

A Generalised Solution to the Out-of-Sample Extension Problem in Manifold Learning

Harry Strange and Reyer Zwiggelaar

Department of Computer Science, Aberystwyth University,
Aberystwyth, Wales, UK SY23 3DB

Abstract

Manifold learning is a powerful tool for reducing the dimensionality of a dataset by finding a low-dimensional embedding that retains important geometric and topological features. In many applications it is desirable to add new samples to a previously learnt embedding, this process of adding new samples is known as the out-of-sample extension problem. Since many manifold learning algorithms do not naturally allow for new samples to be added we present an easy to implement generalized solution to the problem that can be used with any existing manifold learning algorithm. Our algorithm is based on simple geometric intuition about the local structure of a manifold and our results show that it can be effectively used to add new samples to a previously learnt embedding. We test our algorithm on both artificial and real world image data and show that our method significantly out performs existing out-of-sample extension strategies.

Introduction

Manifold learning is a widely researched statistical tool used to reduce the dimensionality of a dataset by projecting the high-dimensional data onto a representative low-dimensional manifold. At its simplest form this low-dimensional manifold can be the hyperplane of maximum variance resulting in the data being projected onto a linear basis (Hotelling 1933). More recent techniques aim to find a non-linear manifold upon which the data can be projected (Tenenbaum, de Silva, and Langford 2000; Roweis and Saul 2000). Non-linear techniques are able to discover more complex manifolds than their linear counterparts and so pave the way for manifold learning to be used as a powerful statistical tool in image processing (Verbeek 2006), data mining (Patwari, III, and Pacholski 2005) and classification (Strange and Zwiggelaar 2009).

One of the open questions within manifold learning is how a new 'unseen' sample can be mapped into a previously learnt embedding. Consider as an example a simple classification problem involving a set of training samples and a separate set of test samples. We wish to use manifold learning to reduce the dimensionality of these data sets so that

we can perform the classification in the lower-dimensional space. The two options open are, either to combine the training and test sets into one and perform manifold learning on this combined dataset before splitting them again in the low-dimensional space, or to run the manifold learning algorithm on the training set and then apply what has been learnt from this manifold learning process to map the test set into the low-dimensional space. The advantage of the latter approach is that it not only potentially less computationally expensive but it also means that new samples can be continually added to the low-dimensional embedding without the need to recompute the low-dimensional manifold every time. This approach is commonly referred to as the out-of-sample extension. It is worth noting that the out-of-sample extension problem can appear similar in many ways to the problem of incremental learning (Law and Jain 2006), where the low-dimensional manifold is incrementally learnt over a number of iterations of new samples being inserted. This is different from the out-of-sample problem where a new sample simply needs to be mapped into the low-dimensional space without affecting the low-dimensional manifold and requiring a re-learning or change in the manifold parameterization for future learning.

Many existing manifold learning techniques do not naturally contain an out-of-sample extension so research has been undertaken to find ways of extending manifold learning techniques to handle new samples. Bengio et al. (Bengio et al. 2003) presented ways of extending some well known manifold learning techniques: ISOMAP (Tenenbaum, de Silva, and Langford 2000), Locally Linear Embeddings (Roweis and Saul 2000), Laplacian Eigenmaps (Belkin and Niyogi 2003) and Multidimensional Scaling (Cox and Cox 2001), to handle the out-of-sample extension problem. The framework developed by Bengio et al relies on phrasing the out-of-sample problem as a kernel problem where a continuous kernel function is defined in order to generalize the existing embeddings to a new data point. Other approaches have been presented that attempt to extend specific manifold learning algorithms to handle new samples (e.g. Maximum Variance Unfolding (Weinberger and Saul 2006; Chin and Suter 2008), LTSA & LLE (Zhang and Zha 2005; Saul and Roweis 2003)) but at present there is little work on creating a generalized solution to the out-of-sample problem. Recently Yang et al. (Yang et al. 2010) proposed a

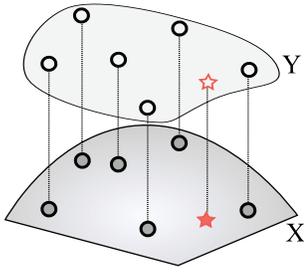


Figure 1: The new sample is attached to its nearest neighborhood in the high-dimensional space and then projected onto the low-dimensional hyperplane defined by the principal components of that neighborhood.

method for a generalized out-of-sample method based on their manifold learning technique Local and Global Regressive Mapping. Regularization is used to learn a model to allow out-of-sample extrapolation and as such they claim that their framework can be applied to any manifold learning algorithm to enable an out-of-sample extension.

