1 Ordinary least-squares problem

1.1 Introduction

An *ordinary least-squares problem* (also called *linear least-squares problem*) is the problem of minimization of the (squared) norm of a vector (hence the name, least squares) where the minimization parameters enter the vector linearly. Reformulation of a given problem in terms of an ordinary least-squares problem is often called the least-squares method. Ordinary least-squares problems can be usually solved using linear algebra methods – a great asset that ensures their extensive applications in science end engineering.

In signal processing least-squares methods are used for smoothing, prediction, deconvolution, error recovery, and de-clipping. In physics it is used, in particular, in fitting a theoretical model to experimental data with uncertainties, and in deconvolution of the detected signal from the detector response.

1.2 Overdetermined linear systems

A system of linear equations is considered *overdetermined* if there are more equations than unknown variables. If all equations of an overdetermined system are linearly independent, the system has no exact solution. However, it is usually possible to find an approximate solution to an overdetermined system using the least-squares method.

Consider a linear system

$$\mathrm{A}\mathbf{c} = \mathbf{b} \,, \tag{1}$$

where A is a tall $n \times m$ matrix, $\mathbf{c}$ is an m-component vector of unknown variables and $\mathbf{b}$ is an n-component vector of the right-hand side terms. If the number of equations n is larger than the number of unknowns m, the system is overdetermined and generally has no solution. However, it is still possible to find an approximate solution — the one where $\mathrm{A}\mathbf{c}$ is only approximately equal $\mathbf{b}$ — in the sense that the Euclidean norm of the difference between $\mathrm{A}\mathbf{c}$ and $\mathbf{b}$ is minimized,

$$\mathbf{c} : \min_{\mathbf{c}} \|\mathrm{A}\mathbf{c} - \mathbf{b}\|^2 \,. \tag{2}$$

The problem (2) is an example of an ordinary least-squares problem. The vector $\mathbf{c}$ that minimizes $\|\mathrm{A}\mathbf{c} - \mathbf{b}\|^2$ is usually called the *least-squares solution*.

Theoretically, the solution to this minimization problem is given by the equation

$$\frac{\partial}{\partial \mathbf{c}^\mathsf{T}} \Big((\mathbf{c}^\mathsf{T}\mathrm{A}^\mathsf{T} - \mathbf{b}^\mathsf{T})(\mathrm{A}\mathbf{c} - \mathbf{b}) \Big) = 2\Big((\mathrm{A}^\mathsf{T}\mathrm{A})\mathbf{c} - \mathrm{A}^\mathsf{T}\mathbf{b} \Big) = 0 \,, \tag{3}$$

with the solution

$$\mathbf{c} = (\mathrm{A}^\mathsf{T}\mathrm{A})^{-1}\mathrm{A}^\mathsf{T}\mathbf{b} \,, \tag{4}$$

where $(\mathrm{A}^\mathsf{T}\mathrm{A})^{-1}\mathrm{A}^\mathsf{T}$ is the left pseudo-inverse of the tall matrix A,

$$(\mathrm{A}^\mathsf{T}\mathrm{A})^{-1}\mathrm{A}^\mathsf{T}\mathrm{A} = \mathbf{1} \,. \tag{5}$$

In practice however instead of calculating the pseudo-inverse one should use QR or SVD decompositions as described below.

1.2.1 Least-squares solution via QR-decomposition

The linear least-squares problem can be solved by QR-decomposition. The matrix A is factorized as $\mathrm{A} = \mathrm{QR}$, where Q is $n \times m$ matrix with orthonormal columns, $\mathrm{Q}^\mathsf{T}\mathrm{Q} = 1$, and R is an $m \times m$ upper triangular matrix. The matrix Q is a semi-orthonormal matrix whose columns span the range (column space) of matrix A.

