Least Square Regression

From maximum likelihood to least square

Maximum likelihood is a method for estimating parameter set θ of a statistical model when given a set of N noisy observations (or data points) { d_1, d_2, d_3, …, d_N } of a random variable (or random vector) D. Likelihood of θ given observations of D is defined as the joint density function of D given θ.

$$L(\theta \mid D = d_1, d_2, d_3, ..., d_N) = p(D = d_1, d_2, d_3, ..., d_N \mid \theta)$$

$$= \prod_{n=1}^{N} p(D = d_n \mid \theta) \quad \text{when observations are independent and identical}$$

$$\ln L(\theta \mid D = d_1, d_2, d_3, ..., d_N) = \sum_{n=1}^{N} \ln p(D = d_n \mid \theta) \quad \text{when log likelihood is easier to work with}$$

Maximum likelihood means estimation of θ by maximizing the likelihood with respect to θ. When maximum likelihood is used in regression, there should be two components in the model : (1) regression model and (2) noise model. Lets consider linear regression AX=B, now we denote observations as a N×M row data matrix A and a N×1 column matrix B, i.e. observations of D = {A, B}, and denote regression parameter X as a M×1 matrix. Besides, there exists **additive** zero mean (correlated or uncorrelated, heteroskedasticity or homoskedasticity) Gaussian noise, thus the model becomes AX+ε=B, where ε is a N×1 matrix, having covariance matrix Σ, i.e. parameter set θ = {X, Σ}.

$$p(A, B \mid X, \Sigma) = \frac{1}{\sqrt{(2\pi)^{rank(\Sigma)} \det(\Sigma)}} \exp(-\frac{1}{2}(AX - B)^T \Sigma^{-1} (AX - B)) \quad \text{for correlated Gaussian}$$

$$= \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp(-\frac{(A_n X - b_n)^2}{2\sigma_n^2}) \quad \text{for uncorrelated Gaussian, heteroskedasticity}$$

$$= \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(A_n X - b_n)^2}{2\sigma^2}) \quad \text{for uncorrelated Gaussian, homoskedasticity}$$

where A_n = n^{th} row vector of matrix A, which is 1×M

b_n = n^{th} element of matrix B, which is a scalar

Homoskedasticity means a constant variance (for different observations), heteroskedasticity means a varying variance (for different observations). By taking log on the likelihood, we will find that maximum likelihood is actually equivalent to different types of least square for the three different cases. Starting from this point, Σ is assumed to be given in this document.

case 1 – uncorrelated Gaussian, homoskedasticity

$$\arg\max_X \ln L(X \mid A, B, \Sigma) = \arg\max_X \ln \left[\prod_{n=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(A_n X - b_n)^2}{2\sigma^2}) \right]$$

$$= \arg\max_X \sum_{n=1}^{N} \left[\ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{(A_n X - b_n)^2}{2\sigma^2} \right]$$

$$= \arg\min_X \sum_{n=1}^{N} (A_n X - b_n)^2$$

$$= \arg\min_X (AX - B)^T (AX - B) \quad \text{known as ordinary least square}$$

case 2 – uncorrelated Gaussian, heteroskedasticity

$$\arg\max_X \ln L(X \mid A, B, \Sigma) = \arg\max_X \ln \left[\prod_{n=1}^{N} \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp(-\frac{(A_n X - b_n)^2}{2\sigma_n^2}) \right]$$

$$= \arg\max_X \sum_{n=1}^{N} \left[\ln \frac{1}{\sqrt{2\pi\sigma_n^2}} - \frac{(A_n X - b_n)^2}{2\sigma_n^2} \right]$$

$$= \arg\min_X \sum_{n=1}^{N} (\frac{A_n X - b_n}{\sigma_n})^2$$

$$= \arg\min_X \sum_{n=1}^{N} (AX - B)^T W (AX - B) \quad \text{where W is diagonal with } w_n = 1/\sigma_n^2$$

known as weighted least square

case 3 – correlated Gaussian

$$\begin{aligned}
\arg\max_X \ln L(X \mid A,B,\Sigma) &= \arg\max_X \ln\left[\frac{1}{\sqrt{(2\pi)^{rank(\Sigma)}\det(\Sigma)}}\exp(-\frac{1}{2}(AX-B)^T\Sigma^{-1}(AX-B))\right] \\
&= \arg\max_X \left[\ln\frac{1}{\sqrt{(2\pi)^{rank(\Sigma)}\det(\Sigma)}}-\frac{1}{2}(AX-B)^T\Sigma^{-1}(AX-B)\right] \\
&= \arg\min_X (AX-B)^T\Sigma^{-1}(AX-B) \qquad \text{known as generalized least square}
\end{aligned}$$

Tikhonov regularization can be introduced to least square method for solving ill posed problem. Least square can also be generalized to non linear regression (i.e. f(A,X)=B instead of AX=B) using Gauss Newton method, or generalized to non Gaussian noise (i.e. noise with other distributions, or even in the existence of outlier) using iterative reweighted least square, or generalized to dynamic system (i.e. time dependent model parameters) using recursive least square.

Least square solution

Here is a summary of different least square methods.

- ordinary least square — linear model, uncorrelated homoskedastic Gaussian
- weighted least square — linear model, uncorrelated heteroskedastic Gaussian
- generalized least square — linear model, correlated Gaussian
- generalized Tikhonov regularization (ridge regression) — generalization to ill posed problem
- non linear least square (Gauss Newton algorithm) — generalization to non linear model
- non linear least square (Levenberg Marquardt algorithm) — generalization to non linear model
- iterative reweighted least square — generalization to non Gaussian noise (LS2.doc)
- recursive least square — generalization to time varying system (LS2.doc)
- two stage least square — generalization to endogenous regressor (LS2.doc)

Ordinary least square

Taking derivative of objective function L wrt X, set it to zero and take transpose.

$$\begin{aligned}
L &= (AX-B)^T(AX-B) \\
0 &= 2A^T(AX_{LS}-B) \\
0 &= (A^TA)X_{LS}-(A^TB) \\
X_{LS} &= (A^TA)^{-1}(A^TB) = FB
\end{aligned} \qquad \text{where linear transformation } F=(A^TA)^{-1}A^T$$

Weighted least square

Taking derivative of objective function L wrt X, set it to zero and take transpose.

$$\begin{aligned}
L &= (AX-B)^TW(AX-B) \\
0 &= 2A^TW(AX_{WLS}-B) \\
0 &= (A^TWA)X_{WLS}-(A^TWB) \\
X_{WLS} &= (A^TWA)^{-1}(A^TWB) = FB
\end{aligned} \qquad \text{where linear transformation } F=(A^TWA)^{-1}(A^TW)$$

Generalized least square

Taking derivative of objective function L wrt X, set it to zero and take transpose.

$$\begin{aligned}
L &= (AX-B)^T\Sigma^{-1}(AX-B) \\
0 &= 2A^T\Sigma^{-1}(AX_{GLS}-B) \\
0 &= (A^T\Sigma^{-1}A)X_{GLS}-(A^T\Sigma^{-1}B) \\
X_{GLS} &= (A^T\Sigma^{-1}A)^{-1}(A^T\Sigma^{-1}B) = FB
\end{aligned} \qquad \text{where linear transformation } F=(A^T\Sigma^{-1}A)^{-1}(A^T\Sigma^{-1})$$

All OLS, WLS and GLS are linear transformations of column matrix B, where F is a M×N matrix that transforms B in N dimensional space to X_{LS} in M dimensional space (i.e. $\Re^N \rightarrow \Re^M$). Hence all OLS, WLS and GLS are classified as linear estimators. OLS minmizes **Euclidean distance** of error vector, while GLS minimizes **Mahalanobis distance** of error vector. Intuitively, Mahalanobis distance is distance in terms of number of standard deviation (or distance normalized by covariance matrix). Hence we have an alternative notation for objective function L :

$$\begin{aligned}
L &= \|AX-B\|^2 &&= (AX-B)^T(AX-B) && \text{i.e. Euclidean distance in OLS} \\
L &= \|AX-B\|^2_{\Sigma^{-1}} &&= (AX-B)^T\Sigma^{-1}(AX-B) && \text{i.e. Mahalanobis distance in GLS (note the inverse)} \\
&&&= (AX-B)^TU^TU(AX-B) && \text{Cholesky decomposition of inverse covariance matrix} \\
&&&= (UAX-UB)^T(UAX-UB) && \text{equivalent to least square of transformed A and B}
\end{aligned}$$

Generalized Tikhonov regularization

Sometimes, A is nearly singular, resulting in unstable estimation of X, we can then stablize the estimation by adding a regularization term, constraining X softly to the expected value X_0. Please review Bayesian framework to see why we consider the distribution of X in Tikhonov regularization, but not in ordinary least square, weighted least square nor generalized least square.

