Chapter 1

Dimension reduction

1.1 Introduction

"Curse of dimensionality" has long been an essential issue in quantitative research fields. The term was first introduced by Richard Bellman in 1957 when he encountered significant obstacles under large dimension of "state variables" in optimization problems. If 10 equally spaced grids are placed in each dimension of a unit cube, the total number of function evaluation for exhaustive search grows exponentially (e.g. 10^3 for three dimensions and 10^{20} for 20 dimensions). He then argued in favor of his dynamic programming method as a solution for particular optimization problems. In many computational problems, the computing complexity also increases exponentially as the dimension increases. In statistics, another type of "curse of dimensionality" arises when estimation of a density function converges very slowly to the truth in high-dimensional space, which means accurate characterization of a high dimensional data set is generally impossible. For example, when using the Gaussian kernel method to estimate the density of a unit d-dimensional multivariate normal distribution, the required sample sizes were calculated for different d if the interest is to estimate density at point $\mathbf{0}$ and the sample size is large enough so that the relative mean square error $E(\hat{f}(\mathbf{0}) - f(\mathbf{0}))^2/f(\mathbf{0})^2$ is less than 0.1 (Silverman, 1986). In one or two dimensional case, the required sample sizes are only 4 and 19. But when d increases to 10, the sample size needed becomes 842,000. A third commonly seen example is from numerical integration of a general function. When a non-smooth function is evaluated on a high-dimensional space, accurate integration of this function becomes hopeless. The dimensionality issue has gained increasing attention since 1980s when more and more high-dimensional and large volume data are accumulated in the modern world.

High-dimensional data can also present peculiar behaviors. A well-known phenomenon is the so-called “concentration of measure”. For example, consider a standard multivariate (d-dimensional) normal distribution. When d=1 or 2, the density is concentrated near the origin. When d goes large, it is, however, easily shown that the distribution is concentrated on a d-dimensional sphere/shell with radius equals $\sqrt{d}$ (see Exercise 1). Such counter-intuitive behavior reminds that high-dimensional data analysis should be done carefully.

To facilitate analysis of high dimensional data, dimension reduction techniques have been rigorously developed and applied for decades. Principal component analysis (PCA) and sigular value decomposition (SVD) are the most commonly used techniques. Section 1.2.1 below will be devoted to introduce theories and algorithms of this important technique. The PCA technique is, however, only useful for data in Euclidean space and has the drawback that it generates both positive and negative loadings in the many dimensions. Alternatives such as multidimensional scaling (MDS) and non-negative matrix factorization (NMF) will be introduced. In contrast to PCA, MDS can take any dissimilarity matrix (does not even have to be a distance matrix) as the input for dimension reduction and NMF generates dimension reduction with only positive loadings for better interpretation. Section 1.3 introduces two additional dimension reduction methods in the presence of a outcome variable (e.g. when data matrix comes from gene expression profile and the outcome variable is disease subtype or time to recurrence). The classifical Fisher's linear discriminant analysis finds reduced dimensions that best discriminate a binary or multi-class outcome variables. Partial least square (PLS) has similar ability to identify reduced space most associated with a continuous outcome variable. In each subsection, we demonstrate the concept, related theory and algorithm, then followed by a breast cancer gene expression profile example.

1.2 Unsupervised dimension reduction

1.2.1 Principal component analysis and singular value decomposition

Taking gene expression profile as an example. A data matrix with expression intensities of G genes and S samples are available. The data matrix is deonted as

$$\mathbf{M} = [x_{gs}]_{1 \le s \le S, 1 \le g \le G} = [\vec{\mathbf{x}}_1^T, \vec{\mathbf{x}}_2^T, \cdots, \vec{\mathbf{x}}_S^T]$$

where $\vec{\mathbf{x}}_s = (x_{s1}, \cdots, x_{sG})$, G is usually in the order of 4k to 30k, and S is typically 10 to 500. To perform dimension reduction, one may project samples onto a low-dimensioinal space (i.e. view S samples on G-dimensionial space and perform dimension reduction). Alternatively, we can also project genes in S-dimensional space onto a low-dimensional space. In the notations here, samples are to be projected.

Before considering observed empirical data $\mathbf{M}$, we first consider the population (random variable) version. Suppose $\vec{\mathbf{X}} = (\mathbf{x_1}, \ldots, \mathbf{x_G})$ is a random vector and data vector $\vec{\mathbf{x}}_s$ for sample s ($1 \le s \le S$)is an observation from distribution $\vec{\mathbf{X}}$ (i.e. $\vec{\mathbf{x}}_1, \vec{\mathbf{x}}_2, \cdots, \vec{\mathbf{x}}_S \sim^D \vec{\mathbf{X}}$). Without loss of generosity, assume $\mathrm{E}(\vec{\mathbf{X}}) = \vec{\mathbf{0}}$ and $\mathrm{Var}(\vec{\mathbf{X}}) = \mathrm{E}(\vec{\mathbf{X}}^T\vec{\mathbf{X}}) = \mathbf{\Sigma}$. $\mathbf{\Sigma}$ is a $G \times G$ symmetric and positive-definite matrix.

Eigen-decomposition and PCA

For any symmetric and positive-definite matrix Σ, there exists matrix V and diagonal matrix D such that

$$\mathbf{\Sigma} = \mathbf{V}\mathbf{D}\mathbf{V}^T$$

, where

$$\mathbf{D} = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & 0 & \\ & 0 & \ddots & \\ & & & \lambda_G \end{pmatrix}$$

, $\lambda_1 \ge \lambda_2 \ge \ldots \ge \lambda_G \ge 0$ are eigenvalues. $\mathbf{V} = (\mathbf{v}_1^T, \mathbf{v}_2^T, \ldots, \mathbf{v}_G^T)$ are eigen vectors. $\mathbf{v}_i \cdot \mathbf{v}_j^T = 0$ when $i \ne j$ and $|\mathbf{v}_g|^2 = \mathbf{v}_g \cdot \mathbf{v}_g^T = 1$, $1 \le g \le G$.

