Pattern Recognition 44 (2011) 2367–2374

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

# Learning context-sensitive similarity by shortest path propagation

Jingyan Wang [a], Yongping Li [a,*], Xiang Bai [b], Ying Zhang [a], Chao Wang [c], Ning Tang [a]

[a] Shanghai Institute of Applied Physics, Chinese Academy of Sciences, 2019 Jialuo Road, Shanghai 201800, PR China
[b] Department of Electronics and Information Engineering, Huazhong University of Science and Technology, 1037 Luoyu Road, Wuhan, Hubei Province 430074, PR China
[c] OGI School of Science and Engineering, Oregon Health & Science University (OHSU), Beaverton, OR 97006, US

ARTICLE INFO

*Article history:*
Received 21 June 2010
Received in revised form
2 February 2011
Accepted 5 February 2011
Available online 12 February 2011

*Keywords:*
Shape retrieval
Contextual similarity learning
Graph transduction
Shortest path propagation

ABSTRACT

In this paper, we introduce a novel shape/object retrieval algorithm shortest path propagation (SSP). Given a query object $q$ and a target database object $p$, we explicitly find the shortest path between them in the distance manifold of the database objects. Then a new distance measure between $q$ and $p$ is learned based on the database objects on the shortest path to replace the original distance measure. The promising results on both MEPG-7 shape dataset and a protein dataset demonstrate that our method can significantly improve the ranking of the object retrieval.



## 1. Introduction

In this paper, we address the problem of finding nearest objects (shape, images, etc.) of the same class in a database, which is called object retrieval/ranking. Finding nearest neighbors is a general idea that underlies many artificial intelligence tasks, including machine learning, data mining, natural language understanding, and information retrieval. Shape retrieval is a very critical problem in computer vision and bioinformatics. Most of the shape retrieval methods [1,15,17–25] focused on pairwise shape similarity measure. Yet, these methods ignore the fact that some differences are more relevant while other differences are less relevant for shape similarity. No matter how good a shape similarity measure is, this problem must be addressed if we want to obtain a better performance. There is now a growing interest in learning context-sensitive or contextual similarity [1,14] from a collection of pairwise similarities among the database (context) objects, which uses the context information to improve the similarity between the query object and the target object.

As pointed out by [1], a "good" similarity between a query $q$ and a known object $p$ should describe the relationship between $q$ and $p$ in the context of the database. The intuition in [1] is to compute a new similarity $s$ for a given similarity measure $s_0$ by a graph transduction method named label propagation (LP). In their method, each query object is considered in the context of the objects of the database, and a new similarity function is learned by implicitly finding the shortest paths on the similarity manifold formed by the similarities among the known objects and the query object. With the similar manner of [1], locally constrained diffusion process (LDCP) [15] was developed to explore the context information, which achieves a better performance on several shape benchmarks than the LP method [1]. Another contribution of [15] is that some ghost objects are also constructed and added into the similarities space to enhance the context information. Recently, Egozi et al. [16] proposed a novel contextual similarity function—meta similarity (MS). It characterizes a given object by its similarity to its KNN objects. One advantage of MS is the low time complexity, as it does not require propagating the similarities. An interesting distance learning method called contextual dissimilarity measure (CDM) [14] is motivated by the observation that a good ranking is usually asymmetric in image search. CDM can significantly improve the distance measure by bag-of-features, but its improvement on shape retrieval is not so obvious, for the shape distance measures has a different distribution from bag-of-features. (We will show the result of CDM on shape dataset in Section 3.)

The methods above-mentioned explore the context information but without explicitly learning the shortest path on the distance manifold, so some redundant context may influence the final results. For example, [1] learns the new similarity $s(p,q)$ by computing the weighted mean of similarities between $q$ and all the other objects (reference objects) in the database, which may bring much redundant information and reduce the influence of the relevant objects. An alternative way could be to replace reference objects with only the neighbors of $p$, but this also

* Corresponding author.
*E-mail addresses:* wangjingyan@sinap.ac.cn (J. Wang), ypli@sinap.ac.cn (Y. Li), xiang.bai@gmail.com (X. Bai), zhangying1@sinap.ac.cn (Y. Zhang), wangcha@csee.ogi.edu (C. Wang), tangning08@gmail.com (N. Tang).

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.02.007

ignores the relationship between $p$ and $q$ by only considering $p$. Thus, the key to computing a "good" similarity $s(q,p)$ depends how to select the *reference object set* $R(q)$ which is relevant to the query and the target object. Intuitively, for a query object $q$, the similarity $s(q,p)$ can be weighted by averaging $s_0(q,p_R), p_R \in R(q)$. The size of $R(q)$ need not to be very large, as long as $R(q)$ can sufficiently describe the relationship between $p$ and $q$.

In this paper, we focus on this problem of selecting an appropriate reference object set for learning a new similarity between $q$ and $p$. We select the nodes in the shortest path between $p$ and $q$ explicitly, which are directly relevant to the query object $q$ and the target object $p$. Then, the new similarity between $q$ and $p$ is computed by propagating the similarities along these reference nodes. Thus, we call the proposed method shortest path propagation (SSP).

The remainder of this paper is organized as follows. Section 2 gives a brief introduction to the graph transduction based context-sensitive similarity framework and the definition of reference objects set. Then, the proposed SSP algorithm is presented. Several experiments are performed in Section 3. Finally, Section 4 concludes the paper.

