Information Sciences 187 (2012) 72–92

Contents lists available at SciVerse ScienceDirect

Information Sciences

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

Extending Sammon mapping with Bregman divergences

Jigang Sun *, Colin Fyfe, Malcolm Crowe

University of the West of Scotland, School of Computing, High Street, Paisley, Renfrewshire PA1 2BE, United Kingdom

ARTICLE INFO

Article history:
Received 19 March 2011
Received in revised form 17 August 2011
Accepted 8 October 2011
Available online 6 November 2011

Keywords:
Metric multidimensional scaling
Sammon mapping
LeftSammon
RightSammon
Bregman divergence
Visualisation quality assessment criterion

ABSTRACT

The Sammon mapping has been one of the most successful nonlinear metric multidimensional scaling methods since its advent in 1969, but effort has been focused on algorithm improvement rather than on the form of the stress function. This paper further investigates using left Bregman divergences to extend the Sammon mapping and by analogy develops right Bregman divergences and reveals the mechanism that improves the performance of scaling over the Sammon mapping. The influence of data space distance preprocessing on optimisation speed is noticed. Non-stress visualisation quality measures are used to compare the configuration quality of the Sammon mapping and its extensions using both Euclidean distance and graph distance on three data sets.

1. Introduction

Metric multidimensional scaling (MMDS) [2] is a group of visualisation and dimensionality reduction methods that projects data points from high dimensional data space to low, typically, two dimensional latent space in which the structure of the original data set can be identified by eye; this can be a preprocessing step for clustering or classification [4,7]. It creates the low dimensional representation by keeping distance between latent points as close to the distances between the original data points as possible by moving the latent points around. The Sammon mapping [15] has been one of the most successful nonlinear MMDS methods since its advent in 1969, which improves the quality of scaling by focusing on small distances in data space: explicitly the Sammon mapping tries to capture the local structure of the data while paying less attention to the global structure of the data. Many articles concentrate on improving the optimisation algorithm, such as Samann (Sammon artificial neural network) [13], triangulation [12], and enhancement [14]. Inspired by Isomap's [19] changing of the metric from Euclidean distance to graph distance, metric replacement has been tried on the Sammon mapping by GeoNLM [22]. By contrast, modification of the stress function is rarely studied. This paper extends our previous work in [18] to improve the Sammon mapping by modifying its stress function.

In this paper we use D_{ij} and G_{ij} to represent the Euclidean and graph distance between points X_i and X_j in data space respectively; L_{ij} represents the distance in latent space between the corresponding mapped points Y_i and Y_j. The size of data set is N.

The main contributions of this paper are that we propose a new extension to the Sammon mapping, the RightSammon mapping; we also provide a further analysis to the LeftSammon mapping proposed, which was in [18]; and a comparison of the two extensions, the LeftSammon and RightSammon mapping, is carried out both theoretically and empirically.

* Corresponding author. Tel.: +44 141 8483305; fax: +44 141 8483616.
E-mail addresses: Jigang.Sun@uws.ac.uk (J. Sun), Colin.Fyfe@uws.ac.uk (C. Fyfe), Malcolm.Crowe@uws.ac.uk (M. Crowe).

0020-0255/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2011.10.013

The paper is organised as follows. In Section 2 the use of Bregman divergences in MMDS is briefly reviewed. In Section 3 a further explanation of the LeftSammon mapping is given. In Section 4 the use of the right Bregman divergence in MMDS is briefly reviewed. In Section 5 a new method, the RightSammon mapping, is proposed with a quick experiment using the Euclidean distance on an illustrative low dimensional open box data set, and features of its stress formation are described. In Section 6 the LeftSammon and RightSammon mapping are compared with respect to stress. In Section 7 non-stress visualisation quality measures are reviewed. In Section 8 experiments on standard data sets using both Euclidean distance and graph distance are presented. Finally Section 9 concludes the paper.

2. Multidimensional scaling and Bregman divergence

2.1. General form of MMDS

The stress functions of MMDS can be generalised as

$$E_{MMDS}(Y) = \frac{1}{C}\sum_{i=1}^{N}\sum_{j=i+1}^{N}(L_{ij} - D_{ij})^2 w_{ij}, \tag{1}$$

where C is a normalisation scalar for stress comparison purposes, and weight $w_{ij} \geqslant 0$; usually $w_{ij} = 0$ for missing data sample (D_{ij} is unknown). The weight w_{ij} can be a constant as in Raw-Stress [8]

$$E_{Raw-Stress}(Y) = \sum_{i=1}^{N}\sum_{j=i+1}^{N}(L_{ij} - D_{ij})^2, \tag{2}$$

where $C = 1$ and $w_{ij} = 1$ in Eq. (1). Usually it is a function of the distance in data space as in the Sammon mapping [15]

$$E_{Sammon}(Y) = \frac{1}{\sum_{i=1}^{N}\sum_{j=i+1}^{N}D_{ij}}\sum_{i=1}^{N}\sum_{j=i+1}^{N}\frac{(L_{ij} - D_{ij})^2}{D_{ij}}, \tag{3}$$

where $C = \sum_{i=1}^{N}\sum_{j=i+1}^{N}D_{ij}$ and $w_{ij} = D_{ij}^{-1}$ in (1).

We discuss weighting the terms by distances in latent space in Section 5.

2.2. Bregman divergence and its application in multidimensional scaling

Consider a strictly convex function $F : S \to \Re$ defined on a convex set $S \subset \Re^d$ ($\Re^d$ denotes d-dimensional real vector space). A Bregman divergence [1] between two points, $\mathbf{p}$ and $\mathbf{q} \in S$, is defined to be

$$d_F(\mathbf{p}, \mathbf{q}) = F(\mathbf{p}) - F(\mathbf{q}) - \langle(\mathbf{p} - \mathbf{q}), \nabla F(\mathbf{q})\rangle, \tag{4}$$

where the angled brackets indicate an inner product and $\nabla F(\mathbf{q})$ is the derivative of F evaluated at $\mathbf{q}$. This can be viewed as the difference between $F(\mathbf{p})$ and its truncated Taylor series expansion around $\mathbf{q}$. Thus it can be used as a measure of the convexity of F. When F is a function of a scalar, the divergence (4) is expressed as $d_F(p, q) = F(p) - F(q) - \frac{dF}{dq}(p - q)$. It is equivalent to the residuals of the Taylor series $d_F(p, q) = \frac{d^2F}{dq^2}\frac{(p-q)^2}{2!} + \frac{d^3F}{dq^3}\frac{(p-q)^3}{3!} + \frac{d^4F}{dq^4}\frac{(p-q)^4}{4!} + \cdots$ which will be useful in the subsequent analysis of the mappings we derive. The following properties determine its usefulness in MMDS:

Non-negativity $d_F(\mathbf{p},\mathbf{q}) \geqslant 0$ and $d_F(\mathbf{p},\mathbf{q}) = 0 \Leftrightarrow \mathbf{p} = \mathbf{q}$. In fact $d_F(\mathbf{p},\mathbf{q})$ steadily decreases as distance between $\mathbf{p}$ and $\mathbf{q}$ decreases.

Asymmetry $d_F(\mathbf{p},\mathbf{q}) \neq d_F(\mathbf{q},\mathbf{p})$ except in special cases such as $F(\cdot) = \|\cdot\|^2$, the squared Euclidean norm.

The non-negativity indicates that the stress function of MMDS can be defined using a divergence. If p is the inter-point distance in data space and q is the inter-point distance in latent space, when p and q are close to each other, the divergence reduces towards zero; or conversely when the divergence is minimised, the two distances are close to each other. The stress function we defined in [18] is:

$$E_{LMMDS}(Y) = \sum_{i=1}^{N}\sum_{j=i+1}^{N}d_F(L_{ij}, D_{ij}) = \sum_{i=1}^{N}\sum_{j=i+1}^{N}\big(F(L_{ij}) - F(D_{ij}) - (L_{ij} - D_{ij})\nabla F(D_{ij})\big). \tag{5}$$

We call this a *left* divergence (Left MMDS (LMMDS)) since the latent points' distances are found in the left position in the Bregman divergences, and these are the terms that we can affect by moving the latent points' positions (a *right* divergence will be defined in Section 5). Concentrating on the higher order terms of the Taylor series expansion, we have

$$E_{LMMDS}(Y) = \sum_{i=1}^{N}\sum_{j=i+1}^{N}\left(\frac{d^2F(D_{ij})}{dD_{ij}^2}\frac{(L_{ij} - D_{ij})^2}{2!} + \frac{d^3F(D_{ij})}{dD_{ij}^3}\frac{(L_{ij} - D_{ij})^3}{3!} + \frac{d^4F(D_{ij})}{dD_{ij}^4}\frac{(L_{ij} - D_{ij})^4}{4!} + \cdots\right). \tag{6}$$

The sum of divergences (5) is used for stress minimisation, i.e. the actual construction of the maps uses (5); the Taylor series expansion (6) is used for analysis of these same maps.

The MMDS stress expression (1) is closely related to (5). In fact, if we choose $F(x) = x^2$, (5) is identical to (2). In many cases (1) can be regarded as first term of (6) up to constant coefficients which are then ignored; and (5) can be regarded as a generalisation of (1).

In [18], we show how, e.g. the Sammon mapping and other MMDS can be generalised with *left* Bregman divergences. In [18] we use the base convex function

$$F(x) = x\ln x, \quad x > 0, \tag{7}$$

to create the LeftSammon (known as ExtendedSammon in [18]) mapping

$$E_{Left}(Y) = \sum_{i=1}^{N}\sum_{j=i+1}^{N}\left(L_{ij}\ln\frac{L_{ij}}{D_{ij}} - L_{ij} + D_{ij}\right) = \sum_{i=1}^{N}\sum_{j=i+1}^{N} T_{Left}(i,j), \tag{8}$$

where $T_{Left}(i,j) = L_{ij}\ln\frac{L_{ij}}{D_{ij}} - L_{ij} + D_{ij}$. Eq. (8) is alternatively expressed with the higher order terms of the Taylor series as

$$\begin{aligned} E_{Left}(Y) &= \sum_{i=1}^{N}\sum_{j=i+1}^{N}\left(\frac{(L_{ij}-D_{ij})^2}{2D_{ij}} - \frac{(L_{ij}-D_{ij})^3}{3!D_{ij}^2} + \cdots + \frac{(-1)^n(n-2)!}{D_{ij}^{n-1}}\frac{(L_{ij}-D_{ij})^n}{n!} + \cdots\right) \\ &= \sum_{i=1}^{N}\sum_{j=i+1}^{N}\left(T_{Sammon}(i,j) + T_3^{Left}(i,j) + \cdots + T_n^{Left}(i,j) + \cdots\right), \end{aligned} \tag{9}$$

where $T_{Sammon}(i,j) = \frac{(L_{ij}-D_{ij})^2}{2D_{ij}}$, $T_n^{Left}(i,j) = \frac{(-1)^n(n-2)!}{D_{ij}^{n-1}}\frac{(L_{ij}-D_{ij})^n}{n!}$, $n = 3, 4, \ldots$ $T_{Left}(i,j)$ in (8) is equivalent to $T_{Sammon}(i,j) + T_3^{Left}(i,j) + \cdots + T_n^{Left}(i,j) + \cdots$

Ignoring constant coefficients, the stress of the Sammon mapping (3) is the first term of (9); thus the Sammon mapping can be considered as an approximation to the LeftSammon mapping. We illustrated via experiment on several data sets in [18] that the LeftSammon mapping outperforms the Sammon mapping; nevertheless for clarity in this paper we replicate the experiment on a benchmark artificial data set, the open box, as shown in Fig. 5(a)–(c). The open box data set consists of 316 data points. Page 15 of [10] states that the open box "is connected but neither compact (in contrast with a cube or closed box) nor smooth (there are sharp edges and corners)." It is implemented using standard Newton's gradient descent method. A high dimensional data set X is projected onto a P dimensional output Y, usually $P = 2$. Y_{ki} stands for the kth dimension of ith output data

$$Y_{ki} \leftarrow Y_{ki} - \alpha\frac{\partial E(Y)}{\partial Y_{ki}}, \quad k = 1, 2, \ldots, P \tag{10}$$

α is a factor that make stress reduce in each iteration; E stands for the stress of a mapping. The implementation is based on Cawley's implementation from http://theoval.cmp.uea.ac.uk/matlab/sammon/sammon.m; the detailed optimisation algorithm is shown Fig. 1; the optimisation is run 25 times and the result with the lowest stress is selected.

```
 1. Randomly initialise the output coordinates $Y$;
 2. Calculate stress and stored it as $error$;
 3. Loop
 4.     Calculate the first order derivatives $\frac{\partial E(Y)}{\partial Y}$;
 5.     Set $\alpha = 1$;
 6.     For $i = 1 : aMaximumTimes$ (we set $aMaximumTimes = 50$)
 7.         By rule (10), the new output coordinates is calculated as
                $$Y_{new} \leftarrow Y - \alpha\frac{\partial E(Y)}{\partial Y};$$
 8.         Calculate the new stress and stored it as $error_{new}$;
 9.         If $error_{new} > error$
10.             $\alpha = \alpha/2$;
11.         Else
12.             $Y = Y_{new}$;
13.             If $\frac{error - error_{new}}{error} < stoppingCriterion$ (we set $stoppingCriterion = 1e - 12$);
14.                 Goto step 20;
15.             EndIf
16.             $error = error_{new}$;
17.         End-If
18.     EndFor
19. EndLoop
20. Termination of calculation.
```

Fig. 1. The optimisation algorithm used for the Sammon, LeftSammon and RightSammon mappings.

