Chapter 1

Least Squares

1.1 Basic Statistics

1.2 Covariance Matrices, Error Ellipses, Confidence Intervals

1.3 Error Propagation (GLOPOV/SLOPOV)

1.4 Derivations of Least Squares

Least Squares regression solves an over-constrained set of equations as a minimization of the residuals squared(v_i). There are a few different least squares regression methods, which will be described in this section.

$$min \sum_{i}^{n} v_i^2$$

1.4.1 Ordinary and Weighted Least Squares

Ghalani provides a good basic example explanation of a simple ordinary least squares problem (Section 11.6.1 Ghalani). Consider the 3 equations provided below.

Equations

The system is overconstrained and there is no solution that satisfies each equation.

$$\begin{aligned} x + y &= 3.0 \\ 2x - y &= 1.5 \\ x - y &= 0.2 \end{aligned}$$

Observation Equations

In order to solve the solution, we add a *residual* for each equation. This residual captures any error in the measurements. These equations are called the observation equations, and are a foundation of least squares.

$$\begin{aligned} x + y &= 3.0 + v_1 \\ 2x - y &= 1.5 + v_2 \\ x - y &= 0.2 + v_3 \end{aligned}$$

Residual Equations

The observation equations can be rewritten as residual equations, shown below.

$$\begin{aligned} v_1 &= x + y - 3.0 \\ v_2 &= 2x - y - 1.5 \\ v_3 &= x - y - 0.2 \end{aligned}$$

The goal of least squares is to determine a solution which minimizes the sum of the square of the residuals... or to get the **least squares solution!**

$$min(\sum_{i=1}^{n} v_i^2)$$

Sum of Squares Equation

$$f(x, y) = (x + y - 3.0)^2 + (2x - y - 1.5)^2 + (x - y - 0.2)^2$$

The minimum of the sum of squares is where the derivative in the x and y dimension is equal to 0. Therefore we derive a normal equation as the partial derivative with respect to each unknown variable, in this case (x,y).

Normal Equation Derivation

$$\begin{aligned}\frac{\delta f(x,y)}{\delta x} &= 2(x+y-3.0)+2(2x-y-1.5)(2)+2(x-y-0.2)\\ \frac{\delta f(x,y)}{\delta y} &= 2(x+y-3.0)+2(2x-y-1.5)(-1)+2(x-y-0.2)(-1)\end{aligned}$$

Normal Equations

$$\begin{aligned}6x-2y-6.2&=0\\ -2x+3y-1.3&=0\end{aligned}$$

Solving for the normal equations yields the 'most probable solution'.

Solving Normal Equations Using Matrices

$$\begin{bmatrix}6 & -2\\ -2 & 3\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix}=\begin{bmatrix}6.2\\ 1.3\end{bmatrix}$$

$$\begin{bmatrix}x\\ y\end{bmatrix}=inv\left(\begin{bmatrix}6 & -2\\ -2 & 3\end{bmatrix}\right)\begin{bmatrix}6.2\\ 1.3\end{bmatrix}$$

$$\begin{bmatrix}x\\ y\end{bmatrix}=\begin{bmatrix}1.514\\ 1.443\end{bmatrix}$$

Derive Normal Equations Using Matrices

If we write our initial observation equations as matrices...

$$AX=L+V$$

$$A=\begin{bmatrix}1 & 1\\ 2 & -1\\ 1 & -1\end{bmatrix}\qquad X=\begin{bmatrix}x\\ y\end{bmatrix}\qquad L=\begin{bmatrix}3.0\\ 1.5\\ 0.2\end{bmatrix}\qquad V=\begin{bmatrix}v_1\\ v_2\\ v_3\end{bmatrix}$$

Pre-multiplying both sides of the equation by A^T, we get the same Normal Equations as above! (V goes away because the normal equation can be solved without residuals). A^TA is sometimes rewritten as N, or the Normal Matrix.

$$\begin{aligned}AX&=L+V\\ A^TAX&=A^TL\rightarrow\text{ sometimes written as }NX=A^TL\end{aligned}$$

$$\begin{bmatrix}1 & 2 & 1\\ 1 & -1 & -1\end{bmatrix}\begin{bmatrix}1 & 1\\ 2 & -1\\ 1 & -1\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix}=\begin{bmatrix}1 & 2 & 1\\ 1 & -1 & -1\end{bmatrix}\begin{bmatrix}3.0\\ 1.5\\ 0.2\end{bmatrix}$$

$$\begin{bmatrix}6 & -2\\ -2 & 3\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix}=\begin{bmatrix}6.2\\ 1.3\end{bmatrix}$$

Notice that this is the same as the normal equations derived above.

Derive Matrix Least Squares Solution (Ordinary Least Squares)

Putting the pieces together, we can now solve for the least squares solution using matrices where X represents the most probable values which are the result of a minimization of the squared residuals of each observation.

$$\begin{aligned}AX&=L+V\\ A^TAX&=A^TL\\ X&=inv(A^TA)A^TL\end{aligned}$$

Add a Weight Term (Stochastic Model)

The weight matrix, W, is used to weight the importance of each observation equation. This is used when you have higher confidence in certain observations, and want those observations to more significantly influence your result. While these weights can be related to anything, it is most common that the W matrix is equal to the variance-covariance matrix for all of the observation equations. Normally each observation is assumed to be statistically independent, which would yield a diagonal matrix (no covariances). However, in an example using GPS surveying, 3 observation equations (X,Y,Z) with covariances are created for each measurement. When Covariances are present, this is often referred to as General Least Squares. Note: Ghilani Textbook does not use this nomenclature.

