Supplemental Material for:

Real-time Semiparametric Regression for Distributed Data Sets

BY JAN LUTS

6th June, 2014

Appendix A: Variational Bayesian Inference for Semiparametric Regression

A mean field approximation is founded upon approximating the posterior density function $p(\boldsymbol{\theta}|\boldsymbol{y})$, e.g. parameter vector $\boldsymbol{\theta} = [\boldsymbol{\beta}, \boldsymbol{u}, a_{u1}, \ldots, a_{ur}, a_\varepsilon, \sigma^2_{u1}, \ldots, \sigma^2_{ur}, \sigma^2_\varepsilon]^T$ for model (3), by a product form $q(\boldsymbol{\theta}) = \prod_{i=1}^d q_i(\boldsymbol{\theta}_i)$. The choice of the $q_i(\boldsymbol{\theta}_i)$ density functions is guided by the notion of Kullback-Leibler divergence

$$\int q(\boldsymbol{\theta}) \log \left\{ \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\boldsymbol{y})} \right\} d\boldsymbol{\theta},$$

such that the distance between $\prod_{i=1}^d q_i(\boldsymbol{\theta}_i)$ and $p(\boldsymbol{\theta}|\boldsymbol{y})$ is minimized. It can be shown that an equivalent optimization problem corresponds to maximizing the so-called lower bound on the marginal likelihood $p(\boldsymbol{y})$,

$$\underline{p}(\boldsymbol{y}; q) \equiv \exp \left[\int q(\boldsymbol{\theta}) \log \left\{ \frac{p(\boldsymbol{\theta}, \boldsymbol{y})}{q(\boldsymbol{\theta})} \right\} d\boldsymbol{\theta} \right].$$

The optimal $q_i^*(\boldsymbol{\theta}_i)$ density functions, in terms of minimizing the Kullback-Leibler divergence, are known to satisfy

$$q_i^*(\boldsymbol{\theta}_i) \propto \exp \left[\int \left\{ \prod_{j \neq i} q_j(\boldsymbol{\theta}_j) \right\} \log p(\boldsymbol{\theta}, \boldsymbol{y}) \, d\boldsymbol{\theta}_{-i} \right],$$

where $\boldsymbol{\theta}_{-i} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{i-1}, \boldsymbol{\theta}_{i+1}, \ldots, \boldsymbol{\theta}_d]^T$.

Although MFVB is limited in its approximation accuracy when compared to MCMC, which can be made arbitrarily accurate by increasing the Monte Carlo sample sizes, the latter is much slower than MFVB. Moreover, the accuracy of MFVB for the models that are considered in the paper is typically excellent (cf. Section 3.1.1).

For the mixed model in (3) the mean field approximation and chosen product form

$$\begin{aligned} &p(\boldsymbol{\beta}, \boldsymbol{u}, a_{u1}, \ldots, a_{ur}, a_\varepsilon, \sigma^2_{u1}, \ldots, \sigma^2_{ur}, \sigma^2_\varepsilon|\boldsymbol{y}) \\ &\qquad \approx q(\boldsymbol{\beta}, \boldsymbol{u}, a_{u1}, \ldots, a_{ur}, a_\varepsilon, \sigma^2_{u1}, \ldots, \sigma^2_{ur}, \sigma^2_\varepsilon) \\ &\qquad \approx q(\boldsymbol{\beta}, \boldsymbol{u}, a_{u1}, \ldots, a_{ur}, a_\varepsilon)\, q(\sigma^2_{u1}, \ldots, \sigma^2_{ur}, \sigma^2_\varepsilon), \end{aligned}$$

lead to the following optimal product density functions: $q^*(\boldsymbol{\beta}, \boldsymbol{u}, a_{u1}, \ldots, a_{ur}, a_\varepsilon)$ is the product of the $N(\boldsymbol{\mu}_{q(\boldsymbol{\beta},\boldsymbol{u})}, \boldsymbol{\Sigma}_{q(\boldsymbol{\beta},\boldsymbol{u})})$ density function, Inverse-Gamma$(1, B_{q(a_{u\ell})})$ density functions, $1 \leq \ell \leq r$, and the Inverse-Gamma$(1, B_{q(a_\varepsilon)})$ density function, while $q^*(\sigma^2_{u1}, \ldots, \sigma^2_{ur}, \sigma^2_\varepsilon)$ is the product of Inverse-Gamma$(\frac{1}{2}(K_\ell + 1), B_{q(\sigma^2_{u\ell})})$ density functions for $1 \leq \ell \leq r$ and the Inverse-Gamma$(\frac{1}{2}(n+1), B_{q(\sigma^2_\varepsilon)})$ density function. Notice that this solution results in so-called induced factorizations. For example, the factorization $q(\boldsymbol{\beta}, \boldsymbol{u}, a_{u1}, \ldots, a_{ur}, a_\varepsilon)$ $= q(\boldsymbol{\beta}, \boldsymbol{u})\, q(a_{u1}), \ldots, q(a_{ur})\, q(a_\varepsilon)$ is not assumed a priori.