## 2. Shortest path similarity propagation

We first describe the classical setting of similarity-based ranking. It has been applied to many retrieval scenarios like key word, document, image, and protein retrieval. Given a set of objects $X=\{x_1,\ldots,x_n\}$ and a similarity function $sim = X \times X \rightarrow R^2$ that assigns a similarity value (a positive value) between each pair of objects, we assume that $x_1$ is a query object (e.g., a query protein shape), and $\{x_2,\ldots,x_n\}$ is a set of known database objects (or a training set). Then, by sorting the values $sim(x_1,x_i)$ in decreasing order for $i=2,\ldots,n$, we obtain a ranking of database objects according to their similarity to the query, i.e., the most similar database object has the highest value and is listed first.

### 2.1. Graph transduction based context-sensitive similarity

Before introducing our method, we review graph transduction (GT) [1] simply. Bai et al. [1] introduces a method to learn a new similarity function $sim_T$ that drastically improves the retrieval results of $sim$ for the given query $x_1$ by using graph transduction. First, a KNN graph $G=\{V,E,W\}$ is constructed. The node set $V$ corresponds to $n$ objects in $X$, $E$ is the edge set, and

$$E=\{(x_i,x_j)|x_j \in KNN(x_i)\} \tag{1}$$

means that the neighboring size $K$ must be large enough to make sure that $G$ is a connected graph.

Let $W=\{w_{ij}\}$ $(i,j=1,\ldots,n)$ be a similarity matrix, which is also called an affinity matrix, and $w_{ij}=sim(x_i,x_j)$ be the weight of edge between $x_i$ and $x_j$. Let $D=(d_{ij})$ be a distance matrix computed by some distance function, which is converted into $w_{ij}$ using a Gaussian kernel:

$$w_{ij}=\exp\left(-\frac{d_{ij}^2}{\sigma_{ij}^2}\right) \tag{2}$$

Kernel size $\sigma_{ij}$ is defined based on the mean distance to $K$-nearest neighborhoods [2]:

$$\sigma_{ij}=\alpha \cdot mean(\{knnd(x_i),knnd(x_j)\}), \tag{3}$$

where $mean(\{knnd(x_i),knnd(x_j)\})$ denotes the mean distance of the $K$-nearest neighbor distance of the sample $x_i$ ,$x_j$ and $\alpha$ is an extra parameter. Both $K$ and $\alpha$ are determined empirically. They are selected by 2-cross validation. The retrieval score or classification accuracy is calculated for each different pair of parameters, and the best parameter pair with the highest score will be selected. As the test dataset in 2-cross validation is half of the whole dataset, we double the $K$ for the whole dataset. An $n \times n$ probabilistic transition matrix $P$ as a row wise normalized matrix $W$ is also defined

$$P_{ij}=\frac{w_{ij}}{\sum_{k=1}^{n} w_{ik}}. \tag{4}$$

Then a new similarity measure $s$ is sought. Since $s$ only needs to be defined as similarity of other elements to query $x_1$, let $f(x_i)=s(x_1,x_i)$, for $i=1,\ldots,n$. A key function in [1] is $f$ and it satisfies

$$f(x_i)=\sum_{j=1}^{n} P_{ij} f(x_j) \tag{5}$$

where $P_{ij}$ is the probability of transit from nodes $i$ to $j$.

Thus, the similarity of $x_i$ to the query $x_1$, expressed as $f(x_i)$, is a weighted average over *all other database objects*, where the weights sum to 1 and are proportional to the similarity of the other database objects to $x_i$.

The solution to (5) is implemented by the following recursive procedure:

$$f_t(x_i)=\sum_{j=1}^{n} P_{ij} f_{t-1}(x_j) \tag{6}$$

for $i=2,\ldots,n$, and we set

$$f_t(x_1)=1. \tag{7}$$

We define a sequence of newly learned similarity functions restricted to $x_1$ as

$$sim_t(x_1,x_i)=f(x_i). \tag{8}$$

Since the goal is ranking the database objects according to their similarity to the query, we stop the computation after a suitable number of iterations $t=T$. Like the usual practice with iterative processes that are guaranteed to converge [1], the computation halts if the difference $|f_t-f_{t-1}|$ becomes very small. The convergence can be observed in Fig. 7(b) in [1], where the differences $|f_t-f_{t-1}|$ as a function of $t$ are plotted. In particular, the curve decreases very slowly after about 1000 iterations, and after 5000 iterations, they are nearly constant.

### 2.2. Shortest path reference set

Let $x_t$ denote a target object in the database, and our goal is to compute an accurate similarity $s(x_1,x_t)$ between query $x_1$ and target $x_t$. Being the same as the formula (5), $s(x_1,x_t)=f(x_t)$.

Now we give the definition of *reference objects*, which is denoted as $R(x_t)$. The similarity $f(x_t)$ between $x_t$ and the query $x_1$ should be the weighted average over $R(x_t)$. Apparently, in [1,15] for each target $x_t$ of the database, $R(x_t)=X(x_t \in X)$ are all the database objects. With the reference objects, Eq. (5) can be rewritten as

$$f(x_t)=\sum_{j \in R(x_t)} P_{tj} f(x_j). \tag{9}$$

We assume that not all the objects are needed to estimate such a similarity, and a few objects including $x_t$ are enough as the reference objects. The $K$ nearest neighbors of the target can be included in such reference objects together with the query $x_1$ and the target $x_t$ itself, which is denoted as local reference set $R^L(x_t)$:

$$R^L(x_t)=\{x_1,x_t\} \cup \{x_j|x_j \in KNN(x_t),\ j=1,\ldots,K\}. \tag{10}$$

$R^L(x_t)$ only includes the neighboring objects of $x_t$ as shown in Fig. 1(c). We call the reference set $R(x_t)$ used in Eq. (9) global reference set $R^G$, since $R^G(x_t)=X$ includes all the database object as shown in Fig. 1(b). Some experiments implemented by Bai et al. [1]

