Proc. Int. Cong. of Math. – 2018
Rio de Janeiro, Vol. 1 (523–550)

DYNAMICAL, SYMPLECTIC AND STOCHASTIC PERSPECTIVES ON GRADIENT-BASED OPTIMIZATION

Michael I. Jordan

Abstract

Our topic is the relationship between dynamical systems and optimization. This is a venerable, vast area in mathematics, counting among its many historical threads the study of gradient flow and the variational perspective on mechanics. We aim to build some new connections in this general area, studying aspects of gradient-based optimization from a continuous-time, variational point of view. We go beyond classical gradient flow to focus on second-order dynamics, aiming to show the relevance of such dynamics to optimization algorithms that not only converge, but converge quickly.

Although our focus is theoretical, it is important to motivate the work by considering the applied context from which it has emerged. Modern statistical data analysis often involves very large data sets and very large parameter spaces, so that computational efficiency is of paramount importance in practical applications. In such settings, the notion of efficiency is more stringent than that of classical computational complexity theory, where the distinction between polynomial complexity and exponential complexity has been a useful focus. In large-scale data analysis, algorithms need to be not merely polynomial, but linear, or nearly linear, in relevant problem parameters. Optimization theory has provided both practical and theoretical support for this endeavor. It has supplied computationally-efficient algorithms, as well as analysis tools that allow rates of convergence to be determined as explicit functions of problem parameters. The dictum of efficiency has led to a focus on algorithms that are based principally on gradients of objective functions, or on estimates of gradients, given that Hessians incur quadratic or cubic complexity in the dimension of the configuration space (Bottou [2010] and Nesterov [2012a]).

More broadly, the blending of inferential and computational ideas is one of the major intellectual trends of the current century—one currently referred to by terms such as “data science” and “machine learning.” It is a trend that inspires the search for new mathematical concepts that allow computational and inferential desiderata to be studied jointly. For example, one would like to impose runtime budgets on data-analysis algorithms as a function of statistical quantities such as risk, the number of data points and the model complexity, while also taking into account computational resource constraints such as number of processors, the communication bandwidth and the degree

MSC2010: primary 65K10; secondary 90C25, 90C26, 37M15, 37J05, 49M99.

of asynchrony. Fundamental understanding of such tradeoffs seems likely to emerge through the development of lower bounds—by establishing notions of "best" one can strip away inessentials and reveal essential relationships. Here too optimization theory has been important. In a seminal line of research beginning in the 1970's, Nemirovskii, Nesterov and others developed a complexity theory of optimization, establishing lower bounds on rates of convergence and discovering algorithms that achieve those lower bounds (Nemirovskii and Yudin [1983] and Nesterov [1998]). Moreover, the model of complexity was a relative one—an "oracle" is specified and algorithms can only use information that is available to the oracle. For example, it is possible to consider oracles that have access only to function values and gradients. Thus the dictum of practical computational efficiency can be imposed in a natural way in the theory.

Our focus is the class of optimization algorithms known as "accelerated algorithms" Nesterov [1998]. These algorithms often attain the oracle lower-bound rates, although it is something of a mystery why they do so. We will argue that some of mystery is due to the historical focus in optimization on discrete-time algorithms and analyses. In optimization, the distinction between "continuous optimization" and "discrete optimization" refers to the configuration ("spatial") variables. By way of contrast, our discussion will focus on continuous time. In continuous time we can give acceleration a mathematical meaning as a differential concept, as a change of speed along a curve. And we can pose the question of "what is the fastest rate?" as a problem in variational analysis; in essence treating the problem of finding the "optimal way to optimize" for a given oracle itself as a formal problem of optimization. Such a variational perspective also has the advantage of being generative—we can derive algorithms that achieve fast rates rather than requiring an analysis to establish a fast rate for a specific algorithm that is derived in an adhoc manner.

Working in continuous time forces us to face the problem of discretizing a continuous-time dynamical system, so as to derive an algorithm that can be implemented on a digital computer. Interestingly, we will find that symplectic integrators, which are widely used for integrating dynamics obtained from variational or Hamiltonian perspectives, are relevant in the optimization setting. Symplectic integration preserves the continuous symmetries of the underlying dynamical system, and this stabilizes the dynamics, allowing step sizes to be larger. Thus algorithms obtained from symplectic integration can move more quickly through a configuration space; this gives a geometric meaning to "acceleration."

It is also of interest to consider continuous-time stochastic dynamics that are in some sense "accelerated." The simplest form of gradient-based stochastic differential equation is the Langevin diffusion. The particular variant that has been studied in the literature is an *overdamped* diffusion that is an analog of gradient descent. We will see that by considering instead an *underdamped* Langevin diffusion, we will obtain a method that is more akin to accelerated gradient descent, and which in fact provably yields a faster rate than the overdamped diffusion.

The presentation here is based on joint work with co-authors Andre Wibisono, Ashia Wilson, Michael Betancourt, Chi Jin, Praneeth Netrapalli, Rong Ge, Sham Kakade, Niladri Chatterji and Xiang Cheng, as well as other co-authors who will be acknowledged in specific sections.

1 Lagrangian and Hamiltonian Formulations of Accelerated Gradient Descent

Given a continuously differentiable function f on an open Euclidean domain $\mathcal{X}$, and given an initial point $\mathbf{x}_0 \in \mathcal{X}$, gradient descent is defined as the following discrete dynamical system:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta \nabla f(\mathbf{x}_k), \tag{1}$$

where $\eta > 0$ is a *step size* parameter.

When f is convex, it is known that gradient descent converges to the global optimum $\mathbf{x}^\star$, assumed unique for simplicity, at a rate of $O(1/k)$ Nesterov [ibid.]. This means that after k iterations, the function value $f(\mathbf{x}_k)$ is guaranteed to be within a constant times $1/k$ of the optimum value $f^\star = f(\mathbf{x}^\star)$. This is a worst-case rate, meaning that gradient descent converges as least as fast as $O(1/k)$ across the function class of convex functions. The constant hidden in the $O(\cdot)$ notation is an explicit function of a complexity measure such as a Lipschitz constant for the gradient.

In the 1980's, a complexity theory of optimization was developed in which rates such as $O(1/k)$ could be compared to lower bounds for particular problem classes (Nemirovskii and Yudin [1983]). For example, an oracle model appropriate for gradient-based optimization might consider all algorithms that have access to sequences of gradients of a function, and whose iterates must lie in the linear span of the current gradient and all previous gradients. This model encompasses, for example, gradient descent, but other algorithms are allowed as well. Nemirovskii and Yudin [ibid.] were able to prove, by construction of a worst-case function, that no algorithm in this class can converge at a rate faster than $O(1/k^2)$. This lower bound is better than gradient descent, and it holds open the promise that some gradient-based algorithm can beat gradient descent across the family of convex functions. That promise was realized by Nesterov [1983], who presented the following algorithm, known as *accelerated gradient descent*:

$$\begin{aligned} \mathbf{y}_{k+1} &= \mathbf{x}_k - \eta \nabla f(\mathbf{x}_k) \\ \mathbf{x}_{k+1} &= (1+\lambda_k)\mathbf{y}_{k+1} - \lambda_k \mathbf{y}_k, \end{aligned} \tag{2}$$

and proved that the algorithm converges at rate $O(1/k^2)$ for convex functions f. Here λ_k is an explicit function of the other problem parameters. We see that the acceleration involves two successive gradients, and the resulting dynamics are richer than those of gradient descent. In particular, accelerated gradient descent is not a descent algorithm—the function values can oscillate.

Nesterov's basic algorithm can be presented in other ways; in particular, we will also use a three-sequence version:

$$\begin{aligned}\mathbf{x}_k &= y_k + \lambda_k \mathbf{v}_k \\ \mathbf{y}_{k+1} &= \mathbf{x}_k - \eta \nabla f(\mathbf{x}_k) \\ \mathbf{v}_{k+1} &= \mathbf{y}_{k+1} - y_k.\end{aligned} \tag{3}$$

We note in passing that in both this version and the two-sequence version, the parameter λ_k is time-varying for some problem formulations and constant for others.

After Nesterov's seminal paper in 1983, the subsequent three decades have seen the development of a variety of accelerated algorithms in a wide variety of other problem settings. These include mirror descent, composite objective functions, non-Euclidean geometries, stochastic variants and higher-order gradient descent. Rates of convergence have been obtained for these algorithms, and these rates often achieve oracle lower bounds. Overall, acceleration has been one of the most productive ideas in modern optimization theory. See Nesterov [1998] for a basic introduction, and Bubeck, Y. T. Lee, and M. Singh [2015] and Allen-Zhu and Orecchia [2014] for examples of recent progress.

And yet the basic acceleration phenomenon has remained somewhat of a mystery. Its derivation and its analysis are often obtained only after lengthy algebraic manipulations, with clever but somewhat opaque upper bounds needed at critical junctures.

In Wibisono, Wilson, and Jordan [2016], we argue that this mystery is due in part to the discrete-time formalism that is generally used to derive and study gradient-based optimization algorithms. Indeed, the notion of "acceleration" seems ill-defined in a discrete-time framework; what does it mean to move more quickly along a sequence of discrete points? Such a notion seems to require an embedding in an underlying flow of time, such that acceleration can be viewed as a diffeomorphism. Moreover, if accelerated optimization algorithms are in some sense optimal, there must be something special about the curve that they follow in the configuration space, not merely the speed at which they move. Such a separate characterization of curve and speed also seem to require continuous time.

