Incorporating Visualisation Quality Measures to Curvilinear Components Analysis

Jigang Sun, Malcolm Crowe, Colin Fyfe

University of the West of Scotland, UK

Abstract
Curvilinear Component Analysis (CCA) is a useful data visualisation method. CCA has the technical property that its optimisation surface, as defined by its stress function, changes during the optimisation according to a decreasing parameter. CCA uses a variant of the stochastic gradient descent method to create a mapping of data. In the optimisation method of CCA, the stress function is only a general guide towards an acceptable mapping. In other multidimensional scaling methods such as Sammon’s mapping, the best mapping among multiple runs from different initialisations can be chosen by selecting the mapping with the lowest stress, whereas in CCA the embedding is simply the result of one run, surely we can have multiple stars. As a consequence of the absence of an objective function to be used as a selection criterion, embedding made by CCA can be poorly optimised. In this paper we present a new way of improving the optimisation of CCA by integrating non-stress data visualisation quality measures into the existing algorithm. We first use data visualisation quality measures to select the best mapping from multiple runs of a standard stochastic gradient descent implementation; then we tune various parameters involved to achieve further enhancement.

Keywords: Curvilinear Component Analysis (CCA); stochastic gradient descent; parameter learning

1. Introduction
In this paper, we consider improvements to methods of extracting information from high dimensional data sets; in particular, we consider methods which project data onto a lower dimensional manifold in order to visualise the structure in a data set by eye. Examples of this type of method include principal component analysis (PCA) [9] and exploratory projection pursuit [7], both of which give linear projections i.e. find linear subspaces. However much modern research has been into methods which find nonlinear manifolds onto which a data set can be projected.

Metric multidimensional scaling (MMDS) [2] is one such group of visualisation and dimensionality reduction methods that projects data points from high-dimensional data space to low-dimensional space in which the structure of the original data set can be identified by eye. It creates the low-dimensional representation by keeping distances between the output space points as close as possible to the distances between the original data points by moving the output space points around in order to minimise a stress function. MMDS can be a preprocessing step for clustering [10] or classification. There exist newer methods such as locally linear embedding (LLE) [15], which projects data points from a high-dimensional space to a low-dimensional space by assuming that a data point is a linear combination of its near neighbours in data space and the linearity still holds in the low-dimensional space, and the self-organising map (SOM) [11], which is an artificial neural network that uses unsupervised competitive learning to produce a low-dimensional discretised representation of the input space. However traditional methods are widely used and still of research interest because a traditional MMDS method can capture the global structure of the data while simultaneously paying attention to local structure.

However it is well known that such methods are prone to finding local optima: in [23, p101], it is shown that, even on the very simple 4-dimensional iris data set of only 150 samples, the global minimum of its stress (or objective)
function is found only 21.3% of the time by the Sammon mapping algorithm over 1000 runs from different random initialisations. That is, almost 80% of the simulations stopped in a configuration which was not the global optimum. Therefore, in this paper, we investigate optimising the parameters of a CCA mapping using different criteria.

The notation used in this paper is as follows. $X$ stands for the $m$ dimensional original data set, $Y$ stands for the configuration of $X$ in the $p$ dimensional output space; $m \gg p$, usually $p = 2$. $D$ stands for the distance matrix in the data space and $L$ stands for the distance matrix in the output space. The vector $X_i = (X_{1i}, X_{2i}, \cdots, X_{mi})$ stands for the data point $i$ in the data space and vector $Y_i = (Y_{1i}, Y_{2i}, \cdots, Y_{pi})$ stands for the projected point in the output space. $D_{ij}$ stands for the inter point distance between $X_i$ and $X_j$; $L_{ij}$ stands for the inter point distance between $Y_i$ and $Y_j$. The size of the data set is $N_i$; $i, j = 1, 2, \cdots, N$.

In section 2, Curvilinear Component Analysis (CCA) [6] is briefly reviewed; its stochastic gradient descent algorithm is reviewed in section 3. Section 4 presents the SOM Toolbox [27] implementation of the optimisation algorithm. Section 5 summarises relevant existing rank based visualisation quality measures. Section 6 introduces ways of using these quality assessment criteria to select the best mapping from multiple runs of the existing algorithm implementation; via investigation of the influences of various parameters experimentally, a parameter learning process is proposed afterwards. Finally section 8 concludes the paper.

2. Curvilinear Component Analysis

Curvilinear Component Analysis (CCA) [6] is an interesting flavour of multidimensional scaling in that unlike the Sammon mapping [16] whose stress function uses weights which are functions of distances in data space, it uses weights which are functions of distances in the output space. It is claimed in [6] to be an improvement to the SOM in that the output space is not a fixed lattice but a continuous space. CCA is similar to SOM in that it uses a weight function parameterised by a decreasing parameter; the weight function corresponds to the neighbourhood function of the SOM.

In [6] the stress function of CCA is generalised as

$$E_{CCA}(Y) = \sum_{i=1}^{N} \sum_{j=i+1}^{N} (L_{ij} - D_{ij})^2 w_{ij}(L_{ij}, \lambda).$$

(1)

The weight function $w_{ij}(L_{ij}, \lambda)$, where $\lambda$ is a parameter, is a function of the distance $L_{ij}$, and the shape of the function can be a decreasing exponential, sigmoid or Lorentz function or a Heaviside step function. As with all MDS methods, the stress function is minimized by moving the positions of the points in the output space around. We can see that CCA is different from the Sammon mapping not only in that the weight function $w_{ij}(L_{ij}, \lambda)$ has as argument the inter-point distance in the low-dimensional space rather than data space as with most MDS algorithms, but also that, like the SOM, the weight has an extra evolving parameter, $\lambda$, which decreases with time during the optimisation. This is to say that the stress function is changing at all times during optimisation. [6] emphasises the situation where the weight function is a Heaviside step function.

2.1. Step CCA

When the weight is the Heaviside step function

$$w_{ij}(L_{ij}, \lambda) = \begin{cases} 1 & \text{if } L_{ij} \leq \lambda \\ 0 & \text{if } L_{ij} > \lambda \end{cases}$$

we call the resulting algorithm Step CCA. It is good at unfolding strongly non-linear or even closed structures by tearing the manifold. Let us illustrate this on the open box which is shown in Fig. 1(a). The open box data set consists of 316 data points. Page 15 of [12] 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).”. This data is not commonly used in the literature, but we have found that it is very useful to illustrate the comparison between different methods by eye. The tearing effects are seen in Figure 1(b).

Step CCA is so successful that it has received a great deal of attention in the literature. In the literature it is used as a standard CCA, such as in [25], [12, p215, p223], where Step CCA is compared with other methods such as the Sammon mapping, PCA [9], Isomap [22], etc..
2.2. Exponential CCA

Besides Step CCA, another version was implemented in the SOM Toolbox [27] in Matlab (as the file CCA.m in the toolbox). The author of the program and the research community as in [12, p91] thought this version used a global exponential weighting in (1) with

\[ w_{ij}(L_{ij}, \lambda) = e^{-\frac{L_{ij}}{\lambda}}; \]  

so that its stress function would be

\[ E_{Exp}(Y) = \sum_{i=1}^{N} \sum_{j=i+1}^{N} (L_{ij} - D_{ij})^2 e^{-\frac{L_{ij}}{\lambda}}; \]  

In fact the stress function for CCA.m is better than this: we present the correct function for the SOM Toolbox implementation below. Let

\[ T_{ij}^{Exp} = \frac{1}{2} (L_{ij} - D_{ij})^2 e^{-\frac{L_{ij}}{\lambda}} \]  

Figure 1: Open box data set and its mappings.
represent the contribution of the data point pair \((i,j)\) to the stress function (3) (ignoring the constant coefficient). \(T^\text{Exp}_{ij}\) is plotted in Figure 2(a) for various values of \(L_{ij}\) when we fix \(D_{ij}\). Since

\[
\frac{dT^\text{Exp}_{ij}}{dL_{ij}} = (L_{ij} - D_{ij})e^{-\frac{L_{ij}}{\lambda}} + \frac{(L_{ij} - D_{ij})^2}{2\lambda} e^{-\frac{L_{ij}}{\lambda}}
= (L_{ij} - D_{ij})e^{-\frac{L_{ij}}{\lambda}} \left(1 - \frac{L_{ij} - D_{ij}}{2\lambda}\right)
\]