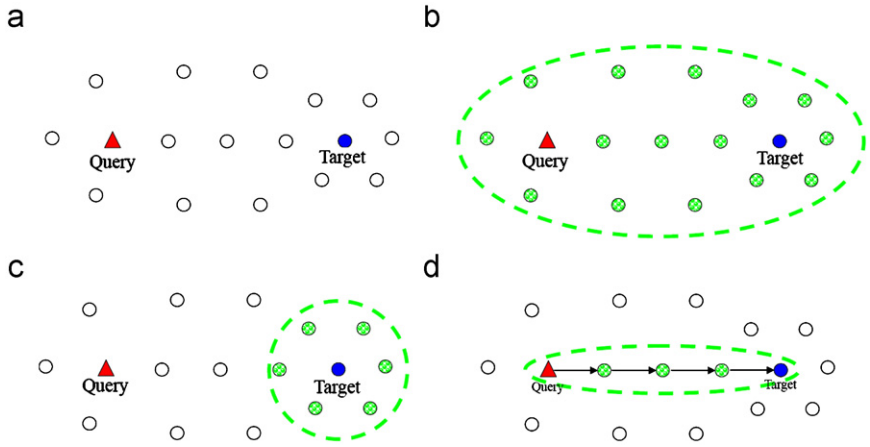


**Fig. 1.** Reference object sets (green circles within the dotted line) used to estimate the similarity between query $x_q$ (red triangle) and target $x_t$ (blue circle): (a) original sets; (b) global reference object set $R^G(x_t)$ containing all the objects in the database; (c) local reference object set $R^L(x_t)$ containing only the objects near $x_t$; (d) extended shortest path reference object set $R^{rESP}(x_t)$ containing the objects near the nodes in the shortest path between $x_q$ and $x_t$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)


**Fig. 2.** A key idea of the proposed distance learning is to replace the original shape distance between (a) and (e) with a distance propagated along a "bridge" or "chain" between (a) and (e). One such path is (a)–(e) in this figure.

show that such $R^L(x_t)$ is inferior to $R^G(x_t)$. A possible reason is that local reference set which only includes the neighboring objects of $x_t$ may neglect the relation between query $x_1$ and target $x_t$.

To solve this problem, we should look for an effective $R(x_t)$, which should satisfy the following conditions:

1. The size should not be too large. There are two reasons for this requirement:
   (a) If the local reference object set $R^L(x_t)$ is too large, (an extreme case is when $K=n-1$, which means all the objects in $X$ are included in $KNN(x_t)$, $R^L(x_t)$ is identical to $R^G(x_t)$). $R^L(x_t)$ will contain many redundant or irrelevant reference objects as a global one. This will reduce the impact of the key reference object, which is a limitation of $R^G(x_t)$.
   (b) In [1], one training object is also in other ones's reference object set, so after the procedure of iterations, all the similarities of the objects to the query is computed as $s(x_1,x_i)=f(x_i)$; but the local $R^L(x_t)$ does not contain all the other objects, after the iterations, only the $s(x_1,x_t)=f(x_t)$ can be returned as similarity between $x_1$ and $x_t$. When the size is too large, the iterations for each object will not be acceptable in time complexity.
2. The relation between $x_1$ and $x_t$ should also be reflected. Apparently, $R^G(x_t)$ contains all the objects so we can make sure that it can satisfy such requirement; however, $R^L(x_t)$ is defined only according to $x_t$, which has a bias of $x_t$, while neglect that there exist a group of objects which can build a bridge between $x_1$ and $x_t$. For example, in Fig. 2, two shape objects (a) and (e) look quite different, but we can find several objects to connect them together as a chain (a)–(e), so that we can discover that they are similar.

According to these two requirements, we can find that $R^L(x_t)$ can fulfill the first requirement but cannot satisfy the second one, while $R^G(x_t)$ is on the contrary. Therefore, we try to set $R(x_t)$ as the set of nodes belonging to the shortest path in $G$ between $x_1$ and $x_t$ with their neighbors.

Our idea is to propagate the similarity along the shortest path from $x_i$ to $x_t$. Here we denote the nodes on the shortest path between $x_1$ and $x_t$ as a set of objects $SP(x_1,x_t)$. The corresponding $R(x_t)$ is called shortest path reference object set, which is denoted as

$$R^{SP}(x_t)=SP(x_1,x_t). \tag{11}$$

To obtain more context information around the shortest path, we extend $R^{SP}(x_t)$ so that it also includes the neighboring objects.

Let $d^{SP}(x_i,x_j)$ denote the shortest path distance between two nodes $x_i$ and $x_j$. The shortest path distance between node $x_i$ and a node set $A\in X$ is defined as

$$d^{SP}(x_i,A)=\min_{x_k\in A} d^{SP}(x_i,x_k). \tag{12}$$

Similar to [5], we consider a sequence of node sets $\{X_r^{SP}(x_t)\}_{r=0}^{R}$, such that

$$R^{SP}(x_t)=X_0^{SP}(x_t)\in X_1^{SP}(x_t)\in X_2^{SP}(x_t)\in\cdots\in X_R^{SP}(x_t)=X. \tag{13}$$

The sets $X_r^{SP}(x_t)$ contain all the nodes that are within a certain distance to the nodes of the shortest path $R^{SP}(x_t)$ and are formally defined as

$$X_r^{SP}(x_t)=\{x_i|d^{SP}(x_i,R^{SP}(x_t))\le q_r\}, \tag{14}$$

where $q_r, r=0,\ldots,R$ is a pre-specified non-negative threshold with $q_0=0$ and $q_R=\max_{x_i\in X}(d^{SP}(x_i,X))$, so that $q_r=(r/R)q_R$. An example of the sequence of node sets $\{X_r^{SP}(x_t)\}_{r=0}^{R}$ is shown in Fig. 3.

