Dimension Reduction

Linear Algebra Project Report, Dr.Ramin Javadi

MohammadSadegh Akhoundzadeh , Maryam Meghdadi

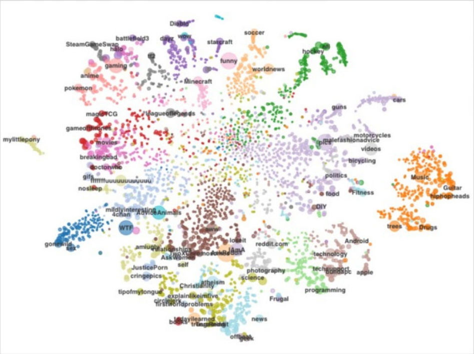
Department of Electrical and Computer Engineering, Isfahan University of Technology

Table of contents

Introduction

Dimensionality reduction : Reducing the number of random variables by obtaining a set of principal variables
The higher the number of features -> the harder it gets to visualize the training set and then work on it

- feature selection
- feature extraction



Linear Methods

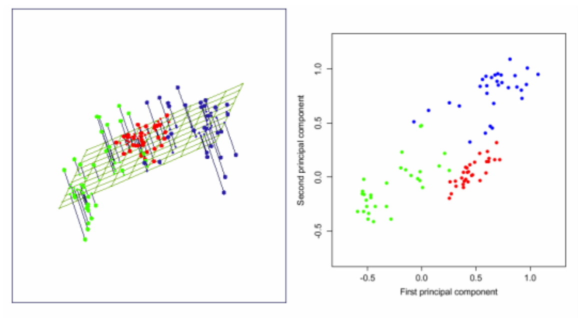
Linear Methods

We discuss and interpret these linear methods:

- PCA
- DUAL PCA
- RANDOM PROJECTION

Principal Component Analysis (PCA)

- Very popular technique
- Find a reduced subspace which maintains most of the variability of the data (n→d)
- d orthogonal vectors that form a new coordinate system which are 'principal components'



Principal Component Analysis (PCA)

Main goal : find maximum variance →choose first eigenvector U_1
Given : n × t matrix X and U_1 is is a combination of X with ω coefficients such that $\omega = [\omega_1 \ldots \omega_n]$

$$U_1 = \omega^T X \tag{1}$$

$$\text{var}(U_1) = \text{var}(\omega^T X) = \omega^T S \omega \tag{2}$$

where S is the n × n sample covariance matrix of X.

$$\max \omega^T \mathrm{S}\, \omega \quad \text{s.t. } \omega^T \omega = 1$$

Using lagrange method and introduce a lagrange multiplier λ we have:

$$L(\omega, \lambda) = \omega^T S \omega - \lambda(\omega^T \omega - 1) \tag{3}$$

By differentiating L with respect to ω we have:

$$S\omega = \lambda_1 \omega \tag{4}$$

Principal Component Analysis (PCA)

Continuing this method for d eigenvectors of covariance matrix S **->** determine the first d principal components Using SVD **->** obtain these eigenvectors:

$$X = U\Sigma V^T \tag{5}$$

where columns of U are eigenvectors of XX^T (covariance matrix).

Algorithm: We define Y as our projected d-dimensional data (d $<$ n):

1. $$Y = U(:, 1:d)^T X \qquad or \qquad Y = U_d^T X \tag{6}$$
2. Reconstruct the training data:
$$\hat{X} = U_d Y \tag{7}$$
3. For a new test example x we compute $y = U^T x$ and then reconstruct it $\hat{x} = Uy$

Dual PCA

- Very similar to PCA
- Faster in the case that the number of features is bigger than the samples
- Goal **:** reduce dependence of our algorithm on n (the number of features)
- Decompose X^TX instead of XX^T
- SVD: X = $U\Sigma V^T$ **->** $XV = U\Sigma$, where the eigenvectors in U corresponds to nonzero singular values in Σ.

The top d eigenvectors can be derived:

$$U = XV\Sigma^{-1} \tag{8}$$

Replace all uses of U in PCA algorithm.

