Computational Linear Algebra for Large Scale Problems: Homework 3

Marco Mungai Coppolino s333969

February 22, 2025

Contents

Abstract

Structured additive regression (STAR) models are an important tool for the applied statistician. They allow adequate modeling of temporal, spatial and spatio-temporal effects as well as non-linear effects for continuous covariates. With advancements in data storage and processing capabilities, the need to fit models with high-dimensional regression coefficients has grown. For instance, in spatial statistics, available spatial information can be incorporated as a latent Gaussian Markov random field (GMRF) using hierarchical models, resulting in regression models with hundreds of thousands of parameters. In such cases, the computational limits of standard workstations are often quickly reached. For this reason, after a brief overview of the STAR model, a Markov chain Monte Carlo (MCMC) framework will then be presented, an algorithm that enables the fitting of such large models, while maintaining low to moderate computational demands. The method integrates a modified sampling scheme with recent advancements in iterative methods for sparse linear systems. This way a solution is given that effectively addresses computational challenges such as calculating the log-determinant of massive precision matrices and sampling from high-dimensional Gaussian distributions.

1 STructured Additive Regression (STAR) models

This chapter provides a concise introduction to the general framework of STAR models, detailing their functionality and laying the groundwork for the computational enhancements discussed in the following section.

1.1 Additive Models

Let's consider a data setting characterized by a vector of conditionally independent response variables $\boldsymbol{y} = (y_1, \dots, y_n)^{\mathrm{T}}$, that follows a probability density function $p(\boldsymbol{y}|\boldsymbol{\theta}) = \prod_{i=1}^{n} p(y_i|\boldsymbol{z_i}, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is a collection of unknown parameters and $\boldsymbol{z_i} = (z_{i1}, \dots, z_{ip})$ is a vector of covariates, s.t. $\boldsymbol{z} = (\boldsymbol{z_1}, \dots, \boldsymbol{z_n})^{\mathrm{T}}$. Given an observation $(y_i, z_{i1}, \dots, z_{1p})$, additive models are defined by the following equation:

$$y_i = \eta_i + \epsilon_i = f_1(z_{i1}) + \cdots + f_p(z_{ip}) + \epsilon_i$$

The conditional expectation $\mu_i = E(y_i|\boldsymbol{z_i}, \boldsymbol{\theta})$ is linked to a linear predictor η_i, that can be rewritten as a linear combination of nonparametric functions $f_1, \dots, f_p$ of $\boldsymbol{z}$, called polynomial splines.

A function $f : [a, b] \to \mathbb{R}$ is a polynomial spline of degree $D \geq 0$, with knots $a = \tau_1 < \cdots < \tau_{m_k} = b$, if it fulfills the following conditions:

1. $f(z)$ is $(D-1)$-times continuously differentiable
2. $f(z)$ is a polynomial of degree D on the intervals $[\tau_k, \tau_{k+1})$, defined by the knots

Before employing polynomial splines in nonparametric regression, it is essential to represent the set of polynomial splines for a given degree and knot configuration. This can be accomplished through various equivalent methods. Without going into the details, the primary goal of these approaches is to identify a set of basis functions B_l that will form the basis of the polynomial spline set. This leads to the following approximation of the functions $f_k, k = 1, \dots, p$:

$$f_k(z_k) = \sum_{j=1}^{m_k} \gamma_{kj} B_j(z_k)$$

The function evaluations $\boldsymbol{f_k} = (f_k(z_{1k}), \dots, f_k(z_{nk}))^{\mathrm{T}}$ can then be expressed as a combination of a $n \times m_k$ design matrix $\boldsymbol{Z}_k$, (which elements are the basis functions evaluated at the observed covariate values, i.e.$\boldsymbol{Z}_k[i, j] = B_j(z_{ik})$), and a $1 \times m_k$ vector of unknown regression coefficients $\boldsymbol{\gamma}_k = (\gamma_{k1}, \dots, \gamma_{km_k})^{\mathrm{T}}$. So we have that:

$$\boldsymbol{f}_k = \boldsymbol{Z}_k \boldsymbol{\gamma}_k$$

And we can rewrite the linear predictor in a compact notation:

$$\boldsymbol{\eta} = \boldsymbol{Z}_1 \boldsymbol{\gamma}_1 + \cdots + \boldsymbol{Z}_p \boldsymbol{\gamma}_p$$

After implementing the above definitions, we can conclude that the set of unknown parameters $\boldsymbol{\theta}$ of the likelihood function $p(\boldsymbol{y}|\boldsymbol{\theta})$ consists of the regression coefficients $\boldsymbol{\gamma}_k, k = 1, \dots, p$, and a dispersion parameter ϕ, that is related to the phenomenon of overdispersion.

