A Bayesian Conjugate Gradient Method

Jon Cockayne, PN Spring School, 29 March 2023

Solving Linear Systems

The Problem

- Goal: find $x^\star$ in the equation

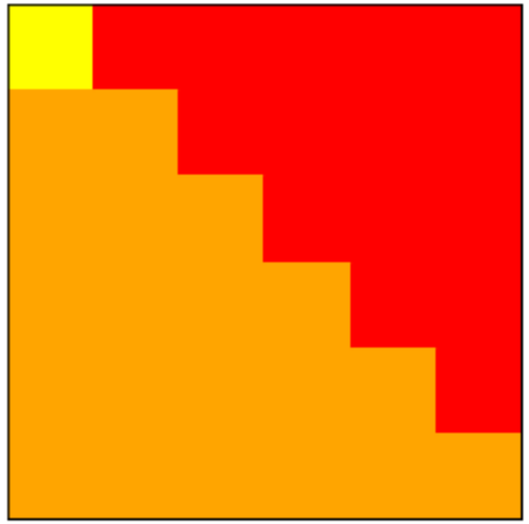
$$Ax^\star = b$$

- $A \in \mathbb{R}^{d \times d}$ invertible (sometimes SPD)
- $x^\star, b \in \mathbb{R}^d$

Direct Methods

- **Direct methods** aim to solve the system in “one shot”
- E.g. **Cholesky factorisation**:

  1. Compute $A = LL^\top$
  2. Compute $Lz = b$
  3. Solve $L^\top x^\star = z$



- (Naive) cost: $\mathcal{O}(d^3)$ computation, $\mathcal{O}(d^2)$ storage.

Iterative Methods

- **Iterative Methods** aim to produce a sequence $(x_m) \to x^\star$ as $m \to \infty$.
- Often possible to elicit an iterative method that is **faster** than a direct method if we are willing to accept a small error in the result.

The Conjugate Gradient Method

Hestenes and Stiefel, 1952

- Consider the functional

$$f(x) = \frac{1}{2} x^\top A x - x^\top b$$

  which has a unique minimum $x^\star$.

- CG arises from performing **modified gradient descent** on this function.

