Graph Embedding and Extensions: A General Framework for Dimensionality Reduction

Yale Chang
Department of ECE, Northeastern University

Abstract

Dimensionality reduction forms a cornerstone of data analysis. A variety of techniques have been proposed to solve this problem. This report first presents a review of these techniques based on their respective motivations, assumptions, strengths and weaknesses. The general framework for dimensionality reduction through graph embedding and its extensions is also assessed by examing assumptions behind the constructions of various similarity measures of different algorithms. Based on the insight of this framework, we first present some heuristics on how to design similarity matrix corresponding to certain data distribution. Then a new algorithm is proposed to overcome the sensitivity to noisy dimensions of nonlinear dimensionality reduction algorithms.

1 Introduction

Dimensionality reduction aims to transform high-dimensional data into a desired low-dimensional representation [1]. We expect the reduced representation should correspond to the intrinsic dimeniona-lity of data, which is the minimum number of parameters needed to account for the observed properties of the data [2]. There exist a variety of dimensionality reduction algorithms [3], which can be categorized according to different perspectives such as supervised vs. unsupervised, linear vs. nonlinear, global vs. local.

Classical methods such as Principle Component Analysis(PCA), Linear Discriminat Analysis(LDA), Multidimensional Scaling(MDS) [4] are linear models. The low-dimensional representations returned by these methods are related to the input feature vectors by a linear transformation. If the data lies on or near a low dimensional subspace, linear methods can recover intrinsic dimensionality. Therefore, "subspace learning" is a subfield of dimensionality reduction with linearity assumption.

However, linear methods will fail if the data lies on a low-dimensional manifold because the data structure becomes highly nonlinear [5]. The degrees of freedom along the sub-manifold corresponds to the intrsinsic dimensionality. In this form, the nonlinear dimensionality reduction(NLDR) is known as "manifold learning" [6]. More generally, manifold learning can be defined as a process that automatically learns the geometric and topological properties of a given manifold [7]. Classical manifold learning algorithms include Isometric Feature Mapping(Isomap) [8], Locally Linear Embedding(LLE) [9, 10], Laplacian Eigenmaps [11].

An alternative for NLDR is to generalize linear methods to their corresponding nonlinear settings by employing "kernel trick" [12]. For example, the nonlinear generalization of PCA can be obtained by replacing dot product in feature space with kernel function, which implicitly maps data from the feature space to high-dimensional Hilbert space and then apply linear method in the new space [13].

Subspace learning, manifold learning and kernel methods can be unified by expressing them in a common framework for dimensionality reduction. Ham et al. [14] show Isomap,LLE and Laplaican Eigenmaps can be described as kernel PCA on specially constructed Gram matrices. Brand [15] equates NLDR to graph embedding with side information. Bengio et al. [16] show a direct equivalence between spectral clustering and kernel PCA, and how both are special cases of a more general learning problem. Based on these previous works, Yan et al. [1] present a general formulation based on graph embedding to unify various dimensionality reduction algorithms.

The outline of the remainder of this report is as follows. In section 2, we review subspace learning algorithms(PCA,LDA),manifold learning algorithms (Isomap,LLE,Laplacian Eigenmaps) and ker-

nel methods(kernel PCA). We will focus on their motivations,assumptions, strengths and weakness. Therefore, algorithms are mostly explained through intuitions and geometric illustrations rather than mathematical formulations. In section 3, we assess the general formulation and the newly proposed algorithm presented in [1]. We also discuss the related formulations in [14–16]. In section 4, we discuss the application of dimensionality reduction to face recognition. In section 5, based on the insight from previous work, we first present some heuristics on how to choose appropriate similarity measure based on data distribution. Then we propose a new algorithm to overcome the sensitivity to noisy dimensions of NLDR algorithms. In section 6, we present our main conclusions.

2 Background

Given a dataset $X = \{x_1, x_2, \cdots, x_N\}, x_i \in \mathbb{R}^D$, where N is the number of samples and D is the number of features, dimensionality reduction can be defined as finding a mapping function $F : x \rightarrow y$ that transforms $x \in \mathbb{R}^D$ into the desired low-dimensional representation $y \in \mathbb{R}^d$. The motivation of dimensionality reduction is to recover intrinsic dimensionality of the data. To better understand the concept of intrinsic dimensionality, consider each human face image in Figure 1 as a sample. The number of original dimension D should be equal to the number of pixels in each image. As is pointed out in [17], although D might be quite high, the set $\mathcal{M}$ of all facial images generated by varying the orientation of a face is a continuous curve in a D-dimensional image space. It is continuous because the image varies smoothly as the face is rotated. It is a curve because it is generated by varying a single degree of freedom, the angle of rotation. Therefore, $\mathcal{M}$ is a one-dimensional manifold in D-dimensional image space. The dimensionality of $\mathcal{M}$ would increase if we were to allow other types of image transformation such as scaling and translation. However, the dimensionality would still be far more less than D. This example illustrates the motivation to conduct dimensionality reduction for human face recognition. Subspace learning assumes intrinsic dimensionality can be obtained by linear projection. While manifold learning assumes intrinsic structure is a manifold. Kernel methods can generalize subspace learning algorithms to nonlinear settings. In the following, we will examine these approaches in detail.