1.2 Prior Specification

In Bayesian statistics the unknown parameter $\boldsymbol{\theta}$ is treated as a random variable, meaning it has a probability distribution $p(\boldsymbol{\theta})$, that express our ignorance about $\boldsymbol{\theta}$. In general it's costume to attribute a multivariate normal distribution to the vector $\boldsymbol{\theta}$, and in our case this results in defining a prior such that:

$$p(\boldsymbol{\gamma}_k|\kappa_k) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_k^{-1})$$

So the vectors of unknown regression coefficients follow a multivariate normal distribution with mean $\boldsymbol{\mu} = \boldsymbol{0}$ and with a precision matrix $\boldsymbol{Q}_k = \boldsymbol{\Sigma}^{-1}$, (the inverse of the covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{Q}_k^{-1}$), that depends on a precision parameter κ_k, i.e. $\boldsymbol{Q}_k = \boldsymbol{Q}_k(\kappa_k)\boldsymbol{K}_k$, where $\boldsymbol{K}_k$ is a penalty matrix. This is the reason behind the "STructured" term in the model's name. By altering the penalty matrix $\boldsymbol{K}_k$, also known as the structured matrix, we can impose different assumptions on the structure of $\boldsymbol{Q}_k$. This flexibility allows for improved prior configurations when modeling temporal or spatial data. Moreover, by looking at the definition (), its clear that each vector $\boldsymbol{\gamma}_k$ can be seen as a GMRF w.r.t. a labelled graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with nodes $\mathcal{V} = \{1, \dots, m_k\}$ and edges $\mathcal{E}$, due to its following of a multivariate normal distribution with a density:

$$p(\boldsymbol{\gamma}_k) = (2\pi)^{-m_k/2} |\boldsymbol{Q}_k|^{1/2} \exp\left(-\frac{1}{2}(\boldsymbol{\gamma}_k - \boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{Q}_k (\boldsymbol{\gamma}_k - \boldsymbol{\mu})\right)$$

In our case $\boldsymbol{\mu} = \boldsymbol{0}$ and $\boldsymbol{Q}_k$ is a symmetric positive definitive matrix, whose nonzero entries defines the dependence structure of the vertices in $\mathcal{G}$. Thus, the non-zero pattern of $\boldsymbol{Q}_k$ directly induce the neighborhood structure of an undirected graph, that is usually used to impose a smoothness-penalty on the elements of $\boldsymbol{\gamma_k}$

1.2.1 Hyperparameters

If the linear predictor contains non fixed effects, (spatial, temporal, or spatial-temporal effects), then the precision parameters κ_k, $k = 1, \ldots, p$ need to be estimated as well. This is done by setting up prior distributions for both the κ_k and for the dispersion parameter ϕ:

- For the precision parameters is common to choose independent gamma distributions with shape and rate parameters a_k and b_k, resulting in the following distribution:

$$p(\kappa_k) = \frac{b_k^{a_k}}{\Gamma(a_k)} \kappa_k^{a_k - 1} exp(-b_k \kappa_k)$$

- Similarly for ϕ we choose a gamma prior with parameters a_ϕ and b_ϕ

1.3 MCMC based Inference for STAR models

In the Bayesian approach, the inference relies solely on the joint posterior, that in our case can be written as follows:

$$\begin{aligned} p(\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_p, \kappa_1, \ldots, \kappa_p, \phi | \boldsymbol{y}) \propto & \prod_{i=1}^{n} p(y_i | \boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_p, \kappa_1, \ldots, \kappa_p, \phi) \\ & \times \prod_{k=1}^{p} |\boldsymbol{Q}_k(\kappa_k)|^{1/2} \exp(-\frac{1}{2} \boldsymbol{\gamma}_k^T \boldsymbol{Q}_k(\kappa_k) \boldsymbol{\gamma}_k) \\ & \times \prod_{k=1}^{p} \kappa_k^{a_k - 1} \exp(-b_k \kappa_k) \\ & \times \phi^{a_\phi - 1} \exp(-b_\phi \phi) \end{aligned}$$

