JMLR: Workshop and Conference Proceedings 1–7

Bayesian Conjugate Gradients

Andrey Savinov ANDREY.SAVINOV@SKOLTECH.RU
Daria Riabukhina DARIA.RIABUKHINA@SKOLTECH.RU
Lusine Airapetyan LUSINE.AIRAPETYAN@SKOLTECH.RU
Skolkovo Institue of Science and Technology

1. Brief Paper review

The **paper** (1) deals with an iterative method for solution of systems of linear equations of the form $Ax_* = b$, where x_* is to be determined. In contrast to classical approaches, the output of the Bayesian conjugate gradient (BCG) is a probability distribution over vectors $x \in R^d$, which reflects knowledge about x_* after expending a limited amount of computational efforts. In a special case, the mode of this distribution coincides with the estimate provided by the standard conjugate gradient method (CG), whilst the probability mass is proven to contract onto x_* as more iterations are performed.

The main contributions of this paper are as follows:

- The Bayesian conjugate gradient method is proposed for solution of linear systems. This is a novel probabilistic numerical method in which both prior and posterior are defined on the solution space for the linear system.
- The specification of the prior distribution is discussed in detail. Several natural prior covariance structures are introduced, motivated by preconditioners or Krylov subspace methods.
- A thorough convergence analysis for the new method is presented, with computational performance in mind. The distributional quantification of uncertainty provided by this method is shown to be conservative in general.
- Practical instance of BCG application is shown.

2. Problem Statement

Let's introduce prior distribution:

$$p(x) = \mathcal{N}(x; x_0, \Sigma_0), \tag{1}$$

The likelihood is given by Dirac distribution:

$$p(y|x) = \delta(y - S_m^\top Ax), \tag{2}$$

where S_m - matrix with columns s_i, $i = 1 \ldots, m \ll d$, called search directions as in Conjugate Gradients method.

Having specified the prior and the likelihood, there exists a unique Bayesian probabilistic numerical method which outputs the conditional distribution $p(x|y_m)$ (2):

$$p(x|y_m = S_m^\top A x^*) = \mathcal{N}(x; x_m, \Sigma_m), \tag{3}$$
$$x_m = x_0 + \Sigma_0 A^\top S_m \Lambda_m^{-1} S_m^\top r_0, \;\; r_0 = (b - Ax_0)$$
$$\Sigma_m = \Sigma_0 - \Sigma_0 A^\top S_m \Lambda_m^{-1} S_m^\top A \Sigma_0, \;\; \Lambda_m = S_m^\top A \Sigma_0 A^\top S_m.$$

The Goal is to obtain x_m, Σ_m iteratively and avoid inverse of Λ_m. Let's compare CG and BCG update formulas:

- CG updates:

$$x_m = x_{m-1} + \langle s_m, r_{m-1} \rangle s_m$$
$$s_m = r_{m-1} - \langle s_{m-1}, r_{m-1} \rangle_A s_{m-1}$$
$$s_m = \frac{s_m}{||s_m||_A}, \;\; r_m = b - Ax_m, \;\; \langle s_i, s_j \rangle_A = \delta_{ij}$$

- BCG updates:

$$x_m = x_{m-1} + \langle s_{m-1}, r_{m-1} \rangle \Sigma_0 A^\top s_m$$
$$s_m = r_{m-1} - \langle s_m, r_{m-1} \rangle_{A\Sigma_0 A^\top} s_{m-1}$$
$$s_m = \frac{s_m}{||s_m||_{A\Sigma_0 A^\top}}, \;\; r_m = b - Ax_m, \;\; \langle s_i, s_j \rangle_{A\Sigma_0 A^\top} = \delta_{ij}$$

For these updates of BCG $\Lambda_m = I$ and s_i, $i = 1, \ldots, m$ are $A\Sigma_0 A^\top$-orthonormal. The goal is to investigate properties of proposed BCG method.

3. Experiments and Description

We implemented an algorithm for BCG, provided in **paper** with overall cost $O(md^2)$ and an algorithm for CG with a constant factor lower cost than BayesCG.

In this project several experiments were conducted. Firstly, there is the simulation study corresponding to the original paper. It verifies theoretical results. Then, we made some experiments on the real data. The results can be found on **GitHub**.

The experiments were conducted on a usual laptop with 2.5 GHz CPU and 8 GB of RAM. Necessary python libraries are NumPy and SciPy of the latest version.

