# A NOVEL METHOD TO FIND THE EXTREME SINGULAR VALUES

A PREPRINT

**James Chok**
School of Mathematics and
Maxwell Institute for Mathematical Sciences,
The University of Edinburgh,
Edinburgh,
EH9 3FD,
United Kingdom
james.chok@ed.ac.uk

February 7, 2025

**ABSTRACT**

In this brief note, I present a novel method to find the extreme singular values based on the theoretical properties of Nesterov's descent. The maximum singular value can be found using projected gradient descent with an unbounded learning rate that yields exponential convergence. Once the maximum singular value is known, the minimum singular value can be found using projected Nesterov's descent which has an optimum learning rate based on the maximum singular value, giving quadratic convergence.

***Keywords*** Singular Values

## 1 Introduction

In this note, we wish to find the maximum and minimum eigenvalues/eigenvectors in magnitude of a matrix $B$. This is equivalent to finding the maximum/minimum eigenvalues/eigenvectors of $A = BB^T$. In particular, $\lambda_{\max}(A) = \lambda^2_{\max}(B)$. As such, we look at finding the maximum and minimum eigenvalues of a symmetric, semi-positive definite matrix $A$. This can be found by maximizing/minimizing the following function

$$f(x) = x^T Ax, \qquad \text{for} \qquad x^T x = 1. \tag{1}$$

LOBPCG is a popular algorithm that uses gradient ascent (or descent) to find the maximum (or minimum) eigenvalues. LOBPCG uses a locally optimum approach in which each iteration performs a line search or a Rayleigh-Ritz method to guarantee each iteration increases (or decreases).

However, for a convex function $f$ with $L$-Lipshitz gradient, it is well known that the optimum learning rate of gradient descent and Nesterov accelerated gradient descent is $1/L$. Since $A$ is semi-positive definite, $f$ is convex with a $L = 2\lambda_{\max}$-Lipshitz continuous gradient, where $\lambda_{\max}$ is the maximum eigenvalue of $A$.

Once $\lambda_{\max}$ is found, the optimum learning rate for gradient descent is known, allowing us to find $\lambda_{\min}$. Moreover, $1/2\lambda_{\max}$ will guarantee that the function value decreases at each iteration, providing a locally optimum algorithm. Thus, we need only look at how to find $\lambda_{\max}$.

Once found, if $\lambda_{\min} > 0$ (which can be assured by convergence properties of Nesterov descent), the problem becomes strongly convex. In turn, projected Nesterov descent can then be used to yield an exponentially converging algorithm for rest of the smallest eigenvalues.

## 2 Method

Once the maximum eigenvalue is found, standard analysis of the projected Nesterov descent yields $\eta = 1/2\lambda_{\max}$ as the optimum convergence rate. In particular, Nesterov descent is the fastest optimization scheme only using first-order information, $\nabla f$.

### 2.1 Approximation of Maximum Eigenvalue

A classical way to approximate the maximum eigenvalue is the power method. Given a random initial vector $x_0$, $A^n x_0 \to x_{\max}$, where $x_{\max}$ is the eigenvector corresponding to the $\lambda_{\max}$.

Another simple method, with negligible extra cost, is to perform projected gradient ascent. This gives the iterations

$$y_{t+1} \; = \; x_t \; + \; 2\alpha A x_t, \qquad x_{t+1} \; = \; y_{t+1}/\|y_{t+1}\|, \tag{2}$$

for some $\alpha > 0$. As with the power method, this has exponential convergence for any $\alpha > 0$.

**Remark 1.** *It may seem counter-intuitive that any $\alpha > 0$ will converge. However, since $A$ is semi-positive definite, $f(x) \geq 0$ and $f(x) \to \infty$ as $\|x\| \to \infty$. As such, maximization occurs when $\|y_{t+1}\| \to \infty$ which allows for any finite learning rate $\alpha$. It is the projection step that keeps this problem well-defined.*