This is a result of the Bayes' Theorem, that says:

$$p(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\boldsymbol{y})} \text{ , where } p(\boldsymbol{y}) \text{ is the marginal likelihood}$$

The exploration of the posterior can be effectively performed using a Markov Chain Monte Carlo (MCMC) algorithm. In Bayesian statistics, MCMC simulation is implemented to generate samples from the joint posterior distribution, which are used for its summarization. In the paper is implemented a Gibbs sampler, in which was included a Metropolis-Hastings step, that is added for solving the problem of not having access to a closed form of the full conditional of the regression coefficients: $p(\boldsymbol{\gamma_k}|\boldsymbol{y}, \kappa_k)$. For general likelihoods a proposal density for $\boldsymbol{\gamma_k}$ can be obtained from a GMRF approximation of the log-likehood. The idea is to perform a taylor expansion of the like log-likelihood: $l(\boldsymbol{\gamma}_k) = \sum_{i=1}^{n} \log p(y_i|\boldsymbol{\gamma}_k)$, around the current state $\boldsymbol{\gamma}_k^c$, which can be written as:

$$l(\boldsymbol{\gamma}_k) \approx a_k^c + (\boldsymbol{b}_k^c)^{\mathrm{T}} \boldsymbol{\gamma}_k - \frac{1}{2} \boldsymbol{\gamma}_k^{\mathrm{T}} \boldsymbol{C}_k^c \boldsymbol{\gamma}_k$$

with coefficients

$$\begin{aligned} a_k^c &= l(\boldsymbol{\gamma}_k^c) - (\boldsymbol{\gamma}_k^c)^{\mathrm{T}} \frac{\partial l(\boldsymbol{\gamma}_k^c)}{\partial \boldsymbol{\gamma}_k} + \frac{1}{2} (\boldsymbol{\gamma}_k^c)^{\mathrm{T}} \frac{\partial^2 l(\boldsymbol{\gamma}_k^c)}{\partial \boldsymbol{\gamma}_k \partial \boldsymbol{\gamma}_k^{\mathrm{T}}} \boldsymbol{\gamma}_k^c \\ \boldsymbol{b}_k^c &= \frac{\partial l(\boldsymbol{\gamma}_k^c)}{\partial \boldsymbol{\gamma}_k} + \boldsymbol{C}_k^c \boldsymbol{\gamma}_k^c \\ \boldsymbol{C}_k^c &= -\frac{\partial^2 l(\boldsymbol{\gamma}_k^c)}{\partial \boldsymbol{\gamma}_k^{\mathrm{T}} \partial \boldsymbol{\gamma}_k} \end{aligned}$$

The approximation is used to rewrite the full conditional:

$$\begin{aligned} p(\boldsymbol{\gamma}_k|\boldsymbol{y}, \kappa_k) &\propto p(\boldsymbol{y}|\boldsymbol{\gamma}_k) p(\boldsymbol{\gamma}_k|\kappa_k) \\ &\propto \exp\left(-\frac{1}{2} \boldsymbol{\gamma}_k^{\mathrm{T}} Q_k \boldsymbol{\gamma}_k + \sum_{i=1}^{n} \log p(y_i|\boldsymbol{\gamma}_k)\right) \\ &\approx \exp\left(-\frac{1}{2} \boldsymbol{\gamma}_k^{\mathrm{T}} Q_k \boldsymbol{\gamma}_k + a_k^c + (\boldsymbol{b}_k^c)^{\mathrm{T}} \boldsymbol{\gamma}_k - \frac{1}{2} \boldsymbol{\gamma}_k^{\mathrm{T}} C_k^c \boldsymbol{\gamma}_k\right) \\ &\propto \exp\left(-\frac{1}{2} \boldsymbol{\gamma}_k^{\mathrm{T}} (Q_k + C_k^c) \boldsymbol{\gamma}_k + (\boldsymbol{b}_k^c)^{\mathrm{T}} \boldsymbol{\gamma}_k\right). \end{aligned}$$

