Restricted Maximum Likelihood Estimation

Lecture Notes

Ahmet Ademoglu, *PhD*
Bogazici University
Institute of Biomedical Engineering

Some concepts and illustrations in this lecture are adapted from the textbooks,

Pattern Recognition and Machine Learning, C. M. Bishop, *Springer*, 2006.

Statistical Parametric Mapping: The Analysis of Functional Brain Images, Editors: K. Friston, J. Ashburner, S. Kiebel, T. Nichols and W. Penny, *Academic Press*, 2006.

The General Linear Model with Normal Residual Error

$$\mathbf{y} = \mathbf{X}\beta + \mathbf{e}$$

with $\mathbf{e} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$ and $\mathbf{y} \sim \mathcal{N}(\mathbf{X}\beta, \boldsymbol{\Sigma})$

Maximum Likelihood Estimation

The log-likelihood function, $\mathcal{L} = \log p(\mathbf{y}|\mathbf{X}, \beta)$ is

$$\mathcal{L} = -N \log(2\pi) - \frac{1}{2} \log |\Sigma| - \frac{1}{2}(\mathbf{y} - \mathbf{X}\beta)^T \Sigma^{-1} (\mathbf{y} - \mathbf{X}\beta)$$

Maximum Likelihood Solution for β and Σ are obtained by solving

$$\frac{\partial}{\partial \beta}\mathcal{L} \quad = 0$$

$$\frac{\partial}{\partial \beta}\mathcal{L} = 2\mathbf{X}^T \Sigma^{-1} \mathbf{X}\beta - 2\mathbf{X}^T \Sigma^{-1} \mathbf{y} = 0$$

which yields $\beta_{ML} = (\mathbf{X}^T \Sigma^{-1} \mathbf{X}^{-1})\mathbf{X}^T \Sigma^{-1} \mathbf{y}$

2-level Hierarchical Model

Effects from N subjects with n replications per subject, the Collapsed Model for the two-level population effect :
$y_{ij} = w_{pop} + z_i + e_{ij}$
e_{ij} : within subject error $\sim \mathcal{N}(0, \sigma_w^2)$
y_{ij} : j^{th} observed effect for subject i
$z_i \sim \mathcal{N}(0, \sigma_b^2)$ between-subject error for the i^{th} subject

Maximum Likelihood Estimate of w_{pop} :

$$\hat{w}_{pop} = \frac{1}{Nn}\sum_{i=1}^{N}\sum_{j=1}^{n} y_{ij}$$

$$y_{ij} \sim \mathcal{N}(w_{pop}, \sigma_w^2 + \sigma_b^2)$$

$$\log p(\mathbf{y}|w_{pop}) = \sum_{i,j=1}^{n,N} \log \mathcal{N}(w_{pop}, \sigma_w^2 + \sigma_b^2)$$

$$= -\frac{1}{2(\sigma_w^2+\sigma_b^2)}\sum_{i,j=1}^{n,N}(y_{ij} - w_{pop})^2 - \frac{nN}{2}\log(\sigma_w^2 + \sigma_b^2) - \frac{nN}{2}\log(2\pi)$$

$$\frac{\partial \log p(\mathbf{y}|w_{pop})}{\partial w_{pop}} = \frac{2}{2(\sigma_w^2+\sigma_b^2)}\sum_{i,j=1}^{n,N}(y_{ij} - w_{pop}) = 0,\ w_{pop}^{ML} = \frac{1}{nN}\sum_{i,j=1}^{n,N} y_{ij}$$

Bayesian Estimation

Consider a D dimensional Normal random vector $\mathbf{y}$ with distribution $\mathcal{N}(\mathbf{y}|\mu, \Sigma)$ in which Σ is known and for which we wish to infer the mean μ from a set of observations $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y_2}, \ldots, \mathbf{y_N}\}$.
Given the prior distribution $p(\mu) = \mathcal{N}(\mu|\mu_0, \Sigma_0)$, find the posterior distribution of μ and its maximum a posteriori estimation.

Using the Bayesian Rule *i.e.* $p(\mu|\mathbf{Y}) = \frac{p(\mathbf{Y}|\mu)p(\mu)}{p(\mathbf{Y})} \propto p(\mathbf{Y}|\mu)p(\mu)$
Taking the log of both sides we get $\log p(\mu|\mathbf{Y}) = \mathcal{N}(\mu|\mu_N, \Sigma_N)$
$= -\frac{1}{2}(\mu - \mu_0)^T \Sigma_0^{-1}(\mu - \mu_0) - \frac{1}{2}\sum_{n=1}^{N}(\mathbf{y}_n - \mu)^T \Sigma^{-1}(\mathbf{y}_n - \mu) +$ const and
by equating it to $-\frac{1}{2}(\mu - \mu_N)^T \Sigma_N^{-1}(\mu - \mu_N)$, we get

$$\mu_N = (\Sigma_0^{-1} + N\Sigma^{-1})^{-1}(\Sigma_0^{-1}\mu_0 + \Sigma^{-1}\sum_{n=1}^{N}\mathbf{y}_n)$$

$$\Sigma_N = \Sigma_0^{-1} + N\Sigma^{-1}$$

Bayesian Estimation of μ

We maximize the logarithm of the posterior distribution w.r.t μ

$$\log p(\mu|\mathbf{Y}) = -\frac{D}{2}\log(2\pi) - \frac{1}{2}|\Sigma_N| - \frac{1}{2}(\mu - \mu_N)^T \Sigma_N^{-1}(\mu - \mu_N)$$
$$\frac{\partial}{\partial\mu}p(\mu|\mathbf{Y}) = \Sigma_N^{-1}\mu - \Sigma_N^{-1}\mu_N = 0 \longrightarrow \hat{\mu} = \mu_N$$
$$\hat{\mu} = (\Sigma_0^{-1} + N\Sigma^{-1})^{-1}(\Sigma_0^{-1}\mu_0 + \Sigma^{-1}\sum_{n=1}^{N}\mathbf{y}_n)$$