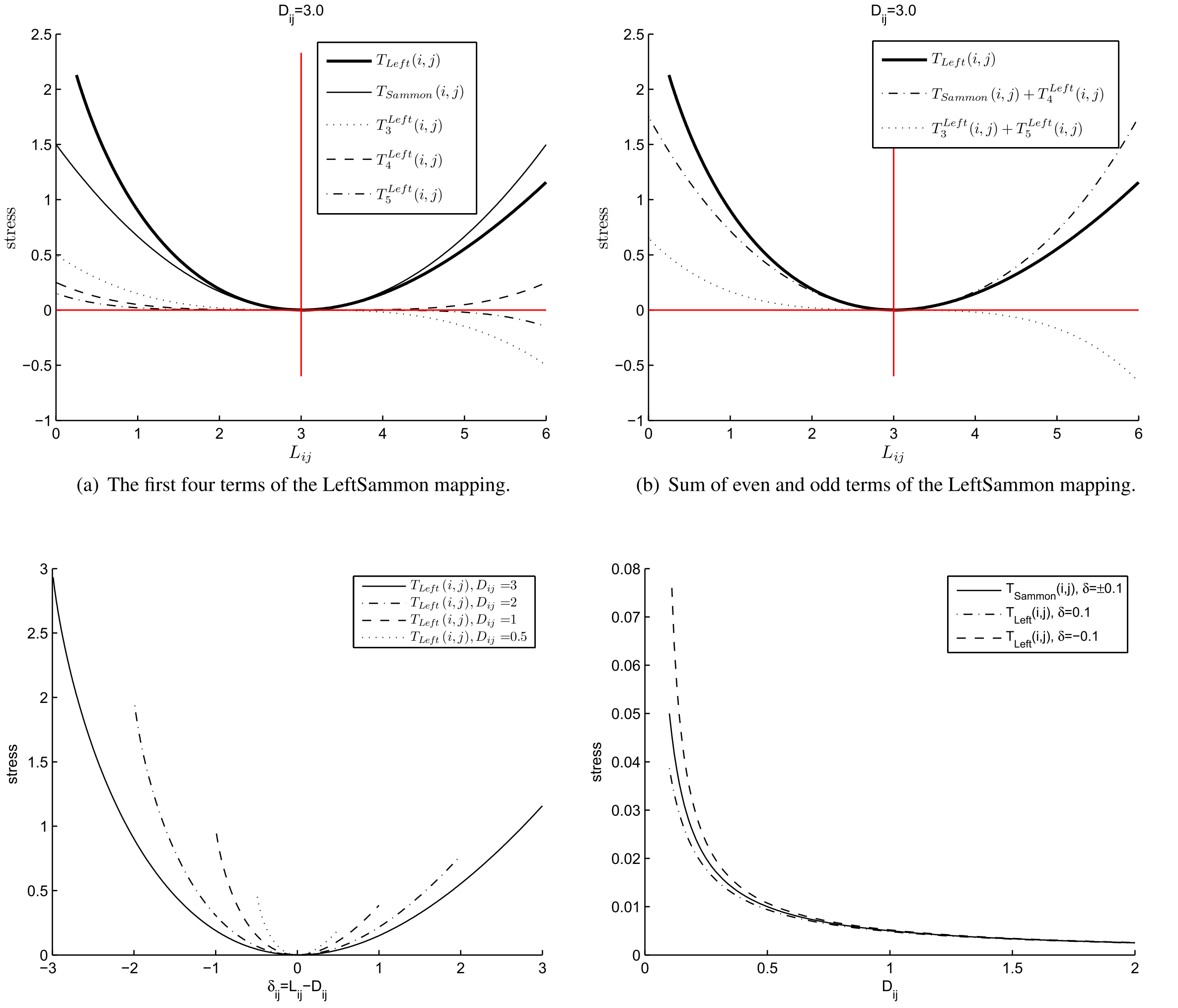
Before doing the experiment, the distance matrix is preprocessed by dividing distances in data space by their mean. In MMDS, dividing distance matrix by a constant does not change the neighbourhood relationships of data points in data space, but has an effect on convergence speed of optimisation and on scaling results if using the typical quasi-Newton optimisation method employed in the Sammon mapping (3).

Fig. 5(b) and (c) show mapping results by the Sammon and the LeftSammon mapping, using the Euclidean distance to calculate L_{ij} and D_{ij} (the first order derivatives of the LeftSammon mapping used in optimisation are given in Section A); the opening of the box projected by the LeftSammon mapping is wider than the opening of the open box mapped by the Sammon mapping; the overlapping of data points between bottom and three vertical sides reduces accordingly.

The underlying reasons of the advantages of the LeftSammon mapping are its stress composition as shown in Fig. 2(a). We can see that the stress of the Sammon mapping is the main part of the LeftSammon mapping and is symmetric with respect to $L_{ij} = D_{ij}$, while it is not the case for the LeftSammon mapping: when $L_{ij} < D_{ij}$, $T_{Left}(i,j) > T_{Sammon}(i,j)$ but when $L_{ij} > D_{ij}$ $T_{Left}(i,j) < T_{Sammon}(i,j)$, which means that on average the LeftSammon mapping tends to map distances longer but the Sammon mapping does not; in other words, the higher order terms in (9) of the LeftSammon mapping deliver an extra power of unfolding over the Sammon mapping. In the following section we will continue to explain the differences in stress.

3. Comparing the LeftSammon vs Sammon mappings in stress

The stress composition of the LeftSammon mapping is further explained in Fig. 2(a) and (b). Fig. 2(a) shows the stress composition of the LeftSammon mapping for $D_{ij} = 3.0$ truncated to the first four terms of the Taylor series expansion. We



(a) The first four terms of the LeftSammon mapping.

(b) Sum of even and odd terms of the LeftSammon mapping.

(c) Stress for different D_{ij} with respect to the same distance error, $0 \le L_{ij} \le 2D_{ij}$.

(d) Focusing on small distances, the the LeftSammon mapping gives low stress when distance is mapped longer.

Fig. 2. Comparing the LeftSammon vs Sammon mapping.

can see that the Sammon mapping provides the main part, while the magnitudes of the third, fourth and fifth terms decrease greatly. Terms of even orders are positive and symmetric with respect to the vertical line $L_{ij} = D_{ij}$, while odd order terms are asymmetric with respect to $(D_{ij}, 0)$. If terms of even order are summed together and similarly the odd terms, we get Fig. 2(b). We can see that for the same magnitude distance error $|\delta_{ij}|$ ($\delta_{ij} = L_{ij} - D_{ij}$), the sum of even orders gives the same stress when δ_{ij} is either positive or negative, as does the Sammon mapping. But for the LeftSammon mapping, the sum of odd orders is positive, i.e. stress increases, when δ_{ij} is negative; the sum of odd orders is negative, i.e. stress decreases, when δ_{ij} is positive. These cause the overall effect that the stress curve of the LeftSammon mapping is tilting rightwards, i.e. the curve of the left part ($L_{ij} < D_{ij}$), is steeper than the right part ($L_{ij} > D_{ij}$), resulting in longer mapped distances giving less stress; in other words, it encourages unfolding.

The stress of the LeftSammon mapping (8) can be expressed as $T_{Left}(i,j) = (D_{ij} + \delta_{ij}) \ln \frac{D_{ij}+\delta_{ij}}{D_{ij}} - \delta_{ij}$. The value of $T_{Left}(i,j)$ for various D_{ij} with respect to δ_{ij} is illustrated in Fig. 2(c). We can see that, like the Sammon mapping, the smaller the distance is, the higher the stress is; intuitively the smaller the distance, the steeper its stress curve. For the same distance error, a small distance tends to contribute higher stress than a long distance; thus small distances are focused on. The focusing on small distance of the Sammon mapping and the LeftSammon mapping is also reflected in Fig. 2(d) when the magnitude of the distance error is fixed. It shows that both the Sammon and the LeftSammon mapping give high stress for small distances, i.e. they both focus on small distances. The difference is that the Sammon mapping does not distinguish between

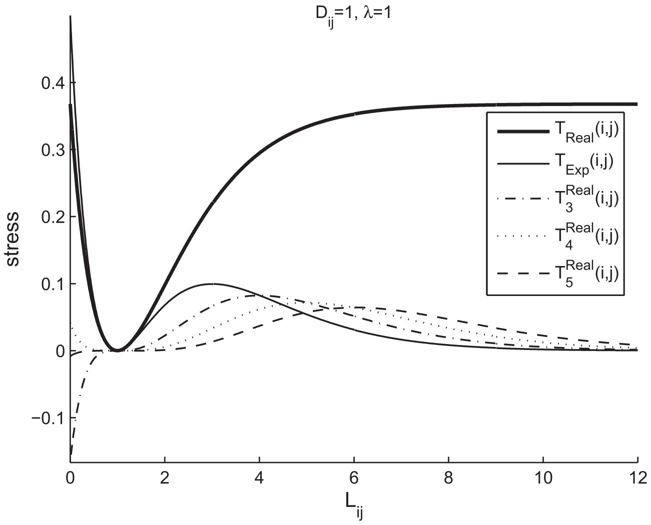


Fig. 3. Term formation of stress of the RightExp, truncated to the first four terms.

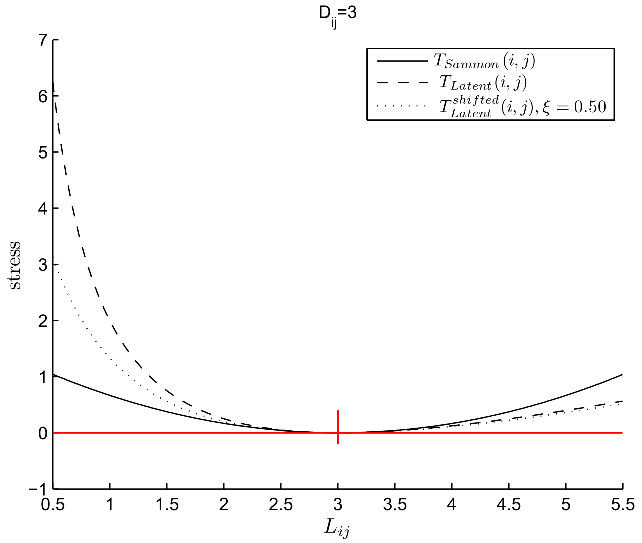
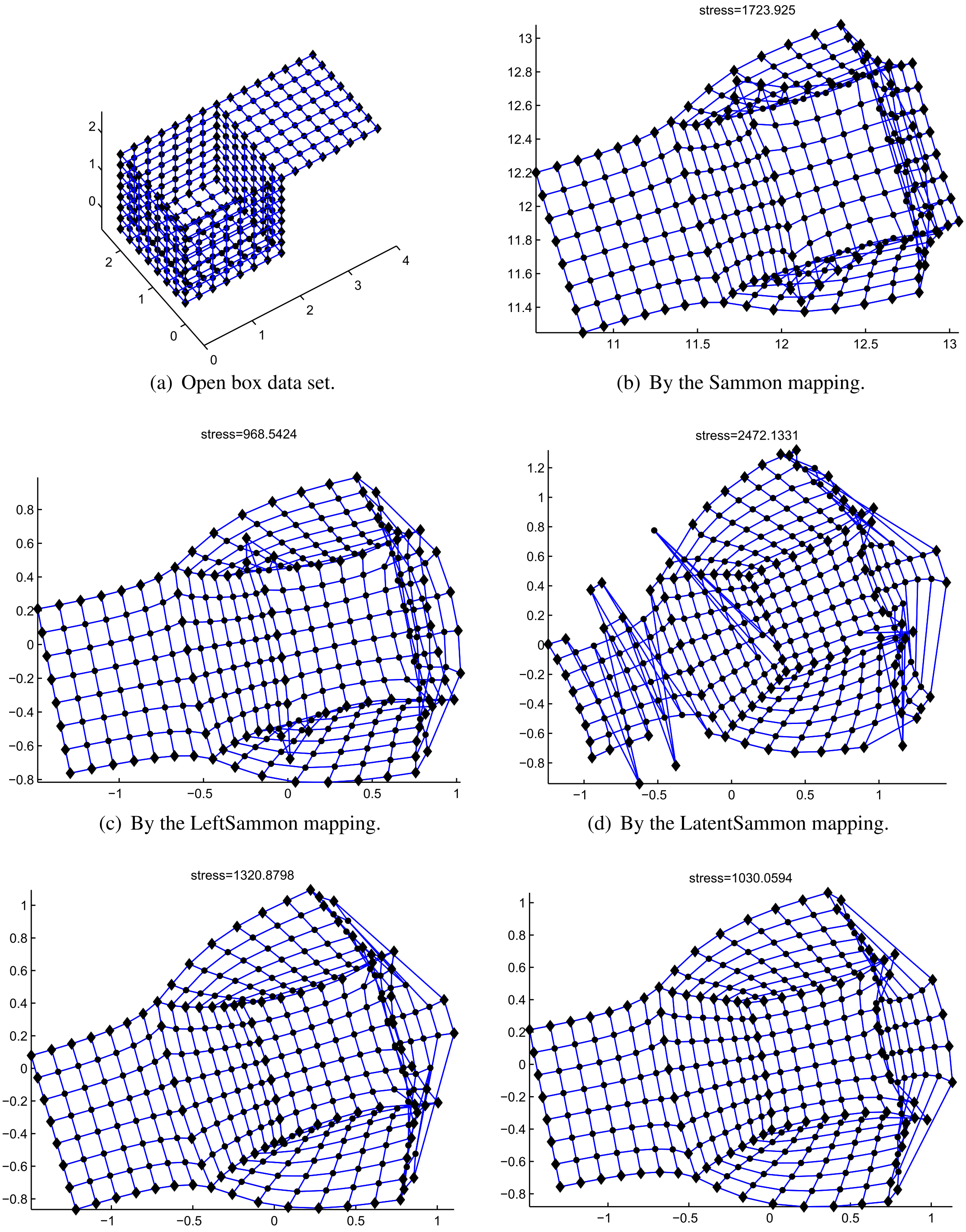


Fig. 4. The Sammon vs LatentSammon mapping.

positive and negative error in the stress. The stress given by the LeftSammon mapping is higher than by the Sammon mapping for $L_{ij} < D_{ij}$ and is lower for $L_{ij} > D_{ij}$, which means that it is easier for the LeftSammon mapping to map distances longer than shorter; this is a further explanation of the unfolding power of the LeftSammon mapping.



