Fitting Linear Statistical Models to Data by Least Squares: Introduction

Radu Balan, Brian R. Hunt and C. David Levermore
University of Maryland, College Park

University of Maryland, College Park, MD

Math 420: *Mathematical Modeling*
January 24, 2024 version

Outline

1. Introduction to Linear Statistical Models

In modeling one is often faced with the problem of fitting data with some analytic expression. Let us suppose that we are studying a phenomenon that evolves over time. Given a set of n times $\{t_j\}_{j=1}^{n}$ such that at each time t_j we take a measurement y_j of the phenomenon. We can represent this data as the set of ordered pairs

$$\{(t_j, y_j)\}_{j=1}^{n} \, .$$

Each y_j might be a single number or a vector of numbers. For simplicity, we will first treat the univariate case when it is a single number. The more complicated multivariate case when it is a vector will be treated later.

1. Introduction to Linear Statistical Models

In modeling one is often faced with the problem of fitting data with some analytic expression. Let us suppose that we are studying a phenomenon that evolves over time. Given a set of n times $\{t_j\}_{j=1}^n$ such that at each time t_j we take a measurement y_j of the phenomenon. We can represent this data as the set of ordered pairs

$$\{(t_j, y_j)\}_{j=1}^n .$$

Each y_j might be a single number or a vector of numbers. For simplicity, we will first treat the univariate case when it is a single number. The more complicated multivariate case when it is a vector will be treated later. The basic problem we will examine is the following. *How can you use this data set to make a reasonable guess about what a measurment of this phenomenon might yield at any other time?*

Model Complexity and Overfitting

Of course, you can always find functions $f(t)$ such that $y_j = f(t_j)$ for every $j = 1, \cdots, n$. For example, you can use Lagrange interpolation to construct a unique polynomial of degree at most $n - 1$ that does this. However, such a polynomial often exhibits wild oscillations that make it a useless fit. This phenomena is called *overfitting*. There are two reasons why such difficulties arise.

Model Complexity and Overfitting

Of course, you can always find functions $f(t)$ such that $y_j = f(t_j)$ for every $j = 1, \cdots, n$. For example, you can use Lagrange interpolation to construct a unique polynomial of degree at most $n-1$ that does this. However, such a polynomial often exhibits wild oscillations that make it a useless fit. This phenomena is called *overfitting*. There are two reasons why such difficulties arise.

- The times t_j and measurements y_j are subject to error, so finding a function that fits the data exactly is not a good strategy.

Model Complexity and Overfitting

Of course, you can always find functions $f(t)$ such that $y_j = f(t_j)$ for every $j = 1, \cdots, n$. For example, you can use Lagrange interpolation to construct a unique polynomial of degree at most $n - 1$ that does this. However, such a polynomial often exhibits wild oscillations that make it a useless fit. This phenomena is called *overfitting*. There are two reasons why such difficulties arise.

- The times t_j and measurements y_j are subject to error, so finding a function that fits the data exactly is not a good strategy.
- The assumed form of $f(t)$ might be ill suited for matching the behavior of the phenomenon over the time interval being considered.

Model fitting

One strategy to help avoid these difficulties is to draw $f(t)$ from a family of suitable functions, which is called a *model* in statistics. If we denote this model by $f(t;\beta_1,\cdots,\beta_m)$ where $m << n$ then the idea is to find values of $\beta_1, \cdots, \beta_m$ such that the graph of $f(t;\beta_1,\cdots,\beta_m)$ best fits the data. More precisely, we will define the *residuals* $r_j(\beta_1,\cdots,\beta_m)$ by the relation

$$y_j = f(t_j;\beta_1,\cdots,\beta_m) + r_j(\beta_1,\cdots,\beta_m)\,, \quad \text{for every } j = 1, \cdots, n\,,$$

and try to minimize the $r_j(\beta_1,\cdots,\beta_m)$ in some sense.

Model fitting

One strategy to help avoid these difficulties is to draw $f(t)$ from a family of suitable functions, which is called a *model* in statistics. If we denote this model by $f(t; \beta_1, \cdots, \beta_m)$ where $m << n$ then the idea is to find values of $\beta_1, \cdots, \beta_m$ such that the graph of $f(t; \beta_1, \cdots, \beta_m)$ best fits the data. More precisely, we will define the *residuals* $r_j(\beta_1, \cdots, \beta_m)$ by the relation

$$y_j = f(t_j; \beta_1, \cdots, \beta_m) + r_j(\beta_1, \cdots, \beta_m), \quad \text{for every } j = 1, \cdots, n,$$

and try to minimize the $r_j(\beta_1, \cdots, \beta_m)$ in some sense.
The problem can be simplified by restricting ourselves to models in which the parameters appear linearly — so-called *linear models*. Such a model is specified by the choice of a basis $\{f_i(t)\}_{i=1}^m$ and takes the form

$$f(t; \beta_1, \cdots, \beta_m) = \sum_{i=1}^{m} \beta_i f_i(t).$$

Polynomial and Periodic Models

Example. The most classic linear model is the family of all *polynomials* of degree less than m. This family is often expressed as

$$f(t; \beta_0, \cdots, \beta_{m-1}) = \sum_{i=0}^{m-1} \beta_i \, t^i \, .$$

Notice that here the index i runs from 0 to $m-1$ rather than from 1 to m. This indexing convention is used for polynomial models because it matches the degree of each term in the sum.

Polynomial and Periodic Models

Example. The most classic linear model is the family of all *polynomials* of degree less than m. This family is often expressed as

$$f(t; \beta_0, \cdots, \beta_{m-1}) = \sum_{i=0}^{m-1} \beta_i t^i .$$

Notice that here the index i runs from 0 to $m-1$ rather than from 1 to m. This indexing convention is used for polynomial models because it matches the degree of each term in the sum.

Example. If the underlying phenomena is *periodic* with period T then a classic linear model is the family of all *trigonometric polynomials* of degree at most L. This family can be expressed as

$$f(t; \alpha_0, \cdots, \alpha_l, \beta_1, \cdots, \beta_l) = \alpha_0 + \sum_{k=1}^{L} \left(\alpha_k \cos(k\omega t) + \beta_k \sin(k\omega t)\right) ,$$

where $\omega = 2\pi / T$ its *fundamental frequency*. Note $m = 2L + 1$.