Given an $r$, we can select an object set from the sequence of sets in (13) as a new reference object set $R(x_t)=X_r^{SP}(x_t)$. We call it "$r$-extended shortest path reference object set", denoted as $R^{rESP}(x_t)=X_r^{SP}(x_t)$. We will use $R^{rESP}(x_t)$ to learn the new similarity function by the same way as in Section 2.1.

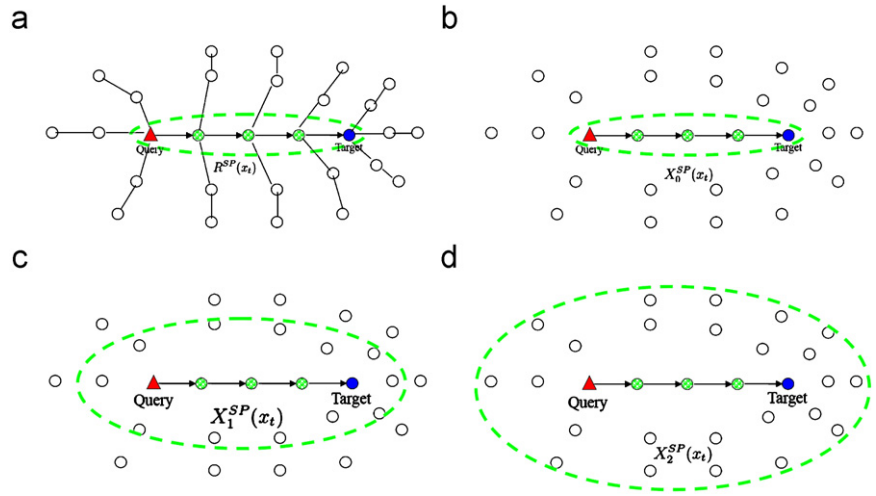


**Fig. 3.** Sequence of sets (green circles within the dotted line): (a) $R^{SP}(x_t)$ = (b) $X_0^{SP}(x_t) \in$ (c) $X_1^{SP}(x_t) \in$ (d) $X_2^{SP}(x_t) = X$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

### 2.3. Similarity learning algorithm

Given a query $x_1$ and a target $x_t$, we try to learn their similarity using the reference object set $R^{rESP}(x_t)$. The first problem is to find the shortest path $SP(x_1,x_t)$ between query $x_1$ and a target $x_t$. Given one node $x_1$ in the graph, Dijkstra algorithm [3] can be used to find the shortest paths $SP(x_1,x_i)$ between any node $x_i, i=2,\ldots,n$. However, Dijkstra algorithm is quite time consuming with the time complexity of $O(n^2)$. Considering an inline query procedure and each object in the database should be compared, the time complexity of indexing one query against all the database objects is $O(n^3)$, which is unacceptable for an inline retrieval system, especially when the database's size is large.

Here we propose an alternative method to obtain $SP(x_1,x_i)$, $i=2,\ldots,n$, which has two steps:

1. *Off-line learning phase*: a pairwise shortest path $SP(x_i,x_j)$, $i,j=2,\ldots,n$ between any two nodes of the database is extracted with Floyd–Warshall algorithm [4], which is a pairwise shortest path routing algorithm. (Though the time complexity of Floyd–Warshall algorithm is $O(n^3)$, it is acceptable for the off-line procedure.)
2. *Online querying phase*: Given a query $x_1$ and a target $x_t$, we search the nearest neighbor of $x_1$: $x_{NN}=NN(x_1)$, and then the shortest path $SP(x_{NN},x_t)$ between $x_{NN}$ and $x_t$ is considered as an approximation of $SP(x_1,x_t)$,

$$SP'(x_1,x_t) \approx x_1 \cup SP(x_{NN},x_t). \tag{15}$$

This method allows for the shortest path routing algorithm off-line, and the online search procedure only finds the nearest neighbor $x_{NN}$ of $x_1$ with a low time complexity of $O(n)$.

The probabilistic transition matrix $P$ in (9) is normalized using $R^G(x_t)$, which is not fit for $R^{rESP}(x_t)$. To solve this problem, we define the smoother probabilistic transition matrix $P^r(x_t)$, whose $(i,j)$th element is given by

$$P_{ij}^r(x_t) = \frac{w_{ij}}{\sum_{k \in R^{rESP}(x_t)} w_{ik}}, \tag{16}$$

where $P^r_{x_t}$ is a $|R^{rESP}(x_t)| \times |R^{rESP}(x_t)|$ stochastic matrix, i.e., all its rows sum up to 1. Using this transition matrix $P^r(x_t)$, we can give the similarity learning algorithm as in Algorithm 1.

**Algorithm 1.** Shortest Path Similarity Propagation Algorithm.

**Require:** The $n \times n$ similarity matrix $w_{ij} = sim(x_i,x_j)$ for $i,j=1,\ldots,n$;
**Require:** The query object $x_1$;
**Require:** The target object $x_t$.
Initialize $f_1(x_1)=1$, and $f_1(x_i)=0, i=2,\ldots,n$, $k=1$;
Find the approximation of $SP(x_1,x_t)$ using (15);
Construct the reference object set $R^{rESP}(x_t)$ using (14);
Compute the smoother probabilistic transition matrix $P^r(x_t)$ for $R^{rESP}(x_t)$ using (16);
**while** $k<T$ **do**
  **for** $i \in R^{rESP}(x_t)$ **do**
    $f_{k+1}(x_i) = \sum_{j \in R^{rESP}(x_t)} P^r_{ij}(x_t) f_k(x_j)$
  **end for**
  $f_{k+1}(x_1)=1$;
  $k$++;
**end while**
Output the learned new similarity $f(x_t)=s(x_1,x_t)$ between query $x_1$ and target $x_t$.

