retrieval accuracy as the evaluation metric, which counts the recall within top-K returned results. In (Donoser and Bischof 2013), K is set to 15 on face datasets. Here we use three different values of K to get comprehensive comparisons with short ranking list ($K = 11$), medium length ranking list ($K = 15$) and long ranking list ($K = 20$) respectively. On MPEG-7 dataset, K is set to 40 conventionally. In the implementation of RDP, Y is set to identity matrix I or the original affinity matrix W.

In Table 1, the comparison with other representative algorithms is given, including Self Diffusion (SD) (Wang and Tu 2012), Locally Constrained Diffusion Process (LCDP) (Yang, Koknar-Tezel, and Latecki 2009), Tensor Product Graph (TPG) diffusion (Yang, Prasad, and Latecki 2013), Manifold Ranking (Zhou et al. 2004b) and Generic Diffusion Process (GDP) (Donoser and Bischof 2013). One should first pay attention to the fact that RDP with $Y = W$ achieves almost 1% percent performance boost compared with $Y = I$. The reason behind is that small Euclidean distances are meaningful in retrieval since they can well approximate the small geodesic distances along the manifold. After graph sparsification, W actually only records those small Euclidean distances. Consequently, we can prevent those meaningful relationship from vanishing by setting $Y = W$, thus yielding more reliable performances.

Among the compared methods, LCDP, TPG and GDP can be considered to work on tensor product graph. LCDP and GDP cannot guarantee the convergence of iteration. Although TPG is guaranteed to converge, it lacks a weighting mechanism to balance the contribution of smoothness term and fitting term. As can be seen, RDP outperforms these variants of diffusion process by a large margin. The performance gain is especially valuable, considering that GDP enumerates 72 variants of diffusion process (4 different affinity initializations, 6 different transition matrices and 3 different update schemes).

Despite those three diffusion processes, the closest work



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 1: The two crosses denote the query points. The retrieval results of manifold ranking are given when iteration number is 5 (a), 10 (b), 20 (c) and 100 (d). The retrieval results of RDP are given when iteration number is 5 (e), 10 (f), 20 (g) and 100 (h). The gray points have zero similarity values with both query points.

Methods	ORL dataset			YALE dataset			MPEG-7 dataset
	$K = 11$	$K = 15$	$K = 20$	$K = 11$	$K = 15$	$K = 20$	$K = 40$
Baseline	58.38	62.35	65.88	62.53	69.48	75.54	85.40
SD	67.65	71.67	74.90	67.05	71.46	77.08	83.09
LCDP	70.50	74.25	77.38	69.42	75.59	80.44	89.45
TPG	68.92	73.90	77.63	70.74	75.32	78.84	89.06
Manifold Ranking	72.67	77.05	80.10	64.52	70.85	77.08	89.26
Manifold Ranking*	73.45	77.58	80.53	71.07	76.91	80.61	92.61
GDP	72.67	77.42	80.58	71.24	77.30	80.83	90.96
RDP (Y=I)	74.17	78.53	81.42	71.24	78.07	81.21	93.77
RDP (Y=W)	75.08	79.27	81.88	71.85	78.24	82.09	93.78

Table 1: The comparison of retrieval accuracy (%) with other representative algorithms on YALE, ORL and MPEG-7 datasets. The best performances are marked in red and the second best performances are marked in blue.

to ours is manifold ranking. Besides the essential difference in the update scheme, the standard manifold ranking has three nuances: 1) it spreads affinities on fully-connected graph, while kNN graph used by RDP is proven more robust; 2) it avoids *self-reinforcement* by setting the diagonal elements of W to zero, while RDP does not; 3) it initializes $Y = I$, while it is demonstrated above that better performances can be achieved with $Y = W$. Hence, we also report the results of a modified version of manifold ranking using the three improvements, referred to Manifold Ranking* in Table 1. As the table presents, the modified manifold ranking achieves much better performances than its standard version. However, the inferior performances of both two versions of manifold ranking to RDP justify the conclusion that diffusion on tensor product graph is more robust in retrieval.

In addition, some other re-ranking algorithms also report the retrieval performances on MPEG-7 dataset using IDSC as the raw descriptor. Compared with them, the proposed RDP is better than Contextual Dissimilarity Measure (Jegou et al. 2010): 88.30, Index-Based Re-Ranking (Pedronette, Almeida, and da Silva Torres 2014): 91.56, Graph Transduction (Bai et al. 2010): 91.61, and RL-Sim Re-Ranking (Pedronette and Torres 2013): 92.62.