In this paper we present a generalized out-of-sample extension (GOoSE) solution. Unlike existing approaches we do not require information to be retained from the learning process, such as the pairwise distance matrix or the resultant eigenvectors, we simply learn the mapping from the original high-dimensional data and its low-dimensional counterpart. As such our method is independent of any specific manifold learning algorithm. The change in local geometry between the high and low-dimensional spaces provides the information needed to compute the transformation of new samples into the low-dimensional space. This simplicity means that our approach can be used to extend any manifold learning technique to handle the out-of-sample extension problem.

The rest of this paper is structured as follows. We begin by outlining the algorithm behind our generalized solution before moving on to show how this generalized solution performs on artificial and real world data. In the Results section we show how using GOoSE can produce comparable results to the existing out-of-sample techniques described above. We end by presenting conclusions and possible directions for future work.

Algorithm

The basic premise of our algorithm is to find the transformation that maps a new unseen sample's neighborhood from the high-dimensional to the low-dimensional spaces. This transformation is equivalent, as far as possible, to running the manifold learning technique on the given sample.

Given the original data set $\mathbf{X} = \{x_i\}_{i=1}^n \in \mathbb{R}^p$ and its low-dimensional representation $\mathbf{Y} = \{y_i\}_{i=1}^n \in \mathbb{R}^q$, where $q \ll p$, we wish to find the low-dimensional approximation, $\varphi \in \mathbb{R}^q$, of an unknown sample, $\phi \in \mathbb{R}^p$, given that $\phi \ni \mathbf{X}$. We assume that \mathbf{X} is sampled from a hidden manifold \mathcal{M} , that is $\mathbf{X} \subseteq \mathcal{M}$, and also that at a local scale \mathcal{M} is linear (i.e. \mathcal{M} is a C^∞ manifold). Since \mathbf{Y} is the result of manifold learning we can describe \mathbf{Y} in terms of a function on \mathbf{X} . That is

$$\mathbf{Y} = f(\mathbf{X}) \quad (1)$$

Unless we are dealing with a linear manifold learning algorithm such as Principal Components Analysis this function will be difficult to learn at a global scale. Instead we can think of \mathbf{Y} as being built up by individual functions for each sample. That is for the i -th datapoint $y_i = f_i(x_i)$. The out-of-sample problem can thus be thought of as finding a function that best approximates the transformation undergone via manifold learning, that is for an unlearned sample $\min(\|\varphi - \varphi'\|)$ where φ is the actual embedding of the sample and φ' is its estimated embedding. This problem is evidently cyclical as we need the actual embedding to be able to find the function to minimize but we need to minimize the function to find the actual embedding.

To solve this problem it is helpful to take a step back and consider the situation where we know the actual low-dimensional representation, y , of a sample x . To re-create the embedding of x we can examine the local geometric structure around x in the high and low-dimensional spaces. If we assume that the result of running a manifold learning algorithm is a local change in the neighboring geometry of a sample then we can reformulate the problem as that of finding a simple linear transformation

$$y = \mathbf{A}\mathbf{V}x \quad (2)$$

where \mathbf{A} is a similarity transformation matrix and \mathbf{V} is a matrix that projects x into the low-dimensional space. We now seek to find \mathbf{A} and \mathbf{V} that best approximates the local transformation. Given that we know the target dimensionality, q , and we take the manifold to be locally linear we can find the projection matrix \mathbf{V} by performing Principal Components Analysis on a local neighborhood of the k -nearest samples according to Euclidean distance around x . We denote the samples in this neighborhood as $\mathbf{X}_{\mathcal{N}_i}$ and so the principal components are found by

$$\Lambda\mathbf{V} = \mathbf{C}\mathbf{V} \quad (3)$$

where \mathbf{C} is the covariance matrix of $\mathbf{X}_{\mathcal{N}_i}$ and \mathbf{V} is a matrix containing as columns the top q -dimensional eigenvectors sorted according to their associated eigenvalues, Λ . We can now find the low-dimensional representation of x by projecting onto the eigenvectors, $y = \mathbf{V}x$ (Figure 1).