## 3. Experiment results

In this section, we show that the proposed approach can improve the performance of retrieval problems. We adopt a shape dataset (MPEG-7 shape dataset) and a 3D protein structure dataset (FSSP/DALI protein dataset) for the experimental evaluation.

### 3.1. The experiments on MPEG-7 shape dataset

MPEG-7 CE Shape-1 Part-B dataset consists of 1400 silhouette images grouped into 70 classes. Each class has 20 different shapes. Some example images of this dataset are shown in Fig. 4. Similar to graph transduction [1], we use IDSC [10] as our baseline distance measure. For MPEG-7 shape dataset, the retrieval performance is often measured with the bulls-eyes test: Each shape in the database is compared to all the other shapes, and the number of shapes from the same category among the 40 most similar shapes is reported. The bulls-eye score is the ratio of the total number of

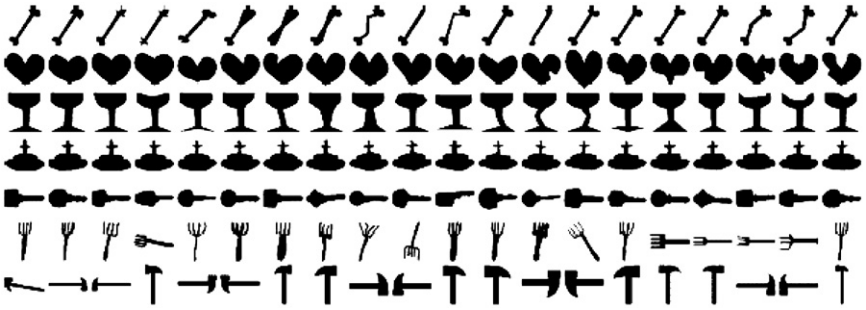

**Fig. 4.** Some images of MPEG-7 CE Shape-1 Part-B dataset.

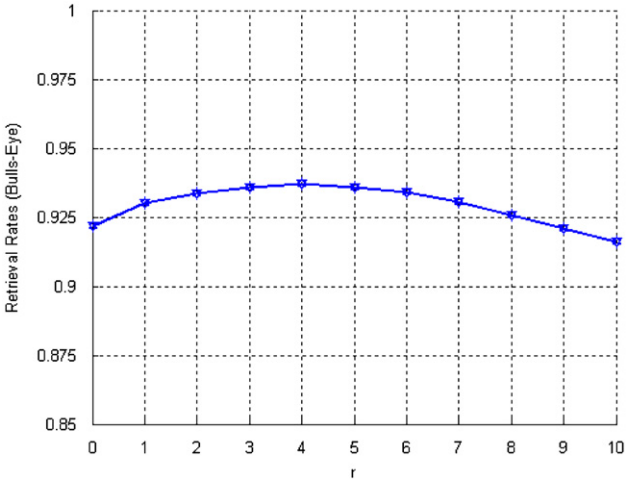


**Fig. 5.** We plot the bulls-eye scores obtained by SSP algorithm for several different sequential thresholds $r$ on the MPEG-7 dataset.

shapes from the same category to the highest possible number (which is $20 \times 1400$).

We use the following parameters to construct the affinity matrix: $\alpha = 0.25$ and the neighborhood size $K$ is 14, which are the same to the settings of [1]. For a fair comparison, we select $T=5000$ as stop condition for SSP, which is also same to the setting of graph transduction [1]. Since [1] has proposed a supervised learning method for determining these parameters in details, we no longer review it here.

An important parameter of the proposed method is $q_r$, hence we do the experiments to show the evolution of the performances according to this parameter. In our experiment, we set $R=10$, so that the whole dataset can be divided into 10 sequential sets. Then $q_r = (r/R)q_R = r/10q_R, r = \{0,1,\ldots,10\}$, and we can select a sequential set $X_r^{SP}(x_t)$ as the reference set by choosing a special $r$. The bulls-eye score on MPEG-7 dataset with different $r$ is given in Fig. 5. A low $q_r=0$ value means that we consider only the shortest paths. Considering only the shortest paths gives small chance to two nodes to appear together on the same path, which leads to a constantly low correlation. While a large one, $q_r=q_R$ will include all the objects in the whole training dataset as reference set. This will make SSP identical to the original graph transduction. Still, the shortest path with $q_r=0$ can achieve a better retrieval performance (92.19%) than graph transduction [1] (91.61%). As shown in Fig. 5, we can discover that the retrieval performance is not sensitive to the parameter $r$. Based on this observation, we use a constant sequential threshold of $r=2$ and $q_r = (2/10q)_R$. These $X_2^{SP}(x_t)$ is then selected as the reference of the target object $x_t$.