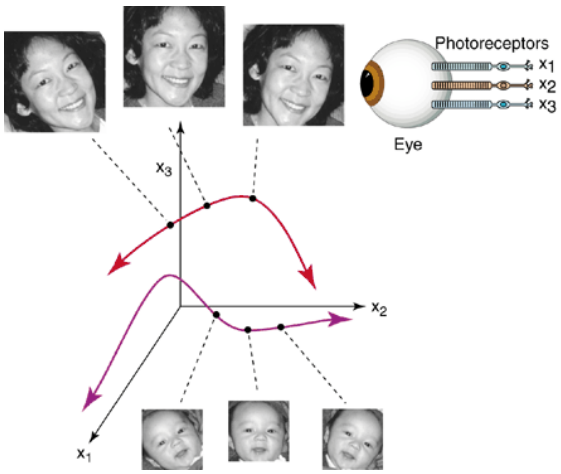


Figure 1: Manifold of Visual Perception [17]

2.1 Subspace Learning

2.1.1 PCA

PCA is based on computing the low-dimensional representation that maximize data variance. In mathematical terms, we first find the linear transformation matrix $\hat{W} \in \mathbb{R}^{D \times d}$ by solving

$$\hat{W} = \underset{W^T W = I}{\arg\max} \operatorname{Tr}\left(W^T \operatorname{Cov}(X) W\right) \tag{1}$$

Besides linearity assumption, PCA assumes principal components with larger associated variances represent interesting structure while those with lower variances represent noise. PCA also assumes the principal components are orthogonal, which makes PCA soluble with eigendecomposition techniques. These are strong, and sometimes incorrect assumptions [18].

2.1.2 LDA

LDA assumes the distribution of each class is Gaussian and the linear projection matrix $\hat{W} \in \mathbb{R}^{D \times d}$ is obtained by maxmizing the ratio between the interclass scatter S_B and intraclass scatters S_W.

$$\hat{W} = \arg\max \operatorname{Tr}\left\{(W^T S_W W)^{-1}(W^T S_B W)\right\} \tag{2}$$

Both problems can be solved by eigendecomposition.The number of low dimension d can be obtained by detecting a prominent gap in the eigenvalue spectrum [5]. The low-dimensional representation $Y = W^T X$. The comparison between PCA

and LDA is shown in Figure 2, the data in two dimensions belonging to two classes shown in red and blue is projected onto a single dimension. PCA chooses the direction of maximum variance, shown by the magenta curve, which leads to strong overlap while LDA takes into account of the class labels and leads to a projection on the green curve giving much better class separation.

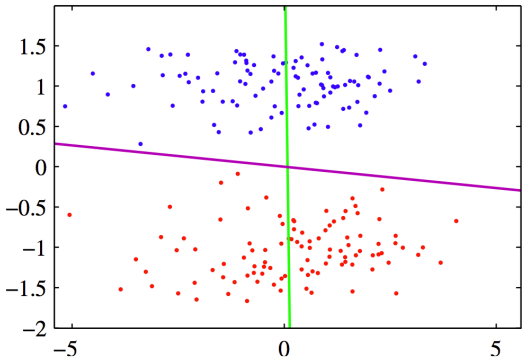


Figure 2: LDA vs PCA [19]

2.1.3 Discussion

Advantages

- Both PCA and LDA are nonparametric, meaning they require no free parameters to set.
- They are computationally more efficient when data lies on a linear subspace and $D < N$.

Disadvantages

- Since PCA, LDA are linear models, they will fail when the true underlying structure of the data is on a nonlinear manifold, such as the "Swiss Roll" shown in Figure 3.
- The nonparametric nature of PCA,LDA can also be seen as a weakness because they assume data distributions are Gaussian, making the algorithm sensitive to outliers.
- Another weakness of LDA is the number of available projection directions is lower than the class number.