Natural Image Retrieval

We also evaluate the proposed algorithm on the widely-used Ukbench dataset. It consists of $2,550$ objects, with each object having 4 different view points. All $10,200$ images are both indexed as queries and database images. The evaluation metric is N-S score, which counts the average recall of the top-4 ranked images. For each image, we pass it into the pre-trained AlexNet and extract the activations of the 5-th convolutional layer and the 7-th fully connected layer. The two activations are L_2 normalized, and Euclidean distance is used to generate two pairwise dissimilarities. Their baseline performances are 3.44 and 3.65 respectively. The two learned similarities of RDP are linearly combined with equal weights. k is set to 4.

The performance comparison is given in Table 2. Besides diffusion processes (*e.g.*, LCMD (Luo et al. 2013) and TPG), we also compare other relevant re-ranking algorithms, including kNN Re-ranking (Shen et al. 2012), RNN Re-ranking (Qin et al. 2011), CDM (Jegou et al. 2010), Graph Fusion (Zhang et al. 2012), Query-Adaptive Late Fu-

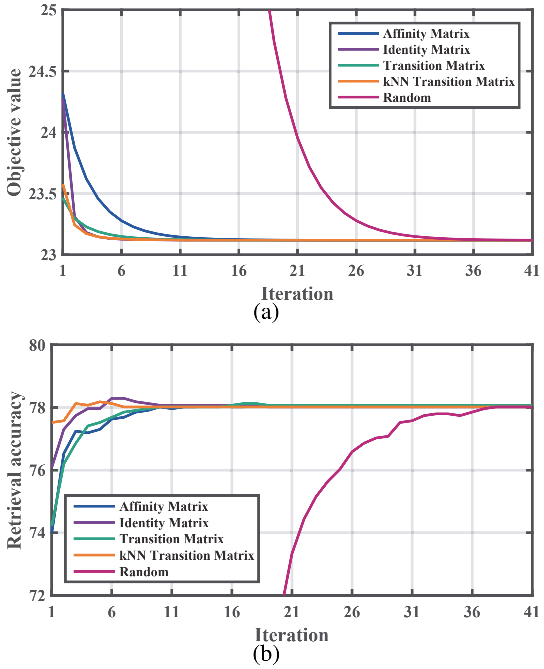


Figure 2: The objective value (a) and retrieval performance (b) of RDP as a function of iteration number on YALE dataset.

Methods	N-S score	Methods	N-S score
kNN Re-ranking	3.56	TPG	3.61
RNN Re-ranking	3.67	CDM	3.68
LCMD	3.70	Graph Fusion	3.77
QALF	3.84	SCA	3.86
Fan *et al.*	3.86	MSCE	3.88
RDP (Y=I)	3.93	RDP (Y=W)	3.94

Table 2: The comparison of N-S score on Ukbench dataset.

sion (QALF) (Zheng et al. 2015), Sparse Contextual Activation (SCA) (Bai and Bai 2016), Fan *et al.* (Yang, Matei, and Davis 2015) and MSCE (Zheng et al. 2016). As Table 2 shows, RDP outperforms the previous state-of-the-art remarkably.

Discussion

In Fig. 2, the influence of iteration number on the objective value defined in Eq. (1) and retrieval performance ($K = 15$) with YALE dataset are given. Here we set $Y = I$. We use five types of initialization $A^{(1)}$, among which the first 4 types are used in generic diffusion process (Donoser and Bischof 2013) and the last one is random values (default setting in our method). A first glance at Fig. 2(a) shows that when propagating affinities on the graph iteratively, RDP tries to minimize the objective function in Eq. (1) until convergence. It reveals the essential behavior of diffusion process on tensor product graph at each iteration. Second, it is observed that different initializations of $A^{(1)}$ will reach the same equilibrium with different convergence speed. Generally, starting from kNN transition matrix leads to the fastest convergence speed while random initializations is the slowest one. Third, the retrieval performances are exactly the same at equilibrium as presented in Fig. 2(b). It demonstrates the robustness of RDP as opposed to the variants summarized in (Donoser and Bischof 2013) that require a careful initialization of $A^{(1)}$.

Conclusions

In this paper, we concern on a family of algorithms called diffusion process. Recent studies have demonstrated that diffusion process on tensor product graph is more capable of retrieval task, but lack theoretical explanations on its essence. By proposing a new variant called Regularized Diffusion Process (RDP), we provide three insightful conclusions to expose the principle of this kind of diffusion process. With the given regularization framework, one can clearly observe that RDP is minimizing a kind of relationships among four tuples at each iteration, so that high order information provided by tensor product graph is necessary.