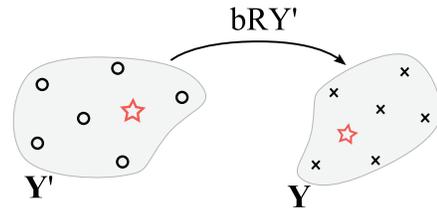


Figure 2: Since we assume that the transformation undergone as a result of manifold learning can be approximated as a local linear transform we aim to find that transform. By applying that transform to the new sample we can find its approximate low-dimensional image.

To find the similarity transformation matrix we need to examine the change in local geometry. We first need to project the local neighborhood $\mathbf{X}_{\mathcal{N}_i}$ into the lower-dimensional space by projecting onto the eigenvectors \mathbf{V} (3). We represent this low dimensional projected neighborhood as \mathbf{Y}' and the same neighborhood of points in the low-dimensional embedding as \mathbf{Y} . For convenience we drop the subscripted \mathcal{N}_i so when referring to \mathbf{Y} we actually mean $\mathbf{Y}_{\mathcal{N}_i}$ and similarly \mathbf{Y}' is $\mathbf{Y}'_{\mathcal{N}_i}$.

We know that

$$\mathbf{Y} \approx \mathbf{A}\mathbf{Y}' \quad (4)$$

and that the transformation matrix \mathbf{A} can be represented in terms of a separate scale and rotation component

$$\mathbf{Y} \approx \mathbf{B}\mathbf{R}\mathbf{Y}' \quad (5)$$

where \mathbf{B} is a non-isomorphic scale matrix and \mathbf{R} is the rotation matrix. The task now becomes to find the scale and rotation that transforms \mathbf{Y}' to \mathbf{Y} (Figure 2).

To find the solution to this problem we use a method from statistical shape theory and find the singular value decomposition (SVD) of the matrix $\mathbf{Y}'^T\mathbf{Y}$.

To find the rotation matrix \mathbf{R} and the scale value \mathbf{b} we first find the singular value decomposition

$$\mathbf{Y}'^T\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad (6)$$

the rotation matrix can then be found by

$$\mathbf{R} = \mathbf{U}\mathbf{V}^T \quad (7)$$

Once the rotation has been applied we find the scale matrix by

$$\mathbf{B} = \begin{bmatrix} \frac{\max(\mathbf{Y}^1) - \min(\mathbf{Y}^1)}{\max(\mathbf{Y}'^1) - \min(\mathbf{Y}'^1)} & \cdots & \cdots \\ \vdots & \ddots & \vdots \\ \cdots & \cdots & \frac{\max(\mathbf{Y}^q) - \min(\mathbf{Y}^q)}{\max(\mathbf{Y}'^q) - \min(\mathbf{Y}'^q)} \end{bmatrix} \quad (8)$$

where \mathbf{Y}^1 indicates the column vector containing all samples along the first dimension of \mathbf{Y} and \mathbf{Y}^q indicates the column vector containing all samples along the q^{th} dimension.

Now we return to the original problem of finding the low-dimensional representation, φ , of an unlearned sample ϕ . As we have shown above, an approximation of the low-dimensional embedding of a neighborhood in the high-dimensional space can be found. So a simple solution to finding the low-dimensional representation of an unlearned sample is to find the rotation and scale transformations of the sample's nearest neighbors and then applying these transformations to the unlearned samples. This can be done by finding the k -nearest neighbors of ϕ in \mathbf{X} , $\mathbf{X}_{\mathcal{N}_\phi}$. We then find the projection matrix of $\mathbf{X}_{\mathcal{N}_\phi}$ according to (3) and the rotation and scale values according to (7) and (8). The low-dimensional representation, φ , of ϕ then becomes

$$\varphi = \mathbf{B}\mathbf{R}\mathbf{V}_{\mathcal{N}_\phi}\phi \quad (9)$$

This process is described in algorithmic form in Algorithm 1.

