Tangent-Corrected Embedding

Ali Ghodsi, Jiayuan Huang
University of Waterloo

Finnegan Southey, Dale Schuurmans
University of Alberta

Abstract

Images and other high-dimensional data can frequently be characterized by a low dimensional manifold (e.g. one that corresponds to the degrees of freedom of the camera). Recently, nonlinear manifold learning techniques have been used to map images to points in a lower dimension space, capturing some of the dynamics of the camera or the subjects. In general, these methods do not take advantage of any prior understanding of the dynamics we might have, relying instead on local Euclidean distances that can be misleading in image space.

In practice, we frequently have some prior knowledge regarding the transformations that relate images (e.g. rotation, translation, etc). We present a method for augmenting existing embedding techniques with additional information derived from known transformations, either in the form of tangent vectors that locally characterize the manifold or distances derived from reconstruction errors. The extra information is incorporated directly into the cost function of the embedding technique. The techniques we augment are largely attractive because there is a closed form solution for their cost optimization. Our approach likewise produces a closed form solution for the augmented cost function. Experiments demonstrate the effectiveness of the approach on a variety of image data.

1 Introduction

Automatically finding low-dimensional manifolds that characterize data observed in high-dimensional space has been a subject of inquiry for several decades. With the abundance of complex data like video and images available today, such methods are more relevant than ever. Traditional techniques such as *principal component analysis* (PCA) [1] and *multidimensional scaling* (MDS) [2] embed data points in a linear subspace of the original space while attempting to preserve relationships amongst the original points (e.g. reconstruction error or pairwise distance). Recently, the basic notions underlying these techniques have been extended to discover nonlinear manifolds that cannot be captured by the simpler methods. Recent nonlinear manifold methods include *kernel PCA* [3], *locally linear embedding* (LLE) [4], Isomap [5], the Laplacian Eigenmap method (LEM) [6], and Semi-definite Embedding [7].

While differing in the relationships they preserve and the exact mechanisms used to preserve relationships, these methods all use Euclidean distance between points as the basis for the embedding. However, if we consider data such as the pixels of grayscale images (a vector of $width \times height$ pixels), even a relatively small and simple transformation (e.g. rotation) of a three-dimensional object can result in a deceptively dramatic change to its image. Between two such consecutive images, the Euclidean distance can be quite large, potentially leading to the false conclusion that the two images are only weakly related.

Image data often has an underlying invariant and associated transformations, like rotation, that naturally imply a manifold on which neighbouring points are small transformations of one another. We can often characterize these transformations based on prior knowledge regarding the source of the images (e.g. video data is likely to contain shifts, rotations, changes of illumination, etc). We propose a method for exploiting the extra information offered by the transformations to correct potentially misleading observations based on Euclidean distance.

Other research has sought to use prior information regarding transformations, particularly with image data, in the context of clustering using probabilistic models that directly incorporate transformations [8], augmented distance measures in a supervised learning context [9], and for tracking [10]. Drawing from this body of research, we extend nonlinear embedding techniques to take advantage of known transformations, allowing for unsupervised learning of embeddings that better reflect the underlying dynamics.

We will start by explaining two methods for constructing distances based on known transformations. The first considers angles between tangent spaces induced by the transformation, and the second uses sequences of transformations that greedily minimize reconstruction error. Next, we briefly review some contemporary embedding techniques and follow this by deriving a new embedding cost function to combine different distance information. Aside from its immediate application as a correction for image embedding, this cost function is a general purpose framework for combining embedding techniques derived from MDS and PCA. We present experiments demonstrating that the augmentation gives embeddings that capture more features of the image data, including clustering together images related to each other by a transformation.

2 Tangent Correction Distance

Consider some high-dimensional data that lives on a lower-dimensional manifold. If the data lie densely along the manifold (e.g. very small steps of rotation) the Euclidean distances between consecutive points may be small enough to capture the manifold (see Figure 1 (a)). If, however, the data are sparsely distributed (Figure 1 (b)), the distances may too large to be informative. Intuitively, one would wish to fill in the gaps along the manifold by generating new points likely to lie on the manifold. It is here that our known transformations come into play.

