Sammon Mapping

Paul Henderson

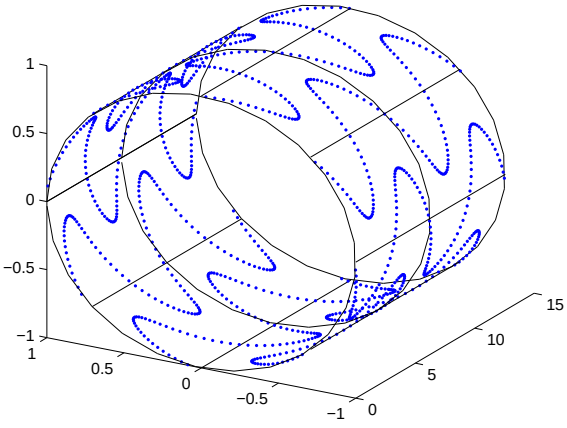
1 Motivation

It is often necessary to reduce the dimensionality of a dataset, in order to make analysis computationally tractable, or to facilitate visualisation. For the purposes of computer vision, we are most often interested in reducing the dimensionality of a large set of real-valued vectors (representing points in some high-dimensional space), and in the course of such reduction, it is useful to *preserve structure* as much as possible — that is, the geometric relations between the original data points should be left intact to the greatest extent feasible.

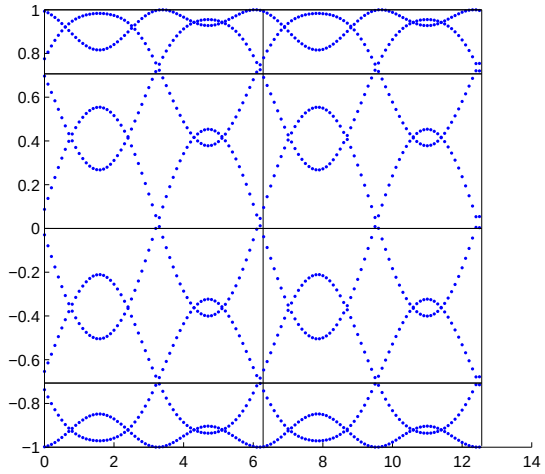
The simplest technique for dimensionality reduction is a straightforward *linear projection*, for example onto the principal components of the data (as in PCA — principal component analysis). While this maximises the amount of the original variance present in the transformed dataset, it will not (in general) preserve 'complex' structures [1] — an example of such a structure being a regular pattern over a curved manifold embedded in the high-dimensional space, see Figure 1 for an example. *Non-linear* projections may therefore be desirable when analysing such data.

This leaves the question of *what* non-linear transformation is optimal for some given dataset. While PCA simply maximises variance, we now want instead to maximise some other measure, that represents the degree to which complex structure is preserved by the transformation. Various such measures exist, and one of these defines the so-called *Sammon Mapping*, named after John Sammon, Jr, who initially proposed it in [2].

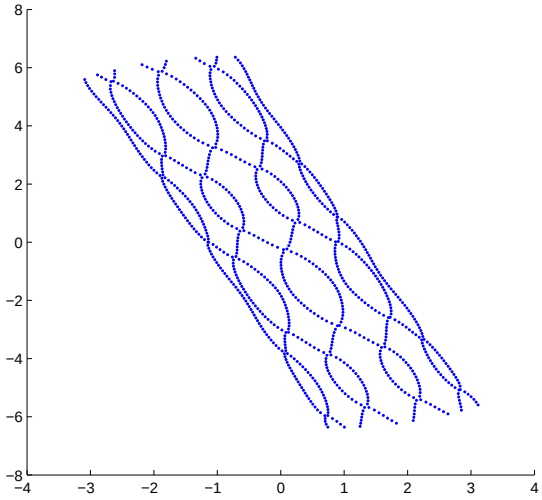
More specifically, the measure used by the Sammon mapping is designed to minimise the differences between corresponding inter-point distances in the two spaces — a transformation is regarded as preferable if it conserves (to the greatest extent possible) the distance between each pair of points. In addition, it attempts to ensure that the mapping does not affect the topology —



