Speeding up Gradient Descent

1 Introduction and Background

Gradient descent has been the optimization algorithm of choice in a large amount of optimization applications, especially those which operate at large scales (where Hessians are too difficult to repeatedly compute). Yet gradient descent is not the fastest converging algorithm, even compared to other algorithms which have just zeroth and first-order information (i.e., knowledge of the function and its gradient). Indeed, Nesterov showed that it is possible to design a sped-up version of gradient descent, called *accelerated gradient descent*, which has asymptotically faster convergence rate than gradient descent [1].[1] This accelerated method is, at least at first, seemingly complicated and unmotivated; the original proof of convergence rate given by Nesterov amounted to a set of mysterious algebraic manipulations. Even one of the most well-known optimization theorists, Sebastian Bubeck, posted on his blog about how understanding Nesterov's acceleration is difficult to understand.

Recently, there have been several lines of research attempting to understand and extend Nesterov's acceleration. One of these attempts uses a generalization of gradient descent called *mirror descent* combined with gradient descent to improve the convergence rate of the algorithm [3], giving an interpretable modification of gradient descent which achieves optimal convergence rates. In this project, we introduce mirror descent and apply it to accelerate gradient descent.

2 Problems

1. Overview of Relevant Literature

In this problem, you will read two research papers: Nesterov's original accelerated gradient descent, and a follow-up which aims to achieve the same convergence rate by mixing gradient descent and mirror descent. The aim is to gain a better understanding of the topics discussed in this project and to get insights into the state-of-the-art development in our understanding of accelerating gradient descent. You will summarize the main results and findings of each paper and answer a few questions about them.

(a) Read and summarize the main results in the paper "A Method of Solving a Convex Programming Problem with Convergence Rate $O(1/k^2)$" by Nesterov [1]. In your summary include answers to the following questions:

i. Describe the main assumptions made in the paper on the function f to optimize. What kinds of functions are shown to be optimized via accelerated gradient descent? (*HINT:* In this paper, Nesterov uses the notation f' to denote the gradient ∇f.)

ii. What is the update rule at the k^{th} step?

iii. After T iterations, what is an upper-bound on $f(\vec{x}_T) - \min_{\vec{x}\in\mathbb{R}^n} f(\vec{x})$?

(b) Read and summarize the main results in the paper "Linear Coupling: An Ultimate Unification of Gradient and Mirror Descent" by Zhu and Orecchia. In your summary include answers to the following questions:

i. What do the authors claim is the kind of progress made by gradient descent? What about mirror descent?

[1] In fact, it achieves asymptotically optimal convergence rates; a proof of this is contained in [2].

ii. Describe the main assumptions made in the paper on the function f to optimize. What kinds of functions are shown to be optimized via the algorithm `AGM` proposed by Zhu and Orecchia?

iii. After T iterations of `AGM`, what is an upper-bound on $f(\vec{y}_T) - \min_{\vec{x} \in Q} f(\vec{x})$?

iv. What is the relationship between this paper [3] and Nesterov's original accelerated gradient descent paper [1]?

2. Mirror Descent

In this problem, we will go through a proof of mirror descent in the case of entropy and ℓ^2 regularization.

Consider the problem

$$f^\star = \min_{\vec{x}\in\mathcal{X}} f(\vec{x}). \tag{1}$$

where $\mathcal{X} \subseteq \mathbb{R}^n$ is a convex subset of $\mathbb{R}^n$. Often, running (projected) gradient descent is not the right thing to do and can lead to slow convergence. This is usually because the convergence rates of Euclidean gradient descent are of the form $f(\vec{x}_k) - f^\star \leq O(L\|\vec{x}_0 - \vec{x}^\star\|_2^2/k)$ and it may be the case for $\mathcal{X}$ that the ℓ^2 distance of the initial point from the optimal and/or the smoothness parameter L that's measured according to the ℓ^2 may be quite large. But taking the "right geometry" into account when *defining the update step* may lead to much faster convergence. For the sake of simplicity, however, for all except one subpart, we will only work for the ℓ^2 case in this problem.