The matrix QQ^T is the projector on the range of matrix A and $(1 - \mathrm{QQ}^\mathsf{T})$ is the corresponding orthogonal projector. Indeed,

$$(1 - \mathrm{QQ}^\mathsf{T})\mathrm{QQ}^\mathsf{T} = \mathrm{QQ}^\mathsf{T}(1 - \mathrm{QQ}^\mathsf{T}) = 0 \,. \tag{6}$$

The vector $\mathrm{A}\mathbf{c} - \mathbf{b}$ can be represented as a sum of two orthogonal components, the one within the range of the matrix and the orthogonal one,

$$\mathrm{A}\mathbf{c} - \mathbf{b} = \mathrm{QQ}^\mathsf{T}(\mathrm{A}\mathbf{c} - \mathbf{b}) + (1 - \mathrm{QQ}^\mathsf{T})(\mathrm{A}\mathbf{c} - \mathbf{b}) \,. \tag{7}$$

The Euclidean norm $\|\mathrm{A}\mathbf{c} - \mathbf{b}\|^2$ is then given as the sum of the norms of the two orthogonal components,

$$\|\mathrm{A}\mathbf{c} - \mathbf{b}\|^2 = \|\mathrm{QQ}^\mathsf{T}(\mathrm{A}\mathbf{c} - \mathbf{b})\|^2 + \|(1 - \mathrm{QQ}^\mathsf{T})(\mathrm{A}\mathbf{c} - \mathbf{b})\|^2 \,. \tag{8}$$

The second term,

$$\|(1 - \mathrm{QQ}^\mathsf{T})(\mathrm{A}\mathbf{c} - \mathbf{b})\|^2 = \|(1 - \mathrm{QQ}^\mathsf{T})\mathbf{b}\|^2 \,, \tag{9}$$

is the norm of the range-orthogonal component of the right-hand-side $\mathbf{b}$: it is independent of the variables $\mathbf{c}$ and can not be reduced by their variations. However, the first term

$$\|\mathrm{QQ}^\mathsf{T}(\mathrm{A}\mathbf{c} - \mathbf{b})\|^2 = \|(\mathrm{R}\mathbf{c} - \mathrm{Q}^\mathsf{T}\mathbf{b})\|^2 \,, \tag{10}$$

can be reduced down to zero by solving the $m \times m$ system of linear equations

$$\mathrm{R}\mathbf{c} = \mathrm{Q}^\mathsf{T}\mathbf{b} \,. \tag{11}$$

The system is right-triangular and can be readily solved by back-substitution.

Thus the solution to the ordinary least-squares problem (2) is given by the solution of the triangular system (11).

1.2.2 Least-squares solution via SVD

Under the *thin singular value decomposition* we shall understand a representation of a tall $n \times m$ $(n > m)$ matrix A in the form

$$\mathrm{A} = \mathrm{USV}^\mathsf{T} \,, \tag{12}$$

where U is a semi-orthonormal ($U^T U = 1$) $n \times m$ matrix that spans the range of matrix A, S is a square $m \times m$ diagonal matrix with non-negative real numbers (called singular values) on the diagonal, and V is a square $m \times m$ orthonormal matrix ($V^T V = VV^T = 1$).

Singular value decomposition can too be used to solve the least squares problem $A\mathbf{c} \approx \mathbf{b}$ the same way as with QR-decomposition only using the U matrix for the projection of the equation on the column-space of the matrix. Specifically,

$$U^T(A\mathbf{c} - \mathbf{b}) = 0 \;\Rightarrow\; SV^T\mathbf{c} = U^T\mathbf{b} \;\Rightarrow\; \mathbf{c} = VS^{-1}U^T\mathbf{b} \,, \tag{13}$$

where one should not build the S^{-1} matrix but should rather divide by the corresponding singular values the components of the vector it acts upon.

Notice that $VS^{-1}U^T$ is the pseudo-inverse A^- of the matrix A. Therefore the solution to the least squares problem can be (theoretically) written as

$$\mathbf{c} = A^-\mathbf{b} \,. \tag{14}$$

1.3 Applications

1.3.1 Signal smoothing

Least squares method can be use to smooth a noisy signal – a sequence of numbers represented by a vector, $\mathbf{y}$. The idea is to find a new signal, $\mathbf{x}$, that is similar to the noisy signal $\mathbf{y}$ but smoother. The smoothness of the signal can be measured by the smallness of its second derivative (indeed a signal with vanishing second derivative is a straight line – the smoothest possible signal). The smoothers based on minimization of the second derivative are the most common in the literature although other derivatives can also be used.