(5)

The finite solutions of \(\frac{dT^\text{Exp}_{ij}}{dL_{ij}} = 0\) are \(L_{ij} = D_{ij}\) and \(L_{ij} = D_{ij} + 2\lambda\). They are the bottom and peak of the stress graph as presented in Figure 2(a). It is clear that the stress function (3) is monotonically decreasing with respect to the low-dimensional space distances \(L_{ij} \in [0, D_{ij}]\); it is increasing for \(L_{ij} \in (D_{ij}, D_{ij} + 2\lambda)\); but beyond the peak at \(L_{ij} = D_{ij} + 2\lambda\) the stress decreases with low-dimensional space distance. Obviously when data are configured infinitely far apart, \(L_{ij} = +\infty\), the stress function is minimised to the lowest value, zero; but this does not give us a solution with which to best visualise the data. The shape of the stress function (3) makes it unfit to be used with any gradient descent method as shown by the mapping of the open box in Figure 1(c): some points ended up projected into poor positions because the low-dimensional space distances from the two escaping points to all other points are so large that they are beyond the peak at \(L_{ij} = D_{ij} + 2\lambda\) as shown in Figure 2(a) and cannot come back to the left basin.

![Graphs](image)

(a) The shape of (4) makes convergence difficult. (b) Term formation of stress of the RightExp, truncated to the first four terms.

Figure 2: The stress of Exponential CCA.

In [18] we proved that the real stress function implemented in the SOM Toolbox is not (3), but

\[
B_{\text{RightExp}}(Y) = \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda^2 \left(e^{\frac{L_{ij}}{\lambda}} - e^{-\frac{L_{ij}}{\lambda}} + (D_{ij} - L_{ij}) \frac{e^{\frac{L_{ij}}{\lambda}} - 1}{\lambda}\right)
\]

(6)

We call the above stress function the RightExp mapping; it is interchangeable with the Exponential CCA in this paper. (6) is equivalent to the truncated Taylor series

\[
B_{\text{RightExp}}(Y) = \sum_{i=1}^{N} \sum_{j=1}^{N} e^{\frac{L_{ij}}{\lambda}} (D_{ij} - L_{ij})^2 \frac{2}{2!} + e^{\frac{L_{ij}}{\lambda}} (D_{ij} - L_{ij})^3 \frac{3}{3!} + \cdots + e^{\frac{L_{ij}}{\lambda}} (D_{ij} - L_{ij})^n \frac{(-\lambda)^n - 1}{n!} + \cdots
\]

(7)

\[
= \sum_{i=1}^{N} \sum_{j=1}^{N} (T^\text{Exp}_{ij} + T^3_{ij} + \cdots + T^n_{ij} + \cdots)
\]

\[
= \sum_{i=1}^{N} \sum_{j=1}^{N} T^\text{RightExp}_{ij}
\]


where \( T_{ij}^3 = -\frac{t_{ij}}{\lambda} (D_{ij}^\alpha L_{ij})^\alpha \), \( T_{ij}^1 = -\frac{t_{ij}}{\lambda^2} (D_{ij}^\alpha L_{ij})^\alpha \), and \( T_{ij}^{\text{RightExp}} = \lambda^2 \left( e^{-\frac{t_{ij}}{\lambda}} - e^{-\frac{t_{ij}}{\lambda} + (D_{ij} - L_{ij}) \frac{t_{ij}}{\lambda}} \right) \).

As illustrated in Figure 2(b), \( T_{ij}^{\text{Exp}} \) is a bad approximation of \( T_{ij}^{\text{RightExp}} \) for \( L_{ij} > D_{ij} \) where other higher order terms, \( T_{ij}^3 \), \( T_{ij}^4 \) and so on, cannot be ignored. The RightExp mapping outperforms (3) in convergence: data points being configured infinitely far apart is no longer encouraged; and beyond \( L_{ij} = D_{ij} + 2\lambda \), \( T_{ij}^{\text{RightExp}} \) will not decrease to zero but instead it levels off; which also indicates that it is good for unfolding, i.e., distances which are mapped larger than their original values are not seriously penalised in stress. It is also worth noting that the graph \( T_{ij}^{\text{RightExp}} \), for the RightExp mapping, is not symmetric: the left part where \( L_{ij} < D_{ij} \) is steeper than the right part where \( L_{ij} > D_{ij} \), so unfolding is easier to achieve. The embedding of the open box by the RightExp mapping is shown in Figure 1(d) which is much better than the mapping created by (3) (shown in Figure 1(c)) in that there is no point pushed out.

The bad projection in Figure 1(c) is the global minima, or one of the acceptable local minima, of stress function (3). The problem is caused by stress function definition. The author of the SOM Toolbox implemented (6) but he thought the stress function he implemented is (3); in other words he could have thought that Figure 1(d) was created by (3) other than (6).

Possibly because of the problems with (3), Exponential CCA has received comparatively little attention by the research community in the literature and more research is needed; so we will focus on Exponential CCA in this paper.

3. Optimisation algorithm

The optimisation method used for CCA is a kind of stochastic gradient descent; the optimisation method is considered as important as the CCA stress functions in [8]. This is because of the decreasing parameter \( \lambda \), and different implementations will produce different embeddings. The basic idea of the CCA optimisation algorithm is that, at each iteration, one output space point \( i \) is randomly chosen and \( Y_i \) is fixed, and all the other points \( j \neq i \) are relocated with respect to this fixed point. Specifically, first the whole stress function is broken into parts: (1) is rewritten as \( E_{\text{CCA}}(Y) = \sum_{i=1}^{N} E_{\text{CCA}}(Y) \). Then, in each iteration when the index \( i \) has been chosen, the updating rule for the output space point \( Y_j \) is

\[
Y_j \leftarrow Y_j - \alpha \frac{\partial E_{\text{CCA}}(Y)}{\partial Y_j} = Y_j - \alpha \frac{\partial E_{\text{CCA}}(Y)}{\partial L_{ij}} \frac{\partial L_{ij}}{\partial Y_j}, \quad j \neq i,
\]

where \( \alpha \) is a decreasing learning rate; one functional form of the decreasing alpha is suggested as \( \alpha(t) = \frac{\alpha_0}{1+t} \) in the original CCA paper [6].

Since the distance between point \( i \) and \( j \) in the low-dimensional space is

\[
L_{ij} = \sqrt{\sum_{h=1}^{p} (Y_{ih} - Y_{jh})^2},
\]

where \( p \) is the dimensionality of the output space, this gives \( \frac{\partial L_{ij}}{\partial Y_i} = \frac{Y_{ih} - Y_{jh}}{L_{ij}} \) or \( \frac{\partial L_{ij}}{\partial Y_j} = \frac{Y_{jh} - Y_{ih}}{L_{ij}} \).

For Step CCA, the updating rule is

\[
\begin{cases}
   Y_j \leftarrow Y_j - \alpha \frac{(L_{ij} - D_{ij})}{L_{ij}} (Y_i - Y_j) & \text{if } L_{ij} \leq \lambda \\
   Y_j \text{ does not change} & \text{if } L_{ij} > \lambda
\end{cases}
\]

For the Exponential CCA, the updating rule is

\[
Y_j \leftarrow Y_j - \alpha \frac{(L_{ij} - D_{ij})}{L_{ij}} e^{-\frac{t_{ij}}{\lambda}} (Y_i - Y_j)
\]

which is consistent with (6) and this gradient descent optimization update rule is implemented in the SOM Toolbox [27] to be reviewed in the next section.