This corresponds to the core of a multivariate normal distribution, thus, the proposal distribution for $\boldsymbol{\gamma}_k$ based on the GMRF approximation has the form:

$$\boldsymbol{\gamma}_k^p|\cdot \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}_k^c), \tilde{\boldsymbol{Q}}_k^c)$$

where $\tilde{\boldsymbol{Q}}_k^c = \boldsymbol{Q}_k + \boldsymbol{C}_k^c$ and the mean $\tilde{\boldsymbol{\mu}}_k^c$ is the solution of the linear system:

$$\tilde{\boldsymbol{Q}}_k^c \tilde{\boldsymbol{\mu}}_k^c = \boldsymbol{b}_k^c$$

Sampling a proposal $\boldsymbol{\gamma}_k^p$ from this distribution requires the evaluation of $\tilde{\boldsymbol{\mu}}_k^c$ and $\tilde{\boldsymbol{Q}}_k^c$ at the current state $\boldsymbol{\gamma}_k^c$. The proposal is accepted with probability:

$$\alpha(\boldsymbol{\gamma}_k^c, \boldsymbol{\gamma}_k^p) = \min\left\{1, \frac{p(\boldsymbol{y}|\boldsymbol{\gamma}_k^p)p(\boldsymbol{\gamma}_k^p|\kappa_k)\varphi(\boldsymbol{\gamma}_k^c|\tilde{\boldsymbol{\mu}}_k^p, \tilde{\boldsymbol{Q}}_k^p)}{p(\boldsymbol{y}|\boldsymbol{\gamma}_k^c)p(\boldsymbol{\gamma}_k^c|\kappa_k)\varphi(\boldsymbol{\gamma}_k^p|\tilde{\boldsymbol{\mu}}_k^c, \tilde{\boldsymbol{Q}}_k^c)}\right\}$$

where $\varphi(\cdot; \boldsymbol{\mu}, \boldsymbol{Q})$ represents the density of a multivariate normal distributed random variable. In order to obtain the acceptance probability the normalizing constant of φ needs to be calculated which requires the computation of the log-determinant of $\tilde{\boldsymbol{Q}}_k^c$. Lastly, the full conditionals of the hyperparameters are again Gamma distributions with updated parameters $\tilde{a}_k = a_k + {}^{rk(\boldsymbol{K}_k)}/_2$ and $\tilde{b}_k = b_k + \boldsymbol{\gamma}_k^{\mathrm{T}} \boldsymbol{K}_k \boldsymbol{\gamma}_k/2$. This is also the case for the full conditional of ϕ that is a Gamma distribution with parameters $\tilde{a}_\phi = a_\phi + {}^{(n\,-\,1)}/_2$ and $\tilde{b}_\phi = b_\phi + (\boldsymbol{y} - \boldsymbol{\eta})^{\mathrm{T}}(\boldsymbol{y} - \boldsymbol{\eta})/2$

1.3.1 Iterated Weighted Least Squares (IWLS) Proposal

In the case where the likelihood belongs to the exponential family, the GMRF approximation corresponds to the Iterated Weighted Least Squares (IWLS) proposal. In essence, this means that the parameters of the proposal derived from the GMRF approximation can be expressed as:

$$\tilde{\boldsymbol{Q}}_k^c = \boldsymbol{Z}_k^{\mathrm{T}} \boldsymbol{W}^c \boldsymbol{Z}_k + \boldsymbol{Q}_k \qquad \text{and} \qquad \tilde{\boldsymbol{Q}}_k^c \tilde{\boldsymbol{\mu}}_k^c = \boldsymbol{Z}_k^{\mathrm{T}} \boldsymbol{W}^c \boldsymbol{Z}_k + \boldsymbol{Q}_k(\tilde{\boldsymbol{y}}^c - \boldsymbol{\eta}_{-k}^c)$$

Here $\boldsymbol{W} = \boldsymbol{D}\boldsymbol{V}^{-1}\boldsymbol{D}$, with $\boldsymbol{D} = \mathrm{diag}\{\partial h(\eta_i)/\partial\eta\ ,\ i = 1, \ldots, n\}$ and $\boldsymbol{V} = \mathrm{diag}\{\phi v(\mu_i)/\omega_i\ ,\ i = 1, \ldots, n\}$, where $v(\mu_i)$ and ω_i are the variance function and weights corresponding to the specific exponential family. Moreover, we have that $\tilde{\boldsymbol{y}} = \boldsymbol{\eta} + \boldsymbol{D}^{-1}(\boldsymbol{y} - \boldsymbol{\mu})$, and η_{-k} is describing the linear predictor without the k-th term.