An example weight matrix for 4 observations equations (a,b,c,d) is shown below.

$$W = \Sigma^{-1} = \begin{bmatrix} \sigma_{aa}^2 & \sigma_{ab}^2 & \sigma_{ac}^2 & \sigma_{ad}^2 \\ \sigma_{ab}^2 & \sigma_{bb}^2 & \sigma_{bc}^2 & \sigma_{bd}^2 \\ \sigma_{ac}^2 & \sigma_{bc}^2 & \sigma_{cc}^2 & \sigma_{cd}^2 \\ \sigma_{ad}^2 & \sigma_{bd}^2 & \sigma_{cd}^2 & \sigma_{dd}^2 \end{bmatrix}^{-1} = \begin{bmatrix} \sigma_{aa}^2 & 0 & 0 & 0 \\ 0 & \sigma_{bb}^2 & 0 & 0 \\ 0 & 0 & \sigma_{cc}^2 & 0 \\ 0 & 0 & 0 & \sigma_{dd}^2 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{\sigma_{aa}^2} & 0 & 0 & 0 \\ 0 & \frac{1}{\sigma_{bb}^2} & 0 & 0 \\ 0 & 0 & \frac{1}{\sigma_{cc}^2} & 0 \\ 0 & 0 & 0 & \frac{1}{\sigma_{dd}^2} \end{bmatrix}$$

Derive Weighted Linear Least Squares

Incorporating the Weight matrix results in the following, where A^TWA is the Normal Matrix.

$$\begin{aligned} WAX &= WL + WV \\ A^TWAX &= A^TWL \\ X &= inv(A^TWA)A^TWL \end{aligned}$$

1.4.2 Nonlinear Least Squares

Each of the previous examples only considered linear observation equations, so the values in the A matrix were constants. When this is not the case, an iterative nonlinear least squares approach must be used. Instead of writing the observation equation as $AX = L + V$ as in the linear case, we write the observation equations as $JX = K + V$. Where J is a matrix of partial derivatives of each observation with respect to each unknown. This method is based on taylor series expansion, and it requires a guess at the initial X (X_0).

The X, in this case, is sometimes written as a ΔX, as it represents how a delta from the initial guess. The delta is applied to the initial guess, and it is solved again. This iterative approach is performed until it stabilizes on a solution. The exit criteria for the loop is normally based on when the reference variance S_0^2 increases. This either means, your solution is diverging to a poor solution, or the deltas have become so small, that they are "bouncing" up and down about the solution.

Weighted Nonlinear Least Squares

Incorporating the Weight Matrix results in the following nonlinear least squares solution.

$$\begin{aligned} WJX &= WK \\ J^TWJX &= J^TWK \\ X &= inv(J^TWJ)J^TWK \end{aligned}$$

1.4.3 Ghilani: "General Least Squares" ("I think" Total Least Squares (TLS))

It is important to recognize that the weight in all of the previous examples is associated with each observation equation, and not each individual variable. For example, when solving a coordinate transformation, it is impossible to have error in both the reference and transformed coordinates. Generalized Least Squares is required to allow residuals in each of the variables.

Consider the least squares fit of a line with the observation equation to fit a linear line through a set of data points. Notice that the residual is associated with the y coordinate.

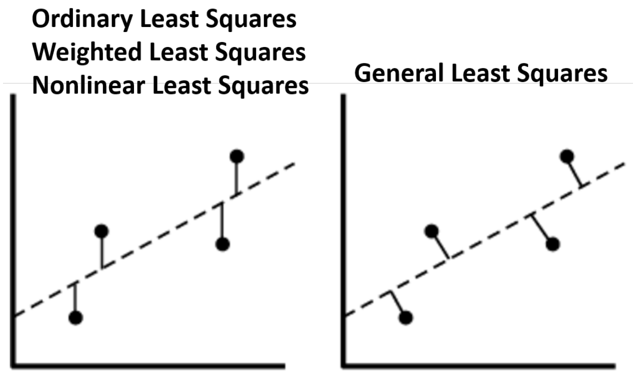
$$mx + b = y + v$$

This is what is commonly performed when you do a linear regression with most software packages. If, however, you want to have error in both the x and y coordinate, you must use general least squares. The observation equation would then be:

$$m(x + v_x) + b = (y + v_y)$$

A visual plot showing how the fit from General Least Squares is different is shown below:

better plot here



"I think" Total Least Squares Solution

Define your observation equations such that all of the observations and residuals are on the left side of the equation.

$$\begin{aligned} mx + b &= y \\ m(x + v_x) + b &= (y + v_y) \\ m(x + v_x) + b - (y + v_y) &= 0 \end{aligned}$$

Now we want to linearize the equation with respect to the unknowns and the observations.

$$F : m(x + v_x) + b - (y + v_y) = 0$$

$$\begin{aligned} \frac{\delta F}{\delta x} v_x + \frac{\delta F}{\delta y} v_y + \frac{\delta F}{\delta m} d_m + \frac{\delta F}{\delta b} d_b &= 0 - (m_0 x + b_0 - y) \\ m v_x - v_y + x d_m + d_b &= 0 - (mx + b - y) \end{aligned}$$

Now if we had 4 points (A,B,C,D), we could write the following equations.

$$\begin{aligned} m v_{x_A} - v_{y_A} + x_A d_m + d_b &= 0 - (m x_A + b - y_A) \\ m v_{x_B} - v_{y_B} + x_B d_m + d_b &= 0 - (m x_B + b - y_B) \\ m v_{x_C} - v_{y_C} + x_C d_m + d_b &= 0 - (m x_C + b - y_C) \\ m v_{x_D} - v_{y_D} + x_D d_m + d_b &= 0 - (m x_D + b - y_D) \end{aligned}$$