From the above we can see that when the point \( j \) is relocated, only the value of the fixed point \( i \) and the current point \( j \) affect the relocation of \( j \), and since \( i \) is constant at this stage, the stress function is actually a \( p \)-dimensional...
function of point \( j \). If \( p = 2 \), during the optimisation process the stress function is treated as two dimensional, which makes jumping out of a local minimum easy and could explain the advantage claimed (\[6\], Figure 4) that convergence is much faster than the standard gradient descent method used by the Sammon mapping. Figure 3 shows an example where, in the high dimensional space, point \( X_4 \) is close to the center of the other three points; in the embedding, \( Y_4 \), corresponding to \( X_4 \), is initialised outside the corresponding locations of the three other points such that \( L_{14} \) (the distance between \( Y_1 \) and \( Y_4 \)) is much greater than its original distance \( D_{14} \). If the optimisation algorithm is the traditional gradient descent, when \( Y_4 \) moves close to \( Y_1 \), \( L_{24} \) and \( L_{34} \) will also become shorter than their original distances such that the overall stress increases to hold back the change. But for the stochastic gradient descent, when point 1 is fixed, this movement has nothing to do with the points 2 and 3, consequently the position changing is faster. The next updates to point 4 in the next iteration will have a new fixed point \( i \) and will thus provide contributions from another point. With the progress of such learning the embedding improves hopefully towards structure appearing. It is worth noting that during the process, when the location of one point \( j \) is changed, only one distance need to be recalculated; but using a standard gradient descent adaptation, for the change of one data point, \( N - 1 \) distances have to be recalculated.

![Figure 3: Possible local minimum occurs during stress optimisation.](image)

One feature of CCA is that \( \lambda \) can decrease during the optimisation. Even using the standard gradient descent algorithm rather than the stochastic one for Step CCA, when \( \lambda \) decreases, more and more terms become zero, i.e., less and less terms are involved, and the change of coordinates in the low-dimensional space becomes easier. For the exponential version, the shape of the stress curve changes dramatically with the decrease of \( \lambda \): more and more terms decrease in magnitude, which leads to a decreasing learning rate, \( \alpha \), being chosen instead of the constant factor adopted by the Sammon mapping.

A decreasing learning rate takes into account the contributions of the stress function at various \( \lambda \). The updating rules do not optimise the stress function for a specific \( \lambda \): the final result is not optimised for the final \( \lambda \) but rather takes into account the trajectory of values based on the varying values of \( \lambda \) seen in training; the stress function at a specific \( \lambda \) is only changed during one iteration of descent and the resulting coordinates of \( Y \) are passed to optimisation with the next value of \( \lambda \). In other words, the final mapping is not simply optimised for the final cost function whose \( \lambda \) is the smallest; it is built from a sequence of cost functions with decreasing learning rate \( \lambda \). The stress function is dynamic with changing \( \lambda \) and thus is no longer used as the final quality assessment criterion, but just as a general but inaccurate guide to the best mapping. The result of the algorithm is that the local structure of the manifold is heavily focused on, and meanwhile the global structure is not totally sacrificed.

The absence of a stopping criterion has been rarely mentioned in the previous literature. We will introduce an alternative to the stress function to improve performance of the stochastic algorithm in section 6.1.
4. SOM Toolbox implementation of the optimisation algorithm

To make the concepts in the following sections concrete, we consider a specific implementation of the Exponential CCA algorithm, that given in the SOM Toolbox [27], which is the only publicly available code we have found so far. The SOM Toolbox implementation of CCA takes the updating rule (11), which uses the stress function of the Exponential CCA as (6). The program is summarised below.

1. Generate the learning rate according to

\[ \alpha(t) = \alpha(0) \left( \frac{\alpha(T - 1)}{\alpha(0)} \right)^{\frac{t}{T - 1}}, t = 0, 1, 2, \cdots, T - 1 \]  

(12)

where \( \alpha(0) \) is the initial rate, 0.5 by default; the final learning rate is set to be \( \alpha(T - 1) = \frac{\alpha(0)}{100} \). \( \alpha(t) \) decreases exponentially with the time \( t \).

2. Generate \( \lambda \) sequences by

\[ \lambda(t) = \lambda(0) \left( \frac{\lambda(T - 1)}{\lambda(0)} \right)^{\frac{t}{T - 1}}, t = 0, 1, 2, \cdots, T - 1. \]  

(13)

The default initial neighbourhood radius is the maximum of standard deviation of each dimension of the data set

\[ \lambda(0) = 3 \max(\text{std}([X_{11}, X_{12}, \cdots, X_{1N}]), \text{std}([X_{21}, X_{22}, \cdots, X_{2N}]), \cdots, \text{std}([X_{m1}, X_{m2}, \cdots, X_{mN}])) \]  

(14)

where \( \text{std}() \) stands for the standard deviation; the final neighbourhood radius, \( \lambda(T - 1) \), is fixed at 0.01. \( \lambda(t) \) decreases exponentially with the time \( t \).

3. Create a uniformly distributed sample index sequence, \( S = \{i \in \{1, 2, \cdots, N\}\} \), i.e. the probability of sample \( i \) in the sequence is \( P(i) = \frac{1}{N} \); the size of \( S \) is \( T = \text{epochs} \times N \). An epoch is a period of traversing all data points; the value of epochs is supplied by the program user.

4. Randomly initialise the output space points, \( Y \).

5. For \( t = 1 : T \) (The body of the stochastic gradient descent)

6. Choose the index of the next fixed data point from \( S \);

7. Update \( L_{ij}(t - 1), j \neq i \) by (9);

8. By (11), all the points except point \( i \) are updated according to

\[ Y_j(t) = Y_j(t - 1) - \frac{\alpha(t) L_{ij}(t - 1) - D_{ij}}{L_{ij}(t - 1)} e^{-\lambda(t-\lambda)}(Y_j(t - 1) - Y_i), j \neq i; \]  

(15)

9. End.

We note that the standard methods for stochastic approximation require that the learning rate, \( \alpha \), should satisfy \( \alpha_t \geq 0, \sum_0^\infty \alpha_t = \infty, \sum_0^\infty \alpha_t^2 < \infty \), however the SOM toolbox uses an exponential decay. It is presumably argued that, in any finite simulation, \( \sum_0^\infty \alpha_t = \infty \) cannot be satisfied.

We can see, as we mentioned in previous section, that unlike the traditional gradient descent method that chooses the embedding whose stress is the lowest, the stress function of Exponential CCA is no longer useful as a quality assessment criterion, because the decreasing \( \lambda \) changes the gradient descent surface constantly. An output obtained in one application of the algorithm may not actually be the best because in addition to a possible effect of the choice of initial output (as with traditional gradient descent) we now have a dependency on the sequence of sample indexes chosen. In the next section we will consider another way of selecting the best mapping, in which we use non-stress visualisation quality criteria as a potential alternative to using the stress function.

In the above, apart from the sample index sequence and the initialisation of output, the implementation also used arbitrary choices for \( \alpha(0), \alpha(T - 1), \lambda(0), \lambda(T - 1) \) and epochs (we will refer them as default parameters). In Section 6 we will try different values for these default parameters to create better mappings.
5. Visualisation quality assessment by rank-based criteria

There are many different dimensionality reduction methods. For example, PCA is linear and a closed form solution can be found; MMDs methods use a stress function as an objective function to assess their outputs. We wish to make a quantitative comparison of mappings projected by two different methods. Certainly we can use our eyes, but comparison of configuration quality between different MMDs methods by eye is not an easy task, nor is it easy to objectively quantify this comparison. Within the family of MMDs, the widely used stress comparison is only valid for projections made by the same method; different MMDs methods use different stress functions so the resulting mapping cannot be compared using them. Clearly if we use the stress function of the Sammon mapping for comparison, then we would expect the Sammon mapping to do best since it is designed to optimise this criterion. We need a criterion independent of the mapping algorithm that can be used for a quantitative assessment of the quality of the mapping that the algorithm finds.

Common rank-based criteria for evaluating mappings by different methods have been introduced in the literature, such as trustworthiness and continuity (T&C) [25], mean relative rank errors (MRREs) [12] which is a revised version of T&C, and local continuity meta-criterion (LCMC) [3]. Both T&C and MRREs are rank based criteria; LCMC evaluates configuration quality in terms of neighbourhood preservation. Unifying work of T&C, MRREs and LCMC is found in [13] and [14]. In this section we will review these criteria.

5.1. Definition of rank

Intuitively we wish to state that the rank of point \( j \) with respect to point \( i \) is 5 if there 4 other points which are closer to \( i \) than \( j \). In [13], the rank of point \( j \) with respect to point \( i \) in data space is defined as

\[
R_{\text{data}}(i, j) = |\{ k : D_{ik} < D_{ij} \} \cup \{ k : D_{ik} = D_{ij}, k < j \}|
\]

similarly the rank of point \( j \) with respect to point \( i \) in the low-dimensional space is defined to be

\[
R_{\text{output}}(i, j) = |\{ k : L_{ik} < L_{ij} \} \cup \{ k : L_{ik} = L_{ij}, k < j \}|
\]

where \(| \cdot |\) denotes set cardinality.