For any convex (doubly)-differentiable function $h : \mathcal{X} \to \mathbb{R}$, define the Bregman divergence of h as

$$D_h(\vec{y};\vec{x}) = h(\vec{y}) - h(\vec{x}) - \langle \nabla h(\vec{x}), \vec{y} - \vec{x}\rangle \tag{2}$$

where $\langle \vec{u}, \vec{v}\rangle$ is the dot product between two vectors $\vec{u}$ and $\vec{v}$.

Algorithm 1 Mirror Descent Algorithm

$\vec{x}_0$ is a uniformly random point in $\mathcal{X}$
$k = 0$
while $k \leq T$ **do**
  $\eta_k > 0$ a step size.
  $\vec{g}_k \leftarrow \nabla f(\vec{x}_k)$
  $\vec{x}_{k+1} = \text{Mirr}(\eta_k \vec{g}_k; \vec{x}_k)$ where $\text{Mirr}(\vec{g};\vec{x}) = \operatorname{argmin}_{\vec{z}\in\mathcal{X}} \{\langle \vec{g}, \vec{z}\rangle + D_h(\vec{z};\vec{x})\}$ is the Mirror Descent step.
  $k \leftarrow k + 1$
end while
return $\overline{x}_T = \frac{1}{T}\sum_{k=0}^{T} \vec{x}_k$

(a) Prove that for any convex set $\mathcal{X}$ and α-strongly convex function $h : \mathcal{X} \to \mathbb{R}$, for any fixed $\vec{x} \in \mathcal{X}$, the Bregman divergence $D_h(\vec{y};\vec{x})$ is a α-strongly convex function of $\vec{y}$. Note that by taking $\alpha = 0$, this proves that if h is convex, the Bregman divergence is convex as well.

(b) Something that's going to be useful in a convergence proof of mirror descent is going to be the so-called Bregman three-point inequality. Formally, prove that

$$\langle \nabla h(\vec{x}) - \nabla h(\vec{y}), \vec{y} - \vec{u}\rangle = D_h(\vec{u};\vec{x}) - D_h(\vec{u};\vec{y}) - D_h(\vec{y};\vec{x}) \tag{3}$$

(c) Let's try to understand the mirror descent update in some special cases. For this part, assume $\mathcal{X} = \{\vec{x} \in \mathbb{R}^n | x_i \geq 0 \ \forall i \in [n] \text{ and } \sum_{i=1}^n x_i = 1\}$ is the n-dimensional probability simplex. We will take $h(\vec{x}) = \sum_{i=1}^n (x_i \log(x_i) - x_i)$, the entropy function. Given $\vec{x} \in \mathcal{X}$ and given some $\vec{g} \in \mathbb{R}^n$ and $\eta > 0$, compute $\text{Mirr}(\eta\vec{g};\vec{x})$. Since $\mathcal{X}$ is constrained to be the simplex, you will have to use a Lagrange multiplier for the $\sum_{i=1}^n x_i = 1$ constraint and eliminate the Lagrange multiplier from the final solution. In this setting, this algorithm goes by a more popular name which is "multiplicative weights update method".

(d) Now, for the rest of the problem, for simplicity, we will assume $\mathcal{X} = \mathbb{R}^n$ and $h(\vec{x}) = \frac{1}{2}\|\vec{x}\|_2^2$. In this case, given a $\vec{g} \in \mathbb{R}^n$, $\eta > 0$ and $\vec{x} \in \mathcal{X}$, compute $\text{Mirr}(\eta\vec{g}; \vec{x})$. Also compute $D_h(\vec{y}, \vec{x})$ in this case. Do these look familiar to something you have already seen?

