COMS 4995: Unsupervised Learning (Summer'18) June 12, 2018

Lecture 7 – Linear Dimensionality Reduction

Instructor: *Nakul Verma* Scribes: *Di Zhang*

Overview: Distance Matrix Learning, Independent Component Analysis(Blind Source Separation), Matrix Factorization and Manifold Embedding

1 Review for Last Lecture

Linear Dimensionality Reduction:
1.RP
2.PCA
3.LDA(supervised technique)
"maximizing" the distance between class means
"minimizing" the inter-cluster varience
4.MDS
Given: $dist(O_i, O_j) = \delta_{ij} \quad x_i, x_j \in R^D$ s.t. $\quad ||x_i - x_j||_2 \doteq \delta_{ij}$
Goal: $minS(x_1, ..., x_n) = \sum_{i<j}(D_{ij} - \delta_{ij})^2$
Question: If new data comes, do we need to do the optimization again or there is a simple way?
Answer: This is a question related to "out of simple" extension.

2 Distance Metric Learning

Given: $x_i, ..., x_n \in R^D \;\; \rho(x_i, x_j) = ||x_i - x_j||_2 = [\sum_{d=1}^{D}(x_{id} - x_{jd})^{1/2}]^{1/2} = [(x_i - x_j)^T I(x_i - x_j)]^{1/2}$
Output: Best Matrix $L \in R^{K \times D}$ for representing the data(improve the classification)
One observation:

$$\rho_L(x_i, x_j) = ||Lx_i - Lx_j||_2 = [(x_i - x_j)^T L^T L(x_i - x_j)]^{1/2}$$

Define $M = L^T L$
"Supervision": $x_1, ...x_n \in R^D; y_1, ...y_n \in \{0, 1\}$
Idea: Find M s.t. distances belonging to same class small and distances belonging to different classes large.
Define 1."similar set" $S = \{(x_i, x_j)\}\, s.t.\, y_i = y_j$ 2."different set" $D = \{(x_i, x_j)\}\, s.t.\, y_i \neq y_j$
Professor came up an objective function:

$$min\Psi(M) = \sum_{(x_i,x_j)\in S} \rho_M^2(x_i, x_j)\frac{1}{|S|} - \lambda \sum_{(x_i,x_j)\in D} \rho_M^2(x_i, x_j)\frac{1}{|D|}$$

The first term can be called "pull term ", the second "push term", λ is a hyper-parameter.
The classic approach is:

$$max \sum_{(x_i,x_j)\in D} \rho_M^2(x_i,x_j)$$

s.t.

$$\sum_{(x_i,x_j)\in S} \rho_M^2(x_i,x_j) \leq 1$$

$$M \geq 0 \qquad [M \in PSD]$$

$$rank(M) \leq k \qquad (\text{``}non-convex'')$$

Note1: $M \geq 0$ is "conic constrain", it can be solved by "semi-definite program", the basic idea is pick up negative eigenvalue and make it to be 0. Figure 1 shows some basic idea about how to deal with it.