When the prior is flat *i.e.*, $\mu = \mathbf{0}$ and $\Sigma_0 = \infty$, Bayesian estimation reduces to the ML $\rightarrow \hat{\mu} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{y}_n$

Restricted Maximum Likelihood Estimation (ReML)

Considering the *General Linear Model*

$$\mathbf{y} = \mathbf{X}\beta + \epsilon \text{ where } \epsilon \sim \mathcal{N}(\mathbf{0}, \Sigma(\lambda))$$

and $\Sigma(\lambda)$ is an $n \times n$ positive definite covariance matrix that depends on unknown parameters that are organized in a parameter vector λ.

For a simple case where $\Sigma(\lambda) = \sigma^2 \mathbf{I}_n$,
The maximum likelihood estimation for β and σ^2 are

$$\begin{aligned} \hat{\beta} &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \\ \hat{\sigma}^2 &= (\mathbf{y} - \mathbf{X}\hat{\beta})^T(\mathbf{y} - \mathbf{X}\hat{\beta})/n \end{aligned}$$

Although $\hat{\beta}$ is an unbiased estimate of β, the $\hat{\sigma}^2$ is biased *i.e.* $E(\hat{\sigma}^2) = \frac{n-r}{n}\sigma^2$ where r is the rank of $\mathbf{X}$.

Self Study Question

Show that for a simple GLM model $\mathbf{y} = \mathbf{X}\beta + \epsilon$ with $\mathbf{X} = \mathbf{1}_N$, $\beta = \mu$ and $\epsilon = \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_N)$,

$$
\begin{aligned}
E[\hat{\mu}] &= \mu \\
E[\hat{\sigma}^2] &= \frac{n-1}{n}\sigma^2
\end{aligned}
$$

Estimation bias in λ originates from the DoF loss in estimating β.
If we estimated variance components with true mean component values, the estimation would be unbiased.
ReML maximizes a modified likelihood that is free of β instead of the original likelihood as in ML.

Error Contrast : a vector $\mathbf{a}$ orthogonal to columns of $\mathbf{X}$ *i.e.* $\mathbf{a^T X} = \mathbf{0}$.
$\mathbf{A} = [\mathbf{a_1}\ \mathbf{a_2}\ \ldots\ \mathbf{a_{N-r}}]^T$, $\mathbf{A^T X} = 0$ and $E\{\mathbf{A^T y}\} = \mathbf{0}$
As a candidate for $\mathbf{A}$, $\mathbf{I_N} - \mathbf{X}(\mathbf{X^T X})^{-1}\mathbf{X^T} = \mathbf{R}$ has rank $N - r$.
As an alternative, we define a contrast vector
$\mathbf{w} = \mathbf{A}^T\mathbf{y}$ such that $\mathbf{A}\mathbf{A}^T = \mathbf{I_N} - \mathbf{X}(\mathbf{X^T X})^{-1}\mathbf{X^T}$ and $\mathbf{A^T A} = \mathbf{I_N}$.
Then $\mathbf{w} = \mathbf{A}^T\mathbf{y} = \mathbf{A^T A}\mathbf{A}^T\mathbf{y} = \mathbf{A^T}(\mathbf{I_N} - \mathbf{X}(\mathbf{X^T X})^{-1}\mathbf{X^T})\mathbf{y}$.

$\mathbf{w} = \mathbf{A^T R y} = \mathbf{A^T}\epsilon$ is a combination of residuals and is free of β with $p_w(\mathbf{w}|\lambda) \sim \mathcal{N}(\mathbf{0}, \mathbf{A^T \Sigma}(\lambda)\mathbf{A})$.

A modified ML Function

ML function of $n - r$ linearly independent error contrasts $\mathbf{w} = \mathbf{A}^{\mathbf{T}}\mathbf{y}$, $p_w(\mathbf{w}|\lambda) \sim \mathcal{N}(\mathbf{0}, \mathbf{A}^{\mathbf{T}}\mathbf{\Sigma}(\lambda)\mathbf{A})$
replace the full likelihood function, $p(\mathbf{y}|\lambda) \sim \mathcal{N}(\mathbf{X}\beta, \mathbf{\Sigma}(\lambda))$

Defining $\mathbf{G}^{\mathbf{T}} = (\mathbf{X}^{\mathbf{T}}\mathbf{\Sigma}(\lambda)^{-\mathbf{1}}\mathbf{X})^{-\mathbf{1}}\mathbf{X}^{\mathbf{T}}\mathbf{\Sigma}(\lambda)^{-\mathbf{1}}$ and $\hat{\beta} = \mathbf{G}^{\mathbf{T}}\mathbf{y}$ and denoting that $\mathcal{N}_{\hat{\beta}}(\beta, \mathbf{G}^{\mathbf{T}}\mathbf{\Sigma}(\lambda)\mathbf{G}) = \mathcal{N}_{\hat{\beta}}(\beta, (\mathbf{X}^{\mathbf{T}}\mathbf{\Sigma}(\lambda)^{-\mathbf{1}}\mathbf{X})^{-1})$
a restricted log likelihood function is maximized as

$$\mathcal{L}_w = \log p_w(\mathbf{A}^{\mathbf{T}}\mathbf{y}|\lambda)$$

Some Useful Properties

If $\mathbf{z} = f(\mathbf{y})$ then $d\mathbf{z} = \frac{\partial \mathbf{f}(\mathbf{y})}{\partial \mathbf{y}} d\mathbf{y} \rightarrow \mathbf{z} = \mathbf{P}\mathbf{y},\ \frac{\partial \mathbf{P}\mathbf{y}}{\partial \mathbf{y}} = \mathbf{P}^T,\ d\mathbf{z} = \mathbf{P}^T d\mathbf{y}$
$\frac{\partial \mathbf{f}(\mathbf{y})}{\partial \mathbf{y}} = \mathbf{J}$: Jacobian
For functions of random variables

$$|p(\mathbf{z})d\mathbf{z}| = |p(\mathbf{y})d\mathbf{y}| \longleftrightarrow p(\mathbf{z})|\mathbf{J}| = p(\mathbf{y})$$