Note that rank \( R_{\text{data}}(i, i) = R_{\text{output}}(i, i) = 0 \), and \( R_{\text{data}}(i, j) \neq R_{\text{data}}(i, k) \), even when \( D_{ik} = D_{ij} \) but \( j \neq k \).

5.2. Trustworthiness and Continuity

In [24] two kinds of errors are mentioned during information visualisation. The first is that points originally far away become neighbours, which decreases trustworthiness, because this creates a nonexistent relationship; the second is the opposite, neighbours become far away after projection. This indicates a lack of continuity and the neighbourhood relationship is lost.

Let \( N_{\text{data}}^k(i) \) and \( N_{\text{output}}^k(i) \) stand for the set of the \( k \) nearest neighbours of points \( i \) in data space and the low-dimensional space respectively. \( \text{Intruders}_k(i) \) is the set of data that are in the neighbourhood of size \( k \) of data \( i \) in the low-dimensional space but not in the original space. \( \text{Intruders}_k(i) = N_{\text{output}}^k(i) \setminus N_{\text{data}}^k(i) \). \( \text{Leavers}_k(i) \) is the data that are in the neighbourhood of size \( k \) of data \( i \) in data space but not in the neighbourhood of size \( k \) of data \( i \) in the output space, \( \text{Leavers}_k(i) = N_{\text{data}}^k(i) \setminus N_{\text{output}}^k(i) \).

Trustworthiness, \( T \), is defined as

\[
T(k) = 1 - \frac{2}{\Gamma_{TC}} \sum_{i=1}^{N} \sum_{j\in\text{Intruders}_k(i)} (R_{\text{data}}(i, j) - k) \tag{16}
\]

Continuity, \( C \), is defined as

\[
C(k) = 1 - \frac{2}{\Gamma_{TC}} \sum_{i=1}^{N} \sum_{j\in\text{Leavers}_k(i)} (R_{\text{output}}(i, j) - k) \tag{17}
\]

where the normalisation factor \( \Gamma_{TC} \) is given by \( \Gamma_{TC} = Nk(2N - 3k - 1) \) if \( k < N/2 \); \( \Gamma_{TC} = N(N - k)(N - k - 1) \) if \( k \geq N/2 \).

The higher that \( T \) is, the less intruding data there are, and the better the projection is; the higher \( C \), the fewer neighbours leaving after projection there are, and the lower the loss of continuity is, and the better the projection is.
5.3. Mean relative rank errors (MRREs)

MRREs [12, p214] is based on a similar principle to T&C; it uses two scalars to measure projection quality as T&C does.

The MRRE with respect to the data space (corresponding to Trustworthiness) is defined as

$$
MRRE_{data}(k) = \frac{1}{\Gamma_{MRRE}} \sum_{i=1}^{N} \sum_{j \in N^k_{data}(i)} \left| \frac{R_{output}(i, j) - R_{data}(i, j)}{R_{data}(i, j)} \right|
$$

$$
= \frac{1}{\Gamma_{MRRE}} \sum_{i=1}^{N} \sum_{j \in N^k_{data}(i)} \left| 1 - \frac{R_{output}(i, j)}{R_{data}(i, j)} \right|
$$

(18)

The MRRE with respect to the low-dimensional space (corresponding to Continuity) is defined as

$$
MRRE_{output}(k) = \frac{1}{\Gamma_{MRRE}} \sum_{i=1}^{N} \sum_{j \in N^k_{output}(i)} \left| \frac{R_{data}(i, j) - R_{output}(i, j)}{R_{output}(i, j)} \right|
$$

$$
= \frac{1}{\Gamma_{MRRE}} \sum_{i=1}^{N} \sum_{j \in N^k_{output}(i)} \left| 1 - \frac{R_{data}(i, j)}{R_{output}(i, j)} \right|
$$

(19)

where the normalisation factor $\Gamma_{MRRE} = N^k \sum_{i=1}^{N} \frac{1}{d - N - 1}$. The smaller the MRRE, the better the configuration quality.

MRRE is zero if, ideally, $R_{output}(i, j) = R_{data}(i, j)$ for $i = 1, 2, \cdots, N$. We can see the difference with T&C is that in MRREs the rank error is weighted; and rank changes of the common points in the $k$ nearest neighbourhood in both data and output space, i.e., $N^k_{data}(i) \cap N^k_{output}(i)$, are taken into account.

5.4. Unified framework of T&C and MRREs

Trustworthiness and continuity can be combined into a single measure [13]

$$
Q_{Lee}^{TC}(k) = \frac{T(k) + C(k)}{2}.
$$

(20)

The higher that $Q_{Lee}^{TC}$ is, the better the quality is. We will use $Q_{Lee}^{TC}$ instead of T&C in this paper.

Similarly, the MRREs can be combined into a single measure

$$
Q_{Lee}^{MRREs}(k) = 1 - \frac{MRRE_{data}(k) + MRRE_{output}(k)}{2}
$$

(21)

The higher $Q_{Lee}^{MRREs}$ is, the better the quality is. We will use $Q_{Lee}^{MRREs}$ instead of MRRE_{data} and MRRE_{output} in this paper.

(20) and (21) are just one way to combine the respective two scalar measures giving each of the scalars equal importance. More weight could be given to the one the user prefers to emphasise.

5.5. Local continuity meta-criterion (LCMC)

LCMC is based on the simple idea that the more original neighbour points are kept in the $k$ nearest neighbourhood after projection, the better the embedding is.

There are two versions of LCMC [4]. Let $\Omega(i, k) = N^k_{data}(i) \cap N^k_{output}(i)$ represent the preserved data points in the $k$-ary neighbourhood. The first version of LCMC, the unadjusted LCMC, is as follows

$$
LCMC_{Unadjusted}(k) = \frac{1}{Nk} \sum_{i=1}^{N} |\Omega(i, k)|.
$$

Since the maximum of $|\Omega(i, k)|$ is $k$, the coefficient $\frac{1}{N}$ averages over all points, and $\frac{1}{k}$ normalises the measure to lie between 0 and 1.
Since whatever mapping method is used, $\text{LCMC}_{\text{unadjusted}}(N-1) \equiv 1$, which does not distinguish projection quality between different methods. The second version of LCMC in [3], the adjusted LCMC, was proposed as follows

$$
\text{LCMC}(k) = \frac{1}{NK} \cdot N \sum_{i=1}^{N} \left( |\Omega(i, k)| - \frac{k^2}{N-1} \right) = \frac{1}{NK} \sum_{i=1}^{N} |\Omega(i, k)| - \frac{k}{N-1}.
$$

(22)

The term $\frac{k}{N-1}$ is added as base-line for the LCMC to adjust random overlap which is modelled by a hypergeometric distribution with $k$ defectives out of $N-1$ items and $k$ draws with the expectation $E(\text{LCMC}_{\text{unadjusted}}(k)) = \frac{k^2}{N-1}$. As the result of adjustment, the adjusted LCMC is 0 when $k = N - 1$. Usually when $k$ increases up to $N - 1$, the unadjusted LCMC approaches 1 but the adjusted LCMC decreases to 0. For both the unadjusted and adjusted LCMC, the higher the scalar, the better the projection quality. We will use the adjusted LCMC, (22), in this paper.

5.6. Classification accuracy

Classification accuracy can be used to assess the projection quality of data by classifying points to class labels based on the output space. The classification accuracy is defined as the proportion of correctly classified points. In this paper we will use the benchmark $k$-nearest-neighbourhood classifier [5] to judge the correct classification of an individual point. The algorithm is simple: in the output space, if the class label of data $i$ in question is same as the label of the majority of its $k$-nearest neighbours, then the classification is correct; otherwise wrong. We will use classification accuracy to evaluate mapping quality of two class-labelled data sets at various neighbourhood sizes $k$.