### 2.2 On Finding the Minimum Eigenvalue

Note that $f$ is a $L$-Lipschitz function with $L = 2\lambda_{\max}$. In Lemma 1 and 2, we see that Nesterov gradient descent requires $L$ to yield an optimal convergence rate. However, the above algorithm only gives an approximation on the eigenvalue $\widetilde{\lambda}_{\max}$ with $0 < \lambda_{\max} - \widetilde{\lambda}_{\max} \ll 1$. Thus, to ensure convergence of Nesterov descent we upper bound the Lipschitz constant by $\widetilde{L} = 2c\widetilde{\lambda}_{\max}$, for $c \geq 1.01$, and taking $t = 1/\widetilde{L}$ yields a known convergence property

$$f(x_k) - f(x^*) \; \leq \; \frac{2 \cdot 4}{(k+1)^2 t}, \tag{3}$$

where the $4$ comes from the fact that $\max_{x,y \in \mathbb{S}^1} \|x - y\| = 2$.

### 2.3 Finding the other extreme eigenvalues

The above algorithms provide a method to produce approximations of the largest and smallest eigenvalues and their corresponding eigenvectors, $v_{\max}, v_{\min}$ respectively. To find the second largest eigenvector, one can apply the subspace projection method

$$\max_x f(x), \qquad \text{for} \qquad x^T x = 1, \text{ and } x^T v_{\max} = 0, \tag{4}$$

and $\|v_{\max}\| = 1$. Similarly defined for the minimum eigenvectors.

Applying projected gradient descent requires us to solve the projection step

$$\min_x \|x - y\|, \qquad x^T x = 1, \text{ and } x^T v_{\max} = 0. \tag{5}$$

The solution can be solved using Lagrange multipliers, giving the solution

$$x \; = \; \frac{y - (y^T v_{\max}) v_{\max}}{\|y - (y^T v_{\max}) v_{\max}\|}. \tag{6}$$

More generally, if $V = [v_1, \ldots, v_m] \in \mathbb{R}^{d \times m}$ with $V^T V = I_m$, and we require $x^T V = 0$, then the projection is given by

$$x \; = \; \frac{y - (y^T V) V^T}{\|y - (y^T V) V^T\|}. \tag{7}$$

## 3 Experiment

We test our solver on a positive definite matrix $A \in \mathbb{R}^{1000 \times 1000}$, with eigenvalues logarithmically spaced from $10^{-4}$ to $10^4$. Our program is written in Python and was run 10 times to compute the time taken. The error results can be seen in Fig. 1 and the times can be seen in Fig. 2.

Unsurprisingly, Nesterov descent (ND) provides the quickest method to compute the maximum eigenvalue. However, the results does not monotonically decrease to the true solution. Moreover, we see Gradient descent (GD) and Nesterov descent lie below the theoretical bounds.

## 4 Conclusion

I present a novel method to compute the minimum and maximum eigenvalues. The method relies on using Gradient ascent to find the maximum eigenvalue, which allows one to approximate the optimum learning rate for Nesterov descent to find the minimum eigenvalue. Since these descent methods require only simple matrix operators, GPUs can be used to accelerate the runtime.