Algorithm 1 Generalized Out-of-Sample Extension

Require: $x \in \mathbb{R}^D$, $\mathbf{X} \in \mathbb{R}^D$, $\mathbf{Y} \in \mathbb{R}^d$, $k \ll |\mathbf{X}|$

- 1: $idx \leftarrow \text{nn}(x, \mathbf{X}, \mathbf{Y}, k)$
- 2: $\mathbf{\Lambda}\mathbf{V} = \mathbf{C}_\mathbf{X}\mathbf{V}$
- 3: $\mathbf{Z}' = \mathbf{X}_{idx}\mathbf{V}_{1\dots d}$
- 4: $(\mathbf{U}\mathbf{\Sigma}\mathbf{V}) \leftarrow \text{svd}(\mathbf{Z}'\mathbf{Y}_{idx})$
- 5: $\mathbf{B} \leftarrow \text{eye}(d, d)$
- 6: $\text{diag}(\mathbf{B}) = \left[\frac{\text{range}(\mathbf{Y}_{idx}^1)}{\text{range}(\mathbf{Z}'_{idx}^1)} \cdots \frac{\text{range}(\mathbf{Y}_{idx}^d)}{\text{range}(\mathbf{Z}'_{idx}^d)} \right]$
- 7: $\mathbf{T} \leftarrow \mathbf{U}\mathbf{V}^T$
- 8: $y \leftarrow x\mathbf{V}_{1\dots d}$
- 9: $y \leftarrow \mathbf{B}y\mathbf{T}$
- 10: **return** y

Discussion & Results

In this section we provide both visual and quantitative evaluation of our method. We begin by defining an embedding error which can be used to analyse the performance of an out-of-sample extension algorithm. We then move on to discuss how GOOSE's only parameter, k , affects the accuracy of the estimated low-dimensional embedding before finally displaying results using both artificial data as well as real world image data.

Embedding Error

To be able to analyse the performance of out-of-sample extensions we need to first define an embedding error. Given a dataset \mathbf{D} we create a training set, \mathbf{B} , and test set, \mathbf{C} , such that $\mathbf{B} \cup \mathbf{C} = \mathbf{D}$, $\mathbf{B} \cap \mathbf{C} = \emptyset$ and $|\mathbf{B}| = |\mathbf{D}| - |\mathbf{C}|$. As in (Yang et al. 2010) we can obtain the low-dimensional embedding, \mathbf{Y} , by running a manifold learning algorithm on the entire dataset \mathbf{D} . We can then express \mathbf{Y} as $\mathbf{Y} = [\mathbf{Y}_{\text{train}}, \mathbf{Y}_{\text{test}}]^T$ where $\mathbf{Y}_{\text{train}}$ and \mathbf{Y}_{test} are the low-dimensional embeddings of the training and test data. Once \mathbf{Y} is known we can use \mathbf{B} to obtain the training set of the manifold and then use an out-of-sample extension method to estimate the low-dimensional embedding of \mathbf{C} . We denote the estimated low-dimensional embedding of the test data $\mathbf{Y}'_{\text{test}}$, we can now define an embedding error based on the root mean square error between the actual and estimated test sets

$$e = \sqrt{\frac{\sum (\mathbf{Y}_{\text{test}} - \mathbf{Y}'_{\text{test}})^2}{n}} \quad (10)$$

where n is the number of elements in the test set and both \mathbf{Y}_{test} and $\mathbf{Y}'_{\text{test}}$ are transformed according to the rotational difference between $\mathbf{Y}_{\text{train}}$ and \mathbf{B} to remove the effect of the manifold learning algorithms mapping the datasets into different low-dimensional spaces¹. This error measure now