5.7. Neighbourhood relationship preservation

The rank order used in T&C and MRREs only indicates the rank position of the current point to the referenced point. Even if the intermediate points are totally different after mapping, if the rank is the same, it still considers this as the perfect embedding. Let there be three data points $q$, $p$ and $i$ satisfying $D_{ip} < D_{iq}$; $i$, $q$ and $p$ are projected into a low-dimensional space by map A or B as shown in Figure 4. The rank order indicates the relation of the current point $p$ with respect to the referenced point $i$; it does not consider their relations with intermediate point $q$. For example, if the rank of the current point $p$ with respect to the referenced point $i$ in the data space is 6, there are 5 points whose distances are between the current point and the referenced point; suppose that the rank of the current point in the low-dimensional space is still 6, however if the original 5 points are all mapped such that their distances to the referenced point are greater than the distance between the reference point and the current point in the low-dimensional space, the 5 points between the point and referenced point are all new intruders. MRREs will show that the rank error of the point $p$ is 0, which is still to say that the mapping of the point is perfect, but the mappings of the other points (the intruders and the leavers) are not perfect, so the overall mapping is far from perfect in this case. So it is necessary to define a new neighbourhood relationship which takes account of the changing of intermediate points, and a measure called neighbourhood relationship preservation (NRP) (known as LSROP) in [17, 21].

In Figure 4, if after projection, $L_{iq} < L_{ip}$ and the difference between the angle $\angle piq$ in the low-dimensional space and in data space is less than $\pi/4$, then we say a neighbourhood relationship, $O_{i}(q, p)$, of $p$ over $q$ with respect to $i$ is strictly preserved, noted as $\mathcal{P}(O_{i}(q, p)) = 1$; otherwise it is noted as $\mathcal{P}(O_{i}(q, p)) = 0$. 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 $\pi/4$. The angle measure depends on the distances, which makes the measure closer to MMDS rationale than a purely rank-based measure. We can see that the distance $L_{pq}$ in map A is closer to $D_{pq}$ than $L_{pq}$ in map B.

\{$p : R_{\text{data}}(i, p) \leq k$\} represents the data points in the $k$-ary neighbourhood of $i$ in data space. \{$q : L_{iq} < L_{ip}$\} is the data points in the sub-neighbourhood in which the rank of $q$ with respect to $i$ is less than the rank of $p$; $\sum_{q : L_{iq} < L_{ip}} \mathcal{P}(O_{i}(q, p))$ represents the number of neighbourhood relationships (of point $q$ with respect to point $p$) in the sub area. The ratio,

$$
\phi(i, k) = \frac{1}{k} \sum_{\{p : R_{\text{data}}(i, p) \leq k\}} \frac{\sum_{q : L_{iq} < L_{ip}} \mathcal{P}(O_{i}(q, p))}{R_{\text{data}}(i, p)}.
$$

10
represents the average percentage of neighbourhood relationships preserved in the $k$-ary neighbourhood of point $i$. The mean of $\phi(i, k)$ over all data points,

$$NRP(k) = \frac{1}{N} \sum_{i=1}^{N} \phi(i, k), \quad (23)$$

represents the visualisation quality of $k$ neighbourhood. It is always between 0 and 1. The higher $NRP$ is, the better is the result.

Both LCMC and NRP consider what is preserved in the $k$ neighbourhood; obviously NRP is stricter than LCMC.

6. Improvement to the stochastic gradient descent algorithm

The value of rank based visualisation quality measures such as NRP and $Q_{\text{Lee}}^{\text{MRRES}}$ in general agree with both common sense and the values of the other evaluation criteria. Now they can be used to validate visualisation quality of different methods, and they can also be used to differentiate between different mappings during the optimisation process.

In the following, we will extend the stochastic gradient descent optimisation algorithm using visualisation quality measures in two steps: firstly the visualisation quality measures, NRP and $Q_{\text{Lee}}^{\text{MRRES}}$, are used to select the best mappings from mappings created during multiple runs of the old stochastic optimisation process shown in Section 4 which uses default parameters; secondly the parameters are changed greedily so that a further enhancement of the mapping is reached.

6.1. Incorporating quality measures into the optimisation algorithm

In Section 4 we saw that the mapping of the stochastic gradient descent is determined by the random initial output and the order of the random sample index sequence. Since the existing SOM toolbox [27] implementation does not use multiple runs from different initializations and does not provide a ready way to choose between such multiple runs, it may not give the best result. One way of improving is to run the mapping multiple times and use NRP or $Q_{\text{Lee}}^{\text{MRRES}}$ to choose a mapping whose quality measure is the highest; the improved algorithm is shown in Figure 5.

The average NRP is taken over all values of $k = [3, 6, 9, \cdots, 30]$. We chose the relatively small upper bound of 30, because often a simulation with high NRP value for small values of $k$ will have relatively (compared to other simulations) lower values for larger values of $k$. We perform the optimisation iteration a number of times, e.g. 40 runs. In each iteration, the initial output, $Y$, is randomly initialised as well as the sample index sequence; then we accept the mapping whose average of NRP is the highest. In Figure 5 the NRP can be substituted with $Q_{\text{Lee}}^{\text{MRRES}}$.

We now show the effects of the improvement on two standard high dimensional data sets, artificial faces and MNIST digits as shown in Figure 6, instead of the 3-dimensional open box.
1. Initialise $\alpha$ and $\lambda$ according to (12) and (13) correspondingly; set $epochs$.
2. $highestQuality = -1$
3. For $run = 1 : NumberOfRuns$ (e.g., $NumberOfRuns = 40$)
   4. Randomly initialise output, $Y$, and sample index sequence
   5. Perform the body of the stochastic gradient descent described in Section 4
   6. Calculate the average NRP for the mapping created and stored in $currentQuality$
   7. If $currentQuality$ is greater than $highestQuality$
      8. set $highestQuality$ as $currentQuality$
      9. reserve current mapping as the best mapping
10. End
11. End

Figure 5: Using visualisation quality measure to improve stochastic gradient descent algorithm. In each run the output initialisation and random sample index sequence are randomly generated, the parameters $\alpha(0)$, $\alpha(T-1)$, $\lambda(0)$, $\lambda(T-1)$ are set in the SOM Toolbox as presented in Section 4, which we call the default parameters. $Q_{MRRE}$ is an alternative to NRP.

(a) Samples from the artificial faces.  
(b) A subset of MNIST digits.

Figure 6: Other two data sets used in testing.

The artificial faces data set was first used in the original Isomap paper [22] and is available at http://isomap.stanford.edu/datasets.html; the size of the data set is 698 image faces. Each artificial face is a centered gray-scale photo of a statue, consisting of $64 \times 64 = 4096$ pixels as in Figure 6(a). 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 a very high (4096) dimensional space; of course, this manifold need not be a linear manifold. We will map onto a plane rather than a 3 dimensional output space since we wish to emulate the standard position in which we do not know a priori the intrinsic dimensionality of the data space and use MMDS to create a two-dimensional projection which captures the distance relations in the original (high-dimensional) data space.

The MNIST handwritten digits database is available from http://cs.nyu.edu/~roweis/data.html. It consists of 8-bit gray scale images of $28 \times 28$ pixels for digit 0 to 9. For our experiment, we choose a subset of 500 samples in total - 50 samples are randomly drawn from each class (Figure 6(b)). Compared with the faces data set, the digits data are more sparse since it consists of images of different figures and consists of 10 classes. Thus since it is likely that all the ’1’s lie in one region of the $28 \times 28$ dimensional space and similarly all the ’2’s etc., there will be areas of this space which contain no data.

The distance matrix is normalised by dividing by the mean distance. Dividing the distance matrix by a constant does not change the neighbourhood relationships of data points in data space, but has an effect on convergence speed.
and on scaling results; empirically, optimisation of a data set with too large an average distance in data space is slow
to converge and results in a poor embedding.

The best and worst mappings of the artificial faces selected by the average NRP are displayed in Figure 7(a) and
Figure 7(b); those selected by the average $Q_{\text{MRRE}}^{\text{Lee}}$ are depicted in Figure 7(c) and Figure 7(d). We divide the mapping
horizontally into 24 layers such that there is at least one point in each of the top and bottom layers, then vertically
draw as many lines to form a grid. If there is only one point in a grid square then display the photo using the full
space of the square; if there is more than one point in a grid square then display the photo of the point that is closest
to the centre of the grid using the full space of the grid square. As the result, there are at most 24 points to be sampled
vertically. We can see that the automatically selected best mappings are far better than the automatically selected
worst ones in posture and lighting.

