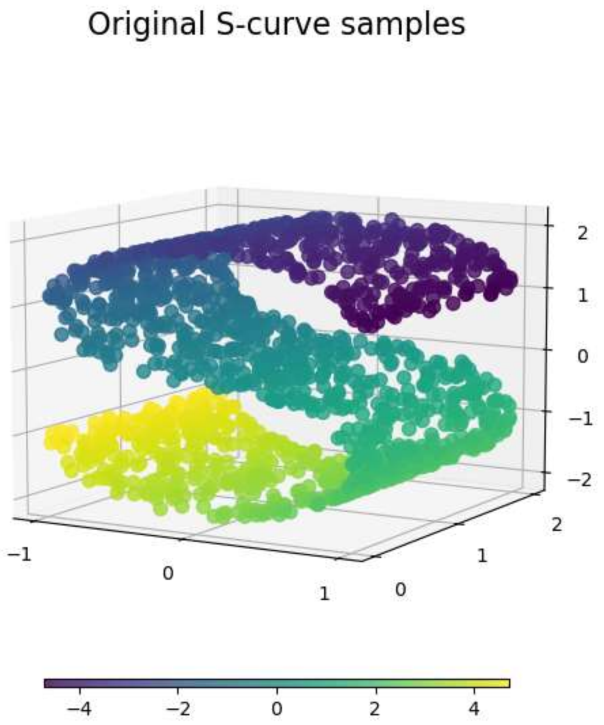
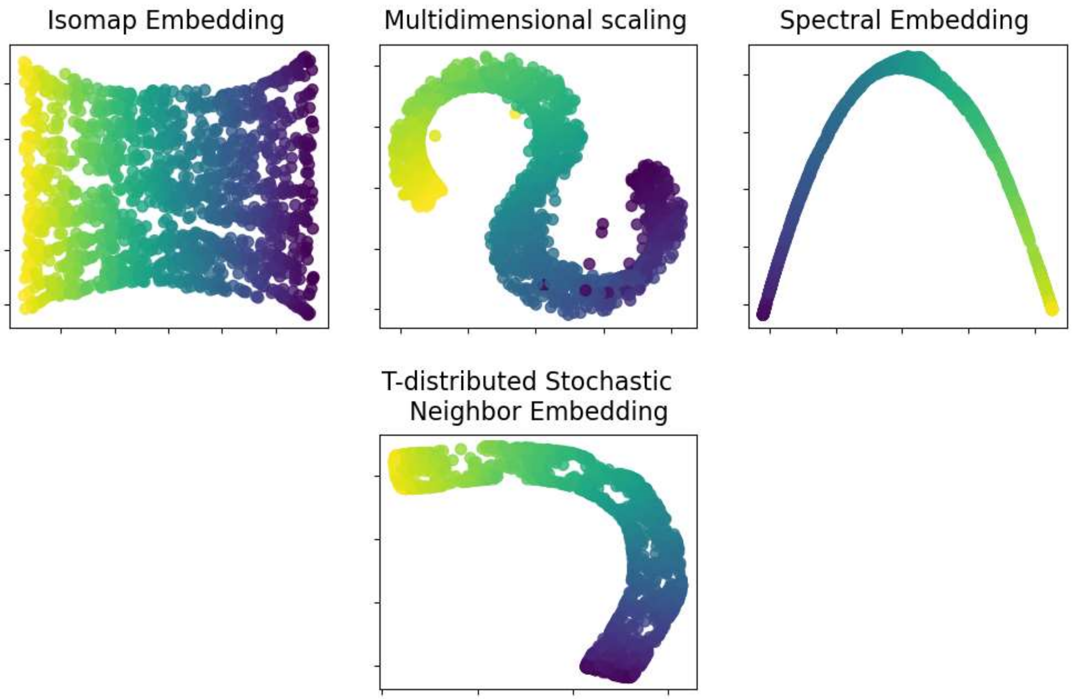
2.2. Manifold learning

Look for the bare necessities
The simple bare necessities
Forget about your worries and your strife
I mean the bare necessities
Old Mother Nature’s recipes
That bring the bare necessities of life

– Baloo’s song [The Jungle Book]



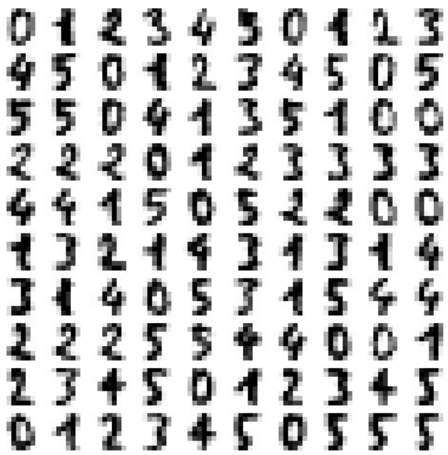


Manifold learning is an approach to non-linear dimensionality reduction. Algorithms for this task a based on the idea that the dimensionality of many data sets is only artificially high.