The second derivative of a discrete signal $\mathbf{x}$ can be approximated by its second-order difference, $D\mathbf{x}$, where the matrix D is given as (pay attention to the first and the last rows)

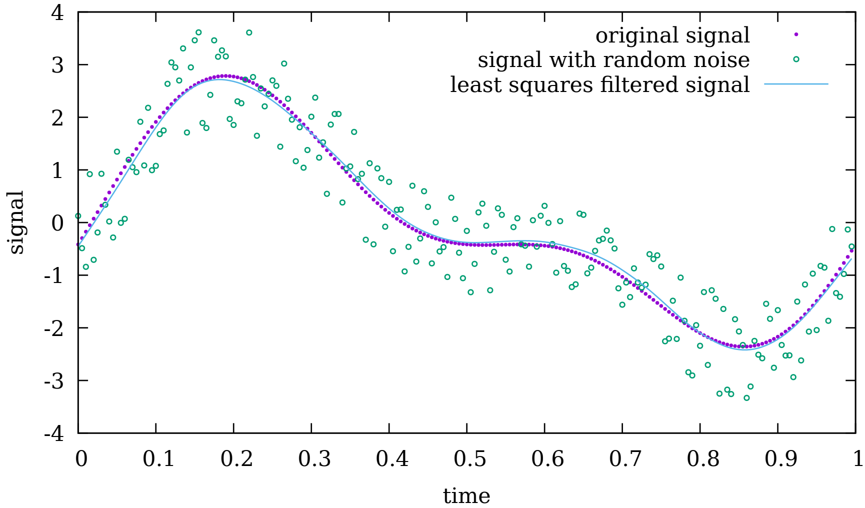
$$D = \begin{pmatrix} 1 & -2 & 1 & & & & \\ 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & & \vdots & & & \\ & & & & 1 & -2 & 1 \\ & & & & 1 & -2 & 1 \end{pmatrix} . \tag{15}$$

The smooth signal $\mathbf{x}$ can then be obtained as a solution to the following least-squares problem,

$$\mathbf{x} : \min_{\mathbf{x}} \left(\|\mathbf{x} - \mathbf{y}\|^2 + \lambda \|D\mathbf{x}\|^2 \right) \,, \tag{16}$$

where $\lambda > 0$ is the smoothing parameter: when $\lambda \to 0$ there is no smoothing and $\mathbf{x}$ converges to $\mathbf{y}$; when $\lambda \to \infty$ the second derivative is infinitely penalized and $\mathbf{x}$ converges to a linear fit to the data.

Figure 1: Noisy signal smoothing using the least-squares algorithm (16).



The minimum of the form (16) is located where its partial derivatives vanish,

$$\frac{\partial}{\partial \mathbf{x}^{\mathsf{T}}}\Big((\mathbf{x}-\mathbf{y})^{\mathsf{T}}(\mathbf{x}-\mathbf{y})+\lambda\mathbf{x}^{\mathsf{T}}\mathrm{D}^{\mathsf{T}}\mathrm{D}\mathbf{x}\Big)=2\Big((\mathrm{I}+\lambda\mathrm{D}^{\mathsf{T}}\mathrm{D})\mathbf{x}-\mathbf{y}\Big)=0\,, \tag{17}$$

(where I is the identity matrix) which gives the (least-squares) solution

$$\mathbf{x}=(\mathrm{I}+\lambda\mathrm{D}^{\mathsf{T}}\mathrm{D})^{-1}\mathbf{y}\,. \tag{18}$$

Of course in practice one should not explicitly calculate the inverse matrix but rather solve the linear equation

$$(\mathrm{I}+\lambda\mathrm{D}^{\mathsf{T}}\mathrm{D})\mathbf{x}=\mathbf{y}\,. \tag{19}$$

The matrix in this linear equation is *banded* (has only few non-zero diagonals) therefore in order to make the method efficient one has to take advantage of this fact (rather than use generic linear solvers).

And example of smoothing using this algorithm is given at Figure (1).