The best and worst mappings of the digits selected by the average NRP are displayed in Figure 8(a) and Figure
8(b); selected by the average $Q_{\text{MRRE}}^{\text{Lee}}$ are depicted in Figure 8(c) and Figure 8(d). We can see, again, that the automatically selected best mappings are far better than the automatically selected worst ones in arranging points from the same cluster to be next to each other.

Figure 9 shows the quality of the stochastic optimisation process by the RightExp mapping on the artificial faces and
MNIST digits data sets for 40 runs each from random initialisations, measured by the average NRP and $Q_{\text{MRRE}}^{\text{Lee}}$
simultaneously. We can see that the mappings vary a lot in quality; the average NRP of the best mapping is more than
10 times that of the worst mapping on the digits data set as shown in Figure 9(c); and on the artificial faces data set
as shown in Figure 9(a); on the same two data sets, the contrast between the average $Q_{\text{MRRE}}^{\text{Lee}}$ of the best mapping and
the worst mapping is also considerable as shown in Figures 9(d) and 9(b). The SOM Toolbox [27] implementation
shown in Section 4 would take one of the realisations randomly. In Figure 9(a) we see that the mapping result of one
randomly chosen is about 50% the lowest in quality; in Figure 9(c) we see that the two best mappings with the highest
NRP have a probability of 2 in 40 be chosen randomly.

6.2. The influence of parameters

We have seen the effects of using two quality measures to select the best mappings to reduce the influence of the
randomness of initial output and sample index sequence, above. A question arises, can we do better than the default
setting of parameters $\lambda(0)$, $\lambda(T - 1)$, $\alpha(0)$, $\alpha(T - 1)$, and epochs. We know that each data set is unique: should these parameters be different so that a best mapping is achieved? In the SOM Toolbox [27] implementation, $\lambda(T - 1)$ is given a fixed value; $\alpha(0)$ and $\alpha(T - 1)$ are supplied by program user otherwise default values are used; epochs is set by the user as well. In the following we will investigate the influence of the parameter experimentally. Our aim is not to find the relation between the values of the parameters and the quality of the mapping, but rather to show that different values of parameters lead to different mappings. In order to do so, all the other parameters and the initialisation of the output space, and the order sample index sequence are fixed when only one parameter is varied. To save space, we only show the result when $\lambda(0)$ and $\lambda(T - 1)$ are set differently, and we will only use the NRP to show the influence of the parameters (we have similar results when $Q_{\text{MRRE}}^{\text{Lee}}$ is used).

6.2.1. The parameter $\lambda(0)$

Figures 10(b) and 10(a) show, when all other parameters as well as initialisation and the sample index sequence are
fixed, the influence of $\lambda(0)$ on the mapping quality measured in terms of average NRP on the digits and artificial
faces respectively.

The original optimisation algorithm described in the SOM Toolbox relates $\lambda(0)$ to the maximum standard deviation of
the dimensionalities. For the digits data set, the maximum standard deviation of the dimensionalities is 0.0439, by
(14) $\lambda(0) = 3 \times 0.0439 = 0.1316$; in Figure 10(b) we can see that, with the increase of $\lambda(0)$, the average NRP decreases
then increases with fluctuations. The average NRP around the default value of $\lambda(0)$ is very low. For the artificial faces
data set, the maximum standard deviation is 0.0187; according to (14), $\lambda(0) = 3 \times 0.0187 = 0.056$; in Figure 10(a) we
can see that the average NRP gained using this value is nearly the lowest.

6.2.2. The parameter $\lambda(T - 1)$

Figures 11(b) and 11(a) illustrate when $\lambda(0)$ is set by (14) and all other parameters, and output initialisation and
sample index sequences are fixed, how the $\lambda(T - 1)$ affects the outcome of the mapping on the digits and artificial
(a) The best mapping selected by the average NRP with default parameters and $epochs = 30$.

(b) The worst mapping selected by the average NRP with default parameters and $epochs = 30$.

(c) The best mapping selected by the average $Q_{MRRE}$ with default parameters and $epochs = 30$.

(d) The worst mapping selected by the average $Q_{MRRE}$ with default parameters and $epochs = 30$.

(e) The best mapping selected by the average NRP with optimised parameters.

(f) The best mapping selected by the average $Q_{MRRE}$ with optimised parameters.

Figure 7: Mappings of the artificial faces data set, maximum 24 points are sampled vertically; the default parameters are the parameters $\alpha(0), \alpha(T - 1), \lambda(0), \lambda(T - 1)$ that are set in the SOM Toolbox as presented in Section 4.
(a) The best mapping selected by the average NRP with default parameters, epochs = 30.

(b) The worst mapping selected by the average NRP with default parameters, epochs = 30.

(c) The best mapping selected by the average Q_{MRRE} with default parameters, epochs = 30.

(d) The worst mapping selected by the average Q_{MRRE} with default parameters, epochs = 30.

(e) The best mapping selected by the average NRP with optimised parameters.

(f) The best mapping selected by the average Q_{MRRE} with optimised parameters.

Figure 8: Mappings of the digits data set, maximum 31 points are sampled vertically; the default parameters are the parameters $\alpha(0)$, $\alpha(T - 1)$, $\lambda(0)$, $\lambda(T - 1)$ that are set in the SOM Toolbox as presented in Section 4.
Figure 9: Multiple runs of SOM Toolbox implementation (the algorithm in Figure 5); in each run the output initialisation and random sample index sequence are randomly generated, $epochs = 30$. 

(a) On the artificial faces, measured by NRP.

(b) On the artificial faces, measured by $Q_{MRREs}^{lee}$.

(c) On the digits, measured by NRP.

(d) On the digits, selected by $Q_{MRREs}^{lee}$.

(e) On the algae data, measured by NRP.

(f) On the algae data, selected by $Q_{MRREs}^{lee}$. 

16
faces data sets. We can see that the default value of $\lambda(T - 1) = 0.01$ is not optimal for digits and artificial faces data sets in terms of average NRP.

![Graph](a) On the artificial faces.  
![Graph](b) On the digits.

Figure 10: The influence of parameter $\lambda(0)$ setting on the mapping quality. $\lambda(T - 1) = 0.01$, epochs = 30, $\alpha(0) = 0.5$, $\alpha(T - 1) = \frac{\alpha(0)}{100}$. The initial output and sample index sequence are fixed.

From the above, we see that we cannot agree that the default setting of $\lambda(0)$ and $\lambda(T - 1)$ is the best. Analogously the choice of $\alpha(0)$, $\alpha(T - 1)$, and epochs has a great impact on the outcome of the optimisation and their corresponding default settings seem to be arbitrary. In the following, we create an algorithm which optimises these values in a greedy manner.

6.3. Parameters learning algorithm

We now introduce in Figure 12, a learning algorithm to optimise these parameters in which, when one parameter is being optimised, all other parameters are fixed, including the initial low-dimensional space positions and the sample index sequence (except when epochs changes).
average NRP $\lambda(0) = 0.055972$

(a) On the artificial faces.

average NRP $\lambda(0) = 0.13161$

(b) On the digits.

