Robust Non-linear Dimensionality Reduction using Successive 1-Dimensional Laplacian Eigenmaps

Samuel Gerber SGERBER@CS.UTAH.EDU
Tolga Tasdizen TOLGA@SCI.UTAH.EDU
Ross Whitaker WHITAKER@CS.UTAH.EDU
Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, UT 84112

Abstract

Non-linear dimensionality reduction of noisy data is a challenging problem encountered in a variety of data analysis applications. Recent results in the literature show that spectral decomposition, as used for example by the Laplacian Eigenmaps algorithm, provides a powerful tool for non-linear dimensionality reduction and manifold learning. In this paper, we discuss a significant shortcoming of these approaches, which we refer to as the *repeated eigendirections problem*. We propose a novel approach that combines successive 1-dimensional spectral embeddings with a data advection scheme that allows us to address this problem. The proposed method does not depend on a non-linear optimization scheme; hence, it is not prone to local minima. Experiments with artificial and real data illustrate the advantages of the proposed method over existing approaches. We also demonstrate that the approach is capable of correctly learning manifolds corrupted by significant amounts of noise.

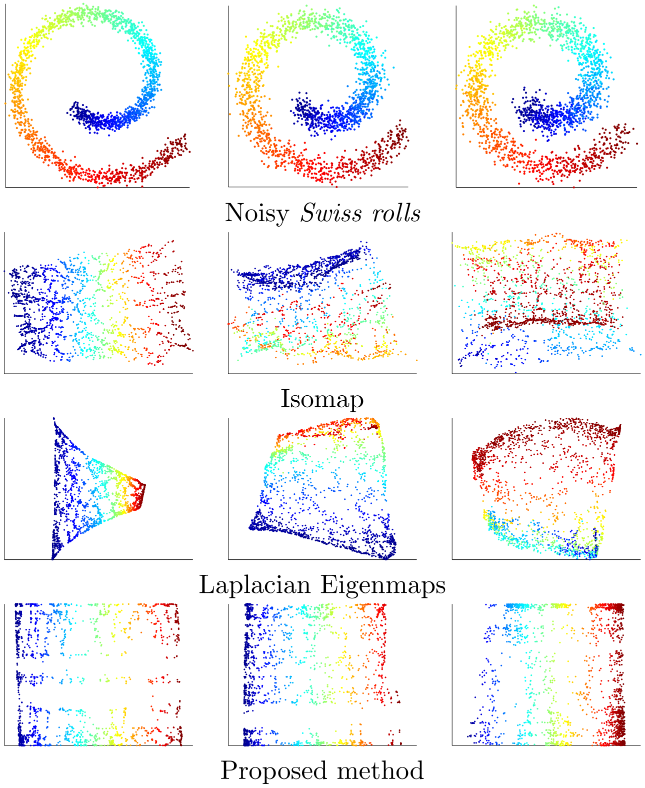
1. Introduction

Dimensionality reduction is an important procedure in various high-dimensional data analysis problems. Applications range from image compression (Ye et al., 2004) to visualization (Vlachos et al., 2002). Often the data lies on a low dimensional manifold embedded in a high dimensional space and the dimensionality of the data can be reduced without significant loss of information. However, in many cases the lower dimensional manifold is non-linear, which makes common linear approaches, such as principal component analysis (PCA) (Jolliffe, 1986) and multidimensional scaling (MDS) (Cox & Cox, 1994) unsuitable. A common example of a non-linear manifold embedded in a high dimensional space is image vectors of an object taken from different angles and lighting directions (Lin et al., 2006). In such a case, the dimensionality is restricted by the degrees of freedom of the physical constraints under which the images were taken. Whereas, the data has a much greater dimensionality depending on the resolution of the image. Analysis of climate data (Gámez et al., 2004) in geophysics or the study of protein motions (Das et al., 2006) in computational biology are other examples.

Isomap (Tenenbaum et al., 2000), chooses a lower dimensional embedding in which Euclidean distances approximate the geodesic distances on the manifold in the high-dimensional space. While this method has many desirable properties, such as distance preservation and a closed-form solution, the geodesic distance approximation depends on the definition of local connectivity for each data point and can be sensitive to noise. Figure 1(top row) shows increasingly noisy versions of a *Swiss roll* data set. The 2nd row illustrates the 2-dimensional embedding obtained by the Isomap algorithm. Tenenbaum et al. propose k-nearest neighbors and ϵ-radius balls as two methods of determining local connectivity. For data with moderate and higher amounts of noise, we were unable to fine tune the algorithmic parameters to recover the correct manifold structure. This can be seen in the 2nd row of Figure 1 where the results for the noisier data sets demonstrate the wrong neighborhood relationships in the lower dimensional embedding (different colors next to each other). A more detailed discussion of the topological stability of Isomap with noisy data can be found in Balasubramanian and Schwartz (2002). Other recent approaches, such as Laplacian

Appearing in *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, OR, 2007. Copyright 2007 by the author(s)/owner(s).