Recall that RDP converts retrieval task to classification task solved by label propagation with only "correct" category label. Thus, one can easily extend it by adding another category labelled with "incorrect" to push these unmatched object pairs far away during iteration. It can be expected that the retrieval performances will be improved further by combining the two kinds of similarity propagation. Moreover, we will also study how to apply RDP to other related topics, such as point matching (Ma et al. 2014). Those can be very promising directions that can be investigated in the future.

Acknowledgements

This work was supported in part by NSFC 61573160, NSFC 61429201, NSF IIS-1302164 and China Scholarship Council; and in part to Dr. Qi Tian by ARO grants W911NF-15-1-0290 and Faculty Research Gift Awards by NEC Laboratories of America and Blippar.

Appendix

Lemma 1. *Assume that A, B and C are three matrices in appropriate sizes, then $vec(ABC^{\mathrm{T}}) = (C \otimes A)vec(B)$.*

Lemma 2. *Assume that A and B are two square matrices of size N. Let λ_A^i $(1 \leq i \leq N)$ be the eigenvalues of A and λ_B^j $(1 \leq j \leq N)$ be those of B, then the eigenvalues of $A \otimes B$ are $\lambda_A^i \lambda_B^j$ $(1 \leq i, j \leq N)$.*

Proposition 1. *The objective function in Eq.* (1) *is equivalent to Eq.* (2).

Proof. The equivalence between the right term of Eq. (1) and Eq. (2) is obvious. Hence, we only focus on the left term below.

Define $\alpha \equiv N(i-1)+k$ and $\beta \equiv N(j-1)+l$. Then the left term of Eq. (1) can be transformed into

$$\begin{aligned}
&\frac{1}{2}\sum_{\alpha,\beta=1}^{N^2} \mathbb{W}_{\alpha\beta}\left(\frac{vec(A)_\alpha}{\sqrt{\mathbb{D}_{\alpha\alpha}}} - \frac{vec(A)_\beta}{\sqrt{\mathbb{D}_{\beta\beta}}}\right)^2 \\
=&\sum_{\alpha,\beta=1}^{N^2} \mathbb{W}_{\alpha\beta}\frac{{vec(A)_\alpha}^2}{\mathbb{D}_{\alpha\alpha}} - \sum_{\alpha,\beta=1}^{N^2} vec(A)_\alpha \frac{\mathbb{W}_{\alpha\beta}}{\sqrt{\mathbb{D}_{\alpha\alpha}\mathbb{D}_{\beta\beta}}} vec(A)_\beta \\
=&\sum_{\alpha=1}^{N^2} {vec(A)_\alpha}^2 - vec(A)^{\mathrm{T}}\mathbb{D}^{-0.5}\mathbb{W}\mathbb{D}^{-0.5}vec(A) \\
=&vec(A)^{\mathrm{T}}\left(I - \mathbb{D}^{-0.5}\mathbb{W}\mathbb{D}^{-0.5}\right)vec(A),
\end{aligned} \tag{15}$$

where $\mathbb{W} = W \otimes W \in \mathbb{R}^{N^2 \times N^2}$ and $\mathbb{D} = D \otimes D \in \mathbb{R}^{N^2 \times N^2}$. Note that the symmetry property of $\mathbb{W}$ is used in Eq. (15). Compared with Eq. (2), we need to present $\mathbb{S} = \mathbb{D}^{-0.5}\mathbb{W}\mathbb{D}^{-0.5}$ now. It holds, since

$$\begin{aligned}
\mathbb{S}_{\alpha\beta} &= S_{ij}S_{kl} = D_{ii}^{-0.5}W_{ij}D_{jj}^{-0.5}D_{kk}^{-0.5}W_{kl}D_{ll}^{-0.5} \\
&= \mathbb{D}_{\alpha\alpha}^{-0.5}\mathbb{W}_{\alpha\beta}\mathbb{D}_{\beta\beta}^{-0.5}.
\end{aligned} \tag{16}$$

The proof is complete. □

References

Bai, S., and Bai, X. 2016. Sparse contextual activation for efficient visual re-ranking. *TIP* 25(3):1056–1069.

Bai, X.; Yang, X.; Latecki, L. J.; Liu, W.; and Tu, Z. 2010. Learning context-sensitive shape similarity by graph transduction. *TPAMI* 32(5):861–874.