Which can be rewritten in Matrix form:

$$BV + JX = K$$

Where:

$$B = \begin{bmatrix} m & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & m & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & m & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & m & -1 \end{bmatrix} \qquad V = \begin{bmatrix} v_{x_A} \\ v_{y_A} \\ v_{x_B} \\ v_{y_B} \\ v_{x_C} \\ v_{y_C} \\ v_{x_D} \\ v_{x_D} \end{bmatrix}$$

$$J = \begin{bmatrix} x_A & 1 \\ x_A & 1 \\ x_A & 1 \\ x_A & 1 \end{bmatrix} \qquad X = \begin{bmatrix} dm \\ db \end{bmatrix} \qquad K = \begin{bmatrix} -(mx_A + b - y_A) \\ -(mx_B + b - y_B) \\ -(mx_C + b - y_C) \\ -(mx_D + b - y_D) \end{bmatrix}$$

With an optional Covariance Matrix for all of the observations equal to:

$$\Sigma = \begin{bmatrix} \sigma^2_{x_A x_A} & \sigma^2_{x_A y_A} & \sigma^2_{x_A x_B} & \sigma^2_{x_A y_B} & \sigma^2_{x_A x_C} & \sigma^2_{x_A y_C} & \sigma^2_{x_A x_D} & \sigma^2_{x_A y_D} \\ \sigma^2_{x_A y_A} & \sigma^2_{y_A y_A} & \sigma^2_{y_A x_B} & \sigma^2_{y_A y_B} & \sigma^2_{y_A x_C} & \sigma^2_{y_A y_C} & \sigma^2_{y_A x_D} & \sigma^2_{y_A y_D} \\ \sigma^2_{x_A x_B} & \sigma^2_{y_A x_B} & \sigma^2_{x_B x_B} & \sigma^2_{x_B y_B} & \sigma^2_{x_B x_C} & \sigma^2_{x_B y_C} & \sigma^2_{x_B x_D} & \sigma^2_{x_B y_D} \\ \sigma^2_{x_A y_B} & \sigma^2_{y_A y_B} & \sigma^2_{x_B y_B} & \sigma^2_{y_B y_B} & \sigma^2_{y_B x_C} & \sigma^2_{y_B y_C} & \sigma^2_{y_B x_D} & \sigma^2_{y_B y_D} \\ \sigma^2_{x_A x_C} & \sigma^2_{y_A x_C} & \sigma^2_{x_B x_C} & \sigma^2_{y_B x_C} & \sigma^2_{x_C x_C} & \sigma^2_{x_C y_C} & \sigma^2_{x_C x_D} & \sigma^2_{x_C y_D} \\ \sigma^2_{x_A y_C} & \sigma^2_{y_A y_C} & \sigma^2_{x_B y_C} & \sigma^2_{y_B y_C} & \sigma^2_{x_C y_C} & \sigma^2_{y_C y_C} & \sigma^2_{y_C x_D} & \sigma^2_{y_C y_D} \\ \sigma^2_{x_A x_D} & \sigma^2_{y_A x_D} & \sigma^2_{x_B x_D} & \sigma^2_{y_B x_D} & \sigma^2_{x_C x_D} & \sigma^2_{y_C x_D} & \sigma^2_{x_D x_D} & \sigma^2_{x_D y_D} \\ \sigma^2_{x_A y_D} & \sigma^2_{y_A y_D} & \sigma^2_{x_B y_D} & \sigma^2_{y_B y_D} & \sigma^2_{x_C y_D} & \sigma^2_{y_C y_D} & \sigma^2_{x_D y_D} & \sigma^2_{y_D y_D} \end{bmatrix}$$

Or with no covariances:

$$\Sigma = \begin{bmatrix} \sigma^2_{x_A} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma^2_{y_A} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma^2_{x_B} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma^2_{y_B} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma^2_{x_C} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma^2_{y_C} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma^2_{x_D} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma^2_{y_D} \end{bmatrix}$$

This system of equations is solved by finding an "equivalent solution" by using an "equivalent weight matrix".

$$W_e = inv(B\Sigma B^T)$$

The "Equivalent Weight Matrix System" is:

$$J^T W_e J X = J^T W_e K$$

$$X = inv(J^T W_e J) J^T W_e K$$

This is a nonlinear problem which must be solved iteratively. See the "General Least Squares" Section for all of the generalized equations and a full example problem.

1.5 Robust and Resistant Regression

Robust least squares attempts to minimize the weight of outliers. So instead of minimizing the square of the residuals (L2 Norm Regression aka Least Squares), one example would be to minimize the absolute value of the residuals (L1 Norm Regression or Least Absolute Difference LAD). This minimizes the impact of outliers, as their residuals are no longer squared. A more sophisticated example would use "**Iterative Reweighted Least Squares**", where each point receives a new weight based on an equation for the residuals.

Resistant Least Squares attempts to remove the outlier observations altogether in an iterative manner.

These methods are useful for datasets where there are likely outliers or blunders.

*I'm not going to go into too much detail on this right now, but this link seems like a good resource for more a more detailed description: https://onlinecourses.science.psu.edu/stat501/node/353

1.6 Variables and Definitions

1.6.1 Calculating Least Squares Parameters

When solving the least squares problem, you can also report other information about the solution. The definition of each of these terms is provided here, and the equation to calculate each term for each least squares method is provided in the summaries.

Residuals (V)

The residuals are easy to compute for ordinary least squares. It is simply the predicted value minus the observed value.