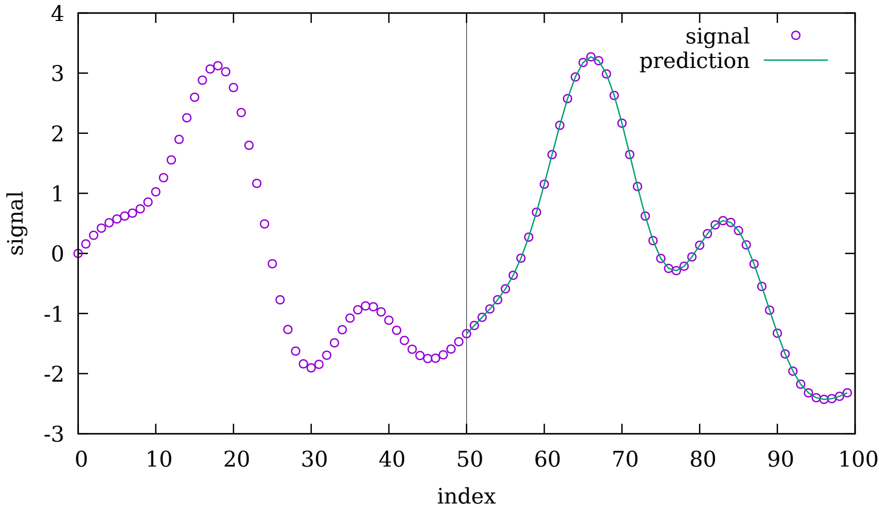
1.3.2 Signal extrapolation (prediction)

Least-squares method can also be used to find patterns (correlations) in a given signal (sequence of numbers) and then use these patterns for signal extrapolation.

Suppose the signal $\{x_i\}_{i=1\dots N}$ is correlated such that the term x_k can be predicted by the n preceding terms,

$$x_k = x_{k-1}a_n + x_{k-2}a_{n-1} + \cdots + x_{k-n}a_1\,. \tag{20}$$

Figure 2: Signal extrapolation (prediction) using the linear prediction anzats (20) with $n = 6$.



Applying this anzats (often called *linear prediction*) to the subset $\{x_{n+1}, x_{n+2}, \dots, x_N\}$ of the signal gives the system of equations

$$\begin{pmatrix} x_1 & x_2 & \dots & x_n \\ x_2 & x_3 & \dots & x_{n+1} \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ x_{N-n} & x_{N-n+1} & \dots & x_{N-1} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} x_{n+1} \\ x_{n+2} \\ \vdots \\ \vdots \\ x_N \end{pmatrix} . \quad (21)$$

This is an overdetermined system the least-squares solution of which determines the correlation parameters $\{a_1, \dots, a_n\}$ which allow extrapolation of the sequence beyond the last term, x_N,

$$x_{N+1} = a_1 x_{N-n+1} + a_2 x_{N-n+2} + \dots + a_n x_N \quad (22)$$

$$x_{N+2} = a_1 x_{N-n+2} + a_2 x_{N-n+3} + \dots + a_n x_{N+1} \quad (23)$$

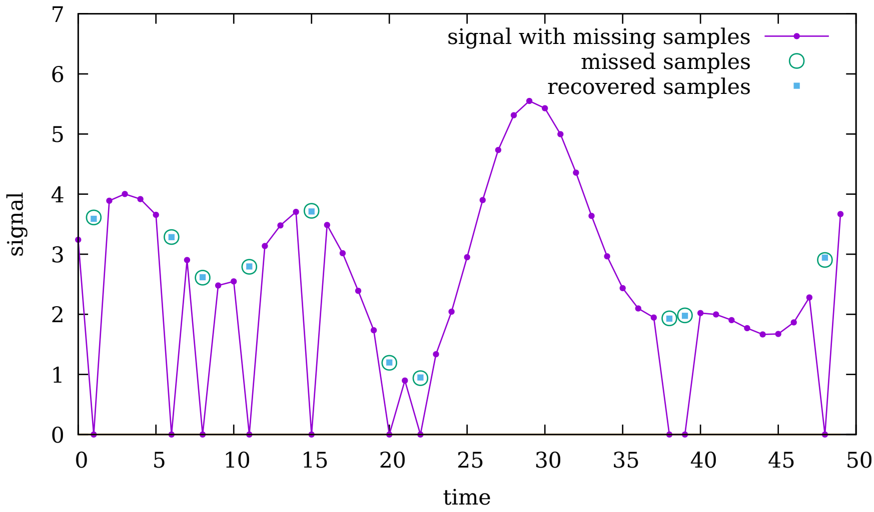
$$\vdots \quad . \quad (24)$$

An illustration of this method is given at figure (2) where the six correlation parameters where determined from the first 50 values of the signal and then extrapolated to the next 50 values (and compared with the exact values). The signal is