Properties

1. Orthogonal projected random variables: Define by $L_g = \mathbf{v_g} \cdot \vec{\mathbf{X}}^T$ the random variable that $\vec{\mathbf{X}}$ projects on $\mathbf{v_g}$. Since $\mathbf{V}^T\Sigma\mathbf{V} = \mathbf{D}$, we can show that $Cov(L_i, L_j) = \mathbf{v}_i\mathbf{\Sigma}\mathbf{v}_j^T = 0$ if $i \ne j$ and $Var(L_g) = \mathbf{v}_g\mathbf{\Sigma}\mathbf{v}_g^T = \lambda_g$.

2. Effective reduced linear space: Suppose $r = \text{rank}(\mathbf{\Sigma}) \Rightarrow \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_r > 0$, $\lambda_{r+1} = \ldots = \lambda_G = 0$. Then $\vec{\mathbf{U}} = (\mathbf{u}_1, \cdots, \mathbf{u}_r) = (\frac{1}{\sqrt{\lambda_1}}\mathbf{v}_1\vec{\mathbf{X}}^T, \cdots, \frac{1}{\sqrt{\lambda_r}}\mathbf{v}_r\vec{\mathbf{X}}^T)$ forms a reduced orthonormal basis of the original linear space spanned by $\vec{\mathbf{X}} = (\mathbf{x_1}, \cdots, \mathbf{x_G})$. In general, we hope $r << G$ (and it is usually the case in many applications). The new reduced r-dimensioinal linear random vector space contains all information of the original G-dimensional linear space.

3. Geometric interpretation of PCA: Consider the following optimization problem where we seek for an optimal projection direction for $\vec{\mathbf{X}}$ such that the variance after projection can be maximized :

$$\max_{\alpha} \text{Var}(\alpha\vec{\mathbf{X}}^T) = \alpha\mathbf{\Sigma}\alpha^T, \text{ subject to } |\alpha| = 1.$$

The solution is the first eigenvector $\alpha^* = \mathbf{v}_1$, and $\text{Var}(\alpha^* \cdot \vec{\mathbf{X}}^T) = \mathbf{v}_1\mathbf{\Sigma}\mathbf{v}_1^T = \mathbf{v}_1\mathbf{V}\mathbf{D}\mathbf{V}^T\mathbf{v}_1^T = \lambda_1$
Proof: Suppose $\alpha = \sum a_g\mathbf{v}_g$, $\sum a_g^2 = 1$, $\text{Var}(\alpha\mathbf{x}) = \sum a_g^2\lambda_a$. Therefore, $(a_1, a_2, \ldots, a_G) = (1, 0, \ldots, 0)$

Similarly, L_g has the maximum variance among all possible projections orthogonal to $L_1, L_2, \ldots, L_{g-1}$, for $1 \leq g \leq G$.

4. Definition of power of a symmetric positive-definite matrix: It can be easily shown that $\mathbf{\Sigma}^{1/2} = \mathbf{V}\mathbf{D}^{1/2}\mathbf{V}^T$ (exercise 2). Extending the proof, we can define the power of matrix $\mathbf{\Sigma}$ as $\mathbf{\Sigma}^m = \mathbf{V}\mathbf{D}^m\mathbf{V}^T$ for $m \in \mathbb{R}$.

5. Percent of variance explained in the dimension reduction: The trace($\mathbf{\Sigma}$), the sum of variance of all original variates, is invariant of changes of basis. Particularly, $\text{trace}(\mathbf{\Sigma}) = \sum \text{Var}(L_g) = \sum \lambda_g$. Define $r_g = \frac{\lambda_g}{\sum \lambda_{g'}}$. r_g represents the proportion of total variance explained by the gth PC. $R_g = \sum_{g'=1}^{g} r_{g'}$ is the cumulative proportion of variance explained by the first g PCs. This is a good index to quantify goodness of dimension reduction.

6. Caution 1: In PCA, demension reduction is achieved by removing dimensions with small variance. But there is no guarantee that the removed dimensions are not essential to the data. In many cases, essential data structure and information can dissapear (e.g. two skewed clusters example in Exercise 7).

7. Caution 2: PCA potentially has standardization issues. If $\text{Var}(\mathbf{x_g})$ are in wide range, standardization should be considered to avoid

dominance of this variable. Consider standardization $\mathbf{x_g^{(s)}} = \frac{1}{\sqrt{\mathrm{Var}(\mathbf{x_g})}}\mathbf{x_g}$. The covariance matrix $\mathbf{\Sigma}^{(s)} = \mathrm{Var}(\mathbf{x_g^{(s)}})$ becomes the correlation matrix and the eigen decomposition is denoted as

$$\mathbf{\Sigma}^{(s)} = \mathbf{V}^{(s)}\mathbf{D}^{(s)}\mathbf{V}^{(s)^T}$$

There is no simple connection between $\mathbf{V}$ and $\mathbf{V}^{(s)}$ (see Exercise 8). That is, PCA on covariance matrix and correlation matrix perform very differently. To standardize or not require careful consideration based on the data.

8. Theoretical convergence under normal assumption: PCA generally does not require normal assumption. Under normal assumption, inference can be made about $\hat{\lambda}_k$ and $\hat{\mathbf{v}}_k$ (e.g. see the theorem below). Under normal assumptions, PCs are independent; without normal assumption, PCs are linearly uncorrelated but could be dependent.