Bai, X.; Wang, B.; Yao, C.; Liu, W.; and Tu, Z. 2012. Co-transduction for shape retrieval. *TIP* 21(5):2747–2757.

Chen, Y.; Li, X.; Dick, A.; and Hill, R. 2014. Ranking consistency for image matching and object retrieval. *Pattern Recognition* 47(3):1349–1360.

Donoser, M., and Bischof, H. 2013. Diffusion processes for retrieval revisited. In *CVPR*, 1320–1327.

Georghiades, A. S.; Belhumeur, P. N.; and Kriegman, D. J. 2001. From few to many: Illumination cone models for face recognition under variable lighting and pose. *TPAMI* 23(6):643–660.

Gopalan, R.; Turaga, P.; and Chellappa, R. 2010. Articulation-invariant representation of non-planar shapes. In *ECCV*, 286–299.

Jegou, H.; Schmid, C.; Harzallah, H.; and Verbeek, J. 2010. Accurate image search using the contextual dissimilarity measure. *TPAMI* 32(1):2–11.

Jiang, J.; Wang, B.; and Tu, Z. 2011. Unsupervised metric learning by self-smoothing operator. In *ICCV*, 794–801.

Kontschieder, P.; Donoser, M.; and Bischof, H. 2009. Beyond pairwise shape similarity analysis. In *ACCV*, 655–666.

Lafon, S., and Lee, A. B. 2006. Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *TPAMI* 28(9):1393–1403.

Latecki, L. J.; Lakämper, R.; and Eckhardt, U. 2000. Shape descriptors for non-rigid shapes with a single closed contour. In *CVPR*, 424–429.

Ling, H., and Jacobs, D. W. 2007. Shape classification using the inner-distance. *TPAMI* 29(2):286–299.

Luo, L.; Shen, C.; Zhang, C.; and van den Hengel, A. 2013. Shape similarity analysis by self-tuning locally constrained mixed-diffusion. *TMM* 15(5):1174–1183.

Ma, J.; Zhao, J.; Tian, J.; Yuille, A. L.; and Tu, Z. 2014. Robust point matching via vector field consensus. *TIP* 23(4):1706–1721.

Page, L.; Brin, S.; Motwani, R.; and Winograd, T. 1999. The pagerank citation ranking: bringing order to the web.

Pedronette, D. C. G.; Almeida, J.; and da Silva Torres, R. 2014. A scalable re-ranking method for content-based image retrieval. *Information Sciences* 265:91–104.

Pedronette, D. C. G., and Torres, R. d. S. 2013. Image re-ranking and rank aggregation based on similarity of ranked lists. *Pattern Recognition* 46(8):2350–2360.

Qin, D.; Gammeter, S.; Bossard, L.; Quack, T.; and Van Gool, L. 2011. Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors. In *CVPR*, 777–784.

Shen, X.; Lin, Z.; Brandt, J.; Avidan, S.; and Wu, Y. 2012. Object retrieval and localization with spatially-constrained similarity measure and k-nn re-ranking. In *CVPR*, 3013–3020.

Wang, B., and Tu, Z. 2012. Affinity learning via self-diffusion for image segmentation and clustering. In *CVPR*, 2312–2319.

Yang, X.; Koknar-Tezel, S.; and Latecki, L. J. 2009. Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval. In *CVPR*, 357–364.

Yang, F.; Matei, B.; and Davis, L. S. 2015. Re-ranking by multi-feature fusion with diffusion for image retrieval. In *WACV*, 572–579.

Yang, X.; Prasad, L.; and Latecki, L. J. 2013. Affinity learning with diffusion on tensor product graph. *TPAMI* 35(1):28–38.

Zhang, S.; Yang, M.; Cour, T.; Yu, K.; and Metaxas, D. N. 2012. Query specific fusion for image retrieval. In *ECCV*, 660–673.

Zheng, L.; Wang, S.; Tian, L.; He, F.; Liu, Z.; and Tian, Q. 2015. Query-adaptive late fusion for image search and person re-identification. In *CVPR*, 1741–1750.

Zheng, L.; Wang, S.; Wang, J.; and Tian, Q. 2016. Accurate image search with multi-scale contextual evidences. *IJCV* 1–13.

Zhou, D.; Bousquet, O.; Lal, T. N.; Weston, J.; and Schölkopf, B. 2004a. Learning with local and global consistency. In *NIPS*, 321–328.

Zhou, D.; Weston, J.; Gretton, A.; Bousquet, O.; and Schölkopf, B. 2004b. Ranking on data manifolds. In *NIPS*, 169–176.