Eigenmaps (Belkin & Niyogi, 2003), exploit proximity relations instead of geodesic distances and use elements of spectral graph theory to compute a lower dimensional embedding. We found that in comparison to Isomap, Laplacian Eigenmaps is more robust in regard to noisy input data. This can be attributed to the lack of a geodesic distance computation. Figure 1 (3rd and 4th row), shows the results of Laplacian Eigenmaps and our method, respectively, on the noisy *Swiss roll* data. Neighborhood relationships are correctly preserved by both methods for the noise levels shown.


Noisy *Swiss rolls*

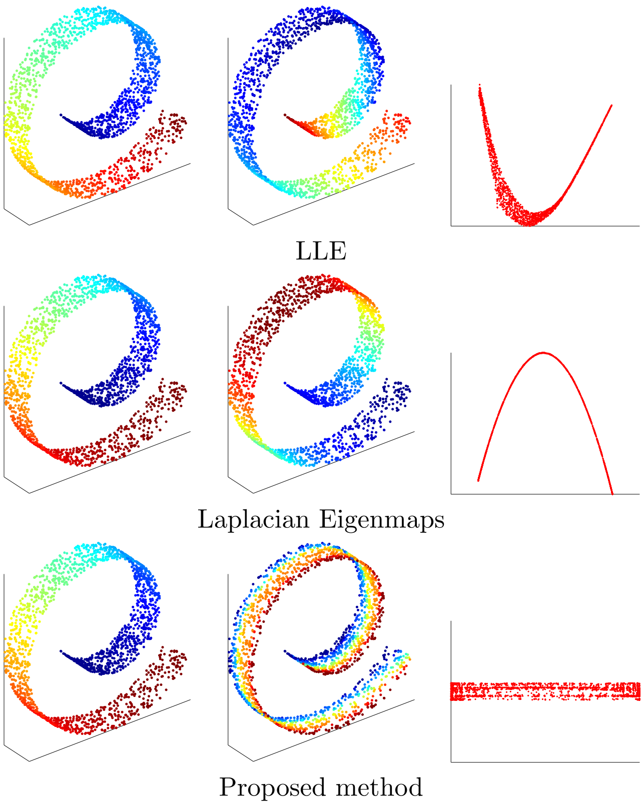
Isomap

Laplacian Eigenmaps

Proposed method

Figure 1. The *Swiss roll* $(\theta\cos(\theta), y, \theta\sin(\theta))$ is a 2D manifold (parametrized by θ and y) in $\mathbb{R}^3$ that is commonly used in dimensionality reduction experiments. Top row: *Swiss rolls* colored by θ, viewed along the y-axis. Independent normally distributed noise with variance increasing from left to right was added. 2D embeddings, also colored by θ, found by Isomap (2nd row), Laplacian Eigenmaps (3rd row) and our approach in the (4th row).

Unfortunately, dimensionality reduction methods based on spectral graph theory implicitly put severe constraints on the shape of the low dimensional manifold. The eigenvectors computed from the graph Laplacian are orthogonal but not independent; in other words, eigenvectors can be strongly correlated locally. This is not sufficient enough to guarantee a satisfactory parametrization. Figure 2 illustrates this problem, which we refer to as *repeated eigendirection*. The two eigenvectors computed from the graph Laplacian and LLE methods recover only the first non-linear dimension of the manifold. The second non-linear dimension is not discovered. The strong local correlations between the two eigenvectors can also be seen by plotting them on the plane (Figure 2, 3rd column). The repeated eigendirection problem occurs when the extent of the underlying non-linear dimensions of the manifold differ significantly. We discuss this problem in more depth in Section 3.


LLE

Laplacian Eigenmaps

Proposed method

Figure 2. Repeated eigendirection problem. The first two columns show the first two eigenvectors colored on the *Swiss roll* computed by LLE, Laplacian Eigenmaps our approach. The last column plots the first against the second eigenvector by each approach. The two eigenvectors display strong local correlations using LLE and Laplacian Eigenmaps, but are independent using the proposed method. Note that the proposed method is also able to recover approximate proportions of the manifold as discussed further in Section 4.

We propose a method that successively collapses the dimensions of the low dimensional manifold based on an advection scheme driven by the eigenvector. The term advection describes the process of transportation in fluids. In our setting we *flow* the individual points along the low dimensional manifold. The *flow* on the low dimensional manifold is defined by a vector field based on the eigenvector computed from the graph Laplacian. This collapses the dimension of the low dimensional manifold defined by this eigenvector and this dimension is therefore eliminated from subsequent eigenvector computations. Our algorithm consists of computing an eigenvector based on the current points, advect the points along the eigenvector and then repeat these two steps (using the advected points) until the embedding is uncovered. The eigenvectors computed at each iteration as described above gives the low dimensional parametrization. Our method resolves the repeated eigendirection problem and is capable of dealing with significant amounts of noise, while still having the desirable properties of eigendecomposition based methods, namely having a solution that does not suffer from local minima. A significant drawback of this approach is the high computational cost for the advection. For the *Swiss roll* example with 2000 points in 3 dimensional space in Figure 1, it takes about 10 minutes to compute the complete low dimensional embedding.