Shift-Invariant Models

Remark. Linear models are linear in the parameters, but are typically nonlinear in the independent variable t. This is illustrated by the foregoing examples: the family of all polynomials of degree less than m is nonlinear in t for $m > 2$; the family of all trigonometric polynomials of degree at most L is nonlinear in t for $L > 0$.

Shift-Invariant Models

Remark. Linear models are linear in the parameters, but are typically nonlinear in the independent variable t. This is illustrated by the foregoing examples: the family of all polynomials of degree less than m is nonlinear in t for $m > 2$; the family of all trigonometric polynomials of degree at most L is nonlinear in t for $L > 0$.

Remark. When there is no preferred instant of time it is best to pick a model $f(t; \beta_1, \cdots, \beta_m)$ that is *translation invariant*. This means for every choice of parameter values $(\beta_1, \cdots, \beta_m)$ and time shift s there exist parameter values $(\beta'_1, \cdots, \beta'_m)$ such that

$$f(t + s; \beta_1, \cdots, \beta_m) = f(t; \beta'_1, \cdots, \beta'_m) \quad \text{for every } t\,.$$

Both models given on the previous slide are translation invariant. Can you show this? Can you find models that are not translation invariant?

Linear Models

It is as easy to work in the more general setting in which we are given data

$$\left\{(\mathbf{x}_j, y_j)\right\}_{j=1}^{n} ,$$

where the $\mathbf{x}_j$ lie within a bounded domain $\mathbb{X} \subset \mathbb{R}^p$ and the y_j lie in $\mathbb{R}$. The problem we will examine now becomes the following.

How can you use this data set to make a reasonable guess about the value of y when $\mathbf{x}$ takes a value in $\mathbb{X}$ that is not represented in the data set?

Linear Models

It is as easy to work in the more general setting in which we are given data

$$\{(\mathbf{x}_j, y_j)\}_{j=1}^{n} ,$$

where the $\mathbf{x}_j$ lie within a bounded domain $\mathbb{X} \subset \mathbb{R}^p$ and the y_j lie in $\mathbb{R}$. The problem we will examine now becomes the following.

How can you use this data set to make a reasonable guess about the value of y when $\mathbf{x}$ takes a value in $\mathbb{X}$ that is not represented in the data set?

We call $\mathbf{x}$ the *independent variable* and y the *dependent variable*. We will consider a linear statistical model with m real parameters in the form

$$f(\mathbf{x}; \beta_1, \cdots, \beta_m) = \sum_{i=1}^{m} \beta_i f_i(\mathbf{x}) ,$$

where each basis function $f_i(\mathbf{x})$ is defined over $\mathbb{X}$ and takes values in $\mathbb{R}$.

Polynomials = linear combinations of monomials

Example. A classic linear model in this setting is the family of all affine functions. If x_i denotes the i^{th} entry of $\mathbf{x}$ then this family can be written as

$$f(\mathbf{x}; a, b_1, \cdots, b_p) = a + \sum_{i=1}^{p} b_i\, x_i\,.$$

Alternatively, it can be expressed in vector notation as

$$f(\mathbf{x}; a, \mathbf{b}) = a + \mathbf{b} \cdot \mathbf{x}\,,$$

where $a \in \mathbb{R}$ and $\mathbf{b} \in \mathbb{R}^p$. Notice that here $m = p + 1$.

Polynomials = linear combinations of monomials

Example. A classic linear model in this setting is the family of all affine functions. If x_i denotes the i^{th} entry of $\mathbf{x}$ then this family can be written as

$$f(\mathbf{x}; a, b_1, \cdots, b_p) = a + \sum_{i=1}^{p} b_i\, x_i\,.$$

Alternatively, it can be expressed in vector notation as

$$f(\mathbf{x}; a, \mathbf{b}) = a + \mathbf{b} \cdot \mathbf{x}\,,$$

where $a \in \mathbb{R}$ and $\mathbf{b} \in \mathbb{R}^p$. Notice that here $m = p + 1$.

Remark. Dimension m for the family of polynomials in p variables of degree at most d grows rapidly:

$$m = \frac{(p+d)!}{p!\,d!} = \frac{(p+1)(p+2)\cdots(p+d)}{d!}\,.$$

Model Residuals or Modeling Noise

Recall that given the data $\{(\mathbf{x}_j, y_j)\}_{j=1}^{n}$ and any model $f(\mathbf{x}; \beta_1, \cdots, \beta_m)$, the residual associated with each $(\mathbf{x}_j, y_j)$ is defined by the relation

$$y_j = f(\mathbf{x}_j; \beta_1, \cdots, \beta_m) + r_j(\beta_1, \cdots, \beta_m) .$$

The linear model given by the basis functions $\{f_i(\mathbf{x})\}_{i=1}^{m}$ is

$$f(\mathbf{x}; \beta_1, \cdots, \beta_m) = \sum_{i=1}^{m} \beta_i f_i(\mathbf{x}) ,$$

for which the residual $r_j(\beta_1, \cdots, \beta_m)$ is given by

$$r_j(\beta_1, \cdots, \beta_m) = y_j - \sum_{i=1}^{m} \beta_i f_i(\mathbf{x}_j) .$$

Model Residuals or Modeling Noise

Recall that given the data $\{(\mathbf{x}_j, y_j)\}_{j=1}^n$ and any model $f(\mathbf{x}; \beta_1, \cdots, \beta_m)$, the residual associated with each $(\mathbf{x}_j, y_j)$ is defined by the relation

$$y_j = f(\mathbf{x}_j; \beta_1, \cdots, \beta_m) + r_j(\beta_1, \cdots, \beta_m)\,.$$

The linear model given by the basis functions $\{f_i(\mathbf{x})\}_{i=1}^m$ is

$$f(\mathbf{x}; \beta_1, \cdots, \beta_m) = \sum_{i=1}^m \beta_i f_i(\mathbf{x})\,,$$

for which the residual $r_j(\beta_1, \cdots, \beta_m)$ is given by

$$r_j(\beta_1, \cdots, \beta_m) = y_j - \sum_{i=1}^m \beta_i f_i(\mathbf{x}_j)\,.$$

The idea is to determine the parameters $\beta_1, \cdots, \beta_m$ in the statistical model by minimizing the residuals $r_j(\beta_1, \cdots, \beta_m)$. In general $m \ll n$ so all the residuals may not vanish.