$$x_k = 2\sin\left(0.9\frac{2\pi k}{N-1}\right) - \sin\left(2.1\frac{2\pi k}{N-1}\right) + \frac{1}{2}\sin\left(3.1\frac{2\pi k}{N-1}\right) , \quad (25)$$

where $N = 50$.

Figure 3: Least-squares missing data recovery



1.3.3 Missing data recovery (error concealment)

Sometimes one can have a signal at hand where several entries are lost due to some sort of data corruption, for example transmission errors (packet loss). Among the techniques to recover the missing data in a corrupted signal (often referred to as *error concealment* techniques) there is one based on the least-squares method.

Let us assume that in a received signal, $\mathbf{y}$, n entries (out of the total N) at positions $m_1, m_2, \dots, m_n$ have been lost and have been replaced with zeros. If one knew the missing entries, $\{z_1 \dots z_n\} \doteq \mathbf{z}$, the recovered signal $\mathbf{x}$ would be given as

$$\mathbf{x} = \mathbf{y} + \mathrm{M}\mathbf{z} \,, \tag{26}$$

where M is the matrix that inserts the elements of $\mathbf{z}$ into the missing positions of $\mathbf{y}$. It is an $N \times n$ matrix with all elements equal zero except for the elements

$$\mathrm{M}_{m_k,k}\Big|_{k=1,\dots,n} = 1 \,. \tag{27}$$

Like in signal smoothing the unknown entries $\mathbf{z}$ can be estimated from the condition that the recovered signal is smooth,

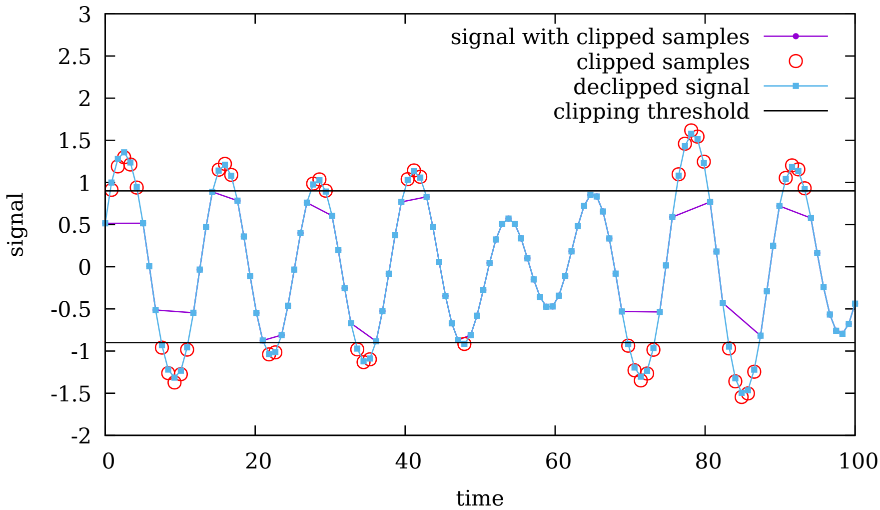
$$\mathbf{z} : \min_{\mathbf{z}} \|\mathrm{D}\mathbf{x}\|^2 = \min_{\mathbf{z}} \|\mathrm{D}(\mathbf{y} + \mathrm{M}\mathbf{z})\|^2 \,. \tag{28}$$

The solution is given by the least-squares solution to the overdetermined system

$$\mathrm{DM}\mathbf{z} = -\mathrm{D}\mathbf{y} \,. \tag{29}$$

Figure (3) illustrates the method.

