Dimension Reduction and Manifold Learning

Xin Zhang
eexinzhang@scut.edu.cn
Content

- Motivation of manifold learning
- Principal component analysis and its extension
- Manifold learning
  - Global nonlinear manifold learning (IsoMap)
  - Local nonlinear manifold learning (LLE)
- Example of applications
Motivations

- Need to analyze large amounts multivariate high dimensional data.
  - Human faces, shape and motion.
  - Speech Waveforms.
  - Global Climate patterns.
  - Gene Distributions.
- Difficult to visualize data in dimensions just greater than three
- “Curse of Dimensionality”
Motivation, Assumption and Goal

- **Motivation**
  - Large volumes of high-dimensional data
  - “Curse of Dimensionality”

- **Assumption**
  - We assume the data lies on an embedded manifold (Euclidean) within the high-dimensional space.

- **Goal**
  - To find meaningful low-dimensional structures hidden in their high-dimensional observations

http://www.seas.upenn.edu/~kilianw/nldrworkshop06/Workshop%20Description.html
Many data sets have the property that the data points all lie close to a manifold of much lower dimensionality than that of the original data space.

Dimension Reduction (continuous latent variable).

- To reduce the degree of freedom (DoF) of the data set.
- To reduce the number of random variable under consideration.
- To find a sub-space that preserves the major property of data.
Major Manifold Learning Algorithms

- Linear
  - Principle Component Analysis
  - Linear Discriminative Analysis

- Non-linear
  - Tensor
  - Graph Embedding
    - Local Linear Embedding
    - ISOMAP
    - Laplacian Eigenmap
    - Conceptual Manifold
  - Kernel

Linear Dimension Reduction: Principal Component Analysis (PCA)

- PCA is defined as the orthogonal projection of the data onto a lower-dimensional *linear space*, such that the variance of the projected data is maximized.
Non-linear Dimensional Reduction (Manifold Learning)

- Manifold learning is the process of exploring a low-dimensional non-linear embedding underlying a set of high-dimensional data.

http://sciencewatch.com/ana/st/face/09mayFacRecHe/
Content

- Motivation of manifold learning
- Principal component analysis and its extension
- Manifold learning
  - Global nonlinear manifold learning (IsoMap)
  - Local nonlinear manifold learning (LLE)
- Example of applications
PCA Formulations

- **Maximum variance formulation**
  - To find a sub-space where the variance of the projected data is maximized.

- **Minimum-error formulation**
  - To find a sub-space where data can be reconstructed with least square error.

*Figure 12.2* Principal component analysis seeks a space of lower dimensionality, known as the principal subspace and denoted by the magenta line, such that the orthogonal projection of the data points (red dots) onto this subspace maximizes the variance of the projected points (green dots). An alternative definition of PCA is based on minimizing the sum-of-squares of the projection errors, indicated by the blue lines.
Maximum Variance Formulation

- Given a D-dimensional data set

\[ \{ x_n \mid n = 1, \ldots, N \} \text{ and } x_n \in \mathbb{R}^D \]

- We project the data on to a M-dimensional (M<<D) subspace with the maximum variance.

- Let’s consider a 1-D space determined by \( u_1 \) with \( u_1^T u_1 = 1 \).

- We compute the variance of the projected data by

\[
\frac{1}{N} \sum_{n=1}^{N} \left( u_1^T (x_n - \overline{x}) \right)^2 \quad \text{where} \quad \overline{x} = \frac{1}{N} \sum_{n=1}^{N} x_n
\]
Variance Computation

\[ \frac{1}{N} \sum_{n=1}^{N} (u_1^T (x_n - \bar{x}))^2 \]

\[ = \frac{1}{N} \sum_{n=1}^{N} (u_1^T (x_n - \bar{x}))u_1^T (x_n - \bar{x}) \]

\[ = \frac{1}{N} \sum_{n=1}^{N} (u_1^T (x_n - \bar{x}))((x_n - \bar{x})^T u_1) \]

\[ u_1 \left( \frac{1}{N} \sum_{n=1}^{N} (x_n - \bar{x})^T (x_n - \bar{x}) \right) u_1^T \]

\[ u_1^T S u_1 \]

\[ S = \frac{1}{N} \left( \sum_{n=1}^{N} (x_n - \bar{x})^T (x_n - \bar{x}) \right) \]
Variance Maximization