Linear Models and Residuals: Matrix Notation

This so-called *fitting problem* can be recast in terms of vectors. Introduce the m-vector $\boldsymbol{\beta}$, the n-vectors $\mathbf{y}$ and $\mathbf{r}$, and the $n \times m$-matrix $\mathbf{F}$ by

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_m \end{pmatrix}, \qquad \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \qquad \mathbf{r} = \begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix},$$

$$\mathbf{F} = \begin{pmatrix} f_1(\mathbf{x}_1) & \cdots & f_m(\mathbf{x}_1) \\ \vdots & \vdots & \vdots \\ f_1(\mathbf{x}_n) & \cdots & f_m(\mathbf{x}_n) \end{pmatrix}.$$

Linear Models and Residuals: Matrix Notation

This so-called *fitting problem* can be recast in terms of vectors. Introduce the m-vector $\boldsymbol{\beta}$, the n-vectors $\mathbf{y}$ and $\mathbf{r}$, and the $n \times m$-matrix $\mathbf{F}$ by

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_m \end{pmatrix}, \qquad \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \qquad \mathbf{r} = \begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix},$$

$$\mathbf{F} = \begin{pmatrix} f_1(\mathbf{x}_1) & \cdots & f_m(\mathbf{x}_1) \\ \vdots & \vdots & \vdots \\ f_1(\mathbf{x}_n) & \cdots & f_m(\mathbf{x}_n) \end{pmatrix}.$$

We will assume the matrix $\mathbf{F}$ has rank m. The fitting problem then becomes the problem of finding a value of $\boldsymbol{\beta}$ that minimizes the "size" of

$$\mathbf{r}(\boldsymbol{\beta}) = \mathbf{y} - \mathbf{F}\boldsymbol{\beta}.$$

But what does “size” mean?

2. Linear Euclidean Least Squares Fitting

One popular notion of the size of a vector is the *Euclidean norm*, which is

$$|\mathbf{r}(\boldsymbol{\beta})| = \sqrt{\mathbf{r}(\boldsymbol{\beta})^{\mathrm{T}}\mathbf{r}(\boldsymbol{\beta})} = \sqrt{\sum_{j=1}^{n} r_j(\beta_1, \cdots, \beta_m)^2}\,.$$

2. Linear Euclidean Least Squares Fitting

One popular notion of the size of a vector is the *Euclidean norm*, which is

$$|\mathbf{r}(\boldsymbol{\beta})| = \sqrt{\mathbf{r}(\boldsymbol{\beta})^{\mathrm{T}}\mathbf{r}(\boldsymbol{\beta})} = \sqrt{\sum_{j=1}^{n} r_j(\beta_1, \cdots, \beta_m)^2}\,.$$

Minimizing $|\mathbf{r}(\boldsymbol{\beta})|$ is equivalent to minimizing $|\mathbf{r}(\boldsymbol{\beta})|^2$, which is the sum of the "squares" of the residuals. For linear models $|\mathbf{r}(\boldsymbol{\beta})|^2$ is a quadratic function of $\boldsymbol{\beta}$ that is easy to minimize, which is why the method is popular. Specifically, because $\mathbf{r}(\boldsymbol{\beta}) = \mathbf{y} - \mathbf{F}\boldsymbol{\beta}$, we minimize

$$\begin{aligned} q(\boldsymbol{\beta}) &= \tfrac{1}{2}|\mathbf{r}(\boldsymbol{\beta})|^2 = \tfrac{1}{2}\mathbf{r}(\boldsymbol{\beta})^{\mathrm{T}}\mathbf{r}(\boldsymbol{\beta}) = \tfrac{1}{2}(\mathbf{y} - \mathbf{F}\boldsymbol{\beta})^{\mathrm{T}}(\mathbf{y} - \mathbf{F}\boldsymbol{\beta}) \\ &= \tfrac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{y} - \boldsymbol{\beta}^{\mathrm{T}}\mathbf{F}^{\mathrm{T}}\mathbf{y} + \tfrac{1}{2}\boldsymbol{\beta}^{\mathrm{T}}\mathbf{F}^{\mathrm{T}}\mathbf{F}\boldsymbol{\beta}\,. \end{aligned}$$

2. Linear Euclidean Least Squares Fitting

One popular notion of the size of a vector is the *Euclidean norm*, which is

$$|\mathbf{r}(\boldsymbol{\beta})| = \sqrt{\mathbf{r}(\boldsymbol{\beta})^{\mathrm{T}}\mathbf{r}(\boldsymbol{\beta})} = \sqrt{\sum_{j=1}^{n} r_j(\beta_1, \cdots, \beta_m)^2}\,.$$

Minimizing $|\mathbf{r}(\boldsymbol{\beta})|$ is equivalent to minimizing $|\mathbf{r}(\boldsymbol{\beta})|^2$, which is the sum of the "squares" of the residuals. For linear models $|\mathbf{r}(\boldsymbol{\beta})|^2$ is a quadratic function of $\boldsymbol{\beta}$ that is easy to minimize, which is why the method is popular. Specifically, because $\mathbf{r}(\boldsymbol{\beta}) = \mathbf{y} - \mathbf{F}\boldsymbol{\beta}$, we minimize

$$\begin{aligned} q(\boldsymbol{\beta}) &= \tfrac{1}{2}|\mathbf{r}(\boldsymbol{\beta})|^2 = \tfrac{1}{2}\mathbf{r}(\boldsymbol{\beta})^{\mathrm{T}}\mathbf{r}(\boldsymbol{\beta}) = \tfrac{1}{2}(\mathbf{y} - \mathbf{F}\boldsymbol{\beta})^{\mathrm{T}}(\mathbf{y} - \mathbf{F}\boldsymbol{\beta}) \\ &= \tfrac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{y} - \boldsymbol{\beta}^{\mathrm{T}}\mathbf{F}^{\mathrm{T}}\mathbf{y} + \tfrac{1}{2}\boldsymbol{\beta}^{\mathrm{T}}\mathbf{F}^{\mathrm{T}}\mathbf{F}\boldsymbol{\beta}\,. \end{aligned}$$

We will use multivariable calculus to minimize this quadratic function.