For Ordinary Least Squares:

$$\text{Residuals} = V = AX - L$$

Unit Variance (S_0^2) (*mse* in matlab)

The unit variance is also called "variance of an observation of unit weight" or the "variance of unit weight." Assuming all of the variances used in the initial weight matrix are correct(accurate stochastic model), the a priori value for the unit variance is 1. With a correct stochastic model, the computer reference variance of the solution after a least squares adjustment should also be equal to 1. If it is too high, it indicates that the initial stochastic model was overly optimistic and that the covariance matrix of the observations needed to be scaled up to characterize the observed residuals. If, the reference variance is too low, the initial stochastic model was too conservative, and the covariance matrix needed to be scaled down to characterize the observed residuals. In the case where the computed reference variance is too low, the test result is often ignored since correcting it will not change the adjusted coordinates significantly. This is basically saying you measured your points more accurately than you thought you did. If the computed reference variance is too high, this indicates that there may either be blunders in the data or you simply you didn't measure your points as accurately as you thought you did. To test if the computed reference variance is statistically different than the a priori reference variance (1), use the goodness of fit test.

For ordinary least squares is:

$$\text{Reference Variance} = S_0^2 = \frac{V'V}{dof}$$

χ^2 Goodness of Fit Test (is S_0^2 statistically equal to 1?)

A wo tailed χ^2 Goodness of Fit Test is used to determine if there is an issue with your least squares adjustment at an α significance level. α values are normally low (eg. 0.01, 0.05).

$$\begin{aligned}\text{Null Hypothesis } H_0 &: S_0^2 = 1\\ \text{Alternative Hypothesis } H_a &: S_0^2 \neq 1\\ \text{Significance } &: \alpha\end{aligned}$$

Test Statistic:

$$\chi^2 = \frac{vS_0^2}{\sigma^2} = \frac{dof \times S_0^2}{1} = dof \times S_0^2$$

Rejection Region:

$$\chi^2 > \chi^2_{(\alpha/2,dof)}$$
$$\chi^2 < \chi^2_{(1-\alpha/2,dof)}$$

If either of the rejection region tests fail at the alpha level, then the null hypothesis is rejected and we can say that the computed reference variance is not equal to the a priori reference variance of 1 at an α significance level. **More importantly**, and somewhat counter-intuitively, is that if both rejection region tests pass, we say that the computed reference variance is not significantly different than the a priori reference variance at an *alpha* significance level, so **we set the computed reference variance equal to 1!** What we are saying here is that the computed reference variance is just an estimate, and it should be equal to 1. Since that estimate is not statistically different than 1 (which would indicate a blunder or poor stochastic model), we assume that there are no blunders and our stochastic model is correct. If there are no blunders and our stochastic model is correct, the computed reference variance is equal to 1.

Cofactor Matrix (Q)

The Cofactor Matrix is the inverse of the Normal Matrix. When the reference variance is equal to 1, it is the same as the covariance matrix.

need more info here

Covariance Matrix or Unknowns (Σ_x)

The Covariance Matrix (also Variance-Covariance) describes the variance and covariances between the computed unknown parameters. The variance is the diagonal element, and the off diagonal elements represent how the computed unknown parameters are related. The standard deviation of the unknowns is just the square root of the variances along the diagonal.

Covariance Matrix of Observations (Σ_l)

The covariance matrix describes the variance and covariance of each of your observation equations. So for example, if you have observation equations for distances between a few points, and solve for the coordinates of the unknown points. You can then use this to determine the uncertainty in the distances. This is basically using Σ_x and GLOPOV.

Model Skill (R^2, sometimes $\hat{S}$)

The Skill of the model, or R^2, is the amount of variance explained by the model. A R^2 equal to 1 indicates the model perfectly fits the data. A R^2 equal to 0 indicates that the model does not explain any of the variance of the data. **Do not use R^2 for nonlinear least squares**, as the value is not valid.

Extra Sum of Squares Test (Large N approximation)

The model skill will always increase as the number of unknowns increases, so the significant of adding another unknown to a model can be tested with a 'extra sum of squares test.' This test requires that a large number of N observations. The equation below shows the hypothesis test that is performed. to compare if the extra parameter added in model B (with M_2 parameters) is significant compared to model A(with M_1 parameters) at the α confidence level. If the test is true, the extra parameter is not significant.

$$\frac{R_B^2 - R_A^2}{1 - R_B^2} \leq \frac{1}{N}\chi^2_{(\alpha, M_2 - M_1)}$$

RMSE (Root Mean Square Error)

RMSE is another metric to summarize the residuals of the model. A value of 0 indicates a perfect fit. Higher values indicate higher residuals and a less accurate model.

1.7 Ordinary Least Squares (OLS)

1.7.1 Theory

Ordinary Least Squares solves a linear, overconstrained system of equations by minimizing the squared residuals of each observation equation.

$$min \sum_{i}^{n} v_i^2$$

1.7.2 Assumptions

- No Outliers/Blunders OLS is not robust to outliers (consider RANSAC/Robust Weighting if outliers)
- System of equations is linear (eg. derivative wrt each unknown is not a function of any of the unknowns)
- System is over-contrained (eg. Number of Observation Equations > Number of Unknowns)
- Error only in dependent variable (eg. mx+b = y + v → error only in y dimension)

1.7.3 Equations

$$AX = L + V$$

$$m = \text{number of observations} \qquad n = \text{number of unknowns}$$

$$dof = \text{degrees of freedom (\# of redundant observations)} = m - n$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{1n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \qquad X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad L = \begin{bmatrix} l_1 \\ l_2 \\ \vdots \\ l_m \end{bmatrix} \qquad V = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix}$$

$$\begin{aligned}
\text{Unknowns} &= \hat{X} = inv(A^T A)A^T L \\
\text{Residuals} &= V = AX - L \\
\text{Reference Variance} &= S_0^2 = \frac{V'V}{dof} \\
\text{Cofactor Matrix} &= Q_{xx} = inv(A^T A) \\
\text{Covariance Matrix of Unkowns} &= \Sigma_{xx} = S_0^2 Q_{xx} \\
\text{Covariance Matrix of Observations} &= \Sigma_{\hat{l}\hat{l}} = A\Sigma_{xx}A^T \\
\text{Standard Deviation of Solved Unknowns} &= \sigma_{\hat{X}} = \sqrt{diag(\Sigma_{xx})} \\
\text{Predicted L} &= \hat{L} = AX \\
\text{R}^2 \text{ (model skill)} &= \frac{var(\hat{L})}{var(L)} \\
\text{RMSE } &= \sqrt{\frac{VV^T}{m}}
\end{aligned}$$