2.2 Manifold Learning

2.2.1 Isomap

Isomap finds low-dimensional representation that most faithfully preserve the pairwise geodesic distances between feature vectors in all scales as measured along the submanifold from which they were sampled. As is shown in Figure 3, the left figure shows Euclidean distance between samples A, B while the right figure shows geodesic distance. There are three steps for Isomap:

1. Construct neighborhood graph on the manifold.
2. Compute the shortest path between pairwise points.
3. Construct low-dimensional embedding by applying MDS [4]

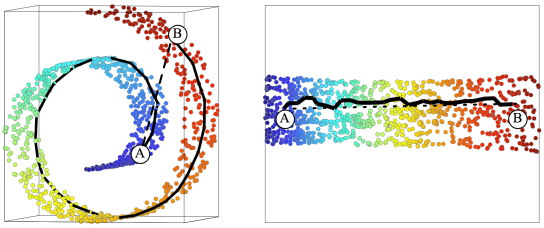


Figure 3: Swiss Roll Dataset [5]

2.2.2 LLE

LLE preserve the local linear structure of nearby feature vectors. As is shown in Figure 4, there are three steps for LLE:

1. Assign neighbors to each data point.
2. Compute the weights W_{ij} that best linearly reconstruct X_i from its neighbors.
3. Compute the low-dimensional embedding Y_i best reconstructed by W_{ij}.

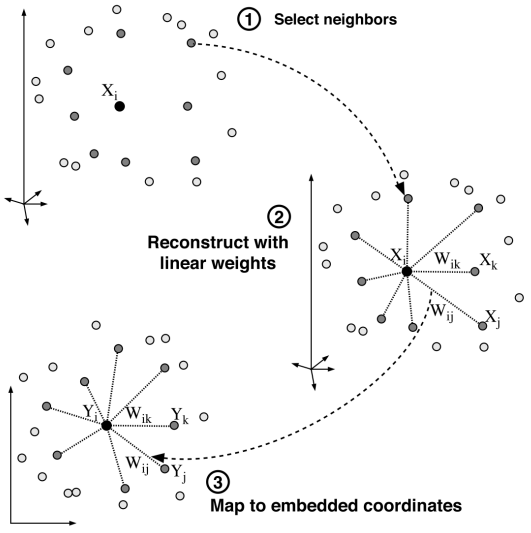


Figure 4: LLE [9]

2.2.3 Laplacian Eigenmaps

Lapalacian Eigenmaps finds the low-dimensional representation that most faithfully preserves local similarity structure in feature space. There are three steps for Laplacian Eigenmaps:

1. Construct neighborhood graph through ϵ-neighorhoods or n nearest neighbors.
2. Choose edge weights using Heat kernel or simply set edge weight to be 1 if connected and 0 otherwise, then we can get weight matrix W.
3. Solve the generalized eigenvector problem

$$L\mathbf{f} = \lambda D\mathbf{f} \tag{3}$$

where D is diagonal weight matrix and its entries are row sums of W, $D_{ii} = \sum_j W_{ij}$, L=D-W is the Laplacian matrix. We leave out the eigenvector $\mathbf{f}_0$ and use the next d eigenvectors for embedding in d-dimensional Euclidean space: $x_i \to (\mathbf{f}_1(i), \cdots, \mathbf{f}_d(i))$.

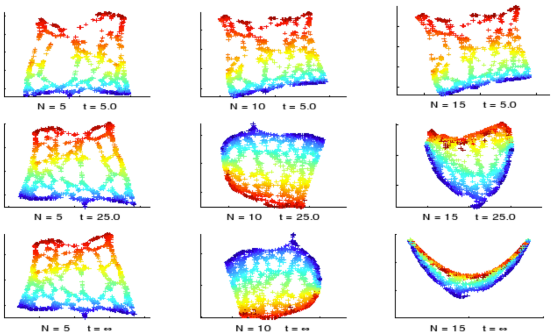


Figure 5: Laplacian Eigenmaps [11]

2.2.4 Discussion

Global vs. Local [6]

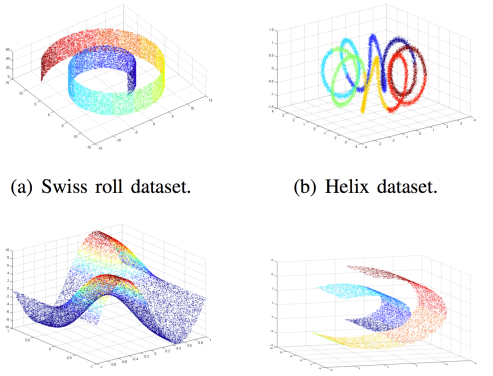
- Local approaches(LLE, Laplacian Eigenmaps) attempt to preserve the local geometry of the data; essentially, they seek to map nearby points on the manifold to nearby points in the low-dimensional representation.
- Global approaches(Isomap) attempt to preserve geometry at all scales, mapping nearby points on the manifold to nearby points in low-dimensional space, and faraway points to faraway points.