Since the optimal parameters in the q^* density functions are interrelated, for example,

$$\boldsymbol{\Sigma}_{q(\boldsymbol{\beta},\boldsymbol{u})} = \left[\mu_{q(1/\sigma_\varepsilon^2)}\, \boldsymbol{C}^T\boldsymbol{C} + \text{blockdiag}\{\sigma_\beta^{-2}\, \boldsymbol{I}_p, \mu_{q(1/\sigma_{u1}^2)}\boldsymbol{I}_{K_1}, \ldots, \mu_{q(1/\sigma_{ur}^2)}\boldsymbol{I}_{K_r}\}\right]^{-1},$$

with $\boldsymbol{C} = [\boldsymbol{X}\, \boldsymbol{Z}]$, the iterative coordinate ascent Algorithm A.1 is used to compute the optimal densities where the logarithm of the lower bound equals

$$\begin{aligned}
\log \underline{p}(\boldsymbol{y}; q) \;=\; & \frac{p + \sum_{\ell=1}^r K_\ell}{2} - \frac{n}{2}\log(2\pi) - (r+1)\log(\pi) - \frac{p}{2}\log(\sigma_\beta^2) + \frac{1}{2}\log(|\boldsymbol{\Sigma}_{q(\boldsymbol{\beta},\boldsymbol{u})}|) \\
& + \log\left(\Gamma\left(\frac{n+1}{2}\right)\right) - \frac{1}{2\sigma_\beta^2}\{||\boldsymbol{\mu}_{q(\boldsymbol{\beta})}||^2 + \text{tr}(\boldsymbol{\Sigma}_{q(\boldsymbol{\beta})})\} - \left(\frac{n+1}{2}\right)\log(B_{q(\sigma_\varepsilon^2)}) \\
& + \mu_{q(1/a_\varepsilon)}\mu_{q(1/\sigma_\varepsilon^2)} - \log(A_\varepsilon) - \log(B_{q(a_\varepsilon)}) + \sum_{\ell=1}^r \Bigg\{ \log\left(\Gamma\left(\frac{K_\ell+1}{2}\right)\right) \\
& - \log(A_{u\ell}) - \log(B_{q(a_{u\ell})}) - \left(\frac{K_\ell+1}{2}\right)\log(B_{q(\sigma_{u\ell}^2)}) + \mu_{q(1/a_{u\ell})}\mu_{q(1/\sigma_{u\ell}^2)} \Bigg\}.
\end{aligned}$$

Algorithm A.1 *Mean field variational Bayes algorithm for obtaining the parameters in the optimal densities for the Gaussian linear mixed model (3).*

Require: $\boldsymbol{C}, \boldsymbol{y}, n, p, K_\ell, \mu_{q(1/\sigma_\varepsilon^2)}, A_\varepsilon, \mu_{q(1/\sigma_{u\ell}^2)}, A_{u\ell}, \sigma_\beta^2$ with $1 \le \ell \le r$

1: **while** the increase in $\log \underline{p}(\boldsymbol{y}; q)$ is significant **do**

2: $\quad \boldsymbol{\Sigma}_{q(\boldsymbol{\beta},\boldsymbol{u})} \leftarrow \left[\mu_{q(1/\sigma_\varepsilon^2)}\, \boldsymbol{C}^T\boldsymbol{C} + \text{blockdiag}\{\sigma_\beta^{-2}\, \boldsymbol{I}_p, \mu_{q(1/\sigma_{u1}^2)}\boldsymbol{I}_{K_1}, \ldots, \mu_{q(1/\sigma_{ur}^2)}\boldsymbol{I}_{K_r}\}\right]^{-1}$