Wibisono, Wilson, and Jordan [ibid.] address these issues via a variational framework that aims to capture the phenomenon of acceleration in some generality. We review this framework in the remainder of this section, discussing the Lagrangian formulation that captures acceleration in continuous time, showing how this formulation gives rise to a family of differential equations whose convergence rates are the continuous-time counterpart of the discrete-time oracle rates. We highlight the problem of the numerical integration of these differential equations, setting up the symplectic integration approach that we discuss in Section 2.

We consider the general non-Euclidean setting in which the space $\mathcal{X}$ is endowed with a distance-generating function $h : \mathcal{X} \to \mathbb{R}$ that is convex and essentially smooth (i.e., h is continuously differentiable in $\mathcal{X}$, and $\|\nabla h(\mathbf{x})\|_* \to \infty$ as $\|\mathbf{x}\| \to \infty$). The function h can be used to define a measure of distance in $\mathcal{X}$ via its *Bregman divergence*:

$$D_h(\mathbf{y}, \mathbf{x}) = h(\mathbf{y}) - h(\mathbf{x}) - \langle \nabla h(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle. \tag{4}$$

The Euclidean setting is obtained when $h(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|^2$.

We use the Bregman divergence to construct a *Bregman kinetic energy* for a dynamical system. We do this by taking the Bregman divergence between a point $\mathbf{x}$ and its translation in the direction of the velocity $\mathbf{v}$ by a time-varying magnitude, $e^{-\alpha_t}$:

$$K(\mathbf{x}, \mathbf{v}, t) := D_h(\mathbf{x} + e^{-\alpha_t}\mathbf{v}, \mathbf{x}). \tag{5}$$

Using the definition in Eq. (4), we see that this kinetic energy can be interpreted as comparison between the amount that h changes under a finite translation, $h(\mathbf{x}+e^{-\alpha_t}\mathbf{v}) - h(\mathbf{x})$, versus an infinitesimal translation, $e^{-\alpha_t}\langle \nabla h(\mathbf{x}), \mathbf{v} \rangle$.

We now define a time-dependent potential energy, $U(\mathbf{x})$:

$$U(\mathbf{x}, t) := e^{\beta_t} f(\mathbf{x}), \tag{6}$$

and we subtract the potential energy from the kinetic energy to obtain the *Bregman Lagrangian*:

$$\begin{aligned} \mathcal{L}(\mathbf{x}, \mathbf{v}, t) &=:= e^{\alpha_t + \gamma_t}\left(K(\mathbf{x}, \mathbf{v}, t) - U(\mathbf{x}, t)\right) \\ &= e^{\alpha_t + \gamma_t}\left(D_h(\mathbf{x} + e^{-\alpha_t}\mathbf{v}, \mathbf{x}) - e^{\beta_t} f(\mathbf{x})\right). \end{aligned} \tag{7}$$

In this equation, the time-dependent factors α_t, β_t and γ_t are algorithmic degrees of freedom that allow the Bregman–Lagrangian framework to encompass a range of different algorithms.

Although α_t, β_t and γ_t can be set independently in principle, we define a set of *ideal scaling conditions* that reduce these three degrees of freedom to a single functional degree of freedom:

$$\begin{aligned} \dot{\beta}_t &\leq e^{\alpha_t} \\ \dot{\gamma}_t &= e^{\alpha_t}. \end{aligned} \tag{8}$$

Wibisono, Wilson, and Jordan [ibid.] show that these conditions are needed to obtain differential equations whose rates of convergence are the optimal rates; see Theorem 1 below.

Given the Bregman Lagrangian, we use standard calculus of variations to obtain a differential equation whose solution is the path that optimizes the time-integrated Bregman Lagrangian. In particular, we form the Euler–Lagrange equations:

$$\frac{d}{dt}\left\{\frac{\partial \mathcal{L}}{\partial \mathbf{v}}(\mathbf{x}_t, \dot{\mathbf{x}}_t, t)\right\} - \frac{\partial \mathcal{L}}{\partial \mathbf{x}}(\mathbf{x}_t, \dot{\mathbf{x}}_t, t) = 0, \tag{9}$$

a computation which is easily done based on Eq. (7). Using the ideal scaling conditions in Eq. (8), the result simplifies to the following master differential equation:

$$\ddot{x}_t + (e^{\alpha_t} - \dot{\alpha}_t)\dot{x}_t + e^{2\alpha_t + \beta_t}\left[\nabla^2 h(x_t + e^{-\alpha_t}\dot{x}_t)\right]^{-1}\nabla f(x_t) = 0. \tag{10}$$

We see that the equation is second order and non-homogeneous. Moreover, the gradient $\nabla f(x_t)$ appears as a force, modified by geometric terms associated with the Bregman distance-generating function h. As we will discuss below, this equation is a general form of Nesterov acceleration in continuous time.

It is straightforward to obtain a convergence rate for the master differential equation. We define the following Lyapunov function:

$$\mathcal{E}_t = D_h(\mathbf{x}^\star, \mathbf{x}_t + e^{-\alpha_t}\dot{\mathbf{x}}_t) + e^{\beta_t}(f(\mathbf{x}_t) - f^\star). \tag{11}$$

Taking a first derivative with respect to time, and asking that this derivative be less than or equal to zero, we immediately obtain a convergence rate, as documented in the following theorem, whose proof can be found in Wibisono, Wilson, and Jordan [2016].

Theorem 1. *If the ideal scaling in Eq. (8) holds, then solutions to the Euler–Lagrange equation Eq. (10) satisfy*

$$f(\mathbf{x}_t) - f^\star \leq O(e^{-\beta_t}).$$

For further explorations of Lyapunov-based analysis of accelerated gradient methods, see Wilson, Recht, and Jordan [2016].

Wibisono, Wilson, and Jordan [2016] studied a subfamily of Bregman Lagrangians with the following choice of parameters, indexed by a parameter $p > 0$:

$$\begin{aligned} \alpha_t &= \log p - \log t \\ \beta_t &= p \log t + \log C \\ \gamma_t &= p \log t, \end{aligned} \tag{12}$$

where $C > 0$ is a constant. This choice of parameters satisfies the ideal scaling condition in Eq. (8). The Euler–Lagrange equation, Eq. (10), reduces in this case to:

$$\ddot{x}_t + \frac{p+1}{t}\dot{x}_t + Cp^2t^{p-2}\left[\nabla^2 h\left(x_t + \frac{t}{p}\dot{x}_t\right)\right]^{-1}\nabla f(x_t) = 0, \tag{13}$$

and, by Theorem 1, it has an $O(1/t^p)$ rate of convergence.

The case $p = 2$ of the Eq. (13) is the continuous-time limit of Nesterov's accelerated mirror descent (Krichene, Bayen, and Bartlett [2015]), the case $p = 3$ is the continuous-time limit of Nesterov's accelerated cubic-regularized Newton's method (Nesterov and Polyak [2006]). In the Euclidean case, when the Hessian $\nabla^2 h$ is the identity matrix, we recover the following differential equation:

$$\ddot{x}_t + \frac{3}{t}\dot{x}_t + \nabla f(x_t) = 0, \tag{14}$$

which was first derived by Su, Boyd, and Candes [2016] as the continuous-time limit of the basic Nesterov accelerated gradient descent algorithm in Eq. (2).

The Bregman Lagrangian has several mathematical properties that give significant insight into aspects of the acceleration phenomenon. For example, the Bregman Lagrangian is closed under time dilation. This means that if we take an Euler–Lagrange curve of a Bregman Lagrangian and reparameterize time so that we travel the curve at a different speed, then the resulting curve is also the Euler–Lagrange curve of another Bregman Lagrangian, with appropriately modified parameters. Thus, the entire family of accelerated methods correspond to a single curve in spacetime and can be obtained by speeding up (or slowing down) any single curve. As suggested earlier, the Bregman Lagrangian framework permits us to separate out the consideration of the optimal curve from optimal speed of movement along that curve.

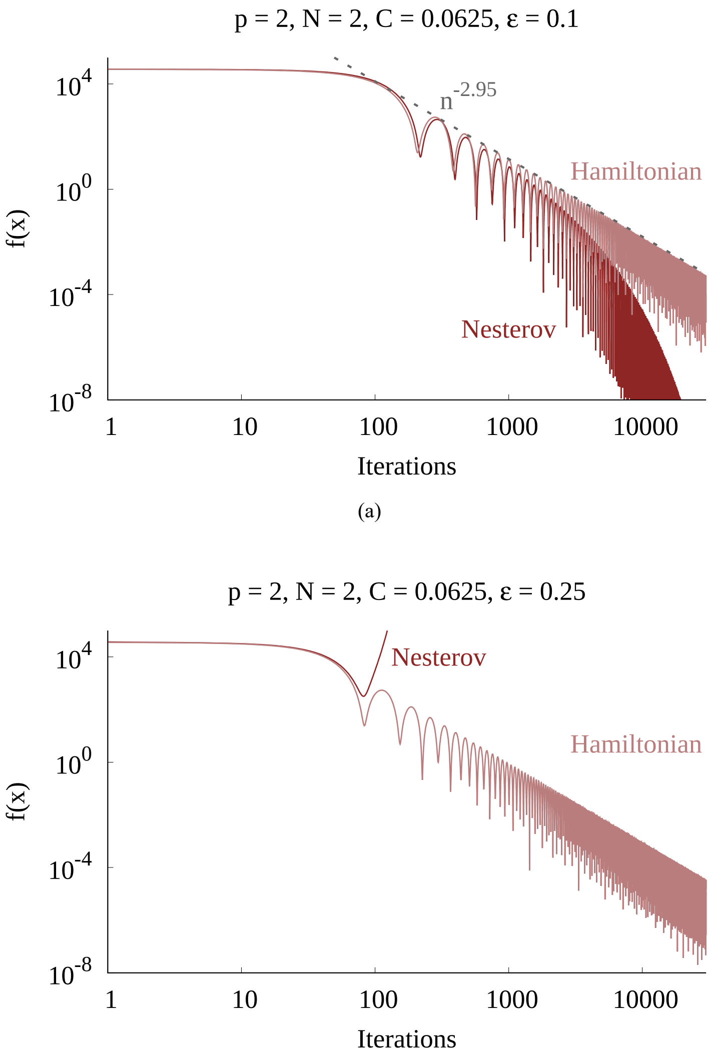
Finally, we turn to a core problem—how to discretize the master differential equation so that it can be solved numerically on a digital computer. Wibisono, Wilson, and Jordan [2016] showed that naïve discretizations can fail to yield stable discrete-time dynamical systems, or fail to preserve the fast oracle rates of the underlying continuous system. Motivated by the three-sequence form of Nesterov acceleration (see Eq. (3)), they derived the following algorithm, in the case of the logarithmic parameterization shown in Eq. (12):

