# PARAMETER ESTIMATION

The fundamental concept of parameter estimation is to determine optimal values of parameters for a numerical model that predicts dependent variable outputs of a function, process or phenomenon based on observations of independent variable inputs. For a given data observation, independent variable inputs are grouped into an input vector of length *ninp* and dependent variable outputs are grouped into an output vector of length *nout*. Corresponding input and output vectors for a given data observation are called a training pair. In general, training pairs from *nobs* number of data observations are pooled to estimate *npar* number of model parameters.

# Table of Contents

# 1.0 Least Squares Methods

##  1.1 Linear Models

 1.1.1 Straight Line

 1.1.2 Single Dependent Variable

 1.1.2.1 Polynomial

 1.1.2.1.1 Quadratic

 1.1.2.1.2 Cubic

 1.1.2.1.3 Quintic

 1.1.2.1.4 Choice Of Polynomial Order

 1.1.2.2 Harmonic

 1.1.2.3 Plane

 1.1.2.4 Bicubic

 1.1.3 Multiple Dependent Variables

 1.1.4 Weighting

 1.1.5 Scaling

#  1.2 Linearized Models

 1.2.1 Newtonian Cooling

 1.2.2 Circle

 1.2.3 Sphere

 1.3 Example Data

 1.3.1 Anscombe's Data Set

# 2.0 Eigenvector Methods

#  2.1 Line

#  2.2 Plane

 2.3 Ellipse

 2.4 Ellipsoid

 2.5 2D Quadric

 2.6 3D Quadric

3.0 Levenberg-Marquardt

4.0 Simplex

5.0 Gradient Search

6.0 Neural Network

7.0 Genetic Algorithms

8.0 Simulated Annealing

9.0 Kalman Filter

# 1.0 LEAST SQUARES METHODS

The least squares optimality criterion minimizes the sum of squares of residuals between actual observed outputs and output values of the numerical model that are predicted from input observations

## 1.1 LINEAR MODELS

Many processes exhibit true linear behavior. Many others operate over such small excursions of input variable values that the output behavior appears linear.

**1.1.1 STRAIGHT LINE**

The classic least squares problem is to fit a straight line model with a single input x and a single output y using parameters b and m as shown in Equation 1. Unfortunately, individual data observations xi and yi may not fit the model perfectly due to experimental measurement error, process variation or insufficient model complexity as shown in Equation 2. Multiple data observations may be concatenated as shown in Equation 3 and represented in matrix form per Equation 4. Even for optimal estimates of model parameters {}, each data observation will have some residual error ei between the observed output yi and the predicted model output as shown in Equations 5 and 6 .

y = b + mx = [ 1 x ] for *ninp*=1, *nout*=1, *npa*r=2 Eq. 1

yi ≈ [ 1 xi ] Eq. 2

 Eq. 3

{Y} ≈ [X] {} for    Eq. 4

 Eq. 5

{e} = {Y} - [X] {} for  Eq. 6

The scalar sum of squares SSQ of residual errors is shown in Equation 7. To minimize the sum of squares, one may set the partial derivative of the sum of squares with respect to the model parameters  equal to zero as shown in Equation 8. Rearranging these terms provides a linear matrix solution for optimal model parameters per Equation 9a. The coefficient of determination  is provided in Equation 9b.

SSQ = {e}T{e} = {Y}T{Y} - 2{}T[X]T{Y} + {}T[X] T[X] {} Eq. 7

∂SSQ / ∂{} = - 2[X]T{Y} + 2[X] T[X] {} = 0 Eq. 8

 Eq. 9a

 for  Eq. 9b

Expanding this matrix solution as shown in Equation 10, provides valuable insights into the nature of least squares estimates. If the input observations xi and the output observations yi are both mean centered as shown in Equation 11, then the offset term b is zero and the slope m is equal to the covariance of xi and yi divided by the variance of xi as shown in Equation 12. Consequently, mean centered data preclude the need to compute an offset term for many linear models.

 Eq. 10

 for xo = = 0 and yo = = 0 Eq. 11

 for xo = 0, yo = 0 Eq. 12

r2 ????

