2.160 IDENTIFICATION, ESTIMATION, AND LEARNING

LECTURE NOTES NO. 4

4. Principal Components and Partial Least Squares Regression

In the previous chapter on Least Squares Estimate we have assumed that the regressor φ's span the whole *m*-dimensional vector space so that the matrix $\sum \varphi(t)\varphi^T(t)$ may be non-singular. This assumption is questionable, when we deal with a high dimensional input space, where the regressor vector contains a large number of components; $m \gg 1$. If we apply the standard linear regression: $\hat{y} = \hat{\theta}^T \varphi$, the same large number of parameters must be determined: $\theta \in \Re^{m\times 1}$.

We may face such a situation in several scenarios.

- One is the growing application areas where abundant sensor information is available. Nowadays inexpensive sensors connected to a network – Internet of Things (IoT), for example - are available everywhere. Costs for acquiring, transmitting, and storing sensor data have reduced dramatically. Extracting useful information from various observed data has become an important issue thanks to technological advance in Sensor Network, Cloud Computing, and Big Data analysis.
- Another situation is the lack of sufficient samples. It is sometimes infeasible or difficult to obtain a large number of cohesive and consistent samples, or it is costly to run many experiments. Biological experiments and clinical trials, for example, are often difficult to repeat many times. The number of sensors, on the other hand, may be increased to observe target phenomena from various perspectives. This may lead to the situation where the dimension of the regressor vector is larger than the number of data samples: $m > N$. As a result, the matrix $\sum \varphi(t)\varphi^T(t)$ becomes singular.

In this section we will address issues associated with high dimensional input data, and introduce a new methodology for extracting significant signal components from the raw data. In other words, we aim to find a small set of variables, referred to as ***Latent Variables***, which are encapsulated in the data but play significant roles in predicting the output. Since the dimension of Latent Variable space is low, the number of parameters to estimate is small.

Before moving into theoretical development, we introduce a preliminary data processing, which is a common practice in ***Statistical Multivariate Analysis***. First, we remove the mean of each variable from the original data: $\varphi_j - \bar{\varphi}_j$, and then normalize it based on its variance σ_j^2. See Figure 1. We deal with the following Mean-Centered, Normalized data:

$$x_j = \frac{\varphi_j - \bar{\varphi}_j}{\sigma_j} \tag{1}$$

A column vector $\boldsymbol{x}$ contains all the components x_j: $x = \left(x_1, \cdots, x_m\right)^T$. Furthermore, *N* samples of $\boldsymbol{x}$ are arranged in an *m* by *N* matrix $\boldsymbol{X}$:

$$X = \left[x^1, \cdots, x^N\right] \in \Re^{m\times N} \tag{2}$$

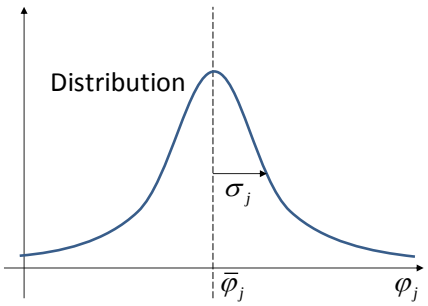

Figure 1 Data distribution

where super scripts 1,…*N* represent sample numbers[1].

In the following we will study three methods for dealing with the large input space problem. We begin with multi-input, single-output problems followed by multi-input, multi-output problems.

4.1 Single-Output, Principal Components Regression

The first method is Principal Components Regression, consisting of two steps of procedure.

Step 1 *Reduction of the Input Space*

The rank of input sample matrix $\boldsymbol{X}$ is less than *m* when *m* > *N*. This implies that the information contained in the *m*-dimensional vectors, $x^1, \cdots, x^N$, can be expresses in a lower dimensional space. To reduce the dimensionality let us first examine how the mean-centered input sample vectors are distributed in the *m*-dimensional vector space. See Figure 2. Let $v \in \Re^{m\times 1}$ be a unit column vector in the *m*-dimensional vector space. The strength of each input sample in the direction of *v* can be expressed with the projection of the sample point onto the unit vector. In total, the squared strength of all the *N* samples in the direction of the unit vector is given by

$$J = \sum_{i=1}^{N} \left| v^T x^i \right|^2 = \sum_{i=1}^{N} v^T x^i (x^i)^T v = v^T \left(\sum_{i=1}^{N} x^i (x^i)^T \right) v = v^T X X^T v \qquad (3)$$

Examine in which direction of *v* this squared strength of the *N* samples becomes maximum.