The Gradient

Recall that the gradient (if it exists) of a real-valued function $q(\boldsymbol{\beta})$ with respect to the m-vector $\boldsymbol{\beta}$ is the m-vector $\partial_{\boldsymbol{\beta}} q(\boldsymbol{\beta})$ such that

$$\frac{\mathrm{d}}{\mathrm{d}s} q(\boldsymbol{\beta} + s\boldsymbol{\gamma})\Big|_{s=0} = \boldsymbol{\gamma}^{\mathrm{T}} \partial_{\boldsymbol{\beta}} q(\boldsymbol{\beta}) \quad \text{for every } \boldsymbol{\gamma} \in \mathbb{R}^m .$$

The Gradient

Recall that the gradient (if it exists) of a real-valued function $q(\boldsymbol{\beta})$ with respect to the m-vector $\boldsymbol{\beta}$ is the m-vector $\partial_{\boldsymbol{\beta}} q(\boldsymbol{\beta})$ such that

$$\frac{\mathrm{d}}{\mathrm{d}s} q(\boldsymbol{\beta} + s\gamma)\Big|_{s=0} = \gamma^{\mathrm{T}} \partial_{\boldsymbol{\beta}} q(\boldsymbol{\beta}) \quad \text{for every } \gamma \in \mathbb{R}^m .$$

In particular, for the quadratic $q(\boldsymbol{\beta})$ arising from our least squares problem we can easily check that

$$q(\boldsymbol{\beta} + s\gamma) = q(\boldsymbol{\beta}) + s\gamma^{\mathrm{T}} \left(\mathbf{F}^{\mathrm{T}}\mathbf{F}\boldsymbol{\beta} - \mathbf{F}^{\mathrm{T}}\mathbf{y} \right) + \tfrac{1}{2} s^2 \gamma^{\mathrm{T}} \mathbf{F}^{\mathrm{T}}\mathbf{F}\gamma .$$

By differentiating this with respect to s and setting $s = 0$ we obtain

$$\frac{\mathrm{d}}{\mathrm{d}s} q(\boldsymbol{\beta} + s\gamma)\Big|_{s=0} = \gamma^{\mathrm{T}} \left(\mathbf{F}^{\mathrm{T}}\mathbf{F}\boldsymbol{\beta} - \mathbf{F}^{\mathrm{T}}\mathbf{y} \right) ,$$

from which we read off that

$$\partial_{\boldsymbol{\beta}} q(\boldsymbol{\beta}) = \mathbf{F}^{\mathrm{T}}\mathbf{F}\boldsymbol{\beta} - \mathbf{F}^{\mathrm{T}}\mathbf{y} .$$

The Hessian

Similarly, the derivative (if it exists) of the vector-valued function $\partial_{\boldsymbol{\beta}} q(\boldsymbol{\beta})$ with respect to the m-vector $\boldsymbol{\beta}$ is the $m \times m$-matrix $\partial_{\boldsymbol{\beta}\boldsymbol{\beta}} q(\boldsymbol{\beta})$ such that

$$\frac{\mathrm{d}}{\mathrm{d}s} \partial_{\boldsymbol{\beta}} q(\boldsymbol{\beta} + s\gamma)\Big|_{s=0} = \partial_{\boldsymbol{\beta}\boldsymbol{\beta}} q(\boldsymbol{\beta})\gamma \quad \text{for every } \gamma \in \mathbb{R}^m .$$

The symmetric matrix-valued function $\partial_{\boldsymbol{\beta}\boldsymbol{\beta}} q(\boldsymbol{\beta})$ is sometimes called the *Hessian* of $q(\boldsymbol{\beta})$.

The Hessian

Similarly, the derivative (if it exists) of the vector-valued function $\partial_{\boldsymbol{\beta}} q(\boldsymbol{\beta})$ with respect to the m-vector $\boldsymbol{\beta}$ is the $m \times m$-matrix $\partial_{\boldsymbol{\beta}\boldsymbol{\beta}} q(\boldsymbol{\beta})$ such that

$$\frac{\mathrm{d}}{\mathrm{d}s} \partial_{\boldsymbol{\beta}} q(\boldsymbol{\beta} + s\gamma)\Big|_{s=0} = \partial_{\boldsymbol{\beta}\boldsymbol{\beta}} q(\boldsymbol{\beta})\gamma \quad \text{for every } \gamma \in \mathbb{R}^m .$$

The symmetric matrix-valued function $\partial_{\boldsymbol{\beta}\boldsymbol{\beta}} q(\boldsymbol{\beta})$ is sometimes called the *Hessian* of $q(\boldsymbol{\beta})$. For the quadratic $q(\boldsymbol{\beta})$ arising from our least squares problem we can easily check that

$$\partial_{\boldsymbol{\beta}} q(\boldsymbol{\beta} + s\gamma) = \mathbf{F}^{\mathrm{T}}\mathbf{F}(\boldsymbol{\beta} + s\gamma) - \mathbf{F}^{\mathrm{T}}\mathbf{y} = \partial_{\boldsymbol{\beta}} q(\boldsymbol{\beta}) + s\mathbf{F}^{\mathrm{T}}\mathbf{F}\gamma .$$

By differentiating this with respect to s and setting $s = 0$ we obtain

$$\frac{\mathrm{d}}{\mathrm{d}s} \partial_{\boldsymbol{\beta}} q(\boldsymbol{\beta} + s\gamma)\Big|_{s=0} = \frac{\mathrm{d}}{\mathrm{d}s} \big(\partial_{\boldsymbol{\beta}} q(\boldsymbol{\beta}) + s\mathbf{F}^{\mathrm{T}}\mathbf{F}\gamma\big)\Big|_{s=0} = \mathbf{F}^{\mathrm{T}}\mathbf{F}\gamma ,$$

from which we read off that

$$\partial_{\boldsymbol{\beta}\boldsymbol{\beta}} q(\boldsymbol{\beta}) = \mathbf{F}^{\mathrm{T}}\mathbf{F}$$

The Hessian

Similarly, the derivative (if it exists) of the vector-valued function $\partial_{\boldsymbol{\beta}} q(\boldsymbol{\beta})$ with respect to the m-vector $\boldsymbol{\beta}$ is the $m \times m$-matrix $\partial_{\boldsymbol{\beta}\boldsymbol{\beta}} q(\boldsymbol{\beta})$ such that

$$\frac{\mathrm{d}}{\mathrm{d}s} \partial_{\boldsymbol{\beta}} q(\boldsymbol{\beta} + s\gamma)\Big|_{s=0} = \partial_{\boldsymbol{\beta}\boldsymbol{\beta}} q(\boldsymbol{\beta})\gamma \quad \text{for every } \gamma \in \mathbb{R}^m .$$