2. Related Work

Traditional methods such as PCA (Jolliffe, 1986) or MDS (Cox & Cox, 1994) work well under the condition that the lower dimensional manifold is close to linear. Various methods have been developed for the non-linear case. Recently there has been an increased amount of interest in non-linear dimensionality reduction methods that do not depend on an iterative minimization procedure that can be prone to local minima. Laplacian Eigenmaps (Belkin & Niyogi, 2003) uses ideas from spectral graph theory, particularly the eigendecomposition of the graph Laplacian $\mathcal{L}$, to preserve local neighborhoods in the low dimensional embedding. LLE (Roweis & Saul, 2000) and its variations (Zhang & Zha, 2005; Yang et al., 2005) also try to find a neighborhood preserving low dimensional embedding. LLE tries to preserve neighborhood relations by first computing weights that describe each point optimally from its nearest neighbors using barycentric coordinates. Then an embedding is computed that minimizes a quadratic error function based on the computed weights. LLE and Laplacian Eigenmaps are closely related; the optimization problem as stated by LLE can be reformulated as trying to find the eigenfunctions of $\mathcal{L}^2$ (Belkin & Niyogi, 2003). Hessian Eigenmaps (Donoho & Grimes, 2003) addresses problems with non-convex manifolds by using the Hessian instead of the Laplacian. We have found that all methods which depend solely on spectral decomposition of a matrix that captures local interactions are susceptible to the repeated eigendirection problem.

Other approaches that have closed form solutions include Isomap (Tenenbaum et al., 2000) a non-linear extension to MDS based on approximate geodesic distances computed by shortest paths on a graph. Isomap does not suffer from the repeated eigendirection problem but is sensitive to noise in the input as discussed in Figure 1. Kernel PCA (Schölkopf et al., 1998) extends PCA to handle non-linear cases using the *kernel trick*. It has been shown that spectral embedding methods are directly related to kernel PCA (Bengio et al., 2004). The difficulty in kernel based methods is to choose an appropriate kernel. Recent work by Weinberger et al. (2004) addresses this problem and provides promising results.

Other approaches to non-linear dimensionality reduction use iterative optimization techniques. Self organizing maps (Kohonen, 1997) learn a mapping from the high dimensional to a fixed lower dimensional space, usually a 2-dimensional Cartesian grid. Topology preserving networks (Martinetz & Schulten, 1994) use aspects from computational geometry combined with a neural network to find a low dimensional embedding. Various other neural network based approaches that utilize bottleneck hidden layers have been proposed (Bishop, 1995; Perantonis & v. Virvilis, 1999; Lluis Garrido, 1999; Lee & Verleysen, 2002; Hinton & Salakhutdinov, 2006). Another approach tries to minimize the Kullback-Leibler distance between a probability density function (PDF) defined in the high dimensional and a PDF in the low dimensional space (Hinton & Roweis, 2002).

3. Algorithm

The goal of dimensionality reduction is to find a set of points $Y \in \mathbb{R}^l$ matched to a set of points $X \in \mathbb{R}^h$, where $l << h$, such that Y embodies the information contained in X. A high level outline of out approach is shown in Algorithm 1 and Figure 3 shows the progression of the algorithm. The first and second step of the algorithm are closely related to the Laplacian Eigenmaps approach; however, we compute only one eigenvector corresponding to the smallest non-zero eigenvalue, thereby avoiding the repeated eigendirection problem. The third step uses this eigenvector to define a flow field on the manifold, which is integrated by an advection scheme, Figure 3 (b)-(d). This step collapses the dimension found by the eigenvector and eliminates this dimension from subsequent eigenvector computations.

Algorithm 1 Successive Laplacian Eigenmaps

```
Set $\tilde{X}^1 = X$
for $i = 1$ to $l$ do
    1. Build graph Laplacian from $\tilde{X}^i$
    2. Compute only the 1st (smallest non-zero eigenvalue) eigenvector from the graph Laplacian and store it in $\mathbf{f}_1^i$.
    3. Collapse the dimension found in step 2 by performing advection along the eigenvector. This leads to new points $\tilde{X}^{i+1}$
end for
Construct $l$-dimensional embedding $Y$ by
    $\mathbf{y}_i \leftarrow (\mathbf{f}_1^1(i), \dots, \mathbf{f}_1^l(i))^T$.
```

3.1. Laplacian Eigenmaps & Weight Selection

The Laplacian Eigenmaps approach is based on the observation that the solution to the minimization problem[1]

$$\arg\min_Y \sum_i \sum_j (\mathbf{y}_i - \mathbf{y}_j)^2 w_{ij} \qquad (1)$$

can be computed by the eigensystem

$$\mathbf{Lf} = \lambda \mathbf{f}, \qquad (2)$$

where $\mathbf{L}$ is the unnormalized graph Laplacian defined by $\mathbf{D}-\mathbf{W}$. $\mathbf{D}$ is diagonal matrix with $d_{ii} = \sum_i w_{ji}$ and $\mathbf{W}$ is an adjacency matrix with an edge between i and j if $x_j \in N_i$ with N_i is the set of neighbors of $\mathbf{x}_i$ weighted by a Gaussian kernel. Then the l-dimensional embedding Y is constructed as $\mathbf{y}_i \leftarrow (\mathbf{f}_1(i), \dots, \mathbf{f}_l(i))^T$, where the $\mathbf{f}_i$ are sorted by ascending λ. The first eigenvector $\mathbf{f}_0$ with $\lambda = 0$ is omitted because it is a constant. Alternatively, one can use the normalized version of the Laplacian ($\mathcal{L} = \mathbf{D}^{-1/2}\mathbf{LD}^{-1/2}$), which amounts to solving the generalized eigenvalue problem $\mathbf{Lf} = \lambda \mathbf{Df}$.