$$\begin{aligned} \mathbf{x}_{k+1} &= \frac{p}{k+p}\mathbf{z}_k + \frac{k}{k+p}\mathbf{y}_k \\ \mathbf{y}_k &= \operatorname*{argmin}_{\mathbf{y}\in\mathcal{X}} \left[f_{p-1}(\mathbf{y};\mathbf{x}_k) + \frac{N}{\epsilon^p\, p}\|\mathbf{y}-\mathbf{x}_k\|^p \right] \\ \mathbf{z}_k &= \operatorname*{argmin}_{\mathbf{z}\in\mathcal{X}} \left[C\, p\, k^{(p-1)} \langle \nabla f(\mathbf{y}_k), \mathbf{z}\rangle + \frac{1}{\epsilon^p} D_h(\mathbf{z},\mathbf{z}_{k-1}) \right]. \end{aligned} \tag{15}$$

Here $f_{p-1}(\mathbf{y};\mathbf{x}_k)$ is the order-p Taylor expansion of the objective function around $\mathbf{x}_k$ and N and C are scaling coefficients and ϵ is a step size. Although Wibisono, Wilson, and Jordan [ibid.] were able to prove that this discretization is stable and achieves the oracle rate of $O(1/k^p)$, the discretization is heuristic and does not flow natural from the dynamical-systems framework. In the next section, we revisit the discretization issue from the point of view of symplectic integration.

2 A Symplectic Perspective on Acceleration

Symplectic integration is a general for the discretization of differential equations that preserves various of the continuous symmetries of the dynamical system (Hairer, Lubich, and Wanner [2006]). In the case of differential equations obtained from mechanics, these symmetries include physically-meaningful first integrals such as energy and momentum. Symplectic integrators exactly conserve these quantities even if the dynamical flow is only approximated. In addition to the appeal of this result from the point of view of physical conservation laws, the preservation of continuous symmetries means that symplectic integrators tend to be more stable than other integration schemes, such that it is possible to use larger step sizes in the discrete-time system. It is this latter fact that suggests a role for symplectic integrators in the integration of the differential equations associated with accelerated optimization methods. This idea has been pursued in recent work by Betancourt, Jordan, and Wilson [2018], whose results we review in this section.



(a)

(b)

Figure 1: (a) When appropriately tuned, both the leapfrog integrator and the three-sequence Nesterov algorithm simulate the same latent Bregman dynamics and hence achieve similar convergence rates, here approximately $\mathcal{O}(k^{-2.95})$. (b) Given a larger step size, the symplectic integrator remains stable and thus converges more quickly, whereas the three-sequence Nesterov algorithm becomes unstable.

While symplectic integrators can be obtained from a Lagrangian framework, they are most naturally obtained from a Hamiltonian framework. We thus begin by transforming the Bregman Lagrangian into a Bregman Hamiltonian. This is readily done via a Legendre transform, as detailed in Wibisono, Wilson, and Jordan [2016] and Betancourt, Jordan, and Wilson [2018]. The resulting Hamiltonian is as follows:

$$H(\mathbf{x}, \mathbf{r}, t) = e^{\alpha(t)+\gamma(t)} \left(D_{h^*}(e^{-\gamma(t)}\mathbf{r} + \frac{\partial h}{\partial \mathbf{x}}(\mathbf{r}), \frac{\partial h}{\partial \mathbf{x}}(\mathbf{x})) + e^{\beta(t)} f(\mathbf{x}) \right), \tag{16}$$

where

$$D_{h^*}(\mathbf{r}, \mathbf{s}) = h^*(\mathbf{r}) - h^*(\mathbf{s}) - \frac{\partial h^*}{\partial \mathbf{r}}(\mathbf{s}) \cdot (\mathbf{r} - \mathbf{s}),$$

and where h^* is the Fenchel conjugate:

$$h^*(r) \underset{\mathbf{v} \in T\mathcal{X}}{=} \sup\left(\langle \mathbf{r}, \mathbf{v} \rangle - h(v)\right).$$

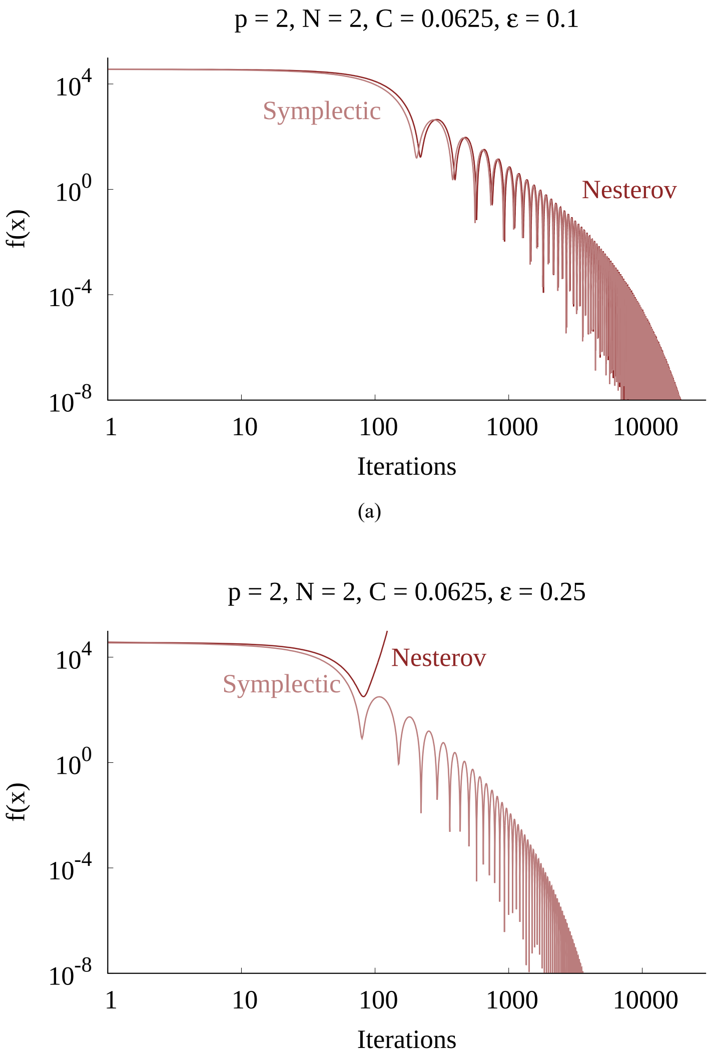
Given the Bregman Hamiltonian in Eq. (16), Betancourt, Jordan, and Wilson [ibid.] follow a standard sequence of steps to obtain a symplectic integrator. First, the Bregman Hamiltonian is time-varying, and it is thus lifted into a time-invariant Hamiltonian on an augmented configuration space that includes time as an explicit variable and includes a conjugate energy variable in the phase space. Second, the Hamiltonian is split into a set of component Hamiltonians, each of which can be solved analytically (or nearly so via simple numerical methods). Third, the component dynamics are composed symmetrically to form the full dynamics. In particular, Betancourt, Jordan, and Wilson [ibid.] illustrate how to form a symmetric leapfrog integrator (a particular kind of symplectic integrator) for the Bregman Hamiltonian. They prove that the error between this integrator and the true dynamics is of order $O(\epsilon^2)$, where ϵ is the step size in the discretization.

Betancourt, Jordan, and Wilson [ibid.] also present empirical results for a quadratic objective function, $f(x) = \langle \Sigma^{-1} x, x \rangle$, on a 50-dimensional Euclidean space, where

$$\Sigma_{ij} = \rho^{|i-j|},$$

and $\rho = 0.9$. This experiment was carried out in the setting of Eq. (12), for various choices of p, C and N. Representative results are shown in Figure 1(a), which compare the leapfrog integrator with the three-sequence version of Nesterov acceleration from Eq. (15). Here we see that both approaches yield stable, oscillatory dynamics whose asymptotic convergence rate is approximately $\mathcal{O}(k^{-2.95})$. Moreover, as shown in Figure 1(b), the symplectic integrator remains stable when a larger step size is chosen, whereas the three-sequence Nesterov algorithm becomes unstable.

Interestingly, however, while the initial rate is the same, the three-sequence algorithm exhibits exponential convergence near the optimum. This behavior does not hold in general (e.g., Betancourt, Jordan, and Wilson [ibid.] show that it does not hold for quartic functions), but it is nonetheless an interesting feature of the three-sequence method in the case of quadratic objectives. Betancourt, Jordan, and Wilson [ibid.] show



(a)

(b)

Figure 2: (a) By incorporating gradient flow into the leapfrog integration of the Bregman Hamiltonian dynamics we recover the same asymptotic exponential convergence near the minimum of the objective exhibited by the dynamical Nesterov algorithm. (b) These modified Hamiltonian dynamics remain stable even as we increase the step size, allowing for more efficient computation and the advantageous asymptotic behavior.

that it arises from an implicit gradient flow that is a side effect of the three-sequence discretization. Moreover, they note that it is possible to mimic this flow within the symplectic integrator. When this is done the results for the quadratic objective are as shown in Figure 2(a). We see that a symplectic integrator that incorporates gradient flow and the three-sequence integrator yield convergence profiles that are essentially equivalent in this case.