(a) Open box data set.

(b) By the Sammon mapping.

(c) By the LeftSammon mapping.

(d) By the LatentSammon mapping.

(e) By the shifted LatentSammon mapping, ξ = standard deviation of D_{ij}.

(f) By the RightSammon mapping.

Fig. 5. Open box data set and experiment on it using Euclidean distance (the stresses displayed are of different stress functions).

4. Using right Bregman divergences in MMDS

The non-symmetry of Bregman divergences indicates that in Eq. (4) when p and q are swapped, a different divergence is created. If the position of D_{ij} and L_{ij} in (8) is swapped, a *right* Bregman divergence is created to define an MMDS stress function

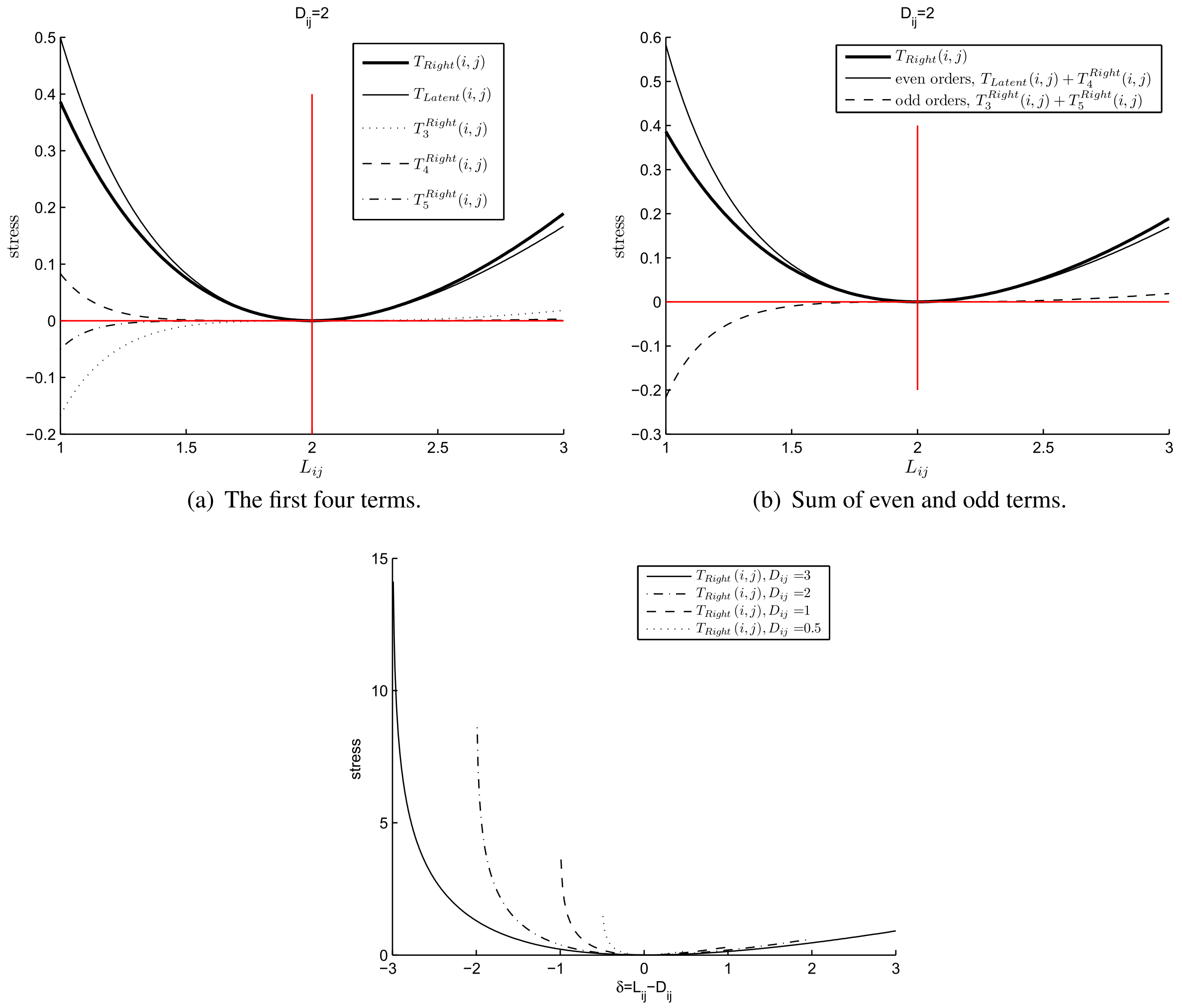
$$E_{RMMDS}(Y) = \sum_{i=1}^{N}\sum_{j=i+1}^{N} d_F(D_{ij}, L_{ij}) = \sum_{i=1}^{N}\sum_{j=i+1}^{N} \left(F(D_{ij}) - F(L_{ij}) - (D_{ij} - L_{ij})\nabla F(L_{ij})\right), \tag{11}$$

where the distances in latent space occupy the right position in the formula. It is equivalent to

$$E_{RMMDS}(Y) = \sum_{i=1}^{N}\sum_{j=i+1}^{N} \left(\frac{d^2F(L_{ij})}{dL_{ij}^2}\frac{(L_{ij} - D_{ij})^2}{2!} + \frac{d^3F(L_{ij})}{dL_{ij}^3}\frac{(L_{ij} - D_{ij})^3}{3!} + \frac{d^4F(L_{ij})}{dL_{ij}^4}\frac{(L_{ij} - D_{ij})^4}{4!} + \cdots\right), \tag{12}$$

which, again, is convenient for analysis.

In fact right Bregman divergences have been used in MMDS as in [17]. As we mentioned before, the weight w_{ij} of the general form of MMDS (1) is usually parameterised in the data space distance. One instance of a latent space parameterisation is a version of Curvilinear Components Analysis (CCA) [5]



(a) The first four terms.

(b) Sum of even and odd terms.

(c) Stress for different D_{ij}.

Fig. 6. Stress for the RightSammon mapping.

$$E_{Exp}(Y)=\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(L_{ij}-D_{ij})^2e^{-\frac{L_{ij}}{\lambda}}, \tag{13}$$

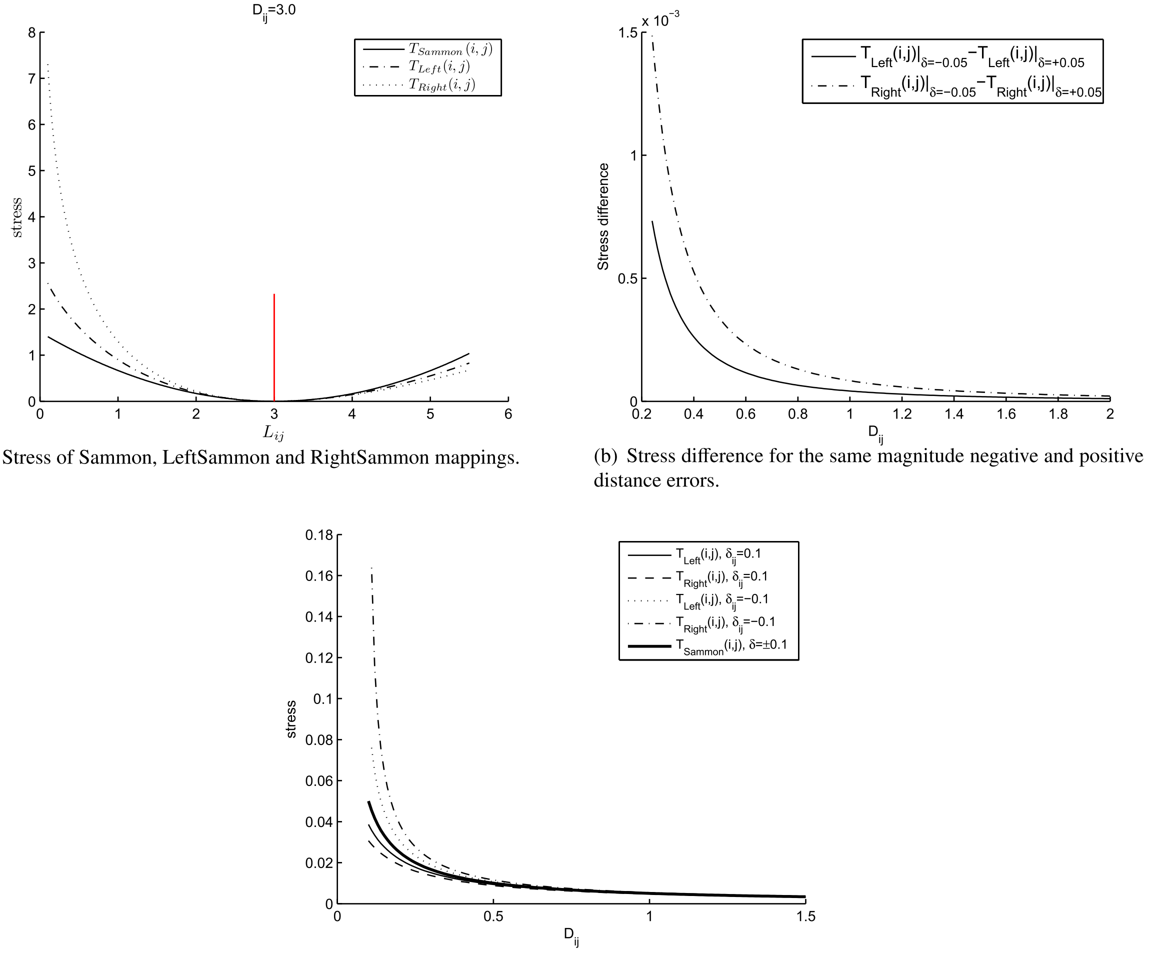
where $C=2,\ w=e^{-\frac{L_{ij}}{\lambda}},\ \lambda$ is regarded as a local neighbourhood radius parameter.

If the base convex function $F(x)=e^{-\frac{x}{\lambda}}$ in (11), we have the new stress function

$$\begin{aligned}
&E_{Real}(Y)\\
&=\lambda^2\sum_{i=1}^{N}\sum_{j=1}^{N}\left(e^{-\frac{D_{ij}}{\lambda}}-e^{-\frac{L_{ij}}{\lambda}}+(D_{ij}-L_{ij})\frac{e^{-\frac{L_{ij}}{\lambda}}}{\lambda}\right) &(14)\\
&=\sum_{i=1}^{N}\sum_{j=1}^{N}\left(e^{-\frac{L_{ij}}{\lambda}}\frac{(D_{ij}-L_{ij})^2}{2!}-\frac{e^{-\frac{L_{ij}}{\lambda}}}{\lambda}\frac{(D_{ij}-L_{ij})^3}{3!}+\cdots+\frac{e^{-\frac{L_{ij}}{\lambda}}}{(-\lambda)^{n-2}}\frac{(D_{ij}-L_{ij})^n}{n!}+\cdots\right)\\
&=\sum_{i=1}^{N}\sum_{j=1}^{N}\left(T_{Exp}(i,j)+T_3^{Real}(i,j)+\cdots+T_n^{Real}(i,j)+\cdots\right), &(15)
\end{aligned}$$

where $T_{Exp}(i,j)=e^{-\frac{L_{ij}}{\lambda}}\frac{(D_{ij}-L_{ij})^2}{2}$ representing (13), $T_3^{Real}(i,j)=-\frac{e^{-\frac{L_{ij}}{\lambda}}}{\lambda}\frac{(D_{ij}-L_{ij})^3}{3!}$, $T_n^{Real}(i,j)=\frac{e^{-\frac{L_{ij}}{\lambda}}}{(-\lambda)^{n-2}}\frac{(D_{ij}-L_{ij})^n}{n!}$.