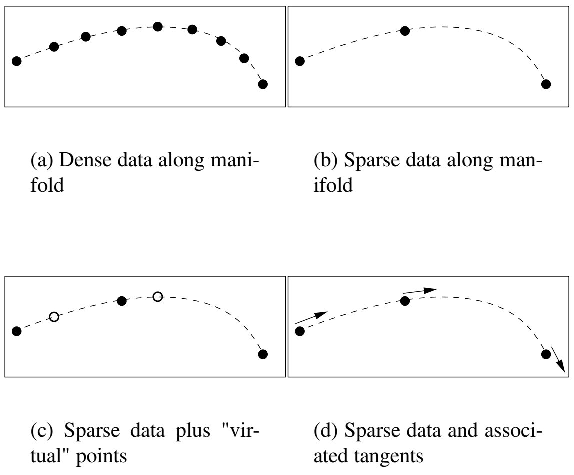
(a) Dense data along manifold
(b) Sparse data along manifold
(c) Sparse data plus "virtual" points
(d) Sparse data and associated tangents

Figure 1. Data and tangents along the manifold

One strategy is to generate "virtual" points by applying transformations to real data points and then learn the manifold using both the real and the virtual points (Figure 1 (c)). In this research, we instead adopt a strategy similar to that used by Chapelle and Schölkopf for using known transformations in support vector machines [11]. Rather than explicitly creating points and adding them, we use the transformations to obtain tangent vectors at the real data points. The underlying assumption here is that the transformation locally characterizes the manifold, so that tangents to the transformation function approximate tangents to the manifold (Figure 1 (d)).

More formally, if we have a set of vectors in the original high dimensional space, $\mathbf{X}$, and a transformation $\mathcal{T}(\mathbf{x}_i, \theta)$, parameterized by a scalar θ, we obtain a tangent vector $d\mathbf{x}_i$ for each point according to $d\mathbf{x}_i = \lim_{\theta \to 0} \frac{1}{\theta}(\mathcal{T}(\mathbf{x}_i, \theta) - \mathcal{T}(\mathbf{x}_i, 0))$ or, alternatively, $d\mathbf{x}_i = \frac{\partial}{\partial \theta} |_{\theta=0} \mathcal{T}(\mathbf{x}_i, \theta)$.

The intuition behind our treatment of these tangent spaces is as follows. If the tangents of two points $\mathbf{x}_i$ and $\mathbf{x}_j$ are unaligned (as in Figure 2 (b)) rather than aligned (as in Figure 2 (a)), the Euclidean distance is probably inaccurate. While there is no guarantee that the Euclidean distance between aligned points is accurate for points that are very distant along the manifold (there may be many curves along the way), over smaller distances the alignment provides some good evidence of the reliability of Euclidean distance. We therefore use the angle between tangent spaces as a correction to the Euclidean distance in our embedding cost functions. A simple way to do this is to compute the sine of the angle between each pair of tangent spaces and treat this as a new measure of distance. Clearly sine gives us a number between 0 and 1 for each pair, where 0 means the pair is parallel (aligned) and 1 means the pair is orthogonal (unaligned). More formally, we construct a matrix, $D^{(\mathcal{T})}$, containing the sine squared of the angle between each pair of tangent vectors.

$$D^{(\mathcal{T})} = 1 - \frac{\langle d\mathbf{x}_i, d\mathbf{x}_j \rangle^2}{\langle d\mathbf{x}_i, d\mathbf{x}_i \rangle \langle d\mathbf{x}_j, d\mathbf{x}_j \rangle}$$

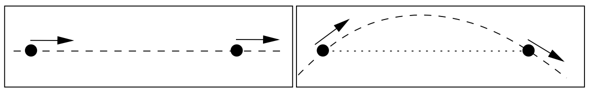
(a) Aligned
(b) Unaligned (dotted line is Euclidean distance)

Figure 2. Alignment of tangent vectors

3 Transformation Reconstruction Distance

An alternative way to use transformations in image comparison is to apply a transformation to an image i and then measure the distance of this transformed image to another image j. We call this distance the *reconstruction error,* representing how well image i can be used to reconstruct the target image j. If the target image j is truly a transformed version of the original, then the corresponding reconstruction error should be very small.

In general, if images on the manifold were the result of repeated applications of the transformation, it should be possible find a sequence of images in the data set with very small reconstruction error at each step in the sequence. Finding such a sequence can be seen a shortest path problem on a graph weighted by the reconstruction errors. This is similar to the approach used by Isomap to obtain so-called geodesic distances [5]. We call the sum of the reconstruction errors along the path the *transformation reconstruction distance*.

Both of these corrections (tangent correction and transformation reconstruction) can be combined with any embedding method that has a cost function similar in form to locally linear embedding (LLE) and the Laplacian Eigenmap method (LEM). We will first provide a brief background on the relevant embedding methods and then derive the combined cost function.