Advantages of Isomap

- It will give a more faithful representation of the data's global structure.
- There are formal guarantees of convergence [8, 20]. When the sample size is large and samples are sampled from a submanifold that is isometric to a convex subset of Euclidean space, the Isomap algorithm will recover this subset up to a rigid motion. Many image manifolds generated by translations, rotations, and articulations can be shown to fit into this framework [5].

Advantages of Local Approaches

- They work on a range of manifolds, whose local geometry is close to Euclidean while global geometry is not.
- They're computationally efficient because the involved neighborhood graphs are usually very sparse.



(a) Swiss roll dataset. (b) Helix dataset.

(c) Twinpeaks dataset. (d) Broken Swiss roll dataset.

Figure 6: Four Artificial Datasets [3]

Disadvantages of Graph-based Approaches

Isomap,LLE,Laplacian Eigenmaps are all based on the construction of neighborhood graph and therefore susceptible to:

- The curse of dimensionality of the embedded manifold because the number of data points that is required to characterize a manifold property grow exponentially with the intrinsic dimensionality of the manifold.
- Local properties of a manifold do not necessarily follow the global structure of the manifold in the presence of noise around the manifold. In other words, local methods suffer from overfitting on the manifold.
- There's no principled way to set k, the size of the neighborhood set. As is shown in Figure 5, the unfolding of Swiss roll is very sensitive to the number of nearest neighbors and the scale pa-

rameter of heat kernel. When k is too high with respect to the sampling density of the manifold, the local linearity assumption will be violated. Wang et al. [21] present an adaptive manifold learning algorithm to solve this problem.
- There's no principled approaches to determining the number of intrinsic dimensionality.
- There's no easy out-of-sample extensions for Isomap,LLE and Laplacian Eigenmaps. An estimation technique has been presented in [22], which inevitably leads to estimation errors in the out-of-sample extensions.

Disadvantages of Isomap
Since the theoretical guarantee of Isomap requires samples need to be sampled uniformly and densely from a manifold with no noise, Isomap is prone to short circuits and topological instabilities if these conditions are not satisfied.
- It is sensitive to noise.
- It will fail for nonconvex parameter space.
- It will fail for spaces with high intrinsic curvature.
- There might not exist an isometric or near isometric embedding in MDS step.
- Since the graph is fully connected, the time compleixity is $\mathcal{O}(N^3)$. Therefore, it will be slow for large training sets.

Disadvantages of Local Approaches
As is pointed out in [3], local approaches such as LLE, Laplacian Eigenmaps perform well on a simple manifold such as Swiss roll dataset. However, this strong performance does not generalize well to more complex datasets, which are shown in Figure 6. The Helix dataset and Twinpeaks dataset are not isometric to Euclidean space. The Broken Swiss Roll dataset is not smooth.
- Local approaches fail on Helix dataset and Twinpeaks dataset because they cannot deal with manifolds that are not isometric to the Euclidean space.
- Local approaches fail on Broken Swiss Roll dataset because the manifold contains discontinuities, making it non-smooth.
- The associate eigenproblems might be hard to solve.
- There's no theoretical guarantees for LLE. However, a variant of LLE known as Hessian LLE [23], can asymptotically recovers the low-dimensional parameterization of any high-dimensional data set whose underlying submanifold is isometric to an open, connected subset of Euclidean space. Unlike Isomap, the subset is not required to be convex. Hessian LLE can work with Broken Swiss Roll dataset.

2.3 Kernel Methods

2.3.1 Kernel PCA

Kernel extension can be applied to algorithms that only need to compute the inner product of data pairs. After replacing the inner product with kernel function, data is mapped implicitly from the original input space to higher dimensional space and then apply linear algorithm in the new feature space. The benefit of kernel trick is data that are not linearly separable in the original space could be separable in new high dimensional space. Kernel PCA is often used for NLDR with polynomial or Gaussian kernels. It is important to realize, however, that these generic kernels are not particular well suited to manifold learning [5]. Figure 7 shows the result of kernel PCA on Swiss roll with polynomial kernel and Gaussian kernel. Both fail to discover the intrinsic dimensionality.

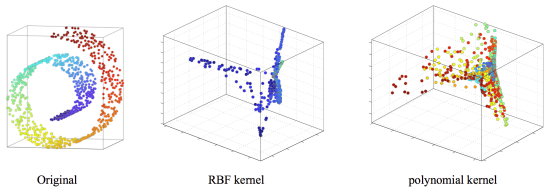


Figure 7: Kernel PCA on Swiss Roll [5]

2.4 Future Work

Based on our review on motivations, assumptions, strengths and weaknesses of dimensionality reduction techniques, the following problems should be the subject of future work.

2.4.1 Topology

There is very little work that is been done on determining the intrinsic topology of high dimensional data. In order to apply manifold learning to practice, it is necessary to decide when natural data actually lie on a manifold.

