The error accumulation in the conjugate gradient method for degenerate problem

Anton Ryabtsev

Optimization Class Project. MIPT

Introduction

Many problems coming from real applications turn out to be degenerate. Build algorithms converging in argument for such tasks in general case turns out to be impossible. The solution of the problem turns out to be unstable to data inaccuracies.

Original problem

We confine ourselves to the simplest task:

$$Ax = b,$$

where exact values of A and b are not available [1], but only $\tilde{A}$ and $\tilde{b}$ are available, where

$$\|\tilde{A} - A\| < \delta_A, \quad \|\tilde{b} - b\| < \delta_b,$$

where $\|C\|_2 = \sqrt{\lambda_{max}(C^T C)}$.

According to the task, you can build the following optimization task:

$$f(x) = \frac{1}{2}\langle \tilde{A}x, x\rangle - \langle \tilde{b}, x\rangle \to \min_{x \in R^n}$$

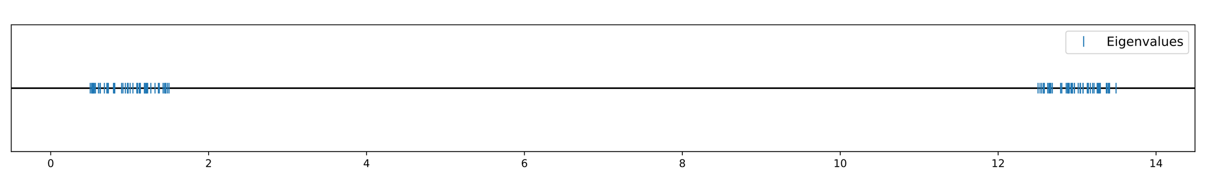
In the case of non symmetric matrix we can make a replacement:

$$A_{simmetric} = \frac{A + A^T}{2}$$

Such a replacement leads to an equivalent minimization, since the value of the objective function will not change from such a replacement.

Generation of ill-conditioned matrix for tests

We choose two numbers (small and big) and then we choose $\frac{N}{2}$ numbers in ϵ-neighborhood. In this way we get a spectre for the matrix with big condition number.



Noisy A matrix generation

$$\|\tilde{A} - A\|_2 < \delta_A$$

$$\tilde{A} = A + \delta \cdot I,$$

$$\tilde{A} - A = \delta \cdot I$$

$$\|\delta \cdot I\|_2 < \delta_A$$

$$0 < \delta < \delta_A$$

$$\delta = random.uniform(0, \delta_A) \quad or \quad \delta = random.gauss(0, \frac{\delta_A}{3})$$

Noisy b-vector generation

$$\|b\|_2 = \sqrt{b_1^2 + b_2^2 + \ldots + b_n^2}$$

$$\|\tilde{b} - b\|_2 = \sqrt{(\tilde{b}_1 - b_1)^2 + (\tilde{b}_2 - b_2)^2 + \ldots + (\tilde{b}_n - b_n)^2} < \delta_b$$

$$\|\tilde{b} - b\|_2 = \sqrt{\Delta_1^2 + \Delta_2^2 + \ldots + \Delta_n^2} < \delta_b$$

$$\Delta_1^2 + \Delta_2^2 + \ldots + \Delta_n^2 < \delta_b^2$$

$$\Delta_1 = random.uniform(-\delta_b, \delta_b)$$

$$\Delta_2 = random.uniform(-\sqrt{\delta_b^2 - \Delta_1^2}, \sqrt{\delta_b^2 - \Delta_1^2})$$

$$\Delta_3 = random.uniform(-\sqrt{\delta_b^2 - \Delta_1^2 - \Delta_2^2}, \sqrt{\delta_b^2 - \Delta_1^2 - \Delta_2^2})$$

...