For our approach we compute the weights as

$$w_{ij} = (1-\alpha)G(\| \mathbf{x}_i - \mathbf{x}_j \|, \sigma) + \alpha I(\mathbf{x}_i, \mathbf{x}_j) \qquad (3)$$

where $G(x, \sigma)$ is a Gaussian function with zero mean and standard deviation σ. $I(x_i, x_j)$ is an indicator function

$$I(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} 1, & \text{if } \mathbf{x}_j \in N_i \text{ or } \mathbf{x}_i \in N_j; \\ 0, & \text{otherwise.} \end{cases}$$

The standard deviation of the Gaussian kernel, σ, plays an important role. If σ is to large, points that are far apart in geodesic distances will get to much weight and cause a short-circuit between separate parts of the manifold. While rare for data lying on non-noisy manifolds, this problem becomes extremely common in noisy data. Figure 4 illustrates the problem: an outlier between two arms of the spiral can cause a short-circuit if its connections are given large weights in Equation (1). For this reason, using equal weights for the k-nearest neighbors is not suitable for noisy data. In contrast, using a Gaussian kernel with an appropriate σ will place small weights for all edges connecting to such outlier points, which reduces their influence in Equation (1) allowing the correct parametrization to be recovered. Furthermore, we have found that the normalized Laplacian also amplifies the influence of noisy points in Equation (1); hence, we use the unnormalized Laplacian.

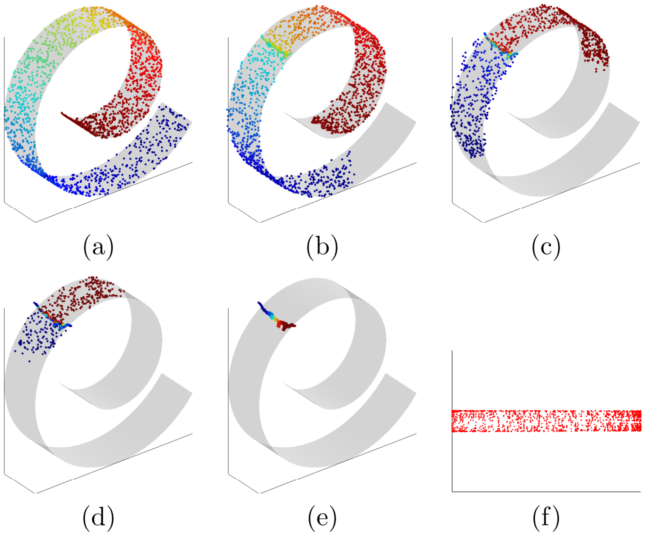
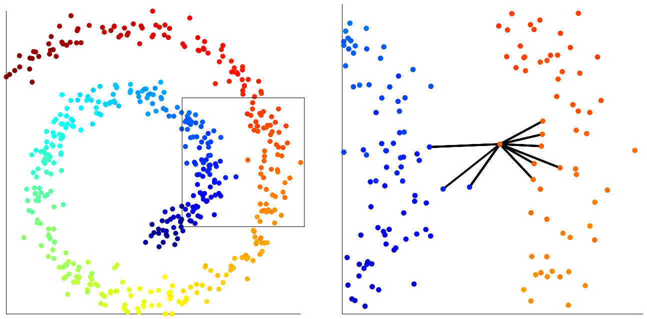


Figure 3. Progression of the algorithm for the *Swiss roll*. (a) Input colored by first eigenvector. (b) - (d) Snapshots of the advection colored by first eigenvector. (e) Completely advected points colored by the new eigenvector computed on the advected points. (f) Final solution formed with the eigenvectors from (a) and (e).

We propose a method to compute σ directly from the input data using ideas from kernel based density estimation techniques. One can view the points $\mathbf{x} \in \mathbb{R}^h$ as drawn from a probability distribution function (PDF). This PDF can be estimated using Parzen window density estimation as $P(\mathbf{x}) \approx \sum_i G(\mathbf{x} - \mathbf{x}_i, \sigma)$. We choose the σ that minimizes the entropy of this Parzen density estimate. It can be shown that this choice of σ guarantees that as the number of points goes to infinity the exact PDF is recovered (Awate & Whitaker, 2006; Parzen, 1962). We have found this approach to be robust in the presence of noisy input data. Typically, outliers get very small edge weights with this approach or can be disconnected from the rest of the points. The

[1] The notation follows closely as used in (Belkin & Niyogi, 2003).

indicator function I is used to avoid (numerically) disconnected components. In all experiments, the weight for the indicator function in Equation (3) is fixed at 0.01 which provides the regularization necessary for avoiding disconnected components but does not overly influence the rest of the data points in the solution of Equation (1).