**1.1.2 SINGLE DEPENDENT VARIABLE**

The methodology developed above may be generalized to polynomials or other linear models requiring *npar ≥ 2* number of linear parameters as shown in Equation 13. Note that the number of data observations *nobs* must be greater than or equal to the number of estimated parameters *npar* to prevent singularity of the symmetric matrix ([X]T[X]).

 for *nout*=1 Eq. 13

**1.1.2.1 POLYNOMIAL**

**1.1.2.1.1 QUADRATIC**

A quadratic model is presented in Equations 14 and 15.

y = a0 + a1 x + a2 x2 = [ 1 x x2 ] for *ninp*=1, *nout*=1, *npa*r=3 Eq. 14

   Eq. 15

**1.1.2.1.2 CUBIC**

A cubic model is presented in Equations 16 and 17.

y = a0 + a1 x + a2 x2 + a3 x3 = [ 1 x x2 x3 ] for *ninp*=1, *nout*=1, *npa*r=4 Eq. 16

   Eq. 17

**1.1.2.1.3 QUINTIC**

A quintic model is presented in Equations 18 and 19.

y = [ 1 x x2 x3 x4 ] for *ninp*=1, *nout*=1, *npa*r=5 Eq. 18

   Eq. 19

**1.1.2.1.4 CHOICE OF POLYNOMIAL ORDER**

|  |  |
| --- | --- |
| Engine speed[rpm] | Engine torque[N.m] |
| 800 | 500 |
| 1000 | 547 |
| 1200 | 636 |
| 1400 | 679 |
| 1600 | 719 |
| 1800 | 724 |
| 2000 | 712 |
| 2200 | 671 |
| 2400 | 606 |
| 2600 | 575 |

**1.1.2.2 HARMONIC**

A linear harmonic model is presented in Equations 17 and 18.

y = [ 1 cos sin cos2 sin2 ] for *ninp*=1, *nout*=1, *npa*r=5 Eq. 17

   Eq. 18

**1.1.2.3 PLANE**

A linear planar model is presented in Equations 19 and 20.

z = a + b x + c y = [ 1 x y ] for *ninp*=2, *nout*=1, *npa*r=3 Eq. 19

   Eq. 20

**1.1.2.3 BICUBIC**

A linear bicubic model is presented in Equations 21 and 22.

 for *ninp*=2, *nout*=1, *npa*r=16 Eq. 21

 Eq. 22a

 Eq. 22b



  Eq. 22c

**1.1.3 MULTIPLE DEPENDENT VARIABLES**

Multiple simultaneous linear outputs for each input data observation may be modeled using this methodology as shown in Equation 23a. Note that the number of data observations *nobs* must be greater than or equal to the number of parameters *npar* (not *npar* times *nout*) to prevent singularity of the symmetric matrix ([X]T[X]). The coefficient of determination  is provided in Equation 23b.

 Eq. 23a

 for  Eq. 23b

pre-inverse or post-inverse ???

A classic example of multiple linear outputs is to compute affine alignment (translation, rotation, size, shear) between two images based on pixel locations of homologues as shown in Eq. 23c and 23d.

 = pixel location of homologue in target image

 = pixel location of matching homologue in image to be aligned with target

 Eq. 23c

   npar = 3 Eq. 23d

**1.1.4 WEIGHTING**

A weighting factor wi may be assigned to each training pair to emphasize relative importance among the data observations. Residuals {e} may be accentuated or attenuated by pre-multiplying with a diagonal weighting matrix [W] to form weighted residuals {} as shown in Equation 23a. The least squares solution for this new set of weighted residuals is shown Equation 23b.

 Eq. 23a

 Eq. 23b

Two concepts are typically used for weights. The simplest form is to use a value of 1 if that data observation is present in the current data set while a value of 0 is used if that observation is missing as shown in Equation 23c. This technique is often used for scientific devices that employ the same data collection protocols for each data set (same *nobs* each time), but in which certain observations may be missing or spurious (e.g. photogrammetry with landmarks that are occasionally not visible).

wi2 = 1 for valid data, wi2 = 0 for missing data Eq. 23c

The second approach is to set the square of each weight equal to the inverse of the expected variance for that observation. The expected variance may be the variance of the respective dependent variable output, the variance of the independent variable input or a pooled value representative of both.

wi2 = 1 / i2 for i2 = variance of observation i Eq. 23d

**1.1.5 SCALING**

# mean center xi\* = xi - x0

# scale largest abs(xi\*) to ±1

# 1.2 LINEARIZED MODELS

It is often tempting to linearize models of nonlinear phenomena in an effort to arrive at estimates for parameters. One should exercise caution however in that linear least squares solutions for linearized models minimize the sum of squares of residuals for nonlinear forms of the dependent variables rather than residuals of those dependent variables themselves.

