Equations for AFNI program 3dLSS.   RWCox – Dec 2011.

Rank 1 Update to Pseudo Inverse

Define $\underline{\underline{P}} = (\underline{\underline{A}}^T \underline{\underline{A}})^{-1} \underline{\underline{A}}^T$ = pseudo inverse of $\underline{\underline{A}} = \underline{\underline{A}}^\psi \Rightarrow \underline{\underline{P}}\,\underline{\underline{A}} = \underline{\underline{I}}$

$\underline{\underline{Q}} = \underline{\underline{I}} - \underline{\underline{A}}\,\underline{\underline{P}}$ = orthogonal symmetric projection onto nullspace of A

$$\left[\underline{\underline{Q}}\,\underline{\underline{A}} = \underline{\underline{0}}\right]$$

Augment $\underline{\underline{A}}$:

$$N\left\{ \left[\overbrace{\underline{\underline{A}}}^{M} \;\middle|\; \overbrace{\underline{c}}^{1} \right] \right. \equiv \underline{\underline{\tilde{A}}}$$

Define $\underline{v} = \underline{\underline{Q}}\,\underline{c} \in \mathbb{R}^N$ (note $\underline{v}^T \underline{\underline{A}} = \underline{c}^T \underline{\underline{Q}}\,\underline{\underline{A}} = \underline{0}^T$)

Then

$$\underline{\underline{\tilde{A}}}^\psi = \begin{array}{r} M\{ \\ \\ 1\{ \end{array} \underbrace{\begin{bmatrix} \underline{\underline{P}}\left(\underline{\underline{I}} - \frac{1}{\underline{c}^T\underline{v}}\,\underline{c}\,\underline{v}^T\right) \\ \hline \frac{1}{\underline{c}^T\underline{v}}\,v^T \end{bmatrix}}_{N} = \begin{bmatrix} P_{ij} - \frac{1}{\underline{c}^T\underline{v}}(\underline{\underline{P}}\,\underline{c})_i\, v_j \\ \hline \left(\frac{1}{c^T v}\right) v_j \end{bmatrix} \quad \begin{array}{l} i = 0 \ldots M-1 \\ j = 0 \ldots N-1 \end{array}$$

Proof

$$\begin{bmatrix} \underline{\underline{P}} - \frac{1}{\underline{c}^T\underline{v}}\,\underline{\underline{P}}\,\underline{c}\,\underline{v}^T \\ \hline \frac{1}{\underline{c}^T\underline{v}}\,\underline{v}^T \end{bmatrix} \left[\underline{\underline{A}} \;\middle|\; \underline{c}\right] = \left[\begin{array}{c|c} \underline{\underline{P}}\,\underline{\underline{A}} - \frac{1}{\underline{c}^T\underline{v}}\,\underline{\underline{P}}\,\underline{c}\,\underline{v}^T\underline{\underline{A}} & \underline{\underline{P}}\,\underline{c} - \frac{1}{\underline{c}^T\underline{v}}\,\underline{\underline{P}}\,\underline{c}\,\underline{v}^T\underline{c} \\ \hline \frac{1}{\underline{c}^T\underline{v}}\,\underline{v}^T\underline{\underline{A}} & \frac{1}{\underline{c}^T\underline{v}}\,\underline{v}^T\underline{c} \end{array}\right]$$

$$= \left[\begin{array}{c|c} \underline{\underline{I}} & \underline{0} \\ \hline \underline{0}^T & 1 \end{array}\right]$$

LSS solution via Rank-1 update

Want to solve

$$\underline{z} \approx \underline{\underline{X}}\,\underline{\beta} + \gamma\underline{b} + \delta\underline{c} = \underline{\underline{X}}\,\underline{\beta} + \gamma(\underline{b} - \underline{c}) + (\gamma+\delta)\,\underline{c}$$

Application has $\underline{b} = \sum_k \underline{c}^{(k)}$, but that doesn't matter here. We want to get $\gamma+\delta$ for each $\underline{c}^{(k)}$; $\underline{\beta}$ and γ and δ are by themselves unimportant.

Let $\underline{\underline{A}} = [\,\underline{\underline{X}} \;\; \underline{b}\,]$ = fixed matrix and $\tilde{\underline{\underline{A}}} = [\,\underline{\underline{A}} \,\vdots\, \underline{c}\,]$ = augmented matrix for 1 subproblem.

From previous page, we can compute γ and δ from the last 2 rows of $\tilde{\underline{\underline{A}}}^{+}$, which are

for γ: $\quad P_{m-1,j} - \frac{1}{\underline{c}^T\underline{v}}(\underline{\underline{P}}\,\underline{c})_{m-1} v_j \qquad$ for $j = 0..N-1$

for δ: $\quad \frac{1}{\underline{c}^T\underline{v}} v_j \qquad$ for $j=0..N-1$

Summing these gives the vector $\underline{s}$ which, when dotted into $\underline{z}$, gives the estimate of $\gamma+\delta$:

$$s_j = \underbrace{P_{m-1,j}}_{\text{last row of } \underline{\underline{P}}} + v_j\Big(1 - \underbrace{(\underline{\underline{P}}\,\underline{c})_{m-1}}_{\text{last element of } \underline{\underline{P}}\underline{c}}\Big) / \underline{c}^T\underline{v} \qquad \text{for } j=0..N-1$$

So, after computing $\underline{\underline{P}}$ & $\underline{\underline{Q}}$ (once for the fixed $\underline{\underline{A}}$), we compute $\underline{s}$ for each $\underline{c}$ by defining N-vector $\underline{p}$ = last row of $\underline{\underline{P}} = \{P_{m-1,j}\}$, then

$$\underline{v} = \underline{\underline{Q}}\,\underline{c}$$

$$\underline{s} = \underline{p} + \underbrace{\frac{(1 - \underline{p}^T\underline{c})}{\underline{c}^T\underline{v}}}_{\text{scalar}}\,\underline{v}$$

This is what the LSS_setup() function does.