We showed in [17] that Eq. (13) is a bad approximation to (14) as shown in Fig. 3 and is not suitable for performing MMDS; the actual stress function of the algorithm of this version of CCA is (14). Note again that (14) can be used in implementation while (15) is useful for analysis.



(a) Stress of Sammon, LeftSammon and RightSammon mappings.

(b) Stress difference for the same magnitude negative and positive distance errors.

(c) Focusing on small distances.

Fig. 7. Stress of the Sammon, LeftSammon and RightSammon mappings.

5. Using a right Bregman divergence to extend the Sammon mapping

Eq. (14) is not the only case of using right Bregman divergences in MMDS. Now we explore the possibility of using the base convex function (7) for the LeftSammon mapping to define a different mapping, which we call the RightSammon mapping:

$$\begin{aligned} E_{Right}(Y) &= \sum_{i=1}^{N}\sum_{j=i+1}^{N}(D_{ij}\ln D_{ij} - L_{ij}\ln L_{ij} - (D_{ij}-L_{ij})(\ln L_{ij}+1)) \\ &= \sum_{i=1}^{N}\sum_{j=i+1}^{N}\left(D_{ij}\ln\frac{D_{ij}}{L_{ij}} - D_{ij} + L_{ij}\right) = \sum_{i=1}^{N}\sum_{j=i+1}^{N} T_{Right}(i,j), \end{aligned} \tag{16}$$

whose equivalent Taylor series expansion is

$$\begin{aligned} E_{Right}(Y) &= \sum_{i=1}^{N}\sum_{j=i+1}^{N}\left(\frac{(D_{ij}-L_{ij})^2}{2L_{ij}} - \frac{(D_{ij}-L_{ij})^3}{3!L_{ij}^2} + \cdots + \frac{(-1)^n(n-2)!}{L_{ij}^{n-1}}\frac{(D_{ij}-L_{ij})^n}{n!} + \cdots\right) \\ &= \sum_{i=1}^{N}\sum_{j=i+1}^{N}\left(T_{Latent}(i,j) + T_3^{Right}(i,j) + \cdots + T_n^{Right}(i,j) + \cdots\right) \end{aligned} \tag{17}$$

where $T_{Latent}(i,j) = \frac{(D_{ij}-L_{ij})^2}{2L_{ij}}$, $T_n^{Right}(i,j) = \frac{(-1)^n(n-2)!}{L_{ij}^{n-1}}\frac{(D_{ij}-L_{ij})^n}{n!}$. We denote $T_{Right}(i,j) = T_{Latent}(i,j) + T_3^{Right}(i,j) + \cdots + T_n^{Right}(i,j) + \cdots$

The first term in (17) resembles the Sammon mapping but is weighted by latent space distance; potentially it can be used for MMDS. We define the LatentSammon mapping with the stress function

$$E_{Latent} = 2\sum_{i=1}^{N}\sum_{j=i+1}^{N}\frac{(D_{ij}-L_{ij})^2}{2L_{ij}} = 2\sum_{i=1}^{N}\sum_{j=i+1}^{N}T_{Latent}(i,j). \tag{18}$$

As shown in Fig. 4, the term $T_{Latent}(i,j)$ is not symmetric. $T_{Latent}(i,j) > T_{Sammon}(i,j)$ when $L_{ij} < D_{ij}$ and $T_{Latent}(i,j)$ becomes infinitely large as $L_{ij} \to 0$. $T_{Latent}(i,j) < T_{Sammon}(i,j)$ when $L_{ij} > D_{ij}$. The stress function (18) is not suitable for MMDS, because empirical minimisation is difficult to stabilise since the graph of $L_{ij} < D_{ij}$ is so steep, as shown by the experiment on the open box in Fig. 5(d) (the first order derivatives of the LatentSammon and RightSammon mappings used in optimisation are given in Section A). The performance is improved significantly if it is regularised (or shifted) with

$$E_{Latent}^{shifted} = \sum_{i=1}^{N}\sum_{j=i+1}^{N}\frac{(D_{ij}-L_{ij})^2}{L_{ij}+\xi}, \quad \xi > 0. \tag{19}$$

As shown in Fig. 4, the stress for $L_{ij} < D_{ij}$ is reduced, and thus convergence is improved as shown by the experiment on the open box illustrated in Fig. 5(e). Compared to the LeftSammon mapping result in Fig. 5(c), the overlapping between bottom and vertical sides reduces, especially the overlapping between the bottom and right side. The quality of the RightSammon mapping is better than the LatentSammon mapping as shown in Fig. 5(f).

There are interesting differences between the RightSammon mapping and the LeftSammon mapping. As stated earlier, in Fig. 2(a) and (b) we can see that for the LeftSammon mapping, the even order terms of the stress function are reflectively

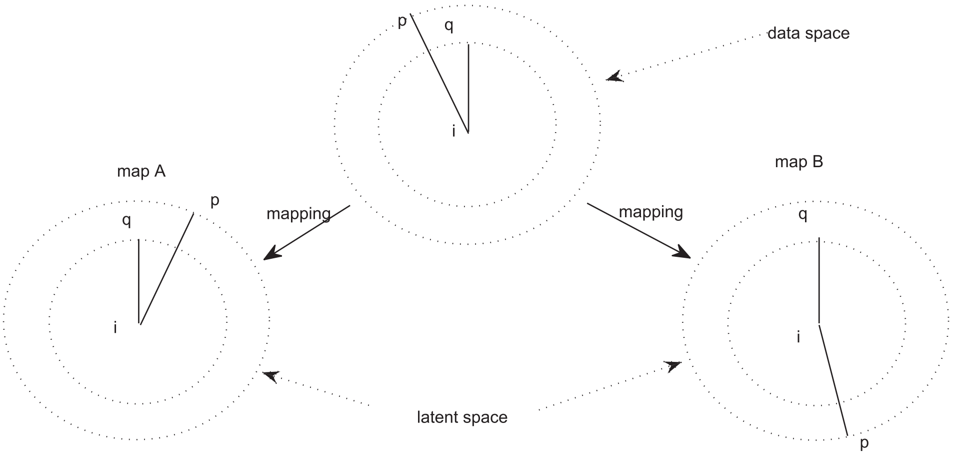


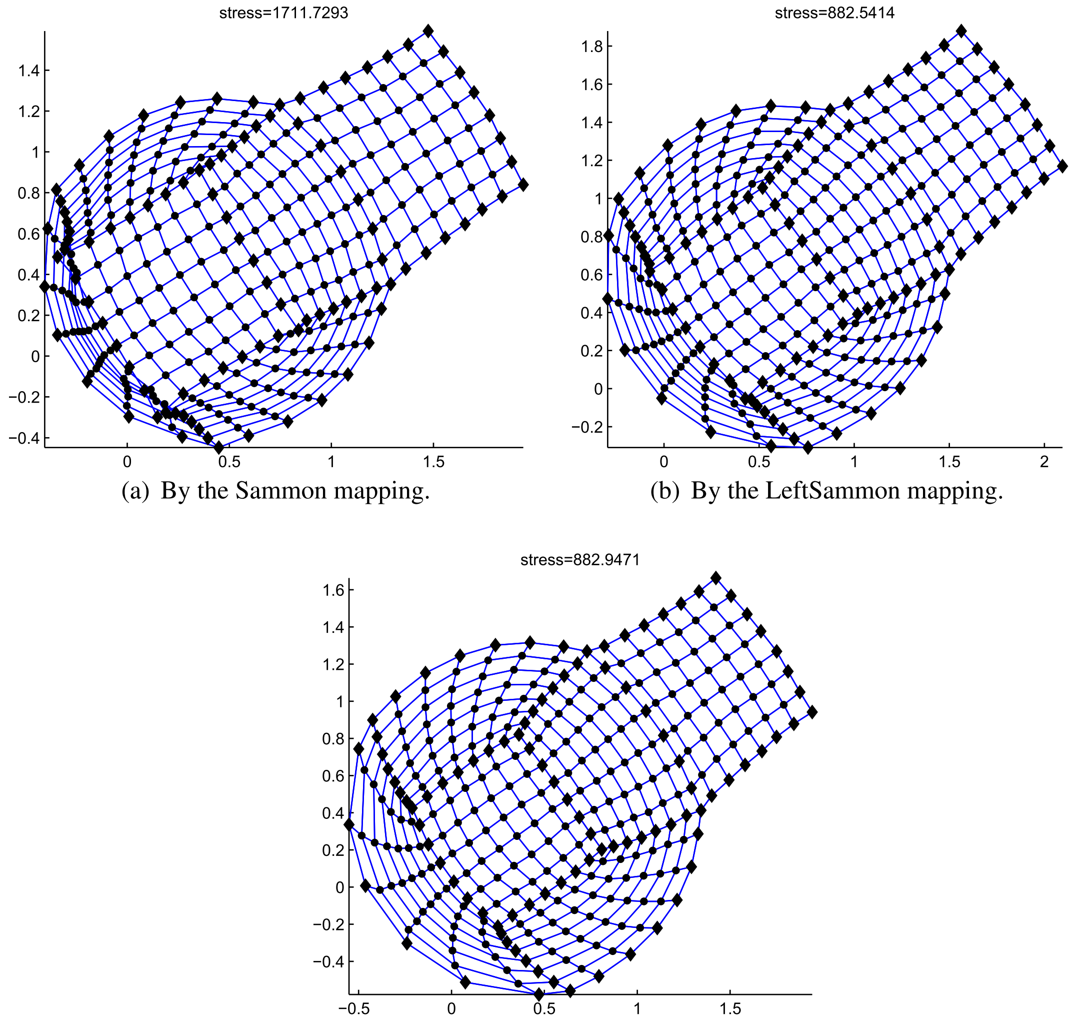
Fig. 8. Mapping of rank order.

symmetric with respect to the vertical line $L_{ij} = D_{ij}$, while the odd orders are rotationally symmetric with respect to $(D_{ij}, 0)$. The odd orders increase the stress for $L_{ij} < D_{ij}$ but decreases the stress for $L_{ij} > D_{ij}$ in order to increase the mapped distances. Fig. 6(a) shows the stress of the RightSammon mapping decomposition into the first four terms of the Taylor expansion; summing of the odd terms and even terms separately, we get Fig. 6(b). In Fig. 6(a) and (b) we can see that, although for the RightSammon mapping all even order terms are positive, they are not symmetric with respect to $L_{ij} = D_{ij}$. The stress in $L_{ij} < D_{ij}$ changes faster than in $L_{ij} > D_{ij}$. Contrary to the LeftSammon mapping, the odd order terms are negative when $L_{ij} < D_{ij}$, reducing the total stress, and are positive when $L_{ij} > D_{ij}$ increasing the total stress.

Fig. 6(c) shows stress with respect to distance error for varying distances in data space. The stresses are large for small distances when $L_{ij} < D_{ij}$ and the stress curves are separate from each other, that is to say, the RightSammon mapping is sensitive to shorter mapping distance error and accordingly gives tighter control. But for $L_{ij} > D_{ij}$ the stress curves are close to each other and stresses are small, so mapped distances can be longer. Again note that these effects are more pronounced for a small D_{ij}: the mapping will concentrate on getting right the latent positions of the points whose distances in data space is small.