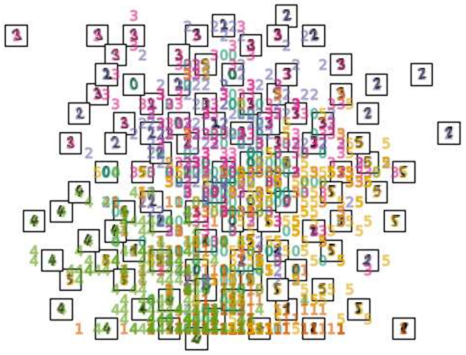
2.2.1. Introduction

High-dimensional datasets can be very difficult to visualize. While data in two or three dimensions can be plotted to show the inherent structure of the data, equivalent high-dimensional plots are much less intuitive. To aid visualization of the structure of a dataset, the dimension must be reduc in some way.

The simplest way to accomplish this dimensionality reduction is by taking a random projection of the data. Though this allows some degree of visualization of the data structure, the randomness o the choice leaves much to be desired. In a random projection, it is likely that the more interesting structure within the data will be lost.

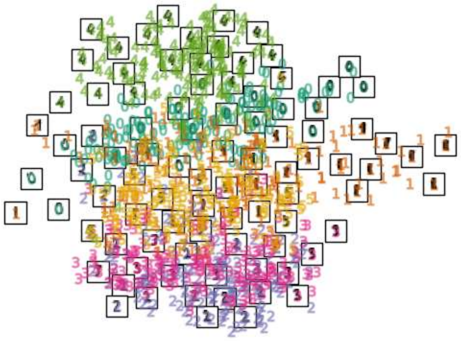


A selection from the 64-dimensional digits dataset

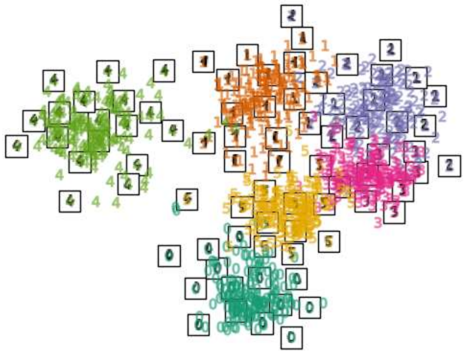


Random projection embedding (time 0.002s)

To address this concern, a number of supervised and unsupervised linear dimensionality reduction frameworks have been designed, such as Principal Component Analysis (PCA), Independent Component Analysis, Linear Discriminant Analysis, and others. These algorithms define specific rubrics to choose an “interesting” linear projection of the data. These methods can be powerful, b often miss important non-linear structure in the data.



Truncated SVD embedding (time 0.003s)



Linear Discriminant Analysis embedding (time 0.007s)

Manifold Learning can be thought of as an attempt to generalize linear frameworks like PCA to be sensitive to non-linear structure in data. Though supervised variants exist, the typical manifold learning problem is unsupervised: it learns the high-dimensional structure of the data from the da itself, without the use of predetermined classifications.

Examples

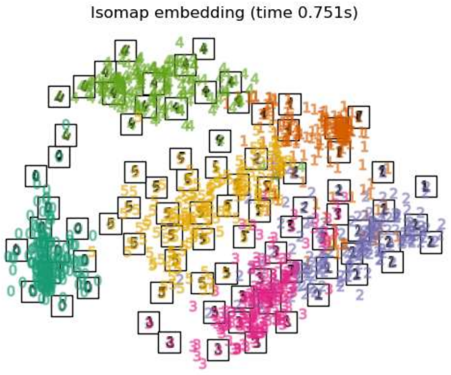
- See Manifold learning on handwritten digits: Locally Linear Embedding, Isomap… for an example of dimensionality reduction on handwritten digits.

