Polynomial Preconditioning for Gradient Methods

EPFL UCLouvain

Nikita Doikov Anton Rodomanov

Optimization Problem

We want to solve **unconstrained minimization** problem:

$$\min_{x \in \mathbb{R}^d} f(x)$$

where f is convex and differentiable

Assumption:

$$\mu B \preceq \nabla^2 f(x) \preceq LB, \qquad \forall x \in \mathbb{R}^d \tag{1}$$

for some $0 \leq \mu \leq L$.

- $B = B^\top \succ 0$ is the **curvature matrix** of size $d \times d$
- $B := I$ (no curvature) $\Rightarrow$ the standard class of strongly convex functions with Lipschitz gradient

Goal: Efficient methods with **cheap iterations** and **provable guarantees** that employ the curvature matrix $B \approx \nabla^2 f(x)$

Examples

Example: Quadratic function

$$f(x) = \tfrac{1}{2}\langle Bx, x\rangle - \langle a, x\rangle$$

satisfies (1) with $L = \mu = 1$

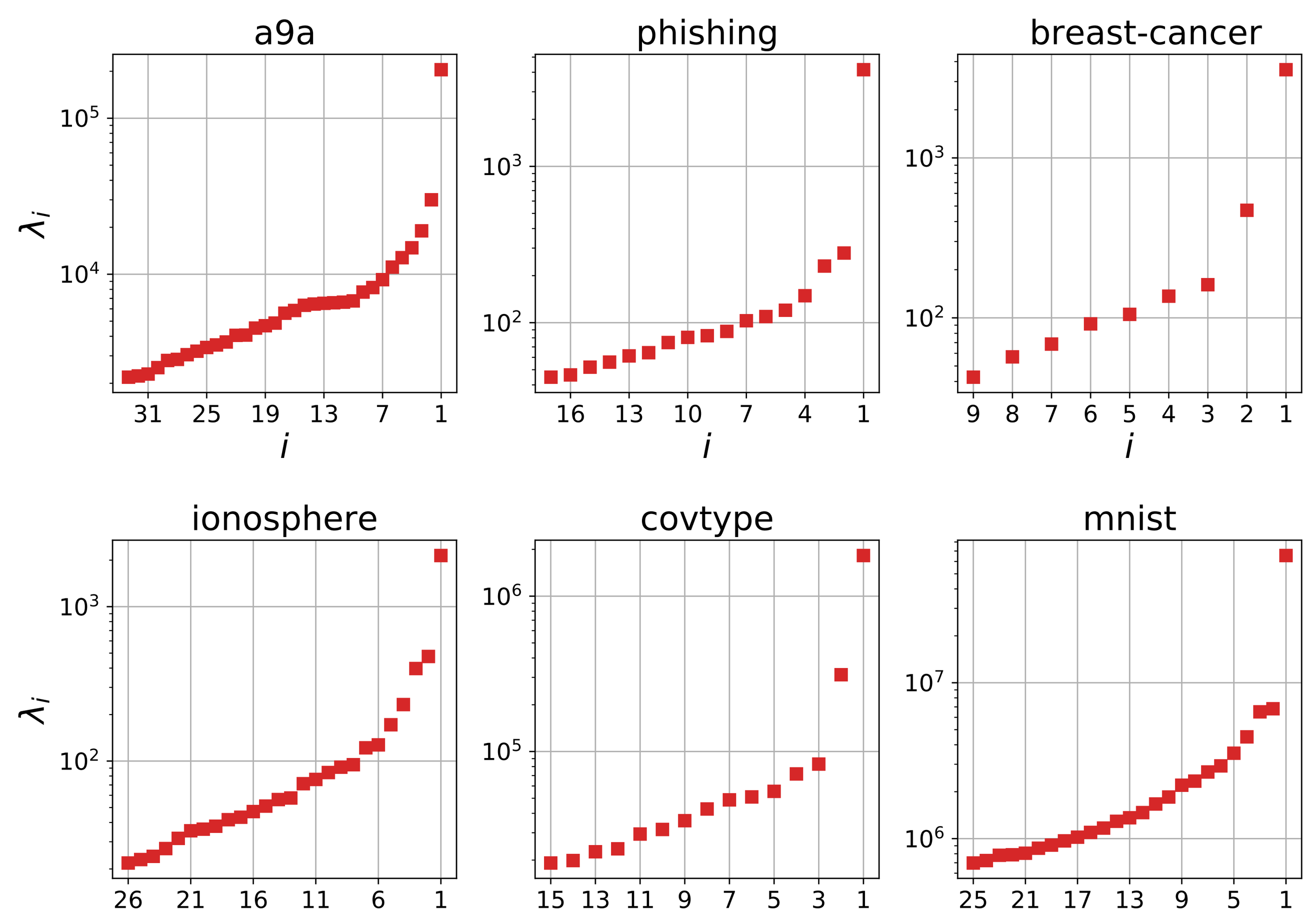
Example: let $f(x) = g(Ax + b)$ with $g(\cdot)$ s.t. $\mu I \preceq \nabla^2 g(x) \preceq LI, \forall x$.
Then (1) is satisfied with