2.4.2 Manifold Learning vs. Kernel Methods

If we have known data lies on nonlinear manifold, which method is more appropriate, manifold learning methods or kernel methods?

2.4.3 Robustness of NLDR Algorithms

Another direction is to make NLDR algorithms more robust to noise and outliers.

2.4.4 Scale

The algorithm should be able to scale up to large datasets.

3 General Framework of Dimensionality Reduction

3.1 Formulation

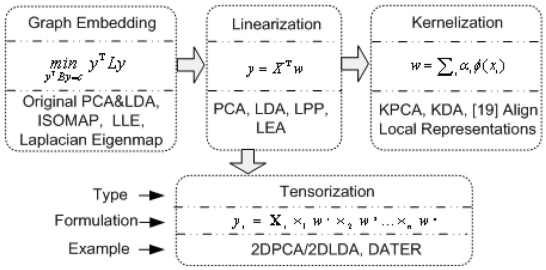


Figure 8: Graph Embedding Framework [1]

Yan et al. [1] propose a general framework of dimensionality reduction based on graph embedding and its linearization/kernelization/tensorization extensions. The graph embedding of the graph G is defined as the desired low-dimensional vector representations that best characterize the similarity relationship between the vertex pairs in G.

$$y^* = \arg\min_{y^T By=d} \sum ||y_i - y_j||^2 W_{ij} \quad (4)$$
$$= \arg\min_{y^T By=d} y^T Ly$$

where d is a constant and B is typically a diagonal matrix for scale normalization and may also the Laplacian matrix of a penalty graph G^p. Theoretical analyais of Laplacian matrix has been presented by Chung [24].

As is shown in Figure 8, manifold learning conducts NLDR for the data lying on or nearly on a lower dimensional manifold. Isomap,LLE and Laplacian Eigenmaps are all direct graph embedding. Subspace learning assumes the low-dimensional vector representations of the vertices can be obtained from a linear projection. PCA, LDA are both linearization of direct graph embedding. Kernel methods implicitly map data into high-dimensional Hilbert space by replacing inner product with kernel function. Kernel PCA, kernel LDA are both kernelization of direct graph embedding. Subspace learning, manifold learning and kernel methods can be expressed in this general framework.

Direct graph embedding and its linearization, kernelization all consider a vector representation of vertices. If the extracted feature from an object contain high-order structure, tensor is more appropriate than vector. Tensor take into account the higer-order structure of an object, making it a better representation method. In uncovering the underlying high-order structure by transforming the input data into a vector as done in most algorithms, which often leads to the curse of dimensionality problem. Compared with the linearization of graph embedding, the feature dimension considered in each iteration of tensorization is much smaller which effectively avoids the curse of dimensionality issue and leads to a significant reduction in computational cost.

3.2 Marginal Fisher Analysis

Based on the disadvantages of LDA we discuss in Section 2.1.3, LDA assumes data distribution of each class to be Gaussian. Marginal Fisher Analysis(MFA) aims to remove this strong assumption by developing new criteria that characterizes intraclass compactness and interclass separability, which are shown in Figure 9. Each sample is connected to its k_1 nearest neighbors in the same class to construct intrinsic graph. Penalty graph is contructed by connecting samples in different classes.

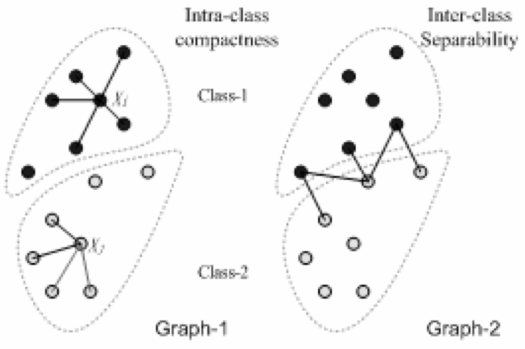


Figure 9: MFA [1]

The performance of MFA is better than LDA

for non-Gaussian data. Figure 10 shows the results of MFA and LDA for non-Gaussian data. MFA can classify the two classes correctly while LDA fails. PCA, LDA are both global, non-parametric methods while MFA is a local, parametric method. The design of k_1 nearest neighbors in MFA makes it become a local approach. Therefore, MFA has the advantage of finding nonlinear embedding compared to linear methods PCA,LDA.