$$\max_v J(v) \quad \text{subject to } |v| = 1 \qquad (4)$$

This is a type of conditional optimization problem that can be solved with use of Lagrange's multiplier λ.

$$L(v) = v^T X X^T v - \lambda (v^T v - 1) \qquad (5)$$

The necessary conditions for this to be maximal are given by

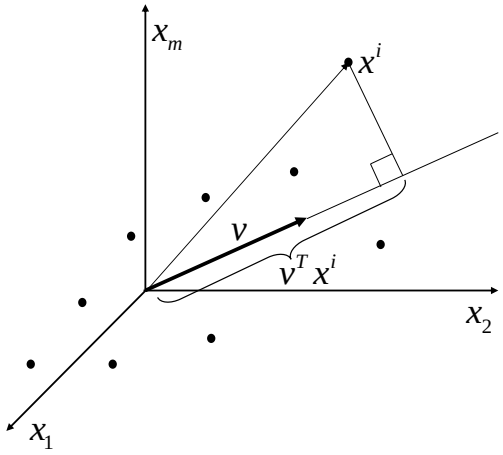

Figure 2 Searching Principal Components

$$\frac{\partial L}{\partial v} = 0, \quad 2XX^T v - 2\lambda v = 0 \quad \text{and} \quad \frac{\partial L}{\partial \lambda} = 0, \quad v^T v - 1 = 0 \qquad (6)$$

From the first equation it can be found that the vector *v* is an eigenvector of the matrix XX^T :

[1] It is a common practice in statistical multivariate analysis that each data vector is represented as a row vector, and the input data matrix is usually the transpose of the above matrix defined in (2). However, in the 2.160 lecture notes we stay with the standard vector matrix notation.

$$XX^T v = \lambda v$$

Since matrix XX^T is a real symmetric matrix and is a positive semi-definite matrix, it possesses all real, non-negative eigenvalues and real eigenvectors. Without loss of generality, we can arrange the eigenvalues in a descending order with the first eigenvalue to be the largest,

$$\begin{array}{l} \lambda_{\max} = \lambda_1 \geq \lambda_2 \geq \ \cdots \geq \lambda_{m^*} \geq \cdots \geq \lambda_m = 0 \\ \quad \vdots \quad \vdots \qquad\qquad \vdots \\ \quad v_1 \perp v_2 \perp \cdots \perp v_{m^*} \end{array} \tag{7}$$

where $v_1, v_2 \cdots$ are eigenvectors associated with eigenvalues $\lambda_1, \lambda_2, \cdots$, and they are orthogonal to each other. The squared input signal strength *J* takes the largest value in the direction of the eigenvector v_1 associated with the largest eigenvalue λ_1 .

Using the set of the eigenvectors as coordinate axes, the matrix XX^T can be diagonalized:

$$\begin{aligned} XX^T &= \begin{pmatrix} v_1 & v_2 \cdots v_m \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \lambda_m \end{pmatrix} \begin{pmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_m^T \end{pmatrix} \\ &= \lambda_1 v_1 v_1^T + \lambda_2 v_2 v_2^T + \cdots + \lambda_{m^*} v_{m^*} v_{m^*}^T + \cdots + \lambda_m v_m v_m^T \end{aligned} \tag{8}$$

Note that, since the matrix XX^T is singular, some of the eigenvalues are zero. (At least (*m*-*N*) eigenvalues are zero if $m > N$). In fact all the sample data are involved in a subspace spanned by a smaller number of eigenvectors. In the last expression of (8), the first term $\lambda_1 v_1 v_1^T$ is most significant, followed by the second term $\lambda_2 v_2 v_2^T$. The contribution of each term $\lambda_i v_i v_i^T$ diminishes as *i* increases. Some later terms with small eigenvalues can be ignored. This allows us to truncate the terms in (8) at an appropriate number, *m**.

$$XX^T \cong \lambda_1 v_1 v_1^T + \lambda_2 v_2 v_2^T + \cdots + \lambda_{m^*} v_{m^*} v_{m^*}^T, \quad m^* < m \tag{9}$$

The percentage accuracy of approximation can be evaluated by

$$\mu = \frac{\lambda_1 + \lambda_2 + \cdots + \lambda_{m^*}}{\lambda_1 + \lambda_2 + \cdots + \lambda_m} \times 100\% \tag{10}$$