Algorithm

We solve this problem using CG-method:

1. Let $i = 0$ and $x_i = x_0$, assume $d_i = d_0 = -\nabla f(x_0)$.
2. We calculate α minimizing $f(x_i + \alpha_i d_i)$ using the formula

$$\alpha_i = -\frac{d_i^T(Ax_i + b)}{d_i^T A d_i}$$

3. Make the algorithm step:

$$x_{i+1} = x_i + \alpha_i d_i$$

4. Update the direction: $d_{i+1} = -\nabla f(x_{i+1}) + \beta_i d_i$, where β_i is calculated by the formula:

$$\beta_i = \frac{\nabla f(x_{i+1})^T A d_i}{d_i^T A d_i}$$

5. Repeat steps 2-4 until the stopping criterion is fulfilled.

Hypothesis

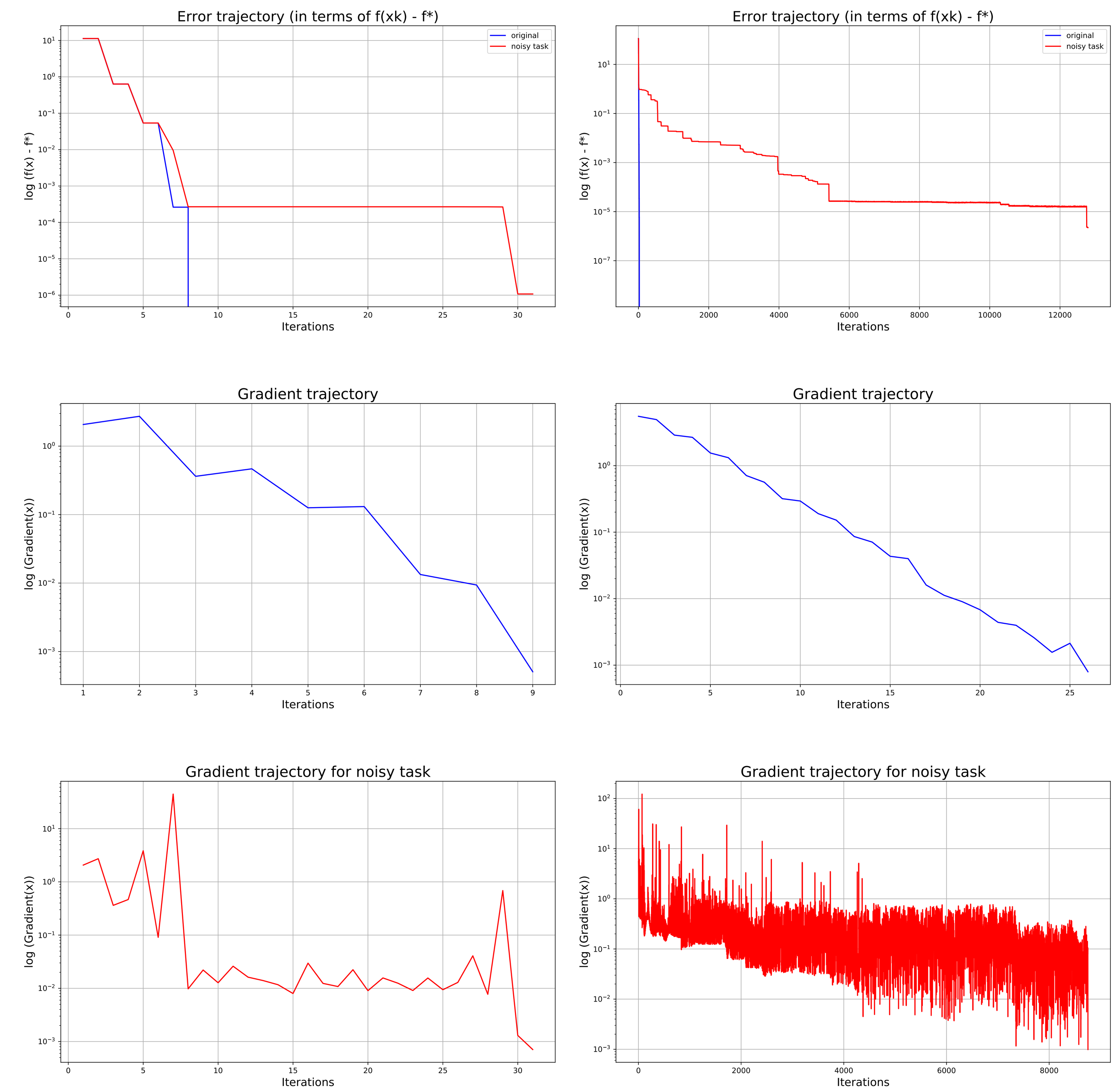
For conjugate gradient method, as well as for their analogues — accelerated methods, we can assume that if the noise in the calculation of the gradient is additive, then it does not accumulate [2]. However, in our case the noise in the gradient is not quite additive (with an increase in the norm of x, it should increase proportionally). But if the generated sequence is limited, then it can be considered as such.