- We now want to maximize the projected variance with respect to $u_1$ with $u_1^T u_1 = 1$.
  $$u_1^* = \arg \max \left( u_1^T Su_1 + \lambda_1 (1 - u_1^T u_1) \right)$$
  $$\frac{d}{du_1} \left( u_1^T Su_1 + \lambda_1 (1 - u_1^T u_1) \right) = 0$$

  \[
  \frac{\partial x^T Ax}{\partial x} = 2Ax \quad \frac{\partial x^T x}{\partial x} = 2x \\
  (A = A^T)
  \]

  $$2Su_1 - 2\lambda u_1 = 0$$
  $$Su_1 = \lambda u_1 \quad \Rightarrow \quad u_1^T Su_1 = \lambda$$

- The variance will be a maximum if $u_1$ equal to the eigen-vector having the largest eigen-value.
M-Dimensional Subspace

For an $M$-dimensional projection space, the optimal linear projection is defined by the $M$ eigenvectors associated with the $M$ largest eigenvalues.

$$\eta_p = \frac{\sum_{i=1}^{p} \lambda_i^2}{N \sum_{i=1}^{N} \lambda_i^2}$$

![Graph showing the relationship between $p$ and $\eta_p$.]
PCA Applications: Data Compression (1)

**Figure 12.3** The mean vector $\bar{x}$ along with the first four PCA eigenvectors $u_1, \ldots, u_4$ for the off-line digits data set, together with the corresponding eigenvalues.

**Figure 12.5** An original example from the off-line digits data set together with its PCA reconstructions obtained by retaining $M$ principal components for various values of $M$. As $M$ increases the reconstruction becomes more accurate and would become perfect when $M = D = 28 \times 28 = 784$. 
PCA Applications:
Data Compression (2)

Figure 12.4  (a) Plot of the eigenvalue spectrum for the off-line digits data set.  (b) Plot of the sum of the discarded eigenvalues, which represents the sum-of-squares distortion $J$ introduced by projecting the data onto a principal component subspace of dimensionality $M$. 

PCA Applications:
Data Pre-processing

Figure 12.6 Illustration of the effects of linear pre-processing applied to the Old Faithful data set. The plot on the left shows the original data. The centre plot shows the result of standardizing the individual variables to zero mean and unit variance. Also shown are the principal axes of this normalized data set, plotted over the range \( \pm \lambda_i^{1/2} \). The plot on the right shows the result of whitening of the data to give it zero mean and unit covariance.

\[
\overline{X} = \frac{1}{N} \sum_{n=1}^{N} X_n \\
Z_{ni} = \frac{x_{ni} - m_i}{\sqrt{v_i}} \\
Y_n = L^{-1/2}U^T(x_n - \overline{X})
\]

\[
L = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \quad U = (u_1 \quad u_2)
\]

\[
\frac{1}{N} \sum_{n=1}^{N} Y_n Y_n^T = I
\]

\[
m_i = \frac{1}{N} \sum_{n=1}^{N} x_{ni} \quad v_i = \frac{1}{N} \sum_{n=1}^{N} (x_{ni} - m_i)^2
\]
PCA Applications: Face Recognition

Original face

PCA-1  PCA-2  PCA-3  PCA-4  PCA-5  PCA-6  PCA-7  PCA-8  PCA-9

eigen face 1  eigen face 2  eigen face 3

eigen face 4  eigen face 5  eigen face 6

eigen face 7  eigen face 8  eigen face 9
Figure 25.7. Not every data set that is well represented by PCA. The principal components of this data set will be relatively unstable, because the variance in each direction is the same for the source. This means that we may well report significantly different principal components for different datasets from this source. This is a secondary issue—the main difficulty is that projecting the data set onto some axis will suppress the main feature, its circular structure.
Kernel PCA

- We introduce a non-linear transformation that converts each data points into a M-dimensional feature space, where we can perform standard PCA.

![Diagram of Kernel PCA](image)

**Figure 12.16** Schematic illustration of kernel PCA. A data set in the original data space (left-hand plot) is projected by a nonlinear transformation $\phi(x)$ into a feature space (right-hand plot). By performing PCA in the feature space, we obtain the principal components, of which the first is shown in blue and is denoted by the vector $v_1$. The green lines in feature space indicate the linear projections onto the first principal component, which correspond to nonlinear projections in the original data space. Note that in general it is not possible to represent the nonlinear principal component by a vector in $x$ space.
Content

- Motivation of manifold learning
- Principal component analysis and its extension
- Manifold learning
  - Global nonlinear manifold learning (IsoMap)
  - Local nonlinear manifold learning (LLE)
- Example of applications
Concepts of Manifold

- “A manifold is a topological space which is locally Euclidean.”
- In general, any object which is nearly "flat" on small scales is a manifold.
- Euclidean space is a simplest example of a manifold.
- Concept of submanifold.
- Manifolds arise naturally whenever there is a smooth variation of parameters [like pose of the face in previous example]
- The dimension of a manifold is the minimum integer number of co-ordinates necessary to identify each point in that manifold.