The symmetric matrix-valued function $\partial_{\boldsymbol{\beta}\boldsymbol{\beta}} q(\boldsymbol{\beta})$ is sometimes called the *Hessian* of $q(\boldsymbol{\beta})$. For the quadratic $q(\boldsymbol{\beta})$ arising from our least squares problem we can easily check that

$$\partial_{\boldsymbol{\beta}} q(\boldsymbol{\beta} + s\gamma) = \mathbf{F}^{\mathrm{T}}\mathbf{F}(\boldsymbol{\beta} + s\gamma) - \mathbf{F}^{\mathrm{T}}\mathbf{y} = \partial_{\boldsymbol{\beta}} q(\boldsymbol{\beta}) + s\mathbf{F}^{\mathrm{T}}\mathbf{F}\gamma .$$

By differentiating this with respect to s and setting $s = 0$ we obtain

$$\frac{\mathrm{d}}{\mathrm{d}s} \partial_{\boldsymbol{\beta}} q(\boldsymbol{\beta} + s\gamma)\Big|_{s=0} = \frac{\mathrm{d}}{\mathrm{d}s}\left(\partial_{\boldsymbol{\beta}} q(\boldsymbol{\beta}) + s\mathbf{F}^{\mathrm{T}}\mathbf{F}\gamma\right)\Big|_{s=0} = \mathbf{F}^{\mathrm{T}}\mathbf{F}\gamma ,$$

from which we read off that

$$\partial_{\boldsymbol{\beta}\boldsymbol{\beta}} q(\boldsymbol{\beta}) = \mathbf{F}^{\mathrm{T}}\mathbf{F} \quad \text{and} \quad \mathbf{F}^{\mathrm{T}}\mathbf{F} > 0.$$

Convexity and Strict Convexity

Because $\partial_{\boldsymbol{\beta}\boldsymbol{\beta}} q(\boldsymbol{\beta})$ is positive definite, the function $q(\boldsymbol{\beta})$ is strictly convex, whereby it has at most one global minimizer. We find this minimizer by setting the gradient of $q(\boldsymbol{\beta})$ equal to zero, yielding

$$\partial_{\boldsymbol{\beta}} q(\boldsymbol{\beta}) = \mathbf{F}^{\mathrm{T}}\mathbf{F}\boldsymbol{\beta} - \mathbf{F}^{\mathrm{T}}\mathbf{y} = \mathbf{0}\,.$$

Convexity and Strict Convexity

Because $\partial_{\boldsymbol{\beta}\boldsymbol{\beta}} q(\boldsymbol{\beta})$ is positive definite, the function $q(\boldsymbol{\beta})$ is strictly convex, whereby it has at most one global minimizer. We find this minimizer by setting the gradient of $q(\boldsymbol{\beta})$ equal to zero, yielding

$$\partial_{\boldsymbol{\beta}}\, q(\boldsymbol{\beta}) = \mathbf{F}^{\mathrm{T}}\mathbf{F}\boldsymbol{\beta} - \mathbf{F}^{\mathrm{T}}\mathbf{y} = \mathbf{0}\,.$$

Because the matrix $\mathbf{F}^{\mathrm{T}}\mathbf{F}$ is positive definite, it is invertible. The solution of the above equation is therefore $\boldsymbol{\beta} = \widehat{\boldsymbol{\beta}}$ where

$$\widehat{\boldsymbol{\beta}} = (\mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1}\mathbf{F}^{\mathrm{T}}\mathbf{y}\,.$$

The fact that $\widehat{\boldsymbol{\beta}}$ is a global minimizer can be seen from the fact $\mathbf{F}^{\mathrm{T}}\mathbf{F}$ is positive definite and the identity

$$\begin{aligned} q(\boldsymbol{\beta}) &= \tfrac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{y} - \tfrac{1}{2}\widehat{\boldsymbol{\beta}}^{\mathrm{T}}\mathbf{F}^{\mathrm{T}}\mathbf{F}\widehat{\boldsymbol{\beta}} + \tfrac{1}{2}(\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}})^{\mathrm{T}}\mathbf{F}^{\mathrm{T}}\mathbf{F}(\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}) \\ &= q(\widehat{\boldsymbol{\beta}}) + \tfrac{1}{2}(\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}})^{\mathrm{T}}\mathbf{F}^{\mathrm{T}}\mathbf{F}(\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}})\,. \end{aligned}$$

Geometric Interpretation. Orthogonal Projections

Remark. The least squares fit has a beautiful geometric interpretation with respect to the associated Euclidean inner product

$$(\mathbf{p} \,|\, \mathbf{q}) = \mathbf{p}^{\mathrm{T}}\mathbf{q}\,.$$

Define $\widehat{\mathbf{r}} = \mathbf{r}(\widehat{\beta}) = \mathbf{y} - \mathbf{F}\widehat{\beta}$. Observe that

$$\mathbf{y} = \mathbf{F}\widehat{\beta} + \widehat{\mathbf{r}} = \mathbf{F}(\mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1}\mathbf{F}^{\mathrm{T}}\mathbf{y} + \widehat{\mathbf{r}}\,.$$

Geometric Interpretation. Orthogonal Projections

Remark. The least squares fit has a beautiful geometric interpretation with respect to the associated Euclidean inner product

$$(\mathbf{p} \mid \mathbf{q}) = \mathbf{p}^{\mathrm{T}}\mathbf{q}\,.$$

Define $\widehat{\mathbf{r}} = \mathbf{r}(\widehat{\beta}) = \mathbf{y} - \mathbf{F}\widehat{\beta}$. Observe that

$$\mathbf{y} = \mathbf{F}\widehat{\beta} + \widehat{\mathbf{r}} = \mathbf{F}(\mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1}\mathbf{F}^{\mathrm{T}}\mathbf{y} + \widehat{\mathbf{r}}\,.$$

The matrix $\mathbf{P} = \mathbf{F}(\mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1}\mathbf{F}^{\mathrm{T}}$ has the properties

$$\mathbf{P}^2 = \mathbf{P}\,, \qquad \mathbf{P}^{\mathrm{T}} = \mathbf{P}\,.$$

This means that $\mathbf{P}\mathbf{y}$ is the orthogonal projection of $\mathbf{y}$ onto the subspace of $\mathbb{R}^n$ spanned by the columns of $\mathbf{F}$, and that $\mathbf{y} = \mathbf{P}\mathbf{y} + \widehat{\mathbf{r}}$ is an orthogonal decomposition of $\mathbf{y}$.