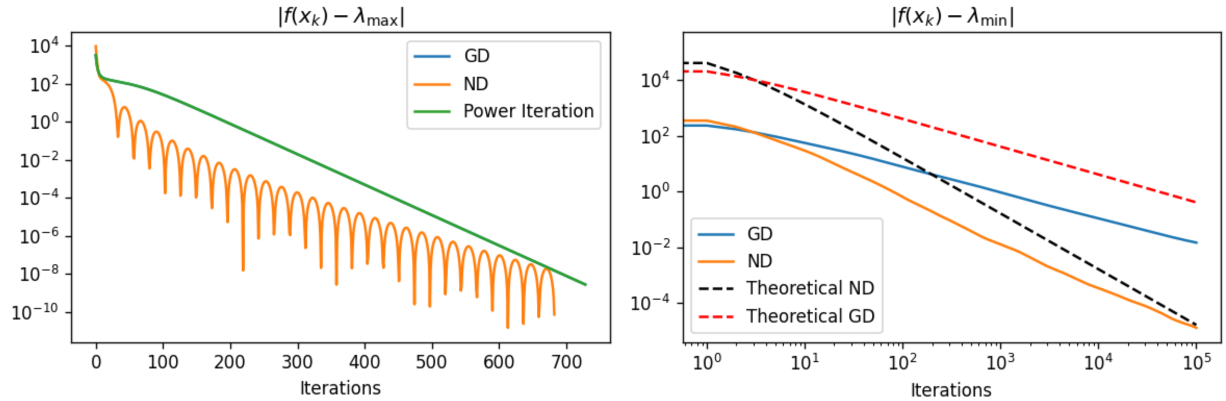


Figure 1: The approximation error to the maximum and minimum eigenvalues of a $1000 \times 1000$ matrix with eigenvalues logarithmically spaced from $10^{-4}$ to $10^{4}$. The theoretical bounds when finding the minimum eigenvalues is plotted in dotted lines.

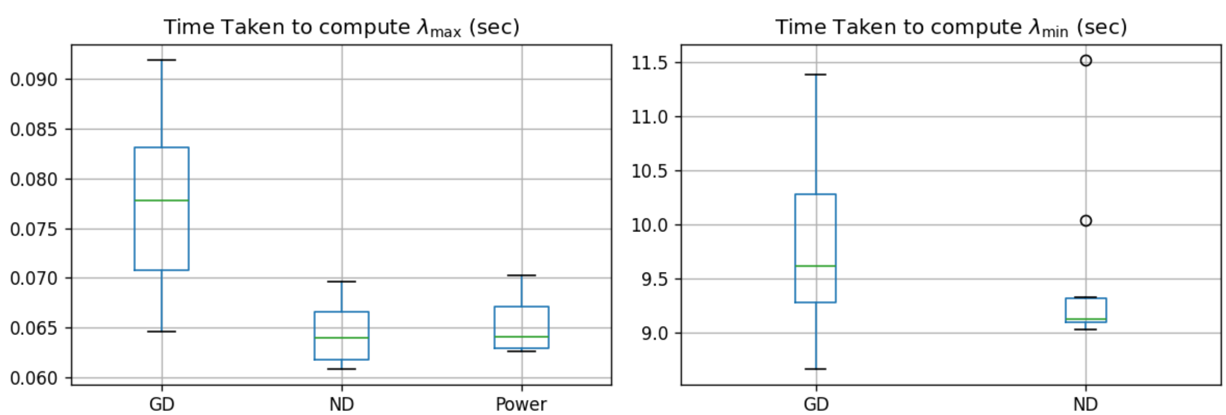


Figure 2: Time taken to compute the maximum and minimum eigenvalues of a $1000 \times 1000$ matrix.

## References

## A Lemmas on Nesterov Descent

**Lemma 1.** *Let $f$ be a $L$-Lipschitz differentiable real-valued convex function. Let $t \in (0, 1/L]$ with $\theta_k = 2/(k+2)$. Nesterov's gradient descent is given by*

$$y = (1-\theta_k)x_k + \theta_k v_k \tag{8}$$

$$x_{k+1} = y - t\nabla f(y) \tag{9}$$

$$v_{k+1} = x_k + \frac{1}{\theta_k}(x_{k+1} - x_k) \tag{10}$$

*where $x_0 \in \mathbb{R}^n$, $\theta_0 = 1$, $v_0 = x_0$. These iterates satisfy*

$$f(x_k) - f(x^*) \leq \frac{2}{(k+1)^2 t}\|x_0 - x^*\|^2, \tag{11}$$

*for all $k \geq 1$.*

**Lemma 2.** *Let $f$ be a $L$-Lipschitz, $m$-strongly convex, differentiable real-valued function. Let $t = 2/(m+L)$, Nesterov's gradient descent is given by*

$$y = x_k + \frac{1-\sqrt{m/L}}{1+\sqrt{m/L}}(x_k - x_{k-1}) \tag{12}$$

$$x_{k+1} = y - (1/L)\nabla f(y), \tag{13}$$

*where $x_0 \in \mathbb{R}^n$, $x_1 = x_0 - (1/L)\nabla f(x_0)$. These iterates satisfy*

$$f(x_k) - f(x^*) \leq \frac{m+L}{2}\|x_0 - x^*\|^2 \left(1 - \sqrt{\frac{m}{L}}\right)^k \tag{14}$$

*for all $k \geq 1$, and some constant $C$ that depends on $m$ and $L$.*