Figure 4: Least-squares data declipping



1.3.4 Signal declipping

A problem very similar to missing data recovery is *signal declipping*. Here the corrupted signal misses those samples, the absolute values of which are larger than certain cutoff (for example, the detector threshold). The recovery is similar to missing data, however in this particular case one minimizes the third derivative, rather than second, forcing the declipped signal to be in the form of a parabola. The corresponding matrix is given by the third order finite difference formulae,

$$\mathrm{D} = \begin{pmatrix} -1 & 3 & -3 & 1 & & & \\ & -1 & 3 & -3 & 1 & & \\ -1/2 & 1 & 0 & -1 & 1/2 & & \\ & -1/2 & 1 & 0 & -1 & 1/2 & \\ & & & \vdots & & & \\ & & -1/2 & 1 & 0 & -1 & 1/2 \\ & & -1 & 3 & -3 & 1 & \\ & & & -1 & 3 & -3 & 1 \end{pmatrix} . \tag{30}$$

Figure (4) shows an example of least-squares declipping.

1.3.5 Ordinary least-squares curve fitting

Ordinary least-squares curve fitting is a problem of fitting n (experimental) data points $\{x_i, y_i \pm \Delta y_i\}_{i=1,\dots,n}$, where Δy_i are experimental errors, by a linear combination, $F_{\mathbf{c}}$, of m functions $\{f_k(x)\}_{k=1,\dots,m}$,

$$F_{\mathbf{c}}(x) = \sum_{k=1}^{m} c_k f_k(x) \;, \tag{31}$$

where the coefficients c_k are the fitting parameters.

The objective of the least-squares fit is to minimize the square deviation, called χ^2, between the fitting function $F_{\mathbf{c}}(x)$ and the experimental data [**?**],

$$\chi^2 = \sum_{i=1}^{n} \left(\frac{F(x_i) - y_i}{\Delta y_i} \right)^2 \;. \tag{32}$$

where the individual deviations from experimental points are weighted with their inverse errors in order to promote contributions from the more precise measurements.

Minimization of χ^2 with respect to the coefficient c_k in (31) is apparently equivalent to the least-squares problem (2) where

$$A_{ik} = \frac{f_k(x_i)}{\Delta y_i} \;,\; b_i = \frac{y_i}{\Delta y_i} \;. \tag{33}$$

Variances and correlations of fitting parameters Suppose δy_i is a small deviation of the measured value of the physical observable at hand from its exact value. The corresponding deviation δc_k of the fitting coefficient is then given as

$$\delta c_k = \sum_i \frac{\partial c_k}{\partial y_i} \delta y_i \;. \tag{34}$$

In a good experiment the deviations δy_i are statistically independent and distributed normally with the standard deviations Δy_i. The deviations (34) are then also distributed normally with *variances*

$$\langle \delta c_k \delta c_k \rangle = \sum_i \left(\frac{\partial c_k}{\partial y_i} \Delta y_i \right)^2 = \sum_i \left(\frac{\partial c_k}{\partial b_i} \right)^2 \;. \tag{35}$$

The standard errors in the fitting coefficients are then given as the square roots of variances,

$$\Delta c_k = \sqrt{\langle \delta c_k \delta c_k \rangle} = \sqrt{\sum_i \left(\frac{\partial c_k}{\partial b_i} \right)^2} \;. \tag{36}$$

The variances are diagonal elements of the *covariance matrix*, Σ, made of *covariances*,

$$\Sigma_{kq} \equiv \langle \delta c_k \delta c_q \rangle = \sum_i \frac{\partial c_k}{\partial b_i} \frac{\partial c_q}{\partial b_i} \;. \tag{37}$$

```
static (vector,matrix) lsfit
(Func<double,double>[] fs, vector x, vector y, vector dy){
   int n = x.size, m=fs.Length;
   var A = new matrix(n,m);
   var b = new vector(n);
   for(int i=0;i<n;i++){
      b[i]=y[i]/dy[i];
      for(int k=0;k<m;k++)A[i,k]=fs[k](x[i])/dy[i];
      }
   vector c = A.solve(b); // solves ||A*c-b||->min
   matrix AI = A.inverse(); // calculates pseudoinverse
   matrix Σ = AI*AI.T;
   return (c, Σ);
   }
```

Table 1: A Csharp implemetation of the ordinary least-squares fit.