4 Embedding Methods

Before presenting tangent corrected embedding, it is necessary to quickly review some background in dimensionality reduction techniques, which we will later combine to form a new embedding cost function. In general, these methods take t points $x_1, \dots, x_t$ in high-dimensional space and map them to points $y_1, \dots, y_t$ in some lower dimensional space. These lower dimensional points are called *codes*. Broadly speaking, the embedding process attempts to preserve some structure inherent in the original set of points by imposing similar structure on the codes.

4.1 Multidimensional Scaling

Multidimensional scaling (MDS) finds a mapping from high-dimensional space to a lower-dimensional space while preserving pairwise distances between points.

Given a $t \times t$ distance matrix D, MDS finds *code vectors* $y_1, \dots, y_t$ in d dimensions such that if d_{ij} denotes the Euclidean distance between x_i and x_j, and $\hat{d}_{ij}$ is the distance between y_i and y_j, then d_{ij} is similar to $\hat{d}_{ij}$ for all pairs of points. We will specifically consider *metric MDS* [2], which tries to minimize,

$$\min_Y \sum_{i=1}^{t} \sum_{j=1}^{t} (d_{ij}^2 - \hat{d}_{ij}^2)^2 \qquad (1)$$

where $d_{ij} = ||x_i - x_j||$ and $\hat{d}_{ij} = ||y_i - y_j||$. One way to achieve this is to convert the distance matrix D to inner products, $X^T X = -\frac{1}{2} H D H^T$, where $H = I - \frac{1}{t} e e^T$ and e is a column vector of all 1's. We can now rewrite the problem as

$$\min_Y \sum_{i=1}^{t} \sum_{j=1}^{t} (x_i^T x_j - y_i^T y_j)^2 \qquad (2)$$

It can be shown [2] that the closed form solution to this minimization problem is $Y = \Lambda^{1/2} V^T$ where Λ is a diagonal matrix containing the top d eigenvalues of $X^T X$, and V contains the corresponding eigenvectors.

4.2 Laplacian Eigenmap Method

Given t points in high-dimensional space, LEM [6] starts by constructing a weighted graph with t nodes and a set of edges connecting neighbouring points. Each edge is weighted by W_{ij}. The embedding map is then provided by the following objective

$$\min_Y \sum_{i=1}^{t} \sum_{j=1}^{t} (\mathbf{y}_i - \mathbf{y}_j)^2 W_{ij}$$

subject to appropriate constraints. This objective can be reformulated as

$$\min_Y \operatorname{Tr}(Y^T Y L)$$

where $L = R - W$, R is diagonal, and $R_{ii} = \sum_{j=1}^{t} W_{ij}$. This L is called the *Laplacian function*. Happily, principle components analysis (PCA) gives us a closed form solution for this minimization problem (i.e. make Y the eigenvectors of L).

4.3 Locally Linear Embedding

Locally linear embedding (LLE) [4] builds a weight matrix by attempting to reconstruct each point using a linear combination of its k nearest neighbours

$$\min_W \sum_{i=1}^{t} ||\mathbf{x}_i - \sum_{j=1}^{k} W_{ij} \mathbf{x}_{N_i(j)}||^2$$

where $N_i(j)$ is the index of the jth neighbour of the ith point. It then selects code vectors so as to preserve the reconstruction weights by solving

$$\min_Y \sum_{i=1}^{t} ||\mathbf{y}_i - \sum_{j=1}^{k} W_{ij} \mathbf{y}_{N_i(j)}||^2$$

This objective can be reformulated as

$$\min_Y \operatorname{Tr}(Y^T Y L)$$

where $L = (I-W)^T(I-W)$. Note that the final objectives for both LEM and LLE have the same form and differ only in how the matrix L is constructed. Therefore, same closed form solution (taking Y to be the eigenvectors of L) works.

It is worth mentioning here that the approach we explain in the next section can be applied to augment any technique that has an objective function of this form (essentially forms of nonlinear PCA, e.g. kernel PCA [3]).

5 Combined Embedding

We will now present a new cost function that can combine two different similarity measures into a single cost function. The first similarity measure is derived from a distance matrix, $D^{(\mathcal{T})}$, which in our case could be one of the two non-Euclidean distances presented above (i.e. tangent correction distance or transformation reconstruction distance). We then use the MDS transformation from Section 4.1 to obtain $M^{(\mathcal{T})} = -\frac{1}{2}HD^{(\mathcal{T})}H$. The second similarity measure is expressed in a matrix, L, derived from a locality-preserving embedding method such as LLE (for which $L = (I-W)^T(I-W)$, where W is the matrix of reconstruction weights) or LEM (in which case L is the Laplacian). These PCA-derived methods attempt to minimize local reconstruction error. Our derivation applies to both LLE and LEM.