1.7.4 Sample Problem

Given the Points:

$$x = [0, 1, 2, 3, 4] \qquad y = [0, 1, 7, 13, 24]$$

Fit a parabola given the observation equation:

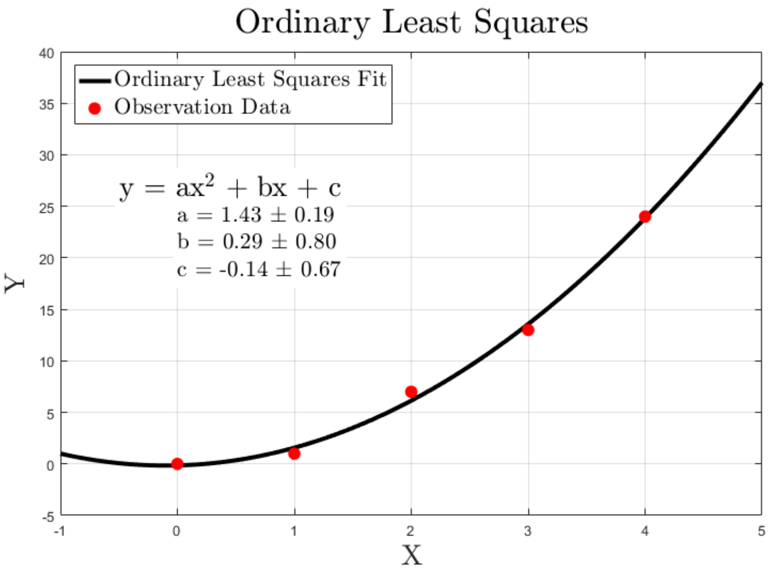
$$y = ax^2 + bx + c$$

Note that the observation equation is linear. The x^2 term is a constant once observation values are substituted.

$$A = \begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \\ x_4^2 & x_4 & 1 \\ x_5^2 & x_5 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 4 & 2 & 1 \\ 9 & 3 & 1 \\ 16 & 4 & 1 \end{bmatrix} \qquad X = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \qquad L = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 7 \\ 13 \\ 24 \end{bmatrix}$$

Use the Equations and solve:

$dof = 2$	$\hat{X} = \begin{bmatrix} 1.43 \\ 0.29 \\ -0.14 \end{bmatrix}$	$V = \begin{bmatrix} -0.14 \\ 0.57 \\ -0.86 \\ 0.57 \\ -0.14 \end{bmatrix}$
$Q = \begin{bmatrix} 0.07 & -0.29 & 0.14 \\ -0.29 & 1.24 & -0.77 \\ 0.14 & -0.77 & 0.89 \end{bmatrix}$	$S_0^2 = 0.71$	$\hat{L} = \begin{bmatrix} -0.14 \\ 1.57 \\ 6.14 \\ 13.57 \\ 23.86 \end{bmatrix}$
$\Sigma_{xx} = \begin{bmatrix} 0.05 & -0.20 & 0.10 \\ -0.20 & 0.89 & -0.55 \\ 0.10 & -0.55 & 0.63 \end{bmatrix}$	$\sigma_{\hat{X}}^2 = \begin{bmatrix} 0.05 \\ 0.89 \\ 0.63 \end{bmatrix}$	$R^2 = 0.9963$
$\Sigma_{\hat{l}\hat{l}} = \begin{bmatrix} 0.05 & -0.20 & 0.10 \\ -0.20 & 0.89 & -0.55 \\ 0.10 & -0.55 & 0.63 \end{bmatrix}$		



1.7.5 Example Matlab Code

check this code (so2 squared when it shouldn't)

show code using matlab lscov

Algorithm 1.1: exampleOLS.m

```
%% Example Ordinary Least Squares
%% observations
x = [0 1 2 3 4];
y = [0 1 7 13 24];
%% Define A and L based on observation equation
A = [x(:).^2 x(:) ones(size(x(:)))];
L = y(:);
%% Calculate Ordinary Least Squares
% X = inv(A'*A)*A'*L; %Replaced 'inv(A)*b' with 'A\b' per Matlab docs
X = (A'*A)\A'*L;                % unknowns
V = A * X - L;                  % residuals
dof = numel(L) - size(A,2);     % degrees of freedom
So2 = V'*V/dof;                 % Reference Variance
Q = inv(A'*A);                  % cofactor
Sx = So2.^2 * Q;                % covariance
Lhat = A * X;                   % predicted L values
r2 = var(Lhat)/var(L);          % R^2 Skill
```