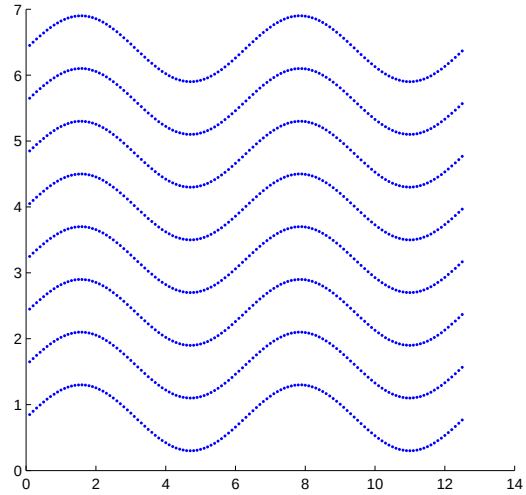
(a) An example three-dimensional dataset, consisting of eight sinusoids on the surface of a cylinder — thus the entire dataset lies on a 2-manifold, and we should be able to obtain a two-dimensional projection which preserves the structural details



(b) Result of applying PCA (a linear projection) to the dataset in (a) — note that the individual sinusoids intersect, thus the topology of the original has not been preserved

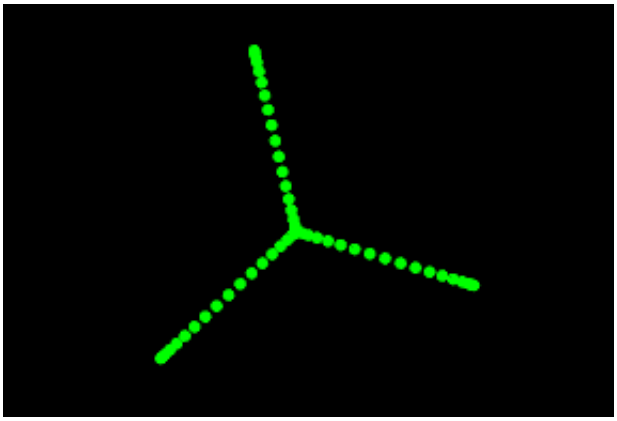


(c) Result of applying the Sammon mapping to the dataset in (a) — this yields a somewhat different result to PCA; there are fewer intersections between sinusoids, though still some (indicating that the topology is being preserved somewhat better — though far from perfectly — by this mapping)

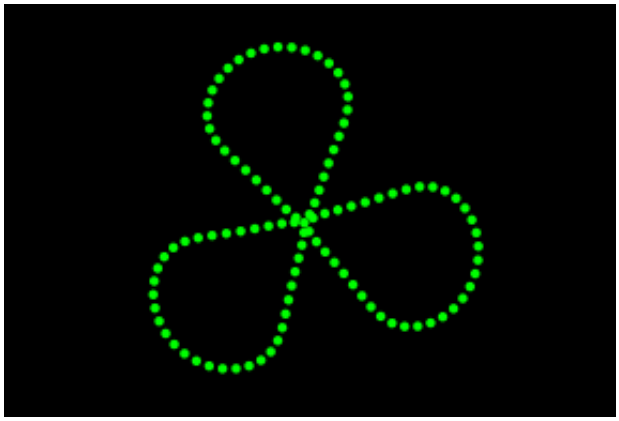


(d) An idealised non-linear projection of the data, with the cylinder 'unrolled', reveals the true structure most clearly. However, the Sammon mapping is unlikely to achieve this particular form, as the topology is not preserved along the cut corresponding to the top and bottom edges of this plot

Figure 1: Various projections of a three-dimensional dataset, where all the points lie on a cylindrical surface



(a) Projection by PCA does not preserve the structure of the dataset — it is unclear that it consists of three circles



(b) The Sammon mapping preserves the topological structure — while the circles become distorted, there are still three closed loops meeting at a single point

Figure 2: PCA and Sammon projections of a six-dimensional 'bouquet of circles', from [3]. The original dataset contains three mutually perpendicular circles in six-dimensional space, meeting at a point

i.e. the importance of preserving relations between *nearby* points is emphasised [3]. Figure 2 illustrates how the mapping can reduce a six-dimensional dataset to a two-dimensional one, while preserving the topological structure.