(e) To prove convergence of mirror descent, it's convenient to introduce a term from online learning, called regret. For any feasible solution $\vec{u} \in \mathcal{X}$, we define regret in the k^{th} iteration as $\text{Reg}_k(\vec{u}) = \langle \eta_k \vec{g}_k, \vec{x}_k - \vec{u} \rangle$. We will first prove an upper bound on the regret of the k^{th} iteration. Formally, prove that

$$\text{Reg}_k(\vec{u}) = \langle \eta_k \vec{g}_k, \vec{x}_k - \vec{u} \rangle = \langle \eta_k \vec{g}_k, \vec{x}_k - \vec{x}_{k+1} \rangle + D_h(\vec{u}; \vec{x}_k) - D_h(\vec{u}; \vec{x}_{k+1}) - D_h(\vec{x}_{k+1}; \vec{x}_k) \tag{4}$$

$$= \frac{\eta_k^2 \|\vec{g}_k\|_2^2}{2} + D_h(\vec{u}; \vec{x}_k) - D_h(\vec{u}; \vec{x}_{k+1}) \tag{5}$$

This inequality will show up again in the proof of accelerated gradient descent in another question in the project.

HINT: Think of the first term on the equality that has to be proven. Looking at that, maybe adding and subtracting from the regret may help?

(f) Now we will consider the total regret over T iterations, i.e., $\text{TotalReg}_T(\vec{u}) = \sum_{i=0}^{T} \langle \eta_k \vec{g}_k, \vec{x}_k - \vec{u} \rangle$. Prove that

$$\text{TotalReg}_T(\vec{u}) \leq \sum_{k=0}^{T} \eta_k^2 \|\vec{g}_k\|_2^2 + D_h(\vec{u}; \vec{x}_0) \tag{6}$$

(g) Now, we will prove a lower bound on the regret in terms of the function value at $\overline{x}_T$ and at $\vec{u}$. Taking $\eta_k = \eta$ for all k, prove that

$$\text{TotalReg}_T(\vec{u}) \geq T\eta(f(\overline{x}_T) - f(\vec{u})) \tag{7}$$

Using this, conclude that

$$f(\overline{x}_T) \leq f(\vec{x}^\star) + \frac{1}{T}\left[\eta \sum_{i=0}^{T} \|\vec{g}_k\|_2^2 + D_h(\vec{x}^\star; \vec{x}_0)/\eta\right] \tag{8}$$

(h) Now, assume that the function is L-Lipschitz (note that this is asking for the function to be Lipschitz and not the gradient of the function to be L-Lipschitz). It can be easily proven (and you may assume so without proof) that $\|\nabla f(\vec{x})\|_2 \leq L$ for all $\vec{x} \in \mathcal{X}$. Conclude that

$$f(\overline{x}_T) \leq f(\vec{x}^\star) + \eta L^2 + \frac{1}{2\eta T}\|\vec{x}_0 - \vec{x}^\star\|_2^2 \tag{9}$$

Show that there exists an $\eta > 0$ such that

$$f(\overline{x}_T) \leq f(\vec{x}^\star) + \frac{\sqrt{2}L\|\vec{x}_0 - \vec{x}^\star\|_2}{\sqrt{T}} \tag{10}$$

HINT: Can you try to optimize the total regret upper bound by optimizing it as a function of η? Hence the convergence is at a rate of $1/\sqrt{T}$.

(i) In the accompanying Jupyter notebook, you will implement the mirror descent update for the entropy regularizer and for the ℓ^2 regularizer. The input to the function will be step size η_k, a vector $\vec{g}$ which is meant to represent the gradient, and the current point $\vec{x}_k$.

Hence the convergence is at a rate of $1/\sqrt{T}$. While we did the proof for the unconstrained setting and with the ℓ^2 geometry, this proof with very few changes can be used to prove similar results in constrained settings and with other geometries, which show up, for example, in the probability simplex case corresponding to the multiplicative weights update method, whose mirror update you calculated above.

3. Accelerated Gradient Descent

In this problem, we will go through a proof of accelerated gradient descent by combining a gradient descent and a mirror descent step. You will also implement this algorithm in an accompanying Jupyter notebook and use it to optimize some specific functions.

Recall that in lecture, in the proof of gradient descent for L-smooth functions, we proved the following inequality:

$$f(\vec{x}_+) \leq f(\vec{x}) - \frac{1}{2L}\|\nabla f(\vec{x})\|_2^2 \tag{11}$$

where $\vec{x}_+ = \vec{x} - \frac{1}{L}\nabla f(\vec{x})$ is the gradient descent step. We will need this inequality as well as the inequality you proved in part (e) in the Mirror Descent problem which bounds the per iteration regret.

