A CONTINUOUS DYNAMICAL SYSTEM ASSOCIATED WITH ITERATIVE ALGORITHMS

YUHAO LI WITH SUPERVISOR MATTHEW TAM

INTRODUCTION

The objective of this project is to discover the properties and behaviors of iterative optimization algorithms such as gradient descent from the perspective of a continuous dynamical system to give a clearer interpretation and deeper understanding of the algorithm The problem is defined as:

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

where $f : \mathbb{R}^n \to \mathbb{R}$ is convex and differentiable. The iterative algorithms include gradient descent, the heavy ball with friction method, Nesterov acceleration, ravine method, and a new method called adaptive gradient descent without descent method [1] would be studied.

DYNAMICAL SYSTEM

There are two types of dynamical system.
Discrete system describes an iterative map (a sequence of x_n with $x_{n+1} = f(x_n)$ for some function f: $\mathbb{R}^n \to \mathbb{R}^n$)

Continuous system is represented by a dynamic(differential equation).The definitions of dynamics could be described as follows:

Definition of dynamic:
A dynamics is defined via a differential equation: $\dot{x}(t) = h(x(t)))$ where $h : \mathbb{R}^\kappa \to \mathbb{R}^\kappa$ is a vector field that assigns the velocity vector $f(x)$ to each point x in space, which specifies the next direction of a point(instantaneous future).

OTHER ALGORITHMS AND INTERPRETATIONS

Other than the first-order dynamical system in Gradient flow, second-order dynamical system is also commonly used in a variety of iterative algorithms.

Heavy ball with friction(HBF)
The heavy ball with friction system could be described as a second-order (in time) dissipative dynamical system, which has mechanical interpretations [2]. This system models the motion of a heavy material point $M(t) = (x(t), f(x(t)))$ sliding on f. By Fig.2 It could be proved that this method would converge to a better local minimum point x by skipping some of the local minimum points $\bar{x}$ due to introduction of momentum. Thus, the trajectories of this method might include damped oscillations before stabilizing.
Continuous Dynamics: $\ddot{x}(t) + \lambda\dot{x}(t) + \nabla f(x) = 0$, where $\dot{x}(0) = \dot{x}_0, x(0) = x_0, \lambda > 0$ as friction parameter that control the number of local minimas the heavy ball could reach asymptotically.

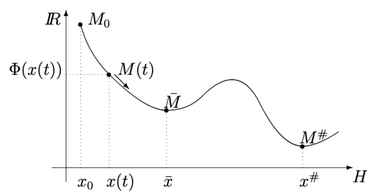


Figure 2: Graphical illustration of heavy ball with friction method [2]

Nesterov Acceleration(NAG)
Similar to the heavy ball with friction method, Nesterov purposed an accelerated gradient method that could result in an $\mathbb{O}(\frac{1}{k^2})$ for a convex optimization problem. This algorithm includes an extrapolation operation, followed by a gradient update step.

Ravine method(RAG)
By reversing the extrapolation and gradient step, the Ravine method is closely related to Nesterov accelerated method. The interpretation of this algorithm could be the flows of water in the mountains that firstly pass by steep ravines rapidly and then followed by flowing to the main branch in the valley by fig.3 [3].

The interpretation of this system could be as a mass-spring-damper system with a curvature-dependent damping term, that directly results in the acceleration and therefore superior performance of NAG[3].

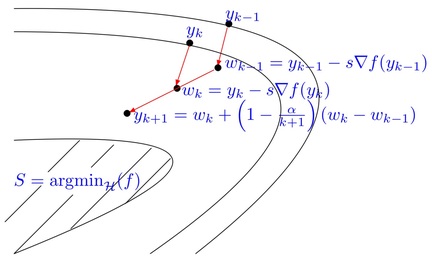


Figure 3: Geometrical demonstration of Ravine method(RAG) [3]

Continuous Dynamics: $\ddot{x}(t) + \frac{\alpha}{t}\dot{x}(t) + \nabla f(x) = 0$, where $\dot{x}(0) = \dot{x}_0, x(0) = x_0, \alpha > 0, t > 0$

Summary

System Name	Continuous Dynamical System
GF	$\dot{x}(t) = -\nabla f(x(t))$
HBF	$\ddot{x}(t) + \lambda\dot{x}(t) + \nabla f(x(t)) = 0$
NAG	$\ddot{x}(t) + \frac{\alpha}{t}\dot{x}(t) + \nabla f(x(t)) = 0$
RAG	$\ddot{x}(t) + \frac{\alpha}{t}\dot{x}(t) + \nabla f(x(t)) = 0$

where α is a positive parameter, $t > 0$,
initial condition $x(0) = x_0$
if system contain second order term $\dot{x}(0) = z_0$

Table 1: Algorithms and their continuous dynamical systems

DERIVING CONTINUOUS DYNAMICS AND DISCRETISATION

From discrete to continuous To begin with an example, the most classical iterative algorithm: Gradient Descent is used.
Gradient Descent: $x_{k+1} = x_k - \lambda\nabla f(x_k)$ where $\lambda > 0$ is step size
Suppose $x(t)$ is a continuous curve with $x(\lambda k) = x_k$, $x(t + \lambda) = x(t) - \lambda\nabla f(x(t))$
Rearranging to get $\frac{x(t+\lambda)-x(t)}{\lambda} = -\nabla f(x(t))$ take limit as $\lambda \to 0$ to get the continuous dynamical system of gradient descent: gradient flow.
Gradient Flow (GF): continuous dynamic of Gradient Descent $\dot{x}(t) = -\nabla f(x(t))$ with initial condition: $x(0) = x_0$.
Figure 1 further illustrates the difference between gradient flow and gradient descent by showing the trajectories of two algorithms that are experimented with for the same logistic regression optimization problem. The trajectory of gradient descent is less smooth than gradient flow.