The Conjugate Gradient Method

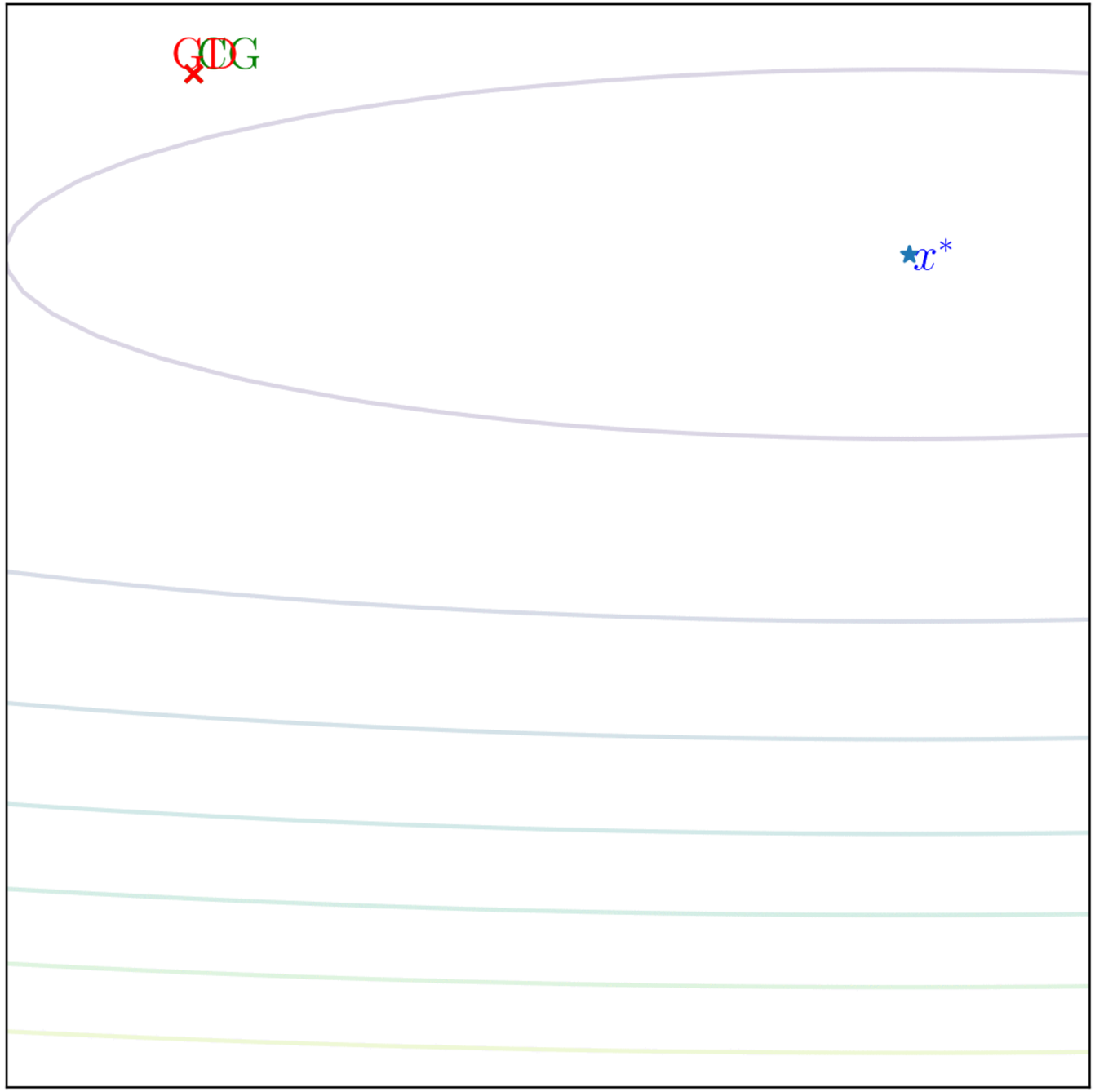
- Raw gradient descent:

$$\tilde{s}_m = b - Ax_{m-1} = r_{m-1}$$

- CG search directions:

$$\tilde{s}_m = r_{m-1} - \langle s_{m-1}, r_{m-1} \rangle_A \times s_{m-1}$$

- Produces a set of search directions that are A-orthonormal (after normalisation)


GD
CDG
x^*

Computational Cost

- $\mathcal{O}(md^2)$ computation (1 matrix-vector product per-iteration)
- $\mathcal{O}(d)$ storage (only need to store 2-3 additional vectors)

Classical Theory

- Introduce the **Krylov Subspace**

$$K_m(A, b) = \text{span}\{b, Ab, \ldots, A^{m-1}b\}$$

Theorem (Krylov Subpace Method)

Let $K_m^\star = x_0 + K_m(A, r_0)$. Then:

$$x_m = \underset{x \in K_m^\star}{\operatorname{argmin}} \|x - x^\star\|_A$$

$$\left(\|z\|_A^2 = z^\top A z \right)$$

Theorem (CG Converges Fast)

It holds that:

$$\frac{\|x_m - x^\star\|_A}{\|x_0 - x^\star\|_A} \leq 2\left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1}\right)^m$$

BayesCG

Probabilistic Linear Solvers

- Start with a Gaussian prior $x \sim \mathcal{N}(x_0, \Sigma_0)$
- Condition on **data** provided by a set of search directions:

$$s_m^\top A x^\star = s_m^\top b =: y_m$$

- Letting $S = (s_1 \quad \dots \quad s_m)$:

Probabilistic Linear Solver (Posterior)

$$x \mid y_1, \ldots, y_m \sim \mathcal{N}(x_m, \Sigma_m)$$

$$x_m = x_0 + \Sigma_0 A^\top S_m \Lambda_m^{-1}(b - Ax_0)$$

$$\Sigma_m = \Sigma_0 - \Sigma_0 A^\top S_m \Lambda_m^{-1} S_m^\top A \Sigma_0$$

where

$$\Lambda_m = S_m^\top A \Sigma_0 A^\top S_m$$

A Problem

- To compute the posterior we must invert $\Lambda_m = S_m^\top A \Sigma_0 A^\top S_m$.
- Note that $(\Lambda_m)_{ij} = \langle s_i, s_j \rangle_{A\Sigma_0 A^\top}$.
- If we can construct search directions to be **orthonormal** in the $A\Sigma_0 A^\top$ inner product, the inverse is trivial.

Theorem (BayesCG)

Let $\tilde{s}_1 = r_0$ and

$$\tilde{s}_m = r_{m-1} - \langle s_{m-1}, r_{m-1} \rangle_{A\Sigma_0 A^\top} \times s_{m-1}$$

Then, after normalisation, $s_1, \ldots, s_m$ are $A\Sigma_0 A^\top$-orthonormal, and

$$x_m = x_{m-1} + \Sigma_{m-1} A^\top s_m \times s_m^\top r_{m-1}$$
$$\Sigma_m = \Sigma_{m-1} - \Sigma_{m-1} A^\top s_m s_m^\top A \Sigma_{m-1}$$

Cost

- $\mathcal{O}(md^2)$ computation (2-3 matrix-vector products per-iteration)
- $\mathcal{O}(md)$ storage (need to store search directions to compute Σ_m)

- **More costly than CG** - but comes with UQ.

Theorem (Krylov Subspace Method)

Let $K_m^\star = x_0 + \Sigma_0 A^\top K_m(A\Sigma_0 A^\top, r_0)$. Then:

$$x_m = \operatorname*{argmin}_{x \in K_m^\star} \|x - x^\star\|_{\Sigma_0^{-1}}$$

Note that setting $\Sigma_0 = A^{-1}$ reproduces CG!

Theorem (Rate of Convergence)

It holds that:

$$\frac{\|x_m - x^\star\|_{\Sigma_0^{-1}}}{\|x_0 - x^\star\|_{\Sigma_0^{-1}}} \leq 2 \left(\frac{\sqrt{\kappa(\Sigma_0 A^\top A)} - 1}{\sqrt{\kappa(\Sigma_0 A^\top A)} + 1} \right)^m$$

Fastest convergence when $\kappa(\Sigma_0 A^\top A) \approx 1$.