3.1. Simulation study

These experiments evaluate convergence and uncertainty quantification of BCG. For this set of experiments we took a matrix A from the original research. It also can be found on our GitHub. This sparse symmetric positive-definite 100x100 matrix was drawn with the MATLAB function sprandsym from random eigenvalues taken from an exponential distribution with the parameter $\gamma = 10$. The proportion of non-zero entries was 20%. The vector x was taken from the normal distribution $\mathcal{N}(0, I)$. The prior mean was equal to zero in all the experiments. There were three types of the prior covariance: $\Sigma_0 = I$, $\Sigma_0 = A^{-1}$ and $\Sigma_0 = (P^T P)^{-1}$, where P was a preconditioner found with an incomplete Cholesky factorization with zero fill-in implemented by us: $P = LL^T$. The computational cost is $O(nz(A)^3)$, $nz(A)$ is the number of non-zero entries.

3.1.1. Point Estimation

Figure 1 shows a comparison between the convergence of the posterior mean x_m for CG and BCG. We drew 100 test problems $x^* \sim \mathcal{N}(0, I)$. The relative error $\frac{||x_m - x^*||_2}{||x_m||_2}$ was computed. The y-axis in on a log-scale.

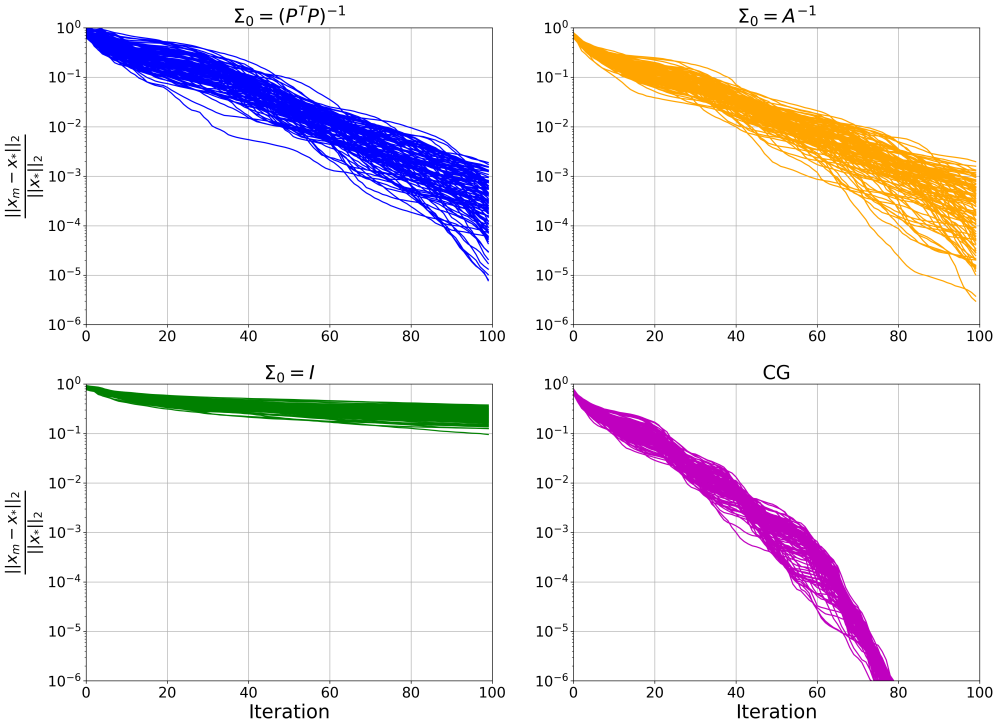


Figure 1: Point estimation for CG and BayesCG

3.1.2. Trace Estimation

Theoretically (1),

$$tr(\Sigma_m \Sigma_0^{-1}) = d - m \tag{4}$$

where d is the number of dimensions and m in the number of iterations. The posterior covariance is evaluated. Experimental setup was the same as in the previous case. The results are presented in the Figure 2. $\frac{tr(\sum_m)}{tr(\sum_0)}$ is plotted. The scale for y-axis is linear.

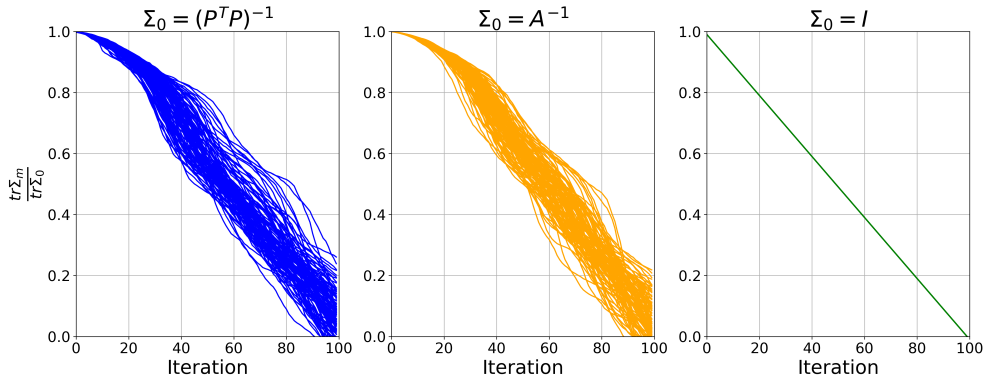


Figure 2: Posterior covariance convergence estimation for BayesCG

3.1.3. Uncertainty Quantification

Here we run the same experiment setup, but with m = 10. We consider 500 test x_* from distribution $\mathcal{N}(0, I)$. $\Sigma_m = UDU^\top$ is SVD decomposition. Therefore, we have a test statistics(1):