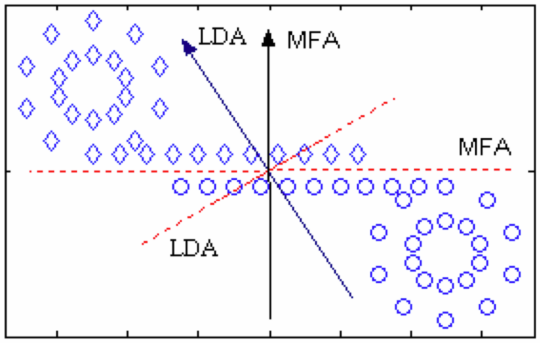


Figure 10: MFA vs. LDA [1]

3.3 Construction of Similarity Matrix

Ham et al. [14] prove Isomap,LLE,Laplacian Eigenmaps can be described as kernel PCA on specially constructed Gram matrices. In the graph embedding framework, different algorithms induce different similarity matrix W.

3.3.1 Isomap

The kernel matrix induced by Isomap is

$$W = \tau(D_G) = -\frac{HSH}{2} \tag{5}$$

where $H = I - \frac{1}{N}ee^T$ and e is a N-dimensional vector filled with 1. S is a squared distance matrix.

3.3.2 LLE

The kernel matrix induced by LLE is

$$W = M + M^T - M^T M \tag{6}$$

where M is the local reconstruction coefficient matrix.

3.3.3 Laplacian Eigenmaps

The kernel matrix induced by Laplacian Eigenmaps is

$$W_{ij} = \exp\left(-\frac{||x_i - x_j||^2}{t}\right) \tag{7}$$

where i is the neighbor of j or j is the neighbor of i.

3.3.4 Discussion

We can draw two conclusions from the formulations above.

- Global and local approaches can be distinguished by looking at their similarity matrices. Global approaches such as PCA will preserve similarity relations at all scales by setting $W_{ij} = 1/N$. While local approaches will only preserve similarity relations in the neighborhood.
- The kernel of manifold learning depends on the data,which is consistent with our intuition. As is pointed out in [16], the right choice of similarity measure is related to the right notion of similarity between data points, which is itself related to the density of the data. Instead of blindly applying a dimensionality reduction algorithm, we can ask the following general question: *Given an arbitrary density model $p(x)$, what is the appropriate similarity measure for graph-based dimensionality reduction algorithms.* One discovery in [16] is methods like spectral clustering empirically appear to capturing such salient features of a data set as its main clusters and sub manifolds, which is unlike previous manifold learning methods like LLE and Isomap which assume a single manifold and have not been designed to say something about the modes of the distribution.

4 Graph Embedding for Image Classification

As is discussed in Section 2, although an image with D pixels can be considered as a point in a D-dimensional image space, the variability of image classes can be represented as low-dimensional manifolds embedded in image space. Rotation only contributes to one degree of freedom. The intrinsic dimensionality of the manifold would increase if we were to allow other types of image transformations such as scaling and translation. However, the dimensionality would still be far more less than D. Graph embedding provides a general framework for dimensionality reduction. Therefore, it's meaningful to apply graph embedding methods to face recognition. In the following we present a few cases of face recognition that use different types of graph embedding and extensions.

4.1 Linear Approaches

Eigenfaces [25] is based on PCA. For linearly embedded subspace, PCA is guaranteed to discover the intrinsic dimensionality. Fisherfaces [26] is based on LDA. It's generally believed LDA outperforms PCA because of supervision. However, [27] shows PCA outperforms LDA when the training set is small and PCA is also less sensitive to different training sets. However, PCA and LDA are both linear methods and therefore will fail when the low-dimensionality lies on a nonlinear manifold.

4.2 Linearization

Manifold learning algorithms, including Isomap, LLE, Laplacian Eigenmaps, does not have straightforward out-of-sample extensions, making them not suitable for face recognition. However, Local Preserving Projection(LPP) [28], which is the linearization of Laplacian Eigenmaps, achieves impressive results in face recognition. Laplacianfaces is based on LPP. Neighborhood Preserving Embedding(NPE) [29] is the linearization of LLE.

4.3 Kernelization

Kernel PCA [30] is ther kernelization of PCA. Kernel LDA [31] is the kernelization of LDA. Compared to their linear form, kernel methods yield lower error rates in face recognition.

4.4 Tensorization

All the above methods treat a face image as a high dimensional vector. They do not consider the spatial correlation of pixels in the image. The real number of degree of freedom will be far less if each image is represented in a 2-order tensor. [32] presents Tensorface algorithm. [33] learns a spatially smooth subspace for face recognition using tensor representation, which gives higher accuracy compared to algorithms above based on vector representations.

5 New Algorithm Design

Besides the possible extensions mentioned in [1], another natural extension is we can treat the penalty graph as a reguarization term and use a control parameter to adjust the contribution of penalty graph.

Based on the insight from the general framework of graph embedding, we will try to solve the following problems:

- Given an arbitrary density model $p(x)$, what is the appropriate similarity measure for graph embedding?
- How to improve the robustness of graph embedding to noisy dimensions?