6. Comparison of stress of the RightSammon, LeftSammon and Sammon mappings

Fig. 7(a) compares the stress of the Sammon, LeftSammon and RightSammon mappings. We can see that for $L_{ij} < D_{ij}$ the stress of the RightSammon mapping is the greatest, but for $L_{ij} > D_{ij}$ the stress of the RightSammon mapping is smallest. Both stress curves are tilting rightwards, as shown in Fig. 7(b), the RightSammon mapping surpasses the LeftSammon mapping in



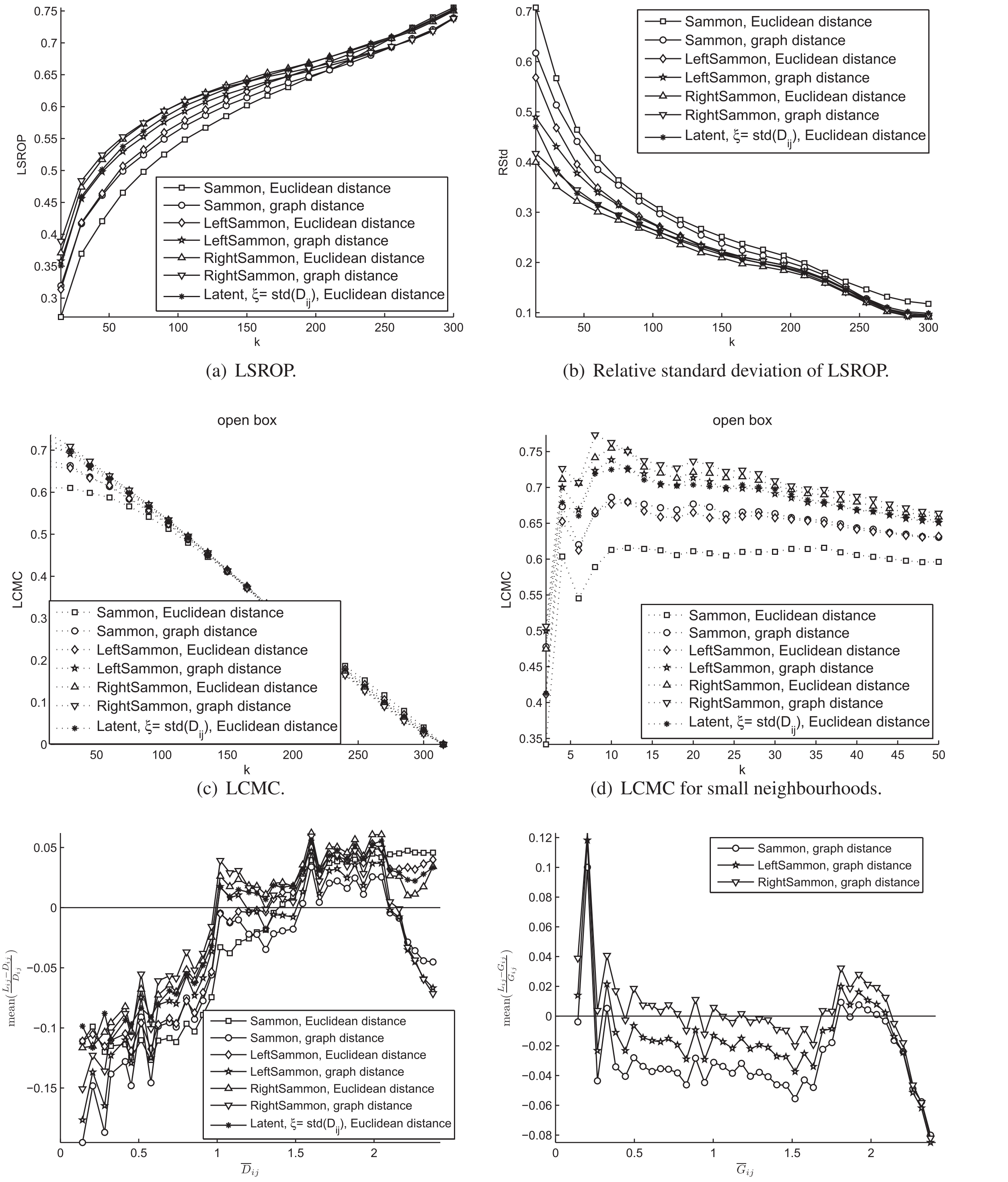
(a) By the Sammon mapping.

(b) By the LeftSammon mapping.

(c) By the RightSammon mapping.

Fig. 9. Experiment on open box using graph distance, $k = 10$ (the stresses displayed are of different stress functions).

its stress difference for the same magnitude negative and positive distance errors. Fig. 7(c) compares the RightSammon and LeftSammon mappings in the focusing power on small distances. It shows that the RightSammon mapping also focuses on small distances. For the same D_{ij}, the RightSammon mapping gives higher stress than the LeftSammon mapping for $L_{ij} < D_{ij}$ but lower stress for $L_{ij} > D_{ij}$, which increases the power of unfolding.



(a) LSROP.

(b) Relative standard deviation of LSROP.

(c) LCMC.

(d) LCMC for small neighbourhoods.

(e) Distance change, D_{ij} represents Euclidean distances.

(f) Relative distance change, G_{ij} represents graph distances.

Fig. 10. Assessment of the experiment on open box faces using Euclidean distance and graph distance.

Due to the above characteristics, apart from the experiment on the open box as depicted in Fig. 5(f) and (c), in Section 8 more examples of experiment will also show that the RightSammon mapping produces better embeddings than the LeftSammon mapping.

7. Visualisation quality assessment of different MMDS methods

Till now we have used the synthetic open box data set to show the performance of different mappings. But the open box data set is low dimensional and the experiment on it is only for illustration purposes. In the real world, data sets are usually high dimensional so we will test our different MMDS methods on high dimensional image data sets.

However, for a high dimensional data set, comparison of configuration quality between different MMDS methods by eye is not an easy task; this subjective comparison needs to be augmented quantitatively and with more analysis which hopefully will reveal why one projection is preferable to another. The question arises as to how to quantitatively assess the quality of the mapping which each method finds. The widely used stress comparison is only valid for projections made using the same criterion; stresses achieved by different methods are not comparable. Clearly if we use the stress function of the Sammon mapping, then we would expect the Sammon mapping scaling to do best since it is designed to optimise this criterion. So some non-stress quality assessment criteria have been proposed in the literature. Trustworthiness and continuity (T&C) [21], mean relative rank errors (MRREs) [10] are rank based criteria. LCMC [3] evaluates configuration quality in terms of neighbourhood preservation. Unifying work of T&C, MRREs and LCMC is found in [11]. We will review LCMC and propose a more strict rank based criterion in the following sections; in this paper we will use these two measures to assess mappings configured by different MMDS (we have similar results using the other criteria).

7.1. Local continuity meta-criterion (LCMC)

The LCMC [3], which is defined to be $\mathrm{LCMC}(K) = \frac{1}{Nk}\sum_{i=1}^{N}\left(C\left(\mathcal{N}_{data}^{k}(i) \cap \mathcal{N}_{latent}^{k}(i)\right) - \frac{k^2}{N-1}\right)$, where $C(\cdot)$ denotes set cardinality. $\mathcal{N}_{data}^{k}(i)$ and $\mathcal{N}_{latent}^{k}(i)$ stand for k nearest neighbours in data space and latent space respectively. The maximum of $C(\mathcal{N}_{data}^{k}(i) \cap \mathcal{N}_{latent}^{k}(i))$ is k, so the coefficient $\frac{1}{N}$ averages over all points, and $\frac{1}{k}$ is normalising the measure between 0 and 1. Since $\frac{1}{Nk}\sum_{i=1}^{N}C\left(\mathcal{N}_{data}^{k}(i) \cap \mathcal{N}_{latent}^{k}(i)\right)$ is 1 when $k = N-1$, $\frac{k^2}{N-1}$ is added as base-line. The higher the LCMC, the larger the number of neighbours of point i in data space remain neighbours in latent space.

7.2. Local strict rank order preservation

LCMC uses a single term to count neighbourhood preservation but it is weaker than rank order preservation. The rank order used in T&C and *MRREs* only considers distance. We propose a method known as local strict rank order preservation (*LSROP*) [16] which counts strictly preserved rank orders in the k-nearest neighbourhood.

In a high dimensional space, let there be three data points q, p and i satisfying $D_{iq} < D_{ip}$ (see Fig. 8); i, q and p are projected into a low dimensional space by map A or B. If $L_{iq} < L_{ip}$ and difference of $|\angle piq|$ in latent space and in data space is less than $\pi/4$, then we say a rank-order (neighbourhood relationship), $O_i(q,p)$, of p over q with respect to i is *strictly* preserved. Bothe map A and map B satisfy $L_{iq} < L_{ip}$, but the map B does not preserve $O_i(q,p)$ while A does, because the difference between $|\angle piq|$ in map B and in data space is greater than $\pi/4$. We require that *LSROP* is consistent with the principle of MMDS, distance preservation: the distance L_{pq} in map A is closer to D_{pq} than in map B.

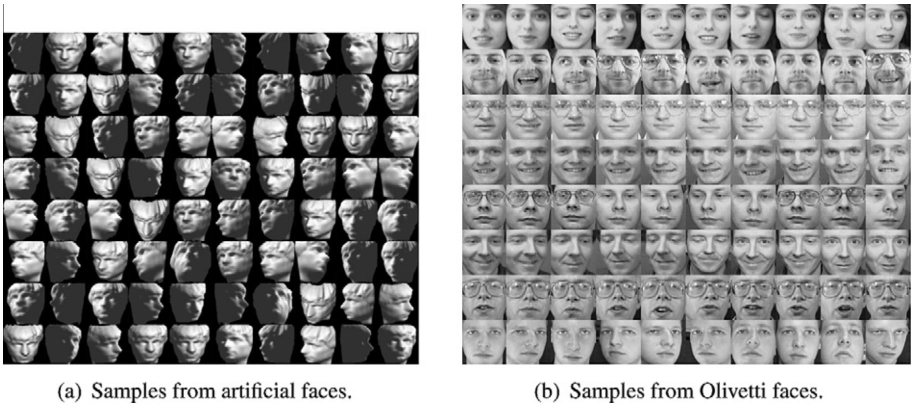
$\mathcal{N}_{data}^{k}(i)$ represents the k nearest neighbour points of point i in data space; $\mathcal{N}_{latent}^{k}(i)$ represents the k nearest neighbour points of point i in latent space. $\Omega(i,k) = \mathcal{N}_{data}^{k}(i) \cap \mathcal{N}_{latent}^{k}(i)$ represents the preserved data points in the neighbourhood, i.e. the common points of the k nearest neighbourhood in data space and latent space. $|\{O_i(q,p)\}|$, $p,q \in \Omega(i,k)$, represents the number of strictly preserved rank orders in a k-ary neighbourhood with respect to i. The ratio of rank orders strictly preserved in the k nearest neighbourhood is represented as

$$\phi(i,k) = \frac{1}{k}\sum_{p,q \in \Omega(i,k)} \frac{|\{O_i(q,p)\}|}{R_{data}(i,p)}$$

LSROP(k) is the mean of $\phi(i,k)$ over all data points

Table 1
Convergence time cost in seconds. It runs 25 times on desktop Dell Optiplex 755.

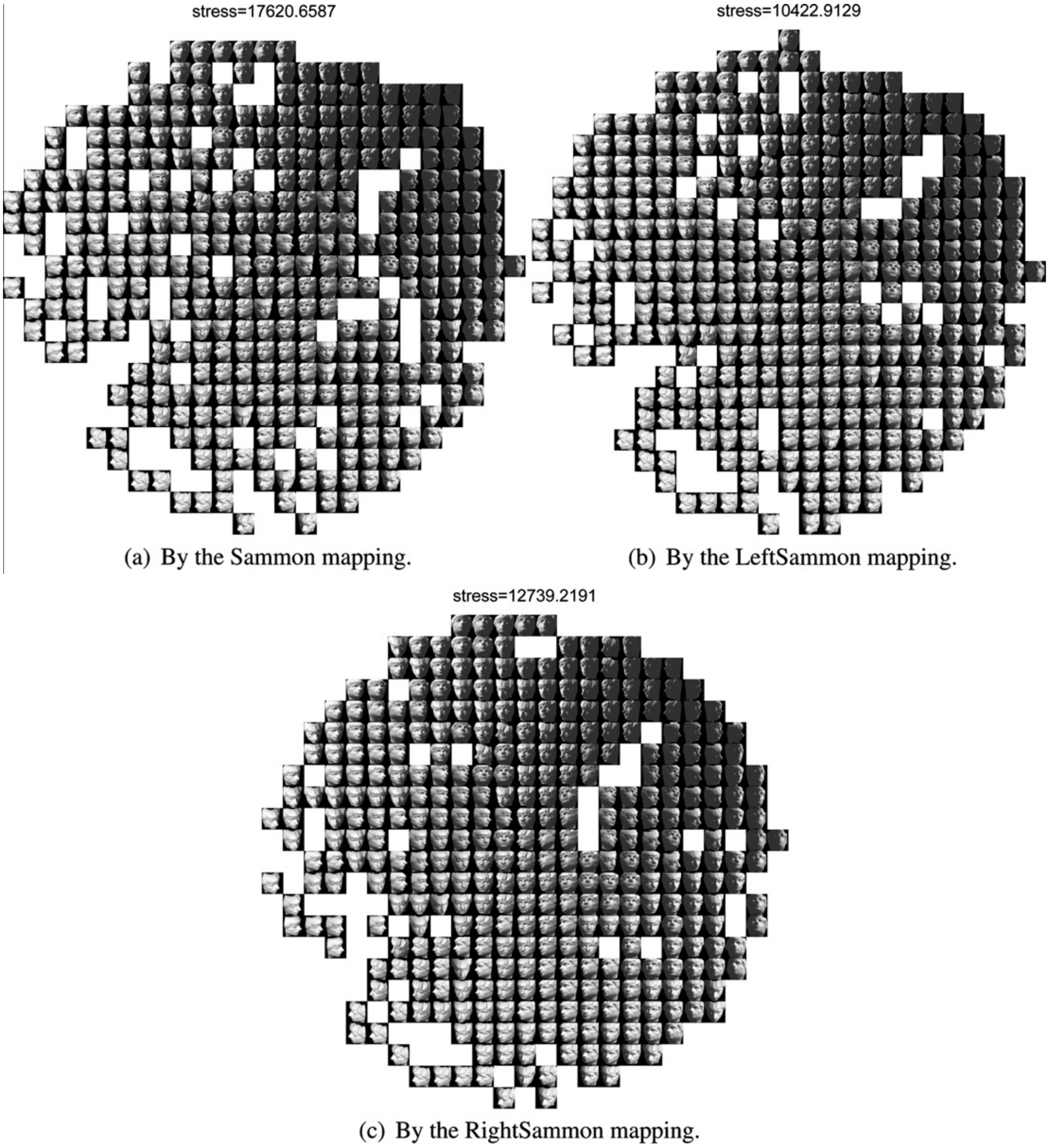
	Using Euclidean distance			Using graph distance		
	Open box	Artificial faces	Olivetti faces	Open box	Artificial faces	Olivetti faces
Sammon	2.4285	45.3577	26.1997	2.0290	29.6062	36.2320
LeftSammon	5.9369	86.5613	43.2182	4.0588	128.1782	53.3570
RightSammon	13.5358	165.7466	113.5680	9.9265	131.9943	63.4774



(a) Samples from artificial faces.