3: $\quad \boldsymbol{\mu}_{q(\boldsymbol{\beta},\boldsymbol{u})} \leftarrow \mu_{q(1/\sigma_\varepsilon^2)}\, \boldsymbol{\Sigma}_{q(\boldsymbol{\beta},\boldsymbol{u})}\, \boldsymbol{C}^T\boldsymbol{y}; \quad \mu_{q(1/a_\varepsilon)} \leftarrow 1/\{\mu_{q(1/\sigma_\varepsilon^2)} + A_\varepsilon^{-2}\}$

4: $\quad \mu_{q(1/\sigma_\varepsilon^2)} \leftarrow \dfrac{n+1}{2\,\mu_{q(1/a_\varepsilon)} + \boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{\mu}_{q(\boldsymbol{\beta},\boldsymbol{u})}^T\boldsymbol{C}^T\boldsymbol{y} + \text{tr}[(\boldsymbol{C}^T\boldsymbol{C})\{\boldsymbol{\Sigma}_{q(\boldsymbol{\beta},\boldsymbol{u})} + \boldsymbol{\mu}_{q(\boldsymbol{\beta},\boldsymbol{u})}\boldsymbol{\mu}_{q(\boldsymbol{\beta},\boldsymbol{u})}^T\}]}$

5: $\quad$ **for** $\ell = 1 \to r$ **do**

6: $\quad\quad \mu_{q(1/a_{u\ell})} \leftarrow 1/\{\mu_{q(1/\sigma_{u\ell}^2)} + A_{u\ell}^{-2}\}; \quad \mu_{q(1/\sigma_{u\ell}^2)} \leftarrow \dfrac{K_\ell + 1}{2\,\mu_{q(1/a_{u\ell})} + \|\boldsymbol{\mu}_{q(\boldsymbol{u}_\ell)}\|^2 + \text{tr}(\boldsymbol{\Sigma}_{q(\boldsymbol{u}_\ell)})}$

7: $\quad$ **end for**

8: **end while**

Appendix B: Computational Efficiency

The following section provides additional details on computational aspects of the algorithms. Since batch Algorithm 1 does not directly process the actual data but only summary statistics, its computational cost is independent of the number of samples n. While the number of hosts h has a linear effect on the computational cost, the dimension of a single sample (i.e. $P = p + \sum_{\ell=1}^r K_\ell$) has a stronger effect: Algorithm 1 is $O(P^3)$ due to the matrix inversion in line 7 and the lower bound computation.

On the other hand, Algorithm 2 is an online approach and therefore the number of time instances clearly has a linear effect on the computational cost. The number of items in the buffer, i.e. B, shows a linear effect since all the summary statistics need to be added together. To illustrate this, Table B.1 summarizes the computational times for online Algorithm 2 for the example from Section 3.2.1. The computational cost is averaged over 100 random synthetic data sets and is presented for various scenarios: increasing number of hosts, increasing number of time instances and number of samples for which hosts transfer summary statistics at each time instance. As expected, the number of time instances has

a linear effect on the computational cost. The number of hosts shows a small effect since the combiner only needs to sum the summary statistics in the buffer together. The number of samples for which the hosts transfer summary statistics has no effect. Finally, note that Algorithm 2 is also $O(P^3)$ because of the matrix inversion in line 7.

Table B.1: *Average computational cost in seconds for applying Algorithm 2 to the synthetic data example in Section 3.2.1 for various scenarios.*

(a) samples = 10

time instance	hosts		
	1	3	9
100	0.57	0.57	0.61
200	0.88	0.91	1.00
300	1.20	1.25	1.44
400	1.51	1.56	1.66
500	1.86	1.86	2.01

(b) samples = 500

time instance	hosts		
	1	3	9
100	0.55	0.57	0.61
200	0.84	0.93	1.02
300	1.14	1.18	1.39
400	1.42	1.47	1.68
500	1.71	1.78	2.12

Matrix Inversion for Grouped Data

As also reported by [1] in the context of frequentist inference for additive mixed models, naïve implementation of the matrix inversion for grouped data as in (4) can be extremely inefficient. Algorithm 1 and Algorithm 2 have time complexity $O(P^3)$, which makes them impractical for large numbers of groups. Moreover, since Algorithm 2 aims to run in an online fashion on large-scale data with potentially many groups, it is important to optimize this line of code. [1] outlines a procedure for which the variance calculations are linear in the number of groups, but omits the computation of correlations between any two groups. Algorithm 2, however, does require calculating these inter-group correlations since the full matrix $\boldsymbol{\Sigma}_{q(\boldsymbol{\beta},\boldsymbol{u})}$ is needed to compute $\boldsymbol{\mu}_{q(\boldsymbol{\beta},\boldsymbol{u})}$, for example. The following paragraphs explain how line 7 can be solved in a more efficient way for grouped data as for example the live example in Section 6.