$$B := A^\top A$$

Example: let $f(x) = \frac{1}{m}\sum_{i=1}^{m} \phi(\langle a_i, x\rangle)$, where $\{a_i\}_{i=1}^m$ are given data vectors (**Logistic Regression:** $\phi(t) = log(1 + e^t)$). Then, we can use

$$B := \sum_{i=1}^{m} a_i a_i^T$$

Leading Eigenvalues of the Curvature Matrix



- We observe **large gaps** between the top eigenvalues

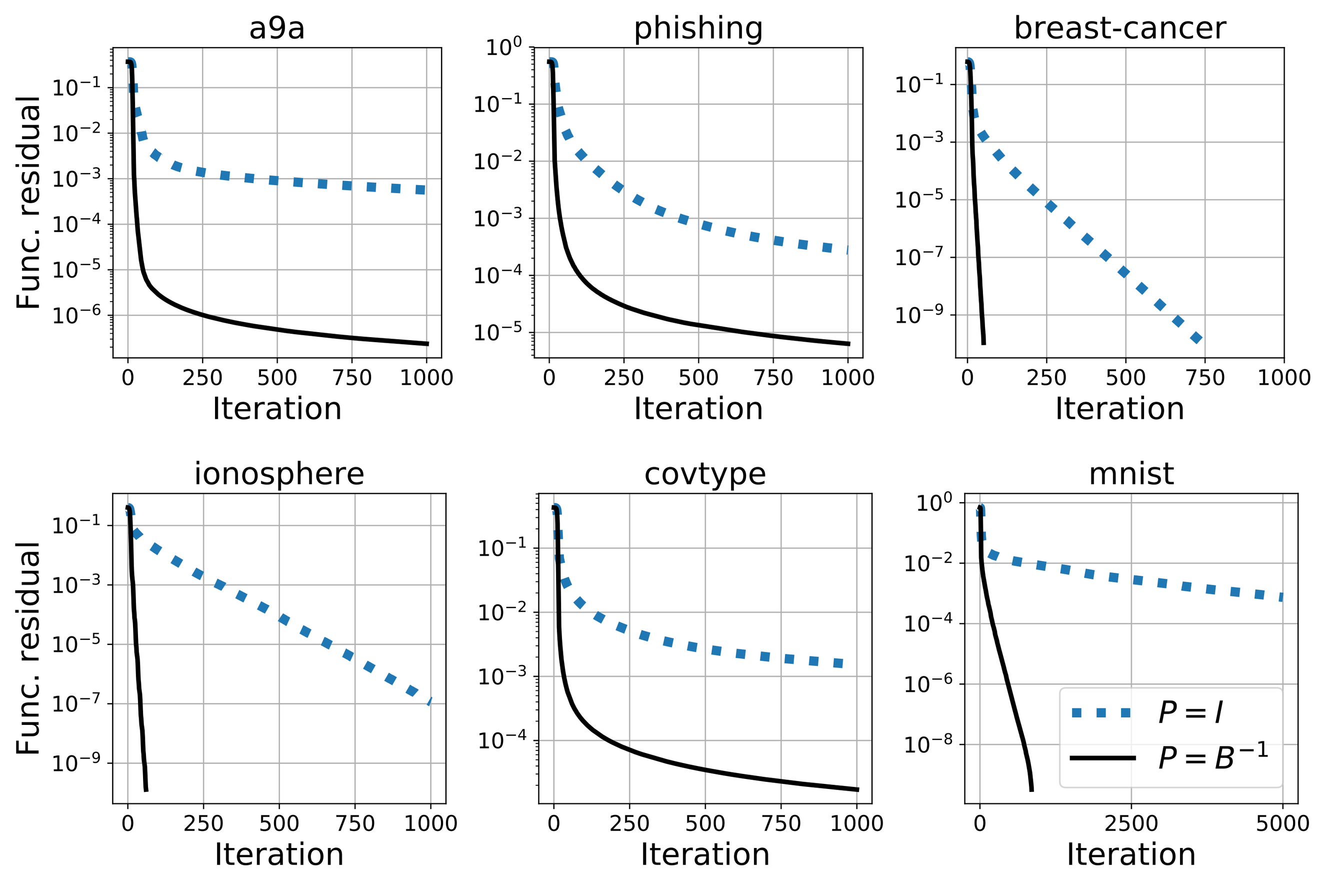
Preconditioned Gradient Method

The Basic Method:

$$x_{k+1} = x_k - \alpha_k P \nabla f(x_k), \qquad k \geq 0 \tag{2}$$

where $\alpha_k > 0$ is a stepsize and $P = P^\top \succ 0$ is a fixed **preconditioned matrix**

- The classical gradient descent: $P := I$
- Ideally: $P \approx B^{-1}$



- Using $P = B^{-1}$ significantly improves the convergence, howere it is **expensive to compute** for large-scale problems

Symmetric Polynomial Preconditioners