- See Comparison of Manifold Learning methods for an example of dimensionality reduction o toy “S-curve” dataset.

The manifold learning implementations available in scikit-learn are summarized below

2.2.2. Isomap

One of the earliest approaches to manifold learning is the Isomap algorithm, short for Isometric Mapping. Isomap can be viewed as an extension of Multi-dimensional Scaling (MDS) or Kernel PC Isomap seeks a lower-dimensional embedding which maintains geodesic distances between all points. Isomap can be performed with the object `Isomap` .



Complexity

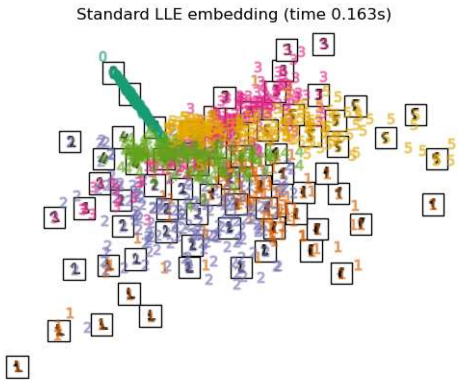
References

- “A global geometric framework for nonlinear dimensionality reduction” Tenenbaum, J.B.; De Silva, V.; & Langford, J.C. Science 290 (5500)

2.2.3. Locally Linear Embedding

Locally linear embedding (LLE) seeks a lower-dimensional projection of the data which preserves distances within local neighborhoods. It can be thought of as a series of local Principal Componen Analyses which are globally compared to find the best non-linear embedding.

Locally linear embedding can be performed with function `locally_linear_embedding` or its object oriented counterpart `LocallyLinearEmbedding` .



Complexity

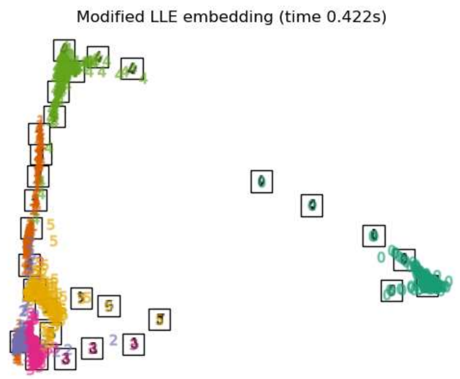
References

- “Nonlinear dimensionality reduction by locally linear embedding” Roweis, S. & Saul, L. Science 290:2323 (2000)

2.2.4. Modified Locally Linear Embedding

One well-known issue with LLE is the regularization problem. When the number of neighbors is greater than the number of input dimensions, the matrix defining each local neighborhood is rank deficient. To address this, standard LLE applies an arbitrary regularization parameter r, which is chosen relative to the trace of the local weight matrix. Though it can be shown formally that as $r \to 0$, the solution converges to the desired embedding, there is no guarantee that the optimal solution will be found for $r > 0$. This problem manifests itself in embeddings which distort the underlying geometry of the manifold.

One method to address the regularization problem is to use multiple weight vectors in each neighborhood. This is the essence of *modified locally linear embedding* (MLLE). MLLE can be performed with function `locally_linear_embedding` or its object-oriented counterpart `LocallyLinearEmbedding` , with the keyword `method = 'modified'` . It requires `n_neighbors > n_components` .



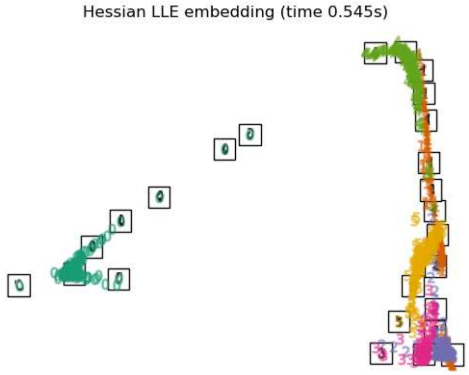
Complexity

References

- “MLLE: Modified Locally Linear Embedding Using Multiple Weights” Zhang, Z. & Wang, J.