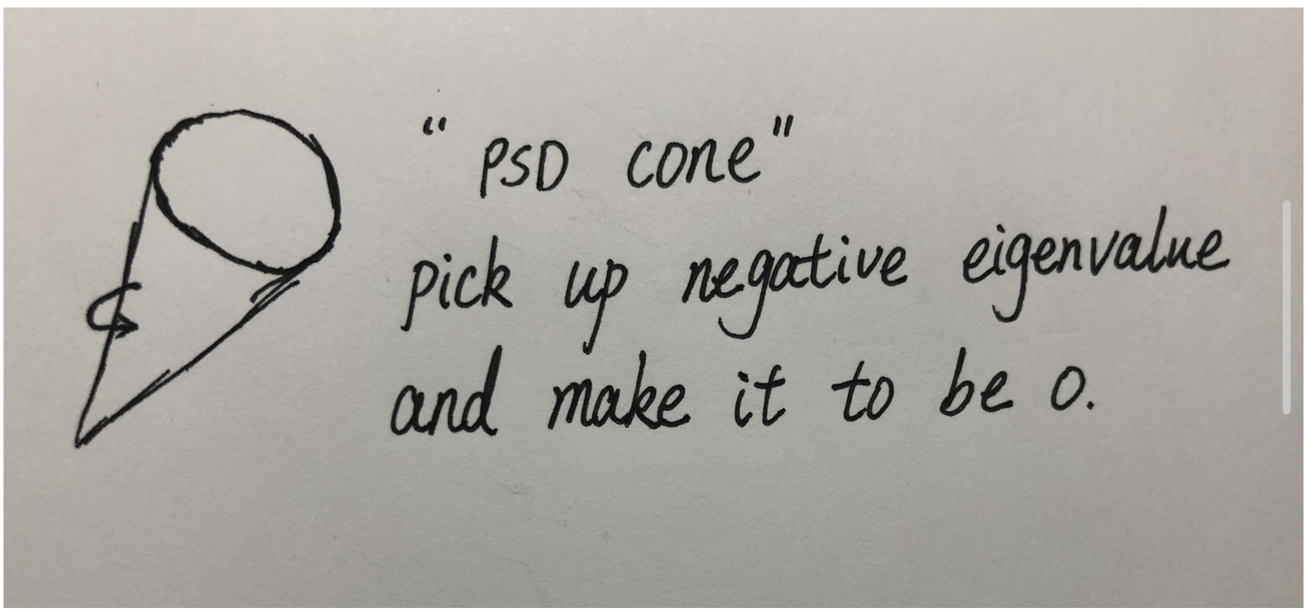

Figure 1

Note2: Rank constraints are $L_0 - type$ and it is non-convex, the nearest convex constraints are $L_1 - type$ i.e. trace constraints($tr(M)$). Therefore, you can replace $rank(M) \leq k$ by $tr(M) \leq k$. However, if rank of L is critical, you have to work with rank(L), making this a Q_2P_2 problem.

3 Independent Component Analysis

Idea: "Maximize the non-gaussian of each dimension"
Example: Try to separate the conversation in a cocktail party using microphone.
Define D: number of microphone; K: number of conversation; T: sound dimension
Let $X = M \times S$, where $X \in R^{D\times T}$ is what you get from all the microphones, $M \in R^{D\times K}$ is the

conversation gained by the microphone. $S \in R^{K \times T}$ is sound signal from K conversations.
Assumption: The assumption is based on CLT, i.e, linear combination of independent random variables is going to be gaussian like. Therefore, X is more gaussian than S(S is independent from each other and X will be more dependent).
Goal: Find WX=S which is less gaussian like.
Question: How to measure gaussian like?
Answer:1.Kurtosis Method 2. Negative Entropy Method 3. Minimize Mutual Information Method

3.1 Kurtosis Method

Define kurtosis for a distribution y, $kertosis(y) := E[y^4] - 3(E[y^2])^2$.
Fact: $g \sim N(0,1) \quad E(g^4) = 3$
$kertosis(y) = 0 \leftrightarrow gaussian$
$kertosis(y) < 0 \leftrightarrow subgaussian$
$kertosis(y) > 0 \leftrightarrow supgaussian$
The objective function:

$$max(kurt(W^T X))^2$$

s.t.

$$var(W^T X) = 1$$

Drawback: Not robust to outliers!

3.2 Negative Entropy Method

Reminder: Entropy $H(y) := -\sum_p P[Y=y] log P[Y=y] = -\int_x plogp\, dx$
Observation:Guassian distribution has least information, i.e. has most entropy of all distribution with the same variance.
The objective function:

$$max - H(W^T X)$$

s.t.

$$Var(W^T X) = 1$$

3.3 Minimize Mutual Information Method

Goal:

$$min \sum_{i<j} I(W_i^T X; W_j^T X)$$

4 Matrix Factorization

Example: Netflix Problem
Description: Suppose we have m users and n movies, each user rates the movies which he has seen. Letr_{ij} be the rating assigned by user i to movies j. Since each user can only rate few movies, The matrix would be super-sparse.

Idea: we assume there are k factors which have vital influence on users and movies, these factors maybe include horror, romance, science, etc.
Define $u_i \in R^k, m_j \in R^k$, then $U \in R^{m\times k}, M \in R^{k\times n}$
Objective function:

$$\min_{U,M} \sum_{r_{ij}\in observed} (r_{ij} - u_i m_j)^2$$

Another way:

$$\min_{U,M} ||R - UM||_F^2$$

5 Manifold Embedding

Definitions:
1. n-dim manifolds: An object $\subseteq R^D$ which locally looks like(homeomorphic) R^n
2. Homeomorphic: continual f and f^{-1} := homeomorphic
3. Diffeomorphic: differentiable f and f^{-1} := diffeomorphic
Manifold hypothesis: $X \subseteq R^D$ measurement are non-linear smoothly related. X is sampled from an underlying(low-dimensional) manifold(perhaps with some noise).
Explain: There are few underlying factors(n independent) which "control" your observations and you make $D \gg n$ different measurement s.t. $x_i \in R^D$.
Figure 2 gives some intuition from R^2 to R^3.