   Theorem (p473, Anderson 2003) Suppose $\vec{\mathbf{X}} \sim N(\mu_G, \Sigma_{G\times G})$, $\mathbf{M}$ is an observed $S \times G$ data matrix from $\vec{\mathbf{X}}$, $\hat{\Sigma}$ is the sample covariance matrix of $\mathbf{M}$ and $\hat{\lambda}_1, \cdots, \hat{\lambda}_G$ are the (distinct) eigenvalues of $\hat{\Sigma}$. Asymptotically, $\hat{\lambda}_g$ is distributed as follows (Exercise 3):

$$\sqrt{n}(\hat{\lambda}_g - \lambda_g) \to N(0, 2\lambda_g^2), \text{for g=1,...,G}$$

Methods to compute eigenvectors and eigenvalues

Characteristic polynomial: To compute eigenvalues and eigenvectors, the characteristic polynomial can be used. This method applies a property that a eigenvector x of $\mathbf{\Sigma}$ should satisfy

$$\mathbf{\Sigma}v = \lambda v.$$

In other words, the eigenvector x does not rotate under transformation of $\mathbf{\Sigma}$ but only scale by λ. This is equivalent to solving $(\mathbf{\Sigma} - \lambda I)v = 0$. or $det(\mathbf{\Sigma} - \lambda I) = 0$. The determinant is a G-degree of polynomial function and the problem converts to finding its roots. This method is, however, often too slow unless G is small.

Power iteration: Power method focuses on finding eigenvectors such that $\mathbf{\Sigma}v = \lambda v$. The method starts with a random vector $v^{(0)}$. It then iteratively calculates $v^{(i)} = \frac{\mathbf{\Sigma}v^{(i-1)}}{|\mathbf{\Sigma}v^{(i-1)}|}$ until the vector converges. The vector usually converges quickly (although the exact converence rate depends on $\mathbf{\Sigma}$) to the eigenvector corresponding to the largest eigenvalue (Exercise 4 to prove it!!). The contribution of the identified eigenvector v_1 is then subtracted from $\mathbf{\Sigma}$ (i.e. $\mathbf{\Sigma}' = \mathbf{\Sigma} - \lambda_1 v_1^T v_1$). The process is repeated to find the next eigenvector.

Exercise 5 will provide a practice to program characteristic polynomial and power iteration algorithms and compare to R standard routine. For eigen decomposition of a very large matrix, the power iteration is still not effective enough. Faster algorithms are available, including QR algorithm and the method combining Householder transformation with LU decomposition. (??add references??)

SVD and relation to eigen-decomposition

Assume data matrix $\mathbf{M} = [x_{gs}]_{1 \le g \le G, 1 \le s \le S}$. In gene expression profile examples, we normally have small sample size compared to large number of genes (i.e. $S << G$, a problem often called "small-n-large-p"). In contrast to eigen-decomposion that works on symmetric positive-definite matrixes, SVD technique directly decompose any $G \times S$ data matrix $\mathbf{M}$ as

$$\mathbf{M} = \mathbf{U} \begin{pmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \mathbf{V^T}.$$

Suppose $1 \le r = rank(\mathbf{M}) \le \min(S, G)$. Then $\mathbf{D}$ is an $r \times r$ diagonal matrix, $\mathbf{U}$ is a $G \times G$ orthogonal transformation matrix and $\mathbf{V}$ is a $S \times S$ orthogonal transformation matrix. Suppose $\mathbf{D} = \begin{pmatrix} \delta_1 & & 0 \\ & \ddots & \\ 0 & & \delta_r \end{pmatrix}$, $\delta_1 \ge \delta_2 \ge \ldots \ge \delta_r > 0$.

- Consider $\mathbf{MM}'$ ($G \times G$ matrix)

  $$\mathbf{U}'\mathbf{MM}'\mathbf{U} = \mathbf{U}' \left(\mathbf{U} \begin{pmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \mathbf{V}'\mathbf{V} \begin{pmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \mathbf{U}' \right) \mathbf{U} = \begin{pmatrix} \mathbf{D}^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}_{G \times G}$$

  Denote eigenvalues of $\mathbf{MM}'$ as $(\lambda_1, \lambda_2, \ldots, \lambda_r, 0, \ldots, 0) \Rightarrow \lambda_i = \delta_i^2$, $1 \le i \le r$.
  Similarly, $\mathbf{V}'\mathbf{M}'\mathbf{MV} = \begin{pmatrix} \mathbf{D}^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}_{S \times S}$

- Suppose $\mathbf{U}^* = \left((\mathbf{u}^{(1)})^T, \ldots, (\mathbf{u}^{(r)})^T\right)_{G \times r}$, $\mathbf{V}^* = \left((\mathbf{v}^{(1)})^T, \ldots, (\mathbf{v}^{(r)})^T\right)_{S \times r}$.

  $$\mathbf{M} = \mathbf{U}^*_{G \times r} \mathbf{D}_{r \times r} (\mathbf{V}^*)^T_{r \times S}$$

  $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(r)}$ are called eigen-samples.
  $\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(r)}$ are called eigen-genes.

Computational advantage of SVD

Consider eigen decomposition of $\mathbf{MM}'$ (a $G \times G$ matrix). Since $S << G$, the eigen decomposition is quite time consuming. Alternatively, one can solve eigen-decomposition of $\mathbf{M}'\mathbf{M}$ first (a $S \times S$ matrix). Identify rank, $\mathbf{V}^*$ and $\mathbf{D}^2$ such that $\mathbf{M}'\mathbf{M} = \mathbf{V}^*\mathbf{D}^2\mathbf{V}^{*'}$. $\mathbf{U}^*$ can be easily generated as $\mathbf{U}^* = \mathbf{MV}^*\mathbf{D}^{-1}$. Finally, we have eigen decomposition of $\mathbf{MM}' = \mathbf{U}^*\mathbf{D}^2\mathbf{U}^{*'}$

With SVD, one can also calculate the Moore-Penrose inverse of $\mathbf{M}$:

$$\mathbf{M}^{-1} = \mathbf{V}\begin{pmatrix} \mathbf{D}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}\mathbf{U}'$$

It is interesting to see that $M^{-1} \neq (M^T)^{-1}$ under the Moore-Penrose inverse since the two inverse matrixes do not even have the same dimension.