Assume that $\boldsymbol{C} = [\boldsymbol{X}\ \boldsymbol{Z}_1\ \boldsymbol{Z}_2]$, where the original design matrix $\boldsymbol{Z}$ is divided into a design matrix that is only related to the K_r random intercepts, i.e. $\boldsymbol{Z}_2$, and a design matrix for all the rest, i.e. $\boldsymbol{Z}_1$, including spline basis functions. This enables the block decomposition

$$\begin{aligned}
\boldsymbol{M} \equiv \boldsymbol{\Sigma}^{-1}_{q(\boldsymbol{\beta},\boldsymbol{u})} &= \mu_{q(1/\sigma^2_\varepsilon)} \left[\begin{array}{cc|c} \boldsymbol{X}^T\boldsymbol{X} + \mu^{-1}_{q(1/\sigma^2_\varepsilon)}\boldsymbol{G}_1 & \boldsymbol{X}^T\boldsymbol{Z}_1 & \boldsymbol{X}^T\boldsymbol{Z}_2 \\ \boldsymbol{Z}_1^T\boldsymbol{X} & \boldsymbol{Z}_1^T\boldsymbol{Z}_1 + \mu^{-1}_{q(1/\sigma^2_\varepsilon)}\boldsymbol{G}_2 & \boldsymbol{Z}_1^T\boldsymbol{Z}_2 \\ \hline \boldsymbol{Z}_2^T\boldsymbol{X} & \boldsymbol{Z}_2^T\boldsymbol{Z}_1 & \boldsymbol{Z}_2^T\boldsymbol{Z}_2 + \mu^{-1}_{q(1/\sigma^2_\varepsilon)}\boldsymbol{G}_3 \end{array}\right] \\
&= \mu_{q(1/\sigma^2_\varepsilon)} \left[\begin{array}{cc} \boldsymbol{M}_{11} & \boldsymbol{M}_{12} \\ \boldsymbol{M}_{21} & \boldsymbol{M}_{22} \end{array}\right],
\end{aligned}$$

where $\boldsymbol{G}_1 = \sigma^{-2}_\beta \boldsymbol{I}_p$, $\boldsymbol{G}_2 = \text{blockdiag}\{\mu_{q(1/\sigma^2_{u1})}\boldsymbol{I}_{K_1}, \ldots, \mu_{q(1/\sigma^2_{ur-1})}\boldsymbol{I}_{K_{r-1}}\}$ and $\boldsymbol{G}_3 = \mu_{q(1/\sigma^2_{ur})}$ $\boldsymbol{I}_{K_r}$. The rules for computing the inverse of a block-partitioned matrix give

$$\boldsymbol{M}^{-1} \equiv \boldsymbol{\Sigma}_{q(\boldsymbol{\beta},\boldsymbol{u})} = \mu^{-1}_{q(1/\sigma^2_\varepsilon)} \left[\begin{array}{cc} \boldsymbol{M}^{11} & \boldsymbol{M}^{12} \\ \boldsymbol{M}^{21} & \boldsymbol{M}^{22} \end{array}\right],$$

with $\boldsymbol{M}^{11} = (\boldsymbol{M}_{11} - \boldsymbol{M}_{12}\boldsymbol{M}_{22}^{-1}\boldsymbol{M}_{21})^{-1}$, $\boldsymbol{M}^{12} = -\boldsymbol{M}^{11}\boldsymbol{M}_{12}\boldsymbol{M}_{22}^{-1}$, $\boldsymbol{M}^{21} = (\boldsymbol{M}^{12})^T$ and $\boldsymbol{M}^{22} = \boldsymbol{M}_{22}^{-1} + \boldsymbol{M}_{22}^{-1}\boldsymbol{M}_{21}\boldsymbol{M}^{11}\boldsymbol{M}_{12}\boldsymbol{M}_{22}^{-1}$ [2]. Dealing with a large number of groups results