$$L = \|AX-B\|_P^2 + \|X-X_0\|_Q^2 = (AX-B)^T P(AX-B) + (X-X_0)^T Q(X-X_0)$$

where $P = \mathrm{cov}(B)^{-1}$

$Q = \mathrm{cov}(X)^{-1}$

$= U^T U$ by Cholesky decomposition, where U is called Tikhonov matrix

$X_0 = E(X)$

Taking derivative of objective function L wrt X, set it to zero and take transpose.

$$\begin{aligned} L &= (AX-B)^T P(AX-B) + (X-X_0)^T Q(X-X_0) \\ 0 &= 2A^T P(AX_{TR}-B) + 2Q(X_{TR}-X_0) \\ 0 &= (A^T PA+Q)X_{TR} - (A^T PB + QX_0) \\ X_{TR} &= (A^T PA+Q)^{-1}(A^T PB + QX_0) \end{aligned}$$

Nonlinear least square - Implemented by Gauss Newton algorithm

When the regression model is nonlinear :

$$B = F(A,X) + \varepsilon$$

where $F(A,X) = \begin{bmatrix} f(A_1,X) \\ f(A_2,X) \\ f(A_3,X) \\ \dots \\ f(A_N,X) \end{bmatrix}$ and $A = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ \dots \\ A_N \end{bmatrix}$

by following log likelihood maximization argument, it becomes minimizing objective function L_1 :

$$\begin{aligned} L_1 &= \|f(A,X)-B\|_{\Sigma^{-1}}^2 \\ &= \left\|f(A,X^{(t)}+\Delta X)-B\right\|_{\Sigma^{-1}}^2 \\ &\sim \left\|f(A,X^{(t)})+J_F\Delta X-B\right\|_{\Sigma^{-1}}^2 \quad \text{by Taylor series expansion, taking the first derivative term} \\ &= \left\|J_F\Delta X-(B-f(A,X^{(t)}))\right\|_{\Sigma^{-1}}^2 \end{aligned}$$

Taking derivative of objective function L_1 wrt ΔX, set it to zero and take transpose.

$$\begin{aligned} L_1 &= (J_F\Delta X-(B-f(A,X^{(t)})))^T\Sigma^{-1}(J_F\Delta X-(B-f(A,X^{(t)}))) \\ 0 &= 2J_F^T\Sigma^{-1}(J_F\Delta X-(B-f(A,X^{(t)}))) \\ 0 &= (J_F^T\Sigma^{-1}J_F)\Delta X-(J_F^T\Sigma^{-1}(B-f(A,X^{(t)}))) \\ \Delta X &= (J_F^T\Sigma^{-1}J_F)^{-1}(J_F^T\Sigma^{-1}(B-f(A,X^{(t)}))) \\ &= (J_F^T\Sigma^{-1}J_F)^{-1}(J_F^T\Sigma^{-1}R^{(t)}) \\ X^{(t+1)} &= X^{(t)}+\Delta X \end{aligned}$$

where $R^{(t)} = B-f(A,X^{(t)})$

and $J_f = \begin{bmatrix} \partial f_1/\partial x_1 & \partial f_1/\partial x_2 & \partial f_1/\partial x_3 & \dots & \partial f_N/\partial x_1 \\ \partial f_2/\partial x_1 & \partial f_2/\partial x_2 & \partial f_2/\partial x_3 & \dots & \partial f_N/\partial x_2 \\ \partial f_3/\partial x_1 & \partial f_3/\partial x_2 & \partial f_3/\partial x_3 & \dots & \partial f_N/\partial x_3 \\ \dots & \dots & \dots & \dots & \dots \\ \partial f_N/\partial x_1 & \partial f_N/\partial x_2 & \partial f_N/\partial x_3 & \dots & \partial f_N/\partial x_M \end{bmatrix}$

This is an iterative method for solving nonlinear least square, known as Gauss Newton algorithm. An alternative is an integration of Gauss Newton together with gradient descent, known as Levenberg Marquardt algorithm.

Nonlinear least square - Implemented by Levenberg Marquardt algorithm

Objective function L_1 is modified by adding a $\lambda \Delta X^T \Delta X$ term to give objective function L_2, the extra term provides double meaning : (1) acts as a regularization term (this is obvious) or (2) acts as gradient descent (we will see that soon).

$$L_2 = \left\| J_F \Delta X - (B - f(A, X^{(t)})) \right\|^2_{\Sigma^{-1}} + \lambda \Delta X^T \Delta X$$

Taking derivative of objective function L_2 wrt ΔX, set it to zero and take transpose.

$$\begin{aligned}
L_2 &= (J_F \Delta X - (B - f(A, X^{(t)})))^T \Sigma^{-1} (J_F \Delta X - (B - f(A, X^{(t)}))) + \lambda \Delta X^T \Delta X \\
0 &= 2 J_F{}^T \Sigma^{-1} (J_F \Delta X - (B - f(A, X^{(t)}))) + 2\lambda I \Delta X \\
0 &= (J_F^T \Sigma^{-1} J_F + \lambda I) \Delta X - (J_F^T \Sigma^{-1} (B - f(A, X^{(t)}))) \\
\Delta X &= (J_F^T \Sigma^{-1} J_F + \lambda I)^{-1} (J_F^T \Sigma^{-1} (B - f(A, X^{(t)}))) \\
&= (J_F^T \Sigma^{-1} J_F + \lambda I)^{-1} (J_F^T \Sigma^{-1} R^{(t)}) \\
X^{(t+1)} &= X^{(t)} + \Delta X
\end{aligned}$$

where $R^{(t)}$ = same as that in Gauss Newton algorithm
and J_f = same as that in Gauss Newton algorithm

Why does Levenberg Marquardt algorithm contain gradient descent? Lets decompose ΔX as :

(1) $(J_F^T \Sigma^{-1} J_F) \Delta X_{Gauss} = J_F^T \Sigma^{-1} R^{(t)}$ i.e. Gauss Newton component

(2) $\lambda \Delta X_{gradient\_descent} = J_F^T \Sigma^{-1} R^{(t)}$ i.e. gradient descent component

The RHS of the above two equations are derivative of objective function L_1 (in Gauss Newton algorithm) with respect to X **(not ΔX!!!)**, thus equation 2 denotes gradient descent, as we restrict the change in X to negative gradient. Lets check, objective function L_1 in Gauss Newton is :

$$\begin{aligned}
L_1 &= (f(A, X) - B)^T \Sigma^{-1} (f(A, X) - B) \\
dL_1 / dX &= 2 J_F^T \Sigma^{-1} (f(A, X) - B) \\
&= 2 J_F^T \Sigma^{-1} (-R) \\
&= -2\lambda \Delta X_{gradient_descent}
\end{aligned}$$

Iterative reweighted least square – Generalized linear model

Generalized linear model extends the generalized least square to (1) non linear regression and (2) non Gaussian noise. Suppose we have the following non linear objective function :

$$L = \sum_{n=1}^{N} f(A_n X - b_n)$$

Taking derivative of objective function L wrt X, set it to zero and take transpose.

$$\begin{aligned}
0 &= \sum_{n=1}^{N} \frac{\partial f(e_n)}{\partial e_n} \frac{\partial e_n}{\partial X} && \text{define } e_n = A_n X - b_n \\
&= \sum_{n=1}^{N} w_n e_n \frac{\partial e_n}{\partial X} && \text{define } w_n = \frac{1}{e_n} \frac{\partial f(e_n)}{\partial e_n} \\
&= \frac{1}{2} \sum_{n=1}^{N} w_n \frac{\partial e_n{}^2}{\partial X} \\
&= \frac{1}{2} \frac{\partial}{\partial X} \sum_{n=1}^{N} w_n e_n{}^2 && \text{hence it becomes weighted least square (is this true?)}
\end{aligned}$$

This is an informal derivation provided by ASM. For details, please refer to the generalized least square section.

Remark

The nonlinear regression in Gauss Newton algorithm (or Levenberg Marquardt algorithm) is f(A,X), while the nonlinear regression in generalized linear model is $g^{-1}(AX)$, which is less general than the former, observation A and parameter X must occur in the form of AX in generalized linear model.