We can now form our combined embedding objective

$$\min_Y (1-\alpha)\mathrm{Tr}(M^{(\mathcal{T})} - Y^TY)^2 + \alpha\mathrm{Tr}(Y^TYL) \qquad (3)$$

where Y is a matrix of code vectors and $0 \le \alpha < 1$.[1] The first trace term in this objective is essentially the MDS objective, trying to preserve the transformation-derived distances. The second trace term is the cost function of a locality-preserving embedding method. The parameter α mixes between the objectives, embedding on the basis of the transformation-based distances as α tends to zero, and on the locality-preserving objective as α tends to 1.

To solve this problem, we start by applying singular value decomposition (SVD) to obtain $M^{(\mathcal{T})} = VPV^T$, where V and P are the eigenvectors/values of $M^{(\mathcal{T})}$. If we also decompose $Y^TY = Q\Lambda Q^T$ (Q and Λ are eigenvectors/values), then $Y = \Lambda^{\frac{1}{2}}Q^T$, allowing us to rewrite (3) as

$$\begin{aligned} & \min\nolimits_{Q,\Lambda}(1-\alpha)\mathrm{Tr}(VPV^T - Q\Lambda Q^T)^2 + \alpha\mathrm{Tr}(Q\Lambda Q^TL) \\ = \ & \min\nolimits_{Q,\Lambda}(1-\alpha)\mathrm{Tr}(P - V^TQ\Lambda Q^TV)^2 + \alpha\mathrm{Tr}(Q\Lambda Q^TL) \end{aligned}$$

[1]In practice, we rescale the matrix $M^{(\mathcal{T})}$ to have a norm similar to L. The objective is to give the two parts of the objective function roughly equal scale so that α is more meaningful. We omit this rescaling from the presentation for the sake of simplicity.

If we now define $G = V^TQ$, we can replace $Q = VG$ to get

$$\begin{aligned} & \min\nolimits_{G,\Lambda}(1-\alpha)\mathrm{Tr}(P - G\Lambda G^T)^2 + \alpha\mathrm{Tr}(VG\Lambda G^TV^TL) \\ = \ & \min\nolimits_{G,\Lambda}(1-\alpha)\mathrm{Tr}(P - G\Lambda G^T)^2 + \alpha\mathrm{Tr}(V^TLVG\Lambda G^T) \\ = \ & \min\nolimits_{G,\Lambda}\mathrm{Tr}((1-\alpha)P^2) + \mathrm{Tr}((1-\alpha)G\Lambda G^TG\Lambda G^T) \\ & -2\mathrm{Tr}((1-\alpha)PG\Lambda G^T) + \mathrm{Tr}(\alpha V^TLVG\Lambda G^T) \end{aligned}$$

Note that P is a constant, so we can drop the first trace term when we are minimizing. Collecting the third and fourth trace terms together, we get

$$\min_{G,\Lambda}\mathrm{Tr}((1-\alpha)G\Lambda G^TG\Lambda G^T) - 2\mathrm{Tr}(BG\Lambda G^T)$$

where $B = (1-\alpha)P - \frac{1}{2}\alpha V^TLV$. Note that B is constant, so we can add $\mathrm{Tr}(\frac{1}{1-\alpha}B^2)$ to complete the square without affecting the minimization (4), and then factor to produce the final objective (5)

$$\begin{aligned} & \min\nolimits_{G,\Lambda}\mathrm{Tr}((1-\alpha)G\Lambda G^TG\Lambda G^T) - 2\mathrm{Tr}(BG\Lambda G^T) \\ & \qquad +\mathrm{Tr}(\tfrac{1}{1-\alpha}B^2) \qquad (4) \\ = \ & \min\nolimits_{G,\Lambda}\mathrm{Tr}(\tfrac{1}{\sqrt{1-\alpha}}B - \sqrt{1-\alpha}G\Lambda G^T)^2 \\ = \ & \tfrac{1}{\sqrt{1-\alpha}}\min\nolimits_{G,\Lambda}\mathrm{Tr}(B - (1-\alpha)G\Lambda G^T)^2 \qquad (5) \end{aligned}$$

This now has the same form as the standard MDS problem, so we can solve it by finding the decomposition $B = USU^T$. For a d dimensional embedding, we then set Λ to be the top d eigenvalues of S rescaled by $(1-\alpha)$ and G to be the corresponding eigenvectors from U.[2] We have now obtained a closed form solution to our mixed objective function.

