Supplementary appendix to "Massively parallel nonparametric regression, with an application to developmental brain mapping," published in the *Journal of Computational and Graphical Statistics*

Philip T. Reiss[1,2], Lei Huang[3], Yin-Hsiu Chen[1], Lan Huo[1], Thaddeus Tarpey[4], and Maarten Mennes[5]

[1]Department of Child and Adolescent Psychiatry, New York University
[2]Nathan S. Kline Institute for Psychiatric Research
[3]Department of Biostatistics, Johns Hopkins University
[4]Department of Mathematics and Statistics, Wright State University
[5]Department of Cognitive Neuroscience, Radboud University Nijmegen Medical Centre

September 10, 2012

1 Diagonalization

The following reparametrization obviates the need to compute and invert the $q \times q$ matrix $\boldsymbol{V}_\gamma$ for $\gamma = \gamma_{(1)}, \ldots, \gamma_{(G)}$. Given the singular value decomposition $\boldsymbol{Z}\boldsymbol{Z}^T = \boldsymbol{U}_*\boldsymbol{D}_*\boldsymbol{U}_*^T$, where $\boldsymbol{U}_*$ is an $n \times n$ orthogonal matrix and $\boldsymbol{D}_* = \text{Diag}\{d_1, \ldots, d_n\}$, we have $\boldsymbol{V}_\gamma^{-1} = \boldsymbol{U}_*\boldsymbol{\Lambda}_\gamma\boldsymbol{U}_*^T$ where $\boldsymbol{\Lambda}_\gamma = \text{Diag}\left\{\frac{1}{1+\gamma d_1}, \ldots, \frac{1}{1+\gamma d_n}\right\}$. Setting $\tilde{\boldsymbol{y}}_\ell = \boldsymbol{U}_*^T\boldsymbol{y}_\ell$ and $\tilde{\boldsymbol{X}} = \boldsymbol{U}_*^T\boldsymbol{X}$ and noting that $d_{q+1} = \ldots = d_n = 0$, the quantities determining $\ell_R(\gamma; \boldsymbol{y}_\ell)$ simplify to

$$\begin{aligned}
\boldsymbol{y}_\ell^T\boldsymbol{M}_\gamma\boldsymbol{y}_\ell &= \tilde{\boldsymbol{y}}_\ell^T\left[\boldsymbol{\Lambda}_\gamma - \boldsymbol{\Lambda}_\gamma\tilde{\boldsymbol{X}}(\tilde{\boldsymbol{X}}^T\boldsymbol{\Lambda}_\gamma\tilde{\boldsymbol{X}})^{-1}\tilde{\boldsymbol{X}}^T\boldsymbol{\Lambda}_\gamma\right]\tilde{\boldsymbol{y}}_\ell \\
\text{and } h_\gamma &= -\sum_{i=1}^{q}\log(1+\gamma d_i) - \log|\tilde{\boldsymbol{X}}^T\boldsymbol{\Lambda}_\gamma\tilde{\boldsymbol{X}}|.
\end{aligned}$$

2 Pointwise interval estimation

The penalized spline model for location ℓ implies that $\boldsymbol{y}_\ell|\boldsymbol{\theta}_\ell \sim N(\boldsymbol{B}\boldsymbol{\theta}_\ell, \sigma_\ell^2 \boldsymbol{I}_n)$. Following Silverman (1985), roughness penalization can be viewed as imposing a prior density proportional to $\exp(-\frac{\lambda_\ell}{2}\boldsymbol{\theta}_\ell^T \boldsymbol{P}\boldsymbol{\theta}_\ell/\sigma_\ell^2)$ on $\boldsymbol{\theta}_\ell$, leading to the posterior distribution $\boldsymbol{\theta}_\ell|\boldsymbol{y}_\ell \sim N[\hat{\boldsymbol{\theta}}_\ell, \sigma_\ell^2(\boldsymbol{B}^T\boldsymbol{B} + \lambda_\ell \boldsymbol{P})^{-1}]$. Consequently, for a given x^*, the posterior variance of $g_\ell(x^*) = \boldsymbol{b}(x^*)^T\boldsymbol{\theta}_\ell$ is $\sigma_\ell^2 \boldsymbol{b}(x^*)^T(\boldsymbol{B}^T\boldsymbol{B} + \lambda_\ell \boldsymbol{P})^{-1}\boldsymbol{b}(x^*)$. We can evaluate a pointwise Bayesian confidence interval for g_ℓ, in the sense of Wahba (1983) and Wood (2006b), at given points $x_1^*, \ldots, x_m^*$, by computing the m-dimensional vector of posterior variance estimates