2.2.5. Hessian Eigenmapping

Hessian Eigenmapping (also known as Hessian-based LLE: HLLE) is another method of solving the regularization problem of LLE. It revolves around a hessian-based quadratic form at each neighborhood which is used to recover the locally linear structure. Though other implementations note its poor scaling with data size, `sklearn` implements some algorithmic improvements which make its cost comparable to that of other LLE variants for small output dimension. HLLE can be performed with function `locally_linear_embedding` or its object-oriented counterpart `LocallyLinearEmbedding`, with the keyword `method = 'hessian'`. It requires `n_neighbors > n_components * (n_components + 3) / 2`.



Complexity

The HLLE algorithm comprises three stages:

> 1. **Nearest Neighbors Search**. Same as standard LLE
> 2. **Weight Matrix Construction**. Approximately $O[DNk^3] + O[Nd^6]$. The first term reflects a similar cost to that of standard LLE. The second term comes from a QR decomposition of the local hessian estimator.
> 3. **Partial Eigenvalue Decomposition**. Same as standard LLE
>
> The overall complexity of standard HLLE is $O[D \log(k) N \log(N)] + O[DNk^3] + O[Nd^6] + O[dN^2]$.
>
> - N : number of training data points
> - D : input dimension
> - k : number of nearest neighbors
> - d : output dimension

References

- "Hessian Eigenmaps: Locally linear embedding techniques for high-dimensional data" Donoh D. & Grimes, C. Proc Natl Acad Sci USA. 100:5591 (2003)

2.2.6. Spectral Embedding

Spectral Embedding is an approach to calculating a non-linear embedding. Scikit-learn implement Laplacian Eigenmaps, which finds a low dimensional representation of the data using a spectral decomposition of the graph Laplacian. The graph generated can be considered as a discrete approximation of the low dimensional manifold in the high dimensional space. Minimization of a cost function based on the graph ensures that points close to each other on the manifold are mapped close to each other in the low dimensional space, preserving local distances. Spectral embedding can be performed with the function `spectral_embedding` or its object-oriented counterpart `SpectralEmbedding` .

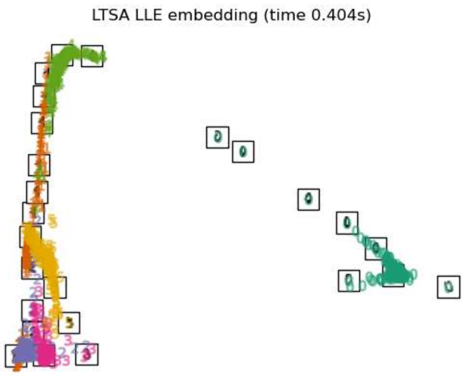
Complexity

References

- “Laplacian Eigenmaps for Dimensionality Reduction and Data Representation” M. Belkin, P. Niyogi, Neural Computation, June 2003; 15 (6):1373-1396

2.2.7. Local Tangent Space Alignment

Though not technically a variant of LLE, Local tangent space alignment (LTSA) is algorithmically similar enough to LLE that it can be put in this category. Rather than focusing on preserving neighborhood distances as in LLE, LTSA seeks to characterize the local geometry at each neighborhood via its tangent space, and performs a global optimization to align these local tange spaces to learn the embedding. LTSA can be performed with function `locally_linear_embedding` its object-oriented counterpart `LocallyLinearEmbedding` , with the keyword `method = 'ltsa'` .



Complexity

References

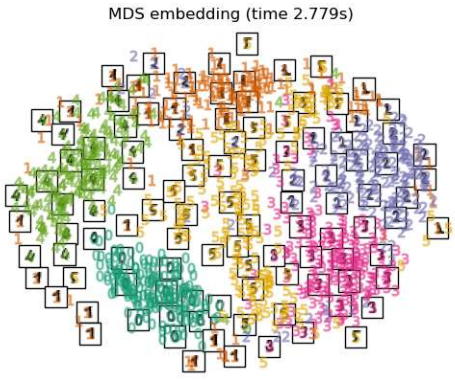
- “Principal manifolds and nonlinear dimensionality reduction via tangent space alignment” Zhang, Z. & Zha, H. Journal of Shanghai Univ. 8:406 (2004)

2.2.8. Multi-dimensional Scaling (MDS)

Multidimensional scaling (`MDS`) seeks a low-dimensional representation of the data in which the distances respect well the distances in the original high-dimensional space.