$p(\mathbf{z}) = p(\mathbf{y})/|\mathbf{J}| \longrightarrow p(\mathbf{z}) = p(\mathbf{y})/|\mathbf{P}^T| = p(\mathbf{y})/|\mathbf{P}|$

If $\mathbf{z} = [\mathbf{A^T}\ \mathbf{G^T}]\mathbf{y}$
$\log\left[\int p_{w,\hat{\beta}}([\mathbf{A^T}\ \mathbf{G^T}]\mathbf{y}|\beta,\lambda)d\beta\right] = \log\left[\frac{1}{\left|[\mathbf{A^T}\ \mathbf{G^T}]\right|}\int p_{\mathbf{y}}(\mathbf{y}|\beta,\lambda)d\beta\right]$

Some Useful Properties

Using the property

$$\begin{vmatrix} \mathbf{A_{11}} & \mathbf{A_{12}} \\ \mathbf{A_{21}} & \mathbf{A_{22}} \end{vmatrix} = |\mathbf{A_{11}}||\mathbf{A_{22}} - \mathbf{A_{21}}\mathbf{A_{11}^{-1}}\mathbf{A_{12}}|$$

$$\left|[\mathbf{A}\ \mathbf{G}]\right| = \left|[\mathbf{A}\ \mathbf{G}]^T\ [\mathbf{A}\ \mathbf{G}]\right|^{1/2}$$

$$\begin{vmatrix} \mathbf{A^T A} & \mathbf{A^T G} \\ \mathbf{G^T A} & \mathbf{G^T G} \end{vmatrix}^{1/2} = |\mathbf{A^T A}|^{1/2}|\mathbf{G^T G} - \mathbf{G^T A}(\mathbf{A^T A})^{-1}\mathbf{A^T G}|^{1/2}$$

$\mathbf{G^T} = (\mathbf{X^T \Sigma^{-1} X})^{-1}\mathbf{X^T \Sigma^{-1}}$ and $\mathbf{A} = \mathbf{I_N} - \mathbf{X}(\mathbf{X^T X})^{-1}\mathbf{X^T}$

$$= |\mathbf{I_N}|^{1/2}\, |\mathbf{G^T G} - \mathbf{G^T}(\mathbf{I_N} - \mathbf{X}(\mathbf{X^T X})^{-1}\mathbf{X^T})\mathbf{G}|^{1/2}$$

$$= |\mathbf{X^T X}|^{-1/2}$$

We define a linear transformation $\mathbf{y} \longrightarrow [\mathbf{w} \;\; \hat{\beta}] = [\mathbf{A^T y} \;\; \mathbf{G^T y}]$

$$cov(\mathbf{w}, \hat{\beta}) = E(\mathbf{w}(\hat{\beta} - \beta)^T) = E(\mathbf{w}\hat{\beta}^T) - E(\mathbf{w}\beta^T) =$$
$$\mathbf{A^T} E(\mathbf{y}\mathbf{y^T})\Sigma^{-1}\mathbf{X}(\mathbf{X^T}\mathbf{\Sigma^{-1}}\mathbf{X})^{-1} - \mathbf{A^T}E(\mathbf{y})\beta^T$$
$$= \mathbf{A^T}(\Sigma + \mathbf{X}\beta\beta^T\mathbf{X^T})\Sigma^{-1}\mathbf{X}(\mathbf{X^T}\mathbf{\Sigma^{-1}}\mathbf{X})^{-1} - \mathbf{A^T}\mathbf{X}\beta\beta^T = 0$$

$$p(\mathbf{y}|\lambda) = p(\mathbf{w}, \hat{\beta}|\lambda, \beta)|\mathbf{J}| = p(\mathbf{w}|\hat{\beta}, \lambda, \beta)p(\hat{\beta}|\lambda, \beta)|\mathbf{J}|$$

Since $\mathbf{w}$ and $\hat{\beta}$ are gaussian and they are uncorrelated, they must be independent as well so that $p(\mathbf{y}|\lambda) = p(\mathbf{w}|\lambda, \beta)p(\hat{\beta}|\lambda, \beta)|\mathbf{J}|$
$p(\mathbf{w}|\lambda) = p(\mathbf{y}|\lambda)/(p(\hat{\beta}|\lambda, \beta)|\mathbf{J}|)$

$$(\mathbf{y} - \mathbf{X}\beta)^T\Sigma(\lambda)^{-1}(\mathbf{y} - \mathbf{X}\beta) =$$
$$(\mathbf{y} - \mathbf{X}\hat{\beta})^T\Sigma(\lambda)^{-1}(\mathbf{y} - \mathbf{X}\hat{\beta}) + (\beta - \hat{\beta})^T(\mathbf{X^T}\mathbf{\Sigma}(\lambda)^{-1}\mathbf{X})(\beta - \hat{\beta})$$