Embed data in a higher dimensional space to a lower dimensional manifold

Concept of Dimensionality Reduction:
Manifold of Perception..Human Visual System

You never see the same face twice.

Preceive constancy when raw sensory inputs are in flux..

$10^6$ optic nerve fibres
Interpolation along Manifold

Time

Manifold Interpolation

http://www.cs.cmu.edu/~efros/courses/AP06/presentations/ThompsonDimensionalityReduction.pdf
Major Manifold Learning Algorithms

- Principle Component Analysis
- Linear Discriminative Analysis
- Local Linear Embedding
- ISOMAP
- Laplacian Eigenmap
- Conceptual Manifold

Linear Dimensionality Reduction Methods

- **Principle component analysis (PCA):**
  - It seeks projection directions with maximal variances. In other words, it removes directions with minimal variances.
  - It can be extended as the non-linear method, like kernel PCA.

- **Linear discriminative analysis (LDA):**
  - It searches for the projection directions that are most effective for discrimination by minimizing the ratio between the intra-class and interclass scatters.

- **Extra information incorporation**
Graph Embedding Manifold Learning

- **Goal:**
  - Introduce geometric property or prior topology knowledge into similarity measurement among data.

- **Assumption:**
  - The global non-linear structure has local linear smoothness.
  - The local similarity can be measured as Euclidean distance.

- **Approach:**
  - Represent each vertex of a graph as a low-dimensional vector that preserves similarities between vertex pairs in the high-dimensional space.
Local Linear Embedding (LLE)

- LLE maps the input data to a lower dimensional space in a manner that preserves the relationship between the neighboring points.
- Fit locally, think globally!
  - Discover the global internal coordinates of the manifold
  - The color coding illustrates the neighborhood preservation

We expect each data point and its neighbours to lie on or close to a locally linear patch of the manifold.

- Each point can be written as a linear combination of its neighbors.
- Weights are chosen to minimize the reconstruction Error.

\[
min_W \| X_i - \sum_{j=1}^{K} W_{ij} X_j \|^2
\]

Derivation on board
Important property...

- The weights that minimize the reconstruction errors are invariant to rotation, rescaling and translation of the data points.
  - Invariance to translation is enforced by adding the constraint that the weights sum to one.

- The same weights that reconstruct the data points in D dimensions should reconstruct it in the manifold in d dimensions.
  - The weights characterize the intrinsic geometric properties of each neighborhood.
Think Globally…

\[ Y_{d \times N} = [Y_1 | Y_2 | \ldots | Y_N] \]

\[ \min_Y \sum_{i=1}^{N} \| Y_i - YW_i \|^2 \]
Local Linear Embedding (LLE)

- Assign neighbors to each data point.
- Find linear weights by minimizing
  \[ \mathcal{E}(W) = \sum_i \left( \tilde{X}_i - \sum_j W_{ij} \tilde{X}_j \right)^2 \]
  that can be solved as a least-square problem with weights sum-to-one constraint.
- Compute the low dimensional embedding vector by minimizing reconstruction error with fixed weights
  \[ \Phi(Y) = \sum_i \left( \tilde{Y}_i - \sum_j W_{ij} \tilde{Y}_j \right)^2 \]
Local Linear Embedding (LLE)

- Assign neighbors to each data point.
- Find linear weights by minimizing
  \[ \mathcal{E}(W) = \sum_i | \tilde{X}_i - \sum_j W_{ij} \tilde{X}_j |^2 \]
  that can be solved as a least-square problem with weights sum-to-one constraint.
- Compute the low dimensional embedding vector by minimizing reconstruction error with fixed weights
  \[ \Phi(Y) = \sum_i | \tilde{Y}_i - \sum_j W_{ij} \tilde{Y}_j |^2 \]
LLE Detailed Algorithm

- **Input X**: D by N matrix consisting of N data items in D dimensions.
- **Output Y**: d by N matrix consisting of d < D dimensional embedding coordinates for the input points.