In general, `MDS` is a technique used for analyzing similarity or dissimilarity data. It attempts to mod similarity or dissimilarity data as distances in a geometric spaces. The data can be ratings of similarity between objects, interaction frequencies of molecules, or trade indices between countrie

There exists two types of MDS algorithm: metric and non metric. In scikit-learn, the class `MDS` implements both. In Metric MDS, the input similarity matrix arises from a metric (and thus respect the triangular inequality), the distances between output two points are then set to be as close as possible to the similarity or dissimilarity data. In the non-metric version, the algorithms will try to preserve the order of the distances, and hence seek for a monotonic relationship between the distances in the embedded space and the similarities/dissimilarities.


MDS embedding (time 2.779s)

Let S be the similarity matrix, and X the coordinates of the n input points. Disparities $\hat{d}_{ij}$ are transformation of the similarities chosen in some optimal ways. The objective, called the stress, is then defined by $\sum_{i<j} d_{ij}(X) - \hat{d}_{ij}(X)$

Metric MDS

Nonmetric MDS

References

- "Modern Multidimensional Scaling - Theory and Applications" Borg, I.; Groenen P. Springer Series in Statistics (1997)
- "Nonmetric multidimensional scaling: a numerical method" Kruskal, J. Psychometrika, 29 (196
- "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis" Kruskal, J Psychometrika, 29, (1964)

2.2.9. t-distributed Stochastic Neighbor Embedding (t-SNE)

t-SNE (TSNE) converts affinities of data points to probabilities. The affinities in the original space a represented by Gaussian joint probabilities and the affinities in the embedded space are represent by Student's t-distributions. This allows t-SNE to be particularly sensitive to local structure and has few other advantages over existing techniques:

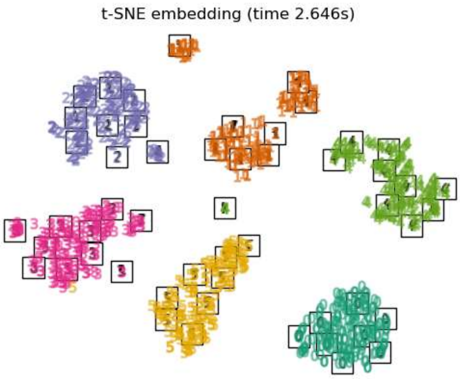
- Revealing the structure at many scales on a single map
- Revealing data that lie in multiple, different, manifolds or clusters
- Reducing the tendency to crowd points together at the center

While Isomap, LLE and variants are best suited to unfold a single continuous low dimensional manifold, t-SNE will focus on the local structure of the data and will tend to extract clustered local groups of samples as highlighted on the S-curve example. This ability to group samples based on the local structure might be beneficial to visually disentangle a dataset that comprises several manifolds at once as is the case in the digits dataset.

The Kullback-Leibler (KL) divergence of the joint probabilities in the original space and the embedded space will be minimized by gradient descent. Note that the KL divergence is not conve i.e. multiple restarts with different initializations will end up in local minima of the KL divergence. Hence, it is sometimes useful to try different seeds and select the embedding with the lowest KL divergence.

The disadvantages to using t-SNE are roughly:

- t-SNE is computationally expensive, and can take several hours on million-sample datasets where PCA will finish in seconds or minutes
- The Barnes-Hut t-SNE method is limited to two or three dimensional embeddings.
- The algorithm is stochastic and multiple restarts with different seeds can yield different embeddings. However, it is perfectly legitimate to pick the embedding with the least error.
- Global structure is not explicitly preserved. This problem is mitigated by initializing points witl PCA (using `init='pca'`).



Optimizing t-SNE

Barnes-Hut t-SNE

References

- “Visualizing High-Dimensional Data Using t-SNE” van der Maaten, L.J.P.; Hinton, G. Journal of Machine Learning Research (2008)
- “t-Distributed Stochastic Neighbor Embedding” van der Maaten, L.J.P.
- “Accelerating t-SNE using Tree-Based Algorithms” van der Maaten, L.J.P.; Journal of Machine Learning Research 15(Oct):3221-3245, 2014.
- “Automated optimized parameters for T-distributed stochastic neighbor embedding improve visualization and analysis of large datasets” Belkina, A.C., Ciccolella, C.O., Anno, R., Halpert, R., Spidlen, J., Snyder-Cappione, J.E., Nature Communications 10, 5415 (2019).

2.2.10. Tips on practical use

Make sure the same scale is used over all features. Because manifold learning methods are