$$\mathcal{L}_w(\mathbf{A^T y}|\lambda) = \log p(\mathbf{w}|\lambda)$$

$$\log\left[\frac{1}{(2\pi)^{N/2}|\Sigma(\lambda)|^{1/2}}e^{(-\frac{1}{2}(\mathbf{y}-\mathbf{X}\hat{\beta})^T\Sigma(\lambda)^{-1}(\mathbf{y}-\mathbf{X}\hat{\beta}))}e^{-\frac{1}{2}(\beta-\hat{\beta})^T(\mathbf{X^T\Sigma^{-1}}(\lambda)\mathbf{X})(\beta-\hat{\beta})}\Big/\left(|\mathbf{X^TX}|^{-\frac{1}{2}}\frac{1}{(2\pi)^{r/2}|\mathbf{X^T}\Sigma(\lambda)\mathbf{X}|^{-1/2}}e^{-\frac{1}{2}(\beta-\hat{\beta})^T(\mathbf{X^T\Sigma^{-1}}(\lambda)\mathbf{X})(\beta-\hat{\beta})}\right)\right]$$

$$\log\left[|\mathbf{X^TX}|^{\frac{1}{2}}\frac{1}{(2\pi)^{N/2}|\Sigma(\lambda)|^{1/2}}\frac{1}{(2\pi)^{-r/2}}\frac{1}{|\mathbf{X^T\Sigma^{-1}}(\lambda)\mathbf{X}|^{1/2}}e^{(-\frac{1}{2}(\mathbf{y}-\mathbf{X}\hat{\beta})^T\Sigma^{-1}(\lambda)(\mathbf{y}-\mathbf{X}\hat{\beta}))}\right]$$

This convenient expression can be maximized as the restricted log-likelihood $\mathcal{L}_w(\mathbf{A^T y}|\lambda)$ w.r.t. variance components λ to obtain an unbiased estimate for the covariance matrix $\Sigma(\lambda)$ and the corresponding regression coefficients $\hat{\beta}$.

If we define the covariance matrix $\Sigma(\lambda) = \sum_{i=1}^{q} \lambda_i \mathbf{Q}_i$ in terms of covariance structures $\mathbf{Q}_i$, we can use the *ReML* to estimate the λ_i

$\mathcal{L}_w(\mathbf{A^T y}|\lambda) =$

$\frac{1}{2}\log|\Sigma(\lambda)^{-1}| - \frac{1}{2}\log|\mathbf{X^T}\Sigma^{-1}(\lambda)\mathbf{X}| - \frac{1}{2}(\mathbf{y}-\mathbf{X}\hat{\beta})^T\Sigma^{-1}(\lambda)(\mathbf{y}-\mathbf{X}\hat{\beta}) + Const$

Iterative Optimization : Fisher scoring

Update λ parameters using the gradient and the Hessian as

$\nabla_{\mathcal{L}_w}(\lambda) = \frac{\partial}{\partial\lambda}\mathcal{L}_w(\lambda)$ and $\mathbf{H}_{ij} = -E\left[\frac{\partial^2}{\partial\lambda_i\partial\lambda_j}\mathcal{L}_w(\lambda)\right]$

Property

If $\mathbf{U} = f(\mathbf{X})$ then $\frac{\partial g(\mathbf{U})}{\partial \mathbf{X}} = \frac{\partial g(f(\mathbf{X}))}{\partial \mathbf{X}}$

Using the chain rule $\frac{\partial g(\mathbf{U})}{\partial \mathbf{X}} = \frac{\partial g(\mathbf{U})}{\partial x_{ij}} = \sum_{k=1}^{M}\sum_{l=1}^{N}\frac{\partial g(\mathbf{U})}{\partial u_{kl}}\frac{\partial u_{kl}}{\partial x_{ij}}$

or in matrix form $\frac{\partial g(\mathbf{U})}{\partial X_{ij}} = Tr\left[\left(\frac{\partial g(\mathbf{U})}{\partial \mathbf{U}}\right)^T \frac{\partial \mathbf{U}}{\partial X_{ij}}\right]$

If $F(\mathbf{X})$ is a differentiable function of each of the elements of $\mathbf{X}$ then $\frac{\partial Tr(F(\mathbf{X}))}{\partial \mathbf{X}} = f(\mathbf{X})^T$ where $f(.)$ is the scalar derivative of $F(.)$.

$\mathcal{L}_w(\lambda|\mathbf{A^T y}) =$

$$\frac{1}{2}\log|\Sigma(\lambda)^{-1}| - \frac{1}{2}\log|\mathbf{X^T}\Sigma^{-1}(\lambda)\mathbf{X}| - \frac{1}{2}Tr[\Sigma^{-1}(\lambda)(\mathbf{y}-\mathbf{X}\hat{\beta})(\mathbf{y}-\mathbf{X}\hat{\beta})^T] + Const$$