**1.2.1 NEWTONIAN COOLING**

A classic example is to model the temperature T of an object as a function of time  during exponential Newtonian cooling from an initial temperature T0 toward ambient temperature T∞ as shown in Equation 24. Taking the logarithm after a simple algebraic rearrangement provides the linearized model in Equation 25. If the ambient temperature T∞ is known, measuring temperatures Ti of the object at times i allows direct estimation of the Newtonian cooling rate b and indirect estimation of the initial temperature difference above ambient with the linearized matrix model in Equations 26. Specifically, this approach minimizes residuals of the nonlinear function ln(T - T∞) rather than residuals of the dependent variable temperature T. Note that simultaneous estimation of ambient temperature T∞ requires an iterative algorithm.

T = T∞ + (T0 - T∞) e-b Eq. 24

ln(T - T∞) = ln(T0 - T∞) - b  Eq. 25

   Eq. 26

**1.2.2 LOGARITHMIC SPIRAL**

The logarithmic spiral often appears in nature such as the cross –section of a nautilus shell, atmospheric low pressure spirals or the arms of galaxies as shown in Equations 27. Given center location xc and yc allows linearization of x and y data points on the spiral as shown in Equations 28 and provides the linearized matrix model of Equations 29. Again, this approach minimizes residuals of the nonlinear function ln(r) rather than residuals of the independent variables x and y. Simultaneous estimation of center location xc and yc requires iterative techniques. Note that there are no actual dependent variables for closed form geometric models estimated from coordinate data.

r = a eb x = xc + r cos y = yc + r cos Eq. 27

 Eq. 28

   Eq. 29

**1.2.3 CIRCLE**

Another example is to estimate the coordinates a and b for the center of a circle of unknown radius r by measuring the coordinates of points x and y on the circle as shown in the familiar nonlinear model in Equation 30. Expanding and rearranging provides the linearized model shown in Equation 31. Measuring coordinates xi and yi for points on the circle allows direct estimation of the center coordinates a and b and indirect estimation of the radius with the linearized model in Equations 32. This approach minimizes residuals of the nonlinear function (x2+y2) rather than residuals for the independent variables x and y.

( x - a )2 + ( y - b )2 = r2 Eq. 30

( x2 + y2 ) = [ 1 x y ] Eq. 31

   Eq. 32

If radius r is known a priori, the problem may be reformulated to find the best fit for center coordinates a and b as shown in Equations 33. The solution for the term (-a2-b2) may not be completely consistent with the solution for terms (2a) and (2b) if the data does not exactly model the given radius r. Again, note that there are no actual dependent variables for closed form geometric models estimated from coordinate data.

   Eq. 33

**1.2.4 SPHERE**

Following the circle example above, one can estimate the coordinates a, b and c for the center of a sphere of unknown radius r as shown in Equations 34.

   Eq. 34

**1.3 EXAMPLE DATA**

**1.3.1 ANSCOMBE'S DATA SET**

# Ref: Anscombe, F.J. (1973) Graphs in statistical analysis. Amer. Statistician 27:17-21

#

# Anscombe's datasets

# x1=[10.00 8.00 13.00 9.00 11.00 14.00 6.00 4.00 12.00 7.00 5.00 ]';

# y1=[ 8.04 6.95 7.58 8.81 8.33 9.96 7.24 4.26 10.84 4.82 5.68 ]';

# x2=x1;

# y2=[ 9.14 8.14 8.74 8.77 9.26 8.10 6.13 3.10 9.13 7.26 4.74 ]';

# x3=x1;

# y3=[ 7.46 6.77 12.74 7.11 7.81 8.84 6.08 5.39 8.15 6.42 5.73 ]';

# x4=[ 8.00 8.00 8.00 8.00 8.00 8.00 8.00 19.00 8.00 8.00 8.00 ]';

# y4=[ 6.58 5.76 7.71 8.84 8.47 7.04 5.25 12.50 5.56 7.91 6.89 ]';2.0 EIGENVECTOR METHODS

The linear methods described above minimize the sum of squares of residuals between observed outputs and output values predicted from multiple input observations. However for some numerical models, the concepts of independent inputs and dependent outputs are not well defined. Fitting geometric models to *k* dimensional point coordinate data is an example. The coordinates for each point are typically grouped into an input vector of length *ninp=k* and there are no dependent variable outputs. In general, input vectors from *nobs* number of data observations are pooled to estimate *npar* number of model parameters.