Approximation by low-rank matrix

When the first r eigenvalues are apparently greater than zero and the remaining $S - r$ eigenvalues are near zero (but not exactly zero), one may suspect that $\mathbf{M}'\mathbf{M}$ may have rank r and the near-zero eigenvalues are actually zero with numerical rounding errors. One may attempt to approximate $\mathbf{M} = \mathbf{UDV}'$ by $\tilde{\mathbf{M}} = \mathbf{U}\tilde{\mathbf{D}}\mathbf{V}'$ where $\tilde{\mathbf{D}}$ is a diagonal matrix equivalent to $\mathbf{D}$ except that the last $S - r$ near-zero eigenvalues are replaced by zero. This intuition is actually valid and supported by the Eckart-Young theorem for low-rank approximation:

Eckart-Young theorem Consider the approximation of $\mathbf{M}$ by any possible r-rank matrix $\mathbf{Z}$. Denote by $||\mathbf{M} - \mathbf{Z}||_F$ the Frobenius norm (which is defined as the square-root of the summation of the squared modulus of all entries in a matrix) of matrix $\mathbf{M} - \mathbf{Z}$. We can show that $\arg\min_Z ||\mathbf{M} - \mathbf{Z}||_F = \tilde{\mathbf{M}}$. That is to say, the best r-rank matrix to approximate $\mathbf{M}$ is exactly the matrix that replaces the $S - r$ smallest eigenvalues as zero (Exercise 6).

A yeast cell-cycle transcriptomic example

Cell cycle is a fundamental mechanism in the control of cell proliferation. A cell often starts from a resting phase G_0 where the cell has left the cycle and stops division. In the first gap G_1 phase, the cell starts to increase in size. The cell then starts to replicate DNA in the synthesis S phase. In the second gap G_2 phase, the cell continues to grow. Finally, the cell divides into two in the mitosis M phase. Several check-point mechanisms in the stages of cell cycle are crucial to examine abnormal mutation, DNA damage and errors in DNA replication. Failure of normal cell cycle controls can accumulate the genetic errors and lead to cancers. Figure 1.1 shows a diagram of the four cell cycle phases.

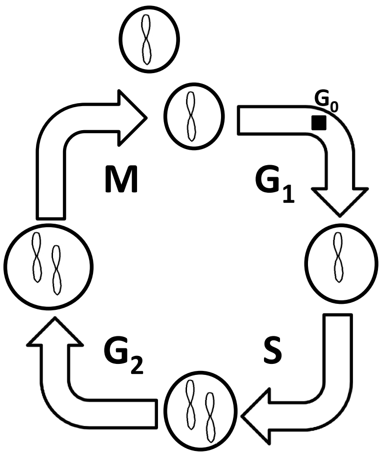

Figure 1.1: The four cell cycle phases.

Spellman et al. (1998) generated a series of yeast cell cycle microarray experiments to identify cell cycle related genes. Yeast cells are first synchronized to the same G_0 stage by four different chemicals: alpha, cdc15, cdc28 and elu. The cells are then released into cell cycle and the array experiments are performed in incremental time intervals (every 7 minutes for alpha, every 10-20 minutes for cdc15, every 10 minutes for cdc28 and every 30 minutes for elu). The data set (obtained from http://genome-www.stanford.edu/cellcycle/) contains the expression profile of 800 genes and their annotated cell cycle stage ("G_1", "S", "S/G_2", "G_2/M" and "M/G_1"). Figure 1.2 shows the PCA projection of the samples using different synchronization chemicals. The result shows clear cyclic patterns in all four synchronization methods. On the other hand, Figure 1.3 shows the PCA projection of the genes (genes are labelled with different annotated cell cycle stages). The result shows clear separation of genes of different stages and a smooth transition from G_1 to S and then S/G_2, G_2/M and M/G_1. See Exercise 11 for steps to reproduce these results.

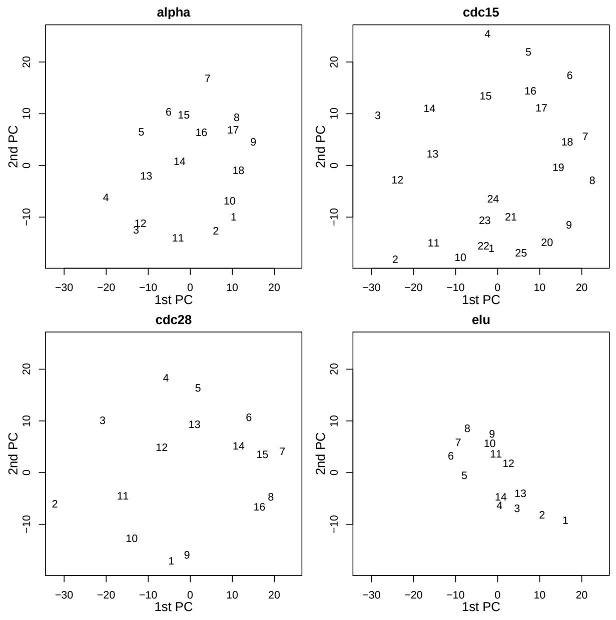

Figure 1.2: PCA projection of the samples (numbers are sequences of time points).