1. Set initial values of the parameters $\lambda(0) = \max(D(\cdot))$, $\lambda(T - 1) = 0.1$, $\alpha(0) = 0.5$, $\alpha(T - 1) = 0.001$, and $\text{epochs} = 50$. Let $\Delta(x)$ represent step interval of parameter $x$, we initially set $\Delta(\text{epochs}) = 15$, $\Delta(\lambda(0)) = \frac{\lambda(0)}{5}$, $\Delta(\alpha(0)) = \frac{\alpha(0)}{5}$, $\Delta(\alpha(T - 1)) = \frac{\alpha(T - 1)}{5}$.
2. Define variable $\text{highestQuality} = -1$ ($\text{highestQuality}$ is used to select the best mapping whose average NRP is the highest)
3. Define $\text{theHighestQualityAchievedWhenOneParameterVaries}$ as an array (This variable is optional - it is used to select the best mapping among the mappings created when one parameter changes)
4. For $i = 1 : \text{NumberOfRounds}$ (we set $\text{NumberOfRounds} = 8$)
5. The 5 parameters are lined in a random order as $\text{parametersInRandomOrder}$, e.g., $\text{parametersInRandomOrder} = [\lambda(0), \alpha(0), \lambda(T - 1), \text{epochs}, \alpha(T - 1)]$
6. Initialise the output randomly
7. $\text{iteration} = 1$
8. For $j = 1 : 5$ (to optimise the 5 parameters, this loop is noted as $\text{iteration}$)
9. Randomly choose a parameter: $x_{\text{previous}} = \text{parametersInRandomOrder}(j)$
10. $\text{theHighestQualityAchievedWhenOneParameterVaries(\text{iteration})} = -1$
11. For $g = 1 : 9$ (to try some values which is around $x_{\text{previous}}$)
12. Try an adjacent value, $x \leftarrow x_{\text{previous}} + (g - 5)\Delta(x)$
13. Perform the body of stochastic gradient descent as described in Section 4
14. Calculate current average NRP and stored in variable $\text{currentQuality}$
15. If the $\text{currentQuality} > \text{highestQuality}$
16. $\text{highestQuality} = \text{currentQuality}$
17. set $x_{\text{previous}}$ as the $x$ (keep the optimised value of parameter)
18. EndIf
19. If $\text{currentQuality} > \text{theHighestQualityAchievedWhenOneParameterVaries(\text{iteration})}$, $\text{theHighestQualityAchievedWhenOneParameterVaries(\text{iteration})} = \text{currentQuality}$
20. EndIf
21. EndFor
22. $\text{iteration} \leftarrow \text{iteration} + 1$
23. EndFor
24. Reduce the increasing steps: $\Delta(x) \leftarrow 0.618\Delta(x)$

Figure 12: Parameter learning algorithm. $Q^\text{MLRM}_{\text{MRRE}}$ is an alternative to NRP.
Figure 13 shows the parameter learning process on the two data sets. We can see that the quality varies generally with time or the evolution of parameters. For the digits data set, measured by the average NRP as shown in Figure 13(c), the quality increases initially with slight fluctuations, then sharply decreases at iteration 21, then increases quickly again; measured by average Q^{Le}_MRRE, as shown in Figure 13(d), the quality increases with time in general at the beginning, then has a slight tendency to drop off slowly. For the faces data set, measured by the average NRP as shown in Figure 13(a), the quality goes up quickly at the start then struggles to hold for a short time then fluctuates as it decreases; measured by average Q^{Le}_MRRE, as shown in Figure 13(b), the quality fluctuates with time.

We can also see that the extent of change in NRP is greater than the corresponding extent of change in Q^{Le}_MRRE. The optimised parameters for the two data sets are summarised in Table 1. Clearly the optimised parameters are data set dependent. We can see that most λ(0) optimised in the table for the two data sets are quite different from the value set by (14) as the default setting in the SOM Toolbox [27]; nor is the case described in [12, p92]” the user can schedule the value of λ: high values, close to the maximal distance measured in the data, can be used in during the first updates” optimal, since the maximum distances of the digits and artificial faces are 1.4918 and 1.7178 respectively; the default setting, λ(T − 1) = 0.01, in the SOM Toolbox, does not hold either. And the optimised α(0) and α(T − 1) are not related proportionally.

6.4. Comparison of the best mappings achieved by algorithm shown in Figure 5 and Figure 12
6.4.1. on the artificial faces data set

Figures 7(e) and 7(f) are the best mappings of the artificial faces correspondingly selected by the average NRP and average Q^{Le}_MRRE, during the parameter learning process. The average NRP of the mapping selected by the average NRP during the parameter learning process as shown in Figure 7(e) is more than two times that of the average NRP of the best mapping selected by the average NRP during the multiple runs using the default parameters as shown in Figure 7(a); the cluster in the embedding of the faces produced by parameter learning is more dense than using default parameters. In contrast, the average Q^{Le}_MRRE of the best mapping selected by the average Q^{Le}_MRRE during the parameter learning process is only 0.6% higher than the average Q^{Le}_MRRE of the best mapping selected by the average Q^{Le}_MRRE during the multi-runs using the default parameters as shown in Figures 7(f) and 7(c); accordingly it is hard to compare the two mappings by eye; nevertheless LCMC in Figure 15(b), NRP in Figure 14(a), Q^{Le}_TC in Figure 15(d) and Q^{Le}_MRRE in Figure 15(f) still show that the parameter learning is obviously better than the default parameters for small neighbourhoods. LCMC in Figure 15(b), NRP in Figure 14(a) show that the mapping selected by the average NRP during the parameter learning process is much better than the mapping selected by the average Q^{Le}_MRRE during the same process as NRP is considered as an improvement to LCMC; it is of no concern to us that the best mapping selected by the average NRP using the optimised parameters is rated as the lowest in quality by both Q^{Le}_TC and Q^{Le}_MRRE in Figures 15(d) and 15(f) since both Q^{Le}_TC and Q^{Le}_MRRE favour the mappings selected by the average Q^{Le}_MRRE as MRREs is a variant of T&C and Q^{Le}_MRRE can be considered a variant of Q^{Le}_TC.

The quality measures for the worst mappings using default parameters as shown in Figures 7(b) and 7(d) are not plotted because their quality is obviously low.

NRP in Figure 14(a) and LCMC in Figure 15(b) also show that the quality of mapping selected by both measures using optimised parameters is much higher than using default parameters. Both Q^{Le}_TC and Q^{Le}_MRRE only show that the best mapping selected by the average Q^{Le}_MRRE during the parameter learning is better than the mapping selected by the average Q^{Le}_MRRE, using default parameters.

NRP in Figure 14(a) shows that, if visualisation quality is high in small neighbourhoods, it is low in large neighbourhoods. It is generally an agreed conclusion in the research community that if a dimensionality reduction method pays more attention to the global structure of a data set, it focuses less on local structure; and vice versa. Q^{Le}_TC in Figure 15(c) and Q^{Le}_MRRE in Figure 15(e) do not reveal this fact.

6.4.2. on the digits data set

The Figures 8(e) and 8(f) reveal the best mappings of the digits data set selected by the average NRP and the average Q^{Le}_MRRE, respectively during the parameter learning process. Comparing Figure 8(e) with the best projection created using the default parameters as shown in Figures 8(a), and comparing Figures 8(f) with with the best projection created using the default parameter as shown in Figure 8(c), we find that, in the mapping created during the parameter learning process, digits such as '6' and '2' are less likely to be mixed with other digits.
Figure 13: The parameter learning process. One parameter is tried 8 adjacent values in each iteration; the vertical axis is the variable, theHighestQualityAchievedWhenOneParameterVaries(iteration), in Figure 12.
As shown in Figures 16 and 17 by all quality measures, for small neighbourhoods, the best mappings created by using default parameters selected by both average NRP and $Q_{\text{MRRE}}^{\text{Lee}}$ are close to each other. For the parameter learning process, in NRP and LCMC, the best mapping is the one selected by the average NRP; but in $Q_{\text{TC}}^{\text{Lee}}$ and $Q_{\text{MRRE}}^{\text{Lee}}$, the best mapping is the one selected by the average $Q_{\text{MRRE}}^{\text{Lee}}$; in either case, the quality of the mapping obtained using the optimised parameters is much higher than the quality of the mapping created using default parameters.

The measures for the worst mappings using default parameters as shown in Figures 8(b) and 8(d) are not plotted because their quality is obviously low, as judged by eye.

Again, one mapping which is high in NRP or LCMC in small neighbourhoods is low in large neighbourhoods; $Q_{\text{MRRE}}^{\text{Lee}}$ do not agree with this at all; $Q_{\text{TC}}^{\text{Lee}}$ can show the tendency but it is not as obvious as with NRP or LCMC.

From the above we see that NRP and LCMC favour the mappings selected by the average NRP and so do $Q_{\text{TC}}^{\text{Lee}}$ and $Q_{\text{MRRE}}^{\text{Lee}}$, with the mappings selected by the average $Q_{\text{MRRE}}^{\text{Lee}}$. Now we use classification accuracy as an independent judge to give an objective assessment. Figures 16(c) and 16(d) are the quality assessment in classification accuracy at various neighbourhood sizes. We can see that the parameter learning method with selection criterion of average NRP is the best for small neighbourhoods, and it is much better than selection by $Q_{\text{MRRE}}^{\text{Lee}}$.