Covariances $\langle \delta c_k \delta c_q \rangle$ are measures of to what extent the coefficients c_k and c_q change together if the measured values y_i are varied. The normalized covariances,

$$\frac{\langle \delta c_k \delta c_q \rangle}{\sqrt{\langle \delta c_k \delta c_k \rangle \langle \delta c_q \delta c_q \rangle}} \tag{38}$$

are called *correlations*.

Using $\mathbf{c} = \mathrm{A}^{-}\mathbf{b}$ the covariance matrix can be calculated as

$$\Sigma = \left(\frac{\partial \mathbf{c}}{\partial \mathbf{b}}\right)\left(\frac{\partial \mathbf{c}}{\partial \mathbf{b}}\right)^{\mathsf{T}} = A^{-}A^{-\mathsf{T}} = (\mathrm{A}^{\mathsf{T}}\mathrm{A})^{-1} \,. \tag{39}$$

The square roots of the diagonal elements of this matrix provide the estimates of the errors $\mathbf{\Delta c}$ of the fitting coefficients,

$$\Delta c_k = \sqrt{\Sigma_{kk}}\,\Big|_{k=1\ldots m} \,, \tag{40}$$

and the (normalized) off-diagonal elements provide the estimates of their correlations.

With SVD the covariance matrix (39) can be calculated as

$$\Sigma = (\mathrm{A}^{\mathsf{T}}\mathrm{A})^{-1} = (\mathrm{V}\mathrm{S}^2\mathrm{V}^{\mathsf{T}})^{-1} = \mathrm{V}\mathrm{S}^{-2}\mathrm{V}^{T} \,. \tag{41}$$

With QR-decomposition the covariance matrix (39) can be calculated as

$$\Sigma = (\mathrm{A}^{\mathsf{T}}\mathrm{A})^{-1} = (\mathrm{R}^{\mathsf{T}}\mathrm{R})^{-1} = \mathrm{R}^{-1}(\mathrm{R}^{-1})^{\mathsf{T}} \,. \tag{42}$$

Table 1.3.5 shows how a Csharp implementation of the ordinary least squares fit via QR decomposition could look like. An illustration of a fit is shown on Figure 5 where a polynomial is fitted to a set of data.

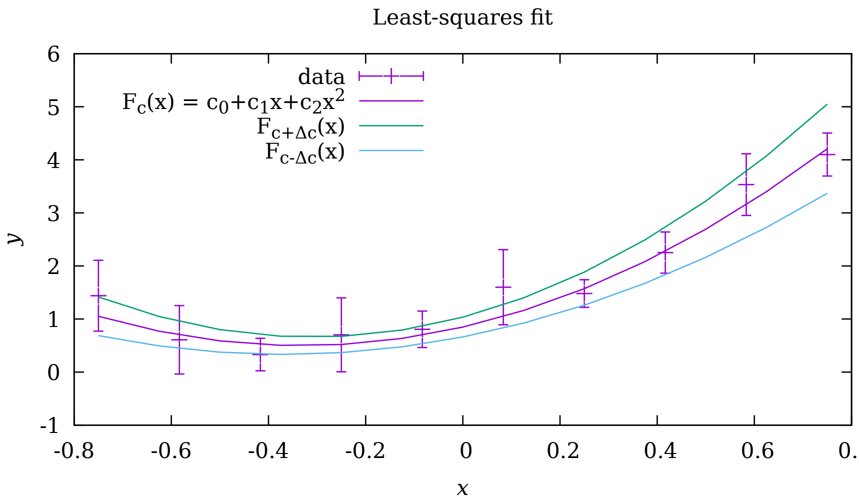


Figure 5: Ordinary least squares fit of $F_{\mathbf{c}}(x) = c_1 + c_2 x + c_3 x^2$ to a set of data. Shown are fits with optimal coefficiens $\mathbf{c}$ as well as with $\mathbf{c} + \Delta\mathbf{c}$ and $\mathbf{c} - \Delta\mathbf{c}$.