Special case : weighted least square in two dimensional space

Consider weighted least square in 2 dimensional space, i.e. matrix A is N×2 and matrix B is N×1.

$$X_{LS} = (A^T WA)^{-1}(A^T WB)$$

or $A^T WAX_{LS} = A^T WB$

where $A^T WA$ = outer product between A and A

$A^T WB$ = outer product between A and B

Suppose we want to fit a straight line with slope intercept, i.e. $X = [m, c]^T$, while matrices A and B are :

$$A = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \dots & \dots \\ x_N & 1 \end{bmatrix} \qquad B = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_N \end{bmatrix}$$

$$W = diag(w_1, w_2, \dots, w_N)$$

Therefore we have :

$$A^T WA = \begin{bmatrix} \sum_{n=1}^N w_n x_n x_n & \sum_{n=1}^N w_n x_n \\ \sum_{n=1}^N w_n x_n & \sum_{n=1}^N w_n \end{bmatrix} = \begin{bmatrix} S_{wxx} & S_{wx} \\ S_{wx} & S_w \end{bmatrix}$$ where S stands for sum

$$A^T WB = \begin{bmatrix} \sum_{n=1}^N w_n x_n y_n \\ \sum_{n=1}^N w_n y_n \end{bmatrix} = \begin{bmatrix} S_{wxy} \\ S_{wy} \end{bmatrix}$$

Using Cramer's rule :

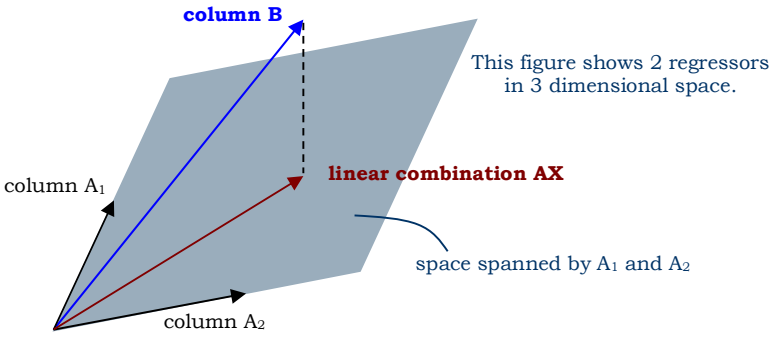
$$m = \frac{\det\begin{bmatrix} S_{wxy} & S_{wx} \\ S_{wy} & S_w \end{bmatrix}}{\det(A^T WA)} = \frac{S_{wxy}S_w - S_{wx}S_{wy}}{S_{wxx}S_w - S_{wx}S_{wx}} = \frac{\text{cov}(x,y)}{\text{var}(x)}$$

$$c = \frac{\det\begin{bmatrix} S_{wxx} & S_{wxy} \\ S_{wx} & S_{wy} \end{bmatrix}}{\det(A^T WA)} = \frac{S_{wxx}S_{wy} - S_{wx}S_{wxy}}{S_{wxx}S_w - S_{wx}S_{wx}} = \frac{S_{wxx}S_{wy} - S_{wx}S_{wxy}}{\text{var}(x)}$$

Geometric interpretation of least square in regressor space

Ordinary least square can be interpreted geometrically in regressor space (do not confuse with data space). Matrix A is considered as M column regressors in N dimensional space (N>M), B is considered as a single vector in N dimensional space, then the objective function min|B-AX| can be interpreted as minimizing the distance between vector B to a weighted sum of regressors in A, where X is the weight vector that we have to find (recall that AX is weighted sum of columns in A). Geometrically, the distance can be minimized when AX equals to the projection of B on the space spanned by columns in A,which is given by PB, where P is the projection matrix $A(A^TA)^{-1}A^T$. Therefore we have :

$$AX = proj_A(B)$$
$$= A(A^T A)^{-1} A^T B$$
$$\Rightarrow \quad X = (A^T A)^{-1} A^T B$$



Remark : We denote the projection of row vector B on the space spanned by rows of A as $proj_{A,row}(B)$ and the projection of column vector B on the space spanned by columns of A as $proj_{A,col}(B)$, then we have :

$$proj_{A,row}(B) = BA^T(AA^T)^{-1}A = BP_{A,row} \quad \text{where} \quad P_{Arow} = A^T(AA^T)^{-1}A$$

$$proj_{A,col}(B) = P_{A,col}B = A(A^T A)^{-1}A^T B \quad \text{where} \quad P_{Acol} = A(A^T A)^{-1}A^T$$

Least square with constraints

(linear constraint 1)	$\min_X \|AX-B\|_W^2$	such that $X_1 = X_1^*$	where $X_1 = column\_matrix(K\times 1)$
(linear constraint 2)	$\min_X \|AX-B\|_W^2$	such that $X_1 = CX_2 + D$	where $X_1 = column\_matrix(K\times 1)$
(linear constraint 3)	$\min_X \|AX-B\|_W^2$	such that $CX = D$	
(quadratic constraint 1)	$\min_{X,y} \|AX-ly\|_W^2$	such that $X^T X = 1$	where $l = ones(N\times 1)$ and $y = scalar = dist\_to\_origin$
(quadratic constraint 2a)	$\min_X \|AX-B\|_W^2$	such that $X^T X = 1$	
(quadratic constraint 2b)	$\min_X \|AX-B\|_W^2$	such that $X^T X = 1$	where $rank(A) = M-1$ (i.e. degenerated case)
(quadratic constraint 3)	$\min_X \|AX-B\|_W^2$	such that $X_1^T X_1 = 1$	where $X_1 = column\_matrix(K\times 1)$

Linear constraint 1

$$
\begin{aligned}
L &= (AX-B)^T W(AX-B) \\
&= (A_1X_1 + A_2X_2 - B)^T W(A_1X_1 + A_2X_2 - B) \\
&= (A_2X_2 - (B - A_1X_1^*))^T W(A_2X_2 - (B - A_1X_1^*))
\end{aligned}
$$

Taking derivative of objective function L wrt X_2, set it to zero and take transpose.

$$
\begin{aligned}
0 &= 2A_2{}^T W(A_2X_2 - (B - A_1X_1^*)) \\
0 &= (A_2{}^T WA_2)X_2 - (A_2{}^T W(B - A_1X_1^*)) \\
X_2 &= (A_2{}^T WA_2)^{-1}(A_2{}^T W(B - A_1X_1^*)) \qquad \text{see Remark\#}
\end{aligned}
$$

Linear constraint 2

$$
\begin{aligned}
L &= (AX-B)^T W(AX-B) \\
&= (A_1X_1 + A_2X_2 - B)^T W(A_1X_1 + A_2X_2 - B) \\
&= (A_1(CX_2 + D) + A_2X_2 - B)^T W(A_1(CX_2 + D) + A_2X_2 - B) \\
&= ((A_1C + A_2)X_2 - (B - A_1D))^T W((A_1C + A_2)X_2 - (B - A_1D))
\end{aligned}
$$

Taking derivative of objective function L wrt X_2, set it to zero and take transpose.

$$
\begin{aligned}
0 &= 2(A_1C + A_2)^T W((A_1C + A_2)X_2 - (B - A_1D)) \\
0 &= ((A_1C + A_2)^T W(A_1C + A_2))X_2 - ((A_1C + A_2)^T W(B - A_1D)) \\
X_2 &= ((A_1C + A_2)^T W(A_1C + A_2))^{-1}((A_1C + A_2)^T W(B - A_1D)) \qquad \text{see Remark\#}
\end{aligned}
$$

Binding X_1 to linear transformation of X_2 and solve for X_2 is equivalent to binding X_2 to linear transformation of X_1 and solve for X_1. The following shows an alternative solution.

$$
\begin{aligned}
L &= (AX-B)^T W(AX-B) \qquad \text{since } X_1 = CX_2 + D \\
&= (A_1X_1 + A_2X_2 - B)^T W(A_1X_1 + A_2X_2 - B) \\
&= (A_1X_1 + A_2(C'X_1 + D') - B)^T W(A_1X_1 + A_2(C'X_1 + D') - B) \\
&= ((A_1 + A_2C')X_1 - (B - A_2D'))^T W((A_1 + A_2C')X_1 - (B - A_2D'))
\end{aligned}
$$

$$
\begin{aligned}
&\Rightarrow \quad X_2 = (C^T C)^{-1}X_1 + (C^T D) \\
&\qquad X_2 = C'X_1 + D' \\
&\text{i.e.} \quad C' = C^T C \text{ and } D' = C^T D
\end{aligned}
$$

Taking derivative of objective function L wrt X_1, set it to zero and take transpose.

$$
\begin{aligned}
0 &= 2(A_1 + A_2C')^T W((A_1 + A_2C')X_1 - (B - A_2D')) \\
0 &= ((A_1 + A_2C')^T W(A_1 + A_2C'))X_1 - ((A_1 + A_2C')^T W(B - A_2D')) \\
X_2 &= ((A_1 + A_2C')^T W(A_1 + A_2C'))^{-1}((A_1 + A_2C')^T W(B - A_2D')) \qquad \text{see Remark\#}
\end{aligned}
$$

Remark\# *For both linear constraint 1 and 2, we have :*

$$
X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \qquad A = \begin{bmatrix} A_1 & A_2 \end{bmatrix}
$$

$$
X_1 = column\_matrix(K\times 1) \qquad A_1 = matrix(N\times K)
$$

$$
X_2 = column\_matrix((M-K)\times 1) \qquad A_2 = matrix(N\times (M-K))
$$

Linear constraint 3

Suppose the rank of C is K (K < M), thus equality constraint CX=D means a dimension reduction of the solution space from $\Re^M$ to $\Re^{M-K}$. Therefore, our first step is either to (1) find orthonormal basis of the constrained solution space in $\Re^{M-K}$ or equivalently (2) find linear transformation that maps row data matrix A (which contains N data in $\Re^M$) in original solution space to row data matrix A' (which contains N data in $\Re^{M-K}$) to new solution space. This is done by performing SVD on C matrix.

