Average-Case Acceleration Through Spectral Density Estimation

Fabian Pedregosa [*1] **Damien Scieur** [*2]

Abstract

We develop a framework for the average-case analysis of random quadratic problems and derive algorithms that are optimal under this analysis. This yields a new class of methods that achieve acceleration given a model of the Hessian's eigenvalue distribution. We develop explicit algorithms for the uniform, Marchenko-Pastur, and exponential distributions. These methods have a simple momentum-like update, in which each update only makes use on the current gradient and previous two iterates. Furthermore, the momentum and step-size parameters can be estimated without knowledge of the Hessian's smallest singular value, in contrast with classical accelerated methods like Nesterov acceleration and Polyak momentum. Through empirical benchmarks on quadratic and logistic regression problems, we identify regimes in which the the proposed methods improve over classical (worst-case) accelerated methods.

1. Introduction

The traditional analysis of optimization algorithms is a worst-case analysis (Nemirovski, 1995; Nesterov, 2004). This type of analysis provides a complexity bound for *any* input from a function class, no matter how unlikely. However, hard-to-solve inputs might rarely occur in practice, in which case these complexity bounds might not be representative of the observed running time.

Average-case analysis provides instead the *expected* complexity of an algorithm over a class of problems, and is more representative of its typical behavior. While the average-case analysis is standard for analyzing sorting (Knuth, 1997) and cryptography (Katz & Lindell, 2014) algorithms, little is known of the average-complexity of optimization algorithms.

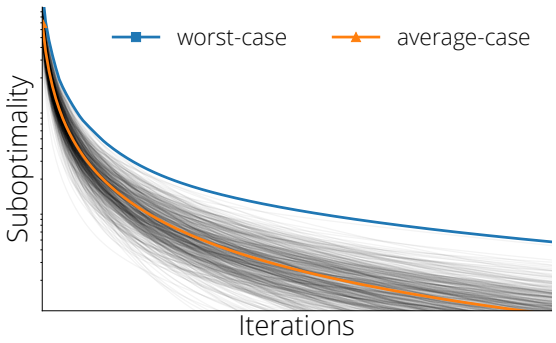

Figure 1. A worst-case analysis can lead to misleading results where the worst-case running times is much worse than the observed running time. Above: convergence of individual random square least squares in gray, while the average suboptimality (orange) is well below the worst-case (blue).

Our **first contribution** (§2) is to develop an average-case analysis of optimization methods on quadratic objectives. A crucial difference with the worst-case analysis is that it depends on the expected spectrum of the Hessian, rather on just the extremal eigenvalues.

While it is unfeasible to assume knowledge of the full spectrum, recent works have shown that the spectrum of large deep learning models is highly predictable and can be approximated using classical models from random matrix theory, such as the Marchenko-Pastur distribution (Sagun et al., 2017; Martin & Mahoney, 2018).

A **second contribution** (§3) is to exploit this regularity to develop practical methods that are optimal under the average-case analysis. We consider different parametric models for the expected spectrum such as the Marchenko-Pastur, uniform, and exponential distributions, and derive for each model an average-case optimal algorithm. These are all momentum-like methods where the hyper-parameter of the distribution can be estimated without knowledge of the smallest eigenvalue.

Finally, we compare these average-case optimal methods on synthetic and real datasets. We identify regimes where average-case exhibits a large computational gain with respect to respect to traditional accelerated methods.

[*]Equal contribution [1]Google Research [2]Samsung SAIT AI Lab, Montreal. Correspondence to: Fabian Pedregosa <pedregosa@google.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

1.1. Related Work

We comment on two of the main ideas behind the proposed methods.

Polynomial-Based Iterative Methods. Our work draws heavily from the classical framework of polynomial-based iterative methods (Fischer, 1996) that can be traced back to the Chebyshev iterative method of Flanders & Shortley (1950) and was later instrumental in the development of the celebrated conjugate gradient method (Hestenes et al., 1952). More recently, this framework has been used to derive accelerated gossip algorithms (Berthier et al., 2018) and accelerated algorithms for smooth games (Azizian et al., 2020), to name a few. Although originally derived for the worst-case analysis of optimization algorithms, in this work we extend this framework to analyze also the average-case runtime.

In concurrent work, Lacotte & Pilanci (2020) derive average-case optimal methods for a different class of methods where the update can be multiplied by a preconditioner matrix.