New family of preconditioners P_τ, indexed by $0 \leq \tau \leq d - 1$

Define: $P_0 \stackrel{\text{def}}{=} I$, and

$$P_\tau \stackrel{\text{def}}{=} \frac{1}{\tau}\sum_{i=1}^{\tau}(-1)^{i-1} P_{\tau-i} U_i, \tag{3}$$

where $U_\tau \stackrel{\text{def}}{=} \mathrm{tr}(B^\tau) I - B^\tau$.

We have

$$P_1 = \mathrm{tr}(B) I - B,$$
$$P_2 = \tfrac{1}{2}\mathrm{tr}(P_1 B) I - P_1 B$$
$$\cdots$$
$$P_{d-1} = \det(B) B^{-1} = \mathrm{Adj}(B)$$

- **Easy to use** for small τ (it requires τ matrix-vector products and evaluating $\mathrm{tr}(B^\tau)$)
- Gradually interpolates between $P = I$ to $P \sim B^{-1}$ (up to a constant factor)

Main Lemma. Let $B = Q\mathrm{Diag}(\lambda)Q^\top$ be the spectral decomposition. Then,

$$P_\tau = Q\mathrm{Diag}(\sigma_\tau(\lambda_{-1}), \ldots, \sigma_\tau(\lambda_{-n}))Q^\top,$$