Step 2 *Construction of a low dimensional regression*

Now that a fewer number of eigenvectors can approximate the *N* samples, let us represent a new input vector x_{new} with the fewer variables. Taking projection of the new input vector onto each of the significant eigenvectors, we can define *m** new variables:

$$\begin{aligned} z_1 &= x_{new}^T v_1 \\ z_2 &= x_{new}^T v_2 \\ &\vdots \\ z_{m^*} &= x_{new}^T v_{m^*} \end{aligned} \tag{11}$$

These new variables are called *Principal Components*, which are encapsulated in the original data, but are significant. Principal components are a type of *Latent Variables*, which will be discussed in detail in the following sections. In many applications, the first 3 ~ 5 principal components can approximate the majority of the original input vectors, which may be of high dimension. By using the principal components, we can formulate a reduced order regression for predicting output *y*:

$$\hat{y} = b_1' z_1 + \cdots + b_{m^*}' z_{m^*} , \tag{12}$$

where the number of parameters to tune, m^*, is much smaller than *m*, and they can be determined from the *N* samples by solving the standard Least Squares problem. To this end we convert all the sample points to principal components:

$$Z = X^T V ,$$

where $V = \left[v_1, v_2, \cdots, v_{m^*} \right]$ is an *m* by *m** vector consisting of the first *m** eigenvectors, and is called Loading, and *Z* is an *N* by *m** input data converted from the original samples. The least squares estimate of the reduced parameters is given by

$$\hat{\theta}' = \begin{pmatrix} b_1' \\ \vdots \\ b_{m^*}' \end{pmatrix} = (Z^T Z)^{-1} Z^T Y \tag{13}$$

where $Y = \begin{pmatrix} y^1 & \cdots & y^N \end{pmatrix}^T \in \Re^{N \times 1}$. Note that the matrix $\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{Z}$ is a nonsingular matrix. This is called Principal Components Regression (PCR).

4.2 Single-Output, Partial Least Squares Regression

The above Principal Components Regression can compress the input space effectively. Output *y* was regressed on the low dimensional principal components, where the regressor consists of a truncated series of principal components having large eigenvalues. Then, the question is whether the principal components selected based on eigenvalues are the most useful set of variables for predicting output *y*. There may be other set of variables that would be more effective to predict the specific set of output data. In the PCR we have quietly assumed that those significant principal components play more important roles in predicting the output than those with smaller eigenvalues. This may be a questionable assumption since it is conceivable that even less significant PCs may be more strongly correlated with the output than those PCs with larger eigenvalues. The principal component analysis ***explains*** the input data, but it does not explain the output and the relationship between the input and output. This section on Partial Least

Squares regression addresses exactly this issue and provides a solution. The basic idea is to find a low-dimensional set of input space variables that is most ***correlated*** with a given set of output data.

4.2.1 Algorithm

We begin with a simple algorithm for predicting a single output. The algorithm consists of three steps.

Step 1 *Finding the most correlated variable*

Similar to the Principal Components analysis, consider a unit vector ***v*** in the *m*-dimensional input space, and take projection onto the unit vector:

$$z = x^T v, \qquad (14)$$

where *z* is a scalar variable, called a *Score* variable. This time the unit vector ***v*** is determined in such a way that the associated score variable *z* may be most correlated with the output *y*. To this end we first project all the input data points onto the unit vector ***v***,

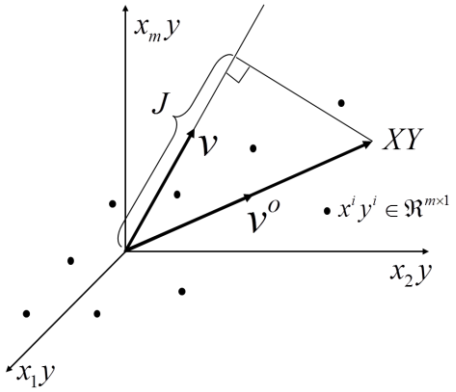

Figure 3 Input – output correlation

$$z^i = (x^i)^T v; \quad i = 1, \cdots, N \qquad (15)$$

and place them in a *N*-dimensional vector,

$$Z_v = \begin{pmatrix} z^1 \\ \vdots \\ z^N \end{pmatrix} = X^T v \in \Re^{N\times 1} \qquad (16)$$

The correlation between the score variable *z* and the output *y* can be evaluated by

$$J = \sum_{i=1}^{N} z^i y^i = Z_v{}^T Y = (v^T X)Y = v^T (X\,Y) \qquad (17)$$

The problem is now to find the direction of the unit vector *v* that maximizes the correlation *J*. Note that in (17) the product $X\,Y$ is an *m*-dimensional column vector. It is clear from Figure 3 that the correlation becomes maximal when the unit vector *v* is aligned with the vector $X\,Y$.

$$v^o = \arg\max_v J(v) = \frac{X\,Y}{|X\,Y|} \qquad (18)$$

where the vector $X\,Y$ is scaled by its absolute value $|X\,Y|$.