Linear least squares methods minimize the sum of squares of residuals parallel to the dependent variable. For eigenvector methods, the residuals are nominally orthogonal to model.

Orthogonal distance fitting …

# 2.1 LINE

The parametric equation for a line in *k* dimensional space describes the location of any point {x} on the line in terms of a given point {p} on the line plus a directed scalar distance s measured from {p} in the unit direction  along the line as shown in Equation 32. However, individual data observations {xi} for points representing the line may not fit the model perfectly due to experimental measurement error, process variation or insufficient model complexity as shown in Equation 33.

 Eq. 32

{xi} ≈ {p} + s  Eq. 33

The optimal least-squares estimate for the given point {p} on the line will be the centroid {xo} of the *nobs* data observations {xi} as shown Equation 34. The symmetric centroidal point distribution matrix [S] may then be formed as shown in Equation 35. Its diagonal eigenvalue matrix [D] and orthogonal eigenvector matrix [V] shown Equation 36 provide least-squares solutions.

{p} = {xo} = {xi} Eq. 34

=({xi}-{xo})T ({xi}-{xo}) Eq. 35

[S] = [V] [D] [V]T for [V] = [{v1}{v2}…{vk}] and [D] = diag( d1 d2 … dk ) Eq. 36

Perfect data observations of points on a line would produce a single non-zero eigenvalue of [S] along a direction parallel to the line. Consequently for real-world data observations, the optimal least-squares estimate for the unit direction  will be parallel to the eigenvector that corresponds to the largest eigenvalue as shown in Equation 37. The largest eigenvalue will be equal to variance of the data points about their centroid along the line. The other eigenvalues are principal values of variance perpendicular to the line.

= {vj} / norm{vj} for {vj} corresponding to largest dj for a line Eq. 37

# 2.2 PLANE

The equation for a plane in *k* dimensional space describes the location of any point {x} on the plane based on the perpendicular distance  from the coordinate origin to the plane as determined by a given point {p} on the plane and the unit normal to plane as shown in Equation 38. However individual data observations {xi} for points representing the plane may not fit the model perfectly due to experimental measurement error, process variation or insufficient model complexity as shown in Equation 39.

{x}T  = {p}T  =  Eq. 38

{xi}T  ≈ {p}T  =  Eq. 39

The optimal least-squares estimate for the given point {p} on the plane will be the centroid {xo} of the *nobs* data observations {xi} as shown Equation 34 above. The symmetric centroidal point distribution matrix [S], diagonal eigenvalue matrix [D] and orthogonal eigenvector matrix [V] shown in Equations 35 and 36 above again provide least-squares solutions per below.

Perfect data observations of points on a plane would produce a single zero eigenvalue of [S] in a direction normal to the plane. Consequently for real-world data observations, the optimal least-squares estimate for the unit normal  will be parallel to the eigenvector that corresponds to the smallest eigenvalue as shown in Equation 40. The smallest eigenvalue will be equal to the variance of the data points perpendicular to the plane. The other eigenvalues are principal values of variance within the plane.

= {vj} / norm{vj} for {vj} corresponding to smallest dj for a plane Eq. 40

**2.3 ELLIPSE**

* 1. **ELLIPSOID**
	2. **2D QUADRIC**

**2.6 3D QUADRIC**

**3.0 LEVENBERG-MARQUARDT**

The Levenberg-Marquardt algorithm iteratively adjusts estimates of model parameters {} to minimize residuals between measured dependent variable outputs {y} and predictions from a numerical model f(.) based on independent variable inputs {x} as shown in Equation 44. For a given set of model parameters {}k at iteration *k* each measured training pair {x}i and {y}i will have residuals {e}i,k as shown in Equation 45. For parameter updates {} shown in Equation 46, the Taylor series expansion for residuals at iteration *k+1* may be written as shown in Equation 47 using the Jacobian [J] of the numerical model with respect to model parameters in Equation 48.

 Eq. 44

 Eq. 45

{}k+1 = {}k + {} Eq. 46

 Eq. 47

 Eq. 48

If only one training pair is available and the number of dependent variable outputs is equal to the number of parameters (*nobs*=1 and *nout*=*npar*), Equation 47 is deterministic and one can try to drive all *nout* residuals {e}i,k+1 to zero using Equation 49. This provides the classical Newton-Raphson root finding algorithm shown in Equation 50.

