- In a pulp digester, relate the liquor concentrations and temperature to the wood composition (e.g., Kappa Number) of the product.

1.5.2 THE METHOD OF LEAST SQUARES

What Is It?

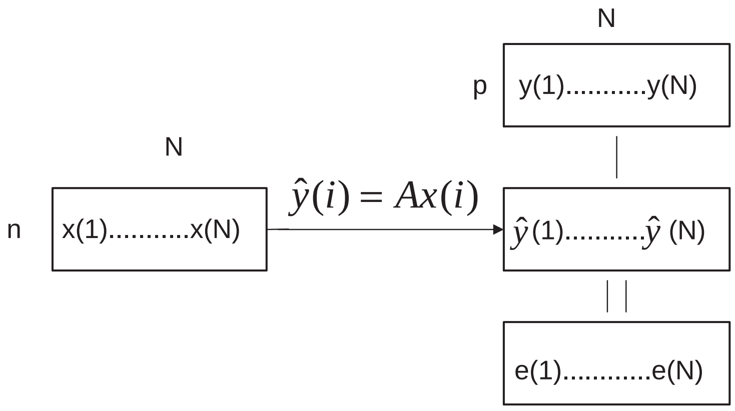
The most widely used is the method of least squares. The least squares method is particularly powerful when one wishes to build a *linear* prediction model of the form

$$\hat{y} = Ax \tag{1.47}$$

With N data points, we can write

$$\underbrace{\begin{bmatrix} y(1) & \cdots & y(N) \end{bmatrix}}_{Y} = A \underbrace{\begin{bmatrix} x(1) & \cdots & x(N) \end{bmatrix}}_{X} + \underbrace{\begin{bmatrix} e(1) & \cdots & e(N) \end{bmatrix}}_{E} \tag{1.48}$$

The last term represents the prediction error (for the prediction model $\hat{y} = Ax$) for the N available data points.



A reasonable criterion for using A is

$$\min_{A}\{\sum_{i=1}^{N} e^T(i)e(i) = \|Y - AX\|_f^2\} \tag{1.49}$$

The solution to the above is

$$A = YX^T(XX^T)^{-1} \tag{1.50}$$

Statistical Interpretation

One can develop the least squares solution from the following statistical argument. Suppose the underlying system (from which the N data set was generated) is

$$y = Ax + \varepsilon \tag{1.51}$$

where ε is a zero-mean, Gaussian random variable vector (covering for the noise and other randomness in the relationship between x and y). Assume also that x is a Gaussian vector. Then, y is also Gaussian due to the linearity. Then,

$$E\{y|x\} = \bar{y} + \text{cov}\{y,x\}\text{cov}^{-1}\{x,x\}(x-\bar{x}) \tag{1.52}$$

Since x and y are both mean-centered variables, $\bar{x} = 0$ and $\bar{y} = 0$. We now approximate the covariances using N data points available to us.

$$\text{cov}(y,x) \approx R_{yx} = \frac{1}{N}\sum_{i=1}^{N} y(i)x^T(i) \tag{1.53}$$

$$\text{cov}(y,x) \approx R_{x} = \frac{1}{N}\sum_{i=1}^{N} x(i)x^T(i) \tag{1.54}$$

Hence,

$$\begin{aligned} E\{y|x\} \approx \hat{y} &= \underbrace{\left(\frac{1}{N}\sum_{i=1}^{N} y(i)x^T(i)\right)\left(\frac{1}{N}\sum_{i=1}^{N} x(i)x^T(i)\right)^{-1}}_{A} x \\ &= \tfrac{1}{N}YX^T(\tfrac{1}{N}XX^T)^{-1}x \end{aligned} \tag{1.55}$$

Note that the above is the same as the predictor that results from the method of least squares.