Algorithm 1.2: exampleOLS.m (plotting Code)

```
%% Plot Results
f = figure(1);clf
xx = -1:0.01:5;
yy = X(1)*xx.^2+X(2)*xx+X(3);
plot(xx,yy,'k','linewidth',3);
hold on
plot(x,y,'r.','markersize',30);
grid on
xlabel('X','fontsize',20,'interpreter','latex')
ylabel('Y','fontsize',20,'interpreter','latex')
title('Ordinary Least Squares','fontsize',24,'interpreter','latex')
h = text(-0.5,25,'y = ax$^2$ + bx + c','interpreter','latex');
h.HorizontalAlignment = 'left';
h.FontSize = 20;
h.VerticalAlignment = 'bottom';
h.BackgroundColor = 'w';
astr = sprintf('a = %.2f $\\pm$ %.2f',X(1),sqrt(Sx(1,1)));
bstr = sprintf('b = %.2f $\\pm$ %.2f',X(2),sqrt(Sx(2,2)));
cstr = sprintf('c = %.2f $\\pm$ %.2f',X(3),sqrt(Sx(3,3)));
h = text(0,25,{astr,bstr,cstr},'interpreter','latex');
h.FontSize = 16;
h.BackgroundColor = 'w';
h.VerticalAlignment = 'top';
hl=legend({'Ordinary Least Squares Fit','Observation Data'},'fontsize',16);
hl.Interpreter = 'latex';
hl.Location = 'northwest';
saveas(f,'OLSexample.png');
```

1.8 Weighted Least Squares (WLS)

1.8.1 Theory

Weighted Least Squares solves a linear, overconstrained system of equations by minimizing the squared residuals of each observation equation, with a weight (W_i) applied to each observation.

$$min \sum_{i}^{n} W_i \times v_i^2$$

*Note that if the scale of the variance in the weight matrix is known to be 1, then the computed reference variance should be inspected to ensure it passes the χ^2 goodness of fit test. If it passes the test, the Covariance matrix should NOT be multiplied by the reference variance. See definition of reference variance for the reasoning.

1.8.2 Assumptions

- No Outliers/Blunders WLS is not robust to outliers (consider RANSAC/Robust Weighting if outliers)
- System of equations is linear (eg. derivative wrt each unknown is not a function of any of the unknowns)
- System is over-contrained (eg. Number of Observation Equations > Number of Unknowns)
- Error only in dependent variable (eg. mx+b = y + v → error only in y dimension)
- No covariance between weight of each observation

1.8.3 Equations

$$WAX = WL + WV$$

$$m = \text{number of observations} \qquad n = \text{number of unknowns}$$

$$dof = \text{degrees of freedom (\# of redundant observations)} = m - n$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{1n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \quad X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad L = \begin{bmatrix} l_1 \\ l_2 \\ \vdots \\ l_m \end{bmatrix} \quad V = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix} \quad W = diag\left(\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix} \text{ or } \begin{bmatrix} \sigma_1^2 \\ \sigma_2^2 \\ \vdots \\ \sigma_m^2 \end{bmatrix}^{-1} \right)$$

$$\text{Unknowns} = \hat{X} = inv(A^T WA)A^T WL$$

$$\text{Residuals} = V = AX - L$$

$$\text{Reference Variance} = S_0^2 = \frac{V'WV}{dof}$$

$$\text{Cofactor Matrix} = Q_{xx} = inv(A^T WA)$$

$$\text{Covariance Matrix of Unkowns} = \Sigma_{xx} = S_0^2 \times Q_{xx}$$

$$\text{Covariance Matrix of Observations} = \Sigma_{\hat{l}\hat{l}} = A\Sigma_{xx}A^T$$

$$\text{Standard Deviation of Solved Unknowns} = \sigma_{\hat{X}} = \sqrt{diag(\Sigma_{xx})}$$

$$\text{Predicted L} = \hat{L} = AX$$

$$\text{R}^2 \text{ (model skill)} = \frac{var(\hat{L})}{var(L)}$$

$$\text{RMSE } = \sqrt{\frac{VV^T}{m}}$$

1.8.4 Sample Problem

1.8.5 Example Matlab Code

1.9 General Least Squares (GLS)

1.9.1 Theory

General Least Squares solves a linear, overconstrained system of equations by minimizing the squared residuals of each observation equation, with a weight matrix defining the variance AND covariance of each observation equation. Note that the equations are the same as WLS, except the Weight Matrix (W) contains covariances.

*Note that if the scale of the variance-covariance in the weight matrix is known to be 1, then the computed reference variance should be inspected to ensure it passes the χ^2 goodness of fit test. If it passes the test, the Covariance matrix should NOT be multiplied by the reference variance. See definition of reference variance for the reasoning.

1.9.2 Assumptions

- No Outliers/Blunders WLS is not robust to outliers (consider RANSAC/Robust Weighting if outliers)
- System of equations is linear (eg. derivative wrt each unknown is not a function of any of the unknowns)
- System is over-constrained (eg. Number of Observation Equations > Number of Unknowns)
- Error only in dependent variable (eg. mx+b = y + v → error only in y dimension)
- Covariance between weight of each observation

1.9.3 Equations

$$WAX = WL + WV$$

$$m = \text{number of observations} \qquad n = \text{number of unknowns}$$

$$dof = \text{degrees of freedom (\# of redundant observations)} = m - n$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \quad X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad L = \begin{bmatrix} l_1 \\ l_2 \\ \vdots \\ l_m \end{bmatrix} \quad V = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix} \quad W = \begin{bmatrix} \sigma_{11}^2 & \sigma_{12}^2 & \dots & \sigma_{1m}^2 \\ \sigma_{21}^2 & \sigma_{22}^2 & \dots & \sigma_{2m}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{m1}^2 & \sigma_{m2}^2 & \dots & \sigma_{mm}^2 \end{bmatrix}^{-1}$$

$$\begin{aligned}
\text{Unknowns} &= \hat{X} = inv(A^T W A) A^T W L \\
\text{Residuals} &= V = AX - L \\
\text{Reference Variance} &= S_0^2 = \frac{V'WV}{dof} \\
\text{Cofactor Matrix} &= Q_{xx} = inv(A^T W A) \\
\text{Covariance Matrix of Unkowns} &= \Sigma_{xx} = S_0^2 \times Q_{xx} \\
\text{Covariance Matrix of Observations} &= \Sigma_{\hat{l}\hat{l}} = A \Sigma_{xx} A^T \\
\text{Standard Deviation of Solved Unknowns} &= \sigma_{\hat{X}} = \sqrt{diag(\Sigma_{xx})} \\
\text{Predicted L} &= \hat{L} = AX \\
\text{R}^2 \text{ (model skill)} &= \frac{var(\hat{L})}{var(L)} \\
\text{RMSE } &= \sqrt{\frac{VV^T}{m}}
\end{aligned}$$