(a) (b)

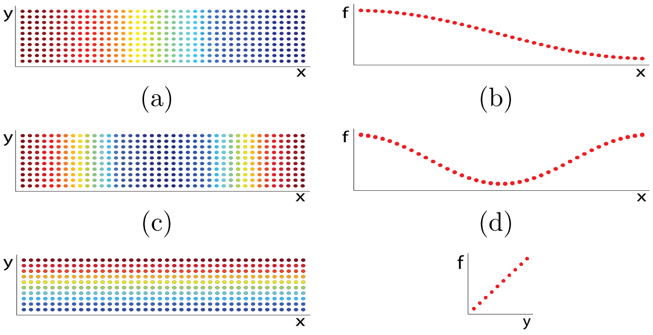
Figure 4. (a) A noisy 2D spiral data set. (b) An outlier between adjacent arms of the spiral can cause a short-circuit if its connections are given large weights in Equation (1).

3.2. Repeated Eigendirection Problem

To understand the repeated eigendirection problem, we can analyze the cost function given in Equation (1). Consider a regular $n \times m$ unit grid as shown in Figure 5. For simplicity assume the points $\mathbf{p}_{x,y} = (x, y)^T$ to be 4-connected with equal weights w. Notice that the number of vertical and horizontal edges is equal. Consider the eigenvector shown in Figure 5 (a,b). This eigenvector can be approximated by a linear function $f(\mathbf{p}_{x,y}) = sx$. In this case, a vertical edge contributes 0 to the cost function. The cost for a horizontal edge is $f(\mathbf{p}_{x+1,y}) - f(\mathbf{p}_{x,y})w = (x + 1 - x)sw = sw$. We can compute s by using the fact that the eigenvector is constrained to have length 1, therefore

$$1 = \sum_{y=1}^{m}\sum_{x=1}^{n} f(\mathbf{p}_{x,y})^2 = m\sum_{x=1}^{n}(xs)^2 = m(\frac{n(n+1)}{2}s)^2$$

and $s = \sqrt{2\,(mn(n+1))^{-1}}$. The total cost for this eigenvector is then $(n+1)(m+1)\sqrt{2\,(mn(n+1))^{-1}}w$. An eigenvector that causes the repeated eigendirection problem is depicted in Figure 5 (c,d). Using the same arguments as above and noting that the *slope* of the eigenvector is approximately doubled, the total cost is found to be $2(n+1)(m+1)\sqrt{2\,(mn(n+1))^{-1}}w$. Finally for an eigenvector in the vertical direction with $f(\mathbf{p}_{x,y}) = ty$ as in Figure 5(e,f) the total cost is $(n+1)(m+1)\sqrt{2\,(nm(m+1))^{-1}}w$. If n is sufficiently larger than m, this cost will be larger than the cost of the eigenvector in Figure 5 (c,d). Therefore, the eigenvector with the repeated direction problem, shown in Figure 5 (c,d), will be preferred by the spectral decomposition.



(a) (b)

(c) (d)

(e) (f)

Figure 5. Eigenvectors on a regular grid. 1st row: 1st eigenvector colored on plane (a) and plotted against x (b). 2nd row: 2nd eigenvector colored on plane (c) and plotted against x (d). 3rd row: Artificial eigenvector that corresponds to horizontal cut colored on plane (e) and plotted against y (f).

The repeated eigendirection problem can also be analyzed from the viewpoint of spectral clustering. The solution of the eigensystem in Equation (2) is the central subject in the study of spectral graph theory (Chung, 1994). Spectral clustering algorithms use the eigenvectors found by this eigensystem to partition the data. In particular, it can be reasoned that the solution found by (2) are approximate solutions to finding the minimal normalized cut of a graph (Shi & Malik, 2000). The normalized cut is defined as

$$Ncut(A_1, \ldots, A_k) = \sum_{i=1}^{k} \frac{cut(A_i, \overline{A}_i)}{vol(A_i)}$$

with $cut(A, B) = \sum_{i\in A, j\in B} w_{ij}$ and $vol(A) = \sum_{i\in A} d_{ii}$ If we analyze the Laplacian Eigenmaps with this in mind, we can easily see how the repeated eigendirection problem occurs. Figure 5 shows a regularly sampled plane with height h and length $4h$. Assuming equal weights between all the 4-way connected points as before, $Ncut$ is much smaller if the second eigenvector cuts the graph *twice* along the shorter side than a *single* cut along the longer side of the plane.

3.3. Collapsing Dimensions by Advection

We use the first eigenvector computed as described in Section 3.1 to construct a flow field on the data to collapse the manifold along this *direction*. Successive repetition of these steps uncover the dimensions of the low dimensional manifold. This process not only avoids the repeated eigendirection problem, but can also be used to recover approximate proportions of the lower dimensional manifold. From the Laplacian Eigenmaps itself it is not possible to infer this information.