While we will not explore it in greater detail here, it is worth noting that this derivation represents a general-purpose way to combine two embedding techniques into one. One of the techniques must have an MDS-like cost function (e.g. Isomap), while the other must have a PCA-like cost function (e.g. LLE, LEM). This means we can potentially create a wide variety of hybrids that combine different similarity measures. For now we will focus on using transformation-derived distances to improve the embedding technique.

6 Transformations

We consider three transformation functions in our experiments. The first, $\mathcal{T}_{shift}$, is a simple *shift* operator that translates the image by a fixed number of pixels in one direction

[2]When B is not positive semi-definite, one can add λI to B, where λ is the absolute value of the largest negative eigenvalue of S.

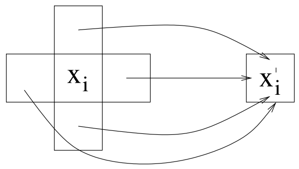

Figure 3. Illustration of local pixel transformation to produce $x^{'}$ from x.

(one pixel, horizontally, in our experiments). This shift operator wraps around at the boundaries. The second transformation, $\mathcal{T}_{rot}$, is a fixed angel rotation about the center of the image.

The third transformation, $\mathcal{T}_{local}$, is a data-dependent transformation that attempts to characterize the relationship between an image and its nearest Euclidean distance neighbour as a parameterized local filter applied over the whole image. Let x be the original image and x' be its nearest Euclidean distance neighbour. Each pixel x_i' in x' is determined by a weighted combination of corresponding neighbouring pixels in the original image, $x_i' = \theta^T x_{N(i)}$, where $x_{N(i)}$ is a vector containing the pixels neighbouring x_i and θ is a weight vector. An illustration is given in Figure 3. When using this transformation, a weight vector is computed for each image that minimizes the Euclidean distance between the transformed original and the nearest neighbour to the original, $min_\theta ||x' - \mathcal{T}_{local}(x, \theta)||$. This is why we call it a “data-dependent” transformation. In all of the experiments presented here, the neighbourhood was the eight immediate neighbours.

7 Experimental Results

We experimented with embeddings for a variety of images using the following methods: LLE, Isomap, LEM, and *transformation-corrected LEM* (TC-LEM) with the tangent correction and the transformation reconstruction correction, respectively. These last two methods correspond to using the combined embedding objective (3), where L is LEM’s Laplacian matrix. Each method was tried with a variety of neighbourhood settings and the best chosen. In most cases LEM, LLE, and Isomap behaved very similarly so we use LEM here as a representative for comparison. Parameters are specified in parentheses after the method (i.e. LEM(k) and TCE-LEM(k, α), where k is the number of neighbours used to build the Laplacian). Note that TCE-LEM(k, 1) is effectively identical to LEM(k).

Figure 5 shows the 2-D embedding using of a set of ten handwritten digit images (0 through 9 - see Figure 4 for the raw images), with each image shifted horizontally against a white background. The TC-LEM embedding used the $\mathcal{T}_{local}$ transformation and the tangent correction distance. Plotting the actual images leads to overlaps, so the plot only shows the number associated with each digit and indicates the degree of shift by the darkness or lightness of the digit (darkest - leftmost and lightest - rightmost). At first glance the manifold produced by LEM seems informative, and it has indeed captured the shift of the images along the horizontal axis. However, the other dimension is essentially meaningless, and LEM is incapable of distinguishing the digits. By contrast, TC-LEM, using $\mathcal{T}_{local}$, captures the shift of the images on the vertical axis and also manages to separate the digits effectively along the horizontal axis.

Figure 6 shows similar results for TC-LEM using $\mathcal{T}_{shift}$ and the reconstruction transformation distance. Again, TC-LEM is able to cluster the digits while still capturing the shift, although there is some overlap in the densest region.

Figure 7 shows the effects of adjusting the mixing parameter, α, for TC-LEM using $\mathcal{T}_{local}$ and tangent correction. Recall that $\alpha = 1$ is pure LEM, which tends to sequence the data by shift. $\alpha = 0$ is using the transformation-derived correction alone, and tends to cluster the data. This tendency toward clustering vs. sequencing is consistent in our experience and suggests that TC-LEM may be viewed as simultaneously trying to cluster and embed. The α parameter can be used to control this tendency in a straightforward fashion.