Matrix Properties for Differentiation

$$\frac{\partial(F(\mathbf{X}))}{\partial\mathbf{X}} = f(\mathbf{X})^T, \ \frac{\partial g(\mathbf{U})}{\partial X_{ij}} = Tr\left[(\frac{\partial g(\mathbf{U})}{\partial\mathbf{U}})^T\frac{\partial\mathbf{U}}{\partial X_{ij}}\right]$$

$$X_{ij}\to\lambda_i, \ \mathbf{g}\to\mathcal{L}_w \text{ and } \Sigma(\lambda) = \sum_{i=1}^{q}\lambda_i Q_i$$

$$\frac{\partial}{\partial\mathbf{Y}}\log|\mathbf{Y}| = \mathbf{Y}^{-T}, \ \frac{\partial\mathbf{Y}^{-1}}{\partial x} = -\mathbf{Y}^{-1}\frac{\partial\mathbf{Y}}{\partial x}\mathbf{Y}^{-1}, \ \frac{\partial Tr[\mathbf{YA}]}{\partial\mathbf{Y}} = \mathbf{A^T}, \ Tr[\mathbf{AB}] = Tr[\mathbf{BA}]$$

$$\frac{\partial\mathcal{L}_w}{\partial\lambda_i} =$$

$$\frac{\partial}{\partial\lambda_i}(\frac{1}{2}\log|\Sigma^{-1}|) - \frac{1}{2}(\frac{\partial}{\partial\lambda_i}(\log|\mathbf{X^T}\Sigma^{-1}\mathbf{X}|) - \frac{\partial}{\partial\lambda_i}Tr(\frac{1}{2}\Sigma^{-1}(\mathbf{y}-\mathbf{X}\hat{\beta})(\mathbf{y}-\mathbf{X}\hat{\beta})^T))$$

1^{st} **term:** $\frac{\partial\log|\Sigma^{-1}|}{\partial\lambda_i} = Tr[(\frac{\partial\log(\Sigma^{-1})}{\partial\Sigma^{-1}})^T\frac{\partial\Sigma^{-1}}{\partial\lambda_i}] = -Tr[\Sigma\Sigma^{-1}\mathbf{Q}_i\Sigma^{-1}] = -Tr[\mathbf{Q}_i\Sigma^{-1}]$

2^{nd} **term:** $\frac{\partial}{\partial\lambda_i}\log|\mathbf{X^T}\Sigma^{-1}\mathbf{X}| = Tr[(\frac{\partial\log|\mathbf{X^T}\Sigma^{-1}\mathbf{X}|}{\partial(\mathbf{X^T}\Sigma^{-1}\mathbf{X})})^T\frac{\partial\mathbf{X^T}\Sigma^{-1}\mathbf{X}}{\partial\lambda_i}]$
$= -Tr[(\mathbf{X^T}\Sigma^{-1}\mathbf{X})^{-1}\mathbf{X^T}\Sigma^{-1}\mathbf{Q}_i\Sigma^{-1}\mathbf{X}]$

3^{rd} **term:** $\frac{\partial}{\partial\lambda_i}Tr[\Sigma^{-1}\overbrace{(\mathbf{y}-\mathbf{X}\hat{\beta})(\mathbf{y}-\mathbf{X}\hat{\beta})^T}^{A}] = Tr[(\frac{\partial Tr[\Sigma^{-1}\mathbf{A}]}{\partial\Sigma^{-1}})^T\frac{\partial\Sigma^{-1}}{\partial\lambda_i}] =$
$-(\mathbf{y}-\mathbf{X}\hat{\beta})(\mathbf{y}-\mathbf{X}\hat{\beta})^T\Sigma^{-1}\mathbf{Q}_i\Sigma^{-1}$

$$\frac{\partial \mathcal{L}_w(\mathbf{A}^\mathbf{T}\mathbf{y}|\lambda)}{\partial \lambda_i} =$$

$$\frac{1}{2} Tr[-\mathbf{Q}_i\Sigma^{-1} + (\mathbf{X}^\mathbf{T}\Sigma^{-1}\mathbf{X})^{-1}\mathbf{X}^\mathbf{T}\Sigma^{-1}\mathbf{Q}_i\Sigma^{-1}\mathbf{X} + (\mathbf{y} - \mathbf{X}\hat{\beta})(\mathbf{y} - \mathbf{X}\hat{\beta})^T\Sigma^{-1}\mathbf{Q}_i\Sigma^{-1}]$$

Self-Study Problem

$$\frac{\partial \mathcal{L}_w}{\partial \lambda} = \frac{1}{2} Tr[-\mathbf{Q}_i\Sigma^{-1} + (\mathbf{X}^\mathbf{T}\Sigma^{-1}\mathbf{X})^{-1}\mathbf{X}^\mathbf{T}\Sigma^{-1}\mathbf{Q}_i\Sigma^{-1}\mathbf{X} + (\mathbf{y} - \mathbf{X}\hat{\beta})(\mathbf{y} - \mathbf{X}\hat{\beta})^T\Sigma^{-1}\mathbf{Q}_i\Sigma^{-1}]$$

If $\mathbf{P} = \Sigma^{-1} - \Sigma^{-1}\mathbf{X}(\mathbf{X}^\mathbf{T}\boldsymbol{\Sigma}^{-1}\mathbf{X})^{-1}\mathbf{X}^\mathbf{T}\Sigma^{-1}$

it can be shown that

$$\frac{\partial \mathcal{L}_w}{\partial \lambda} = \nabla_{\mathcal{L}_w}(\lambda) = -\frac{1}{2} Tr[\mathbf{P}\mathbf{Q}_i] + \frac{1}{2} Tr[\mathbf{P}\mathbf{y}\mathbf{y}^\mathbf{T}\mathbf{P}^\mathbf{T}\mathbf{Q}_i]$$

Computing the Hessian $\mathbf{H}_{ij} = -E\left[\frac{\partial^2}{\partial\lambda_i\partial\lambda_j}\mathcal{L}_w(\lambda)\right] = -E\left[\frac{\partial}{\partial\lambda_j}\mathcal{L}_w(\lambda_i)\right]$

1. $\frac{\partial}{\partial\lambda_j}Tr[\Sigma^{-1}\mathbf{Q}_i] = Tr[(\frac{\partial Tr[\Sigma^{-1}\mathbf{Q}_i]}{\partial\Sigma^{-1}})^T\frac{\partial\Sigma^{-1}}{\partial\lambda_j}] = -Tr[\mathbf{Q}_i^T(\Sigma^{-1}\mathbf{Q}_j\Sigma^{-1})]$
$= -Tr[\Sigma^{-1}\mathbf{Q}_i\Sigma^{-1}\mathbf{Q}_j]$