For the first problem, there is not a principled technique yet. However, some heuristics can be used [34]. For the second problem, we propose a method that combines linear transformation of the original dimensions and graph embedding to do feature selection.

5.1 Data Sensitive Similarity Measure Construction

As is shown in Figure 11: 1) it's difficult to set ϵ for ϵ-graph when we have data on different scales because points on the dense region are already very tightly connected while the points in sparse region are barely connected. 2) However, k nearest neighbor graph can connect points on different scales. 3) While mutual k nearest graph is able to connect points on different scales, it does not mix those scales with each other. Therefore, the mutual k-nearest neighbor graph seems particular well-suited if we want to detect clusters of different densities [34]. 4) When the data include multiple scales and when the clusters are placed within a cluttered background, local scale parameter will lead to better clustering results for spectral clustering [35]. 5) To do spectral clustering for data on multiple scales, a diffusion based measure is presented in [36].

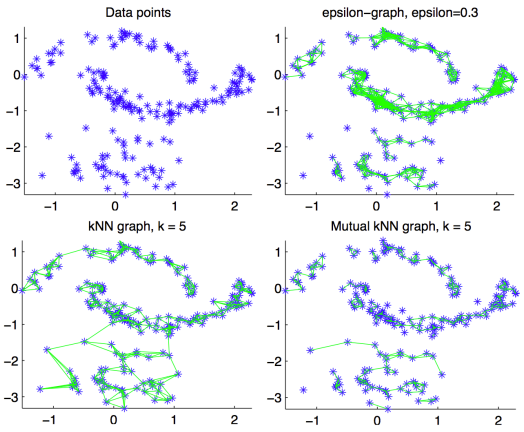


Figure 11: Similarity Graphs [34]

5.2 Feature Extraction and Selection for Graph Embedding

The downstream tasks of dimensionality reduction include supervised learning(classification, regres-

sion), unsupervised learning(clustering) and semi-supervised learning. Before graph embedding is applied, we need to compute pairwise similarities between original feature vectors. When some features are irrelevant for the downstream tasks, they will act as noise and distort the original similarity structure [37]. The main idea is to learn relevant features in the original input feature space. While all the existing NLDR algorithms assume all the original features are useful, which might not be true in practice.

As is suggested by Maaten et al. [3], the work on NLDR should shift towards the algorithms with objective functions that can be optimized well, such as kernel PCA, spectral clustering in the following general form.

$$\arg\min_{Y \in \mathbb{R}^{d \times N}} Q(Y; X, \theta) \tag{8}$$

where Q represents the objective function that need to be minimized, $X \in \mathbb{R}^{D \times N}$ represents the original dataset, $Y \in \mathbb{R}^{d \times N}$ represents the desired low-dimensional embedding, θ represents the free parameters that can be manually set. Before plugging X into this optimization problem, we can apply linear transformation $W \in \mathbb{R}^{D \times q}$ to X such that $Z = W^T X$. Then we solve the following optimization problem instead:

$$\arg\min_{\substack{Y \in \mathbb{R}^{d \times N} \\ W \in \mathbb{R}^{D \times q}}} Q(Y; W^T X, \theta) \tag{9}$$

We can add constraint to projection matrix W to avoid trivial solution.

An alternative is to pursue feature selection instead of feature extraction if we want to keep the original features. To achieve this goal, we can add l_1/l_∞ regularization term [38] and solve the following optimization problem:

$$\arg\min_{\substack{Y \in \mathbb{R}^{d \times N} \\ W \in \mathbb{R}^{D \times q}}} Q(Y; W^T X, \theta) + \lambda \sum_{j=1}^{D} ||W_j||_\infty \tag{10}$$

where W_j represents the j-th row of linear transformation matrix W.

6 Conclusions

Dimensionality reduction algorithms can be unified into a common graph embedding framework. Different algorithms are based on different motivations, assumptions and therefore have their respective strengths and weaknesses. These approaches cannot serve as a "black box algorithm" which automatically find the intrinsic dimensionality in any given dataset. The challenge is to design a principled algorithm to construct similarity measure for the graph given arbitrary data distribution. After discussing the motivation and methods of applying the proposed general framework to face recognition, we first present some heuristics on choosing similarity measures that correspond to different data distributions. Then a general method to conduct feature extraction and feature selection for the original features is proposed. The new algorithm is expected to make graph embedding robust to noisy dimensions.

References

[1] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: a general framework for dimensionality reduction," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 1, pp. 40–51, 2007.

[2] K. Fukunaga, *Introduction to statistical pattern recognition*. Academic press, 1990.

[3] L. J. van der Maaten, E. O. Postma, and H. J. van den Herik, "Dimensionality reduction: A comparative review," *Journal of Machine Learning Research*, vol. 10, no. 1-41, pp. 66–71, 2009.