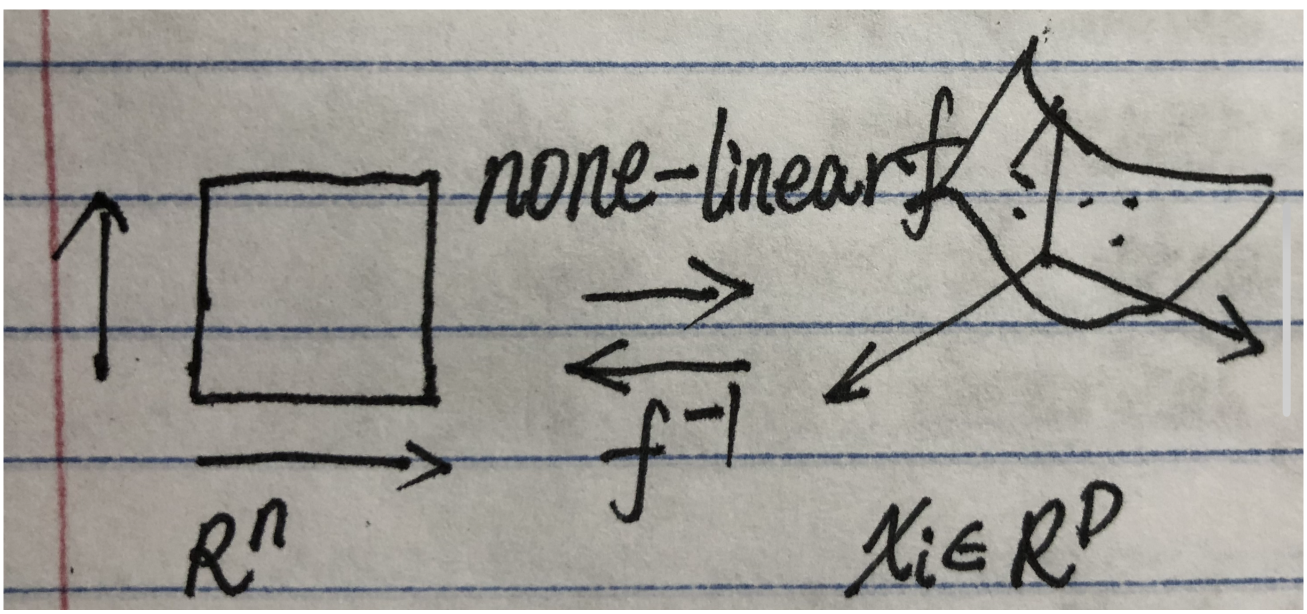

Figure 2

Goal of manifold embedding: find f^{-1} or at least find $f^{-1}(x_i)\ \forall x_i \in X$
Figure 3 gives some intuition from R^2 to R^1.

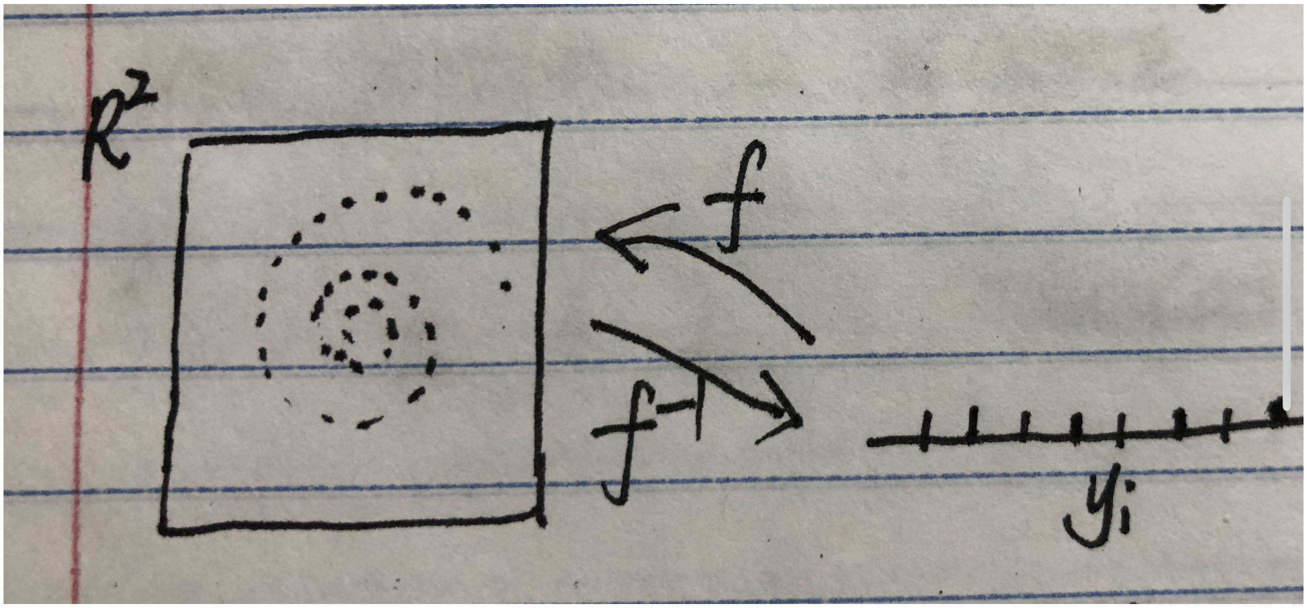

Figure 3

Approach: Isometric mapping
1. Create K-NN graph to approximate geodesic distance.

$$\rho(x_i, x_j) = geo(x_i, x_j)$$

2. Run MDS on the geodesic distance.

$$\min S(y_1, ..., y_n) = \sum_{i<j} (D(y_i, y_j) - \delta_{ij})^2$$

Note: Other approaches such as t-SNE, LLE, Max var unfolding will be discussed in the next few lecture.