Algorithm:

1. Compute Y:
$$Y = U^T X = \Sigma V^T \tag{9}$$
2. Reconstruct data:
$$\hat{X} = UY = U\Sigma V^T = XV\Sigma^{-1}\Sigma V^T = XVV^T \tag{10}$$
3. Project out of sample point x:
$$y = U^T x = \Sigma^{-1} V^T X^T x \tag{11}$$
4. Finally reconstruct this sample point:
$$\hat{x} = Uy = UU^T x = XV\Sigma^{-2} V^T X^T x \tag{12}$$

Random Projection

- for the cases with large number of samples

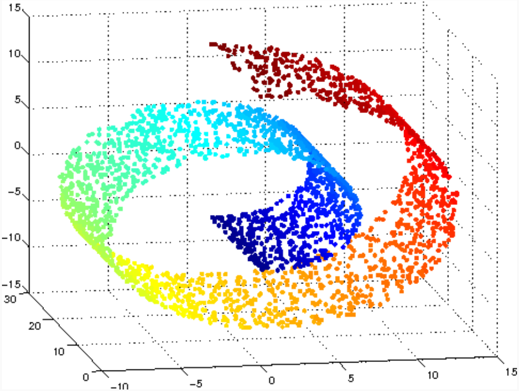
- What does it guarantee?
  after projecting the points from the n-dimensional space to K dimensional space using the randomly drawn matrix W, the distances between the high dimensional points is preserved in the lower dimensional projections up to some approximation factor

This matrix W is:

$$W[i,j] = \left\{ \begin{array}{ll} +\frac{1}{\sqrt{K}} & \text{with probability } 1/2 \\ -\frac{1}{\sqrt{K}} & \text{with probability } 1/2 \end{array} \right\}$$

Nonlinear Methods

What happen's if we have a manifold ?



Distance

- Euclidean distance
- Geodesic distance

Algorithm:

1. Construct a k-nearest neighbor graph on n data points
2. Compute shortest path between all the points (DG) as an estimation of geodesic distance
    - Dijkstra's algorithm
    - Floyd-Warshall
3. Compute $k = -\frac{1}{2}H(D^G)^2H$. find eigenvectors of k and call it V. then find the top p eigenvalues of k and form $\hat{\Lambda}$. The final solution is $Y = \hat{\Lambda}^{\frac{1}{2}}V^T$.

Laplacian Eigenmap (Spectral Embedding)

1. Using Spectral Clustering
2. Affinity Matrix
   - The neighbourhood graph can be constructed by finding the k nearest neighbours and computing the adjacency matrix
   - Gaussian Distance

$$W_{ij} = e^{-\frac{||x_i - x_j||^2}{\gamma}} \tag{13}$$

3. Objective Function

$$\min_{Y} \sum_{i=1}^{t} \sum_{j=1}^{t} (y_i - y_j)^2 W_{ij} \tag{14}$$

From spectral clustering we know that:

$$\sum_{i,j} (y_i - y_j)^2 W_{ij} = Y^T L Y \tag{15}$$

$$\min_{Y} Tr(YLY^T) \tag{16}$$

$$\text{s.t. } YY^T = I$$

Experiment

start by generating three different data to compare the algorithms

- simple ”Hello” in three dimension
- curved ”Hello”
- S-shape data

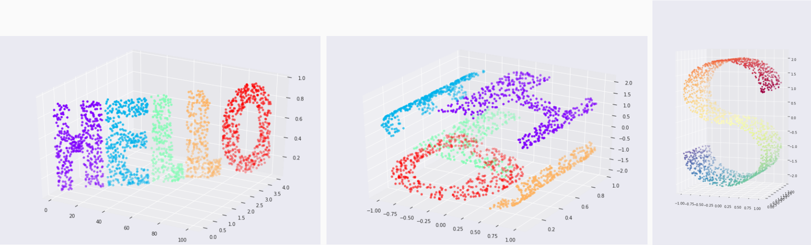


Figure 1: data points that are used

The result of algorithms on the first data point

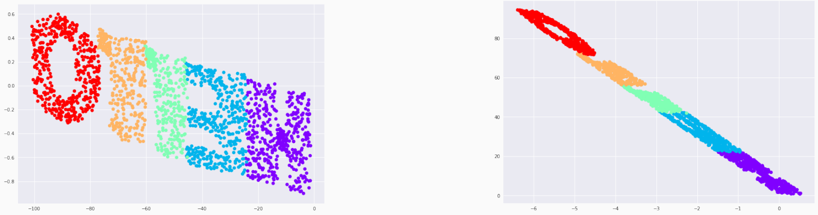


Figure 2: PCA and Random Projection applied to the first data point

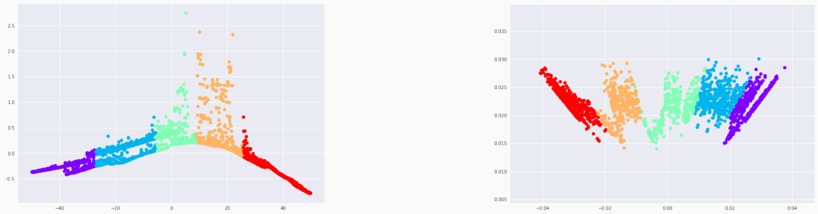


Figure 3: Isomap and Laplacian eigenmap applied to the first data point

The result of algorithms on the second data point

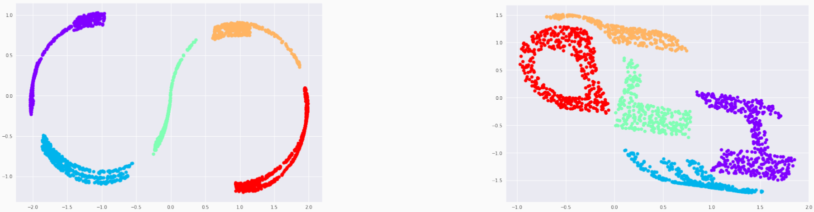


Figure 4: PCA and Random Projection applied to the second data point

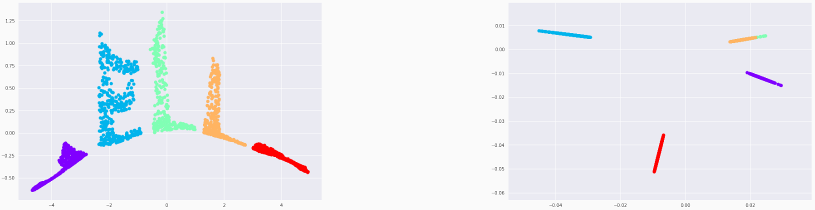


Figure 5: Isomap and Laplacian eigenmap applied to the second data point

The result of algorithms on the third data point

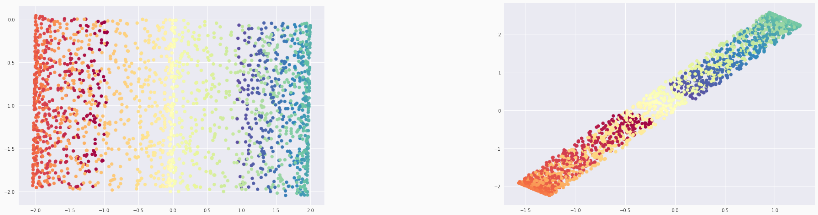


Figure 6: PCA and Random Projection applied to the third data point

Figure 7: Isomap and Laplacian eigenmap applied to the third data point

Source

Get the source of the project from here:

`https://github.com/Linear-Algebra-Course/dimension-reduction`

References I

[1] [2] [3]

Ali Ghodsi.
Dimensionality Reduction A Short Tutorial.
Science, page 25, 2006.

E Kokiopoulou, J Chen, and Y Saad.
Trace Optimization and eigenproblem in dimension reduction methods.
Numerical Linear Algebra with Applications, pages 1–35, 2011.

Random Projections.
Machine Learning for Data Science (CS 4786) Why Random Projection Works ?!
(Cs 4786):1–5.