$$U_{d-m}D_{d-m}^{-\frac{1}{2}}U_{d-m}^\top(x_m - x_*) \sim \mathcal{N}(0, I_{d-m}) \tag{5}$$

$$Z(x_*) = ||D^{-\frac{1}{2}}U_{d-m}^\top(x_m - x_*)||_2 \sim \chi^2_{d-m} \tag{6}$$

We draw several test problems $x^* \sim \mathcal{N}(0, I)$ and compare the kernel density estimation of $Z(x^*)$ with χ^2_{d-m}. The results are presented in the Figure 3.

3.2. Corner case

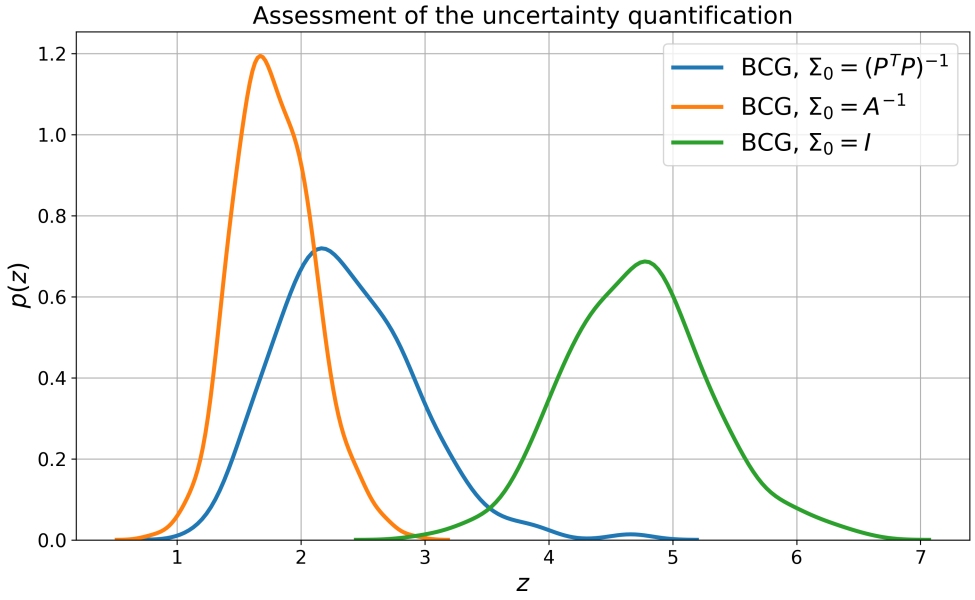
In BCG the matrix A is supposed to be invertible. Here we checked that the method doesn't work for a singular case. We took real data for the Google PageRank problem, which can be downloaded **here**, and solved it as an eigenvalue problem with the NetworkX Python library. Then we considered it as a linear system with the a singular matrix and applied BCG. Depending on choice of the vector b we observed either singular or constant values of norms.

3.3. 1-D Poisson Equation

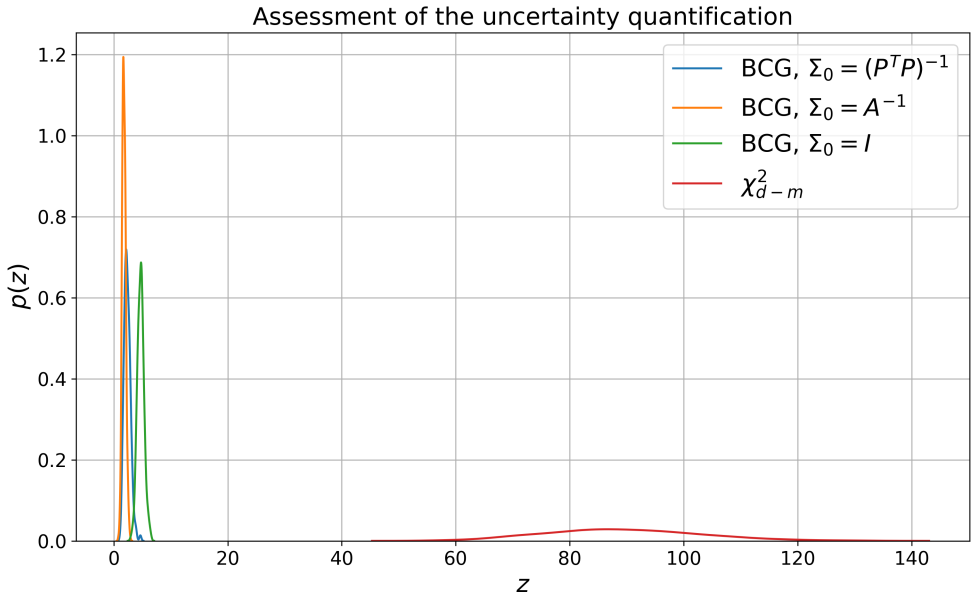
Differential equations may be represented as linear systems due to finite-element discretization()first. We chose 1-D Poisson equation that describes physical phenomena.