Let's now describe an accelerated gradient descent algorithm. We will work in the unconstrained optimization setting so that $\mathcal{X} = \mathbb{R}^n$ and will use the ℓ^2 geometry and we assume f is convex and L-smooth.

Algorithm 2 Acceleration via Combining Gradient and Mirror Descent

$\vec{x}_0 = \vec{y}_0 = \vec{z}_0$ is a uniformly random point in $\mathcal{X}$
$k = 0$
while $k \leq T$ **do**
  $\vec{x}_{k+1} = \tau_k \vec{z}_k + (1-\tau_k)\vec{y}_k$ for $\tau_k = 2/(k+2)$
  $\vec{y}_{k+1} \leftarrow x_{k+1} - \frac{1}{L}\nabla f(\vec{x}_{k+1})$
  $\vec{z}_{k+1} = \text{Mirr}(\eta_{k+1}\nabla f(\vec{x}_{k+1}); \vec{z}_k)$ where $\eta_{k+1} = (k+2)/2L = 1/(\tau_k L)$
  $k \leftarrow k+1$
end while
  return y_T

Here the h function defining the Bregman divergence for the mirror descent step is just $h(\vec{x}) = \frac{1}{2}\|x\|_2^2$.

(a) We first understand the regret on the mirror update. Formally prove that,

$$\langle \eta_{k+1}\nabla f(\vec{x}_{k+1}), \vec{z}_k - \vec{u}\rangle = \frac{\eta_{k+1}^2}{2}\|\nabla f(\vec{x}_{k+1})\|_2^2 + D(\vec{u};\vec{z}_k) - D(\vec{u};\vec{z}_{k+1}) \tag{12}$$

HINT: In part (e) of the mirror descent question, would the proof still work if $\vec{g}_k$ was something other than $\nabla f(\vec{z}_k)$?

(b) Now, we try to understand the regret of $\vec{x}_{k+1}$. Formally, prove that,

$$\langle \eta_{k+1}\nabla f(\vec{x}_{k+1}), \vec{x}_{k+1} - \vec{u}\rangle \tag{13}$$

$$= \frac{(1-\tau_k)\eta_{k+1}}{\tau_k}\langle \nabla f(\vec{x}_{k+1}), \vec{y}_k - \vec{x}_{k+1}\rangle + \frac{\eta_{k+1}^2}{2}\|\nabla f(\vec{x}_{k+1})\|_2^2 + D(\vec{u};\vec{z}_k) - D(\vec{u};\vec{z}_{k+1}). \tag{14}$$

Furthermore, show that one can upper bound the RHS above by

$$\frac{(1-\tau_k)\eta_{k+1}}{\tau_k}(f(\vec{y}_k) - f(\vec{x}_{k+1})) + \eta_{k+1}^2 L(f(\vec{x}_{k+1}) - f(\vec{y}_{k+1})) + D(\vec{u};\vec{z}_k) - D(\vec{u};\vec{z}_{k+1}). \tag{15}$$

HINT: In the $\vec{x}_{k+1} - \vec{u}$ term, add and subtract $\vec{z}_k$ and use the previous part along with the definition of $\vec{x}_{k+1}$.

(c) Now deduce that

$$\eta_{k+1}^2 L f(\vec{y}_{k+1}) - (\eta_{k+1}^2 L - \eta_{k+1}) f(\vec{y}_k) - D(\vec{u}; \vec{z}_k) + D(\vec{u}; \vec{z}_{k+1}) \leq \eta_{k+1} f(\vec{u}) \tag{16}$$

HINT: You will need to use the specific values of η_k and τ_k as defined in the algorithm definition to observe some cancellations

(d) Now, summing up the inequality in the previous part and plugging in values for η_{k+1}, conclude that

$$f(\vec{y}_T) \leq f(\vec{x}^\star) + \frac{2L\|\vec{x}^\star - \vec{x}_0\|_2^2}{(T+1)^2} \tag{17}$$

(e) In the accompanying Jupyter notebook, implement the above acceleration via gradient plus mirror descent step. Run the algorithm on a given low rank quadratic optimization problem. Report how the algorithm performs as compared to Gradient Descent, Adam, Adagrad algorithms. In the logistic regression `fval` function, inside the logarithm terms, you should add a small epsilon like `1e-10` in order to ensure there are no NaNs in the output.