$$
\begin{aligned}
C &= USV^T = USZ \\
U &= matrix(K \times K) \\
S &= matrix(K \times M) \\
V &= matrix(M \times M) \\
V^T &= \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} \\
Z_1 &= matrix(K \times M) = \text{span of constraint} \\
Z_2 &= matrix((M-K) \times M) = \text{null space of constraint}
\end{aligned}
$$

The original regression AX=B under constraint that CX=D is then transformed to a new regression. Please note that we apply transformation V to the RHS of row data matrix A and apply transformation V^T to the LHS of column vector X, both denote the **same transformation**.

$$
\begin{aligned}
& AX = B \text{ such that } CX = D \\
\Rightarrow\ & AVV^T X = B \text{ such that } USV^T X = D \\
\Rightarrow\ & A'X' = B \text{ such that } USX' = D \quad \text{where } A' = AV \text{ and } X' = V^T X \\
\Rightarrow\ & A'X' = B \text{ such that } X' = S^{-1}U^T D \\
\Rightarrow\ & A'X' = B \text{ such that } \begin{bmatrix} X'_1 \\ X'_2 \end{bmatrix} = \begin{bmatrix} S_1^{-1} \\ S_2^{-1} \end{bmatrix} U^T D \quad \text{ignore null space of constraint} \\
\Rightarrow\ & A'X' = B \text{ such that } X'_1 = S_1^{-1}U^T D
\end{aligned}
$$

where
$$
\begin{aligned}
X'_1 &= matrix(K \times 1) = \text{constrained components of X} \\
X'_2 &= matrix((M-K) \times 1) = \text{free-to-move components of X} \\
S_1^{-1} &= matrix(K \times K) = \begin{bmatrix} 1/s_1 & 0 & 0 & \dots & 0 \\ 0 & 1/s_2 & 0 & \dots & 0 \\ 0 & 0 & 1/s_3 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1/s_K \end{bmatrix} \\
S_2^{-1} &= zeros((M-K) \times K)
\end{aligned}
$$

In short, we have performed such a transformation :

$$
\begin{aligned}
& \arg\min_X (AX-B)^T W (AX-B) \\
& s.t.\ \ CX = D
\end{aligned}
\quad \Rightarrow \quad
\begin{aligned}
& \arg\min_{X'} (A'X'-B)^T W (A'X'-B) \\
& s.t.\ \ X'_1 = S_1^{-1}U^T D
\end{aligned}
$$

The transformed regression is exactly the same as linear constraint 1. Therefore we have :

$$
\begin{aligned}
L &= (AX-B)^T W (AX-B) \\
&= (A'X'-B)^T W (A'X'-B) \\
&= (A'_1 X'_1 + A'_2 X'_2 - B)^T W (A'_1 X'_1 + A'_2 X'_2 - B) \\
&= (A'_2 X'_2 - (B - A'_1 X'_1))^T W (A'_2 X'_2 - (B - A'_1 X'_1))
\end{aligned}
$$

Taking derivative of objective function L wrt X'_2, set it to zero and take transpose.

$$
\begin{aligned}
0 &= 2A'^T_2 W (A'_2 X'_2 - (B - A'_1 X'_1)) \\
0 &= (A'^T_2 W A'_2) X'_2 - (A'^T_2 W (B - A'_1 X'_1)) \\
X'_2 &= (A'^T_2 W A'_2)^{-1} - (A'^T_2 W (B - A'_1 X'_1))
\end{aligned}
$$

Finally, we have :

$$
X = VX' = \begin{bmatrix} VX'_1 \\ VX'_2 \end{bmatrix} = \begin{bmatrix} V(S_1^{-1}U^T D) \\ V(A'^T_2 W A'_2)^{-1}(A'^T_2 W (B - A'_1 X'_1)) \end{bmatrix}
$$

Quadratic constraint 1

Lagrange function is set up as the following :

$$L = (AX-ly)^T W(AX-ly)+\lambda(1-X^T X) \quad \text{where } l=ones(N\times 1) \text{ and } y=scalar=dist\_to\_origin$$

Taking derivative of objective function L wrt y, set it to zero and take transpose.

$$\begin{aligned} 0 &= l^T W(AX-ly) \\ 0 &= (l^T WAX)-(l^T Wl)y \\ y &= (l^T Wl)^{-1}(l^T WAX) && \text{where } l^T Wl=\sum_{n=1}^{N} w_n = scalar \\ &= \frac{l^T WAX}{l^T Wl} && \text{where } l^T WA=\sum_{n=1}^{N} w_n A_n \\ &= \bar{A}X && \text{where } \bar{A}=(\sum_{n=1}^{N} w_n A_n)/(\sum_{n=1}^{N} w_n) \end{aligned}$$

Substitute this result into objective function L :

$$\begin{aligned} L &= (AX-ly)^T W(AX-ly)+\lambda(1-X^T X) \\ &= (AX-l\bar{A}X)^T W(AX-l\bar{A}X)+\lambda(1-X^T X) \\ &= ((A-l\bar{A})X)^T W((A-l\bar{A})X)+\lambda(1-X^T X) \end{aligned}$$

Taking derivative of objective function L wrt X, set it to zero and take transpose.

$$\begin{aligned} 0 &= 2(A-l\bar{A})^T W(A-l\bar{A})X-2\lambda X \\ \lambda X &= (A-l\bar{A})^T W(A-l\bar{A})X \\ \lambda X &= \mathrm{cov}(A)X && \text{suppose } l^T Wl=1 \end{aligned}$$

Therefore X and λ are the major eigenvector and the major eigenvalue of cov(A), and since cov(A) is a covariance matrix, X is principal component of A. In conclusion :

$$\arg\min_{X,y}\|AX-ly\|_W^2 \text{ such that } X^T X=1 \quad \Leftrightarrow \quad X=eigenvector(\mathrm{cov}(A)) \text{ and } y=\bar{A}X$$

$$\Leftrightarrow \quad X=principal\_component(A) \text{ and } y=\bar{A}X$$

Quadratic constraint 2a

Lagrange function is set up as the following :

$$L = (AX-B)^T W(AX-B)+\lambda(1-X^T X)$$

Taking derivative of objective function L wrt X, set it to zero and take transpose.

$$\begin{aligned} 0 &= 2A^T W(AX-B)-2\lambda X \\ 0 &= A^T WAX-A^T WB-\lambda X \\ 0 &= (A^T WA-\lambda I)X-(A^T WB) \\ X &= (A^T WA-\lambda I)^{-1}(A^T WB) \end{aligned}$$

Taking derivative of objective function L wrt λ, set it to zero.

$$\begin{aligned} 1 &= X^T X \\ &= ((A^T WA-\lambda I)^{-1}(A^T WB))^T((A^T WA-\lambda I)^{-1}(A^T WB)) \\ f(\lambda) &= ((A^T WA-\lambda I)^{-1}(A^T WB))^T((A^T WA-\lambda I)^{-1}(A^T WB))-1 \end{aligned}$$

This is a M[th] order polynomial of λ, which can be solved by Newton Raphson. However if the problem degenerates :

$$rank(A) = rank(A^T A) = M-1$$

then there will be infinity many solutions, then it is better to go for quadratic constraint 2b in the next section.