$$\hat{\sigma}_\ell^2 \text{Diag}\left[\boldsymbol{B}^*(\boldsymbol{B}^T\boldsymbol{B} + \lambda_\ell \boldsymbol{P})^{-1}\boldsymbol{B}^{*T}\right], \tag{1}$$

where $\hat{\sigma}_\ell^2$ is an estimate of the ℓth-location error variance and $\boldsymbol{B}^* = \begin{bmatrix} \boldsymbol{b}(x_1^*)^T \\ \vdots \\ \boldsymbol{b}(x_m^*)^T \end{bmatrix}$. Demmler-Reinsch orthogonalization, as applied in the main text, can speed the computation here in two respects. First, a natural error variance estimator is $\hat{\sigma}_\ell^2 = \|\boldsymbol{y}_\ell - \boldsymbol{B}\hat{\boldsymbol{\theta}}_\ell\|^2/(n - \text{df}_\ell)$, where df_ℓ is the effective degrees of freedom of the ℓth-location smooth; and df_ℓ can be obtained for all ℓ by

$$(\text{df}_1, \ldots, \text{df}_L) = \boldsymbol{1}_K^T \boldsymbol{M},$$

where $\boldsymbol{1}_K$ is a vector of K 1s. Second, for a fixed ℓ, we need not compute the entire $m \times m$ matrix whose diagonal appears in (1), since

$$\text{Diag}\left[\boldsymbol{B}^*(\boldsymbol{B}^T\boldsymbol{B} + \lambda_\ell \boldsymbol{P})^{-1}\boldsymbol{B}^{*T}\right] = (\boldsymbol{B}^*\boldsymbol{R}^{-1}\boldsymbol{U})^{\odot 2}\left(\frac{1}{1 + \lambda_\ell \boldsymbol{\tau}}\right),$$

where $\boldsymbol{A}^{\odot 2} = \boldsymbol{A} \odot \boldsymbol{A}$.

3 Implementation of functional data clustering

The voxelwise curve estimates created by our massively parallel procedure can be represented as a single functional data object using the `fda` package in R (Ramsay et al., 2009). This object is determined by a matrix, each column of which gives the coefficient of the curve for one voxel pair with respect to a B-spline basis. The `fda` function `deriv.fd` was used to generate a similar object representing the first derivatives of the curves, and functional principal component analysis was performed on the latter object via the `fda` function `pca.fd`. Due to the relatively large number of curves, we did not find it necessary to regularize the eigenfunctions by means of a roughness penalty. Ordinary k-means clustering was then applied to the matrix of functional principal component scores. Interactive visualization of the clusters is implemented with the R package `rpanel` (Bowman et al., 2007).

4 Timing estimates

A straightforward way to perform RLRT for a single scatterplot smooth using standard software is:

1. use the `gamm` function in R package `mgcv` (Wood, 2006a) to express the nonparametric regression model as a linear mixed-effects object;

2. input this object to the R package `RLRsim` (Scheipl, 2010), which simulates the null distribution for the RLRT, and refers the observed test statistic to this distribution.

Simulating the null distribution in step 2 takes a fraction of a second, and need not be repeated for each voxel since the null distribution is the same in each case. Therefore, our timing estimate for the naïve voxelwise RLRT is based on repeating step 1 for 500 voxel pairs, and extrapolating the elapsed time to 71287 voxel pairs. For naïve voxelwise smoothing, we used the `gam` function in `mgcv` to perform nonparametric regression with smoothing parameter selection by REML (Wood, 2011) for 500 voxel pairs, and again extrapolated to the entire brain.

References

Bowman, A., Crawford, E., Alexander, G., and Bowman, R. (2007). rpanel: Simple interactive controls for R functions using the tcltk package. *Journal of Statistical Software*, 17(9):1–18.

Ramsay, J., Hooker, G., and Graves, S. (2009). *Functional Data Analysis with R and MATLAB*. New York: Springer.

Scheipl, F. (2010). RLRsim: Exact (restricted) likelihood ratio tests for mixed and additive models. R package version 2.0-5.

Silverman, B. (1985). Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *Journal of the Royal Statistical Society, Series B*, 47(1):1–52.

Wahba, G. (1983). Bayesian "confidence intervals" for the cross-validated smoothing spline. *Journal of the Royal Statistical Society, Series B*, 45:133–150.

Wood, S. (2006a). *Generalized Additive Models: An Introduction with R*. Boca Raton, FL: Chapman & Hall.

Wood, S. (2006b). On confidence intervals for generalized additive models based on penalized regression splines. *Australian & New Zealand Journal of Statistics*, 48(4):445–464.

Wood, S. (2011). Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society: Series B*, 73(1):3–36.