But it is also true that the symplectic approach is topologically more stable than the three-sequence method, a fact which is revealed if one chooses a more aggressive step size. This is exhibited in Figure 2(b), where the symplectic integrator converges while the three-sequence Nesterov method diverges when the step size is increased.

In summary, this section has exhibited a connection between symplectic integration and the acceleration phenomenon in optimization. When the latter is construed as a continuous-time phenomenon, symplectic integration appears to provide an effective and flexible way to obtain discrete-time approximations. Much remains to be done, however, to tighten this link. In particular, we would like to obtain necessary and sufficient conditions for stable integrators that achieve oracle rates, and it is not yet clear what role symplectic geometry will play in uncovering those conditions.

3 Acceleration and the Escape from Saddle Points in Nonconvex Optimization

In this section we turn to *nonconvex* optimization. Although the general nonconvex setting harbors many intractable problems about which little can be said regarding computational or statistical efficiency, it turns out that for a wide range of problems in statistical learning, there is sufficient mathematical structure present in the nonconvex setting that useful mathematical results can be obtained. Indeed, in many cases the ideas and algorithms from convex optimization—suitably modified—can be carried over to the nonconvex setting. In particular, for gradient-based optimization, the same algorithms that perform well in the convex setting also tend to yield favorable performance in the nonconvex setting. In this sense, convex optimization has served as a laboratory for nonconvex optimization, in addition to having many natural applications of its own.

A useful first foothold on nonconvex optimization is obtained by considering the criterion of *first-order stationarity*. Given a differentiable function $f : \mathcal{X} \to \mathbb{R}$, on some well-behaved Euclidean domain $\mathcal{X}$ of dimension d, we define first-order stationary points to be those points $\mathbf{x} \in \mathcal{X}$ where the gradient vanishes: $\|\nabla f(\mathbf{x})\| = 0$. Although first-order stationary points can in general be associated with many kinds of topological singularity, for many statistical learning problems it suffices to consider the categorization into points that are global minima, local minima, local maxima and saddle points. Of these, local maxima are rarely viewed as problematic—simple modifications of gradient descent, such as stochastic perturbation, can suffice to ensure that algorithms do not get stuck at local maxima. Local minima have long been viewed as the core concern in nonconvex optimization for statistical learning problems. Recent work has shown, however, that in a wide range of nonconvex statistical learning problems, local minima are provably absent, or, in empirical studies, even when local minima are present they do not appear to be discovered by gradient-based algorithms. Such results have been obtained for smooth semidefinite programs (Boumal, Voroninski, and Bandeira [2016]), matrix completion (Ge, J. D. Lee, and Ma [2016]), synchronization and Max-Cut (Bandeira, Boumal, and Voroninski [2016] and Mei, Misiakiewicz, Montanari, and Oliveira [2017]), multi-layer neural networks (Choromanska, Henaff, Mathieu, Arous, and LeCun [2014] and Kawaguchi [2016]) matrix sensing Bhojanapalli, Neyshabur, and Srebro [2016] and robust principal components analysis (Ge, Jin, and Y. Zheng [2017]).

As for global minima, while they are unambiguously the desirable end states for optimization algorithms, when there are multiple global minima it will generally be necessary to impose additional criteria (e.g., statistical) to single out preferable global minima, and to ask that an optimization algorithm respect this preference. We will not discuss these additional criteria here.

It remains to consider saddle points. Naively one might view these as akin to local maxima, in the sense that it is plausible that a simple perturbation could suffice for a gradient-based algorithm to roll down a direction of negative curvature. Such an argument has support from a recent theoretical result: J. D. Lee, Simchowitz, Jordan, and Recht [2016] have shown that under regularity conditions gradient descent will converge asymptotically and almost surely to a (local) minimum and thus avoid saddle points. In particular, a gradient-based algorithm that is initialized at a random point in $\mathcal{X}$ will avoid any and all saddle points in the asymptotic limit. While this result helps to emphasize the strength of gradient descent, it is of limited practical in that it is asymptotic (providing no rate of convergence); moreover, critically, it does not provide any insight into the rate of escape of saddle points as a function of dimension. While under suitable regularity all directions are escape directions for local maxima, it could be that only one direction is an escape direction for a saddle point. The computational burden of finding that direction could be significant; perhaps exponential in dimension. Given that modern statistical learning problems can involve many hundreds of thousands or millions of dimensions, such a burden would be fatal.

We thus focus our discussion on saddle points. To tie the discussion here to the discussion of the previous section, we take a dynamical systems perspective and study the extent to which acceleration (second-order dynamics) is able to improve the rate of escape of saddle points in gradient-based optimization. Intuitively, there is a narrow region around a saddle point in which the flow is principally directed towards the saddle point, and it seems plausible that an accelerated algorithm is able to bypass such a region more effectively than a non-accelerated algorithm. Whether this is actually true has been an open question in the literature.

To understand how dynamics and geometry interact in the neighborhood of saddle points, it is necessary to go beyond first-order stationarity, which lumps saddle points together with local minima, and to impose a condition that excludes saddle points. We define a *second-order stationary point* to be a point $\mathbf{x} \in \mathcal{X}$ such that $\|\nabla f(\mathbf{x})\| = 0$

and $\lambda_{\min}(\nabla^2 f(\mathbf{x})) \geq 0$. This definition includes local minima, but it also allows degenerate saddle points, in which the smallest eigenvalue of the Hessian is zero, and so we also define a *strict saddle point* to be a point $\mathbf{x} \in \mathcal{X}$ for which $\lambda_{\min}(\nabla^2 f(\mathbf{x})) < 0$. These two definitions jointly allow us to separate local minima from most saddle points. In particular, if all saddle points are strict, then an algorithm that converges to a second-order stationary point necessarily converges to a local minimum. (See Ge, Huang, Jin, and Yuan [2015] for further discussion.)

It turns out that these requirements are reasonable in practical applications. Indeed, it it has been shown theoretically that all saddle points are strict in many of the nonconvex problems mentioned earlier, including tensor decomposition, phase retrieval, dictionary learning and matrix completion (Ge, Huang, Jin, and Yuan [2015], J. Sun, Qu, and J. Wright [2016b,a], Bhojanapalli, Neyshabur, and Srebro [2016], and Ge, J. D. Lee, and Ma [2016]). Coupled with the fact (mentioned above) that there is a single global minimum in such problems, we see that an algorithm that converges to a second-order stationary point will actually converge to a global minimum.

To obtain rates of convergence, we need to weaken the definitions of stationarity to allow an algorithm to arrive in a ball of size $\epsilon > 0$ around a stationary point, for varying ϵ. We define an *ϵ-first-order stationary point* as a point $\mathbf{x} \in \mathcal{X}$ such that $\|\nabla f(\mathbf{x})\| \leq \epsilon$. Similarly we define an *ϵ-second-order stationary point* as a point $\mathbf{x} \in \mathcal{X}$ for which $\lambda_{\min}(\nabla^2 f(\mathbf{x})) \geq -\sqrt{\rho\epsilon}$, where ρ is the Hessian Lipschitz constant. (We have followed Nesterov and Polyak [2006] in using a parameterization for the Hessian that is relative to size of the gradient.)

Finally, we need to impose smoothness conditions on f that are commensurate with the goal of finding second-order stationary points. In particular, we require both the gradient and the Hessian to be Lipschitz:

$$\|\nabla f(\mathbf{x}_1) - \nabla f(\mathbf{x}_2)\| \leq \ell\|\mathbf{x}_1 - \mathbf{x}_2\| \tag{17}$$

$$\|\nabla^2 f(\mathbf{x}_1) - \nabla^2 f(\mathbf{x}_2)\| \leq \rho\|\mathbf{x}_1 - \mathbf{x}_2\|, \tag{18}$$

for constants $0 < \ell, \rho < \infty$, and for all $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$.

Before turning to algorithmic issues, let us calibrate our expectations regarding achievable rates of convergence by considering the simpler problem of finding an ϵ-first-order stationary point, under a Lipschitz condition solely on the gradient. Nesterov [1998] has shown that if gradient descent is run with fixed learning rate $\eta = \frac{1}{\ell}$, and the termination condition is $\|\nabla f(\mathbf{x})\| \leq \epsilon$, then the output will be an ϵ-first-order stationary point, and the algorithm will terminate within the following number of iterations:

$$O\left(\frac{\ell(f(\mathbf{x}_0) - f^\star)}{\epsilon^2}\right),$$

where $\mathbf{x}_0$ is the point at which the algorithm is initialized. While the rate here is less favorable than in the case of smooth convex functions—where it is $O(\epsilon)$[1]—the rate

[1] In Section 1 we expressed rates in terms of the achieved ϵ after a given number of iterations; here we use the inverse function, expressing the number of iterations in terms of the accuracy. Expressed in the former way, the rate here is $O(1/\sqrt{k})$.