Geometric Interpretation. Orthogonal Projections

Remark. The least squares fit has a beautiful geometric interpretation with respect to the associated Euclidean inner product

$$(\mathbf{p} \,|\, \mathbf{q}) = \mathbf{p}^{\mathrm{T}}\mathbf{q} \,.$$

Define $\widehat{\mathbf{r}} = \mathbf{r}(\widehat{\beta}) = \mathbf{y} - \mathbf{F}\widehat{\beta}$. Observe that

$$\mathbf{y} = \mathbf{F}\widehat{\beta} + \widehat{\mathbf{r}} = \mathbf{F}(\mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1}\mathbf{F}^{\mathrm{T}}\mathbf{y} + \widehat{\mathbf{r}} \,.$$

The matrix $\mathbf{P} = \mathbf{F}(\mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1}\mathbf{F}^{\mathrm{T}}$ has the properties

$$\mathbf{P}^2 = \mathbf{P} \,, \qquad \mathbf{P}^{\mathrm{T}} = \mathbf{P} \,.$$

This means that $\mathbf{P}\mathbf{y}$ is the orthogonal projection of $\mathbf{y}$ onto the subspace of $\mathbb{R}^n$ spanned by the columns of $\mathbf{F}$, and that $\mathbf{y} = \mathbf{P}\mathbf{y} + \widehat{\mathbf{r}}$ is an orthogonal decomposition of $\mathbf{y}$. Since $\mathbf{F}^{\mathrm{T}}\mathbf{P} = \mathbf{F}^{\mathrm{T}}$ we get $\mathbf{F}^{\mathrm{T}}\widehat{\mathbf{r}} = 0$. This says that residual $\widehat{\mathbf{r}}$ is orthogonal to every column of $\mathbf{F}$; recall that each of these columns corresponds to a basis function. Thus, $\widehat{\mathbf{r}}$ will have mean zero if the constant function 1 is one of the basis functions.

A 2-dimensional Example

Example. Least Squares for the affine model $f(t;\alpha,\beta) = \alpha + \beta t$ and data $\{(t_j, y_j)\}_{j=1}^n$. Matrix **F** has the form

$$\mathbf{F} = \begin{pmatrix} \mathbf{1} & \mathbf{t} \end{pmatrix}, \qquad \text{where} \quad \mathbf{1} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}, \quad \mathbf{t} = \begin{pmatrix} t_1 \\ \vdots \\ t_n \end{pmatrix}.$$

Define

$$\bar{t} = \frac{1}{n}\sum_{j=1}^n t_j\,, \qquad \overline{t^2} = \frac{1}{n}\sum_{j=1}^n t_j^2\,, \qquad \sigma_t^2 = \frac{1}{n}\sum_{j=1}^n (t_j - \bar{t}\,)^2\,,$$

A 2-dimensional Example

Example. Least Squares for the affine model $f(t;\alpha,\beta) = \alpha + \beta t$ and data $\{(t_j, y_j)\}_{j=1}^n$. Matrix $\mathbf{F}$ has the form

$$\mathbf{F} = \begin{pmatrix} \mathbf{1} & \mathbf{t} \end{pmatrix}, \qquad \text{where} \quad \mathbf{1} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}, \quad \mathbf{t} = \begin{pmatrix} t_1 \\ \vdots \\ t_n \end{pmatrix}.$$

Define

$$\bar{t} = \frac{1}{n}\sum_{j=1}^n t_j, \qquad \overline{t^2} = \frac{1}{n}\sum_{j=1}^n t_j^2, \qquad \sigma_t^2 = \frac{1}{n}\sum_{j=1}^n (t_j - \bar{t}\,)^2,$$

To obtain:

$$\mathbf{F}^{\mathrm{T}}\mathbf{F} = \begin{pmatrix} \mathbf{1}^{\mathrm{T}}\mathbf{1} & \mathbf{1}^{\mathrm{T}}\mathbf{t} \\ \mathbf{t}^{\mathrm{T}}\mathbf{1} & \mathbf{t}^{\mathrm{T}}\mathbf{t} \end{pmatrix} = n\begin{pmatrix} 1 & \bar{t} \\ \bar{t} & \overline{t^2} \end{pmatrix},$$

$$\det(\mathbf{F}^{\mathrm{T}}\mathbf{F}) = n^2\left(\overline{t^2} - \bar{t}^2\right) = n^2\sigma_t^2 > 0.$$

Notice that $\bar{t}$ and σ_t^2 are the sample mean and variance of t respectively.

The 2-dimensional Example: Explicit Formulas

Then the $\widehat{\alpha}$ and $\widehat{\beta}$ that give the least squares fit are given by

$$\begin{pmatrix} \widehat{\alpha} \\ \widehat{\beta} \end{pmatrix} = \widehat{\boldsymbol{\beta}} = (\mathbf{F}^{\mathrm{T}}\mathbf{F})^{-1}\mathbf{F}^{\mathrm{T}}\mathbf{y} = \frac{1}{n}\frac{1}{\sigma_t^2}\begin{pmatrix} \overline{t^2} & -\bar{t} \\ -\bar{t} & 1 \end{pmatrix}\begin{pmatrix} \mathbf{1}^{\mathrm{T}} \\ \mathbf{t}^{\mathrm{T}} \end{pmatrix}\mathbf{y}$$

$$= \frac{1}{\sigma_t^2}\begin{pmatrix} \overline{t^2} & -\bar{t} \\ -\bar{t} & 1 \end{pmatrix}\begin{pmatrix} \bar{y} \\ \overline{ty} \end{pmatrix} = \frac{1}{\sigma_t^2}\begin{pmatrix} \overline{t^2}\,\bar{y} - \bar{t}\,\overline{ty} \\ \overline{ty} - \bar{t}\,\bar{y} \end{pmatrix},$$

where

$$\bar{y} = \frac{1}{n}\mathbf{1}^{\mathrm{T}}\mathbf{y} = \frac{1}{n}\sum_{j=1}^{n} y_j\,, \qquad \overline{yt} = \frac{1}{n}\mathbf{t}^{\mathrm{T}}\mathbf{y} = \frac{1}{n}\sum_{j=1}^{n} y_j t_j\,.$$

These formulas for $\widehat{\alpha}$ and $\widehat{\beta}$ can be expressed simply as

$$\widehat{\beta} = \frac{\overline{yt} - \bar{y}\,\bar{t}}{\sigma_t^2}\,, \qquad \widehat{\alpha} = \bar{y} - \widehat{\beta}\bar{t}\,.$$

Notice that $\widehat{\beta}$ is the ratio of the covariance of y and t to the variance of t.