1. **Find neighbors in X space (e.g. using KNN)**
   for i=1:N
   compute the distance from Xi to every other point Xj
   find the K smallest distances
   assign the corresponding points to be neighbors of Xi
   End

2. **Solve for reconstruction weights W.**
   for i=1:N
   create matrix Z consisting of all neighbors of Xi (in columns)
   subtract Xi from every column of Z
   compute the local covariance C=Z'*Z
   solve linear system C*w = 1 for w (1 denotes column vector of all ones)
   set elements in the i-th row of w equal to w/sum(w);
   end

3. **Compute embedding coordinates Y using weights W.**
   create sparse matrix M = (I-W)'*(I-W)
   find bottom d+1 eigenvectors of M
   (corresponding to the d+1 smallest eigenvalues)
   set the qthRow of Y to be the q+1 smallest eigenvector
   (discard the bottom eigenvector [1,1,1,1...] with eigenvalue zero)
LLE Result and Discussion

- PCA (top) and LLE (bottom) comparison of the manifold learning
  - Applied two methods to a set of image generated by a single face translated across a 2-D noisy background.
  - LLE maps the images with corner faces to the corners of its 2-D embedding while PCA fails to preserve the neighborhood structure of nearby images.

- Pros and Cons of LLE
  - Incremental, fast, one free parameter
  - Simple linear algebra operations
  - Might distort global structure
  - No mapping relationship
ISOMAP

- ISOMAP finds the low-dimensional representations for a data set by approximately preserving the **geodesic distances** of the data pairs.

- Assumption: only geodesic distances reflect the true geometry of the manifold and preserve the intrinsic geometry of the data.

---

ISOMAP Algorithm

I. Construct neighborhood graph
   - Build a sparse graph with K-nearest neighbors for every point

II. Compute shortest path
   - Infer other inter-point distance by finding shortest path on the graph

III. Construct d-dimensional embedding
   - A d-dimensional space to preserve inter-point distances by using the top eigenvectors scaled by their eigenvalues.

\[ y_i = [\sqrt{\lambda_1}v_1^i, \sqrt{\lambda_2}v_2^i, \ldots, \sqrt{\lambda_m}v_m^i] \]
ISOMAP Result
ISOMAP Discussions

Pros:
- It preserves global structure
- One parameter for neighborhood determination

Cons:
- Sensitive to noise, noise edges
- Computationally expensive
## Comparison

<table>
<thead>
<tr>
<th>ISOMAP</th>
<th>LLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do MDS on the geodesic distance matrix.</td>
<td>Model local neighborhoods as linear a patches and then embed in a lower dimensional manifold.</td>
</tr>
<tr>
<td>Global approach</td>
<td>Local approach</td>
</tr>
<tr>
<td>Might not work for nonconvex manifolds with holes</td>
<td>Nonconvex manifolds with holes</td>
</tr>
<tr>
<td>Extensions: Landmark, Conformal &amp; Isometric ISOMAP</td>
<td>Extensions: Hessian LLE, Laplacian Eigenmaps etc.</td>
</tr>
</tbody>
</table>

Both needs manifold finely sampled.
Brief Summary of Graph Embedding

- The low-dimensional vector representations relationship best characterize the similar graphic or geometric relationship between the high dimensional data pairs.

Extensions

- Prior knowledge guided manifold learning
- Kernel manifold learning
- Incremental manifold learning
- Out-of-sample manifold learning
- Multiple output manifold learning
**Conceptual Manifold**

- Rather than learning a manifold from data, a conceptual manifold is proposed to ideally characterize the intrinsic geometry of data.
- Torus manifold can simultaneous inference the view and body pose based.

---

Motivation of manifold learning
Principal component analysis and its extension
Manifold learning
  - Global nonlinear manifold learning (IsoMap)
  - Local nonlinear manifold learning (LLE)
Example of applications
Manifold Learning Applications

- Manifold learning has produced successful results on
  - Image denoise
  - Face recognition
  - Face expression recognition and interpolation
  - Character recognition
  - Gesture recognition
  - Activity recognition
  - Pose estimation
  - Dynamic appearance modeling for tracking
  - and so on
Face Interpolation

Face Expression Recognition

Pose Estimation

- Appearance manifold
  - Single view appearance manifold
  - Multiple-view appearance manifold

People Recognition and Image De-noise

Pose estimation using Torus
Dynamic Appearance Modeling and Tracking

Conclusion

- Manifold learning is an efficient tool to discover the embedding space with the intrinsic structure.
- The graphical, geometric, topological information is important prior knowledge.
- There are still some issues, like the neighborhood selection, the parameter optimization, the noisy and sparse data set, the mapping relationship between low-dimensional representation and the high-dimensional data.

READ

- An introduction to Monte Carlo Sampling by McKay
- Chapter 11 of Pattern Recognition and Machine Intelligence by Bishop