1.2.2 Multi-dimensional scaling

Multi-dimensional scaling (MDS) aims to map data from high dimension to a low-dimensional space (usually 2D or 3D Euclidean space) such that the distance (or dissimilarity) strucutre is best preserved. The loss function to be minimized is

$$L = \min \sum_{i<j} (d_{ij} - \delta_{ij})^2,$$

where δ_{ij} is the dissimilarity measure between object i and j in the original data space and d_{ij} is the distance between the two objects after mapping to the targeted low-dimensional (usually Euclidean) space. The first classical MDS (CMDS) development (Torgerson, 1954) considered a symmetric and quantitative dissimilarity matrix with no missingness (see pp121, Alpaydin 2004). It also required the data at the ratio level of measurement but this requirement was later generalized to interval level (Messick, 1956) (see definition of ratio and interval scales in Appendix). In the loss fuction above, large distances can dominate the optimization

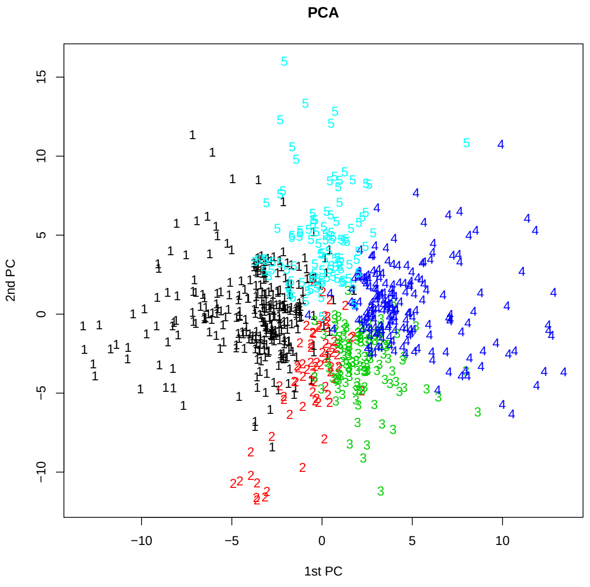

Figure 1.3: PCA projection of the genes. (1: G_1; 2: S; 3: S/G_2; 4: G_2/M; 5: M/G_1)

and ignore the structure for pairs of short distances. A standardized loss function below is called Sammon's stress (Sammon, 1969) and is usually preferred to capture local structures:

$$L = \min \frac{1}{\sum_{i<j} \delta_{ij}} \sum_{i<j} \frac{(d_{ij} - \delta_{ij})^2}{\delta_{ij}}.$$

The Sammon's stress can be explained as an averaged and standardardized mapping error and is a good measure to quantify success of dimension reduction. The minimization can be performed by gradient descent or other iterative methods.

Table 1.1 lists the flight mileage of ten cities obtained from "http://www.webflyer.com/travel/mileage_calculator/". Since cities are located on the globe and the mileages are calculated as flight mileages, the table provides a dissimilarity matrix of the ten cities rather than an Euclidean distance matrix. Several R functions are available to implement different types of MDS algorithms: cmdscale, sammon and isoMDS. Figure 1.4 shows the MDS mapping of the ten cities on a 2D Euclidean space (Exercise 12). Although classical MDS (cmdscale) and Sammon mapping (sammon) gave seemingly similar results, they differ greatly in local structures. For example, the distances between Pittsburgh and DC and between Pittsburgh and Chicago are 183 and 394 miles, respectively. The mapping from cmdscale is quite distorted while the Sammon mapping better preserve this local distance structure. In this example, the Sammon stress is as low as 0.0042, showing a good preservation of the dissimilarity structure in the mapping.

Note that any "reflection, translation and/or rotation" (i.e. isometry) of an MDS solution is also an MDS solution since MDS only considers the preservation of the dissimilarity structure.

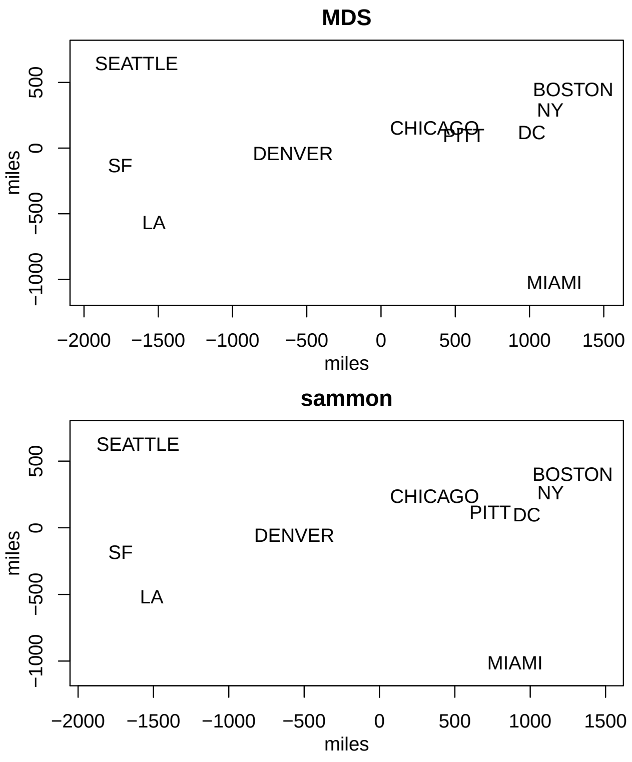

Figure 1.4: MDS mapping of the ten cities on a 2D Euclidean space. Upper: classical MDS; lower: sammon's stress.

Below we go over an algorithm for the simplest "classical mulditimen-

Table 1.1: Flight mileages between pairs of ten cities.