Least Squares for the General Linear Model

The best fit is therefore

$$\widehat{f}(t) = \widehat{\alpha} + \widehat{\beta} t = \bar{y} + \widehat{\beta}(t - \bar{t}\,) = \bar{y} + \frac{\overline{yt} - \bar{y}\,\bar{t}}{\sigma_t^2}(t - \bar{t}\,)\,.$$

Least Squares for the General Linear Model

The best fit is therefore

$$\widehat{f}(t) = \widehat{\alpha} + \widehat{\beta}t = \bar{y} + \widehat{\beta}(t - \bar{t}\,) = \bar{y} + \frac{\overline{yt} - \bar{y}\,\bar{t}}{\sigma_t^2}\,(t - \bar{t}\,)\,.$$

Remark. In the above example we inverted the matrix $\mathbf{F}^{\mathrm{T}}\mathbf{F}$ to obtain $\widehat{\boldsymbol{\beta}}$. This was easy because our model had only two parameters in it, so $\mathbf{F}^{\mathrm{T}}\mathbf{F}$ was only 2×2. The number of paramenters m does not have to be too large before this approach becomes slow or unfeasible. However for fairly large m you can obtain $\widehat{\boldsymbol{\beta}}$ by using Gaussian elimination or some other direct method to efficiently solve the linear system

$$\mathbf{F}^{\mathrm{T}}\mathbf{F}\boldsymbol{\beta} = \mathbf{F}^{\mathrm{T}}\mathbf{y}\,.$$

Such methods work because the matrix $\mathbf{F}^{\mathrm{T}}\mathbf{F}$ is positive definite. As we will soon see, this step can be simplified by constructing the basis $\{f_i(t)\}_{i=1}^{m}$ so that $\mathbf{F}^{\mathrm{T}}\mathbf{F}$ is diagonal.

3. Auto-Regressive Processes

Consider a time-series $(x(t))_{t=-\infty}^{\infty}$ where each sample $x(t)$ can be scalar or vector. We say that $(x(t))_t$ is the output of an *Auto-Regressive process of order p*, denoted $AR(p)$, if there are (scalar or matrix) constants $a_1, \ldots, a_p$ so that

$$x(t) = a_1 x(t-1) + a_2 x(t-2) + \cdots a_p x(t-p) + \nu(t).$$

Here $(\nu(t))_t$ is a different time-series called the *driving noise*, or the *excitation*.

3. Auto-Regressive Processes

Consider a time-series $(x(t))_{t=-\infty}^{\infty}$ where each sample $x(t)$ can be scalar or vector. We say that $(x(t))_t$ is the output of an *Auto-Regressive process of order p*, denoted $AR(p)$, if there are (scalar or matrix) constants $a_1, \ldots, a_p$ so that

$$x(t) = a_1 x(t-1) + a_2 x(t-2) + \cdots a_p x(t-p) + \nu(t).$$

Here $(\nu(t))_t$ is a different time-series called the *driving noise*, or the *excitation*.

Compare the two type of 'noises' we have seen so far:
Measurement Noise: $y_t = Fx_t + r_t$

3. Auto-Regressive Processes

Consider a time-series $(x(t))_{t=-\infty}^{\infty}$ where each sample $x(t)$ can be scalar or vector. We say that $(x(t))_t$ is the output of an *Auto-Regressive process of order p*, denoted $AR(p)$, if there are (scalar or matrix) constants $a_1, \ldots, a_p$ so that

$$x(t) = a_1 x(t-1) + a_2 x(t-2) + \cdots a_p x(t-p) + \nu(t).$$

Here $(\nu(t))_t$ is a different time-series called the *driving noise*, or the *excitation*.

Compare the two type of 'noises' we have seen so far:
Measurement Noise: $y_t = Fx_t + r_t$ *Driving Noise*: $x_t = A(x(t-)) + \nu_t$

Scalar AR(p) process

Given a time-series $(x_t)_t$, the least squares estimator of the parameters of an $AR(p)$ process solves the following minimization problem:

$$\min_{a_1,\ldots,a_p} \sum_{t=1}^{T} |x_t - a_1 x(t-1) - \cdots - a_p x(t-p)|^2$$

Scalar AR(p) process

Given a time-series $(x_t)_t$, the least squares estimator of the parameters of an $AR(p)$ process solves the following minimization problem:

$$\min_{a_1,\ldots,a_p} \sum_{t=1}^{T} |x_t - a_1 x(t-1) - \cdots - a_p x(t-p)|^2$$

Expanding the square and rearranging the terms we get $a^T Ra - 2a^T q + \rho(0)$ where

$$R = \begin{bmatrix} \rho(0) & \rho(-1) & \cdots & \rho(p-1) \\ \rho(1) & \rho(0) & \cdots & \rho(p-2) \\ \vdots & & \ddots & \vdots \\ \rho(p-1) & \rho(p-2) & \cdots & \rho(0) \end{bmatrix}, \quad q = \begin{bmatrix} \rho(1) \\ \rho(2) \\ \vdots \\ \rho(p-1) \end{bmatrix}$$

and $\rho(\tau) = \sum_{t=1}^{T} x_t x_{t-\tau}$ is the auto-correlation function.

Scalar AR(p) process

Computing the gradient for the minimization problem

$$\min_{a=[a_1,\ldots,a_p]^T} \quad a^T R a - 2a^T q + \rho(0)$$

produces the closed form solution

$$\hat{a} = R^{-1} q$$

that is, the solution of the linear system $Ra = q$ called the *Yule-Walker system*.
An efficient adaptive (on-line) solver is given by the Levinson-Durbin algorithm.