Experimental Results

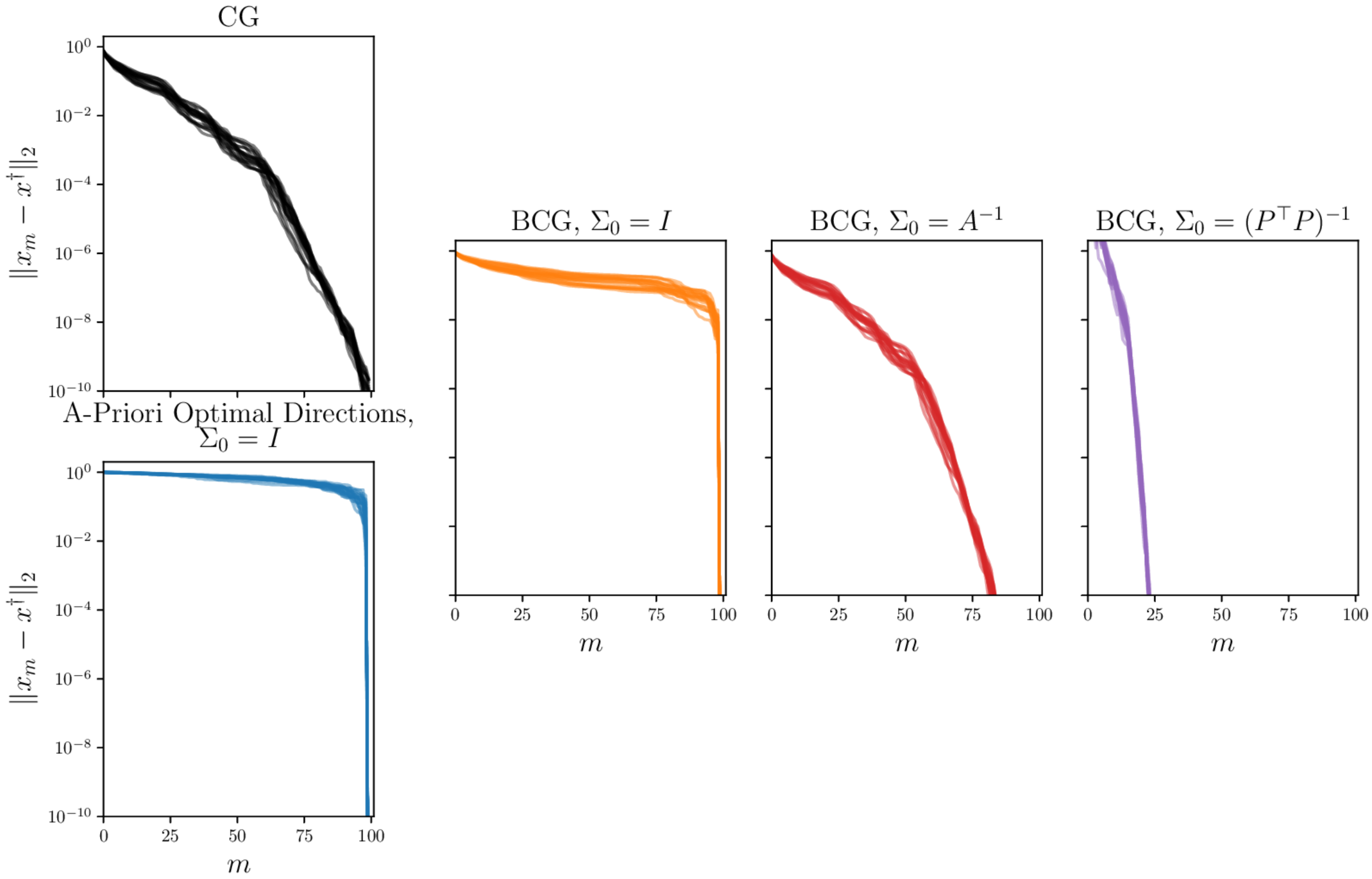
Priors Considered

- $\Sigma_0 = A^{-1}$ - replicates CG.
- $\Sigma_0 = I$ - “uninformative” prior.
- **Preconditioner Prior** - Given a preconditioner for A, take $\Sigma_0 = (P^\top P)^{-1}$.

(Left) preconditioner is a matrix P such that P^{-1} is easily computable, and $\kappa(P^{-1}A) \ll \kappa(A)$.

Experimental Setup

- A a random sparse matrix.
- $d = 100$
- Draw test problems $x^\star \sim \mathcal{N}(0,I)$.
- Apply BayesCG to $m = 100$.
- Compare to CG and “A-Priori Optimal” (essentially random) directions.