The flow field is constructed to point along the direction of the gradient $\mathbf{g}$ of the eigenvector while constrained to the tangent space of the manifold. The points are then advected towards the mean of the eigenvector $\mathbf{f}$ by the iterative process

$$\mathbf{x}_k^{i+i} = t[\mathbf{f}(\mathbf{x}_k^i) - \mathbf{f}_a]\frac{\mathbf{g}(\mathbf{x}_k^i)}{\| \mathbf{g}(\mathbf{x}_k^i) \|,}$$

where $\mathbf{x}_k^i$ is the kth point at the ith iteration of the advection procedure, t is an adaptive time step and $\mathbf{f}_a$ is the mean of the eigenvector values. For computational efficiency purposes, we compute $\mathbf{f}(\mathbf{x}_k)$ and $\mathbf{g}(\mathbf{x}_k)$ only on the original, unadvected points. Then, values for the advected points can be computed by interpolation using their nearest neighbors in the original point set. Once $\mathbf{g}$ is computed, we use the Cash-Karp integration scheme (a Runge-Kutta variation) with adaptive step size control (Press et al., 1992) for the advection. Advection is performed until all points reach the mean value of the eigenvector. Note that while advection is an iterative process for numerical integration of a flow field, it does not suffer from any local minima problems. The gradient direction of $\mathbf{f}$ at each point $\mathbf{x}_k$ is then computed by the following steps:

1. Define a local neighborhood for each point: Let N_k be the set of n closest points to $\mathbf{x}_k$ Perform PCA on the local neighborhood N_k and build principal component matrix from the eigenvectors corresponding to the l largest eigenvalues $V = \left[\mathbf{v}_0, \ldots, \mathbf{v}_{l-1}\right]$, where l is the dimensionality of the manifold. Transform $\mathbf{x} \in N_k$ into principal components coordinates by $\tilde{\mathbf{x}} = V^T\mathbf{x}$.

2. Fit a linear function in $\mathbb{R}^l$ to eigenvector $\mathbf{f}$ defined on $\tilde{\mathbf{x}}_i$ using weighted linear least squares. We use weights based on the Euclidean distance between points and the distance of the eigenvector values of the points. For $x \in N_k$ we use the weights

$$w(x) = G(\| \mathbf{x} - \mathbf{x}_k \|, \sigma_k)G(\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{x}_k), \sigma_f) \qquad (4)$$

where G are Gaussian kernels with zero mean and $\sigma_k = \frac{\max_{\mathbf{x}_i \in N_k} \|\mathbf{x}_k - \mathbf{x}_i\|}{3}$ and $\sigma_f = \frac{\max(\mathbf{f}) - \min(\mathbf{f})}{10}$. These weights allow for larger neighborhoods which give more faithful linear fits. Possible bridges between points that have large geodesic distances on the low dimensional embedding but small Euclidean distances in the high dimensional space will still get small weight due to the contribution from the second term in Equation (4), and therefore not distort the linear fit.

3. Compute the gradient of the linear function fitted in the previous step to obtain $\tilde{\mathbf{g}}_k$ and transform it into ambient space $\mathbf{g}_k = V\tilde{\mathbf{g}_k}$, which is the advection direction at point $\mathbf{x}_k$.

Note that the advection process described above depends on l, the dimensionality of the target manifold. In this paper, we do not address the problem of choosing the dimensionality of the manifold.

4. Results

Properties of the proposed algorithm and its advantages over other methods were previously discussed in the context of experiments with artificial noisy data. It could be argued that the repeated eigendirection can be resolved by globally scaling the data. However, in general, it is not possible to know the extent of the non-linear dimensions of the manifold without knowing the parametrization of the manifold itself. An example of such a manifold, a plane rolled into a corkscrew shape, is shown in Figure 6. In this case, Laplacian Eigenmaps fails because of the repeated eigendirection problem. Similarly, LLE exhibits the repeated eigendirection problem. The amount of noise leads to short circuits in the shortest path computation in Isomap. The proposed algorithm is able to correctly recover the lower dimensional manifold. Note that the distance traveled by points during the advection scheme described in Section 3.3 allows us to recover the approximate proportions of the manifold as shown in Figure 6(e). In other words, the proposed algorithm is not proven to preserve geodesic distances as in the case of Isomap, but the distance traveled by a point in the advection process can be used as an approximation.

The next experiment uses the artificial head image data set from the Isomap website. This data set shows a single computer rendered head from different vertical and horizontal camera positions, and lighting directions varying from left to right. The intrinsic dimensionality of the images should therefore be approximately three. Our algorithm finds these dimensions as shown in Figure 7.