(b) Samples from Olivetti faces.

Fig. 11. Samples randomly drawn from artificial faces and Olivetti faces data set.



(a) By the Sammon mapping.

(b) By the LeftSammon mapping.

(c) By the RightSammon mapping.

Fig. 12. Experiment on artificial faces using Euclidean distance, maximum 23 data are sampled vertically (the stresses displayed are of different stress functions).

$$LSROP(k) = \frac{1}{N}\sum_{i=1}^{N}\phi(i,k) \tag{20}$$

LSROP is always between 0 and 1. The higher *LSROP* is, the better is the result. We will see empirically in Section 8 that *LSROP* differentiates mapping quality more precisely than LCMC.

The relative standard deviation of $\phi(i,k)$ is

$$RStd(k) = \frac{\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(\phi(i,k) - LSROP(k))^2}}{LSROP(k)},$$

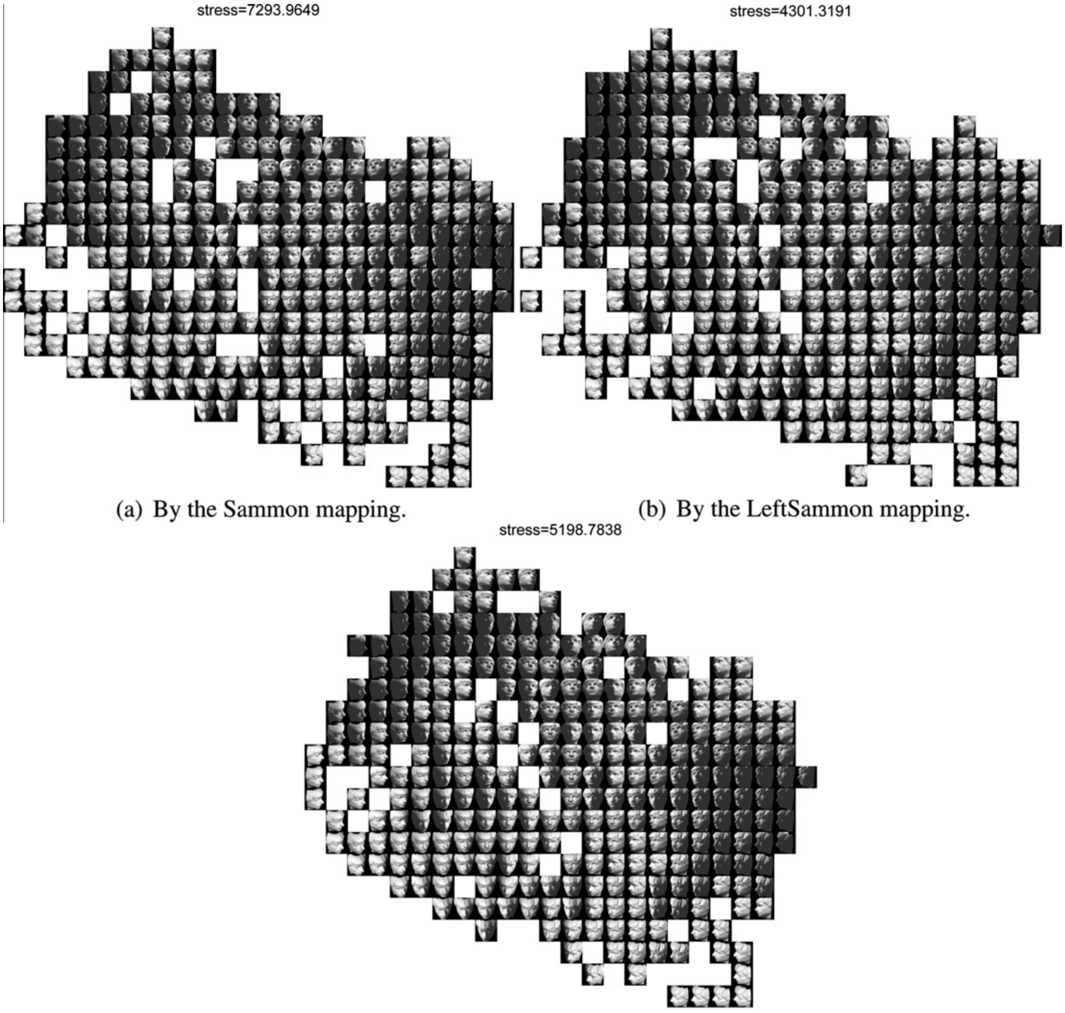
where $\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(\phi(i,k) - LSROP(k))^2}$ is the standard deviation of $\phi(i,k)$.

RStd is an auxiliary criterion to indicate the degree of diversity of *LSROP*. The lower *RStd* is, the lower is the diversity of the preserved strict orders.

8. Experiment on standard data sets

8.1. Use of graph distance in MMDS

Isomap [19] extends classical MMDS [20] by replacing Euclidean distance in data space with graph distance as an approximation to geodesic distance, on the theoretical basis that the data lies on a low dimensional manifold, assumed to be

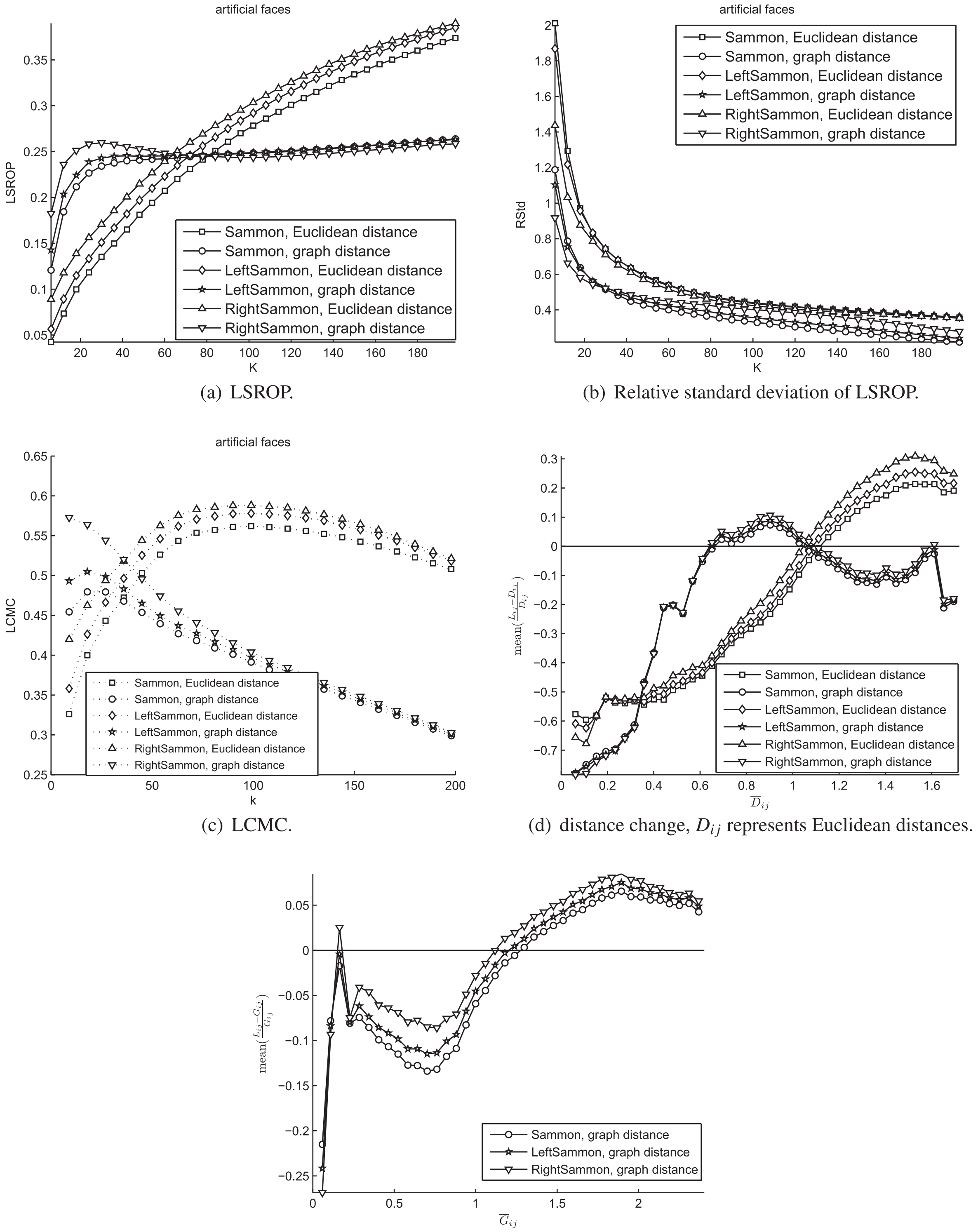


(a) By the Sammon mapping.

(b) By the LeftSammon mapping.

(c) By the RightSammon mapping.

Fig. 13. Experiment on artificial faces using graph distances, $k = 8$, maximum 20 data are sampled vertically (the stresses displayed are of different stress functions).



(a) LSROP.

(b) Relative standard deviation of LSROP.

(c) LCMC.

(d) distance change, D_{ij} represents Euclidean distances.

(e) Relative distance change, G_{ij} represents graph distances.

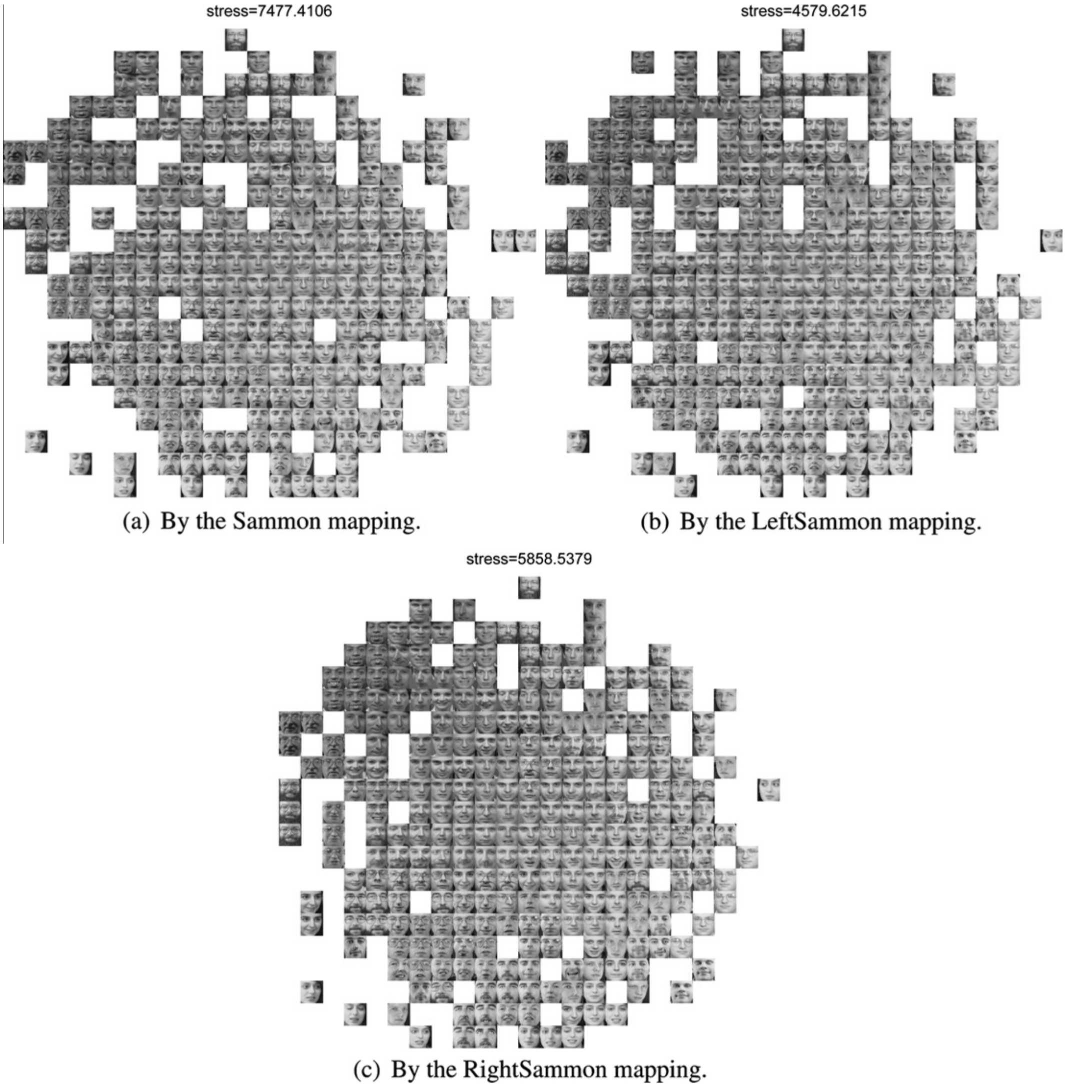
Fig. 14. Assessment of the experiment on the artificial faces using Euclidean distance and graph distance.

embedded in the high dimensional data space, and the graph distance approximates the distance across this manifold. After Isomap's success the metric replacement was applied to other distance preservation methods, as GeoNLM [22] extends the Sammon mapping, and CDA [9] extends CCA [6].