Quadratic constraint 2b
Suppose A is not full rank :

$$
\begin{array}{lll}
rank(A) & = & K \\
A & = & [A_1 \quad A_2] \\
A_2 & = & A_1 V \qquad \text{(i.e. expressed in terms of linear combination of columns in } A_1\text{)} \\
\Rightarrow \quad V & = & (A_1{}^T A_1)^{-1}(A_1{}^T A_2)
\end{array}
$$

where $A_1 = matrix(N \times K)$ (i.e. N row data vectors in K dimension)
$A_2 = matrix(N \times (M-K))$ (i.e. N row data vectors in M–K dimension)
$V = matrix(K \times (M-K))$

then
$$
\begin{aligned}
L &= (AX-B)^T W (AX-B) \\
&= (A_1X_1 + A_2X_2 - B)^T W (A_1X_1 + A_2X_2 - B) \\
&= (A_1X_1 + A_1VX_2 - B)^T W (A_1X_1 + A_1VX_2 - B) \\
&= (A_1(X_1 + VX_2) - B)^T W (A_1(X_1 + VX_2) - B)
\end{aligned}
$$

where $X_1 = matrix(K \times 1)$
$X_2 = matrix((M-K) \times 1)$

Taking derivative of objective function L wrt X_1+VX_2, set it to zero and take transpose.

$$
\begin{aligned}
0 &= 2A_1{}^T W (A_1(X_1 + VX_2) - B)) \\
0 &= (A_1{}^T WA_1)(X_1 + VX_2) - (A_1{}^T WB) \\
X_1 &= (A_1{}^T WA_1)^{-1}(A_1{}^T WB) - VX_2
\end{aligned}
$$

which represents K linear equations, we need M–K extra information to solve for all M parameters. For example : if K = M–1, and we are given that :

$$
\begin{aligned}
0 &= f(x_1, x_2, x_3, ..., x_M) \\
0 &= f(X_1, X_2) \\
0 &= f((A_1{}^T WA_1)^{-1}(A_1{}^T WB) - VX_2, X_2) \\
\Rightarrow \quad 0 &= f((A_1{}^T WA_1)^{-1}(A_1{}^T WB) - (A_1{}^T A_1)^{-1}(A_1{}^T A_2)X_2, X_2)
\end{aligned}
$$

where X_2 is a scalar, we can solve f for X_2 using numerical method, then get X by :

$$
X_1 = (A_1{}^T WA_1)^{-1}(A_1{}^T WB) - (A_1{}^T A_1)^{-1}(A_1{}^T A_2)X_2
$$

Quadratic constraint 3
Lagrange function is set up as the following :

$$
\begin{aligned}
L &= (AX-B)^T W (AX-B) + \lambda(1 - X_1{}^T X_1) \\
&= (A_1X_1 + A_2X_2 - B)^T W (A_1X_1 + A_2X_2 - B) + \lambda(1 - X_1{}^T X_1)
\end{aligned}
$$

where $X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$ $\qquad A = [A_1 \quad A_2]$

$X_1 = column\_matrix(K \times 1)$ $\qquad A_1 = matrix(N \times K)$
$X_2 = column\_matrix((M-K) \times 1)$ $\qquad A_2 = matrix(N \times (M-K))$

Taking derivative of objective function L wrt X_2, set it to zero and take transpose.

$$
\begin{aligned}
0 &= 2A_2{}^T W (A_2X_2 - (B - A_1X_1)) \\
0 &= (A_2{}^T WA_2)X_2 - (A_2{}^T W(B - A_1X_1)) \\
X_2 &= (A_2{}^T WA_2)^{-1}(A_2{}^T W(B - A_1X_1)) \\
&= (A_2{}^T WA_2)^{-1}(A_2{}^T WB - A_2{}^T WA_1X_1) \\
&= -(A_2{}^T WA_2)^{-1}(A_2{}^T WA_1)X_1 + (A_2{}^T WA_2)^{-1}(A_2{}^T WB) \\
&= CX_1 + D \qquad \text{(equation 1)}
\end{aligned}
$$

where $C = -(A_2{}^T WA_2)^{-1}(A_2{}^T WA_1)$
$D = +(A_2{}^T WA_2)^{-1}(A_2{}^T WB)$

Substitute this result into objective function L :

$$\begin{aligned} L &= (A_1X_1 + A_2(CX_1 + D) - B)^T W(A_1X_1 + A_2(CX_1 + D) - B) + \lambda(1 - X_1{}^T X_1) \\ &= ((A_1 + A_2C)X_1 - (B - A_2D))^T W((A_1 + A_2C)X_1 - (B - A_2D)) + \lambda(1 - X_1{}^T X_1) \\ &= (EX_1 - F)^T W(EX_1 - F) + \lambda(1 - X_1{}^T X_1) \end{aligned}$$

where $E = A_1 + A_2C$

$F = B - A_2D$

This is the same formulation as quadratic constraint 2a or 2b, that is :

$$\min_{X_1} \|EX_1 - F\|_W^2 \text{ such that } X_1{}^T X_1 = 1 \qquad \text{(transformed problem)}$$

$E = matrix(N \times K)$

$F = matrix(N \times 1)$

Case 1 : when E is full rank

Taking derivative of objective function L wrt X_1, set it to zero and take transpose.

$$\begin{aligned} 0 &= 2E^T W(EX_1 - F) - 2\lambda X_1 \\ 0 &= E^T WEX_1 - E^T WF - \lambda X_1 \\ 0 &= (E^T WE - \lambda I)X_1 - (E^T WF) \\ X_1 &= (E^T WE - \lambda I)^{-1}(E^T WF) \qquad \text{(equation 2)} \end{aligned}$$

Taking derivative of objective function L wrt λ, set it to zero.

$$\begin{aligned} 1 &= X_1{}^T X_1 \\ &= ((E^T WE - \lambda I)^{-1}(E^T WF))^T((E^T WE - \lambda I)^{-1}(E^T WF)) \\ f(\lambda) &= ((E^T WE - \lambda I)^{-1}(E^T WF))^T((E^T WE - \lambda I)^{-1}(E^T WF)) - 1 \qquad \text{(equation 3)} \end{aligned}$$

This is a M^{th} order polynomial in λ, which can be solved by Newton Raphson. Thus the solution is :

- solve equation 3 for λ and then
- solve equation 2 for X_1 and finally
- solve equation 1 for X_2.

Case 2 : when rank of E is K–1

$$\begin{aligned} rank(E) &= K - 1 \\ E &= [E_1 \quad E_2] \\ E_2 &= E_1V \\ \Rightarrow \quad V &= (E_1{}^T E_1)^{-1}(E_1{}^T E_2) \end{aligned}$$

then

$$\begin{aligned} L &= (EX_1 - F)^T W(EX_1 - F) \\ &= (E_1X_{11} + E_2X_{12} - F)^T W(E_1X_{11} + E_2X_{12} - F) \\ &= (E_1X_{11} + E_1VX_{12} - F)^T W(E_1X_{11} + E_1VX_{12} - F) \\ &= (E_1(X_{11} + VX_{12}) - F)^T W(E_1(X_{11} + VX_{12}) - F)) \end{aligned}$$

Taking derivative of objective function L wrt X_{11}+VX_{12}, set it to zero and take transpose.

$$\begin{aligned} 0 &= 2E_1{}^T W(E_1(X_{11} + VX_{12}) - F)) \\ 0 &= (E_1{}^T WE_1)(X_{11} + VX_{12}) - (E_1{}^T WF) \\ X_{11} &= (E_1{}^T WE_1)^{-1}(E_1{}^T WF) - VX_{12} \end{aligned}$$

which represents K–1 linear equations, we need one extra information to solve for all K parameters.

$$\begin{aligned} 1 &= X_1{}^T X_1 \\ &= X_{11}{}^T X_{11} + X_{12}{}^T X_{12} \\ &= ((E_1{}^T WE_1)^{-1}(E_1{}^T WF) - VX_{12})^T((E_1{}^T WE_1)^{-1}(E_1{}^T WF) - VX_{12}) + X_{12}{}^T X_{12} \\ f(x) &= ((E_1{}^T WE_1)^{-1}(E_1{}^T WF) - VX_{12})^T((E_1{}^T WE_1)^{-1}(E_1{}^T WF) - VX_{12}) + X_{12}{}^T X_{12} - 1 \end{aligned}$$

where X_{12} is a scalar, we can solve f for X_{12} using numerical method. Lets consider a special case : K=2.

Thus the solution is (replace capital letters by small letters for scalars) :

$$
\begin{aligned}
v &= (E_1{}^T E_1)^{-1}(E_1{}^T E_2) \\
x_{11} &= (E_1{}^T W E_1)^{-1}(E_1{}^T W F) - v x_{12} \\
&= u - v x_{12} && \text{(equation 4)} \\
x_{11}^2 + x_{22}^2 &= 1 && \text{(equation 5)}
\end{aligned}
$$

Hence we need to solve the simultaneous equation 4 and 5, note that u and v are known.

$$
\begin{aligned}
1 &= (u - v x_{12})^2 + (x_{12})^2 \\
0 &= (1+v^2)x_{12}^2 - 2uvx_{12} + u^2 - 1 \\
x_{12} &= \frac{2uv \pm \sqrt{(2uv)^2 - 4(1+v^2)(u^2-1)}}{2(1+v^2)} \\
&= \frac{2uv \pm \sqrt{4u^2v^2 - 4u^2v^2 - 4u^2 + 4v^2 + 4}}{2(1+v^2)} = \frac{uv \pm \sqrt{1+v^2-u^2}}{1+v^2} \\
x_{11} &= u - v\frac{uv \pm \sqrt{1+v^2-u^2}}{1+v^2} = \frac{u \mp v\sqrt{1+v^2-u^2}}{1+v^2}
\end{aligned}
$$

Finally solve equation 1 for X_2.