CG
A-Priori Optimal Directions, $\Sigma_0 = I$
BCG, $\Sigma_0 = I$
BCG, $\Sigma_0 = A^{-1}$
BCG, $\Sigma_0 = (P^\top P)^{-1}$
$\|x_m - x^\dagger\|_2$
m
10^0
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
0
25
50
75
100

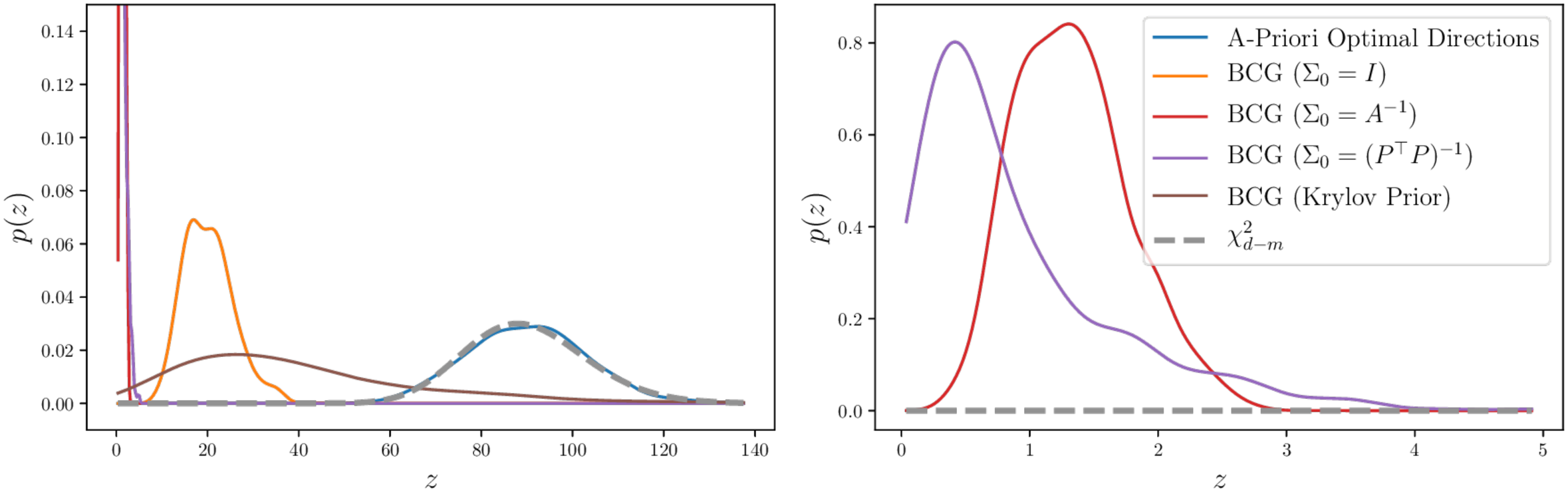
Posterior Calibration

- We say that the posterior is “well-calibrated” if $x^\star$ typically looks like a draw from the posterior.

- To assess this we compute the Z-statistic:

$$Z(x^\star) = \|x_m(x^\star) - x^\star\|^2_{\Sigma_m^\dagger(x^\star)}$$

- If the posterior is well-calibrated we can prove that $Z(X) \sim \chi^2_{d-m}$, when X is distributed according to the prior.


0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
$p(z)$
0
20
40
60
80
100
120
140
z
0.8
0.6
0.4
0.2
0.0
$p(z)$
0
1
2
3
4
5
z
A-Priori Optimal Directions
BCG ($\Sigma_0 = I$)
BCG ($\Sigma_0 = A^{-1}$)
BCG ($\Sigma_0 = (P^\top P)^{-1}$)
BCG (Krylov Prior)
χ^2_{d-m}

A Crime Against Bayes

- When we applied Bayes theorem we cheated!

$$s_m = r_{m-1} - \langle s_{m-1}, r_{m-1} \rangle \times s_{m-1}$$
$$r_{m-1} = b - Ax_{m-1} = A(x^\star - x_m)$$

so

$$s_m^\top b = s_m (x^\star)^\top A x^\star$$

Conclusions

- Mitigating poor UQ using (e.g.) empirical Bayes.
- Using BayesCG in applications (e.g. IterGP).

Thanks!