¹This is something that is not considered by Yang et al. in (Yang et al. 2010) but without this step the results obtained are meaningless as the two low-dimensional embeddings are in different coordinate spaces

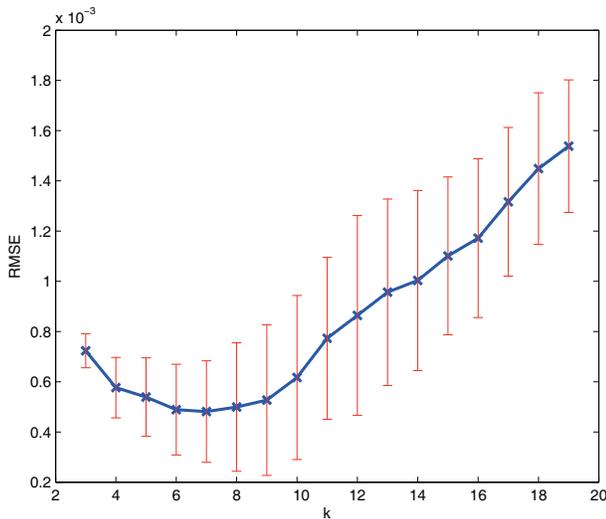


Figure 3: The effect of the neighborhood size parameter k on the embedding error of a dataset with a known low-dimensional manifold.

provides us with a basis of analysing the performance of an out-of-sample extension method, with a low value of e signifying that the estimated test embedding is closer to the actual test embedding than that of a test embedding with a high value of e .

Parameter Selection

Our algorithm has only one free parameter, the neighborhood size k . To test how this parameter affects the performance we ran a set of experiments on a known manifold with a known low-dimensional embedding. We used the Swiss Roll manifold with 2000 samples and the low-dimensional embedding learnt by LTSA (we could have used any manifold learning algorithm but LTSA produces the most faithful result as shown in Figure 4). The data was randomly split into training and test sets with each set having a size of 1000. For each permutation of training and test we used the GOoSE algorithm to try and embed the test set into the low-dimensional space with varying parameters of k within the range [3, 19]. The RMSE of the test data for each value of k was recorded and averaged over a series of 10 runs.

Figure 3 shows the results of this test. The graph is shown with associated error bars indicating the standard deviation of the results per value of k . The results show that a minima is reached around $k = 7 \pm 2$, after this point the RMSE increases along with the standard deviation meaning that results obtained with a larger value of k are more unstable. Although this optimum value of k will change depending on what dataset is used, experiments do show that a local minima will always exist. Since the GOoSE algorithm is fast to run it is easy to find an optimum value of k by performing a simple parameter search.

Results

To test our algorithm we use 3 main datasets: a 3-dimensional Swiss Roll, a moving image dataset and the ISOMAP faces data. Each of these datasets presents a different challenge for a manifold learning algorithm and subsequently an out-of-sample extension algorithm.

Swiss Roll The Swiss Roll dataset consists of a 2-dimensional manifold embedded within \mathbb{R}^3 . This 2-dimensional manifold is a highly-curved plane that is rolled up to resemble a Swiss Roll (Figure 4). A manifold learning algorithm should be able to ‘unwrap’ this Swiss Roll and embed it into \mathbb{R}^2 . We used 2000 points sampled from the Swiss Roll and this was randomly split into 1000 samples for training and 1000 samples for test. We used four different manifold learning algorithms (LLE (Roweis and Saul 2000), LTSA (Zhang and Zha 2005), Eigenmaps (Belkin and Niyogi 2003) and LGRM (Yang et al. 2010)) to learn the low-dimensional training embedding before applying GOoSE to estimate the test set’s low-dimensional embedding. These algorithms were chosen due to the fact that they all either inherently contain, or have been extended to cope with, the out-of-sample extension problem. For LLE, LTSA and Eigenmaps the neighborhood size parameter was set to 8 and for LGRM we used the parameters shown in (Yang et al. 2010). The results of running our algorithm on the Swiss Roll dataset using the GOoSE k parameter of $k = 7$ are shown in Figure 4. The top row shows the 1000 training samples and the bottom row shows the results of running GOoSE on the test samples. In all cases GOoSE is able to embed the novel samples within the trained manifold to obtain a meaningful embedding of the test set. It is worth noting that the failure of Laplacian Eigenmaps, and to some extent LLE, to produce meaningful low-dimensional embeddings is due to the fact that the problem is under-sampled. As such these techniques are unable to build an adequate model of the manifold from the training set leading to an incorrect low-dimensional embedding. However, this does enable us to show that even in the case of a distorted embedding GOoSE is able to embed novel samples according to the shape of the trained embedding.