**Table 1**
Retrieval rates (bulls-eye) of different methods on the MPEG-7 dataset. Symbols in the table: MDS—multidimensional scaling, SC—shape context, DP—dynamic programming, EMD—Earth mover's distance, GM—geometric matching, CDM—contextual dissimilarity measure.

| Methods | Bulls-eye score (%) |
|---|---|
| Visual parts [17] | 76.45 |
| Shape contexts [18] | 76.51 |
| Curve edit distance [19] | 78.17 |
| Generative models [20] | 80.03 |
| MDS+SC+DP [21] | 84.35 |
| Planar graph cuts [22] | 85.00 |
| IDSC+DP [21] | 85.40 |
| IDSC+DP+EMD-$L_1$[23] | 86.56 |
| Shape-tree [11] | 87.70 |
| GM+IDSC [16] | 87.47 |
| GM+SC [16] | 88.11 |
| IDSC+CDM [14] | 88.30 |
| Contour flexibility [24] | 89.31 |
| IDSC+graph transduction [1] | 91.61 |
| GM+IDSC+meta descriptor [16] | 91.46 |
| GM+SC+meta descriptor [16] | 92.51 |
| Locally constrained diffusion [15] | 93.32 |
| IDSC+SSP | **93.35** |

From the retrieval rates collected in Table 1, we can observe that the similarity learning method using graph transduction including original [1] and our SSP produces better results than the IDSC [10] method. The shape similarity learned using graph transduction [1] is able to improve the IDSC retrieval rate to 91.61%, while the similarity proposed in this paper using shortest path propagation achieve the retrieval rate of 93.35%, which ranks as the top one in the list. The second highest result is 93.32% obtained by Locally Constrained Diffusion [15]. This is a solid evidence that similarity methods by SSP do achieve promising effects.

In order to visualize the gain in retrieval rates by our method as compared to IDSC, we plot the recall-precision curves in Fig. 6. Clearly, the graph transduction [1] and the SSP outperform the original IDSC [10] approaches on the MPEG-7 datasets. This means that the contextual similarity learning increases not only the bulls-eye score, but also the ranking of the shapes. The proposed SSP method consistently achieves better performance than the graph transduction [1]. These results suggest that using the proposed algorithm with the nodes in the shortest path as reference object set is still better than performing weighted average among all the other objects.

For each query, the average running time of graph transduction [1] method on MEPG-7 is about 30 s in Matlab. As comparison, the running time of the our SSP is about 97 s for each query, which is almost three times of graph transduction [1]. However, it is still acceptable for a online shape retrieval system. As mentioned

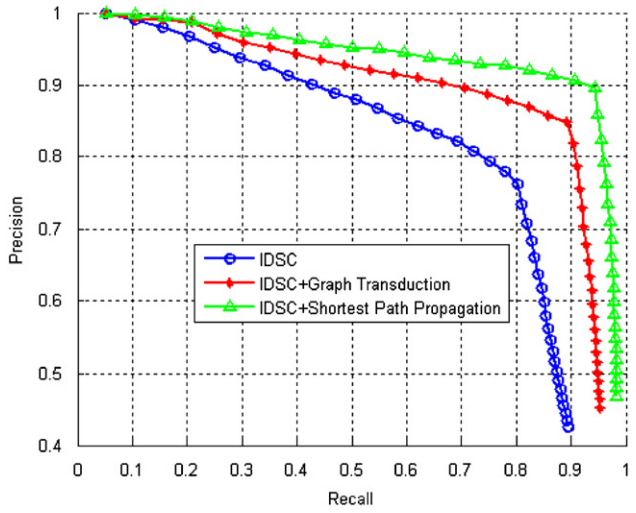


**Fig. 6.** A comparison of recall–precision curves between IDSC [10] (blue circles), the result improved by Graph Transduction [1] (red stars) and the proposed SSP method (green triangles) for MPEG-7. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

**Table 2**
Results of 1NN classification improvement.

| Similarity | Classification rate (%) |
|---|---|
| IDSC | 94.71 |
| IDSC+graph transduction [1] | 95.71 |
| IDSC+SSP | **96.29** |

**Table 3**
Overall Protein classification accuracy using the different protein similarity.

| Similarity | Classification rate (%) |
|---|---|
| Euclidean distance | 98.52 |
| Euclidean distance+graph transduction [1] | 98.90 |
| Euclidean distance+SSP | **99.09** |

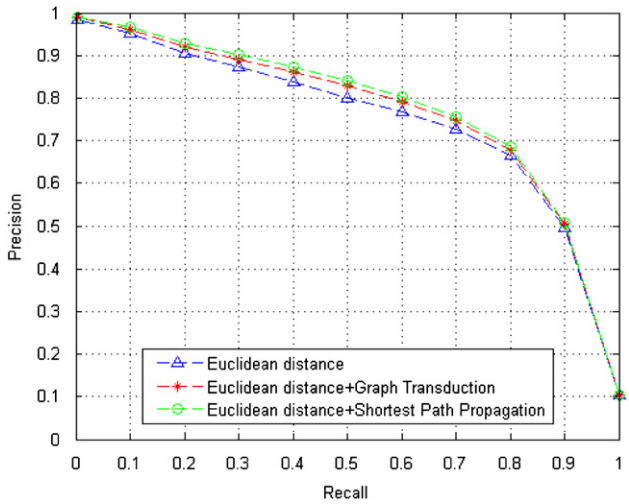


**Fig. 7.** A comparison of retrieval between Euclidean distance (green squares), and the result improved by the graph transduction [1] (blue stars) and the proposed shortest path propagation (red circles) for a part of the FSSP/DALI database. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

in Section 2.2, for graph transduction [1], all the similarities of the objects to the query is computed after the one iteration, while for SSP, after one iteration, only the $s(x_1,x_t)=f(x_t)$ can be returned as similarity between $x_1$ and $x_t$, so it is understandable that the running time of SSP is several times more than graph transduction. Nonetheless, when the SSP algorithm is applied to object retrieval on a large size database (for example, with millions or billions of objects), it is not practical to perform SSP among the whole database. In such a situation, we can first compute a short-list of objects for a query using a pairwise distance function, and then learn similarities of each short-listed objects to re-rank them using SSP. The shortest path information used by SSP can be learned in an off-line procedure, which will not affect the online querying procedure.