Algorithm 1 Perturbed Gradient Descent (PGD)

```
for $k = 0, 1, \ldots,$ do
    if $\|\nabla f(\mathbf{x}_k)\| < g$ and no perturbation in last $\mathcal{T}$ steps then
        $\mathbf{x}_k \leftarrow \mathbf{x}_k + \xi_k \qquad \xi_k \sim \text{Unif}(\mathbb{B}_0(r))$
    $\mathbf{x}_{k+1} = \mathbf{x}_k - \eta \nabla f(\mathbf{x}_k)$
```

retains the essential feature from the convex setting that it is independent of dimension. Recall, however, that saddle points are ϵ-first-order stationary points, and thus this result describes (inter alia) the rate of *approach* to a saddle point. The question that we now turn to is the characterization of the rate of *escape* from a saddle point, where we expect that dimensionality will rear its head.

Turning to algorithmic considerations, we first note that—in contradistinction to the convex case—pure gradient descent will *not* suffice for convergence to a local minimum. Indeed, in the presence of saddle points, the rate of convergence of gradient descent can depend exponentially on dimension (Du, Jin, J. D. Lee, Jordan, Poczos, and A. Singh [2018]). Thus we need to move beyond gradient descent to have a hope of *efficient* escape from saddle points. We could avail ourselves of Hessians, in which case it would be relatively easy to identify directions of escape (as eigenvectors of the Hessian), but as discussed earlier we wish to avoid the use of Hessians on computational grounds. Instead, we focus on gradient descent that is augmented with a stochastic perturbation. Ge, Huang, Jin, and Yuan [2015] and Jin, Ge, Netrapalli, Kakade, and Jordan [2017] studied such an augmentation in which a homogeneous stochastic perturbation (uniform noise in a ball) is added sporadically to the current iterate. Specifically, noise is added when: (1) the norm of the gradient at the current iterate is small, and (2) the most recent such perturbation is at least $\mathcal{T}$ steps in the past, where $\mathcal{T}$ is an algorithmic hyperparameter. We refer to this algorithm as "perturbed gradient descent" (PGD); see Algorithm 1.

We can now state a theorem, proved in Jin, Ge, Netrapalli, Kakade, and Jordan [ibid.], that provides a convergence rate for PGD. Note that PGD has various algorithm hyperparameters, including r (the size of the ball from which the perturbation is drawn), $\mathcal{T}$ (the minimum number of time steps between perturbations), η (the step size), and g (the bound on the norm of the gradient that triggers a perturbation). As shown in Jin, Ge, Netrapalli, Kakade, and Jordan [ibid.], all of these hyperparameters can be specified as explicit functions of the Lipschitz constants ℓ and ρ. The only remaining hyperparameters are a constant quantifying the probability statement in the theorem, and a universal scaling constant.

Theorem 2 (Theorem)**.** *Assume that the function f is ℓ-smooth and ρ-Hessian Lipschitz. Then, with high probability, an iterate $\mathbf{x}_k$ of PGD (Algorithm 1) will be an ϵ-second order stationary point after the following number of iterations k:*

$$O\left(\frac{\ell(f(\mathbf{x}_0) - f^\star)}{\epsilon^2} \log^4\left(\frac{d}{\epsilon^2}\right)\right).$$

We see that the first factor is exactly the same as for ϵ-first-order stationary points with pure gradient descent. The penalty incurred by incorporating the perturbation—and thereby avoiding saddle points—is quite modest—it is only polylogarithmic in the dimension d.

With this convergence result as background, we turn to the main question of this section: Does acceleration aid in the escape from saddle points? In particular, can we improve on the rate in Theorem 2 by incorporating acceleration into the perturbed gradient descent algorithm? We will see that the answer is "yes"; moreover, we will see that a continuous-time dynamical-systems perspective will play a key role in establishing the result.

A major challenge in analyzing accelerated algorithms is that the objective function does not decrease monotonically as is the case for gradient descent. In the convex case, we met this challenge by exploiting the Lagrangian/Hamiltonian formulation to design a Lyapunov function; see Eq. (11). These Lyapunov functions, however, involve the global minimum $\mathbf{x}^{\star}$, which is unknown to the algorithm. This is not problematic in the convex setting, as terms involving $\mathbf{x}^{\star}$ can be bounded using convexity; in the nonconvex setting, however, it is fatal.

To overcome this problem in the nonconvex setting, Jin, Netrapalli, and Jordan [2017] developed a Hamiltonian that is appropriate for the analysis of nonconvex problems. Specializing to Euclidean geometry, the function takes the following form:

$$E_t := \frac{1}{2\eta}\|\mathbf{v}_t\|^2 + f(\mathbf{x}_t); \tag{19}$$

a sum of kinetic energy and potential energy terms. Let us consider using this Hamiltonian to analyze the following second-order differential equation:

$$\ddot{\mathbf{x}} + \tilde{\theta}\dot{\mathbf{x}} + \nabla f(\mathbf{x}) = 0. \tag{20}$$

Integrating both sides, we obtain:

$$f(\mathbf{x}(t_2)) + \frac{1}{2}\dot{\mathbf{x}}(t_2)^2 = f(\mathbf{x}(t_1)) + \frac{1}{2}\dot{\mathbf{x}}(t_1)^2 - \tilde{\theta}\int_{t_1}^{t_2} \dot{\mathbf{x}}(t)^2 \mathrm{d}t. \tag{21}$$

The integral shows that the Hamiltonian decreases monotonically with time t, and the decrease is given by the *dissipation* term $\tilde{\theta}\int_{t_1}^{t_2} \dot{\mathbf{x}}(t)^2 \mathrm{d}t$. Note that Eq. (21) holds regardless of the convexity of $f(\cdot)$.

Although the Hamiltonian decreases monotonically in continuous time, Jin, Netrapalli, and Jordan [ibid.] show that this is *not* the case in discrete time (in the nonconvex setting). Thus, once again, the complexity associated with acceleration manifests itself principally in the transition to discrete time. Jin, Netrapalli, and Jordan [ibid.] were able to resolve this problem by isolating a condition under which the change of the Hamiltonian is indeterminate (it may decrease or increase), and show that under the complement of this condition, the Hamiltonian necessarily decreases. Roughly speaking, the condition arises when the function is "too nonconvex." Moreover, this condition can be assayed algorithmically, and the algorithm can be modified to ensure decrease of the Hamiltonian when the condition arises.

Algorithm 2 Perturbed Accelerated Gradient Descent (PAGD)

```
1: $\mathbf{v}_0 = 0$
2: **for** $k = 0, 1, \ldots,$ **do**
3:     **if** $\|\nabla f(\mathbf{x}_k)\| \leq \epsilon$ and *no perturbation in last* $\mathcal{T}$ *steps* **then**
4:         $\mathbf{x}_k \leftarrow \mathbf{x}_k + \xi_k \qquad \xi_k \sim \text{Unif}(\mathbb{B}_0(r))$
5:     $\mathbf{y}_k = \mathbf{x}_k + \lambda_k \mathbf{v}_k$
6:     $\mathbf{x}_{k+1} = \mathbf{y}_k - \eta \nabla f(\mathbf{y}_k)$
7:     $\mathbf{v}_{k+1} = \mathbf{x}_{k+1} - \mathbf{x}_k$
8:     **if** $f(\mathbf{x}_k) \leq f(\mathbf{y}_k) + \langle \nabla f(\mathbf{y}_k), \mathbf{x}_k - \mathbf{y}_k \rangle - \frac{\gamma}{2}\|\mathbf{x}_k - \mathbf{y}_k\|^2$ **then**
9:         $(\mathbf{x}_{k+1}, \mathbf{v}_{k+1})$ = Negative-Curvature-Exploitation($\mathbf{x}_k, \mathbf{v}_k$)
```

Perturbation	lines 3-4
Standard AGD	lines 5-7
Negative Curvature Exploitation (NCE)	lines 8-9

The overall algorithm, Perturbed Accelerated Gradient Descent (PAGD), is presented in Algorithm 2. Steps 3 and 4 are identical to the PGD algorithm. Steps 5, 6 and 7 replace gradient descent in the latter algorithm with accelerated gradient descent (cf. Eq. (3)). Step 8 measures the "amount of nonconvexity," and if it is too large, makes a call to a function called "Negative-Curvature-Exploitation." This function does one of two things: (1) if the momentum is large, it zeros out the momentum; (2) if the momentum is small, it conducts a local line search along the direction of the momentum. Jin, Netrapalli, and Jordan [2017] prove that this overall algorithm yields monotone decrease in the Hamiltonian. Moreover, they use this result to prove the following theorem regarding the convergence rate of PAGD. (As in the case of PGD, we are not specifying the settings of the various algorithm hyperparameters; we refer to Jin, Netrapalli, and Jordan [ibid.] for these settings. We note, however, that all hyperparameters are functions of the Lipschitz constants ℓ and ρ, with the exception of a constant quantifying the probability statement in the theorem, and a universal scaling constant.)

Theorem 3. *Assume that the function f is ℓ-smooth and ρ-Hessian Lipschitz. Then, with high probability, at least one of iterates, $\mathbf{x}_k$, of PAGD (Algorithm 2) will be an ϵ-second order stationary point after the following number of iterations:*

$$O\left(\frac{\ell^{1/2}\rho^{1/4}(f(\mathbf{x}_0) - f^*)}{\epsilon^{7/4}} \log^6\left(\frac{d}{\epsilon}\right)\right).$$

Comparing this result to Theorem 2, we see that the rate has improved from $1/\epsilon^2$ to $1/\epsilon^{7/4}$. Thus we see that acceleration provably improves upon gradient descent in the nonconvex setting—acceleration hastens the escape from saddle points. We also see that, once again, there is a mild—polylogarithmic—dependence on the dimension d.