Spectral Density Estimation. The realization that deep learning networks behave as linear models in the infinite-width limit (Jacot et al., 2018; Novak et al., 2018) has sparked a renewed interest in the study of spectral density of large matrices. The study of spectral properties of these very large models is made possible thanks to improved tools (Ghorbani et al., 2019) and more precise theoretical models (Jacot et al., 2019; Pennington & Worah, 2017).

Notation. Throughout the paper we denote vectors in lowercase boldface ($\boldsymbol{x}$) and matrices in uppercase boldface letters ($\boldsymbol{H}$). Probability density functions and eigenvalues are written in Greek letters (μ, λ), while polynomials are written in uppercase Latin letter (P, Q). We will sometimes omit integration variable, with the understanding that $\int \varphi \,\mathrm{d}\mu$ is a shorthand for $\int \varphi(\lambda)\,\mathrm{d}\mu(\lambda)$.

2. Average-Case Analysis

In this section we introduce the average-case analysis framework for random quadratic problems. The main result is Theorem 2.1, which relates the expected error with other quantities that will be easier to manipulate, such as the residual polynomial. This is a convenient representation of an optimization method that will allow us in the next section to pose the problem of finding an optimal method as a best approximation problem in the space of polynomials.

Let $\boldsymbol{H} \in \mathbb{R}^{d\times d}$ be a random symmetric positive-definite matrix and $\boldsymbol{x}^\star \in \mathbb{R}^d$ a random vector. These elements determine the following (random) quadratic minimization problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \left\{ f(\boldsymbol{x}) \stackrel{\text{def}}{=} \frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}^\star)^\top \boldsymbol{H}(\boldsymbol{x}-\boldsymbol{x}^\star) \right\}. \tag{OPT}$$

Our goal is to quantify the expected error $\mathbb{E}\,\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|$, where $\boldsymbol{x}_t$ is the t-th update of a first-order method starting from $\boldsymbol{x}_0$ and $\mathbb{E}$ is the expectation over the random $\boldsymbol{H}, \boldsymbol{x}_0, \boldsymbol{x}^\star$.

Remark 1 *Problem OPT subsumes the quadratic minimization problem* $\min_{\boldsymbol{x}} \boldsymbol{x}^\top \boldsymbol{H}\boldsymbol{x} + \boldsymbol{b}^\top \boldsymbol{x} + c$*, but the notation above will be more convenient for our purposes.*

Remark 2 *The expectation in the expected error* $\mathbb{E}\,\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2$ *is over the inputs and not over any randomness of the algorithm, as would be common in the stochastic literature. In this paper we will only consider deterministic algorithms.*

To solve OPT, we will consider *first-order methods*. These are methods in which the sequence of iterates $\boldsymbol{x}_t$ is in the span of previous gradients, i.e.,

$$\boldsymbol{x}_{t+1} \in \boldsymbol{x}_0 + \mathbf{span}\{\nabla f(\boldsymbol{x}_0), \ldots, \nabla f(\boldsymbol{x}_t)\}\,. \tag{1}$$

This class of algorithms includes for instance gradient descent and momentum, but not quasi-Newton methods, since the preconditioner could allow the iterates to go outside of the span. Furthermore, we will only consider *oblivious* methods, that is, methods in which the coefficients of the update are known in advance and don't depend on previous updates. This leaves out some methods that are specific to quadratic problems like conjugate gradient.

From First-Order Method to Polynomials. There is an intimate link between first order methods and polynomials that simplifies the analysis on quadratic objectives. Using this link, we will be able to assign to each optimization method a polynomial that determines its convergence. The next Proposition gives a precise statement:

Proposition 2.1 *(Hestenes et al., 1952) Let* $\boldsymbol{x}_t$ *be generated by a first-order method. Then there exists a polynomial* P_t *of degree* t *such that* $P_t(0) = 1$ *that verifies*

$$\boldsymbol{x}_t - \boldsymbol{x}^\star = P_t(\boldsymbol{H})(\boldsymbol{x}_0 - \boldsymbol{x}^\star)\,. \tag{2}$$

Following Fischer (1996), we will refer to this polynomial P_t as the *residual polynomial*.