1.9.4 Sample Problem

1.9.5 Example Matlab Code

1.10 Nonlinear Least Squares

1.10.1 Theory

Nonlinear Least Squares solves a nonlinear, overconstrained system of equations by minimizing the squared residuals of each observation equation. A Taylor Series expansion is utilized to linearize the equation and iteratively calculate the gradient of the function to determine a local minima. Optionally, a weight matrix(W) may be used. Remove the W term, or replace with an identity matrix if a Weight matrix is not desired. Covariances may be 0s if not known.

*Note that if the scale of the variance-covariance in the weight matrix is known to be 1, then the computed reference variance should be inspected to ensure it passes the χ^2 goodness of fit test. If it passes the test, the Covariance matrix should NOT be multiplied by the reference variance. See definition of reference variance for the reasoning.

1.10.2 Assumptions

- No Outliers/Blunders. Nonlinear Least Squares is not robust to outliers (consider RANSAC/Robust Weighting if outliers)
- System of equations is nonlinear (eg. derivative wrt at least one unknown is a function of one of the other unknowns)
- System is over-constrained (eg. Number of Observation Equations > Number of Unknowns)
- Error only in dependent variable (eg. mx+b = y + v → error only in y dimension)
- X_0 must be a reasonable guess, otherwise the solution might settle on an incorrect local minima, rather than the global minimum.

1.10.3 Equations

$$AX = L + V$$

$$WJ\Delta X = WK + WV$$

$$m = \text{number of observations} \qquad n = \text{number of unknowns}$$

$$dof = \text{degrees of freedom (\# of redundant observations)} = m - n$$

$$i = \text{loop iteration}$$

$$\text{Observation Equations} = F_m(x_1, x_2, ..., x_n) = l_m \text{ (note: } AX \text{ is obs eqn, L is } [l_1, l_2 ..., l_m] \text{)}$$

$$J = \begin{bmatrix} \frac{\delta F_1}{\delta x_1} & \frac{\delta F_1}{\delta x_2} & \cdots & \frac{\delta F_1}{\delta x_n} \\ \frac{\delta F_2}{\delta x_1} & \frac{\delta F_2}{\delta x_2} & \cdots & \frac{\delta F_2}{\delta x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\delta F_m}{\delta x_1} & \frac{\delta F_m}{\delta x_2} & \cdots & \frac{\delta F_m}{\delta x_n} \end{bmatrix} \quad \Delta X = \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_n \end{bmatrix} \quad K = \begin{bmatrix} l_1 - F_1(x_1, x_2, ..., x_n) \\ l_2 - F_2(x_1, x_2, ..., x_n) \\ \vdots \\ l_m - F_m(x_1, x_2, ..., x_n) \end{bmatrix} \quad V = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix}$$

$$W = \begin{bmatrix} \sigma_{11}^2 & \sigma_{12}^2 & \cdots & \sigma_{1m}^2 \\ \sigma_{21}^2 & \sigma_{22}^2 & \cdots & \sigma_{2m}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{m1}^2 & \sigma_{m2}^2 & \cdots & \sigma_{mm}^2 \end{bmatrix}^{-1} \qquad \text{Initial Guess } X_0 = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Loop Equations:

Loop until ΔX is small, or more robustly, loop until S_0^2 increases. S_0^2 will increase slightly when you get down to really really small numbers and the cpu starts rounding. Caveat: it will also increase if you have a really bad initial guess, and it starts diverging.

$$\begin{aligned}
\text{Loop Delta Estimate} &= \Delta X = inv(J^T W J) J^T W K \\
\text{Loop Estimate} &= \hat{X}_i = X_{i-1} + \Delta X \\
\text{Residuals} &= V = J\hat{X}_i - K \\
\text{Reference Variance} &= S_0^2 = \frac{V^T W V}{dof}
\end{aligned}$$

Final Calculations

$$\begin{aligned}
\text{Unknowns} &= \hat{X} = \hat{X}_i \text{ (Final Loop Estimate)} \\
\text{Cofactor Matrix} &= Q_{xx} = inv(J^T W J) \\
\text{Covariance Matrix of Unkowns} &= \Sigma_{xx} = S_0^2 \times Q_{xx} \\
\text{Covariance Matrix of Observations} &= \Sigma_{\hat{l}\hat{l}} = J \Sigma_{xx} J^T \\
\text{Standard Deviation of Solved Unknowns} &= \sigma_{\hat{X}} = \sqrt{diag(\Sigma_{xx})} \\
\text{Predicted L} &= \hat{L} = AX \\
\text{R}^2 \text{ (model skill)} &= \frac{var(\hat{L})}{var(L)} \\
\text{RMSE } &= \sqrt{\frac{VV^T}{m}}
\end{aligned}$$

1.10.4 Sample Problem

1.10.5 Example Matlab Code

1.11 Total Least Squares (TLS)

1.11.1 Theory

Total Least Squares solves a nonlinear, overconstrained system of equations by minimizing the squared residuals of each observation. A Taylor Series expansion is utilized to linearize the equation and iteratively calculate the gradient of the function to determine a local minima. A Weight Matrix, with a column/row for each observation rather than observation equation, is used to allow for error in all dimensions. Traditionally, this will be a predicted variance for each observation.