Figure 8 show the 2-D embedding of a set of eighteen images of a teapot. The teapot is viewed from two different angles (see Figure 4 for the raw images) and each subset is rotated in the plane in nine steps through to 180 degrees. LEM captures the rotation (along the horizontal axis) but fails to distinguish between the two views (the vertical axis has no readily apparent meaning and the two sets overlap almost perfectly). TC-LEM, using $\mathcal{T}_{local}$ and tangent correction, captures both rotation in the plane (as a “loop” of images) and the two distinct views of the teapot (there are two loops). Figure 9 shows similar results for $\mathcal{T}_{rot}$ and transformation reconstruction distance. In this case, however, it broke the set into four clusters instead of two, suggesting that paths found in reconstruction space did not completely connect the images within a cluster.

In general, all methods are capable of identifying meaningful relationships in the data but only TC-LEM is reliably capable of capturing both the dimension corresponding to the sequencing of images and the other distinguishing features (e.g. different objects).

8 Conclusions

We have demonstrated that known transformations inherent in the image domain can be used to augment nonlinear embedding techniques by correcting potentially misleading Euclidean distances. Transformation corrected em-

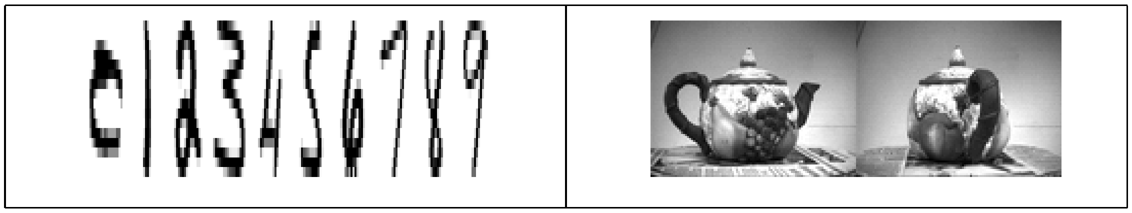

Figure 4. Original digit and teapot images used in experiments.

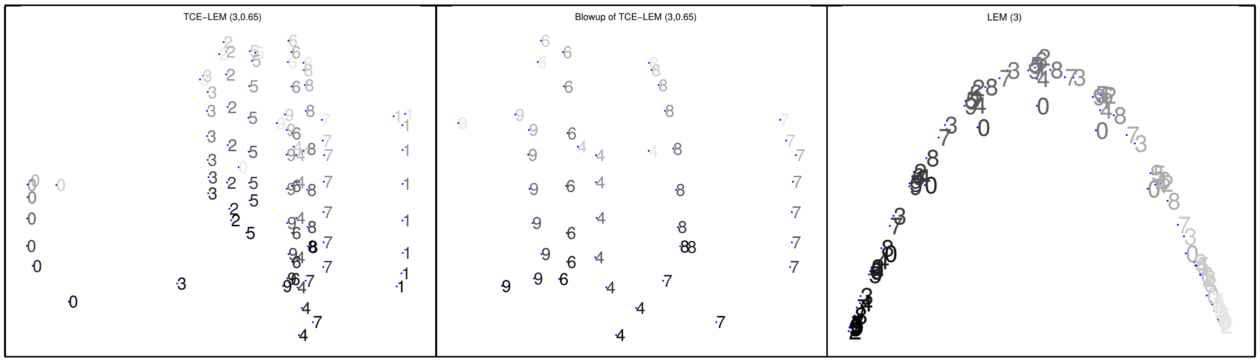

Figure 5. Schematic plot of two-dimensional manifold found by (from left to right) TC-LEM$(3, 0.65)$ using $\mathcal{T}_{local}$ and tangent correction (left), a blowup of the dense part of that plot (middle), and LEM(3) (right) for images of all ten digits with multiple images for each digit shifted horizontally by varying amounts. Actual images are not plotted but the shift is indicated by the darkness of the digit (leftmost - darkest; rightmost - lightest).