2 Application of Krylov Subspace Methods for Sampling and Inference

The methods discussed in the previous section are widely utilized, either in their original form or with modifications. However, when dealing with one or more high-dimensional regression coefficients, it becomes problematic to sample from equation $\boldsymbol{\gamma}_k^p|\cdot \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}_k, \tilde{\boldsymbol{Q}}_k^{-1})$ and compute the log-determinant of $\tilde{\boldsymbol{Q}}_k$ for the evaluation of the acceptance. In this section, we present solutions to address these challenges.

2.1 Iterative Sampling from large GMRF

Sampling from the distribution $\boldsymbol{\gamma}_k^p|\cdot \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}_k, \tilde{\boldsymbol{Q}}_k^{-1})$ within the adopted MCMC scheme can be divided into solving equation $\tilde{\boldsymbol{Q}}_k \tilde{\boldsymbol{\mu}}_k = \boldsymbol{b}_k$ with respect to $\tilde{\boldsymbol{\mu}}_k$ and sampling from $\mathcal{N}(\boldsymbol{0}, \tilde{\boldsymbol{Q}}_k^{-1})$, which requires to solve the system $\tilde{\boldsymbol{Q}}_k^{1/2}\boldsymbol{x} = \boldsymbol{z}$ with $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. For both tasks the Cholesky decomposition of $\tilde{\boldsymbol{Q}}_k = \boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$ is usually used. However, if the precision matrix is too large it may be too computationally expensive to obtain such factorization. In this situation, Krylov subspace methods, a class of iterative procedures, can be a computational efficient alternative for solving linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$. Given a non-vanishing vector $\mathbf{v} \in \mathbb{R}^n$ we define the Krylov Subspace as follows:

$$\mathcal{K}_m(\mathbf{A}, \mathbf{v}) = \mathrm{span}(\mathbf{v}, \mathbf{A}\mathbf{v}, \mathbf{A}^2\mathbf{v} \ldots, \mathbf{A}^{m-1}\mathbf{v})$$

Krylov Subspace methods are orthogonal projection methods onto Krylov Subspaces, that seek to find an approximate $\mathbf{x}_m$ of the solution $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$, from an affine space $\mathbf{x}_0 + \mathcal{L}$, where $\mathcal{L} := \mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)$, with $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$ being the residual of the system at the arbitrary initial guess $\mathbf{x}_0$. This can be achieved by imposing the:

$$\text{Galerkin Ortogonality Condition: } \mathbf{r}_m = \mathbf{b} - \mathbf{A}\mathbf{x}_m \perp \mathcal{L} = \mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)$$

The objective of the krylov subspace methods is constructing an orthonormal basis $\mathbf{v}_1, \ldots, \mathbf{v}_m$ of $\mathcal{K}_m(\mathbf{A}, \mathbf{v}_1)$, where $\mathbf{v}_1 = {}^{\mathbf{r}_0}/_\beta$ with $\beta = \|\mathbf{r}_0\|_2$. We define with $\mathbf{V}_m$ the matrix whose column vectors are the basis $\mathbf{v}_1, \ldots, \mathbf{v}_m$. In case the (modified) Gram-Schmidt orthogonalization is used to build $\mathbf{V}_m$, we obtain the Arnoldi algorithm:

Algorithm 1 Arnoldi Algorithm

Data: $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$, $\beta = \|\mathbf{r}_0\|_2$, $\mathbf{v}_1 = {}^{\mathbf{r}_0}/_{\beta}$
for $j = 1, \ldots, m$ **do**
  $\mathbf{w}_j = \mathbf{A}\mathbf{v}_j$
  for $i = 1, \ldots, j$ **do**
    $h_{i,j} = \mathbf{w}_j^{\mathrm{T}}\mathbf{v}_i$
    $\mathbf{w}_j = \mathbf{w}_j - h_{i,j}\mathbf{v}_i$
  end
  $h_{j+1,j} = \|\mathbf{w}_j\|_2$
  if $h_{j+1,j} == 0$ **then break**
  $\mathbf{v}_{j+1} = {}^{\mathbf{w}_j}/_{h_{j+1,j}}$
end