We performed a similar experiment on images of a real head which we acquired with a fixed light source. In this experiment, the head rotates ±90 degrees in the left-right direction and a smaller range of rotations in the up-down direction. The size of the 2-dimensional manifold described by this movement has an aspect ratio from about 3:1. Figure 8 shows the embeddings found by Laplacian Eigenmaps and the proposed approach. In the Laplacian Eigenmaps results, the first eigenvector finds the left to right dimension, but the

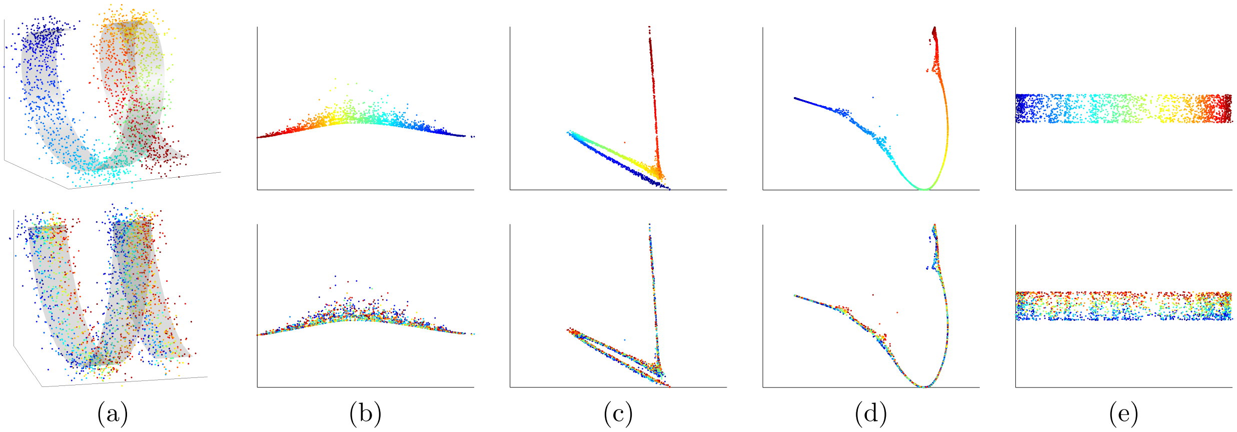


Figure 6. Corkscrew plane (a) with a significant amount of independent normally distributed noise. The two images in the two rows are colored by the two correct parameters of the underlying plane. Embeddings found by (b) Isomap, (c) LLE, (d) Laplacian Eigenmaps and (e) proposed algorithm

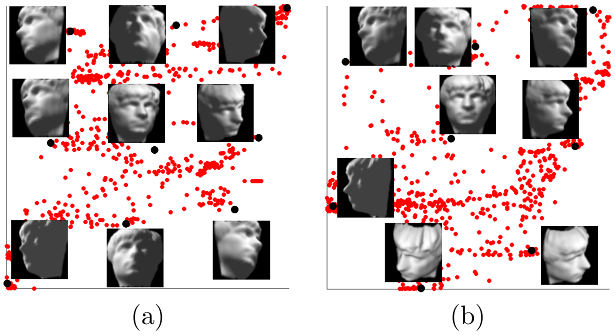


Figure 7. The plots show the 1st vs. 2nd and 1st vs 3rd dimensions of the 3 dimensional embedding found by our approach on the Isomap head data set. (a) The 1st vs. 2nd eigenvector recovers the left-right motion and the left-right lighting. (b) The 1st vs. 3rd eigenvector reveals the left-right and up-down motion.

second eigenvector repeats the same direction. The proposed approach is able to find the left and right and up and down directions. We believe that it also recovers the approximate trajectories that the head followed during the acquisition of the images.

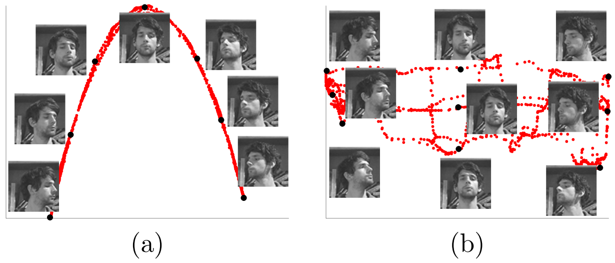


Figure 8. Real head images. Embeddings found by (a) Laplacian Eigenmaps and (b) proposed method.

It is also informative to observe the advection process that a single image from the data set undergoes, as shown in Figure 9. This advection process is the same as the advection on the Swiss roll depicted in Figure 3; however, it is performed in a much higher dimensional space. Observe that image at the end of the advection shown in Figure 9 is recognizable as a face; therefore, we can conclude that the data points remain on the manifold during the advection. This observation suggests the possibility for a new approach to a variety of problems such as face alignment and recognition.

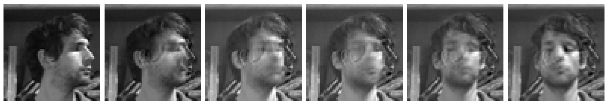


Figure 9. Sequence of a head image during the advection with increasing duration from left to right.