	BOST	NY	DC	MIAM	CHIC	SEAT	SF	LA	DENV	PITT
BOSTON	0	206	429	1504	963	2976	3095	2979	1949	495
NY	206	0	233	1308	802	2815	2934	2786	1771	340
DC	429	233	0	1075	671	2684	2799	2631	1616	183
MIAMI	1504	1308	1075	0	1329	3273	3053	2687	1010	2037
CHICAGO	963	802	671	1329	0	2013	2142	2054	996	394
SEATTLE	2976	2815	2684	3273	2013	0	808	1131	1307	2120
SF	3095	2934	2799	3053	2142	808	0	379	1235	2250
LA	2979	2786	2631	2687	2054	1131	379	0	1059	2130
DENVER	1949	1771	1616	2037	996	1307	1235	1059	0	1280
PITT	495	340	183	1010	394	2120	2250	2130	1280	0

sional scaling" where the mapping is equivalent to PCA eigen-decomposition. Suppose the observed dissimilarity matrix $\mathbf{D} = \{\delta_{ij}\}_{1 \le i,j \le n}$ is derived from the Euclidean distances of an unknown $n \times q$ data matrix $\mathbf{X}$ (q is also unknown). Given $\mathbf{D}$, how can we identify the original data matrix $\mathbf{X}$? Note again that any rotation and reflection of a solution $\mathbf{X}$ produces the same $\mathbf{D}$ and is also a soultion. We fix the center of each column in $\mathbf{X}$ at origin (i.e. $\sum_{i=1}^n x_{ik} = 0$, for $1 \le k \le q$).

Define $\mathbf{B} = \mathbf{X}\mathbf{X}^T$. The elements of $\mathbf{B}$ can be written as linear combinations of elements of $\mathbf{X}$: $b_{ij} = \sum_{k=1}^q x_{ik} \cdot x_{jk}$. It can be shown that $d_{ij}^2 = b_{ii} + b_{jj} - 2b_{ij}$ and finally $b_{ij} = -\frac{1}{2}[d_{ij}^2 - d_{i\cdot}^2 - d_{\cdot j}^2 + d_{\cdot\cdot}^2]$, where $d_{i\cdot}^2 = (\sum_{j=1}^n d_{ij}^2)/n$, $d_{\cdot j}^2 = (\sum_{i=1}^n d_{ij}^2)/n$ and $d_{\cdot\cdot}^2 = (\sum_{i=1}^n \sum_{j=1}^n d_{ij}^2)/n^2$. Now eigen-decomposition of $\mathbf{B}$ can rewrite it as $\mathbf{B} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$. If the original data matrix $\mathbf{X}$ is indeed of rank q, the eigenvalues in $\mathbf{\Lambda}$ will show exactly q non-zero eigen-values. The final mapping solutation is $\mathbf{X} = \mathbf{V}\mathbf{\Lambda}^{1/2}$. Exercise 13 asks to complete the proof and code the implementation of this algorithm.

1.2.3 Non-negative matrix factorization

(This section is now copied and reorganized from Wikipedia.) Non-negative matrix factorization (NMF) is a group of algorithms where a non-negative matrix, $\mathbf{X}$, is factorized into two matrices, $\mathbf{W}$ and $\mathbf{H}$ with the constraint that the factors W and H must be non-negative, i.e., all elements must be equal to or greater than zero. Usually the number of columns of W and the number of rows of H in NMF are selected so the product WH will become an approximation to X. The full decomposition of X then amounts to the two non-negative matrices W and H as well as a residual U, such that: X = WH + U. The elements of the residual matrix can either be negative or positive. The method has been applied in chemometrics, text mining and bioinformatics (Devarajan, 2008).

There are different types of non-negative matrix factorizations. The different types arise from using different cost functions for measuring the divergence between X and WH and possibly by regularization of the W and/or H matrices. Two simple divergence functions studied by Lee and Seung are the squared error (or Frobenius norm) and an extension of the Kullback-Leibler divergence to positive matrices (the original Kullback-Leibler divergence is defined on probability distributions). Each divergence leads to a different NMF algorithm, usually minimizing the divergence using iterative update rules. The factorization problem in the squared error version of NMF may be stated as: Given a nonnegative matrix $\mathbf{X}$ find nonnegative matrices $\mathbf{W}$ and $\mathbf{H}$ such that $F(\mathbf{W}, \mathbf{H}) = |\mathbf{X} - \mathbf{WH}|_F^2$ is minimized.

1.2.4 Comparison of PCA, MDS and NMF

The three unsupervised dimension reduction methods are useful in many applications. They have their own pros and cons depending on different applications. In general, PCA and NMF look for an optimal reduced space to project to. NMF differs from PCA in that the loadings are restricted to be positive for better interpretation. MDS, on the other hand, focuses on preserving the dissimilarity (distance) structure of the data. PCA and NMF work only on data sets of high-dimensional Euclidean space. MDS is more flexible for any data set where the dissimilarity of every pair of objects (dissimilarity matrix) is defined. PCA and NMF "project" the data onto a low-dimensional space while MDS "maps" the data instead. A key consequence is that a newly added subject can be easily projected to the existing PCA or NMF induced dimension reduction. For MDS, additional ad hoc optimization procedure will be needed to map the new subject to the reduced space. We note again that dimension reduction always runs the risk of losing important information in the data. Figure 1.5 shows an example of two skewed clusters where dimension reduction by PCA loses the cluster structure (Exercise 7).