The nearest neighbor (NN) algorithm is among the simplest of all machine learning algorithms. The proposed similarity learning algorithm could also improve the recognition rate of NN classification. The classification results of MPEG-7 dataset have shown the improvement. We divided the MPEG-7 dataset into two sets: training set and testing set. For each class, 10 shapes are chosen as the training samples and the remaining 10 shapes are then used for testing. The results are shown in Table 2. We observe that the NN classification performance on these datasets has improved by SSP and graph transduction [1]. In particular, SSP always outperforms graph transduction [1] and yields the best classification rate 96.29%.

### 3.2. The experiments on FSSP/DALI protein dataset

In order to evaluate the performance of the proposed method following [6], a portion of the FSSP database [7] is also used. This database is constructed according to the DALI algorithm [8,9] and consists of 2635 proteins classified into 27 classes. For feature extraction, we use the descriptor $Kraw_{00}$ & $FT_{02}$ proposed in [6]. The benchmark metric of similarity is based on the Euclidean distance between the $Kraw_{00}$ & $FT_{02}$ descriptor vectors.

The performance of the classification was evaluated in terms of overall classification accuracy using leave-one-out experiment. The overall classification accuracy is the percentage of the correctly predicted class labels among all 2635 proteins of the database. First, the contextual similarity learning methods are evaluated, which rely on the Euclidean distance measure. We use the following parameters to construct the affinity matrix for SSP and graph transduction [1]: $\alpha=0.4$ ,the neighborhood size is $K=20$. Moreover, we set $q_r=2/10q_R$ to construct the shortest path reference set, and stop condition $T=1000$. The overall classification accuracy results are very satisfactory (Table 3). The overall performance of SSP is close to 100% of the best. This suggests that, for context-sensitive shape similarity by graph transduction, is the most useful information contained in shortest path between query and target object to be compared.

The efficiency of the proposed shape similarity learning method is also evaluated in terms of information retrieval performance. Each model of the database is used as query and the retrieved proteins are ranked in terms of shape similarity to the query. For the presentation of the results, the information retrieval precision–recall curve is used. Fig. 7 depicts the information retrieval precision–recall curve for all geometrical descriptor vectors used. From these results, we can see that SSP and graph transduction [1] outperform Euclidean distance in protein datasets. However, the difference in terms of performance is very little. The protein descriptor $Kraw_{00}$ & $FT_{02}$ used in this experiment is very powerful. Based on this descriptor, the basic Euclidean distance has archived a good retrieval performance,

leaving little space to improvement by SSP and graph transduction. In many settings, SSP achieves retrieval precision that is 0.60% to over 4.33% better than the precisions attained by the Euclidean distance. The SSP can improve protein accuracy for FSSP/DALI beyond the corresponding graph transduction [1] based similarity learning algorithms. The results demonstrate that the SSP has the potential to incorporate the knowledge of intrinsic shape differences, which is explicitly expressed by shortest paths on the Euclidean distance data space formed by the known shapes and the query shape.

In term of timing analysis, except for protein descriptor computation and Euclidean distance computation, all the other steps of the protein search system, i.e., iterative propagation, are computed several times instead of only once in graph transduction [1]. In our experiment, for each query protein, the average running time of graph transduction [1] method on MEPG-7 is about 50 s in Matlab. The running time of the SSP is about 3 min for each query, which is almost three times of graph transduction [1]. For a large protein database, it is still acceptable to rank 2000 proteins in 3 min, without any program optimization.

## 4. Conclusion

Our main contribution is the development of a contextual similarity learning algorithm for the retrieval and classification of shape. By defining a new reference object set, the proposed similarity algorithm propagates the similarity along the shortest path from the query to the target, in the [1] manner.

We then validated this contextual similarity algorithm on shape and protein datasets, in which nearest neighbor classification and retrieval tasks are performed. Experimental results have shown the advantages of this algorithm win over [1]. When the similarity propagation strategies avoid the redundancy produced by using the whole database as the reference, the performance of SSP is improved over graph transduction. We also note that the improvements are data dependent, which are also affected by the features used to describe the shapes. However, we can consistently observe improvements by replacing the global reference with shortest path.

## Acknowledgments

The authors would like to thank the anonymous reviewers and the editor for comments that helped them to improve this paper. This project is funded by the Natural Science Foundation of China (60903096).