The first step in using graph distances is to get a k-ary or ϵ-radius neighbourhood for every data point in data space. To get k-ary neighbourhoods, in each column of the distance matrix D, we set the distance value of points outside a k-ary neighbourhood to be infinitely large. Since the density of data in data space is not even, the neighbourhood relationship is not always bi-directional. If j is in i's neighbourhood, but i might not be in j's neighbourhood, this means that the modified data matrix D is usually not symmetric, i.e. $D_{ij} = D_{ji}$ will no long hold for some i,j. Our method to make the distance matrix symmetric, or the neighbouring relationship bi-directional again is to set D_{ij} or D_{ji} to be the smaller of the two. After these processes, the minimum number of neighbours is k. Finally Dijkstra's algorithm is used to calculate the shortest path between all data points. The graph distance between point i and j is represented as G_{ij}.

8.2. Experiment on the open box using graph distance

The experiment using graph distances on the open box data set by the Sammon, LeftSammon and RightSammon mappings is illustrated in Fig. 9. Compared with using Euclidean distances as shown in Fig. 5, the overlap between the bottom and vertical sides is greatly reduced. Just as when using Euclidean distances, of the three methods, the result mapped with the RightSammon mapping is the best of all, and the LeftSammon mapping is better than the Sammon mapping.



(a) By the Sammon mapping.

(b) By the LeftSammon mapping.

(c) By the RightSammon mapping.

Fig. 15. Experiment on Olivetti faces using Euclidean distance, maximum 20 data are sampled vertically (the stresses displayed are of different stress functions).

We can see clearly in Fig. 10(a) that, in terms of *LSROP*, the RightSammon mapping outperforms the LeftSammon mapping and the LeftSammon is better than the Sammon mapping using both the Euclidean distance and graph distance; the RightSammon mapping also outperforms the Latent mapping using Euclidean distance with ξ as the standard deviation of distances in data space. The figure also shows that using graph distance achieves a better mapping than using Euclidean distance. LCMC in Fig. 10(c) and (d) also agree with the above in small neighbourhoods. Fig. 10(b) positively shows that a method which is high in *LSROP* has a low diversity of *LSROP*.

In order to compare distances in data space and latent space, in Fig. 10(e) the distances D_{ij} in data space are quantised into 40 sets; the mean of D_{ij} of each set, $\overline{D}_{ij}$, is represented on the horizontal axis. The vertical axis shows mean of $\frac{L_{ij}-D_{ij}}{D_{ij}}$ which indicates the relative distance change; it more accurately reflects the change in distances than $L_{ij} - D_{ij}$. In Fig. 10(e) we can see that using graph distances the short and long Euclidean distances in data space between data points are compressed more than using Euclidean distances. In Fig. 10(f), the distance changes are plotted against graph distance, where we can see that on average the graph distances are mapped shorter; for each method the longest graph distances are the most compressed. Of the three method, the RightSammon mapping is the least in graph distance compressing.

The average convergence time is shown in Table 1. It can be seen that the time needed by the RightSammon mapping is the longest; the Sammon mapping uses the least time; the computational cost using graph distances is slightly less than using Euclidean distances.

8.3. Experiment on artificial faces

The artificial faces data set is illustrated in Fig. 11(a); it was first used in the original Isomap paper [19]. It consists of 698 grey-scale images; an image consists of 64 × 64 = 4096 pixels and each pixel is stored as a dimension digitally. Although the image data is very high dimensional, its intrinsic dimensionality was designed to be only three: faces are posed up and down, left and right, plus a direction defined by the source of the light. Thus all the images are thought to be on a three dimensional manifold embedded in very high (4096) dimensional space. We will map onto a plane using Euclidean distance and graph distance respectively rather than a 3 dimensional latent space to save space in this paper.

Fig. 12(a)–(c) show mapping results by the Sammon, LeftSammon, and RightSammon mapping using Euclidean distance. As each MMDS method uses a different stress function to assess visualisation quality between multiple runs, the stress value could not be used to compare projection quality between different methods. We first identify the experimental results by eye then give quantitative mapping quality assessments. We can see that in the three figures, similar pictures are configured close to one another; lighting is strongest in the bottom left part, and changes to the darkest on the top right of each figure. In both the bottom left and top right areas we find nearby faces with opposite poses. The changing of lighting is comparatively smooth. In changes to both pose and lighting, the LeftSammon mapping produces smoother mapping than the Sammon mapping, and the RightSammon mapping produces the smoothest result. Fig. 13(a)–(c) show mapping results by the Sammon, LeftSammon, and RightSammon mapping using graph distance. We can see that all three figures are roughly in a parallelogram shape; interestingly the largest variations in both pose and lighting are well separated. This is the result of unfolding; the distances of these data points must be small in the Euclidean embedding, but large within the manifold. In the horizontal direction, pose changes from left to right; vertically faces are looking up to down, changing gradually; shades of light are in distinct areas of the mapping, changing frequently. Of the three methods, the RightSammon mapping gives the smoothest changes in both shade of lighting and pose, and its faces are most compactly allocated.

Fig. 14(a) depicts mapping quality measured in *LSROP* by the three methods using Euclidean distance and graph distance. Using Euclidean distance, for both small and large neighbourhood, the quality of all three methods improves, and the RightSammon mapping outperforms the LeftSammon mapping, and the Sammon mapping is the worst. Encouragingly the relative standard deviation of *LSROP* of the RightSammon mapping is the lowest for large k as in Fig. 14(b) indicating that its *LSROP* is evenly distributed. Using graph distances, the quality of the three methods increases rapidly up to about $k = 25$, then the quality of the Sammon and LeftSammon mappings levels off and the quality rises at a very low rate afterwards; the quality of the RightSammon mapping is higher than both the other two methods until about $k = 80$. For neighbourhood size up to about $k = 25$, the relative standard deviation of *LSROP* of the RightSammon mapping using graph distance is lower than the LeftSammon mapping as shown in Fig. 14(b). The quality using graph distance is much higher than using Euclidean distance until about $k \leqslant 60$. Fig. 14(c) shows that the mapping quality measured by LCMC using Euclidean distance increases for approximately $k < 85$, and the quality of using graph distance is higher than using Euclidean distance up to approximately $k < 30$. As can be seen, for small neighbourhoods, the mapping qualities measured by both *LSROP* and LCMC using graph distances are better than using Euclidean distance.

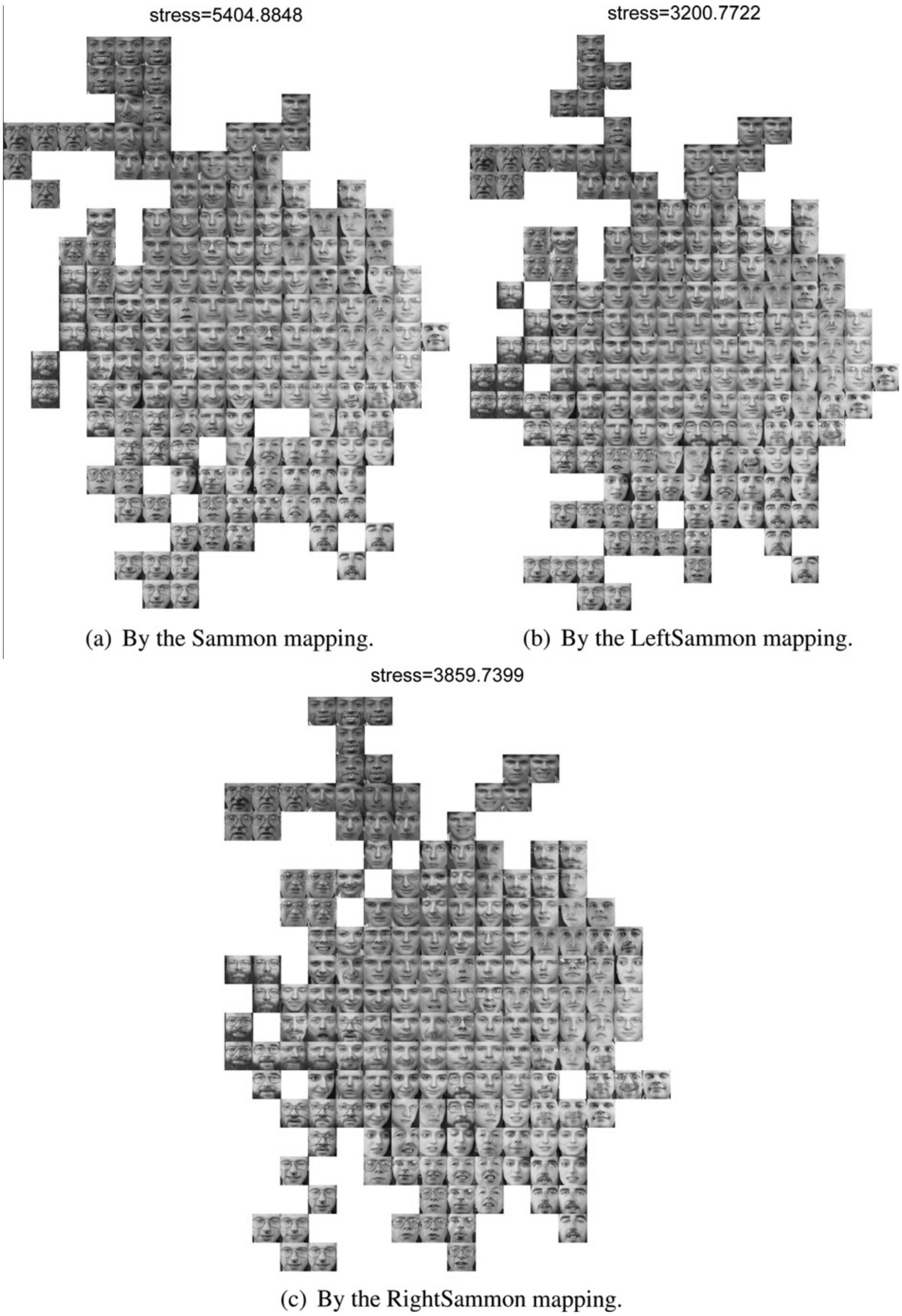
Fig. 14(d) shows the relative distance change of the three methods using two the kinds of distances. It worth noting that for the smallest distances ($D_{ij} < 0.2$) using the metrics, the RightSammon mapping compresses distance the most. For using the Euclidean distance, the latent distance mapped by the RightSammon mapping is the longest, then followed by the LeftSammon and Sammon mapping. Fig. 14(e) tells us that the shortest graph distances are compressed badly; the RightSammon mapping does it to the greatest degree. Both Fig. 14(d) and (e) on average verifies the conclusion that the RightSammon mapping maps distance longer than the LeftSammon mapping; similarly the LeftSammon mapping to the Sammon mapping. Using graph distance, the difference of distance change by the three methods are very small for $0.2 < D_{ij} < 0.7$. In

Fig. 14(d) using graph distance tends to compress the longest Euclidean distances while using Euclidean distance stretches the longest distances.

In Table 1, in return for a better mapping, the LeftSammon and RightSammon cost more time than the Sammon mapping using both the Euclidean and graph distances on the artificial faces data set.

8.4. Experiment on Olivetti faces

The Olivetti faces data set is images of 40 people. Ten different facial expression images are taken for each person; the size of the data set is 400 (examples in Fig. 11(b)). Each image is 64 × 64 grey scale. Its intrinsic dimensionality is unknown. The data set is available from http://cs.nyu.edu/∼roweis/data.html.