The core of the proofs of Theorems 2 and 3 revolves around the study of the local geometry around saddlepoints and its interaction with gradient-descent dynamics. As depicted in Figures 3 and 4, there is a slab-like region in the neighborhood of a saddle

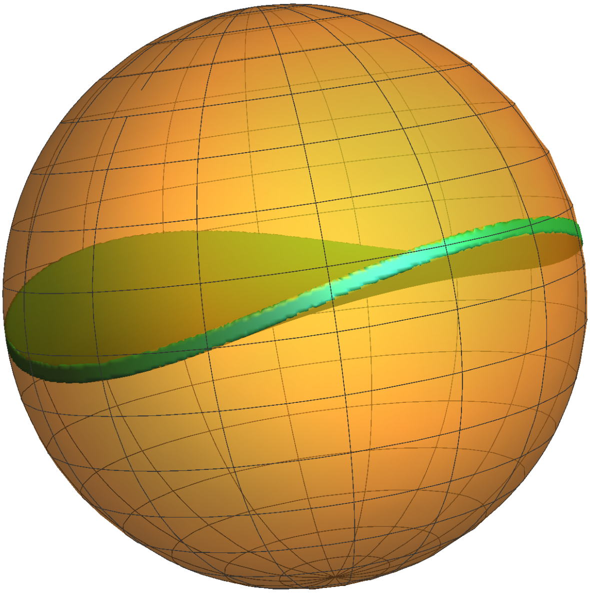


Figure 3: Perturbation ball in three dimensions and "thin pancake" stuck region.

point in which gradient descent will be "stuck"—taking an exponential amount of time to escape. This region is not flat, but instead varies due to the variation of the Hessian in this neighborhood. The Lipschitz assumption gives us control over this variation. To analyze the width of the stuck region, and thus its volume as a fraction of the perturbation ball, Jin, Netrapalli, and Jordan [ibid.] study the rate of escape of a pair of gradient-descent (or accelerated-gradient-descent) sequences that start on the sides of the stuck region. These initial points are a distance r apart along the direction given by the minimum eigenvector of the Hessian, $\nabla^2 f(\mathbf{x})$, at the saddle point. The critical value r for which at least one of the two sequences escapes the stuck region quickly can be computed, and this provides an estimate of the volume of the stuck region. The overall result is that this volume is small compared to that of the perturbation ball, and thus the perturbation is highly likely to cause the optimization algorithm to leave the stuck region (and not return).

4 Underdamped Langevin Diffusion

Our focus thus far has been on dynamical systems which are deterministic, with stochasticity introduced in a limited way—as a perturbation that ensures fast escape from a

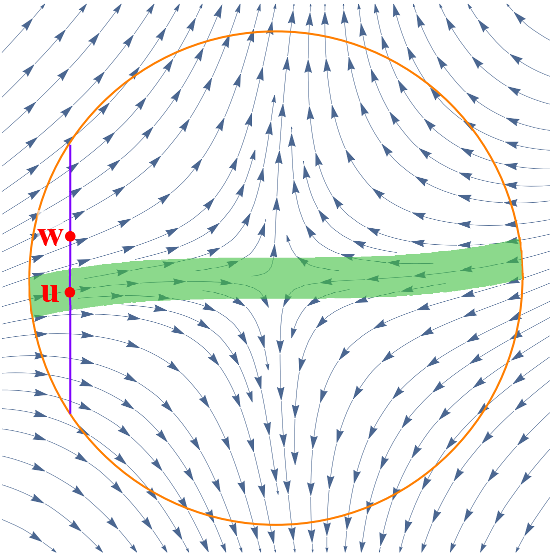


Figure 4: Gradient flow and “narrow band” stuck region in two dimensions.

saddle point. The particular perturbation that we have analyzed—a uniform perturbation in a ball—is sufficient for fast escape, but it is not necessary. Given the success of this simple choice, however, we are motivated to study more thoroughgoing stochastic approaches to our problem. We may wish to investigate, for example, whether a less homogeneous perturbation might suffice. Also, recalling the statistical learning setting that motivates us, even if the optimization problem is formulated as a deterministic one, the underlying data that parameterize this problem are best viewed as random, such that algorithm tractories become stochastic processes, and algorithm outputs become random variables. Thus, taking a stochastic-process point of view opens the door to connecting algorithmic results to inferential results.

We thus turn to a discussion of stochastic dynamics. Given our continuous-time focus, these stochastic dynamics will be expressed as stochastic differential equations. Moreover, we will again be interested in second-order (“momentum”) dynamics, and will investigate the extent to which such dynamics can yield improvements over first-order dynamics.

The classical connection between gradient descent and stochastic differential equations is embodied in the *overdamped Langevin diffusion*:

$$d\mathbf{x}_t = -\nabla f(\mathbf{x}_t)dt + \sqrt{2}dB_t,$$

where $\mathbf{x}_t \in \mathbb{R}^d$ and B_t is d-dimensional standard Brownian motion. Under mild regularity conditions, it is known that the invariant distribution of this continuous-time process is proportional to $p^\star(\mathbf{x}) \propto \exp(-(f(\mathbf{x})))$. Thus samples from $p^\star(\mathbf{x})$ can be obtained by solving the diffusion numerically. Such a numerical solution is generally referred to as “Langevin Markov chain Monte Carlo” or “Langevin MCMC.”

Asymptotic guarantees for overdamped Langevin MCMC were established in Gelfand and Mitter [1991]. The first explicit proof of non-asymptotic convergence of overdamped Langevin MCMC for log-smooth and strongly log-concave distributions was given by Dalalyan [2017], who showed that discrete, overdamped Langevin diffusion achieves ϵ error, in total variation distance, in $O(d/\epsilon^2)$ steps. Following this, Durmus and Moulines [2016] proved that the same algorithm achieves ϵ error, in 2-Wasserstein distance, in $O(d/\epsilon^2)$ steps.[2]

Our second-order perspective motivates us to consider *underdamped Langevin diffusion*:

$$\begin{aligned} d\mathbf{v}_t &= -\gamma \mathbf{v}_t dt - u\nabla f(\mathbf{x}_t)dt + \sqrt{2\gamma u}dB_t \\ d\mathbf{x}_t &= \mathbf{v}_t dt, \end{aligned} \tag{23}$$

where u and γ are parameters. It can be shown that the invariant distribution of this continuous-time process is proportional to $\exp(-(f(\mathbf{x})+\|\mathbf{v}\|_2^2/2u))$. Thus the marginal distribution of $\mathbf{x}$ is proportional to $\exp(-f(\mathbf{x}))$ and samples from $p^\star(\mathbf{x}) \propto \exp(-f(\mathbf{x}))$ can be obtained by solving the underdamped Langevin diffusion numerically and ignoring the momentum component.

Underdamped Langevin diffusion is interesting because it is analogous to accelerated gradient descent; both are second-order dynamical systems. Moreover, it contains a Hamiltonian component, and its discretization can be viewed as a form of Hamiltonian MCMC. Hamiltonian MCMC has been empirically observed to converge faster to the invariant distribution compared to standard Langevin MCMC (Betancourt, Byrne, Livingstone, and Girolami [2017]).

In recent work, Cheng, Chatterji, Bartlett, and Jordan [2017] have analyzed underdamped Langevin diffusion. They have shown that—in the same setting analyzed by Durmus and Moulines [2016] for overdamped Langevin diffusion—that the underdamped algorithm achieves a convergence rate of $O(\sqrt{d}/\epsilon)$ in 2-Wasserstein distance. This is a significant improvement over the $O(d/\epsilon^2)$ rate of overdamped Langevin diffusion, both in terms of the accuracy parameter ϵ and the dimension d.

5 Discussion

The general topic of gradient-based optimization, and its application to large-scale statistical inference problems, is currently very active. Let us highlight one particular set of questions that appear likely to attract ongoing attention in coming years. Note that optimization methods are used classically in the statistical setting to solve point estimation problems, where the core problem is to output a single point in the configuration

[2] Recall that the Wasserstein distance, $W_2(\mu,\eta)$, between probability measures μ and η is defined as follows:

$$W_2(\mu,\eta) := \left(\inf_{\zeta\in\Gamma(\mu,\eta)} \int \|\mathbf{x}-\mathbf{y}\|_2^2 d\zeta(\mathbf{x},\mathbf{y})\right)^{1/2}, \tag{22}$$

where ζ ranges over the set of transference plans $\Gamma(\mu,\eta)$.

space that has desirable statistical properties. But the broader problem is to provide in addition an indication of the uncertainty associated with that output, in the form of some summary of a probability distribution. Optimization ideas can be relevant here as well, by considering a configuration space that is a space of probability distributions. Relatedly, one can ask to converge not to a single point, but to a distribution over points. The Hamiltonian approach naturally yields oscillatory solutions, and, as we have seen, some work is required to obtain algorithms that converge to a point. This suggests that the Hamiltonian approach may in fact be easier to employ in the setting of distributional convergence than in the point estimation setting, and thereby provide an algorithmic bridge between point estimation and broader inference problems. Indeed, in Bayesian inference, Hamiltonian formulations (and symplectic integration, in the form of leapfrog integrators) have been successfully employed in the setting of Markov chain Monte Carlo algorithms, where the momentum component of the Hamiltonian (empirically) provides faster mixing. Deeper connections between acceleration and computationally-efficient inference are clearly worth pursuing.

References

Alekh Agarwal, Sahand Negahban, and Martin J Wainwright (2010). “Fast global convergence rates of gradient methods for high-dimensional statistical recovery”. In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 37–45.

Naman Agarwal, Zeyuan Allen-Zhu, Brian Bullins, Elad Hazan, and Tengyu Ma (2017). “Finding approximate local minima faster than gradient descent”. In: *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, pp. 1195–1199. MR: 3678262.

Zeyuan Allen-Zhu and Elad Hazan (2016). “Variance reduction for faster non-convex optimization”. In: *International Conference on Machine Learning (ICML)*, pp. 699–707.