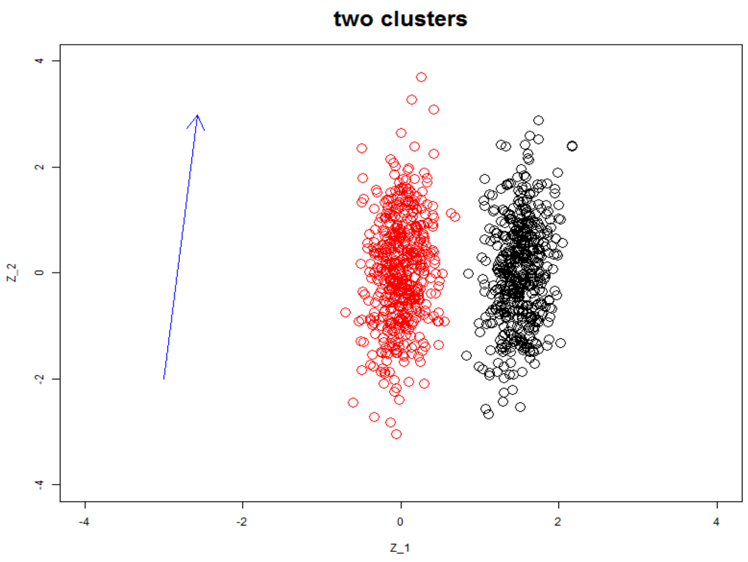

Figure 1.5: An example of two thin clusters where PCA loses the cluster information.

1.3 Supervised dimension reduction (with response variable)

When a response variable is available to the data, it is possible to perform optimal dimension reduction to best visualize the spread of the response variable in the observations. Fisher's linear discriminant analysis is applicable for discrete class labels (e.g. patients of several disease subtypes) and partial least squares (PLS) method is useful for general continuous response variable. Below we illustrate the methods starting from finding the first reduced dimension.

1.3.1 Fisher's linear discriminant analysis

When class labels are available, Fisher's discriminant analysis attempts to project data onto a subspace such that groups of different class labels are optimally separated. Denote by $B = \sum_c (\mu_c - \bar{x})(\mu_c - \bar{x})^T$ the between-class variance of the data (where μ_c is the mean of class c and $\bar{x}$ is the overall mean) and $W = \sum_c \sum_{i \in c} (x_i - \mu_c)(x_i - \mu_c)^T$ the within-class variance. Consider the following target function

$$\max_a \frac{a^T B a}{a^T W a}.$$

The numerator $a^T Ba$ represents the between-class variance after the data are projected to direction a. Similarly, the denominator $a^T W a$ represents the with-class variance of data projected to a. As a result, the target function aims to maximize the between-class separation while minimize the within-class concentration. Note that the target function is scale invariant (i.e. the target function is invariant for scaling from a vector a to $c \cdot a$ for a constant c). Perform eigen-decomposition for $W = VDV^T$ such that $W^{\frac{1}{2}} = VD^{\frac{1}{2}}V^T$ and $W^{-\frac{1}{2}} = VD^{-\frac{1}{2}}V^T$. Define $b = W^{\frac{1}{2}}a$. The target function can be rewritten as

$$\max_b \frac{b^T (W^{-\frac{1}{2}})^T B W^{-\frac{1}{2}} b}{b^T b}.$$

Define $B^* = (W^{-\frac{1}{2}})^T B W^{-\frac{1}{2}}$. The problem becomes a simple eigendecomposition of B^* to obtain eigenvectors (denoted as $b_1, b_2, ...$). Solution of a can be calculated as $a_i = W^{-\frac{1}{2}} b_i$.

1.3.2 Partial least square

Partial least square regression originated from social sciences (Wold, 1966). It was first presented similar to the power method but then the statistical framework was later established. In contrast to PCA, the method aims to find the projection direction such that the covaraince of projected data with the class label or continuous outcome Y is the largest:

$$\max_w \mathrm{Cov}(w\vec{\mathbf{X}}^T, \mathbf{Y}), \text{ subject to } ||w|| = 1.$$

In general, if Y is also multi-variate, PLS performs the following optimization:

$$max(Xw)^T(Yv),$$

subject to the constraints $||w|| = w^T w = 1$ and $||v|| = v^T v = 1$.

For more references, check "http://www.ats.ucla.edu/stat/sas/library/pls.pdf".

1.4 Association between predictors and responses

Consider a multi-variate linear regression

$$Y_i = \vec{\beta} \cdot \vec{X}_i, 1 \leq i \leq n$$

where Y_i is the response (dependent) variable and $\vec{X}_i$ contains p covariates (independent variables). When p is small, n is large and X is full rank (i.e. no colinearity between covariates), conventional multi-variate linear regression can work well. When p is very large (very common in genomic applications), $\vec{X}$ is often singular and the regression result is unstable (multi-collinearity). One conventional solution is to perform eigen-decomposition of $\vec{X}$ and select the top eigen-vectors as predictors. This approach (often called PCA regression) selects eigenvectors as covariates that best explain $\vec{X}$ but has no guarantee that they will explain Y well. Alternatively, PLS identifies components from $\vec{X}$ such that it explains as much as possible of the covariance between $\vec{X}$ and Y and the PLS components are used for regression (so-called PLS regressioin). See Nguyen and Rocke (2002) for a microarray class prediction example using PLS dimension reduction.

See “http://www.casact.org/pubs/dpp/dpp08/08dpp76.pdf” for more details of PCA regression and PLS regression.

Exercise:

1. (a) Consider $X = (X_1, \cdots, X_d)^T = N((\mu_1, \cdots, \mu_d)^T, I_d)$. Show that the L2-norm $||X||_2 = \sqrt{X_1^2 + \cdots + X_d^2}$ converges to $\sqrt{d}$ when d goes to infinity. The convergence is actually very fast. This result shows that the high-dimensional standard multivariate normal distribution is populated in concentration on the sphere with radius $\sqrt{d}$. (b) Perform simulation to verify (a). For a given d, simulate 1000 observations from X and calculate their L2-norms. Simulate d from 1 to 100. Plot a scatter plot of d on the x-axis and the box-plot of the L2-norms of data for each d on the y-axis. Draw a line of $y = \sqrt{d}$. The result should show a fast convergence.