Step 2 *Predicting output y from score variable z*

Our objective is to predict output *y*. Now that we have found that the score variable *z* associated with the unit vector v^o has the highest correlation with output *y*, the following linear prediction based on *z* is the best prediction using a single variable.

$$\hat{y} = c^o z \tag{19}$$

where c^o is the scalar coefficient minimizing the mean squared error:

$$c^o = \arg\min_c \sum_{i=1}^{N} (y^i - cz^i)^2 = \frac{Y^T Z_v}{|Z_v|^2} \tag{20}$$

Figure 4 illustrates the correlation between the sampled score variables and the sampled outputs, and shows how the two are linked with the slope, i.e. the parameter *c*.

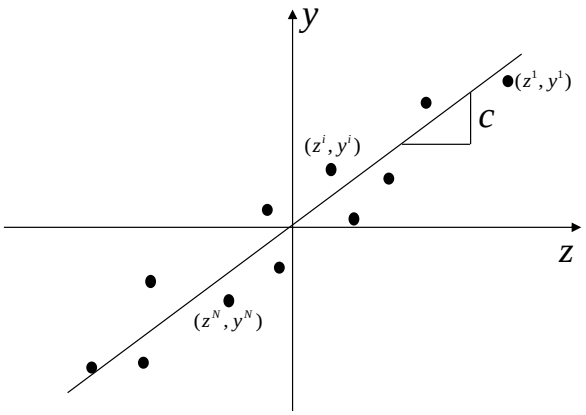

Figure 4 Plot of score variable and output: the slope *c* is determined by least squares

Step 3 *Repeating the process for the residue*

The prediction of output *y* based on the single score variable *z* may be limited in accuracy. The data *X* and *y* may contain more correlation in other directions, which we should explore. We can expect that the prediction error may be reduced by using two score variables, say *z* and *z*'. The second score variable z' is responsible for predicting the output in relation to the *residue* of the input data that the first score variable could not exploit,

$$y' = y - c^o z \tag{21}$$

For *N* samples, we use the following *N*-dimensional output residue vector:

$$Y' = Y - c^o Z_v \tag{22}$$

As for the input data, we have already used $Z_v = X^T v$ for creating the prediction $c^o Z_v$. Therefore, we have to subtract this used information from the whole input data matrix *X* in order to find the second score variable. Consider the *m* x *N* matrix *X* as a collection of row vectors $\xi_j = \left(x_j^1, \cdots, x_j^N\right) \in \Re^{1\times N}$, $j = 1, \cdots, m$, in the *N*-dimensional vector space. The score vector transpose $Z_v^T \in \Re^{1\times N}$ in the *N*-dimensional vector space indicates the direction in which the information encapsulated in *X* has been used. The residue that has not yet been extracted is the subspace that is orthogonal to the score vector Z_v^T. Projection of each row vector ξ_j, $j = 1, \cdots, m$ onto the unit vector $Z_v^T / |Z_v|$ provides the magnitude of each row vector : $a_j = \xi_j Z_v / |Z_v|$ in this direction. Subtracting $a_j Z_v^T / |Z_v|$ from the original row vector ξ_j yields

$$\xi_j - a_j \frac{Z_v^T}{|Z_v|} = \xi_j \left(I - \frac{Z_v Z_v^T}{|Z_v|^2} \right) \tag{23}$$

See Figure 5. Concatenating the above row vectors vertically, we can obtain the residue of input data matrix X:

$$X' = X - X\frac{Z_v Z_v^T}{|Z_v|^2} = X\left(I - \frac{Z_v Z_v^T}{|Z_v|^2}\right) \tag{24}$$

where the N x N matrix $\left(I - \frac{Z_v Z_v^T}{|Z_v|^2}\right)$ is a projection matrix that projects the whole input data matrix X onto the subspace orthogonal to the unit vector $Z_v^T / |Z_v|$. The residue of input data matrix, X', does not contain any information that has already been used for the first score variable. The used information, on the other hand, is contained in the matrix:

$$\begin{pmatrix} a_1 \\ \vdots \\ a_m \end{pmatrix}\frac{Z_v^T}{|Z_v|} = \begin{pmatrix} \xi_1 \\ \vdots \\ \xi_m \end{pmatrix}\frac{Z_v Z_v^T}{|Z_v|^2} = pZ_v^T, \tag{25}$$

where $p = \frac{XZ_v}{|Z_v|^2}$ is called the *Loading* vector associated with the first score variable.