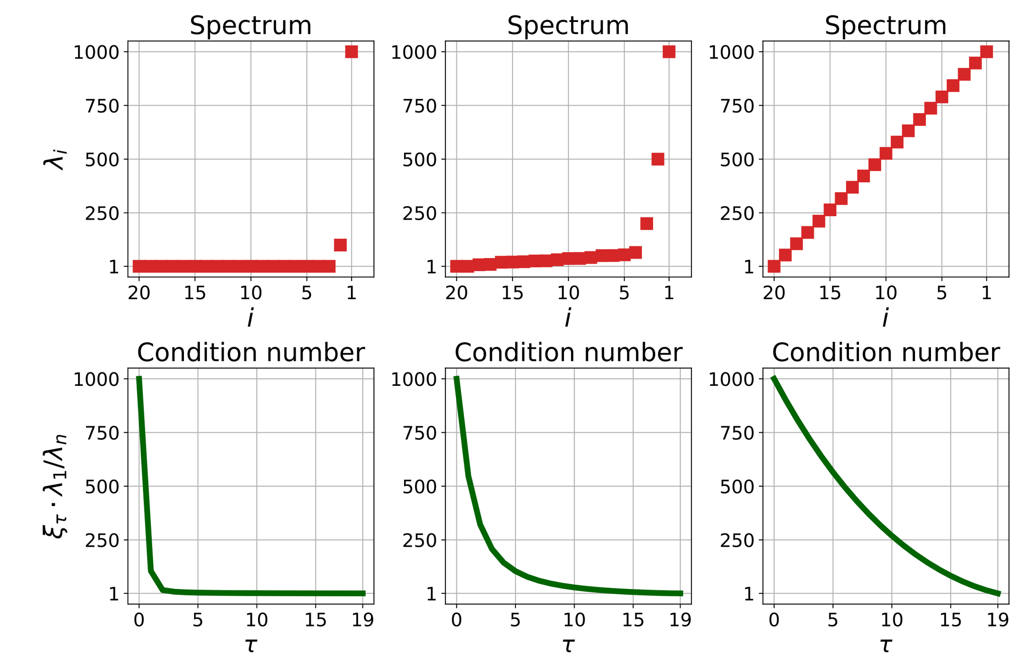
where $\lambda_{-i} \in \mathbb{R}^{n-1}$ is the vector of all eigenvalues except λ_i and $\sigma_\tau(\cdot)$ is the **elementary symmetric polynomial**

Main Properties

For any τ, we have $\alpha_\tau B^{-1} \preceq P_\tau \preceq \beta_\tau B^{-1}$ with **condition number**

$$\frac{\beta_\tau}{\alpha_\tau} = \frac{\lambda_1}{\lambda_n} \cdot \xi_\tau(\lambda), \quad \text{where} \quad \xi_\tau \stackrel{\text{def}}{=} \frac{\sigma_\tau(\lambda_{-1})}{\sigma_\tau(\lambda_{-n})}$$

- $\xi_\tau(\lambda) \leq 1$ is an **improvement** of the condition number by using Symmetric Polynomial Preconditioner of order τ
- $\xi_0(\lambda) = 1, \xi_{n-1}(\lambda) = \frac{\lambda_n}{\lambda_1}$
- $\xi_\tau(\lambda)$ **monotonically decreases** with τ
- $\xi_\tau(\lambda) \to 0$ when $\frac{\lambda_1}{\lambda_{\tau+1}} \to \infty$
- Explicit bound: $\xi_\tau(\lambda) \leq \left(\sum_{i=\tau+1}^{n} \lambda_i\right) / \left(\lambda_1 + \sum_{i=\tau+1}^{n-1} \lambda_i\right)$



Provably improves the condition number in case of **large gaps** between the top eigenvalues

Global Complexity

Theorem. The Gradient Method (2) with preconditioner (3) of order τ:

$$K = \mathcal{O}\Big(\tfrac{L}{\mu} \cdot \tfrac{\lambda_1}{\lambda_n} \cdot \xi_\tau(\lambda) \cdot \log \tfrac{1}{\varepsilon}\Big)$$

iterations. Using the Fast Gradient Method [Nesterov, 1983], we get

$$K = \mathcal{O}\Big(\sqrt{\tfrac{L}{\mu} \cdot \tfrac{\lambda_1}{\lambda_n} \cdot \xi_\tau(\lambda)} \cdot \log \tfrac{1}{\varepsilon}\Big)$$

Experiments