*Note that if the scale of the variance-covariance in the weight matrix is known to be 1, then the computed reference variance should be inspected to ensure it passes the χ^2 goodness of fit test. If it passes the test, the Covariance matrix should NOT be multiplied by the reference variance. See definition of reference variance for the reasoning.

1.11.2 Assumptions

- No Outliers/Blunders. Nonlinear Least Squares is not robust to outliers (consider RANSAC/Robust Weighting if outliers)
- System is over-constrained (eg. Number of Observation Equations > Number of Unknowns)
- Error is in all measurements variable (eg. mx+b = y + v → error only in x and y dimension)
- X_0 must be a reasonable guess, otherwise the solution might settle on an incorrect local minima, rather than the global minimum. If a linear problem, one method is to solve the unweighted OLS, then use that to initialize X_0 in TLS.

1.11.3 Equations

$$AX = L \qquad \text{(note: residuals in both X and L)}$$

$$BV + J\Delta X = K$$

$$m = \text{number of observations} \qquad n = \text{number of unknowns}$$

$$dof = \text{degrees of freedom (\# of redundant observations)} = m - n$$

$$i = \text{loop iteration}$$

$$\text{Observation Equations} = F_m(x_1, x_2, ..., x_n) = l_m \text{ (note: } AX \text{ is obs eqn, L is } [l_1, l_2 ..., l_m] \text{)}$$

$$B = \begin{bmatrix} \frac{\delta F_1}{\delta v_1} & \frac{\delta F_1}{\delta v_2} & \dots & \frac{\delta F_1}{\delta v_n} & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\delta F_2}{\delta v_1} & \frac{\delta F_2}{\delta v_2} & \dots & \frac{\delta F_2}{\delta v_n} & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & \frac{\delta F_m}{\delta v_1} & \frac{\delta F_m}{\delta v_2} & \dots & \frac{\delta F_m}{\delta v_n} \end{bmatrix} \qquad V = \begin{bmatrix} v_{11} \\ v_{21} \\ \vdots \\ v_{n1} \\ v_{12} \\ v_{22} \\ \vdots \\ v_{n2} \\ \vdots \\ v_{1m} \\ v_{2m} \\ \vdots \\ v_{nm} \end{bmatrix}$$

$$J = \begin{bmatrix} \frac{\delta F_1}{\delta x_1} & \frac{\delta F_1}{\delta x_2} & \cdots & \frac{\delta F_1}{\delta x_n} \\ \frac{\delta F_2}{\delta x_1} & \frac{\delta F_2}{\delta x_2} & \cdots & \frac{\delta F_2}{\delta x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\delta F_m}{\delta x_1} & \frac{\delta F_m}{\delta x_2} & \cdots & \frac{\delta F_m}{\delta x_n} \end{bmatrix} \qquad \Delta X = \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_n \end{bmatrix} \qquad K = \begin{bmatrix} l_1 - F_1(x_1, x_2, ..., x_n) \\ l_2 - F_2(x_1, x_2, ..., x_n) \\ \vdots \\ l_m - F_m(x_1, x_2, ..., x_n) \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} \sigma_{11}^2 & \sigma_{12}^2 & \cdots & \sigma_{1(n\times m)}^2 \\ \sigma_{21}^2 & \sigma_{22}^2 & \cdots & \sigma_{2(n\times m)}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{(n\times m)1}^2 & \sigma_{(n\times m)2}^2 & \cdots & \sigma_{(n\times m)(n\times m)}^2 \end{bmatrix} \qquad \text{Initial Guess } X_0 = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Loop Equations:

Loop until ΔX is small, or more robustly, loop until S_0^2 increases. S_0^2 will increase slightly when you get down to really really small numbers and the cpu starts rounding. Caveat: it will also increase if you have a really bad initial guess, and it starts diverging.

$$\begin{aligned} \text{Equivalent Weight} = W_{eq} &= inv(B\Sigma B^T) \\ \text{Loop Delta Estimate} = \Delta X &= inv(J^T W_{eq} J) J^T W_{eq} K \\ \text{Loop Estimate} = \hat{X}_i &= X_{i-1} + \Delta X \\ \text{Equivalent Residuals} = V_{eq} &= J\hat{X}_i - K \\ \text{Observation Residuals} = V &= \Sigma B^T W_{eq} V_{eq} \\ \text{Reference Variance} = S_0^2 &= \frac{V_{eq}^T W_{eq} V_{eq}}{dof} \end{aligned}$$

Final Calculations

$$\begin{aligned} \text{Unknowns} = \hat{X} &= \hat{X}_i \text{ (Final Loop Estimate)} \\ \text{Cofactor Matrix} = Q_{xx} &= inv(J^T W_{eq} J) \\ \text{Covariance Matrix of Unkowns} = \Sigma_{xx} &= S_0^2 \times Q_{xx} \\ \text{Covariance Matrix of Observations} = \Sigma_{\hat{l}\hat{l}} &= J\Sigma_{xx} J^T \\ \text{Standard Deviation of Solved Unknowns} = \sigma_{\hat{X}} &= \sqrt{diag(\Sigma_{xx})} \\ \text{RMSE } &= \sqrt{\frac{V_{eq} V_{eq}^T}{m}} \end{aligned}$$

1.11.4 Sample Problem

1.11.5 Example Matlab Code

1.12 Example: TPU linear line fit

$$y = mx + b$$

TPU GLOPOV vs SLOPOV comparison of uncertainty from fit

1.13 Example: RANSAC Plane Fit

1.14 Example: 3D Conformal Transformation with TLS

1.15 Example: Space Resection

1.16 Example: TPU Ortho From Space Resection

1.17 Example: Space Intersection

1.18 Example: SfM

1.19 References