Zeyuan Allen-Zhu and Lorenzo Orecchia (2014). “Linear coupling: An ultimate unification of gradient and mirror descent”. arXiv: 1407.1537. MR: 3754927 (cit. on p. 526).

Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M. Kakade, and Matus Telgarsky (2014). “Tensor Decompositions for Learning Latent Variable Models”. *Journal of Machine Learning Research* 15, pp. 2773–2832. MR: 3270750.

Sanjeev Arora, Rong Ge, Ankur Moitra, and Sushant Sachdeva (2012). “Provable ICA with unknown Gaussian noise, with implications for Gaussian mixtures and autoencoders”. In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 2375–2383.

Kazuoki Azuma (1967). “Weighted sums of certain dependent random variables”. *Tohoku Mathematical Journal, Second Series* 19.3, pp. 357–367. MR: 0221571.

Pierre Baldi and Kurt Hornik (1989). “Neural networks and principal component analysis: Learning from examples without local minima”. *Neural Networks* 2.1, pp. 53–58. MR: 0936601.

Afonso S Bandeira, Nicolas Boumal, and Vladislav Voroninski (2016). "On the low-rank approach for semidefinite programs arising in synchronization and community detection". In: *Conference on Computational Learning Theory (COLT)*, pp. 361–382 (cit. on p. 534).

Amir Beck and Marc Teboulle (2009). "A fast iterative shrinkage-thresholding algorithm for linear inverse problems". *SIAM Journal on Imaging Sciences* 2.1, pp. 183–202. MR: 2486527.

Yoshua Bengio (2009). "Learning deep architectures for AI". *Foundations and Trends® in Machine Learning* 2.1, pp. 1–127. MR: 3617773.

Michael Betancourt, Simon Byrne, Sam Livingstone, and Mark Girolami (2017). "The geometric foundations of Hamiltonian Monte Carlo". *Bernoulli* 23, pp. 2257–2298. MR: 3648031 (cit. on p. 541).

Michael Betancourt, Michael I Jordan, and Ashia C Wilson (2018). "On symplectic optimization". arXiv: 1802.03653 (cit. on pp. 529, 531).

Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro (2016). "Global optimality of local search for low rank matrix recovery". In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 3873–3881 (cit. on pp. 534, 535).

Léon Bottou (2010). "Large-scale machine learning with stochastic gradient descent". In: *Proceedings of COMPSTAT'2010*. Springer, pp. 177–186. MR: 3362066 (cit. on p. 523).

Nicolas Boumal, Vlad Voroninski, and Afonso S Bandeira (2016). "The non-convex Burer-Monteiro approach works on smooth semidefinite programs". In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 2757–2765 (cit. on p. 534).

Stephen Boyd and Lieven Vandenberghe (2004). *Convex optimization*. Cambridge university press. MR: 2061575.

Alan J Bray and David S Dean (2007). "Statistics of critical points of gaussian fields on large-dimensional spaces". *Physical Review Letters* 98.15, p. 150201. MR: 2346609.

Sébastien Bubeck et al. (2015). "Convex optimization: Algorithms and complexity". *Foundations and Trends® in Machine Learning* 8.3-4, pp. 231–357.

Sébastien Bubeck, Yin Tat Lee, and Mohit Singh (2015). "A geometric alternative to Nesterov's accelerated gradient descent". arXiv: 1506.08187 (cit. on p. 526).

Emmanuel J Candes, Xiaodong Li, and Mahdi Soltanolkotabi (2015). "Phase retrieval via Wirtinger flow: Theory and algorithms". *IEEE Transactions on Information Theory* 61.4, pp. 1985–2007. MR: 3332993.

Emmanuel J Candes and Yaniv Plan (2011). "Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements". *IEEE Transactions on Information Theory* 57.4, pp. 2342–2359. MR: 2809094.

J-F Cardoso (1989). "Source separation using higher order moments". In: *Acoustics, Speech, and Signal Processing*. IEEE, pp. 2109–2112.

Yair Carmon and John C Duchi (2016). "Gradient Descent Efficiently Finds the Cubic-Regularized Non-Convex Newton Step". arXiv: 1612.00547.

Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford (2016). "Accelerated Methods for Non-Convex Optimization". arXiv: 1611.00756. MR: 3814027.

Yair Carmon, Oliver Hinder, John C Duchi, and Aaron Sidford (2017). "Convex until Proven Guilty: Dimension-Free Acceleration of Gradient Descent on Non-Convex Functions". arXiv: 1705.02766.

Coralia Cartis, Nicholas Gould, and Ph L Toint (2010). "On the complexity of steepest descent, Newton's and regularized Newton's methods for nonconvex unconstrained optimization problems". *SIAM Journal on Optimization* 20.6, pp. 2833–2852. MR: 2721157.

Louis Augustin Cauchy (1847). "Méthode générale pour la résolution des systémes d'équations simultanees". *C. R. Acad. Sci. Paris*.

Xiang Cheng, Niladri Chatterji, Peter Bartlett, and Michael I Jordan (2017). "Underdamped Langevin MCMC: A non-asymptotic analysis". arXiv: 1707.03663 (cit. on p. 541).

Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun (2014). "The Loss Surface of Multilayer Networks". *arXiv:1412.0233* (cit. on p. 534).

Pierre Comon (2002). "Tensor decompositions". *Mathematics in Signal Processing V*, pp. 1–24. MR: 1931400.

Pierre Comon, Xavier Luciani, and André LF De Almeida (2009). "Tensor decompositions, alternating least squares and other tales". *Journal of Chemometrics* 23.7-8, pp. 393–405.

Frank E Curtis, Daniel P Robinson, and Mohammadreza Samadi (2014). "A trust region algorithm with a worst-case iteration complexity of $O(\epsilon^{-3/2})$ for nonconvex optimization". *Mathematical Programming*, pp. 1–32. MR: 3612930.

Arnak S Dalalyan (2017). "Theoretical guarantees for approximate sampling from smooth and log-concave densities". *Journal of the Royal Statistical Society: Series B* 79, pp. 651–67. MR: 3641401 (cit. on p. 541).

Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio (2014). "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization". In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 2933–2941.

Simon Du, Chi Jin, Jason D Lee, Michael I Jordan, Barnabas Poczos, and Aarti Singh (2018). "Gradient descent can take exponential time to escape saddle points". In: *Advances in Neural Information Processing Systems (NIPS) 30* (cit. on p. 536).

Alain Durmus and Eric Moulines (2016). "Sampling from strongly log-concave distributions with the Unadjusted Langevin Algorithm". arXiv: 1605.01559 (cit. on p. 541).

Alan Frieze, Mark Jerrum, and Ravi Kannan (1996). "Learning linear transformations". In: *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pp. 359–359. MR: 1450634.

Yan V Fyodorov and Ian Williams (2007). “Replica symmetry breaking condition exposed by random matrix calculation of landscape complexity”. *Journal of Statistical Physics* 129.5-6, pp. 1081–1116. MR: 2363390.

Rong Ge, Furong Huang, Chi Jin, and Yang Yuan (2015). “Escaping From Saddle Points—Online Stochastic Gradient for Tensor Decomposition”. In: *Conference on Computational Learning Theory (COLT)* (cit. on pp. 535, 536).

Rong Ge, Chi Jin, and Yi Zheng (2017). “No Spurious Local Minima in Nonconvex Low Rank Problems: A Unified Geometric Analysis”. arXiv: 1704.00708 (cit. on p. 534).

Rong Ge, Jason D Lee, and Tengyu Ma (2016). “Matrix completion has no spurious local minimum”. In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 2973–2981 (cit. on pp. 534, 535).

Saul B Gelfand and S Mitter (1991). “Recursive stochastic algorithms for global optimization in $\mathbb{R}^d$”. *SIAM Journal on Control and Optimization* 29, pp. 999–1018. MR: 1110084 (cit. on p. 541).

Saeed Ghadimi and Guanghui Lan (2016). “Accelerated gradient methods for nonconvex nonlinear and stochastic programming”. *Mathematical Programming* 156.1-2, pp. 59–99. MR: 3459195.

Ernst Hairer, Christian Lubich, and Gerhard Wanner (2006). *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*. New York: Springer. MR: 2221614 (cit. on p. 529).

Morgan A Hanson (1999). “Invexity and the Kuhn–Tucker theorem”. *Journal of Mathematical Analysis and Applications* 236.2, pp. 594–604. MR: 1704601.

Radoslav Harman and Vladimír Lacko (2010). “On decompositional algorithms for uniform sampling from n-spheres and n-balls”. *Journal of Multivariate Analysis* 101.10, pp. 2297–2304. MR: 2719863.

Richard A Harshman (1970). “Foundations of the PARAFAC procedure: Models and conditions for an “explanatory” multi-modal factor analysis”. *UCLA Working Papers in Phonetics* 16.1, p. 84.

Daniel Hsu and Sham M Kakade (2013). “Learning mixtures of spherical gaussians: moment methods and spectral decompositions”. In: *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*. ACM, pp. 11–20. MR: 3385380.

Furong Huang, UN Niranjan, Mohammad Umar Hakeem, and Animashree Anandkumar (2013). “Fast Detection of Overlapping Communities via Online Tensor Methods”. *arXiv:1309.0787*.

Aapo Hyvarinen (1999). “Fast ICA for noisy data using Gaussian moments”. In: *Circuits and Systems*. Vol. 5, pp. 57–61.

Aapo Hyvärinen, Juha Karhunen, and Erkki Oja (2004). *Independent component analysis*. Vol. 46. John Wiley & Sons.