1.5.3 LIMITATIONS OF LEAST SQUARES

Possibility of Ill-Conditioning

Recall the least squares solution

$$\begin{aligned} \hat{y} &= YX^T(XX^T)^{-1}x \\ &= R_{yx}R_x^{-1}x \end{aligned} \tag{1.56}$$

Since R_x is a symmetric, positive (semi)-definite matrix, it has the decomposition in the form of

$$R_x^{-1} = \begin{bmatrix} v_1 & \cdots & v_n \end{bmatrix} \begin{bmatrix} \frac{1}{\sigma_1^2} & & \\ & \ddots & \\ & & \frac{1}{\sigma_n^2} \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_n^T \end{bmatrix} \tag{1.57}$$

In the case that the x data are highly correlated, $\sigma_1 \gg \sigma_n$ and some of σ's will be very small (in a relative sense).

Implication of Ill-Conditioned Information Matrix

This has the following implications.

- *Possibility of Artificially High Gains Due to Poor Signal to Noise Ratio*
  Note that R_{yx} and R_x are only approximations of the covariance matrices based on N data points. Due to the error in the data, they both contain errors. When $\frac{1}{\sigma_i^2}$'s are large, errors in R_{yx} can get amplified greatly leading to a bad predictor (*e.g.*, a predictor with artificially high gains).

- *Sensitivity to Outliers and Noise*
  Also, even if the covariance matrices were estimated perfectly, the prediction can still be vulnerable to errors in the x data due to the high gain.

- *Statistical Viewpoint*

R_x (actually XX^T) is called information matrix. σ_i represents the amount of information contained in the data X for a particular linear combination of x (given by $v_i^T x$). Hence, small σ_i means small amount of information. Naturally, extracting the correlation between $v_i^T x$ and y from the very small amount of data can lead to trouble.

Examples

CONSIDER A TWO-DIMENSIONAL CASE WITH AN ILL-CONDITIONED INFORMATION MATRIX. GRAPHICALLY ILLUSTRATE THE DATA DISTRIBUTION AND HOW IT RELATES TO THE SVD, RESULTING ESTIMATE, etc.

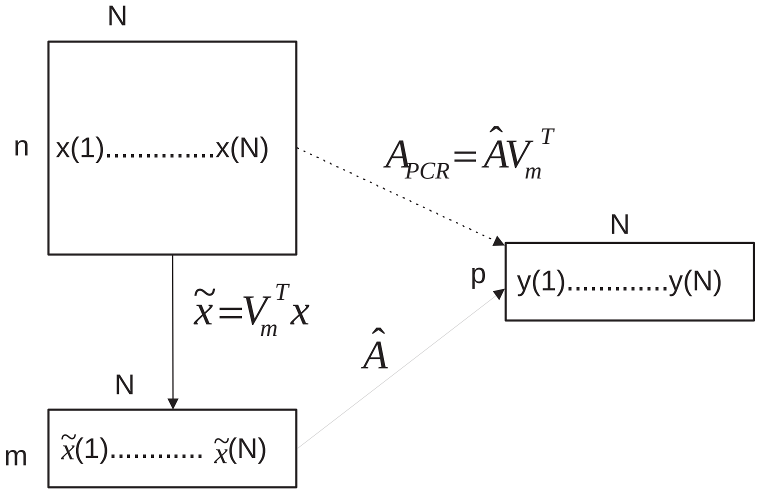
1.5.4 PRINCIPAL COMPONENT REGRESSION

Main Idea

Partition the decomposition of the matrix R_x as

$$\left[\begin{array}{ccc|ccc} v_1 & \cdots & v_m & v_{m+1} & \cdots & v_n \end{array}\right] \left[\begin{array}{ccc|ccc} \sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_m & & & \\ \hline & & & \sigma_{m+1} & & \\ & & & & \ddots & \\ & & & & & \sigma_n \end{array}\right] \left[\begin{array}{c} v_1^T \\ \vdots \\ v_m^T \\ \hline v_{m+1}^T \\ \vdots \\ v_n^T \end{array}\right] \tag{1.58}$$

The main idea is to project the data down to the reduced dimensional space defined by $v_1, \cdots, v_m$ (which also represents the space for which a large amount of data are available). This is illustrated graphically as follows:



We can write the projection as

$$\tilde{x} \triangleq V_m^T x = \begin{bmatrix} v_1^T \\ \vdots \\ v_m^T \end{bmatrix} x \tag{1.59}$$

$\tilde{x}$ represents the principal components of x. Note that, in the case that x is of very high dimension, it is likely that $\dim\{\tilde{x}\} \ll \dim\{x\}$. We can write the least squares estimator as

$$\begin{aligned} \hat{y} &= \underbrace{Y\tilde{X}^T(\tilde{X}\tilde{X}^T)^{-1}}_{\tilde{A}}\tilde{x} \\ &= \underbrace{\tilde{A}V_m^T}_{A_{PCR}} x \end{aligned} \tag{1.60}$$

This is called *principal component regression.*

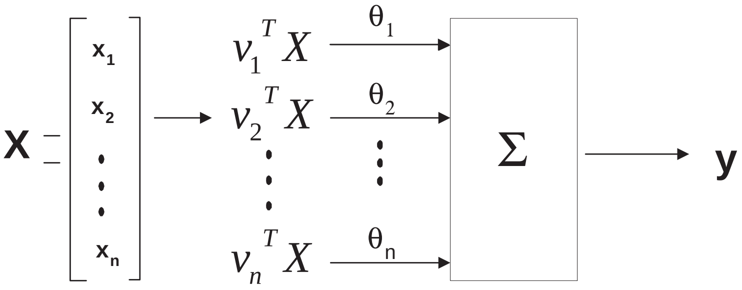
Statistical Viewpoint

In a statistical sense, it can be interpreted as accepting *bias* for reduced *variance.* We are *a priori* setting the correlation between

$v_i^T x, i = m+1, \cdots, n$ and y to be zero, i.e.,

$$y = \theta_1 \underbrace{v_1^T x}_{\tilde{x}_1} + \cdots + \theta_m \underbrace{v_m^T x}_{\tilde{x}_m} + 0 \times \underbrace{v_{m+1}^T x}_{\tilde{x}_{m+1}} + \cdots + 0 \times \underbrace{v_n^T x}_{\tilde{x}_n}$$

since computing the correlation based on data can introduce substantial variances which are thought to be much more harmful to estimation than the bias.



Example

TAKE THE PREVIOUS EXAMPLE, DO THE PRINCIPAL COMPONENT REGRESSION AND SHOW THE VARIANCE VS. BIAS TRADE-OFF.

1.5.5 PARTIAL LEAST SQUARES (PLS)

Main Idea

PLS is similar to PCR in that they are both biased regressions or subspace regression. The difference is that, in PLS, the subspace (consisting of m directions) of the regressor space is chosen to maximize XY^TYX^T rather than XX^T. In other words, in choosing the m directions, one looks at

- not only how much a certain modes contributes to the X data,
- but also how much it is correlated with the Y data.

In this sense, it can be thought as a middle ground between the PCR and the regular least squares.

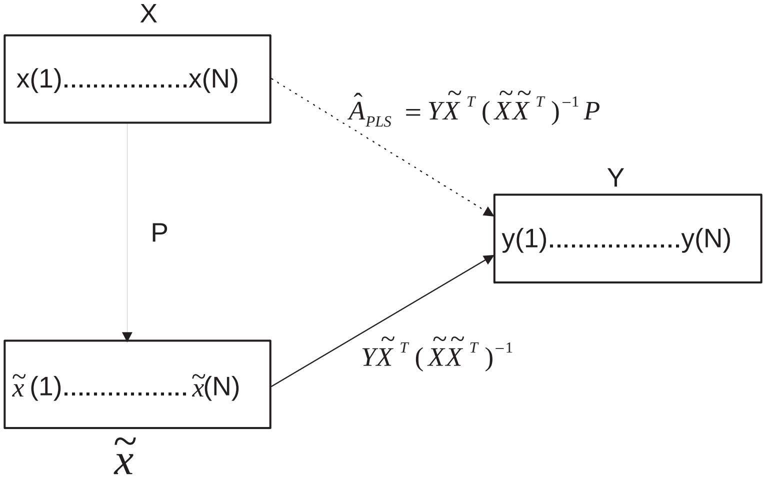
Procedure

The PLS procedure can be explained as follows:

1. Set $i = 1$. $X_1 = X$ and $Y_1 = Y$.
2. Choose the principal direction v_i for $X_i^T Y_i Y_i^T X_i$.
3. $v_i^T X_i = \tilde{X}_i$
4. Compute the LS prediction of Y_i based on $\tilde{X}_i$.
$$\hat{Y}_i = Y_i \tilde{X}_i^T \frac{1}{(\tilde{X}_i \tilde{X}_i^T)} \tilde{X}_i \tag{1.61}$$
5. Compute the residuals
$$\begin{aligned} Y_{i+1} &= Y_i - \hat{Y}_i \\ X_{i+1} &= X_i - v_i \tilde{x}_i \end{aligned} \tag{1.62}$$
6. If the residual Y_{i+1} is sufficiently small, stop. If not, set $i = i + 1$ and go back to Step 2.

From the above, with m iteration, the m-dimensional regression space is defined. Create a data matrix

$$\tilde{X} = \begin{bmatrix} \tilde{X}_1 \\ \vdots \\ \tilde{X}_m \end{bmatrix}$$


X
x(1).................x(N)
$\hat{A}_{PLS} = Y\tilde{X}^T(\tilde{X}\tilde{X}^T)^{-1}P$
Y
y(1).................y(N)
P
$Y\tilde{X}^T(\tilde{X}\tilde{X}^T)^{-1}$
$\tilde{x}$(1).................$\tilde{x}$(N)
$\tilde{X}$

$\tilde{X}$ can be expressed as a linear projection of X:

$$\tilde{X} = PX \tag{1.63}$$

where $P \in \mathcal{R}^{m \times n}$. Then, the PLS predictor can be written as

$$\begin{aligned} \hat{y} &= Y\tilde{X}^T(\tilde{X}\tilde{X}^T)^{-1}\tilde{x} \\ &= \underbrace{Y\tilde{X}^T(\tilde{X}\tilde{X}^T)^{-1}P}_{A_{PLS}}\, x \end{aligned} \tag{1.64}$$

The above is not the most efficient algorithm from a computational standpoint. The most widely used is the NIPALS (Nonlinear Iterative Partial Least Squares) algorithm described in (Geladi and Kowalski, Analytica Chimica Acta, 1986)

1.5.6 NONLINEAR EXTENSIONS

Regression needs not be confined to just linear relationships. More generally, one can search for a prediction model of the form $\hat{y} = f(x)$ where f can be a nonlinear function.

Finite Dimensional Parameterization

To reduce the problem to a parameter estimation, one defines a search set with a finite dimensional parameterization for the nonlinear function. The search set can take on different forms.

- **Functional Expansion**

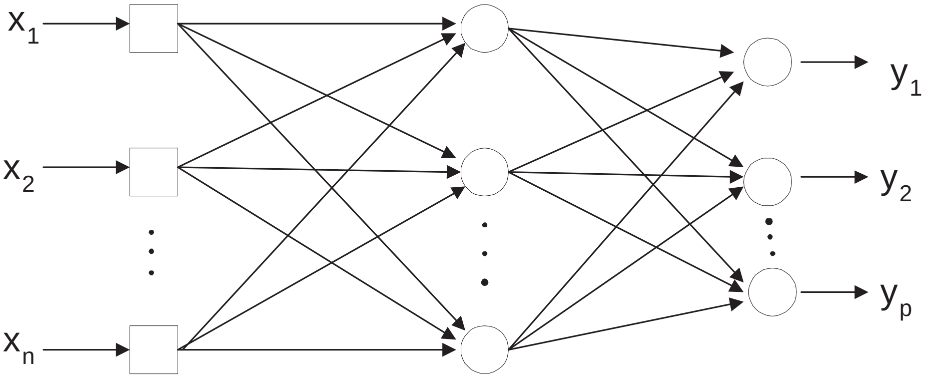
  $$\hat{y} = \sum_{i=1}^{n} c_i \phi_i(x) \tag{1.65}$$