Numerical example

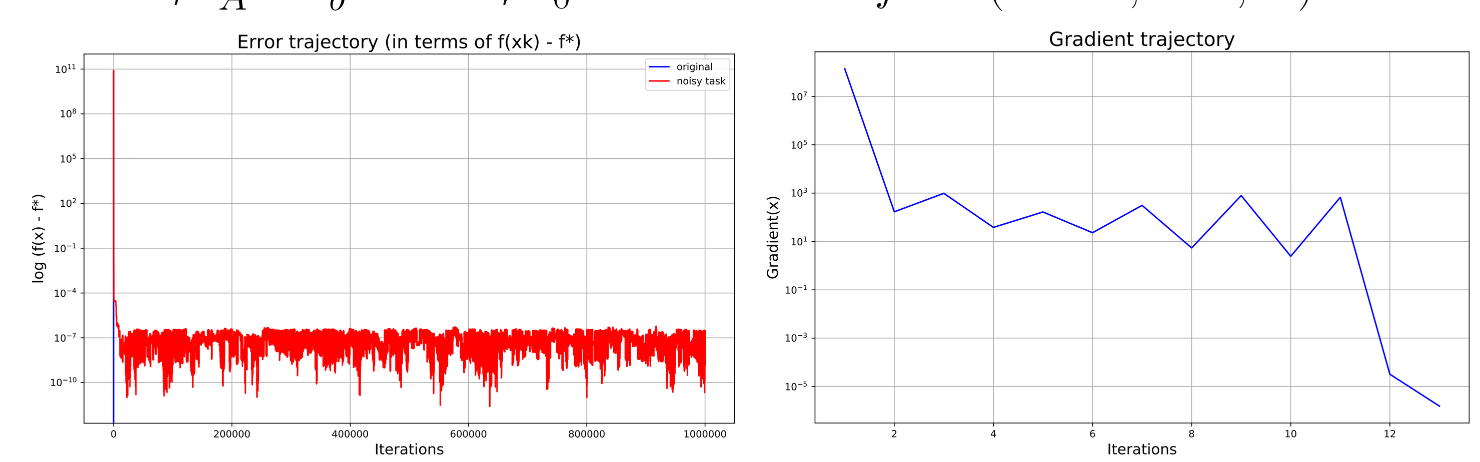
Consider a numerical example with $f(x) = \frac{1}{2}\langle \tilde{A}x, x\rangle - \langle \tilde{b}, x\rangle \to \min_{x \in R^n}$ with $A \in \mathbf{R}^{N \times N}$ and $b \in \mathbf{R}^N$. Entries of A and b are generated as independent samples from a uniform distribution. A is symmetric and positive definite. **Break criterion:** The gradient norm at the point x_{k+1} is less than 10^{-3} or more than $100 \cdot N$ iterations passed.

(a) $N = 10$, $\delta_A = \delta_b = 10^{-5}$, $x_0 = \vec{0}$
Decision error: $1.08 \cdot 10^{-6}$

(b) $N = 100$, $\delta_A = \delta_b = 10^{-5}$, $x_0 = \vec{0}$
Decision error: $2.23 \cdot 10^{-6}$



$N = 10$, $\delta_A = \delta_b = 10^{-5}$, $x_0 = random.uniform(-1000, 1000, N)$



Decision error: $1.12 \cdot 10^{-7}$

Results

A series of experiments were carried out with different sizes of matrices and different noises. There is reason to believe that with noise, less than a certain value, the method will converge anyway, and the faster, the less noise. You can see code here [3].

References

[1] Gasnikov Alexander. Modern numerical optimization methods. universal gradient descent method. *M.: MIPT*, 2018.

[2] Darina Dvinskikh and Alexander Gasnikov. Decentralized and parallelized primal and dual accelerated methods for stochastic convex programming problems. *arXiv preprint arXiv:1904.09015*, 2019.

[3] https://github.com/kosmo-tony/optimization-methods.