2. Suppose a symmetrix positive-definite matrix Σ and its eigen-decomposition $\Sigma = VDV^T$. Prove that $\Sigma^m = VD^mV^T$, $\forall m \in \mathbb{N}$. Extend the proof to $\forall m \in \mathbb{R}$.

3. Prove the theorem that derived eigenvalues convergence to the true eigenvalues at the rate of $\sqrt{n}$ under the multivariate normal assumption.

4. Prove that the power itenration method will converge to find the eigenvector with the largest eigenvalue if the initial random vector is not orthogonal to it.

5. (a) Write an R routine to implement the characteristic polynomial algorithm for eigen decoomposition. (b) Write an R routine to implement the power iteration algorithm for eigen decomposition. (c) Apply both routines to a small example and compare the result with the R standard routine "eigen".

6. Prove Eckart-Young theorem for low-rank matrix approximation.

7. (a) Simulate a two-cluster skewed distribution. Assume $Z = pX = (1-p)Y$. $p \sim Binomial(1, 0.5)$,
$$X = (X_1, X_2)^T \sim N\left(\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}0.05 & 0.045\\0.045 & 1\end{pmatrix}\right)$$
and
$$Y = (Y_1, Y_2)^T \sim N\left(\begin{pmatrix}1.5\\0\end{pmatrix}, \begin{pmatrix}0.05 & 0.045\\0.045 & 1\end{pmatrix}\right)$$
Simulate 1000 observations. (b) Perform PCA method and plot the first PC on the scatter plot, and the PC projection of one-dimension. (c) Perform MDS, Fisher discriminant analysis and PLS to reduce the two-dimensional data to one-dimension. Compare their performance in maintaining the two-cluster structure.

8. Following the simulation in Exercise 7, standardize the data and perform PCA (i.e. perform eigen-decomposition on correlation matrix instead of covariance matrix). How does the result differ?

9. (a) Load in "iris" data set from R package "datasets". Perform PCA to project the first four numeric variables to a two-dimensional space. Draw scatter plot for the projected data. Label the classes (the fifthe variable in the data) with different colors or labels. (b) Apply power iteration method to perform PCA in (a). (c) Similarly perform partial least square and Fisher's discriminant analysis to the iris data.

10. Breast cancer used to be considered as one type of cancer. Recent molecular studies have found that it contains several distinct subtypes with different prognosis and desired treatment strategies. Gene expression profile is a powerful tool for a genome-wide classification tool towards "personalized medicine". In this exercise, the

breast cancer gene expression data from TCGA with five breast cancer subtype labels are prepared in “TCGA_breastCancer.txt”. Load in the data, perform PCA to project samples and label samples of different subtypes with different colors.

11. Load the tab-delimited ”yeastCellCycle.txt” file into R. The sample annotation denotes the type of chemicals used for cell cycle synchronization (alhpa, cdc15, cdc28 and elu). (a) Perform SVD and project the samples from 800 dimensional space onto a 2D space. Reproduce Figure 1.2 by drawing the scatter plots with samples of the four chemicals separately. (b) Project genes to a 2D eigen space and reproduce Figure 1.3. (c)Redo dimension reduction on samples in (a) using MDS. (d) Redo dimension reduction on genes in (b) using MDS.

12. Load the tab-delimited ”Mileage.txt.txt” file into R and using R functions to perform MDS analysis to map the ten cities onto a 2D Euclidean space.

13. (a) Complete the proof of the classical multidimensional scaling (CMDS) algorithm. (b) Write an R function for CMDS based on the algorithm. Simulation a small data set and compare the result of your function with the “cmdscal” function in R.

Appendix:
Psychologist Stanley Smith Stevens categorized all measurements in science into four types of scales: nominal, ordinal, interval and ratio (Stevens, 1946). Nominal scale uses ”labels” instead of ordered or quantitative scale. Examples include “sex”, “race” and “color”. Variables assessed on a nominal scale are called categorical variables. Ordinal scale describes data in rank order, rather than relative size or degree of differences. Examples include “tumor grade” and “pain scale”. Interval scale considers variables with meaningful difference between the levels. A famous example of interval scale is temparature. It is defined as 0 and 100 degree at freezing and boiling water and the unit (degree) is defined as 1/100 temparature difference between the two extremes. Note that IQ score should be considered ordinal but not interval scale since difference between 160 and 130 IQ scores is not comparable tween 130 and 100 IQ scores. Ratio scale covers most measurement in physical sciences and engineering that defines the measure as the ratio between the observed magnitude and the unit magnitude. Mass, length and angle are all examples. The major difference between interval and ratio scales is that calculating ratio between two observations is meaningful in ratio scale but not in interval scale. For

example, taking ratio between 40 degree and 20 degree is meaningless.

References

Richard Ernest Bellman; Rand Corporation (1957). Dynamic programming. Princeton University Press. ISBN 9780691079516.

Torgerson. W. S. (1952). Psychometrika. 17. 401-419.

Messick. S. J. and Abelson. R. P. (1956). Psychometrika. 21, 1-17.

Sammon, Jr., J. W. (1969) A nonlinear mapping for data structure analysis. IEEE Transactions on Computers, 18:401-409.

Stevens, S.S (June 7, 1946). "On the Theory of Scales of Measurement". Science 103 (2684): 677680.

B. W. Silverman (1986). Density Estimation for statistics and data analysis. Chapman & Hall.

Karthik Devarajan (2008). Nonnegative Matrix Factorization: An Analytical and Interpretive Tool in Computational Biology. PLoS Computational Biology. 4(7): e1000029.

Nguyen DV, Rocke DM. (2002) Tumor classification by partial least squares using microarray gene expression data. Bioinformatics. 2002 Jan;18(1):39-50.