### 6.4.3. on the algae data set

We now examine another data set well known in the literature, the algae data set. This is a set of 118 samples from a scientific study of various forms of algae, some of which have been manually identified. Each sample is recorded as an 18 dimensional vector representing the magnitudes of various pigments. 72 samples have been identified as belonging to specific classes of algae which are labelled from 1 to 9. 46 samples have yet to be classified and these are labelled 0. Before the test, a normalisation is performed. Firstly, each dimension is centered by subtracting its mean; then the standard deviation of each dimension is transformed to 1 by dividing each sample by the standard deviation; finally the mean distance between data points is transformed to 1.

Using default parameters and selecting the mapping by the average NRP, Figure 18(a) shows the worst mapping and Figure 18(b) illustrates the best result. In Figure 18(a) we see that 8, 4 and 3 are dispersed in three locations; 7, 9, 5 are in two locations; 2s are mixed with one cluster of 3. In Figure 18(b), 7 and 9 are in two places; 6 is scattered across the plot. Clusters are obvious in Figure 18(c) which use the optimised parameters; 1s are separated by a strip formed by 0s and 5s; one 6 is far from other 6s.

Using default parameters and selecting the mapping by the average $Q_{\text{MRRE}}^{\text{Lee}}$, Figure 19(a) shows the worst mapping and Figure 19(b) illustrate the best result. In Figure 19(a) we can see that that 1s are well separated by 0s into two groups; 2s and 3s are also separated. There are improvements in clustering in Figure 19(b), it shows that one 7 is on the left of the map, two 7s are on the right part of the map, other three 7s are on the left-top location. One 9 is on the
Figure 15: Quality assessment of the projections of the artificial faces data set by LCMC, $Q^L_{TC}$ and $Q^L_{MRRE}$ (paras stands for parameters).
Table 1: Optimised parameters using average NRP.

<table>
<thead>
<tr>
<th></th>
<th>$\lambda(0)$</th>
<th>$\lambda(T-1)$</th>
<th>$\alpha(0)$</th>
<th>$\alpha(T-1)$</th>
<th>epochs</th>
<th>$Q$</th>
<th>$Q^\text{err}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>artificial faces</td>
<td>1.2124</td>
<td>0.1171</td>
<td>0.9764</td>
<td>0.0006</td>
<td>142</td>
<td>0.2359</td>
<td>0.9711</td>
</tr>
<tr>
<td>digits</td>
<td>1.4123</td>
<td>0.1247</td>
<td>0.5236</td>
<td>0.0021</td>
<td>168</td>
<td>0.0969</td>
<td>0.9236</td>
</tr>
</tbody>
</table>

Figure 16: Quality assessment of the projections of the digits data set in NRP and classification accuracy (paras stands for parameters).
Figure 17: Quality assessment of the projections of the digits data set by LCMC, $Q_{\text{TC}}^{\text{Lee}}$ and $Q_{\text{MRRE}}^{\text{Lee}}$ (paras stands for parameters).
(a) The worst mapping with default parameters and \(epochs = 30\).

(b) The best mapping with default parameters and \(epochs = 30\).

(c) The best mapping with optimised parameters.

Figure 18: Mappings of the algae data set selected by the average NRP.
Figure 19: Mappings of the algae data set selected by the average $Q_{\text{MRRE}}^{\text{Lee}}$. 

(a) The worst mapping with default parameters and $epochs = 30$. 

(b) The best mapping with default parameters and $epochs = 30$. 

(c) The best mapping with optimised parameters.
right, other two are on the top. One 4 is on the left, all other 4s are on the top-right. In Figure 19(c) with optimised parameters the digits are much better grouped; 1s are still separated by a strip formed by 0s and 5s and one 6 is far from other 6s, which is similar to Figure 18(c).

Figures 20 and 21 assess the quality using various measures; for the classification accuracy, we only take account of the 72 labelled samples. We see that the best mappings selected by the two quality criteria, average NRP and $Q_{Lee}^{MRRE}$, using the same kind of parameters setting are close to each other. The contrast in quality is great using different setting of parameters: the parameter learning method is superior to using the default parameters and the variances in quality are big for using the default parameters.

6.4.4. Summary

In the above we showed that, on the MNIST digits that measured in NRP, LCMC, $Q_{Lee}^{MRRE}$ and $Q_{Lee}^{TC}$ and selected by both average NRP and $Q_{Lee}^{MRRE}$, the parameter learning algorithm shown in Figure 12 further improves mapping quality on the basis of the algorithm of selecting a better mapping from the multiple runs shown in Figure 5. On the artificial faces, if measured in NRP and LCMC, the conclusion holds; if measured in $Q_{Lee}^{MRRE}$ and $Q_{Lee}^{TC}$, it is still true when selected by average $Q_{Lee}^{MRRE}$, but selected by average NRP the parameter learning method is worse than the algorithm using the default parameters. Due to the curse of dimensionality [1], in reality an observed data set is often sparse like the MNIST digits data set; so we conclude that the parameter learning method works on typical real data sets. This is also true of the sparse algae data set.

7. Comparison with other methods

Figures 23(b), 23(c), 23(d) and 23(c) show the mapping of algae data set by Sammon, LeftSammon, RightSammon and LeftExp mappings respectively. Figures 23(e) and 23(f) assess the mapping quality in classification accuracy. It shows that the mappings produced by all methods are similar in quality except that the RightExp using default parameters creates lower quality mapping.

From above we have seen that the incorporation of non-stress quality measure into the optimisation process has greatly improved the data visualisation. We now compare the improved algorithm (parameter learning) with the Sammon [16], LeftSammon [21], RightSammon [21] mappings, and the LeftExp mapping (known as Exp in [19]) on the digits and algae data sets based on classification accuracy.

Figures 22(b), 22(c), 22(d) and 22(c) show the mapping of the digits by Sammon, LeftSammon, RightSammon and LeftExp mappings respectively. Figures 22(e) and 22(f) assess the mapping quality in classification accuracy. In Figure 22(f) we see that the improved algorithm is the best for small neighbourhoods, the second best is the RightSammon. The best mapping selected by the average NRP using default parameters is slightly inferior to the RightSammon mapping. The LeftSammon mapping produces better visualisation than the LeftExp mapping.

8. Conclusions and future work

In this paper we have integrated visualisation quality measures into the existing stochastic gradient descent algorithm for CCA; we have also shown that the SOM toolbox implementation of this method is very dependent on the specific parameters chosen during learning and have shown that a greedy algorithm can find the optimal parameters based on optimising any of a number of different measures of goodness of the mappings. The improved algorithm was tested on Exponential CCA; we will show in the future that it is also applicable to Step CCA.

In this paper we have focused on the improvements of the optimisation method of CCA itself; the effects of improvement are judged by eye as well as by quality measures; theoretical and empirical comparison of CCA with other methods such as LLE [15], Isomap [22], the Sammon mapping [16] and its extensions using Bregman divergence [21], and Neighbourhood Retrieval Visualiser(NeRV) [26] remains as future work, part of which has been done in [20].

The code used in this paper is freely available from http://cis.uws.ac.uk/research/JigangSun/index.html.
Figure 20: Quality assessment of the projections of the algae data set in NRP and classification accuracy (paras stands for parameters).
Figure 21: Quality assessment of the projections of the algae data set by LCMC, $Q_{TC}^{Le}$, and $Q_{MRRE}^{Le}$ (paras stands for parameters).
Figure 22: Quality assessment of the projections of the digits data set in classification accuracy.
Figure 23: Quality assessment of the projections of the algae data set in classification accuracy.
References


Jigang Sun received his Bachelor in Management Engineering from the University of Beijing Aeronautics and Astronautics (BUAA), Beijing, in 1988; and received his MSc in Information Technology from the University of the West of Scotland (UWS), Paisley, in 2008; successfully completed his Ph.D. at UWS in July 2011. He has been focused on using Bregman divergences to improve metric multidimensional scaling, non-stress function scaling quality assessment criteria, and recently neural networks. His research work is ongoing while seeking his next funded project.

Colin Fyfe is a professor in computational intelligence with over 300 refereed publications. He has supervised 22 Ph.D. students and is on the editorial boards of 5 international journals. His current research interests are unsupervised machine learning.

Malcolm Crowe is a Professor in the School of Computing at the University of the West of Scotland since 1985. He still finds time to do some research.