To obtain quantitative analysis of our algorithm and to compare it against existing approaches we measured the embedding error of our algorithm using a 10 fold cross validation approach. The data was randomly split into 10 folds with 9 being used for training and 1 for test. This was repeated until all folds had been used as a test set. A manifold learning algorithm was then used to obtain the full low-dimensional embedding as well as the training set’s low-dimensional embedding. For each run the RMSE was recorded when using GOoSE and also when using the given manifold learning algorithm’s out-of-sample extension. For LLE and LTSA we used the out-of-sample approach outlined in (Saul and Roweis 2003); for Eigenmaps we used the approach outlined in (Bengio et al. 2003) and we used LGRM’s built in approach (Yang et al. 2010). Thus for each run of the experiment using a given manifold learning algorithm we obtain two different error scores: the error obtained from using the out-of-sample extension associated with the given algorithm

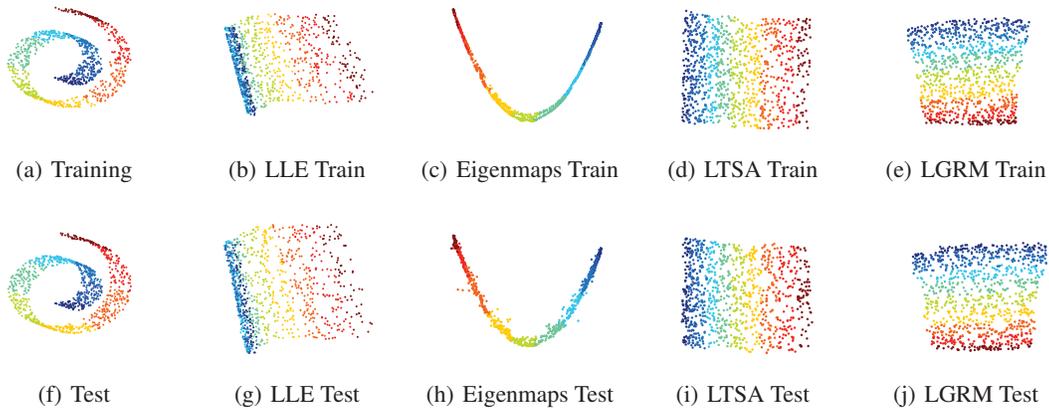


Figure 4: Results of running the GOOSE algorithm on the Swiss Roll data with 1000 samples used from training and 1000 samples used for test. GOOSE’s k parameter was set to $k = 7$. The neighborhood size parameter for LLE, LTSA and Eigenmaps was set to 8 while the parameters for LGRM were set according to (Yang et al. 2010).

(the default method) and the error obtained from running GOOSE as the out-of-sample method. For each test GOOSE was run multiple times with differing values of k and the minimum RMSE was taken. The averaged results are shown in the graph in Figure 5. When compared with existing out-of-sample approaches our algorithm is able to consistently outperform current methods. The average RMSE across all methods for GOOSE is $e = 0.0002$ when compared with $e = 0.0043$ for using the algorithms’ built in out-of-sample methods. There is also large variation in the performance of existing out-of-sample methods, $\sigma = 0.0044$, with LLE performing the worst ($e = 0.010$) and LGRM performing the best ($e = 0.0008$). The variation between different manifold learning algorithms when using GOOSE is considerably lower, $\sigma = 0.0001$. This shows the stability of GOOSE and its effectiveness regardless of what manifold learning algorithm is used to learn the low-dimensional embedding.