General least square vs General linear model vs Generalized linear model
It is easy to get confused by these three linear models, they are different :

- general least square = least square that considers convariance among model parameters
- general linear model = least square with design matrix
- generalized linear model = iterative reweighted least square (generalize to non Gauss noise by link function)

More about general linear model
General linear model is a linear model B=AX+ε with (1) single or multiple regressors (i.e. variables X) and (2) univariate or multivariate Gaussian noise ε. Matrix A is called **design matrix**, which can either be dense or sparse, depending on specific applications. Noise ε can be uncorrelated or correlated, homoskedastic or heteroskedastic, different settings result in different implementations :

- uncorrelated and homoskedastic = ordinary least square
- uncorrelated and heteroskedastic = weighted least square
- correlated = general least square

When noise ε does not follow Gaussian distribution, it becomes generalized linear model, which can be implemented with iterative reweighted least square. Please read document for 'Generalized linear model'. Here are some examples for design matrix of general linear model (from algorithm team, vision group, ASM).

Example : concentric circle fitting
Suppose there are M concentric circles, there are M+2 regressors.

$$
\begin{aligned}
r_m^2 &= (x - x_c)^2 + (y - y_c)^2 && \forall m \in [1, M] \\
r_m^2 &= x^2 - 2xx_c + x_c^2 + y^2 - 2yy_c + y_c^2 && \forall m \in [1, M] \\
x^2 + y^2 &= 2x(x_c) + 2y(y_c) - (r_m^2 - x_c^2 - y_c^2) && \forall m \in [1, M]
\end{aligned}
$$

$$
A = \begin{bmatrix} 2x_1 & 2y_1 & -\delta_{c_1} \\ 2x_2 & 2y_2 & -\delta_{c_2} \\ 2x_3 & 2y_3 & -\delta_{c_3} \\ \dots & \dots & \dots \\ 2x_N & 2y_N & -\delta_{c_N} \end{bmatrix} \quad X = \begin{bmatrix} x_c \\ y_c \\ r_1^2 - x_c^2 - y_c^2 \\ \dots \\ r_M^2 - x_c^2 - y_c^2 \end{bmatrix} \quad B = \begin{bmatrix} x_1^2 + y_1^2 \\ x_2^2 + y_2^2 \\ x_3^2 + y_3^2 \\ \dots \\ x_N^2 + y_N^2 \end{bmatrix}
$$

where δ_m = 1×M matrix with value one at location m, and zero otherwise
c_n = circle that data point n belongs to

Example : rectangle fitting

There are two parallel line pairs in a rectangle, having four intercepts while sharing the same orientation, hence there are five regressors in total.

$$y = mx + c_1 \qquad y = -m^{-1}x + c_2 \qquad \Rightarrow \qquad x = -my + mc_2$$
$$y = mx + c_3 \qquad y = -m^{-1}x + c_4 \qquad \Rightarrow \qquad x = -my + mc_4$$

Suppose there are 4 data points, each belongs to a line on the rectangle, then we have :

$$A = \begin{bmatrix} x_1 & 1 & 0 & 0 & 0 \\ -y_2 & 0 & 1 & 0 & 0 \\ x_3 & 0 & 0 & 1 & 0 \\ -y_4 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad X = \begin{bmatrix} m \\ c_1 \\ mc_2 \\ c_3 \\ mc_4 \end{bmatrix} \qquad B = \begin{bmatrix} y_1 \\ x_2 \\ y_3 \\ x_4 \end{bmatrix}$$

Example : homogenous inspection

Suppose we have a region defined by four corners (x_L,y_L), (x_L,y_U), (x_U,y_L) and (x_U,y_U), the intensity at the four corners are a_{00}, a_{01}, a_{10} and a_{11} respectively, we have :

$$A = \begin{bmatrix} (1-r_{x_1})(1-r_{y_1}) & (1-r_{x_1})r_{y_1} & r_{x_1}(1-r_{y_1}) & r_{x_1}r_{y_1} \\ (1-r_{x_2})(1-r_{y_2}) & (1-r_{x_2})r_{y_2} & r_{x_2}(1-r_{y_2}) & r_{x_2}r_{y_2} \\ (1-r_{x_3})(1-r_{y_3}) & (1-r_{x_3})r_{y_3} & r_{x_3}(1-r_{y_3}) & r_{x_3}r_{y_3} \\ \dots & \dots & \dots & \dots \\ (1-r_{x_N})(1-r_{y_N}) & (1-r_{x_N})r_{y_N} & r_{x_N}(1-r_{y_N}) & r_{x_N}r_{y_N} \end{bmatrix} \qquad X = \begin{bmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{bmatrix}$$

$$B = \begin{bmatrix} ins(x_1, y_1) \\ ins(x_2, y_2) \\ ins(x_3, y_3) \\ \dots \\ ins(x_N, y_N) \end{bmatrix} \qquad \Rightarrow \qquad \begin{bmatrix} r_{x_n} \\ r_{y_n} \end{bmatrix} = \begin{bmatrix} (x_n - x_L)/(x_U - x_L) \\ (y_n - y_L)/(y_U - y_L) \end{bmatrix}$$

Example : template inspection

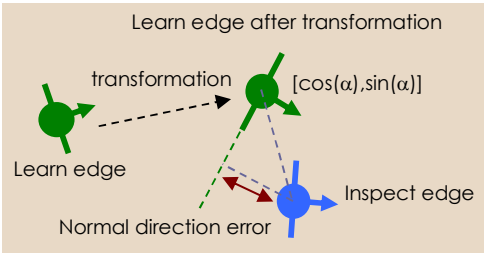
Suppose there are M learn images, with intensity $Lrn_m(x,y)$ at location (x,y), then we have :

$$A = \begin{bmatrix} lrn_1(x_1, y_1) & lrn_2(x_1, y_1) & \dots & lrn_M(x_1, y_1) & 1 \\ lrn_1(x_2, y_2) & lrn_2(x_2, y_2) & \dots & lrn_M(x_2, y_2) & 1 \\ lrn_1(x_3, y_3) & lrn_2(x_3, y_3) & \dots & lrn_M(x_3, y_3) & 1 \\ \dots & \dots & \dots & \dots & \dots \\ lrn_1(x_N, y_N) & lrn_2(x_N, y_N) & \dots & lrn_M(x_N, y_N) & 1 \end{bmatrix} \qquad X = \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_M \\ b \end{bmatrix}$$

$$B = \begin{bmatrix} ins(x_1, y_1) \\ ins(x_2, y_2) \\ ins(x_3, y_3) \\ \dots \\ ins(x_N, y_N) \end{bmatrix}$$

Example : shape alignment with normal distance

$$L = \sum_{n=1}^{N} [(x_n^{ins} - x_n^{lrnTrans}) \times \cos(\alpha_n^{lrn}) + (y_n^{ins} - y_n^{lrnTrans}) \times \sin(\alpha_n^{lrn})]^2$$
$$= \sum_{n=1}^{N} \begin{bmatrix} (x_n^{ins} - (a \times x_n^{lrn} - b \times y_n^{lrn} + \Delta x)) \times \cos(\alpha_n^{lrn}) + \\ (y_n^{ins} - (b \times x_n^{lrn} + a \times y_n^{lrn} + \Delta y)) \times \sin(\alpha_n^{lrn}) \end{bmatrix}^2$$



With the above objective function for alignment, we have design matrix :

$$A = \begin{bmatrix} x_1^{lrn}\cos(\alpha_1^{lrn}) + y_1^{lrn}\sin(\alpha_1^{lrn}) & -y_1^{lrn}\cos(\alpha_1^{lrn}) + x_1^{lrn}\sin(\alpha_1^{lrn}) & \cos(\alpha_1^{lrn}) & \sin(\alpha_1^{lrn}) \\ x_2^{lrn}\cos(\alpha_2^{lrn}) + y_2^{lrn}\sin(\alpha_2^{lrn}) & -y_2^{lrn}\cos(\alpha_2^{lrn}) + x_2^{lrn}\sin(\alpha_2^{lrn}) & \cos(\alpha_2^{lrn}) & \sin(\alpha_1^{lrn}) \\ \dots & \dots & \dots & \dots \\ x_N^{lrn}\cos(\alpha_N^{lrn}) + y_N^{lrn}\sin(\alpha_N^{lrn}) & -y_N^{lrn}\cos(\alpha_N^{lrn}) + x_N^{lrn}\sin(\alpha_N^{lrn}) & \cos(\alpha_N^{lrn}) & \sin(\alpha_N^{lrn}) \end{bmatrix}$$

$$B = \begin{bmatrix} x_1^{ins}\cos(\alpha_1^{lrn}) + y_1^{ins}\sin(\alpha_1^{lrn}) \\ x_2^{ins}\cos(\alpha_2^{lrn}) + y_2^{ins}\sin(\alpha_2^{lrn}) \\ \dots \\ x_N^{ins}\cos(\alpha_N^{lrn}) + y_N^{ins}\sin(\alpha_N^{lrn}) \end{bmatrix} \qquad X = \begin{bmatrix} a \\ b \\ \Delta x \\ \Delta y \end{bmatrix} \qquad \text{and} \qquad a^2 + b^2 = 1$$

Bias and variance of regression estimation

	$AX+\varepsilon$	=	B
where	Σ	=	$E(\varepsilon\varepsilon^T)$
then	X_{LS}	=	$(A^TA)^{-1}(A^TB)$
	X_{GLS}	=	$(A^T\Sigma^{-1}A)^{-1}(A^T\Sigma^{-1}B)$

where X is deterministic, ground truth,
A is deterministic, observation,
B is stochastic, dependent on ε,
Σ is stochastic.