Example 1 ***(Gradient descent).*** *The residual polynomial associated with the gradient descent method has a remark-*

ably simple form. Subtracting $\boldsymbol{x}^\star$ on both sides of the gradient descent update $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \gamma \boldsymbol{H}(\boldsymbol{x} - \boldsymbol{x}^\star)$ gives

$$\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star = (I - \gamma \boldsymbol{H})(\boldsymbol{x}_t - \boldsymbol{x}^\star) \tag{3}$$
$$= \cdots = (\boldsymbol{I} - \gamma \boldsymbol{H})^t (\boldsymbol{x}_0 - \boldsymbol{x}^\star) \tag{4}$$

and so the residual polynomial is $P_t(\lambda) = (1 - \gamma\lambda)^t$.

A convenient way to collect statistics on the spectrum of a matrix is through its *empirical spectral distribution*, which we now define.

Definition 3 ***(Empirical/Expected spectral distribution).*** *Let $\boldsymbol{H}$ be a random matrix with eigenvalues $\{\lambda_1, \ldots, \lambda_d\}$. The **empirical spectral distribution** of $\boldsymbol{H}$, called $\mu_{\boldsymbol{H}}$, is the probability measure*

$$\mu_{\boldsymbol{H}}(\lambda) \overset{def}{=} \tfrac{1}{d} \textstyle\sum_{i=1}^d \delta_{\lambda_i}(\lambda) \,, \tag{5}$$

where δ_{λ_i} is the Dirac delta, a distribution equal to zero everywhere except at λ_i and whose integral over the entire real line is equal to one.

Since $\boldsymbol{H}$ is random, the empirical spectral distribution $\mu_{\boldsymbol{H}}$ is a random measure. Its expectation over $\boldsymbol{H}$,

$$\mu \overset{def}{=} \mathbb{E}_{\boldsymbol{H}}[\mu_{\boldsymbol{H}}] \,, \tag{6}$$

*is called the **expected spectral distribution**.*

Example 2 (Marchenko-Pastur distribution) *Consider a matrix $\boldsymbol{A} \in \mathbb{R}^{n \times d}$, where each entry is an i.i.d. random variables with mean zero and variance σ^2. Then it is known that the expected spectral distribution of $\boldsymbol{H} = \frac{1}{n}\boldsymbol{A}^\top \boldsymbol{A}$ converges to the Marchenko-Pastur distribution (Marchenko & Pastur, 1967) as n and $d \to \infty$ at a rate in which $d/n \to r \in (0, \infty)$. The Marchenko-Pastur distribution μ_{MP} is defined as*

$$\max\{1 - \tfrac{1}{r}, 0\}\delta_0(\lambda) + \frac{\sqrt{(L-\lambda)(\lambda-\ell)}}{2\pi\sigma^2 r \lambda} 1_{\lambda \in [\ell, L]} \, \mathrm{d}\lambda, \tag{7}$$

where $\ell \overset{def}{=} \sigma^2(1 - \sqrt{r})^2$, $L \overset{def}{=} \sigma^2(1 + \sqrt{r})^2$ are the extreme nonzero eigenvalues and δ_0 is a Dirac delta at zero (which disappears for $r \geq 1$).

In practice, the Marchenko-Pastur distribution is often a good approximation to the spectral distribution of high-dimensional models, even for data that might not verify the i.i.d. assumption, like the one in Figure 2.

Before presenting the main result of this section, we state one simplifying assumption on the initialization that we make throughout the rest of the paper.

Assumption 1 *We assume that $\boldsymbol{x}_0 - \boldsymbol{x}^\star$ is independent of $\boldsymbol{H}$ and*

$$\mathbb{E}\,(\boldsymbol{x}_0 - \boldsymbol{x}^\star)(\boldsymbol{x}_0 - \boldsymbol{x}^\star)^\top = R^2 \boldsymbol{I} \,. \tag{8}$$

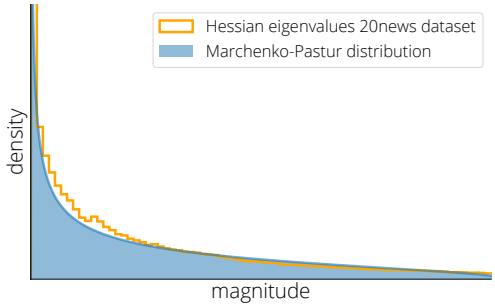

Figure 2. Eigenvalue distribution of high dimensional problems can often be approximated with simple models. Above, a fit to the Marchenko-Pastur distribution (blue) of the Hessian's eigenvalues histogram (yellow) in a least square problem with data the News20 dataset (20k features, Keerthi & DeCoste (2005)).