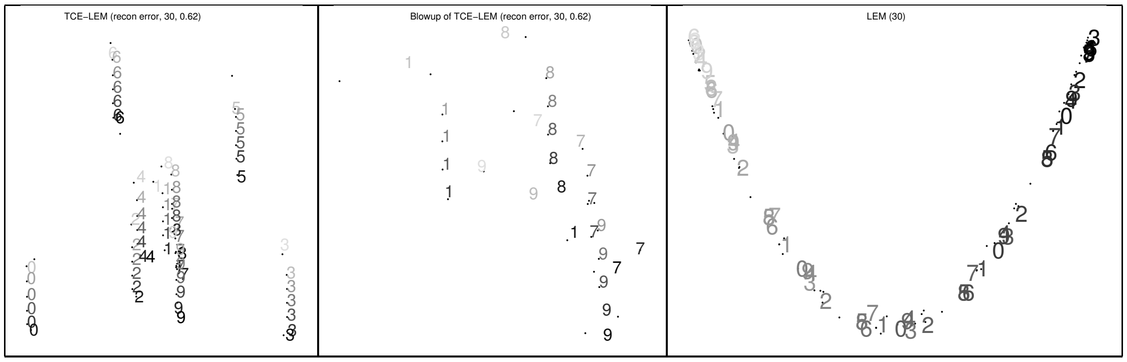

Figure 6. Schematic plot of two-dimensional manifold found by (from left to right) TC-LEM$(30, 0.62)$ with $\mathcal{T}_{shift}$ and reconstruction error (left), a blowup of the dense part of that plot (middle), and LEM(30) (right) for images of all ten digits with multiple images for each digit shifted horizontally in one pixel increments. Actual images are not plotted but the shift is indicated by the darkness of the digit (leftmost - darkest; rightmost - lightest). For readability, only every fifth digit position is plotted.

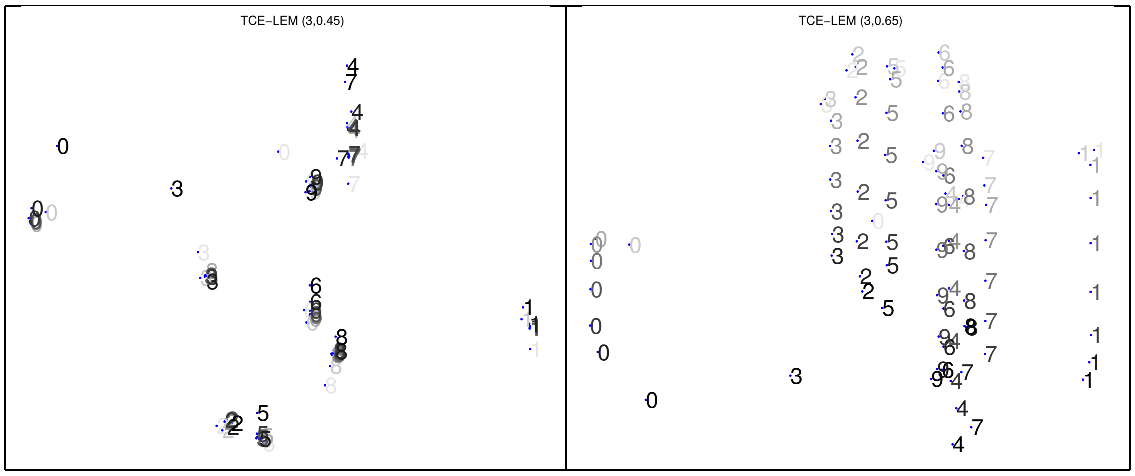

Figure 7. TC-LEM on digit data using $\mathcal{T}_{local}$ and tangent correction for two values of $\alpha = 0.45$ and $\alpha = 0.65$, showing the shift between clustering and spreading depending on α.