We then have :

X_{LS}	=	$(A^TA)^{-1}(A^TB)$	
	=	$(A^TA)^{-1}(A^T(AX+\varepsilon))$	
	=	$(A^TA)^{-1}(A^TA)X+(A^TA)^{-1}(A^T\varepsilon)$	
	=	$X+(A^TA)^{-1}(A^T\varepsilon)$	(equation 6a)
X_{GLS}	=	$(A^T\Sigma^{-1}A)^{-1}(A^T\Sigma^{-1}B)$	
	=	$(A^T\Sigma^{-1}A)^{-1}(A^T\Sigma^{-1}(AX+\varepsilon))$	
	=	$(A^T\Sigma^{-1}A)^{-1}(A^T\Sigma^{-1}A)X+(A^T\Sigma^{-1}A)^{-1}(A^T\Sigma^{-1}\varepsilon)$	
	=	$X+(A^T\Sigma^{-1}A)^{-1}(A^T\Sigma^{-1}\varepsilon)$	(equation 6b)

Here are the bias and variance of using least square for correlated noise.

⇒ $E[X_{LS}]$	=	$E[X+(A^TA)^{-1}(A^T\varepsilon)]$	
	=	$X+(A^TA)^{-1}A^TE[\varepsilon]$	
	=	X	(i.e. ground truth, hence unbiased)
⇒ $\mathrm{cov}[X_{LS}]$	=	$E[(X_{LS}-E[X_{LS}])(X_{LS}-E[X_{LS}])^T]$	
	=	$E[(X_{LS}-X)(X_{LS}-X)^T]$	
	=	$E[(A^TA)^{-1}(A^T\varepsilon)((A^TA)^{-1}(A^T\varepsilon))^T]$	(using equation 6a)
	=	$E[(A^TA)^{-1}A^T\varepsilon\varepsilon^T((A^TA)^{-1}A^T)^T]$	
	=	$(A^TA)^{-1}A^T\Sigma((A^TA)^{-1}A^T)^T$	
	=	$(A^TA)^{-1}A^T\Sigma A((A^TA)^{-1})^T$	
	=	$(A^TA)^{-1}A^T\Sigma A(A^TA)^{-1}$	(using $((A^TA)^{-1})^T=((A^TA)^T)^{-1}=(A^TA)^{-1}$)

Here are the bias and variance of using general least square for correlated noise.

⇒ $E[X_{GLS}]$	=	$E[X+(A^T\Sigma^{-1}A)^{-1}(A^T\Sigma^{-1}\varepsilon)]$	
	=	$X+(A^T\Sigma^{-1}A)^{-1}(A^T\Sigma^{-1})E[\varepsilon]$	
	=	X	(i.e. ground truth, hence unbiased)
⇒ $\mathrm{cov}[X_{GLS}]$	=	$E[(X_{GLS}-E[X_{GLS}])(X_{GLS}-E[X_{GLS}])^T]$	
	=	$E[(X_{GLS}-X)(X_{GLS}-X)^T]$	
	=	$E[(A^T\Sigma^{-1}A)^{-1}(A^T\Sigma^{-1}\varepsilon)((A^T\Sigma^{-1}A)^{-1}(A^T\Sigma^{-1}\varepsilon))^T]$	(using equation 6b)
	=	$E[(A^T\Sigma^{-1}A)^{-1}(A^T\Sigma^{-1})\varepsilon\varepsilon^T((A^T\Sigma^{-1}A)^{-1}(A^T\Sigma^{-1}))^T]$	
	=	$(A^T\Sigma^{-1}A)^{-1}(A^T\Sigma^{-1})\Sigma((A^T\Sigma^{-1}A)^{-1}(A^T\Sigma^{-1}))^T$	(equation 7, using $E[\varepsilon\varepsilon^T]=\Sigma$)
	=	$(A^T\Sigma^{-1}A)^{-1}(A^T)((A^T\Sigma^{-1}A)^{-1}(A^T\Sigma^{-1}))^T$	
	=	$(A^T\Sigma^{-1}A)^{-1}(A^T)(A^T\Sigma^{-1})^T((A^T\Sigma^{-1}A)^{-1})^T$	
	=	$(A^T\Sigma^{-1}A)^{-1}(A^T\Sigma^{-1}A)(A^T\Sigma^{-1}A)^{-1}$	(using $\Sigma^T=\Sigma$ and $(A^T\Sigma^{-1}A)^T=A^T\Sigma^{-1}A$)
	=	$(A^T\Sigma^{-1}A)^{-1}$	

This is reasonable, as A^TA is the outer product (weighted covariance of data matrix A), the above formula shows that the covariance of estimated parameters is inversely proportional to the **spread of data**, and directly proportional to the **noise of data**. This is intuitive in straight line fitting. We can also see that under correlated noise, ordinary least square estimation can still provide an unbiased result, yet its variance is not the minimum (generalized least square estimation is a better choice), this can be proved by the Gauss Markov Theorem as shown in the next page.

Gauss Markov Theorem

Gauss Markov Theorem states that given a model :

$$AX + \varepsilon = B$$

$$\Rightarrow \quad X_{GLS} = (A^T \Sigma^{-1} A)^{-1} (A^T \Sigma^{-1} B)$$

and $\Sigma = E(\varepsilon \varepsilon^T)$

where X is deterministic, ground truth,
A is deterministic, observation,
B is stochastic, dependent on ε,
Σ is stochastic.

then least square estimate is the best linear unbiased estimate (BLUE) of X, in the sense that (1) it is unbiased and (2) it has minimum variance among all linear unbiased estimators.

Proof 1 : Direct proof

Suppose UB is an unbiased linear estimate of X, where U is a M×N matrix, we then have :

$$X_{unbias} = UB$$
$$= U(AX + \varepsilon)$$

$$\Rightarrow \quad E[X_{unbias}] = X$$
$$E[U(AX + \varepsilon)] = X$$
$$UAX = X$$
$$UA = I$$

$$\Rightarrow \quad \text{cov}[X_{unbias}] = E[(X_{unbias} - E[X_{unbias}])(X_{unbias} - E[X_{unbias}])^T]$$
$$= E[(X_{unbias} - X)(X_{unbias} - X)^T]$$
$$= E[(UAX + U\varepsilon - X)(UAX + U\varepsilon - X)^T]$$
$$= E[(U\varepsilon)(U\varepsilon)^T] \qquad \text{(since } UA = I \text{ for unbiased estimation)}$$
$$= E[U\varepsilon\varepsilon^T U^T]$$
$$= U\Sigma U^T$$
$$= (U' + (A^T\Sigma^{-1}A)^{-1}(A^T\Sigma^{-1}))\Sigma(U' + (A^T\Sigma^{-1}A)^{-1}(A^T\Sigma^{-1}))^T \qquad \text{(since } U' = U - (A^T\Sigma^{-1}A)^{-1}(A^T\Sigma^{-1})\text{)}$$
$$= (U'\Sigma U'^T) + (A^T\Sigma^{-1}A)^{-1}(A^T\Sigma^{-1}))\Sigma(A^T\Sigma^{-1}A)^{-1}(A^T\Sigma^{-1})^T \qquad \text{(using remark)}$$
$$= (U'\Sigma U'^T) + \text{cov}(X_{GLS}) \qquad \text{(using equation 7)}$$

We need to show that every term inside cov(X_{unbias}) is larger than the corresponding term inside cov(X_{GLS}). The term in y-th column and x-th row of a matrix can be extracted by the following product :

$$\text{cov}(X_{unbias})_{y,x} = \delta_y^T \text{cov}(X_{unbias})\delta_x \qquad \text{(where } \delta_x \text{ and } \delta_y \text{ are M×1 column matrix)}$$
$$= \delta_y^T ((U'\Sigma U'^T) + \text{cov}(X_{GLS}))\delta_x$$
$$= \delta_y^T (U'\Sigma U'^T)\delta_x + \delta_y^T \text{cov}(X_{GLS})\delta_x$$
$$\geq \delta_y^T \text{cov}(X_{GLS})\delta_x \qquad \text{(since } U'\Sigma U'^T \text{ is semi positive definite)}$$
$$= \text{cov}(X_{GLS})_{y,x} \qquad \text{(since } Z^T(U'\Sigma U'^T)Z \geq 0 \text{ for all non zero Z)}$$