Masato Inoue, Hyeyoung Park, and Masato Okada (2003). “On-Line Learning Theory of Soft Committee Machines with Correlated Hidden Units–Steepest Gradient Descent and Natural Gradient Descent–”. *Journal of the Physical Society of Japan* 72.4, pp. 805–810.

Prateek Jain, Chi Jin, Sham M Kakade, and Praneeth Netrapalli (2015). "Computing matrix squareroot via non convex local search". arXiv: 1507.05854.

Prateek Jain, Sham M Kakade, Rahul Kidambi, Praneeth Netrapalli, and Aaron Sidford (2017). "Accelerating Stochastic Gradient Descent". arXiv: 1704.08227. MR: 3827111.

Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi (2013). "Low-rank matrix completion using alternating minimization". In: *Proceedings of the Forty-Fifth annual ACM symposium on Theory of Computing*, pp. 665–674. MR: 3210828.

Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan (2017). "How to escape saddle points efficiently". In: *International Conference on Machine Learning (ICML)* (cit. on p. 536).

Chi Jin, Praneeth Netrapalli, and Michael I Jordan (2017). "Accelerated gradient descent escapes saddle points faster than gradient descent". arXiv: 1711.10456 (cit. on pp. 537–539).

David S Johnson, Christos H Papadimitriou, and Mihalis Yannakakis (1988). "How easy is local search?" *Journal of Computer and System Sciences* 37.1, pp. 79–100. MR: 0973658.

Hamed Karimi, Julie Nutini, and Mark Schmidt (2016). "Linear convergence of gradient and proximal-gradient methods under the Polyak-Lojasiewicz condition". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, pp. 795–811.

Kenji Kawaguchi (2016). "Deep learning without poor local minima". In: *Advances In Neural Information Processing Systems (NIPS)*, pp. 586–594 (cit. on p. 534).

Krzysztof C Kiwiel (2001). "Convergence and efficiency of subgradient methods for quasiconvex minimization". *Mathematical Programming* 90.1, pp. 1–25. MR: 1819784.

Tamara G Kolda (2001). "Orthogonal tensor decompositions". *SIAM Journal on Matrix Analysis and Applications* 23.1, pp. 243–255. MR: 1856608.

Walid Krichene, Alexandre Bayen, and Peter Bartlett (2015). "Accelerated mirror descent in continuous and discrete time". In: *Advances in Neural Information Processing Systems (NIPS) 27* (cit. on p. 528).

Jason D Lee, Max Simchowitz, Michael I Jordan, and Benjamin Recht (2016). "Gradient descent only converges to minimizers". In: *Conference on Computational Learning Theory (COLT)*, pp. 1246–1257 (cit. on p. 534).

Yin Tat Lee and Aaron Sidford (2013). "Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems". In: *Foundations of Computer Science (FOCS)*. IEEE, pp. 147–156. MR: 3246216.

Kfir Y Levy (2016). "The Power of Normalization: Faster Evasion of Saddle Points". arXiv: 1611.04831.

Huan Li and Zhouchen Lin (2017). "Provable Accelerated Gradient Method for Nonconvex Low Rank Optimization". arXiv: 1702.04959.

Olvi L Mangasarian (1965). "Pseudo-convex functions". *Journal of the Society for Industrial & Applied Mathematics, Series A: Control* 3.2, pp. 281–290. MR: 0191659.

Song Mei, Theodor Misiakiewicz, Andrea Montanari, and Roberto I Oliveira (2017). "Solving SDPs for synchronization and MaxCut problems via the Grothendieck inequality". In: *Conference on Computational Learning Theory (COLT)*, pp. 1476–1515 (cit. on p. 534).

Ion Necoara, Yurii Nesterov, and Francois Glineur (2015). "Linear convergence of first order methods for non-strongly convex optimization". arXiv: 1504.06298.

Arkadii Nemirovskii and David Yudin (1983). *Problem Complexity and Method Efficiency in Optimization*. New York: John Wiley. MR: 0702836 (cit. on pp. 524, 525).

Yurii Nesterov (1983). "A method of solving a convex programming problem with convergence rate O(1/k2)". *Soviet Mathematics Doklady* 27, pp. 372–376 (cit. on p. 525).

– (1998). *Introductory Lectures on Convex Programming*. New York: Springer. MR: 2142598 (cit. on pp. 524–526, 535).

– (2004). *Introductory Lectures on Convex Optimization*. Vol. 87. Springer Science & Business Media. MR: 2142598.

– (2008). "Accelerating the cubic regularization of Newton's method on convex problems". *Mathematical Programming* 112.1, pp. 159–181. MR: 2327005.

– (2012a). "Efficiency of coordinate descent methods on huge-scale optimization problems". *SIAM Journal of Optimization* 22, pp. 341–362. MR: 2968857 (cit. on p. 523).

– (2012b). "Efficiency of coordinate descent methods on huge-scale optimization problems". *SIAM Journal on Optimization* 22.2, pp. 341–362. MR: 2968857.

Yurii Nesterov and Boris T Polyak (2006). "Cubic regularization of Newton method and its global performance". *Mathematical Programming* 108.1, pp. 177–205. MR: 2229459 (cit. on pp. 528, 535).

Praneeth Netrapalli, Prateek Jain, and Sujay Sanghavi (2013). "Phase retrieval using alternating minimization". In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 2796–2804. MR: 3385838.

Michael O'Neill and Stephen J Wright (2017). "Behavior of Accelerated Gradient Methods Near Critical Points of Nonconvex Problems". arXiv: 1706.07993.

Bruno A Olshausen and David J Field (1997). "Sparse coding with an overcomplete basis set: A strategy employed by V1?" *Vision Research* 37.23, pp. 3311–3325.

Panos M Pardalos (1993). *Complexity in numerical optimization*. World Scientific. MR: 1358836.

Dohyung Park, Anastasios Kyrillidis, Constantine Caramanis, and Sujay Sanghavi (2017). "Non-square matrix sensing without spurious local minima via the Burer-Monteiro approach". In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 65–74.

Boris T Polyak (1963). "Gradient methods for the minimisation of functionals". *USSR Computational Mathematics and Mathematical Physics* 3.4, pp. 864–878. MR: 0158568.

Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan (2012). “Making Gradient Descent Optimal for Strongly Convex Stochastic Optimization”. In: *ICML*, pp. 449–456.

Magnus Rattray, David Saad, and Shun-ichi Amari (1998). “Natural gradient descent for on-line learning”. *Physical Review Letters* 81.24, p. 5461.

Sashank J Reddi, Ahmed Hefny, Suvrit Sra, Barnabas Poczos, and Alex Smola (2016). “Stochastic variance reduction for nonconvex optimization”. In: *International Conference on Machine Learning (ICML)*, pp. 314–323.

Clément W Royer and Stephen J Wright (2017). “Complexity analysis of second-order line-search algorithms for smooth nonconvex optimization”. arXiv: 1706.03131. MR: 3799071.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams (1988). “Learning representations by back-propagating errors”. *Cognitive Modeling* 5.

David Saad and Sara A Solla (1995). “On-line learning in soft committee machines”. *Physical Review E* 52.4, p. 4225.

Andrew M Saxe, James L McClelland, and Surya Ganguli (2013). “Exact solutions to the nonlinear dynamics of learning in deep linear neural networks”. *arXiv:1312.6120*.

Shai Shalev-Shwartz, Ohad Shamir, Karthik Sridharan, and Nathan Srebro (2009). “Stochastic convex optimization”. In: *Conference on Computational Learning Theory (COLT)*.

Shai Shalev-Shwartz and Tong Zhang (2014). “Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization”. In: *International Conference on Machine Learning (ICML)*, pp. 64–72. MR: 3033340.

Weijie Su, Stephen Boyd, and Emmanuel J Candes (2016). “A differential equation for modeling Nesterov’s accelerated gradient method: theory and insights”. *Journal of Machine Learning Research* 17.153, pp. 1–43. MR: 3555044 (cit. on p. 528).

Ju Sun, Qing Qu, and John Wright (2016a). “A geometric analysis of phase retrieval”. In: *IEEE International Symposium on Information Theory (ISIT)*. IEEE, pp. 2379–2383 (cit. on p. 535).

– (2016b). “Complete dictionary recovery over the sphere I: Overview and the geometric picture”. *IEEE Transactions on Information Theory* 63, pp. 853–884. MR: 3604646 (cit. on p. 535).

Ruoyu Sun and Zhi-Quan Luo (2016). “Guaranteed matrix completion via non-convex factorization”. *IEEE Transactions on Information Theory* 62.11, pp. 6535–6579. MR: 3565131.

Andre Wibisono, Ashia C Wilson, and Michael I Jordan (2016). “A variational perspective on accelerated methods in optimization”. *Proceedings of the National Academy of Sciences* 133, E7351–E7358. MR: 3582442 (cit. on pp. 526–529, 531).

Ashia C Wilson, Benjamin Recht, and Michael I Jordan (2016). “A Lyapunov analysis of momentum methods in optimization”. arXiv: 1611.02635 (cit. on p. 528).

Stephen J Wright and Jorge Nocedal (1999). *Numerical optimization*. Vol. 2. Springer New York. MR: 1713114.

Qinqing Zheng and John Lafferty (2016). "Convergence Analysis for Rectangular Matrix Completion Using Burer-Monteiro Factorization and Gradient Descent". arXiv: 1605.07051.

James Y. Zou, Daniel Hsu, David C. Parkes, and Ryan P. Adams (2013). "Contrastive Learning Using Spectral Methods". In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 2238–2246.

Received 2017-12-01.

Michael I. Jordan
jordan@cs.berkeley.edu
jordan@stat.berkeley.edu
University of California, Berkeley