in the relationship $K_r \gg p + \sum_{\ell=1}^{r-1} K_l$. In these circumstances, the straightforward matrix multiplications $\boldsymbol{X}^T\boldsymbol{X}$, $\boldsymbol{X}^T\boldsymbol{Z}_1$ and $\boldsymbol{Z}_1^T\boldsymbol{Z}_1$ are relatively inexpensive. As also explained in [1], $\boldsymbol{Z}_2$ has a special structure because of the random intercept design, thereby making the computation of $\boldsymbol{X}^T\boldsymbol{Z}_2$ and $\boldsymbol{Z}_1^T\boldsymbol{Z}_2$ efficient. The biggest inverse that is needed for computing $\boldsymbol{M}^{-1}$ is $\boldsymbol{M}_{22}^{-1}$, but since $\boldsymbol{M}_{22}$ is diagonal it can be obtained in K_r steps. The final step to obtain $\boldsymbol{\Sigma}^{-1}_{q(\boldsymbol{\beta},\boldsymbol{u})}$ is computing $\boldsymbol{M}^{22}$. Whereas [1] only computes the diagonal elements of this matrix, Algorithm 2 requires all unique entries of this symmetric matrix. Denoting $\boldsymbol{M}_{12} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_{K_r}]$, the elements of $\boldsymbol{M}^{22}$ are

$$\begin{aligned}
\boldsymbol{M}^{22}_{ii} &= \frac{\mu_{q(1/\sigma^2_\varepsilon)}}{n_i\mu_{q(1/\sigma^2_\varepsilon)}+\mu_{q(1/\sigma^2_{ur})}}\left(1 + \frac{\mu_{q(1/\sigma^2_\varepsilon)}\,\boldsymbol{h}_i^T\boldsymbol{M}^{11}\boldsymbol{h}_i}{n_i\mu_{q(1/\sigma^2_\varepsilon)}+\mu_{q(1/\sigma^2_{ur})}}\right),\\
\boldsymbol{M}^{22}_{ij} &= \frac{\mu^2_{q(1/\sigma^2_\varepsilon)}\,\boldsymbol{h}_i^T\boldsymbol{M}^{11}\boldsymbol{h}_j}{\left(n_i\mu_{q(1/\sigma^2_\varepsilon)}+\mu_{q(1/\sigma^2_{ur})}\right)\left(n_j\mu_{q(1/\sigma^2_\varepsilon)}+\mu_{q(1/\sigma^2_{ur})}\right)}, \quad i \neq j,
\end{aligned}$$

with n_i the number of subjects in group i. In summary, the time complexity to compute $\boldsymbol{M}^{11}$ and $\boldsymbol{M}^{12}$ (or $\boldsymbol{M}^{21}$) is $O(K_r)$, while computing $\boldsymbol{M}^{22}$ is $O(K_r^2)$. Therefore, optimizing the matrix inverse in line 7 reduces the time complexity of Algorithm 2 to $O(K_r^2)$.

Parallelization

Although the algorithms are sequential, further optimization can be achieved by parallelization. For example, all the summations for the summary statistics can be done independently. To compute the matrix inverse in line 7 of Algorithm 2, the $K_r(K_r+1)/2$ unique entries of $\boldsymbol{M}^{22}$ can be computed in parallel. Parallel matrix multiplication algorithms can also speed up lines 8 and 9. In addition, $\mu_{q(1/a_{u\ell})}$ and $\mu_{q(1/\sigma^2_{u\ell})}$ can also be computed in parallel for each ℓ.

Transferring Summary Statistics

As explained in Section 3.1, the total number of parameters that each host has to send equals $P(P+1)/2+P+2$ in Algorithm 1. The first term of this sum can further be reduced depending on the particular structure of $\boldsymbol{C}_g$. For example, in case of random intercept model (4), the matrix $\boldsymbol{Z}_{2_g}$ (i.e. the design matrix for the random intercepts from host g) is extremely sparse since each row only contains a single non-zero entry, i.e. 1. As $\boldsymbol{Z}_{2_g}^T\boldsymbol{Z}_{2_g}$ is a diagonal matrix with the number of samples per group (i.e. n_i) on its diagonal, a significant reduction (i.e. $K_r(K_r-1)/2$) in entries to be transferred from host to combiner can be obtained.

References

[1] A. D. A. C. Smith and M. P. Wand, "Streamlined variance calculations for semiparametric mixed models," *Statistics in Medicine*, vol. 27, no. 3, pp. 435–448, 2008.

[2] D. A. Harville, *Matrix Algebra From a Statistician's Perspective*. New York, USA: Springer, 2000.