## References

[1] X. Bai, X. Yang, L.J. Latecki, W. Liu, Z. Tu, Learning context-sensitive shape similarity by graph transduction, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (5) (2010) 861–874.
[2] J. Wang, S.-F. Chang, X. Zhou, T.C.S. Wong, Active microscopic cellular image annotation by superposable graph transduction with imbalanced labels, in: Proceedings of IEEE CS Conference on Computer Vision and Pattern Recognition, 2008.
[3] F. Zhang, J. Liu, An algorithm of shortest path based on Dijkstra for huge data, in: Proceedings of the 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2009), 2009, pp. 244–247.
[4] C. Papadimitriou, M. Sideri, On the Floyd–Warshall algorithm for logic programs, Journal of Logic Programming 41 (1) (1999) 129–137.
[5] M. Culp, G. Michailidis, Graph-based semisupervised learning, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (2008) 174–179.
[6] P. Daras, D. Zarpalas, A. Axenopoulos, et al., Three-dimensional shape-structure comparison method for protein classification, IEEE—ACM Transactions on Computational Biology and Bioinformatics 3 (3) (2006) 193–207.
[7] L. Holm, S. Kaariainen, P. Rosenstrom, et al., Searching protein structure databases with DaliLite v.3, Bioinformatics 24 (23) (2008) 2780–2781.
[8] L. Holm, C. Sander, The FSSP database: fold classification based on structure–structure alignment of proteins, Nucleic Acids Research 24 (1996) 206–210.
[9] L. Holm, C. Sander, Touring protein fold space with Dali/FSSP, Nucleic Acids Research 26 (1998) 316–319.
[10] H.B. Ling, D.W. Jacobs, Shape classification using the inner-distance, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (2) (2007) 286–299.
[11] P.F. Felzenszwalb, J. Schwartz, Hierarchical matching of deformable shapes, in: Proceedings of IEEE CS Conference on Computer Vision and Pattern Recognition, 2007.
[14] H. Jegou, H. Harzallah, C. Schmid, A contextual dissimilarity measure for accurate and efficient image search, in: CVPR '07. IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 9–16.
[15] 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 CVPR, 2009.
[16] A. Egozi, Y. Keller, H. Guterman, Improving shape retrieval by spectral matching and meta similarity, IEEE Transactions on Image Processing 19 (5) (2010) 1319–1327.
[17] L.J. Latecki, R. Lakamper, T. Eckhardt, Shape descriptors for non-rigid shapes with a single closed contour, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, 2000, pp. 424–429.
[18] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (4) (2002) 509–522.
[19] T.B. Sebastian, P.N. Klein, B.B. Kimia, On aligning curves, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (1) (2003) 116–125.
[20] Z. Tu, A. Yuille, Shape matching and recognition using generative models and informative features, European Conference on Computer Vision, vol. 1, 2004, pp. 195–209.
[21] H. Ling, D.W. Jacobs, Shape classication using the inner-distance, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (2) (2007) 286–299.
[22] F.R. Schmidt, E. Toeppe, D. Cremers, Efficient planar graph cuts with applications in computer vision, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, Florida, 2009.
[23] H. Ling, K. Okada, An efficient earth mover's distance algorithm for robust histogram comparison, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (5) (2007) 840–853.
[24] C. Xu, J. Liu, X. Tang, 2D shape matching by contour flexibility, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (1) (2009) 180–186.
[25] X. Yang, X. Bai, L.J. Latecki, Z. Tu, Improving shape retrieval by learning graph transduction, European Conference on Computer Vision, vol. 4, 2008, pp. 788–801.

**Jingyan Wang** received B.E. degree in EE from Central South University, Changsha, China. Currently, he is a Ph.D. candidate at Shanghai Institute of Applied Physics, Chinese Academy of Science, Shanghai. His research is focused on machine learning with applications on bioinformatics, medical imaging and computer vision. Jingyan Wang is a Student Member of IEEE and IEEE Engineering in Medicine and Biology Society.

**Yongping Li** received M.S. degree in the Electrical Engineering in 1989 from the Graduate School of Chinese Academy of Sciences. He worked as a Research Fellow at Shanghai Institute of Applied Physics (SINAP) for various research projects on computerized instrumentations from 1989 to 1995. He joined the Center for Vision Speech and Signal Processing, University of Surrey, UK in 1996 with a Royal Society fellowship, and received Ph.D. in Pattern Recognition, with emphasis on Biometrics, in 2000 under the supervision of Professor Josef Kittler. He worked as a Principal Researcher on speech signal processing at a startup company in Silicon Valley, California from 2001 to 2002. From 2003 to present, he is a Research Professor and Ph.D. student supervisor at Shanghai Institute of Applied Physics, and a part-time Ph.D. student supervisor at the Center for Biometrics and Security Research (CBSR), Institute of Automation, Chinese Academy of Sciences. His current research interests include digital signal processing for MCU and DSP based instrumentations, biometric algorithms for face, voiceprint and fingerprint recognition, multiple biometrics, biometric solution integration, biometrics system performance evaluation, testing, and standardization. Dr. Li is a member of the IEEE, IEEE computer society and IEEE instrumentation and measurement.

**Xiang Bai** received the B.S. and M.S. degrees both in Electronics and Information Engineering from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2003 and in 2005, respectively. In October 2009, he obtained his Ph.D. from HUST. From January 2006 to May 2007, he worked in the Department of Computer Science

and Information, Temple University. From October 2007 to October 2008, he worked in University of California, Los Angeles as a joint Ph.D. student. Now he is a faculty member of EI Department of HUST. His research interests include computer graphics, computer vision, and pattern recognition.

**Ying Zhang** received the B.E. degree in computer science and technology from Henan University, China in 2008. She is currently pursuing the M.S. degree in signal and information processing at Shanghai Institute of Applied Physics, Chinese Academy of Science, Shanghai, China. Her research interests are Machine Learning, Bioinformatics and Face Recognition.

**Chao Wang** received Ph.D. degree in CS from Graduate of Chinese Academy of Sciences and the master degree in CS from Shandong University, China. Currently, he is a Postdoc at Oregon Graduate Institute (OGI), Oregon Health & Science University (OHSU), US. His research interests include machine learning, computer vision, digital image processing, pattern recognition. Especially, he mainly study on face recognition.

**Ning Tang** received the B.E. degree in EE from Nankai University, Tianjin, China. Currently, he is pursuing a masters degree at Shanghai Institute of Applied Physics, Chinese Academy of Science, Shanghai. His research interests include machine learning, computer vision and bioinformatics.