$$-\frac{\partial^2 u}{\partial x^2} = f(x), \;\; u(0) = 0, \;\; u(2\pi) = 0 \tag{7}$$

The real data were used (**source**). In Figure 4 $A \in \mathrm{R}^{101\times101}$ is a tridiagonal matrix; $u, f \in \mathrm{R}^{101\times1}$.



(a)



(b)

Figure 3: Assessment of the uncertainty quantification

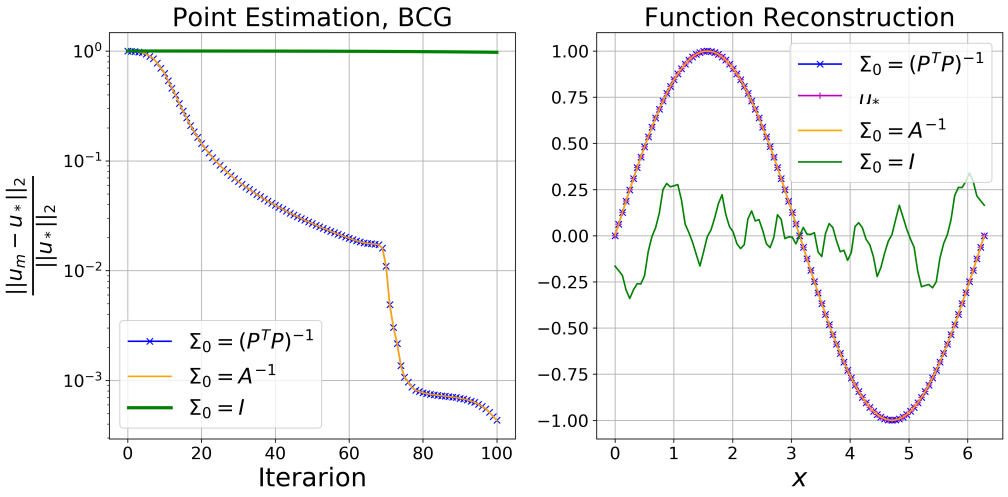


Figure 4: 1-D Poisson equation point estimation and function reconstruction.

4. Discussion of results

4.1. Point Estimation

In Figure 1 the convergence of the posterior mean x_m from BayesCG is contrasted with that of the output of CG, for many test problems x with a fixed sparse matrix A. The convergence of the BayesCG mean vector in all cases is slower than in CG in contradiction to original paper results for the case of $\Sigma_0 = (P^T P)^{-1}$.

4.2. Trace Estimation

In this section the full posterior output from BayesCG is evaluated. In Figure 2, the convergence rate of $tr(\Sigma_m)$ is plotted for the same set of problems. $tr(\Sigma_m)$ appears to contract at a roughly linear rate, in contrast to the exponential rate observed for x_m. Again, reported convergence rate for $\Sigma_0 = (P^T P)^{-1}$ case from the paper is not reproduced.

4.3. Uncertainty Quantification

In Figure 3 the empirical distribution of the statistic z was compared to χ^2_{d-m} with different prior covariances. For BayesCG the UQ provided by the posterior was overly-conservative. Furthermore, note that the quality of the UQ seems to worsen as the convergence rate for x_m improves.

4.4. 1-D Poisson Equation

Figure 4 demonstrates that BCG approach managed to reconstruct the exact solution, except for the case $\Sigma_0 = I$.

5. Contribution of each team member

- Andrey Savinov: responsible for usual CG and incomplete Cholesky decomposition implementation, experiments with Trace and Point estimation (posterior covariance and mean), Uncertainty Quantification. Realization of solution to 1-D Poisson equation. Presentation and report preparation.
- Daria Riabukhina: Point estimation, Google PageRank, Presentation, Report
- Lusine Airapetyan: Realization of BCG method, Report

References

[1] Jon Cockayne, Chris J. Oates, Ilse C.F. Ipsen, Mark Girolami. A Bayesian Conjugate-Gradient Method, 2018. **https://arxiv.org/pdf/1801.05242.pdf**

[2] Jon Cockayne, Chris Oates, Tim Sullivan, Mark Girolami. Bayesian Probabilistic Numerical Methods, 2017. **https://arxiv.org/pdf/1702.03673.pdf**