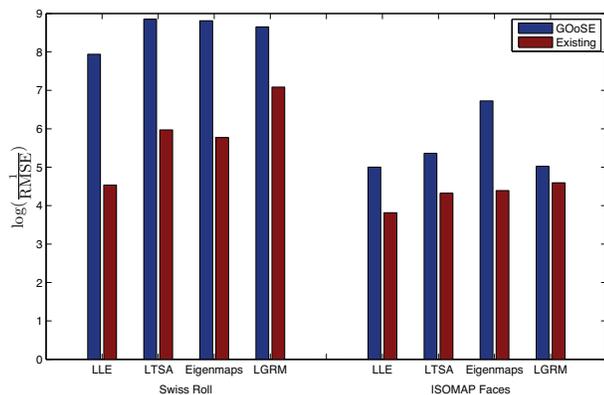


Figure 5: Average embedding quality for each of the out-of-sample extension algorithms on the Swiss Roll and ISOMAP faces datasets. Due to the large range in results the plot is shown with the Y axis scaled as $\log(\frac{1}{RMSE})$, meaning a larger value indicates a better quality embedding.

Moving Image To test our algorithm on image data we use two different datasets the first of which consists of an image moving across a black background. The dataset contains 4096 images of size 96×96 pixels and so the high-dimensional data lies in \mathbb{R}^{9216} . The training data consists of 2048 randomly selected samples and the remaining samples are used as test. Local Tangent Space Alignment (Zhang and Zha 2005) with parameter $k = 8$ was used to learn the low-dimensional embedding of the training set as it was able to find a meaningful low-dimensional embedding of the data. Figure 6 shows the resulting 2-dimensional embedding with the training data indicated by blue dots (\bullet) and the test data indicated by a red plus-signs ($+$). As can be seen the test samples fit nicely into the ‘gaps’ of the training data as would be expected. The data consists of dense regions of samples around the corners and a sparser region of samples in the center (this is due to the manifold being curved at the edges and so is not truly 2-dimensional). Even with the more sparsely sampled central region GOOSE manages to place the unlearned samples into the correct regions.

ISOMAP Faces The second image dataset used is the ISOMAP faces dataset (Tenenbaum, de Silva, and Langford 2000) consisting of a set of 698 faces in \mathbb{R}^{4096} under different pose and illumination conditions. This dataset is interesting as it has intrinsic dimensionality of 4 (Kégl 2002) meaning the quality of results are not visually assessable. Therefore our algorithm’s performance on this dataset along with the performance of other out-of-sample methods is shown in Figure 5. The neighborhood size parameters for LLE, LTSA and Eigenmaps were set to 8 while the parameters for LGRM were set according to (Yang et al. 2010). GOOSE was run multiple times with differing values of k and the minimum RMSE was taken. As with the Swiss Roll dataset we used a 10 fold cross validation approach and averaged the results (the full details are described in the Swiss Roll section above). Again GOOSE is able to outperform existing out-of-sample techniques. The average embedding error for GOOSE on the ISOMAP faces data is $e = 0.0048$ with

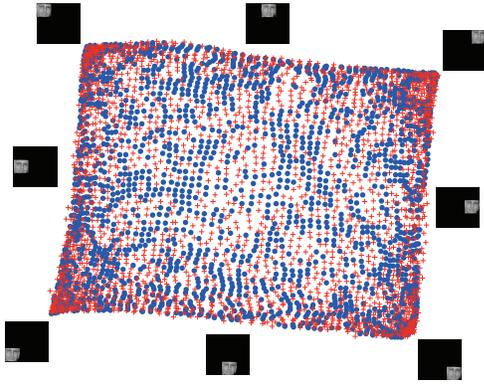


Figure 6: Result of running the GOoSE algorithm on high-dimensional image data. The data consists of 4096 images of size 92×92 split randomly using half as training and half as test. The blue dots represent the training samples and the red plus-signs represent the samples learnt using GOoSE.

standard deviation of $\sigma = 0.0026$. For the existing out-of-sample methods the average error is $e = 0.0144$ with standard deviation of $\sigma = 0.0052$. Although the standard deviation of the results from the GOoSE algorithm on this dataset is higher it is still able to consistently out perform existing out-of-sample methods.