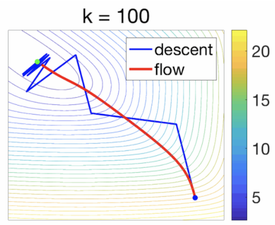


Figure 1: Comparison between Gradient Descent and Gradient flow from: https://francisbach.com/gradient-flows/

From continuous to discrete: Discretisation
After obtaining the continuous dynamics, various discretisation methods that discrete the differential equation could explore new optimisation method with similar convergence rate. There are mainly two types of discretisation method.
Forward Euler:

$$x(t_{k+1}) \approx x_{k+1} = x_k + h\nabla f(x_k, t_k)\forall k \in \mathbb{N}$$

By using forward euler discretization to gradient flow by rearranging equation, we could obtain back to the gradient descent algorithm
Backward Euler:

$$x(t_{k+1}) \approx x_{k+1} = x_k + h\nabla f(x_{k+1}, t_{k+1})\forall k \in \mathbb{N}$$

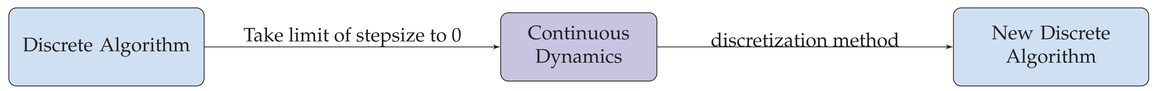
By using backward euler discretization that use gradient value of the next iteration, we could yield another algorithm **proximal point algorithm**. The derivation is provided as follows

$$x_{k+1} = x_k - \lambda\nabla f(x_{k+1})$$

$$x_{k+1} + \lambda\nabla f(x_{k+1}) = x_k$$

$$(I + \lambda\nabla(f))(x_{k+1}) = x_k$$

Since $I + \lambda\nabla(f)$ is invertible, and $(I + \lambda\nabla(f))^{-1}$ is equivalent to proximity operator $\operatorname{prox}_{\lambda f}$
So $x_{k+1} = \operatorname{prox}_{\lambda f} x_k = \operatorname{argmin}_{y\in\mathbb{R}^n}\{f(y) + \frac{1}{2\lambda}(||y - x_k||)^2\}$



FUTURE WORK

The traditional gradient descent algorithm is not ideal for converging to a local minimum even for a convex objective function, which also not only requires a global Lipschitz condition to prevent the explosion of gradient size everywhere in the domain but also the appropriate choice of step size to prevent the possible slow convergence.

Adaptive Gradient Descent Without Descent Method
The algorithm in fig.4 was presented by Yura [1] in 2019, which replaced the global Lipschitz condition with a local Lipschitz condition so that more functions such as $tan(x)$ could be applied. Furthermore, it chooses the step size automatically by ensuring the energy of the system is decreasing for each iteration.

Work in progress
Due to the adaptive choice of step size and recursive relation between step size for each iteration, the continuous dynamics of this system are difficult to derive. It becomes an obstacle to understand why local Lipschitz is sufficient and find its mechanical interpretations without knowing the exact continuous dynamical system.
Apart from that, certain discretization methods that yield from a continuous dynamic of adaptive gradient descent method with similar convergence properties would also be the research goal in the future.

Algorithm 1 Adaptive gradient descent

Input: $x^0 \in \mathbb{R}^d$, $\lambda_0 > 0$, $\theta_0 = +\infty$
$x^1 = x^0 - \lambda_0\nabla f(x^0)$
for $k = 1, 2, \ldots$ **do**
  $\lambda_k = \min\left\{\sqrt{1 + \theta_{k-1}}\lambda_{k-1}, \frac{\|x^k - x^{k-1}\|}{2\|\nabla f(x^k) - \nabla f(x^{k-1})\|}\right\}$
  $x^{k+1} = x^k - \lambda_k\nabla f(x^k)$
  $\theta_k = \frac{\lambda_k}{\lambda_{k-1}}$
end for

Figure 4: Adaptive Gradient Descent Algorithm [1]

REFERENCES

[1] Y. Malitsky and K. Mishchenko. Adaptive gradient descent without descent. In International Conference on Machine Learning, 2020.

[2] H. Attouch, X. Goudou, and P. Redont. The heavy ball with friction method, i. the continuous dynamical system: Global exploration of the local minima of a real-valued function by asymptotic analysis of a dissipative dynamical system. *Communications in Contemporary Mathematics*, 02(01):1–34, 2000.

[3] H. Attouch and J. Fadili. From the Ravine method to the Nesterov method and vice versa: a dynamical system perspective, 2022. arXiv:2201.11643.

[4] Michael Muehlebach and Michael Jordan. A dynamical systems perspective on Nesterov acceleration. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4656–4662. PMLR, 09–15 Jun 2019.

ACKNOWLEDGEMENT

I would like to express my deep thanks to my supervisor: Matthew Tam for constant guidance, detailed explanation, and extreme patience. This invaluable vacation scholar research experience directs my further interest and passion for optimization algorithms and their applications in different dynamical systems.

CONTACT DETAIL

Email yuhao3@student.unimelb.edu.au