Let $\bar{\mathbf{H}}_m$ be the $(m+1) \times m$ matrix whose non-zero entries are the h defined by the algorithm, let $\mathbf{H}_m$ be the same matrix without its last row, then is true that:

$$\bullet\ \mathbf{V}_m^{\mathrm{T}}\mathbf{A}\mathbf{V}_m = \mathbf{H}_m \qquad \bullet\ \mathbf{V}_m^{\mathrm{T}}\mathbf{r}_0 = \mathbf{V}_m^{\mathrm{T}}(\beta\mathbf{v}_1) = \beta\mathbf{e}_1$$

Any vector of $\mathcal{K}_m$ can be written as a linear combination of its basis, i.e. the columns of $\mathbf{V}_m$, meaning the approximation $\mathbf{x}_m$ can be written as follows:

$$\mathbf{x}_m = \mathbf{x}_0 + \mathbf{V}_m\mathbf{y}_m$$

where $\mathbf{y}_m$ is a vector of coefficients, that need to be computed. To do so we impose the condition that the residual $\mathbf{r}_m$ has to be $\perp$ to $\mathcal{K}_m(A, \mathbf{r}_0)$, meaning it has to be $\perp$ to all the vectors of the basis of $\mathcal{K}_m$. This implies that:

$$\mathbf{V}_m^{\mathrm{T}}\mathbf{r}_m = \mathbf{0} \quad \underset{\substack{\downarrow \\ \text{by applying the two} \\ \text{above equations}}}{\Longrightarrow} \quad \mathbf{H}_m\mathbf{y}_m = \beta\mathbf{e}_1$$

We have reduced the problem of finding an approximation $\mathbf{x}_m$ to the research of a solution $\mathbf{y}_m$ of a linear system much smaller than the one we originally had. Moreover is true that $\mathbf{r}_m = \mathbf{b} - \mathbf{A}\mathbf{x}_m = -h_{m+1,m}\mathbf{v}_{m+1}\mathbf{e}_m^{\mathrm{T}}\mathbf{y}_m$. This is a consequence of the fact that $\mathbf{A}\mathbf{V}_m = \mathbf{V}_m\mathbf{H}_m + \mathbf{w}_m\mathbf{e}_m^{\mathrm{T}}$. So we can repeat the algorithm, increasing at each step m by 1, until $\|\mathbf{r}_m\|$ satisfies a certain tolerance. Higher m will produce more accurate approximations, but if it becomes too big, we might not be able to store the large matrices $\mathbf{V}_m$ and $\mathbf{H}_m$. To avoid this problem we can stop the algorithm if m reaches a fixed M and restart it by using the last found $\mathbf{x}_m$ as the new $\mathbf{x}_0$

2.1.1 Iterative Methods for Symmetric Matrices

When the matrix $\mathbf{A}$ is symmetric, (that's the case with the matrices $\boldsymbol{Q}$ and $\boldsymbol{Q}^{1/2}$), we have that the matrix $\mathbf{H}_m$ is symmetric and tri-diagonal:

$$\mathbf{H}_m = \mathbf{T}_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & & \\ \beta_2 & \alpha_2 & \beta_3 & & 0 & \\ & \beta_3 & \ddots & \ddots & & \\ & & \ddots & \ddots & \beta_{m-1} & \\ & 0 & & \beta_{m-1} & \alpha_{m-1} & \beta_m \\ & & & & \beta_m & \alpha_m \end{pmatrix}$$

This means that we can rewrite the Arnoldi Algorithm in a different way, obtaining the Lanczos Algorithm:

Algorithm 2 Lanczos Algorithm

Data: $v_0 = 0$, v_1 s.t. $\|\mathbf{v}_1\|_2$, $\beta_1 = 0$.
for $j = 1, \ldots, m$ **do**
  $\mathbf{w}_j = A\mathbf{v}_j - \beta_j\mathbf{v}_{j-1}$
  $\alpha_j = \mathbf{w}'\mathbf{v}_j$
  $\mathbf{w}_j = \mathbf{w}_j - \alpha_j\mathbf{v}_j$
  $\beta_{j+1} = \|\mathbf{w}_j\|_2$
  if $\beta_{j+1} == 0$ **then break**
  $\mathbf{v}_{j+1} = \mathbf{w}_j/\beta_{j+1}$
end