Remark :

$$U' = U - (A^T\Sigma^{-1}A)^{-1}(A^T\Sigma^{-1})$$
$$U'A = UA - (A^T\Sigma^{-1}A)^{-1}(A^T\Sigma^{-1}A)$$
$$= UA - I$$
$$= 0$$

$$U'\Sigma((A^T\Sigma^{-1}A)^{-1}(A^T\Sigma^{-1}))^T = U'\Sigma(A\Sigma^{-1})^T((A^T\Sigma^{-1}A)^{-1})^T$$
$$= U'\Sigma\Sigma^{-1}A((A^T\Sigma^{-1}A)^{-1})^T$$
$$= U'A((A^T\Sigma^{-1}A)^{-1})^T$$
$$= 0$$
$$((A^T\Sigma^{-1}A)^{-1}(A^T\Sigma^{-1}))\Sigma U'^T = (A^T\Sigma^{-1}A)^{-1}(A^T\Sigma^{-1})\Sigma U'^T$$
$$= (A^T\Sigma^{-1}A)^{-1}A^T U'^T$$
$$= (A^T\Sigma^{-1}A)^{-1}(U'A)^T$$
$$= 0$$

Proof 2 : Indirect proof (in fact, I do not prefer this one)

Instead of direct estimation of X, its proof starts with estimation of a linear transformation of model parameters WX, where W is a given K×M matrix, which maps $\Re^M \rightarrow \Re^K$. Suppose UB is an unbiased linear estimate of WX, where U is a K×N matrix, we then have :

$$\begin{aligned}
\Rightarrow \quad E(UB) &= WX \\
E(U(AX+\varepsilon)) &= WX \\
UAX &= WX \\
UA &= W \qquad \text{(equation 8)}
\end{aligned}$$

Since $A^T\Sigma^{-1}A$ is full rank in M dimensional space, W can be expressed as the result of applying linear transformation V on $A^T\Sigma^{-1}A$'s columns, where V is a K×M matrix.

$$\begin{aligned}
W &= VA^T\Sigma^{-1}A \qquad \text{(equation 9)} \\
WX_{GLS} &= V(A^T\Sigma^{-1}A)(A^T\Sigma^{-1}A)^{-1}(A^T\Sigma^{-1}B) \\
&= V(A^T\Sigma^{-1}B) \qquad \text{(equation 10)}
\end{aligned}$$

Lets find the variance of UB, and show that it is greater than the variance of WX_{GLS}. Please beware of our assumption that A is non stochastic, while B is stochastic.

$$\begin{aligned}
\Rightarrow \quad \mathrm{var}(UB) &= \mathrm{var}(UB - WX_{GLS} + WX_{GLS}) \\
&= \mathrm{var}(UB - WX_{GLS}) + \mathrm{var}(WX_{GLS}) + 2\,\mathrm{cov}(UB - WX_{GLS}, WX_{GLS}) && \text{(using equation 10)} \\
&= \mathrm{var}(UB - WX_{GLS}) + \mathrm{var}(WX_{GLS}) + 2\,\mathrm{cov}(UB - VA^T\Sigma^{-1}B, VA^T\Sigma^{-1}B) \\
&= \mathrm{var}(UB - WX_{GLS}) + \mathrm{var}(WX_{GLS}) + 2\,\mathrm{cov}((U - VA^T\Sigma^{-1})B, (VA^T\Sigma^{-1})B) \\
&= \mathrm{var}(UB - WX_{GLS}) + \mathrm{var}(WX_{GLS}) + 2(U - VA^T\Sigma^{-1})\,\mathrm{var}(B)(VA^T\Sigma^{-1})^T && \text{(using remark)} \\
&= \mathrm{var}(UB - WX_{GLS}) + \mathrm{var}(WX_{GLS}) + 2(U - VA^T\Sigma^{-1})\,\mathrm{var}(AX+\varepsilon)\Sigma^{-1}AV^T && \text{(since } \Sigma \text{ is symmetric)} \\
&= \mathrm{var}(UB - WX_{GLS}) + \mathrm{var}(WX_{GLS}) + 2(U - VA^T\Sigma^{-1})\Sigma\Sigma^{-1}AV^T \\
&= \mathrm{var}(UB - WX_{GLS}) + \mathrm{var}(WX_{GLS}) + 2(UA - VA^T\Sigma^{-1}A)V^T \\
&= \mathrm{var}(UB - WX_{GLS}) + \mathrm{var}(WX_{GLS}) + 2(W - W)V^T && \text{(using equation 8 and 9)} \\
&= \mathrm{var}(UB - WX_{GLS}) + \mathrm{var}(WX_{GLS}) \\
&\geq \mathrm{var}(WX_{GLS}) && \text{(since } \mathrm{var}(UB - WX_{GLS}) \geq 0\text{)}
\end{aligned}$$

Remark :

Suppose P and Q are both K×N deterministic linear transformation, which transform B from N dimensional space to K dimensional space, then we have :

$$\begin{aligned}
\mathrm{cov}(PB, QB) &= E[(PB - E(PB))(QB - E(QB))^T] \\
&= E[P(B - E(B))(B - E(B))^T Q^T] \\
&= PE[(B - E(B))(B - E(B))^T]Q^T \\
&= P\,\mathrm{var}(B)Q^T \\
&= \begin{cases} \sigma^2 I & Uncorrelated, Homoskedastic \\ W^{-1} & Uncorrelated, Heteroskedastic \\ \Sigma & Correlated \end{cases} && \text{(W is diagonal, with } w_n = 1/\sigma_n^2\text{)}
\end{aligned}$$

Bias and variance of density estimation

Given a set of data, unbiased mean estimation and unbiased variance estimation are given by :

$$
\begin{aligned}
x_n &\sim \varepsilon(\mu,\sigma) \qquad n \in [1, N] \\
\mu_{est} &= \textstyle\sum_{n=1}^{N} x_n / N \\
\sigma_{est}^2 &= \textstyle\sum_{n=1}^{N} (x_n - \mu_{est})^2 / (N-1) \\
&= \textstyle\sum_{n=1}^{N} (x_n^2 - 2\mu_{est} x_n + \mu_{est}^2) / (N-1) \\
&= (\textstyle\sum_{n=1}^{N} x_n^2 - 2\mu_{est} \sum_{n=1}^{N} x_n + N\mu_{est}^2) / (N-1) \\
&= (\textstyle\sum_{n=1}^{N} x_n^2 - 2N\mu_{est}^2 + N\mu_{est}^2) / (N-1) \\
&= (\textstyle\sum_{n=1}^{N} x_n^2 - N\mu_{est}^2) / (N-1)
\end{aligned}
$$

Thus : $E(x_n) = \mu$

$\mathrm{var}(x_n) = \sigma^2$ i.e. $E(x_n^2) = \mathrm{var}(x_n) + E^2(x_n) = \sigma^2 + \mu^2$

$\mathrm{cov}(x_n x_m) = 0$ i.e. $E(x_n x_m) = \mathrm{cov}(x_n x_m) + E(x_n)E(x_m) = \mu^2$

$$
\begin{aligned}
\Rightarrow \quad E(\mu_{est}) &= E(\textstyle\sum_{n=1}^{N} x_n / N) \\
&= \textstyle\sum_{n=1}^{N} E(x_n) / N \\
&= \textstyle\sum_{n=1}^{N} \mu / N \\
&= \mu \qquad \text{i.e. This is unbiased estimation of mean.}
\end{aligned}
$$

$$
\begin{aligned}
\Rightarrow \quad \mathrm{var}(\mu_{est}) &= \mathrm{var}(\textstyle\sum_{n=1}^{N} x_n / N) \qquad \text{i.e. } E(\mu_{est}^2) = \mathrm{var}(\mu_{est}) + E^2(\mu_{est}) = \sigma^2/N + \mu^2 \\
&= \textstyle\sum_{n=1}^{N} \mathrm{var}(x_n) / N^2 \\
&= \textstyle\sum_{n=1}^{N} \sigma^2 / N^2 \\
&= \sigma^2 / N
\end{aligned}
$$

$$
\begin{aligned}
\Rightarrow \quad E(\sigma_{est}^2) &= E((\textstyle\sum_{n=1}^{N} x_n^2 - N\mu_{est}^2) / (N-1)) \\
&= (\textstyle\sum_{n=1}^{N} E(x_n^2) - NE(\mu_{est}^2)) / (N-1) \\
&= (\textstyle\sum_{n=1}^{N} (\mu^2 + \sigma^2) - N(\mu^2 + \sigma^2/N)) / (N-1) \\
&= (N(\mu^2 + \sigma^2) - N(\mu^2 + \sigma^2/N)) / (N-1) \\
&= (N(\sigma^2 - \sigma^2/N)) / (N-1) \\
&= \sigma^2 \qquad \text{i.e. This is unbiased estimation of variance.}
\end{aligned}
$$

Reference

(1) Gauss Markov theorem, Kenny Yu.