Multivariate AR(1) Processes

The Multivariate AR(1) process is defined by the linear process:

$$\mathbf{x}(t) = W\mathbf{x}(t-1) + \nu(t)$$

where $\mathbf{x}(t)$ is the n-vector describing the state at time t, and $\nu(t)$ is the driving noise vector at time t. The $n \times n$ matrix W is the unknown matrix of coefficients.

Multivariate AR(1) Processes

The Multivariate AR(1) process is defined by the linear process:

$$\mathbf{x}(t) = W\mathbf{x}(t-1) + \nu(t)$$

where $\mathbf{x}(t)$ is the n-vector describing the state at time t, and $\nu(t)$ is the driving noise vector at time t. The $n \times n$ matrix W is the unknown matrix of coefficients.

In general the matrix W may not have to be symmetric.

However there are cases when we are interested in symmetric $AR(1)$ processes. One such case is furnished by undirected weighted graphs. Furthermore, the matrix W may have to satisfy additional constraints. One such constraint is to have zero main diagonal. Alternate case is for W to have constant 1 along the main diagonal.

LSE for Vector AR(1) with zero main diagonal

$$\textit{LS Estimator}: \quad \min_{W \in \mathbb{R}^{n \times n}} \sum_{t=1}^{T} \|\mathbf{x}(t) - W\mathbf{x}(t-1)\|^2$$

$$\text{subject to}: W = W^T$$

$$diag(W) = 0$$

LSE for Vector AR(1) with zero main diagonal

$$\textit{LS Estimator}: \quad \min_{W \in \mathbb{R}^{n\times n}} \sum_{t=1}^{T} \|\mathbf{x}(t) - W\mathbf{x}(t-1)\|^2$$
$$\text{subject to}: W = W^T$$
$$diag(W) = 0$$

How to find W: Rewrite the criterion as a quadratic form in variable $z = vec(W)$, the independent entries in W. If $\mathbf{x}(t) \in \mathbb{R}^n$ is n-dimensional, then z has dimension $m = n(n-1)/2$:

$$z^T = \begin{bmatrix} W_{12} & W_{13} & \cdots & W_{1n} & W_{23} & \cdots & W_{n-1,n} \end{bmatrix}$$

Let $A(t)$ denote the $n \times m$ matrix so that $W\mathbf{x}(t) = A(t)z$. For $n = 3$:

$$A(t) = \begin{bmatrix} \mathbf{x}(t)_2 & \mathbf{x}(t)_3 & 0 \\ \mathbf{x}(t)_1 & 0 & \mathbf{x}(t)_3 \\ 0 & \mathbf{x}(t)_1 & \mathbf{x}(t)_2 \end{bmatrix}$$

LSE for Vector AR(1) with zero main diagonal

Then

$$J(W) = \sum_{t=1}^{T} (\mathbf{x}(t) - A(t)z)^T (\mathbf{x}(t) - A(t)z) = z^T R z - 2z^T q + r_0$$

where

$$R = \sum_{t=1}^{T} A(t)^T A(t) \ , \ \ q = \sum_{t=1}^{T} A(t)^T \mathbf{x}(t) \ , \ \ r_0 = \sum_{t=1}^{T} \|\mathbf{x}(t)\|^2.$$

The optimal solution solves the linear system

$$Rz = q \quad \Rightarrow \quad z = R^{-1} q.$$

Then the Least Square estimator W is obtained by reshaping z into a symmetric $n \times n$ matrix of 0 diagonal.

LSE for Vector AR(1) with unit main diagonal

$$LS\ Estimator: \min_{W \in \mathbb{R}^{n\times n}} \sum_{t=1}^{T} \|\mathbf{x}(t) - W\mathbf{x}(t-1)\|^2$$

$$\text{subject to}: W = W^T$$

$$diag(W) = ones(n,1)$$

LSE for Vector AR(1) with unit main diagonal

$$\textit{LS Estimator}: \quad \min_{W \in \mathbb{R}^{n\times n}} \sum_{t=1}^{T} \|\mathbf{x}(t) - W\mathbf{x}(t-1)\|^2$$
$$\text{subject to}: W = W^T$$
$$diag(W) = ones(n,1)$$

How to find W: Rewrite the criterion as a quadratic form in variable $z = vec(W)$, the independent entries in W. If $\mathbf{x}(t) \in \mathbb{R}^n$ is n-dimensional, then z has dimension $m = n(n-1)/2$:

$$z^T = \begin{bmatrix} W_{12} & W_{13} & \cdots & W_{1n} & W_{23} & \cdots & W_{n-1,n} \end{bmatrix}$$

Let $A(t)$ denote the $n \times m$ matrix so that $W\mathbf{x}(t-1) = A(t)z + \mathbf{x}(t-1)$. For $n = 3$:

$$A(t) = \begin{bmatrix} \mathbf{x}(t-1)_2 & \mathbf{x}(t-1)_3 & 0 \\ \mathbf{x}(t-1)_1 & 0 & \mathbf{x}(t-1)_3 \\ 0 & \mathbf{x}(t-1)_1 & \mathbf{x}(t-1)_2 \end{bmatrix}$$

LSE for Vector AR(1) with unit main diagonal

Then

$$J(W) = \sum_{t=1}^{T}(\mathbf{x}(t)-A(t)z-\mathbf{x}(t-1))^T(\mathbf{x}(t)-A(t)z-\mathbf{x}(t-1)) = z^T Rz-2z^T q+r_0$$

where

$$R = \sum_{t=1}^{T} A(t)^T A(t) \ , \ \ q = \sum_{t=1}^{T} A(t)^T(\mathbf{x}(t)-\mathbf{x}(t-1)) \ , \ \ r_0 = \sum_{t=1}^{T} \|\mathbf{x}(t) - \mathbf{x}(t-1)\|^2$$

The optimal solution solves the linear system

$$Rz = q \ \ \Rightarrow \ \ z = R^{-1}q.$$

Then the Least Square estimator W is obtained by reshaping z into a symmetric $n \times n$ matrix with 1 on main diagonal.

Further Questions

We have seen how to use least squares to fit linear statistical models with m parameters to data sets containing n pairs when $m << n$. Among the questions that arise are the following.

- How does one pick a basis that is well suited to the given data?
- How can one avoid overfitting?
- Do these methods extended to nonlinear statistical models?
- Can one use other notions of smallness of the residual? Maximum Likelihood Estimation.