Moreover, if $\mathbf{A}_m$ is symmetric positive definitive then $\exists\, \mathbf{L}_m, \mathbf{U}_m$ s.t.$\mathbf{T}_m = \mathbf{L}_m\mathbf{U}_m$

$$\mathbf{H}_m = \mathbf{T}_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & & \\ \beta_2 & \alpha_2 & \beta_3 & & 0 & \\ & \beta_3 & \ddots & \ddots & & \\ & & \ddots & \ddots & \beta_{m-1} & \\ & 0 & & \beta_{m-1} & \alpha_{m-1} & \beta_m \\ & & & & \beta_m & \alpha_m \end{pmatrix} = \mathbf{L}_m\mathbf{U}_m = \begin{pmatrix} 1 & & & & & \\ \lambda_2 & 1 & & & 0 & \\ & \lambda_3 & \ddots & & & \\ & & \ddots & \ddots & & \\ & 0 & & \lambda_{m-1} & 1 & \\ & & & & \lambda_m & 1 \end{pmatrix} \begin{pmatrix} \eta_1 & \gamma_2 & & & & \\ & \eta_2 & \gamma_3 & & 0 & \\ & & \ddots & \ddots & & \\ & & & \ddots & \gamma_{m-1} & \\ & 0 & & & \eta_{m-1} & \gamma_m \\ & & & & & \eta_m \end{pmatrix}$$

This means that we can rewrite the Lanczos Algorithm in a different way, obtaining the D-Lanczos Algorithm. This can be done in the following way:

$$\beta_1 = \lambda_1 = \gamma_1 = 0,\ \gamma_j = \beta_j,\ \lambda_j = \tfrac{\beta_j}{\eta_{j-1}},\ \eta j = \alpha_j - \tfrac{\beta_j^2}{\eta_{j-1}}$$

Moreover $\mathbf{T}_m = \mathbf{L}_m, \mathbf{U}_m \implies \mathbf{L}_m, \mathbf{z}_m = \beta\mathbf{e}_1$, where $\mathbf{z}_m := \mathbf{U}_m\mathbf{y}_m$ and is s.t. $z_{mj} = (-1)^{i+1}\prod_{i=1}^{j}\lambda_i\beta$

The benefit of this method is that we don't need the full-basis of the krylov space to find the approximation, we only need the last 2, meaning we can save a lots of storage. In our problem, we can solve the $\tilde{\boldsymbol{Q}}_k\tilde{\boldsymbol{\mu}}_k = \boldsymbol{b}_k$ equation with the D-Lanczos algorithm, and then solve the $\tilde{\boldsymbol{Q}}_k^{1/2}\boldsymbol{x} = \boldsymbol{z}$ system with the Lanczos algorithm.

2.2 Calculation of the log-determinant

In order to increase the acceptance rates of the above MCMC sampler within the GLM framework, we can replace $\boldsymbol{\gamma}_k^c$ with the current posterior mode $\tilde{\boldsymbol{\mu}}_k^c$ in the calculation of the current linear predictor $\boldsymbol{\eta}^c$:

$$\boldsymbol{\eta}^c = \boldsymbol{\eta}^c + \boldsymbol{Z}_k(\tilde{\boldsymbol{\eta}}_k^c - \boldsymbol{\gamma}_k^c)$$

In this way, the proposal becomes independent of the current state $\boldsymbol{\gamma}_k^c$, meaning that it is not more required to calculate the log-determinant of $\tilde{\boldsymbol{Q}}_k$ when evaluating the proposal density. As a result, the acceptance rates of the generated chains will be significantly higher than typically recommended. However, for high-dimensional, highly structured regression coefficients, this can be a beneficial attribute.

However, another important thing that can improve the results of the MCMC is the mixing of the chain, a process that would require the calculation of the log-determinant. An approach that can be implemented to solve efficiently this problem comes again from the Krylov Subspace methods. Specifically, the Lanczos algorithm can approximate the eigenvalues of large symmetric matrices, facilitating the calculation of the log-determinant. While this approach increases computational expense, it could lead to a more accurate solution.

References

[1] P. SCHMIDT, M. M¨UHLAU, V. SCHMID (2017) *Fitting large-scale structured additive regression models using Krylov subspace methods*

[2] P. SCHMIDT (2016) *Bayesian inference for structured additive regression models for large-scale problems with applications to medical imaging*

[3] L. FAHRMEIR ET AL. (2004) *Regression Models, Methods and Applications*