 Eq. 49

 Eq. 50

If only one training pair is available and the number of dependent variable outputs is larger than the number of parameters (*nobs*=1 and *nout*>*npar*), residuals {e}i,k+1 at iteration *k+1* can be minimized by the standard linear least squares solution shown in Equation 51.

 Eq. 51

However, if the number of parameters is greater than the number of dependent variable outputs, Equation 51 is row insufficient and multiple training pairs are required (*nobs*>1 for *nout*<*npar*). Residuals from all training pairs at iteration *k* shown in Equation 45 may be concatenated as shown in Equation 52 providing an aggregate sum of squares SSQ over all observations. Similarly all residuals predicted at iteration *k+1* for update {} in Equation 47 may be concatenated as shown Equation 53. The linear least squares solution for parameter updates that will minimize the predicted aggregate SSQ after at iteration *k+1* is then shown in Equations 54 and 55.

 Eq. 52

 Eq. 53

 Eq. 54

 Eq. 55

Equation 55 provides rapid second order Newtonian convergence but can become unstable if the square Jacobian summation is nearly singular. Levenberg and Marquardt showed that a positive factor  added to the diagonal elements of the square Jacobian summation matrix as shown in Equation 56 can provide both rapid and stable convergence. For very small values of , this provides Newtonian convergence similar to Equation 55. For larger values of , this provides small but stable steps along the gradient shown in Equation 57.

 Eq. 56

 Eq. 57

If parameter updates provide a stable step with smaller aggregate SSQ than prior iterations, factor  is reduced in preparation for the next iteration. If parameter updates provide an unstable step with larger aggregate SSQ than prior iterations, those updates are rejected, factor  is increased and the process is repeated. Typically  is started at a value of 0.1, is reduced by a factor of 10 for stable steps, and is increased by a factor of 10 for unstable steps.

Convergence may be assessed by observing when absolute values of parameter updates are small while the aggregate SSQ approaches the expected standard deviation of residuals. Observing the

progression of factor  can also help indicate convergence.

The algorithm may be summarized as follows.

1) Postulate initial estimates for parameters {}

2) Evaluate aggregate SSQ over all training pairs for initial parameter estimates (Equation 52)

3) Set factor  = 0.1

4) Proceed through all training pairs

 a) Evaluate all residuals {e}i,k (Equation 45)

 b) Evaluate all Jacobians [J]i.k (Equation 48)

 c) Accumulate summations  (Equation 55)

5) Add factor  to diagonal and compute parameter updates {} (Equation 56)

6) Update parameters {}k+1 (Equation 46)

7) Evaluate aggregate SSQ over all training pairs for new parameter estimates (Equation 52)

8) If aggregate SSQ has been reduced:

 a) Reduce factor NEW = OLD / 10

 b) Proceed with the next iteration at step 4)

9) If aggregate SSQ has increased:

 a) Discard the new parameter estimates and use immediate prior values

 b) Increasee factor NEW = OLD \* 10

 c) Proceed with the next iteration at step 5)

Because of its robust performance, Levenberg-Marquardt method is often used with finite difference numerical approximations for the Jacobian [J] of the numerical model with respect to model parameters. Note that this Jacobian must be re-evaluated for each training pair for each iteration whether analytically or numerically.

Penalty functions may be added to residuals to impose explicit or implicit inequality constraints on parameters. However as with any gradient technique, convergence may dither across constraint boundaries if the minimum SSQ is at a constraint boundary.

Levenberg, K. "A Method for the Solution of Certain Problems in Least Squares." *Quart. Appl. Math.* **2**, 164-168, 1944.

Marquardt, D. "An Algorithm for Least-Squares Estimation of Nonlinear Parameters." *SIAM J. Appl. Math.* **11**, 431-441, 1963.

**4.0 SIMPLEX**

Nedler, J.A. and Mead, R. "A Simplex Method for Function Minimization." Computer Journal, **7**, 308-???, 1965.

**5.0 GRADIENT SEARCH**

**6.0 NEURAL NETWORK**

**7.0 GENETIC ALGORITHMS**

**8.0 SIMULATED ANNEALING**

**9.0 KALMAN FILTER**