3 Rubric

Here's what the rubric looks like:

- To get a B: read and give a two-sentence summary for the papers in both problems and answer the questions associated to the two papers at the beginning of the problems, have a mostly correct implementation for problems 2(i) and 3(f), correctly solve any five parts from 2(a) — 2(h), and correctly solve any three parts from 3(a) — 3(e).
- To get a B+/A-: read and give a two-sentence summary for the papers in both problems, have a correct implementation for problems 2(i) and 3(f), correctly solve all of 2(a) — 2(h), and correctly solve all of 3(a) — 3(e).
- To get a A: complete all problems as stated above, plus one of the extensions below.

To get an A, you should complete one of the following extensions. You should include your extension(s) in a separate report that you attach to your project writeup. Exceptional projects that go above and beyond may receive extra credit at our discretion.

- Read the paper [4] and provide a summary of the key ideas of the SAM procedure, including especially why it may improve performance over gradient descent. Also, provide an implementation of the SAM procedure, including a performance evaluation on the benchmark functions used in the main project.
- Read the paper [5] and provide a summary of the key ideas of the optimal algorithm given in the paper, including especially the geometric intuition of what happens at each step of the algorithm. Also, provide an implementation of this algorithm, including a performance evaluation on the benchmark functions used in the main project.
- Read the paper [6] which shows how to solve min-max saddle point problems in special cases and summarize the algorithm and also why this algorithm (and the analysis) is called a mirror proximal algorithm. Implement the mirror prox algorithm in the special case which corresponds to Example 1 (page 17 of the paper) of matrix games.
- Propose your own extension of a similar level of interest and difficulty that you get approved by course staff.

3.1 Deliverables

Your submission should be in the form of one PDF with the following parts:

1. Your project writeup.
2. A PDF printout of the completed Jupyter notebook `main.ipynb`.
3. A report of any project extension you choose to complete. Your report should follow the template available on the course website under "Projects". The report must have a minimum of 1000 words and should include the following sections:
    - Introduction section: includes literature review of any relevant papers you summarized and identifies open problems in the understanding of the SAM procedure/accelerated gradient descent/optimization in min-max or other settings.

- Methodology section: includes description of the methodology you follow in the project extension you chose to implement. Make sure you include description of the algorithms, the assumptions on the function to be optimized, the error guarantees of the algorithms.
- Results section: summarizes the results you obtained from the project extension you implement. If you focused on an optimization algorithm, you should talk about the proofs of the theorem statements. You may use some of the visualizations from your project writeup for performance comparison for the algorithm implementation part.

4. A post-mortem survey about your project experience.

In addition to the PDF submission, please do submit any code or Jupyter notebook you write for any project extension.

References

[1] Y. E. Nesterov, "A method of solving a convex programming problem with convergence rate o\bigl(k^2\bigr)," in *Doklady Akademii Nauk*, Russian Academy of Sciences, vol. 269, 1983, pp. 543–547.

[2] Y. Nesterov *et al.*, *Lectures on convex optimization*. Springer, 2018, vol. 137.

[3] Z. Allen-Zhu and L. Orecchia, "Linear coupling: An ultimate unification of gradient and mirror descent," *arXiv preprint arXiv:1407.1537*, 2014.

[4] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, "Sharpness-aware minimization for efficiently improving generalization," *arXiv preprint arXiv:2010.01412*, 2020.

[5] S. Bubeck, Y. T. Lee, and M. Singh, "A geometric alternative to nesterov's accelerated gradient descent," *arXiv preprint arXiv:1506.08187*, 2015.

[6] A. Nemirovski, "Prox-method with rate of convergence o (1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems," *SIAM Journal on Optimization*, vol. 15, no. 1, pp. 229–251, 2004.