[4] T. F. Cox and M. A. Cox, *Multidimensional scaling*. CRC Press, 2000.

[5] L. K. Saul, K. Q. Weinberger, J. H. Ham, F. Sha, and D. D. Lee, "Spectral methods for dimensionality reduction," *Semisupervised learning*, pp. 293–308, 2006.

[6] V. De Silva and J. B. Tenenbaum, "Global versus local methods in nonlinear dimensionality reduction," *Advances in neural information processing systems*, pp. 721–728, 2003.

[7] T. Lin and H. Zha, "Riemannian manifold learning," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 5, pp. 796–809, 2008.

[8] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[9] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[10] L. K. Saul and S. T. Roweis, "Think globally, fit locally: unsupervised learning of low dimensional manifolds," *The Journal of Machine Learning Research*, vol. 4, pp. 119–155, 2003.

[11] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[12] B. Scholkopf and A. Smola, "Learning with kernels," 2002.

[13] B. Scholkopf, A. Smola, and K.-R. Müller, "Kernel principal component analysis," in *Advances in kernel methods-support vector learning*, Citeseer, 1999.

[14] J. Ham, D. D. Lee, S. Mika, and B. Schölkopf, "A kernel view of the dimensionality reduction of manifolds," in *Proceedings of the twenty-first international conference on Machine learning*, p. 47, ACM, 2004.

[15] M. Brand, "Continuous nonlinear dimensionality reduction by kernel eigenmaps," in *IJCAI*, pp. 547–554, 2003.

[16] Y. Bengio, P. Vincent, J.-F. Paiement, O. Delalleau, M. Ouimet, and N. LeRoux, "Learning eigenfunctions of similarity: linking spectral clustering and kernel pca," *Dpartement dinformatique et recherche oprationnelle, Universit de Montral, Tech. Rep*, 2003.

[17] H. S. Seung and D. D. Lee, "The manifold ways of perception," *Science*, vol. 290, no. 5500, pp. 2268–2269, 2000.

[18] J. Shlens, "A tutorial on principal component analysis," *Systems Neurobiology Laboratory, University of California at San Diego*, vol. 82, 2005.

[19] C. M. Bishop *et al.*, *Pattern recognition and machine learning*, vol. 1. springer New York, 2006.

[20] D. L. Donoho and C. Grimes, *When does ISOMAP recover the natural parameterization of families of articulated images?* Department of Statistics, Stanford University, 2002.

[21] J. Wang, Z. Zhang, and H. Zha, "Adaptive manifold learning.," in *NIPS*, vol. 2004, 2004.

[22] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. Le Roux, and M. Ouimet, "Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering," *Mij*, vol. 1, p. 2, 2003.

[23] D. L. Donoho and C. Grimes, "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data," *Proceedings of the National Academy of Sciences*, vol. 100, no. 10, pp. 5591–5596, 2003.

[24] F. R. Chung, *Spectral graph theory*, vol. 92. American Mathematical Soc., 1997.

[25] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pp. 586–591, IEEE, 1991.

[26] P. N. Belhumeur, J. P. Hespanha, and D. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 7, pp. 711–720, 1997.

[27] A. M. Martínez and A. C. Kak, "Pca versus lda," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 2, pp. 228–233, 2001.

[28] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face recognition using laplacianfaces," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 3, pp. 328–340, 2005.

[29] X. He, D. Cai, S. Yan, and H.-J. Zhang, "Neighborhood preserving embedding," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2, pp. 1208–1213, IEEE, 2005.

[30] M.-H. Yang, "Face recognition" sing kernel methods," 2002.

[31] Q. Liu, R. Huang, H. Lu, and S. Ma, "Face recognition using kernel-based fisher discriminant analysis," in *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pp. 197–201, IEEE, 2002.

[32] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear subspace analysis of image ensembles," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 2, pp. II–93, IEEE, 2003.

[33] D. Cai, X. He, Y. Hu, J. Han, and T. Huang, "Learning a spatially smooth subspace for face recognition," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1–7, IEEE, 2007.

[34] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[35] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering.," in *NIPS*, vol. 17, p. 16, 2004.

[36] B. Nadler and M. Galun, "Fundamental limitations of spectral clustering," *Advances in Neural Information Processing Systems*, vol. 19, p. 1017, 2007.

[37] D. Niu, J. G. Dy, and M. I. Jordan, "Dimensionality reduction for spectral clustering," in *International Conference on Artificial Intelligence and Statistics*, pp. 552–560, 2011.

[38] M. Masaeli, J. G. Dy, and G. M. Fung, "From transformation-based dimensionality reduction to feature selection," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 751–758, 2010.