Conclusions & Future Work

We have presented a novel and simple technique to solve the generalized out-of-sample extension problem in manifold learning. Our algorithm, GOoSE, applies the local geometric change between neighborhoods in the high and low-dimensional space to any unlearned sample to obtain its low-dimensional embedding. The method works by learning the transformation that maps the neighborhood of the unlearned sample from the high to the low-dimensional space. This transformation is then applied to the new sample to obtain an estimation of its low-dimensional embedding.

The results show that this method is able to successfully embed new datapoints into non-linear manifolds. We have shown that the GOoSE algorithm is able to embed new samples into previously learnt manifolds regardless of the manifold learning technique used. GOoSE also significantly outperforms existing out-of-sample techniques when tested on artificial and real world data. This makes GOoSE a powerful and versatile tool for statistical learning as it is independent of the manifold learning technique used and only requires access to the original data and the learnt low-dimensional embedding.

We are currently working on a version of the GOoSE algorithm that reverses the out of sample process, that is it aims to solve the pre-image problem (given a sample in the low-dimensional space we wish to find its high-dimensional image). Using similar methodology outlined in this paper we aim to produce a pre-image algorithm that can be combined with the GOoSE algorithm to form a framework for easily mapping between the high and low-dimensional spaces.

References

- Belkin, M., and Niyogi, P. 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation* 15(6):1373–1396.
- Bengio, Y.; Païement, J.-F.; Vincent, P.; Delalleau, O.; Roux, N. L.; and Ouimet, M. 2003. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. In *Advances in Neural Information Processing Systems 16, NIPS 2003*.
- Chin, T.-J., and Suter, D. 2008. Out-of-sample extrapolation of learned manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30:1547–1556.
- Cox, T. F., and Cox, M. 2001. *Multidimensional Scaling*. Chapman and Hall.
- Hotelling, H. 1933. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology* (24):417–441,498–520.
- Kégl, B. 2002. Intrinsic dimension estimation using packing numbers. In *Advances in Neural Information Processing Systems 15, NIPS 2002*.
- Law, M. H., and Jain, A. K. 2006. Incremental nonlinear dimensionality reduction by manifold learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28:377–391.
- Patwari, N.; III, A. O. H.; and Pacholski, A. 2005. Manifold learning visualization of network traffic data. In *Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data, MineNet '05*, 191–196.
- Roweis, S. T., and Saul, L. K. 2000. Nonlinear dimensionality reduction by locally linear embedding. *SCIENCE* 290:2323–2326.
- Saul, L. K., and Roweis, S. T. 2003. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *J. Mach. Learn. Res.* 4:119–155.
- Strange, H., and Zwiggelaar, R. 2009. Classification performance related to intrinsic dimensionality in mammographic image analysis. In *Proceedings of the Thirteenth Annual Conference on Medical Image Understanding and Analysis*, 219–223.
- Tenenbaum, J. B.; de Silva, V.; and Langford, J. C. 2000. A global geometric framework for nonlinear dimensionality reduction. *SCIENCE* 290:2319–2323.
- Verbeek, J. 2006. Learning nonlinear image manifolds by global alignment of local linear models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(8):1236–1250.
- Weinberger, K. Q., and Saul, L. K. 2006. An introduction to nonlinear dimensionality reduction by maximum variance unfolding. In *proceedings of the 21st national conference on Artificial intelligence - Volume 2*, 1683–1686.
- Yang, Y.; Nie, F.; Xiang, S.; Zhuang, Y.; and Wang, W. 2010. Local and global regressive mapping for manifold learning with out-of-sample extrapolation. In *Twenty-Fifth American Association for Artificial Intelligence Conference 2010, (AAAI-2010)*.
- Zhang, Z., and Zha, H. 2005. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM J. Sci. Comput.* 26:313–338.