2. $\frac{\partial}{\partial\lambda_j}Tr[(\mathbf{X}^\mathbf{T}\Sigma^{-1}\mathbf{X})^{-1}\mathbf{X}^\mathbf{T}\Sigma^{-1}\mathbf{Q}_i\Sigma^{-1}\mathbf{X}] = \frac{\partial}{\partial\lambda_j}Tr[\Sigma^{-1}\mathbf{X}(\mathbf{X}^\mathbf{T}\Sigma^{-1}\mathbf{X})^{-1}\mathbf{X}^\mathbf{T}\Sigma^{-1}\mathbf{Q}_i]$
$= \frac{\partial}{\partial\lambda_j}Tr[\Sigma^{-1}\underbrace{\mathbf{X}(\mathbf{X}^\mathbf{T}\boldsymbol{\Sigma}^{-1}\mathbf{X})^{-1}\mathbf{X}^\mathbf{T}\Sigma^{-1}\mathbf{Q}_i}_{\mathbf{A}}] = Tr[(\frac{\partial Tr[\Sigma^{-1}\mathbf{A}]}{\partial\Sigma^{-1}})^T\frac{\partial\Sigma^{-1}}{\partial\lambda_j}]$
$= -Tr[\mathbf{A}\Sigma^{-1}\mathbf{Q_j}\Sigma^{-1}] = -Tr[\mathbf{X}(\mathbf{X}^\mathbf{T}\boldsymbol{\Sigma}^{-1}\mathbf{X})^{-1}\mathbf{X}^\mathbf{T}\Sigma^{-1}\mathbf{Q}_i\Sigma^{-1}\mathbf{Q_j}\Sigma^{-1}]$

3. $\frac{\partial}{\partial\lambda_j}Tr[(\mathbf{X}^\mathbf{T}\Sigma^{-1}\mathbf{X})^{-1}\mathbf{X}^\mathbf{T}\Sigma^{-1}\mathbf{Q}_i\Sigma^{-1}\mathbf{X}] = \frac{\partial}{\partial\lambda_j}Tr[\mathbf{Q}_i\Sigma^{-1}\mathbf{X}(\mathbf{X}^\mathbf{T}\Sigma^{-1}\mathbf{X})^{-1}\mathbf{X}^\mathbf{T}\Sigma^{-1}]$
$= \frac{\partial}{\partial\lambda_j}Tr[\underbrace{\mathbf{Q}_i\Sigma^{-1}\mathbf{X}(\mathbf{X}^\mathbf{T}\boldsymbol{\Sigma}^{-1}\mathbf{X})^{-1}\mathbf{X}^\mathbf{T}}_{\mathbf{B}}\Sigma^{-1}] = Tr[(\frac{\partial Tr[\mathbf{B}\Sigma^{-1}]}{\partial\Sigma^{-1}})^T\frac{\partial\Sigma^{-1}}{\partial\lambda_j}]$
$= -Tr[\mathbf{B}\Sigma^{-1}\mathbf{Q}_j\Sigma^{-1}] = -Tr[\mathbf{Q}_i\Sigma^{-1}\mathbf{X}(\mathbf{X}^\mathbf{T}\boldsymbol{\Sigma}^{-1}\mathbf{X})^{-1}\mathbf{X}^\mathbf{T}\Sigma^{-1}\mathbf{Q}_j\Sigma^{-1}] =$

4. $\frac{\partial}{\partial\lambda_j}Tr[\overbrace{\mathbf{I_N}}^{A}(\mathbf{X}^\mathbf{T}\boldsymbol{\Sigma}^{-1}\mathbf{X})^{-1}\overbrace{\mathbf{X}^\mathbf{T}\Sigma^{-1}\mathbf{Q}_i\Sigma^{-1}\mathbf{X}}^{C}]$ $\quad \frac{\partial Tr[\mathbf{AY}^{-1}\mathbf{C}]}{\partial\mathbf{Y}} = -\mathbf{Y}^{-\mathbf{T}}\mathbf{A}^\mathbf{T}\mathbf{C}^\mathbf{T}\mathbf{Y}^{-T}$,
$= Tr[\frac{\partial Tr(\mathbf{I_N}(\mathbf{X}^\mathbf{T}\Sigma^{-1}\mathbf{X})^{-1}\mathbf{C}}{\partial(\mathbf{X}^\mathbf{T}\Sigma^{-1}\mathbf{X})})^T\frac{\partial\mathbf{X}^\mathbf{T}\boldsymbol{\Sigma}^{-1}\mathbf{X}}{\partial\lambda_j}]$
$= Tr[(\mathbf{X}^\mathbf{T}\Sigma^{-1}\mathbf{X})^{-1}\mathbf{X}^\mathbf{T}\Sigma^{-1}\mathbf{Q}_i\Sigma^{-1}\mathbf{X}(\mathbf{X}^\mathbf{T}\Sigma^{-1}\mathbf{X})^{-1}(\mathbf{X}^\mathbf{T}\Sigma^{-1}\mathbf{Q}_j\Sigma^{-1}\mathbf{X})]$

5. $\frac{\partial}{\partial \lambda_j} Tr[(\mathbf{y} - \mathbf{X}\hat{\beta})(\mathbf{y} - \mathbf{X}\hat{\beta})^T \Sigma^{-1} \mathbf{Q}_i \Sigma^{-1}]$

Question

In fact, we do not need to determine the above term 5 to compute **H**. Explain why not?