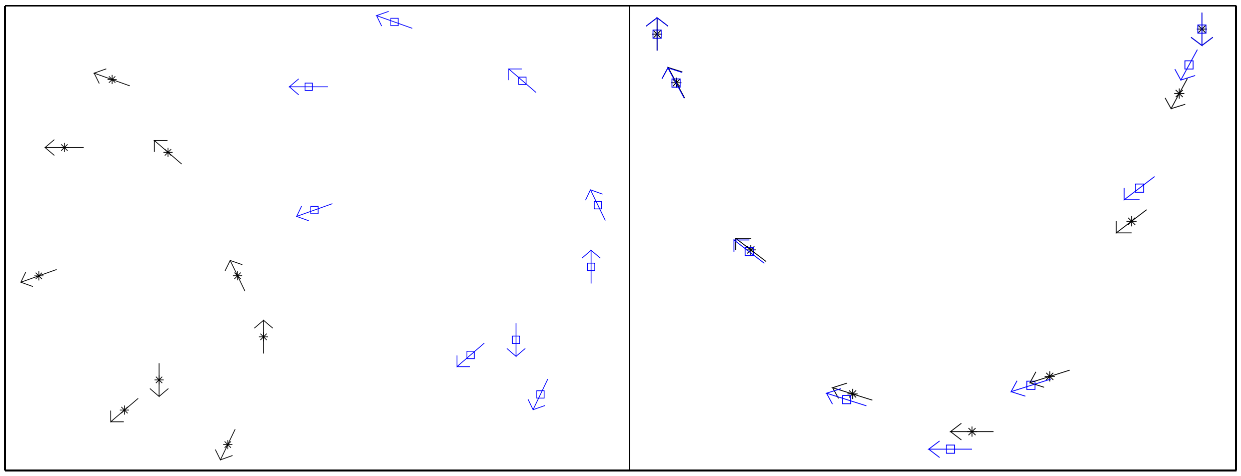

Figure 8. Schematic plot of two-dimensional manifold found by (left) TC-LEM(0.2) using $\mathcal{T}_{local}$ and tangent correction, and (right) LEM for images of a teapot viewed from two different angles and rotated in the plane. Arrows indicate the angle of rotation for each image and points (square or star) indicate the teapot image.

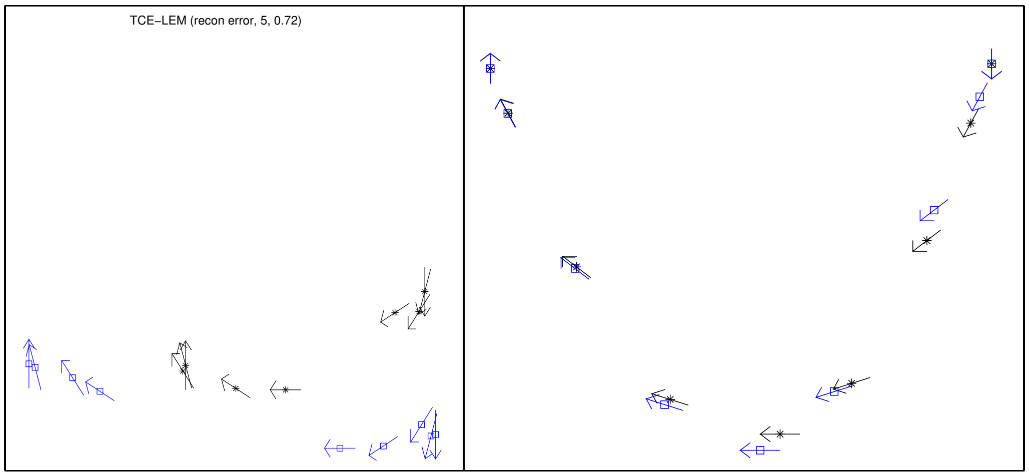

Figure 9. Schematic plot of two-dimensional manifold found by (left) TC-LEM(0.2) using $\mathcal{T}_{rot}$ and transformation reconstruction distance, and (right) LEM for images of a teapot viewed from two different angles and rotated in the plane.

beddings reliably capture the dimension corresponding to a sequence of such transformations and other distinguishing features along with it. Computationally, the method retains the advantages of the techniques we augment by having a closed form solution for the optimization and represents a general purpose method for combining MDS-derived and PCA-derived embedding methods. Future work consists in extending the approach to use multiple transformations simultaneously, more sophisticated methods for characterizing the local manifold, and trying new combinations of embedding techniques.

Acknowlegements

Supported by the Alberta Ingenuity Centre for Machine Learning, NSERC, MITACS, and the CRC programme.

References

[1] I. Jolliffe. *Principal Component Analysis*. Springer-Verlag, 1986.

[2] T. Cox and M. Cox. *Multidimensional Scaling*. Chapman Hall, 2nd edition edition, 2001.

[3] S. Mika, B. Schölkopf, A. Smola, K.-R. Müller, M. Scholz, and Rätsch G. Kernel PCA and de-noising in feature spaces. In *Advances in Neural Information Processing Systems 11 (NIPS'98)*, 1999.

[4] Sam Roweis and Lawrence Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, December 2000.

[5] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, December 2000.

[6] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[7] K. Q. Weinberger and L. K. Saul. Unsupervised learning of image manifolds by semidefinite programming. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR-04)*, volume II, pages 988–995, 2004.

[8] Brendan J. Frey and Nebojsa Jojic. Transformation-invariant clustering and dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–17, January 2003.

[9] Patrice Simard, Yann LeCun, and John S. Denker. Efficient pattern recognition using a new transformation distance. In *Advances in Neural Information Processing Systems 5 (NIPS-92)*, pages 50–58, 1993.

[10] M. J. Black and A. Jepson. EigenTracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision*, 26(1):63–84, 1998.

[11] O. Chapelle and B. Schölkopf. Incorporating invariances in non-linear support vector machines. In *Advances in Neural Information Processing Systems 14*, 2002.