  or

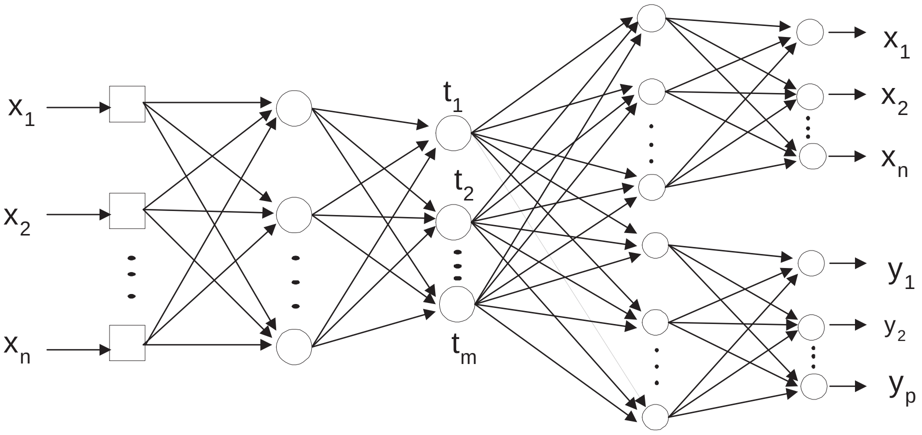
  $$\hat{y} = \sum_{i=1}^{n} \phi_i(x, c_i) \tag{1.66}$$

  $\{\phi_i(x), i = 1, \cdots, n\}$ are basis functions (polynomials, sinusoids, wavelets, Gaussian functions, etc.). The problem is reduced to finding c_i. The former leads to a linear regression problem while the latter to a nonlinear problem. The order can be determined on an iterative basis, that is, by examining the prediction error as more and more terms are introduced.

- **Network Based Approach**
  For instance, shown below is the so called Artificial Neural Network (ANN) inspired by biological neural systems. The parameters are the various weights which must be selected on the basis of the available data. This is referred to as the *learning* in the ANN parlance. The usual criterion is again the least squares or its extensions.



Nonlinear PLS



1.5.7 EXTENSIONS TO THE DYNAMIC CASE

Suppose x and y have dynamic correlations:

$$y(k) = f(x(k), x(k-1), \cdots, \cdots) \tag{1.67}$$

Different structures can be envisioned:

- *Time Series:* construct a predictor of form

$$\begin{aligned} \hat{y}(k) &= a_1\hat{y}(k-1) + \cdots + a_n\hat{y}(k-n) \\ &\quad +b_0x(k) + b_1x(k-1) + \cdots + b_mx(k-n) \\ y(k) &= \hat{y}(k) + \varepsilon(k) \end{aligned} \tag{1.68}$$

  $a_1, \cdots, a_n, b_0, \cdots, b_m$ can be found to minimize the prediction error using the available data. Note that, since we don't have data for $\hat{y}(k-1), \cdots, \hat{y}(k-n)$, and they depend on the choice of the parameters, this is a nonlinear regression problem. Therefore, it is pretty much limited to SISO problems.

- *State-Space Model:* For MIMO systems, use Subspace ID to create

$$\begin{aligned} z(k+1) &= Az(k) + \varepsilon_1(k) \\ \begin{bmatrix} x(k) \\ y(k) \end{bmatrix} &= \begin{bmatrix} C_x \\ C_y \end{bmatrix} z(k) + \varepsilon_2(k) \end{aligned} \tag{1.69}$$

  Then, build the Kalman filter that uses the measurement x to predict y can be written as

$$\begin{aligned} \hat{z}(k) &= A\hat{z}(k-1) + K(x(k) - C_xA\hat{z}(k-1)) \\ \hat{y}(k) &= C_y\hat{z}(k) \end{aligned} \tag{1.70}$$