Self-Study Problem

Remembering that $\mathbf{P} = \Sigma^{-1} - \Sigma^{-1}\mathbf{X}(\mathbf{X^T}\boldsymbol{\Sigma}^{-1}\mathbf{X})^{-1}\mathbf{X^T}\Sigma^{-1}$,

show that the Hessian $\mathbf{H}_{ij} = -E[\frac{\partial^2}{\partial\lambda_i\partial\lambda_j}\mathcal{L}_w(\lambda)] = -E[\frac{\partial}{\partial\lambda_j}\nabla_{\mathcal{L}_w}]$

$$= -\frac{1}{2}Tr[(\Sigma^{-1}\mathbf{Q}_i^T\Sigma^{-1}\mathbf{Q}_j)]$$
$$+\frac{1}{2}Tr[\Sigma^{-1}\mathbf{X}(\mathbf{X^T}\boldsymbol{\Sigma}^{-1}\mathbf{X})^{-1}\mathbf{X^T}\Sigma^{-1}\mathbf{Q}_i\Sigma^{-1}\mathbf{Q}_j]$$
$$+\frac{1}{2}Tr[\Sigma^{-1}\mathbf{Q}_i\Sigma^{-1}\mathbf{X}(\mathbf{X^T}\boldsymbol{\Sigma}^{-1}\mathbf{X})^{-1}\mathbf{X^T}\Sigma^{-1}\mathbf{Q}_j]$$
$$-\frac{1}{2}Tr[\Sigma^{-1}\mathbf{X}(\mathbf{X^T}\Sigma^{-1}\mathbf{X})^{-1}\mathbf{X^T}\Sigma^{-1}\mathbf{Q}_i\boldsymbol{\Sigma}^{-1}\mathbf{X}(\mathbf{X^T}\Sigma^{-1}\mathbf{X})^{-1}\mathbf{X^T}\Sigma^{-1}\mathbf{Q}_j]$$

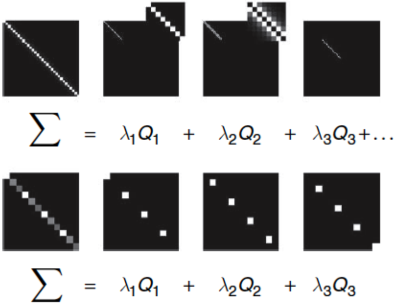
can be simplified as

$$\mathbf{H}_{ij} = \frac{1}{2}Tr[\mathbf{P}\mathbf{Q}_i\mathbf{P}\mathbf{Q}_j]$$

Estimating the Covariance Components of $\Sigma = \sum_i \lambda_i \mathbf{Q}_i$

$$\mathbf{y} = \mathbf{X}\beta + \mathbf{e}$$

with $\mathbf{e} \sim \mathcal{N}(0, \Sigma)$ and $\mathbf{y} \sim \mathcal{N}(\mathbf{X}\beta, \Sigma)$



AR(1) Model

3 measures taken
from a group of subjects.

The **ReML** Algorithm can be used to estimate the hyperparameters λ_i, $\Sigma = \sum_i \lambda_i \mathbf{Q}_i$.

$$\mathbf{y} = \mathbf{X}\beta + \mathbf{e}$$

with $\mathbf{e} \sim \mathcal{N}(0, \Sigma)$ and

ReML Algorithm

Initialize λ

Compute $\Sigma = \sum_i \lambda_i \mathbf{Q}_i$ and $\mathbf{P} = \Sigma^{-1} - \Sigma^{-1}\mathbf{X}(\mathbf{X^T}\mathbf{\Sigma}^{-1}\mathbf{X})^{-1}\mathbf{X^T}\Sigma^{-1}$

Compute the gradient and Hessian

$\mathbf{g} = \frac{\partial \mathcal{L}_w}{\partial \lambda_i} = -\frac{1}{2} Tr[\mathbf{P}\mathbf{Q}_i] + \frac{1}{2} Tr[\mathbf{P}\mathbf{y}\mathbf{y}^\mathbf{T}\mathbf{P}^\mathbf{T}\mathbf{Q}_i]$

$\mathbf{H} = -\frac{\partial^2 \mathcal{L}_w}{\partial \lambda_i \lambda_j} = \frac{1}{2} Tr[\mathbf{P}\mathbf{Q}_i\mathbf{P}\mathbf{Q}_j]$

Update λ until convergence

$\lambda \longrightarrow \lambda + \mathbf{H}^{-1}\mathbf{g}$

Autoregressive model for colored noise

AR(1) model for colored noise $w[n] = aw[n-1] + v[n]$ where $v[n] \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

$$\begin{bmatrix} w[0] \\ w[1] \\ w[2] \\ \vdots \\ w[N-1] \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ a & 0 & 0 & 0 & \dots & 0 \\ 0 & a & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & a & \dots & 0 \end{bmatrix} \begin{bmatrix} w[0] \\ w[1] \\ w[2] \\ \vdots \\ w[N-1] \end{bmatrix} + \begin{bmatrix} v[0] \\ v[1] \\ v[2] \\ \vdots \\ v[N-1] \end{bmatrix}$$

$\mathbf{W} = \mathbf{AW} + \mathbf{V} \longrightarrow \mathbf{W}(\mathbf{I} - \mathbf{A}) = \mathbf{V}$

$Cov[\mathbf{W}] = E[\mathbf{WW}^T] = E[(\mathbf{I} - \mathbf{A})^{-1}\mathbf{VV}^T(\mathbf{I} - \mathbf{A})^{-T}] = (\mathbf{I} - \mathbf{A})^{-1}E[\mathbf{VV}^T](\mathbf{I} - \mathbf{A})^{-T}$

$Cov[\mathbf{W}] = (\mathbf{I} - \mathbf{A})^{-1}(\mathbf{I} - \mathbf{A})^{-T}$

Once $\mathbf{W}$ is estimated by ReML, the first off-diagonal values of $\mathbf{I} - \mathbf{A}$ can be used as an estimate of AR(1) coefficient a.