A drawback of our algorithm is the performance. For instance, the real head data set consists of 1041 images with a resolution of 64 × 64 pixels. First, we use PCA to reduce the 4096 dimensions to 1041 without loss of information. On this reduced data set the algorithm takes about 10 hours on a 3.2 GHz Intel Xeon to collapse a single dimension. The bottleneck is the advection process. The computational cost for solving a ordinary differential equation for the advection with accuracy e is exponential in ln(e) (Werschulz, 1991). Roughly we can say, that the run time depends linearly on the input size, the size of the high-dimensional space, the dimension of the manifold and the number of steps during the advection, which is dependent on the curvature and the length of the advection path.

5. Conclusions

In this paper, we demonstrated the repeated eigendirection problem and proposed a novel approach to non-linear dimensionality reduction that resolves this problem. The proposed algorithm also offers significant advantages over existing methods such as Isomap in cases with large amount of noise and where the topology of the manifold is prone to short circuits, such as the *Swiss roll*. A future research direction will be to investigate whether manifolds with such topologies are common in various real world applications. A drawback of the algorithm is the large computational expense of the advection process and the use of the dimensionality of the target manifold in the advection process. Faster advection schemes which do not depend on this knowledge will be another focus of future research.

Acknowledgments

This work was supported by the National Institutes of Health (NIH grant RO1 EB005832). We also thank Xavier Tricoche for the helpful discussions and the anonymous reviewers for many helpful comments.

References

Awate, S. P., & Whitaker, R. T. (2006). Unsupervised, information-theoretic, adaptive image filtering for image restoration. *IEEE Trans. PAMI*, *28*, 364–376.

Balasubramanian, M., & Schwartz, E. L. (2002). The isomap algorithm and topological stability. *Science*, *295*.

Belkin, M., & Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, *15*, 1373–1396.

Bengio, Y., Delalleau, O., Roux, N. L., Paiement, J.-F., Vincent, P., & Ouimet, M. (2004). Learning eigenfunctions links spectral embedding and kernel pca. *Neural Computation*, *16*, 2197–2219.

Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford University Press.

Chung, F. R. K. (1994). *Spectral graph theory*. AMS.

Cox, T. F., & Cox, M. A. A. (1994). *Multidimensional scaling*. Monographs on Statistics and Applied Probability 59. London: Chapman and Hall.

Das, P., Moll, M., Stamati, H., Kavraki, L. E., & Clementi, C. (2006). Low-dimensional, free-energy landscapes of protein-folding reactions by nonlinear dimensionality reduction. *PNAS*, *103*, 9885–9890.

Donoho, D., & Grimes, C. (2003). Hessian eigenmaps: locally linear embedding techniques for high dimensional data. *Proc. of N. A. of Sciences*, *100*, 5591–5596.

Gámez, A. J., Zhou, C. S., Timmermann, A., & Kurths, J. (2004). Nonlinear dimensionality reduction in climate data. *Nonlinear Processes in Geophysics*, *11*, 393–398.

Hinton, G., & Roweis, S. T. (2002). Stochastic neighbor embedding. *NIPS* (pp. 833–840).

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*, 504 – 507.

Jolliffe, I. T. (1986). *Principal component analysis*. Springer-Verlag.

Kohonen, T. (1997). *Self-organizing maps*. New York: Springer-Verlag.

Lee, J., & Verleysen, M. (2002). Nonlinear projection with the isotop method. *ICANN*.

Lin, T., Zha, H., & Lee, S. U. (2006). Riemannian manifold learning for nonlinear dimensionality reduction. *ECCV*.

Lluis Garrido, Sergio Gomez, J. R. (1999). Non-linear dimensionality reduction with input distances preservation. *ANN*.

Martinetz, T., & Schulten, K. (1994). Topology representing networks. *Neural Netw.*, *7*, 507–522.

Parzen, E. (1962). On the estimation of a probability density function and the mode. *AMS*, *33*, 1065–1076.

Perantonis, J. S., & v. Virvilis (1999). Dimensionality reduction using a novel neural network based feature extraction method. *IJCNN*, *2*, 1195–1198.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (1992). *Numerical recipes in c: The art of scientific computing*. Cambridge University Press.

Roweis, S., & Saul, L. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, *290*.

Schölkopf, B., Smola, A., & Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, *10*, 1299–1319.

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE PAMI*, *22*, 888–905.

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

Vlachos, M., Domeniconi, C., Gunopulos, D., Kollios, G., & Koudas, N. (2002). Non-linear dimensionality reduction techniques for classification and visualization. *KDD '02* (pp. 645–651). Edmonton, Alberta, CA: ACM Press.

Weinberger, K. Q., Sha, F., & Saul, L. K. (2004). Learning a kernel matrix for nonlinear dimensionality reduction (p. 106.). New York, NY, USA: ACM Press.

Werschulz, A. G. (1991). *The computational complexity of differential and integral equations*. Oxford Press.

Yang, J., Li, F., & Wang, J. (2005). A better scaled local tangent space alignment algorithm (pp. 1006–1011.).

Ye, J., Janardan, R., & Li, Q. (2004). Gpca: an efficient dimension reduction scheme for image compression and retrieval. *KDD '04* (pp. 354–363). ACM Press.

Zhang, Z., & Zha, H. (2005). Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM J. Sci. Comput.*, *26*, 313–338.