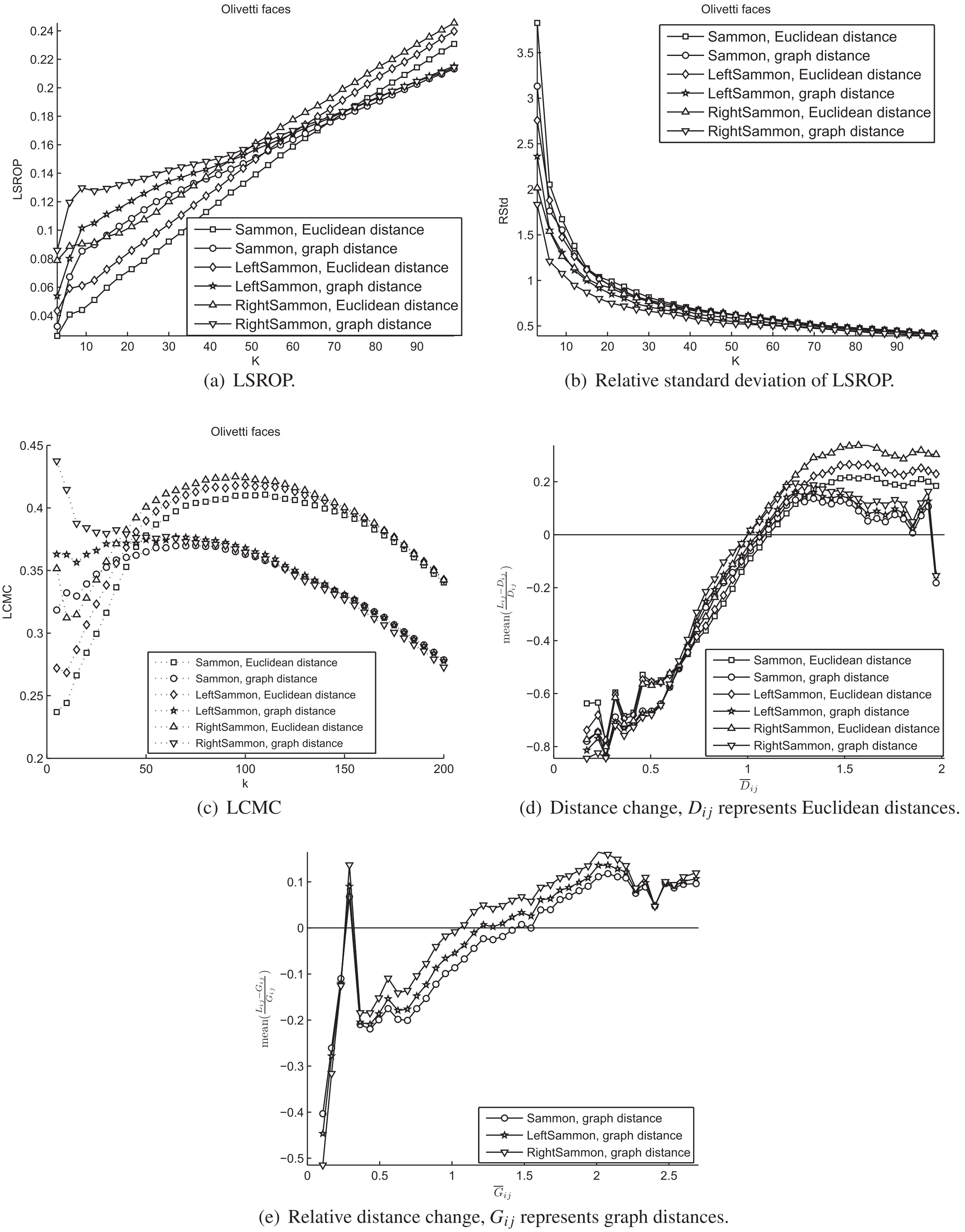


(a) By the Sammon mapping.

(b) By the LeftSammon mapping.

(c) By the RightSammon mapping.

Fig. 16. Experiment on Olivetti faces using graph distance, $k = 7$, maximum 20 data are sampled vertically (the stresses displayed are of different stress functions).



(a) LSROP.

(b) Relative standard deviation of LSROP.

(c) LCMC

(d) Distance change, D_{ij} represents Euclidean distances.

(e) Relative distance change, G_{ij} represents graph distances.

Fig. 17. Assessment of the experiment on Olivetti faces using Euclidean distance and graph distance.

Using Euclidean distances produces results displayed in Fig. 15(a)–(c). We can see that similar faces, for example happy faces, are allocated close to each other, and there is a changing of shade of grey scale from the upper-left corner to lower-right part, from darkest to lightest. The results using graph distances are shown in Fig. 16, similar images are further tightly clustered. The darkest faces are also on the upper-left and light images are still on the bottom-right positions.

Similar conclusions with the artificial faces can be drawn from Fig. 17. In Fig. 17(a) and (c) we can see that for small neighbourhoods using graph distances outperforms using Euclidean distances. Using both metrics, the RightSammon mapping is the best, and the LeftSammon mapping is the second best. From Fig. 17(b) we can see that *LSROP* of the best method, the RightSammon mapping using graph distances, is the most stable for a large size of k; the LeftSammon mapping using graph distances or the RightSammon mapping using Euclidean distance comes second best. Fig. 17(d) shows that the RightSammon mapping compresses the smallest Euclidean distances and long distances are mapped longer than other methods. Fig. 17(e) shows that the RightSammon mapping compresses smallest graph distances most.

From Table 1 we can see that the time cost of the RightSammon mapping is nearly triple that of the LeftSammon mapping in convergence using Euclidean distance; the time cost does not differ from each other so much when the distance is graph distance.

9. Conclusion

We have applied Bregman divergences in the context of metric multidimensional scaling in order to achieve better mappings for visualisation of high dimensional data sets. In doing so, we have shown that the popular Sammon mapping can be improved upon by using the left Bregman divergence with a specific underlying convex function. By considering the higher order expansion terms of the implicit Taylor series expansion in the Bregman formula, we were able to show why the new mapping outperforms the standard Sammon mapping.

We then created a new mapping called the RightSammon mapping because it uses the same underlying convex function though now the parameters to be adjusted, the latent point distances via the latent points' positions, appears in the right position in the divergences. Again we have analysed the convergence properties of this algorithm using the higher order terms of the implicit Taylor series expansion. This new mapping was shown to outperform even the LeftBregman divergence which itself outperforms the Sammon mapping. We showed this both through the mappings formed and by using both the LCMC and the newly proposed *LSROP* criterion which measures the goodness of the graphs.

Finally we have investigated using graph distances in data space to form the distances D_{ij} between data points and used the same algorithms as before on these distances. These new algorithms were shown to be the most effective of all in capturing both global and local structure of higher dimensional data in a lower dimensional projection.

Appendix A. Derivatives of stress functions used for stress optimisation

The distance between point i and j in latent space is $L_{ij} = \sqrt{\sum_{k=1}^{P}(Y_{ki} - Y_{kj})^2}$, where P is the dimensionality of latent space, usually $P = 2$; $\frac{\partial L_{ij}}{\partial Y_{ki}} = \frac{Y_{ki} - Y_{kj}}{L_{ij}}$.

A.1. The LeftSammon mapping

$$\frac{\partial E_{Left}}{\partial Y_{ki}} = \frac{\partial E_{Left}}{\partial L_{ij}} \frac{\partial L_{ij}}{\partial Y_{ki}} = \sum_{j=i+1}^{N} \ln \frac{L_{ij}}{D_{ij}} \frac{Y_{ki} - Y_{kj}}{L_{ij}} = \sum_{j=i+1}^{N} \frac{\ln L_{ij} - \ln D_{ij}}{L_{ij}} (Y_{ki} - Y_{kj}).$$

A.2. The RightSammon mapping

$$\frac{\partial E_{Right}}{\partial L_{ij}} = 1 - \frac{D_{ij}}{L_{ij}},$$

$$\frac{\partial E_{Right}}{\partial Y_{ki}} = \frac{\partial E_{Right}}{\partial L_{ij}} \frac{\partial L_{ij}}{\partial Y_{ki}} = \sum_{j=i+1}^{N} \left(1 - \frac{D_{ij}}{L_{ij}}\right) \frac{Y_{ki} - Y_{kj}}{L_{ij}}.$$

A.3. The LatentSammon mapping

$$\frac{\partial E_{Latent}}{\partial L_{ij}} = \frac{2(L_{ij} - D_{ij})L_{ij} - (D_{ij} - L_{ij})^2}{L_{ij}^2} = 1 - \frac{D_{ij}^2}{L_{ij}^2},$$

$$\frac{\partial E_{Latent}}{\partial Y_{ki}} = \frac{\partial E_{Latent}}{\partial L_{ij}} \frac{\partial L_{ij}}{\partial Y_{ki}} = \sum_{j=i+1}^{N} \left(1 - \frac{D_{ij}^2}{L_{ij}^2}\right) \frac{Y_{ki} - Y_{kj}}{L_{ij}} = \sum_{j=i+1}^{N} \frac{L_{ij}^2 - D_{ij}^2}{L_{ij}^3} (Y_{ki} - Y_{kj}),$$

$$\frac{\partial E^{shifted}_{Latent}}{\partial L_{ij}} = 1 - \frac{(D_{ij}+\xi)^2}{(L_{ij}+\xi)^2},$$

$$\frac{\partial E^{shifted}_{Latent}}{\partial Y_{ki}} = \frac{\partial E^{shifted}_{Latent}}{\partial L_{ij}} \frac{\partial L_{ij}}{\partial Y_{ki}} = \sum_{j=i+1}^{N} \left(1 - \frac{(D_{ij}+\xi)^2}{(L_{ij}+\xi)^2}\right) \frac{Y_{ki} - Y_{kj}}{L_{ij}}.$$

References

[1] Arindam Banerjee, Srujana Merugu, Inderjit S. Dhillon, Joydeep Ghosh, Clustering with Bregman divergences, Journal of Machine Learning Research 6 (2005) 1705–1749.

[2] Ingwer Borg, Patrick J.F. Groenen, Modern Multidimensional Scaling, second ed., Springer, New York, 2005.

[3] Lisha Chen, Andreas Buja, Local multidimensional scaling for nonlinear dimension reduction, graph drawing and proximity analysis, PhD Thesis, University of Pennsylvania, 2006.

[4] Aynur Dayanik, Feature interval learning algorithms for classification, Knowledge-Based Systems 23 (5) (2010) 402–417.

[5] Pierre Demartines, Jeanny Hérault, Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets, IEEE Transactions on Neural Networks 8 (1) (1997).

[6] Jeanny Hérault, Claire Jausions-Picaud, Anne Guérin-Dugué, Curvilinear component analysis for high-dimensional data representation: I. Theoretical aspects and practical use in the presence of noise, in: IWANN (2), 1999, pp. 625–634.

[7] Jiani Hu, Weihong Deng, Jun Guo, Weiran Xu, Learning a locality discriminating projection for classification, Knowledge-Based Systems 22 (8) (2009) 562–568.

[8] Joseph B. Kruska, Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis, Psychometrika 29 (1964) 1–27.

[9] John A. Lee, Amaury Lendasse, Michel Verleysen, Curvilinear distance analysis versus Isomap, in: Proceedings of ESANN 2002, 10th European Symposium on Artificial Neural Networks, 2000, pp. 185–192.

[10] John A. Lee, Michel Verleysen, Nonlinear Dimensionality Reduction, first ed., Information Science and Statistics, Springer, 2007.

[11] John A. Lee, Michel Verleysen, Quality assessment of dimensionality reduction: Rank-based criteria, Neurocomputing 72 (7–9) (2009) 1431–1443.

[12] R.C.T. Lee, J.R. Slagle, H. Blum, A Triangulation Method for Sequential Mapping of Points from N-Space to Two-Space, IEEE Transactions on Computers 6 (1977) 288–292.

[13] Jianchang Mao, A.K. Jain, Artificial neural networks for feature extraction and multivariate data projection, IEEE Transactions on Neural Networks 6 (2) (1995) 296–317.

[14] Elzbieta Pekalska, Dick de Ridder, Robert P.W. Duin, Martin A. Kraaijveld, A new method of generalizing Sammon mapping with application to algorithm speed-up, in: 5th Annual Conference of the Advanced School for Computing and Imaging, ASCI 1999, Heijen, NL, June 1999, pp. 221–228.

[15] John W. Sammon, A nonlinear mapping for data structure analysis, IEEE Transactions on Computing 18 (1969).

[16] Jigang Sun, Extending metric multidimensional scaling with Bregman divergences, PhD Thesis, University of the West of Scotland, Paisley, Scotland, March 2011.

[17] Jigang Sun, Malcolm Crowe, Colin Fyfe, Curvilinear component analysis and Bregman divergences, in: European Symposium on Artificial Neural Networks (Esann), d-side publications, 2010, pp. 81–86.

[18] Jigang Sun, Malcolm Crowe, Colin Fyfe, Extending metric multidimensional scaling with Bregman divergences, Pattern Recognition 44 (5) (2011) 1137–1154.

[19] Joshua B. Tenenbaum, Vin de Silva, John C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.

[20] Warren S. Torgerson, Multidimensional scaling: I. theory and method, Psychometrika 17 (1952) 401–419.

[21] Jarkko Venna, Samuel Kaski, Local multidimensional scaling, Neural Networks 19 (6) (2006) 889–899.

[22] Li Yang, Sammon's nonlinear mapping using geodesic distances, in: ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04), Washington, DC, USA, vol. 2, IEEE Computer Society, 2004, pp. 303–306.