2 Details

Unlike traditional linear dimensionality reduction techniques (such as PCA), the Sammon mapping does not explicitly represent the transformation function. Instead, it simply provides a measure of how well the *result* of a transformation (i.e. some lower-dimensional dataset having the same number of points as the original) reflects the structure present in the original dataset, in the sense described above. In other words, we are attempting not to find an optimal mapping to apply to the original data, but rather to construct a *new* lower-dimensional dataset, which has structure as similar to the first dataset as possible.

Following the paper [2] introducing the mapping, we will represent the original dataset as N vectors in L-dimensional space, given by $X_i, i = 1, \ldots, N$. We seek to map these into d-dimensional space (with $d < L$), to give vectors $Y_i, i = 1, \ldots, N$. For simplicity, write d_{ij} for the pairwise distance between Y_i and Y_j, and similarly d^*_{ij} for the distance between X_i and X_j (Sammon

assumes the metric here to be Euclidean, though this is not strictly necessary).

The amount of structure present in the original but lost in the transformed dataset is then measured by an error E, defined as

$$E = \frac{1}{\sum_{i<j} d^*_{ij}} \sum_{i<j}^{n} \frac{(d^*_{ij} - d_{ij})^2}{d^*_{ij}}$$

Essentially, the error is given by summing up the squared differences (before versus after transformation) in pairwise distances between points; the summations are over the range $i < j$ so that each pairwise distance is counted once (and not a second time with i and j swapped). The tendency to preserve topology (mentioned above) is due to the factor of d^*_{ij} in the denominator of the main summation, ensuring that if the original distance between two points is small, then the weighting given to their squared difference is greater.

3 Implementation

The error E provides us with a measure of the quality of any given transformed dataset. However, we still need to determine the optimal such dataset, in terms of minimising E. Strictly speaking, this is an implementation detail and the Sammon mapping itself is simply defined as the optimal transformation; however, in the original paper [2], Sammon describes one method for performing the optimisation. The transformed dataset Y_i is first initialised by performing PCA on the original data (an arbitrary, random initialisation is sufficient, but using the principal components improves performance somewhat — see below). Then, we repeatedly update the Y_i using steepest descent, considering the gradient of E with respect to the Y_i, until satisfactory convergence is achieved.

This optimisation problem has rather high dimensionality (proportional to the number of data points), and hence other more modern techniques, such as simulated annealing, could beneficially be applied in order to achieve better convergence properties and avoid local minima.

More recently, neural network implementations of the Sammon Mapping have been proposed — see, for example, [4] and [5]. This approach has the advantage of being able to generalise — given a new vector in the high-dimensional space, the network is able to map it to a lower-dimensional vector in a fashion consistent with the rest of the dataset; this is in contrast to the basic

form of the mapping, where there is no mechanism for dealing with new data aside from recomputing the entire transformation. Also, in this context, [6] discusses the merits of random versus PCA-based initialisation (see above); it seems that using a small number of principal components to perform the initialisation shortens the training and decreases the final error achieved.

References

[1] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Prentice Hall, 1988.

[2] J. W. Sammon, Jr, "A nonlinear mapping for data structure analysis," *IEEE Transactions on Computers*, vol. C-18, no. 5, pp. 401–409, 1969.

[3] S. Hotton and J. Yoshimi, "About HiSee – Projection Algorithms." `http://hisee.sourceforge.net/about.html`.

[4] J. Mao and A. K. Jain, "Artificial neural networks for feature extraction and multivariate data projection," *IEEE Transactions on Neural Networks*, vol. 6, no. 2, pp. 296–317, 1995.

[5] S. Ivanikovas, V. Medvedev, and G. Dzemyda, "Parallel realizations of the SAMANN algorithm," in *Adaptive and Natural Computing Algorithms*, pp. 179–188, Springer Berlin / Heidelberg, 2007.

[6] B. Lerner, H. Guterman, M. Aladjem, and I. Dinstein, "On the initialisation of Sammon's nonlinear mapping," *Pattern Analysis and Applications*, vol. 3, pp. 61–68, 2000.