A ReML Simulation for Error Covariance Estimation

```
% Error Covariance  Estimation for a GLM y=Xbeta + e using ReML
M = 100; % number of time points
% Hyperparameters to be estimated Error Covariance
lambda = [0.25 0.1 ];
Q1 = diag(ones(M,1)); % Diagonal Noise Component (white)
Q2 = diag(ones(M-1,1),1) +  diag(ones(M-1,1),-1) ;%off-diagonal component
Ce = Q1*lambda(1) + Q2*lambda(2) ; % Error Covariance Matrix
w = rand(M,1); % generate white noise
w = w-mean(w); % correct for zero mean
w= w/std(w); % normalize for unit variance
F = chol(Ce) % Cholesky decomposition of Ce=F' * F for color filtering
e = F*w; %  Filtering the white noise to obtain colored error
X(:,1) = [zeros(M/4,1); ones(M/2,1); zeros(M/4,1) ]; X(:,2) = ones(M,1); % Design Matrix
beta = [1 ; 1]; Y = X*beta + e; %Signal + Error
% Solution using ReML
lambda_1 = [1 ;1];   % Initialize hyperparameters
iterNum = 0;
while iterNum < 10,
C_1e = Q1*lambda_1(1) + Q2*lambda_1(2) ;
C_1e1 =  pinv(C_1e);
C_t = pinv(X' * C_1e1 * X) ;
P = C_1e1*( eye(M,M) - X*C_t*X'*C_1e1 );
g(1,1) = 0.5* (-trace (P*Q1) + Y'*P*Q1*P*Y ); g(2,1) =  0.5* (-trace (P*Q2) + Y'*P*Q2*P*Y );
H(1,1) = 0.5 * trace (P*Q1*P*Q1);H(1,2) = 0.5 * trace (P*Q1*P*Q2);
H(2,1) = 0.5 * trace (P*Q2*P*Q1);H(2,2) = 0.5 * trace (P*Q2*P*Q2);
Delta = pinv(H) * g; lambda_1 = lambda_1 + Delta
iterNum = iterNum +1;
end;
```

Self-Study Question: Estimate AR(1) coefficient using ReML Algorithm.

```
% ReML Estimation of Noise Covariance for a simple GLM model with AR noise
clear all
N = 100; % number of time points
M = 1; % number of realizations
E0 = randn(N,M); % Gaussian White Noise
E0 = (E0 - mean(E0)*ones(1,M))./(std(E0*ones(1,M)));
a = [ 0.8]; % AR Coefficient with  model x(n) = ax(n-1) + w(n)
I_A = eye(N,N);
I_A = I_A - full(spdiags(ones(N,1)*a,-1:-1:-length(a),N,N)) ;
% whitening filter [I-A]^{-1} X = E
I_A_1 = pinv(I_A) ; % Coloring Filter
%S= I_A_1+I_A_1'-2*diag(diag(I_A_1)); % KK' colored part of Q
S = I_A_1*I_A_1';
E = I_A_1 * E0; % Colored Noise based on AR(1) model
X= [ [zeros(1,N/4) ones(1,N/4) zeros(1,N/4) ones(1,N/4)] ;ones(1,N) ]'; % design matrix using a regresso
beta = [2 1]';
Y = (X * beta)* ones(1,M) + E ; % Signal + Colored Noise generated by AR(1) model
Q{1} = eye(N,N);
for i=2:6,
% you can play with the number of colored component sub matrices but a few seems to be enough
Q{i} = full(spdiags(1*ones(N,2),[ -i+1 i-1 ],N,N));;
end;

% Write the code for RemL Algorithm to solve the hyperparameters, lambda for Q_i,
%construct the  estimated error covariance matrix  from which you can  extract the AR(1) parameter.
```

Solution of AR(1) problem using ReML

```
lambda_1 = ones(length(Q),1); % Initialize hyperparameters
iterNum = 0; lambda_p=0; Error =1;
while Error>1e-2,
C_1e = zeros(N,N);
for j =1:length(Q), C_1e = Q{j}*lambda_1(j) + C_1e ; end;
C_1e1 = pinv(C_1e);
C_t = pinv(X' * C_1e1 * X) ;
P = C_1e1*( eye(N) - X*C_t*X'*C_1e1 );
for p = 1:length(Q)
  g(p,1) = 0.5* (-trace(P*Q{p}) + Y'*P*Q{p}*P*Y );
  for q=1:length(Q),
    H(p,q) = 0.5 * trace (P*Q{p}*P*Q{q});
  end;
end;
Delta = pinv(H) * g;
lambda_1 = lambda_1 + Delta;
Error = norm(lambda_p-lambda_1);
lambda(iterNum+1)= Error;
lambda_p = lambda_1;
iterNum = iterNum +1;
end;
C_h = zeros(N,N); for j =1:length(Q), C_h = Q{j}*lambda_1(j) + C_h ; end;
V0 = chol(C_h);  %  V0 = I_A_1'
V0= inv(V0');
% Alternatively, you can use SPM_RML as
%  [V,h] = spm_reml(N*Y*Y',X,Q,N); V0 = chol(V); V0= inv(V0');
% The AR parameter is in the first lower diagonal region of V0
```