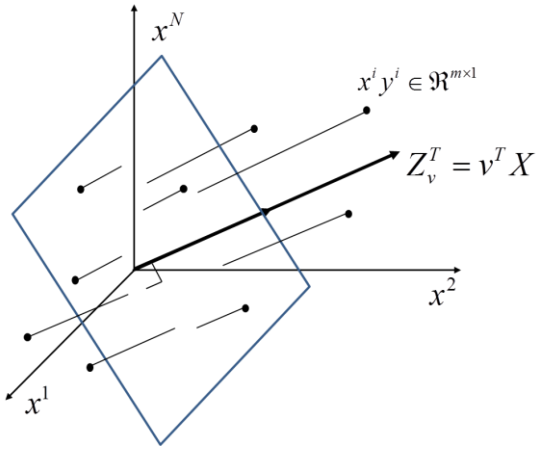

Figure 5 Deflation of input data matrix via projection onto the subspace orthogonal to the score vector

Now that the components of the input and output data associated with the first score variable have been filtered out in the new input and output data, X', Y', the second score variable can be determined by repeating Step 1 where the input data matrix X is replaced by X' and the output Y is by Y'. Let z' be the second score variable that maximizes the correlation between X' and Y', and v' be the unit vector generating the second score variable. Again by repeating Step 2, the output prediction is given by $\hat{y}' = c'v'$.

Each iteration of the above procedure extracts a set of variables, including unit vector v, score z, and loading vector p, from the residue. These variables, encapsulated in the original input data, are found to be significant in correlating the input and to the output. This set of variables is referred to as ***Latent Variables*** for a single output. Repeating the above steps m^* times ($m^* \le \min[m, N]$) until the correlation between the residues X', Y' diminishes (the residue becomes totally random and no useful correlation can be found), we can obtain a series of latent variables: unit vectors $v(1), \cdots, v(m^*)$, score variables $Z_{v1}(1), \cdots, Z_{vm^*}(m^*)$, and loading vectors $p(1), \cdots, p(m^*)$ as well as the least squares coefficients $c(1), \cdots, c(m^*)$. Using all these latent variables we can predict output y as

$$\hat{y} = c(1)z(1) + \cdots + c(m^*)z(m^*) = B \cdot x \tag{26}$$

where

$$B = \sum_{i=1}^{m^*} c(i)v^T(i) \ . \tag{27}$$

4.2.2 Properties

Latent Variables constructed with the above procedure possess some good properties.

Orthogonality of unit vectors: It is important to check whether the first and the second unit vectors are orthogonal to each other. First, the unit vector of the second latent variable is in the direction of:

$$v' \propto X'Y' = X\left(I - \frac{Z_v Z_v^T}{|Z_v|^2}\right)(Y - c^o Z_v)$$
$$= X\left(I - \frac{Z_v Z_v^T}{|Z_v|^2}\right)Y - X\left(c^o Z_v - c^o \frac{Z_v Z_v^T Z_v}{|Z_v|^2}\right) = X\left(I - \frac{Z_v Z_v^T}{|Z_v|^2}\right)Y \tag{28}$$

Now computing the inner product,

$$v^T v' = v^T X\left(I - \frac{Z_v Z_v^T}{|Z_v|^2}\right)Y = \left(Z_v^T - \frac{Z_v^T Z_v Z_v^T}{|Z_v|^2}\right)Y = 0 \tag{29}$$

Z_v^T

Therefore, the two unit vectors are orthogonal to each other.

Orthogonality of Score Vectors: Similarly, we can show that the two score vectors, Z_v and $Z'_{v'} = \left[z'^1, \cdots, z'^N\right]^T$, too, are orthogonal to each other.

$$Z_v^T Z'_{v'} = Z_v^T \left[X\left(I - \frac{Z_v Z_v^T}{|Z_v|^2}\right)\right]^T v' = \left(Z_v^T - \frac{Z_v^T Z_v Z_v^T}{|Z_v|^2}\right)X^T v' = \left(Z_v^T - Z_v^T\right)X^T v' = 0 \tag{30}$$