Pattern Recognition 42 (2009) 1012 – 1013

Contents lists available at ScienceDirect

Pattern Recognition

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

A comment on “Using locally estimated geodesic distance to optimize neighborhood graph for isometric data embedding”

Caiming Zhong[a,b,c,*], Duoqian Miao[a,c]

[a]School of Electronics and Information Engineering, Tongji University, Shanghai 201804, PR China
[b]College of Science and Technology, Ningbo University, Ningbo 315211, PR China
[c]Tongji Branch, National Engineering and Technology Center of High Performance Computer, Shanghai 201804, PR China

ARTICLE INFO

Article history:
Received 21 July 2008
Accepted 11 November 2008

Keywords:
Triangle inequality
Geodesic distance
Euclidean distance

ABSTRACT

A geodesic distance-based approach to build the neighborhood graph for isometric embedding is proposed to deal with the highly twisted and folded manifold by Wen et al. [Using locally estimated geodesic distance to optimize neighborhood graph for isometric data embedding, Pattern Recognition 41 (2008) 2226–2236]. This comment is to identify the error in their example and the ineffectiveness of their algorithm.

Wen et al. recently proposed an approach which deals with highly twisted and folded manifold for isometric data embedding [1]. The approach employs locally estimated geodesic distances to optimize a neighborhood graph which is usually constructed with Euclidean distances in some isometric embedding methods such as Isomap [2]. Unfortunately, the example given in Ref. [1] is incorrect, and the algorithm OptimizeNeighborhoodbyGeod(X, k, m, d) is ineffective. This comment aims at identifying the errors.

In Ref. [1], the initial neighborhood is determined by Euclidean distance, and then the local geodesic distance is estimated. Fig. 1, which is corresponding to Fig. 2 in Ref. [1], illustrates the process of the estimation. Let $N(x)$ be a set of Euclidean distance based three nearest neighbors of a data point x, then $N(x)=\{x_1,x_2,x_3\}$, and $N(x_1)=\{x_{11},x_{12},x_{13}\}$. Let $d(x,y)$ be the Euclidean distance between data point x and y. As x_1 is a neighbor of x, and x_{11} is neighbor of x_1, applying triangle inequality theorem, we have

$$d(x,x_{11}) \leqslant d(x,x_1)+d(x_1,x_{11}). \tag{1}$$

Furthermore, x_{11} is not a neighbor of x, that implies

$$d(x,x_{11}) > d(x,x_i), \quad i=1,2,3. \tag{2}$$

From (1) and (2), we obtain

$$d(x,x_1)+d(x_1,x_{11}) > d(x,x_i), \quad i=1,2,3. \tag{3}$$

That is to say, in Fig. 1, $d(x,x_1)=2$, $d(x_1,x_{11})=5$ and $d(x,x_3)=12$ cannot exist simultaneously. Accordingly, x_3 cannot be optimized into x_{11}, and for the same reason, x_2 cannot be optimized into x_{12}.

Based on the above analysis, the algorithm OptimizeNeighborhoodbyGeod(X, k, m, d) given in Ref. [1] is ineffective. The algorithm is as follows.

Algorithm 1. OptimizeNeighborhoodbyGeod(X, k, m, d).

/* $X=x_i$ be the high dimensional data set, k be the neighborhood size, m be the scope for locally estimating geodesic distances, d be the dimension of the embedding space, and $m<k$. The output is the optimized neighborhood set $N=\{N(x_i)\}$ for all points */

(1) Calculate the neighborhood $N(x_i)$ for any point x_i using Euclidean distance d_e, where $N(x_i)$ is sorted ascendingly. Let $d_g(x_i,x_j)=d_e(x_i,x_j)$ for the pairs of all points.
(2) For $i=1$ to $|X|$, where $|X|$ is the number of points in X.
(3) · For $j=1$ to k
(4) ·· Select jth point from $N(x_i)$, denoted as x_{ij}
(5) ·· For $p=1$ to m
(6) ··· Select pth point from $N(x_{ij})$, denoted as x_{ijp}
(7) ···If $d_g(x_i,x_{ij})+d_g(x_{ij},x_{ijp})<d_g(x_i,x_{ik})$ and $x_{ijp}\notin N(x_i)$ and parent $(x_{ijp})\in N(x_i)$
(8) ···· Delete x_{ik} from $N(x_i)$
(9) ····· Insert x_{ijp} into $N(x_i)$ ascendingly

DOI of original articles: 10.1016/j.patcog.2007.12.015, 10.1016/j.patcog.2008.11.002.

*Corresponding author at: School of Electronics and Information Engineering, Tongji University, Shanghai 201804, PR China. Tel.: +86 21 69589867.
E-mail addresses: charman_zhong@hotmail.com, zhongcaiming@nbu.edu.cn (C. Zhong).

0031-3203/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2008.11.007

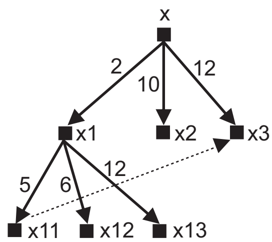

Fig. 1. Example to optimizing the neighborhood of the point *x*.

(10) · · · · · $d_g(x_i, x_{ijp}) = d_g(x_i, x_{ij}) + d_g(x_{ij}, x_{ijp})$
(11) · · · · · Let $j = 1$ and break
(12) · · · End
(13) ·· End
(14) · End
(15) End
(16) $N = N(x_i)$ be the optimized neighborhood for all points in X

In the step 1, since $d_g(x_i, x_j) = d_e(x_i, x_j)$ for the pairs of all points, i.e. all the local geodesic distances are initialized to corresponding Euclidean distances. According to the analysis on the example, the condition of step 7 is never satisfied, and the block from steps 8 to 11 is never executed. Consequently, the algorithm is ineffective.

References

[1] G. Wen, L. Jiang, J. Wen, Using locally estimated geodesic distance to optimize neighborhood graph for isometric data embedding, Pattern Recognition 41 (2008) 2226–2236.

[2] J.B. Tenenbaum, V. de Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (2000) 2319–2323.

About the Author—CAIMING ZHONG is currently pursuing his Ph.D. at Tongji University, Shanghai, China. His research interests include cluster analysis, manifold learning and image segmentation.

About the Author—DUOQIAN MIAO is a professor of Department of Computer Science and Technology at Tongji University, Shanghai, China. He has published more than 40 papers in international proceedings and journals. His research interests include soft computing, rough sets, pattern recognition, data mining, machine learning and granular computing.