This assumption is verified for instance when both $\boldsymbol{x}_0$ and $\boldsymbol{x}^\star$ are drawn independently from a distribution with scaled identity covariance. It is also verified in a least squares problem of the form $\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2$, where the target vector verifies $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}^\star$.[1]

Theorem 2.1 *Let $\boldsymbol{x}_t$ be generated by a first-order method, associated to the polynomial P_t. Then we can decompose the expected error at iteration t as*

$$\mathbb{E}\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 = \overbrace{R^2}^{\text{initialization}} \int_{\mathbb{R}} \underbrace{P_t^2}_{\text{algorithm}} \overbrace{\mathrm{d}\mu}^{\text{problem}} \,. \tag{9}$$

This identity represents the expected error of an algorithm in terms of three interpretable quantities:

1. The distance to optimum at initialization enters through the constant R, which is the diagonal scaling in Assumption 1.
2. The optimization method enters in the formula through its residual polynomial P_t. The main purpose of the rest of the paper will be to find optimal choices for this polynomial.
3. The difficulty of the problem class enters through the expected spectral distribution $\mu \overset{\text{def}}{=} \mathbb{E}_{\boldsymbol{H}}[\mu_{\boldsymbol{H}}]$.

Remark 4 *Although we will not use it in this paper, similar formulas can be derived for the objective and gradient*

[1] After the first version of this paper appeared online, Paquette et al. (2020) generalized these results to least squares problems with a noisy target vector $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}^\star + \boldsymbol{\eta}$.

suboptimality:

$$\mathbb{E}[f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star)] = R^2 \int_{\mathbb{R}} P_t^2(\lambda)\lambda \,\mathrm{d}\mu(\lambda) \tag{10}$$

$$\mathbb{E}\,\|\nabla f(\boldsymbol{x}_t)\|^2 = R^2 \int_{\mathbb{R}} P_t^2(\lambda)\lambda^2 \,\mathrm{d}\mu(\lambda)\,. \tag{11}$$

3. Average-Case Acceleration

The framework developed in the previous section opens the door to exploring the question of optimality with respect to the average-case complexity. Does a method exist, that is optimal in this analysis? If so, what is this method?

In this section is to give a constructive answer. We will first introduce some concepts from the theory of orthogonal polynomials which will be necessary to develop optimal methods.

Definition 5 *Let α be a non-decreasing function such that $\int Q \,\mathrm{d}\alpha$ is finite for all polynomials Q. We will say that the sequence of polynomials $P_0, P_1, \ldots$ is orthogonal with respect to $\mathrm{d}\alpha$ if P_i has degree i and*

$$\int_{\mathbb{R}} P_i\, P_j \,\mathrm{d}\alpha \begin{cases} = 0 & \text{if } i \neq j \\ > 0 & \text{if } i = j \end{cases}. \tag{12}$$

*Furthermore, if they verify $P_i(0) = 1$ for all i, we call these **residual orthogonal** polynomials.*

While a degree t polynomial needs in principle t scalars to be fully specified, residual orthogonal polynomials admit a compressed recursive representation in which each polynomial can be represented using only *two* scalars and the previous two residual orthogonal polynomials. This compressed representation is known as the *three-term recurrence* and will be crucial to ensure optimal methods enjoy the low memory and low per-iteration cost of momentum methods. The following lemma states this property more precisely.

Lemma 6 (Three-term recurrence) *(Fischer, 1996, §2.4) Any sequence of residual orthogonal polynomials $P_1, P_2, \ldots$ verifies the following three-term recurrence*

$$P_t(\lambda) = (a_t + b_t\lambda)P_{t-1}(\lambda) + (1-a_t)P_{t-2}(\lambda) \tag{13}$$

for some scalars a_t, b_t, with $a_{-1} = a_0 = 1$ and $b_{-1} = 0$.

Remark 7 *Although there exist algorithms to compute the coefficients a_t and b_t recursively (e.g., Fischer (1996, Algorithm 2.4.2)), numerical stability and computational costs make these methods unfeasible for our purposes. In Sections 4–6 we will see how to compute these coefficients for specific distributions of μ.*

We have now all ingredients to state the main result of this section, a simple algorithm with optimal average-case complexity.

Theorem 3.1 *Let P_t be the residual orthogonal polynomials of degree t with respect to the weight function $\lambda \,\mathrm{d}\mu(\lambda)$, and let a_t, b_t be the constants associated with its three-term recurrence. Then the algorithm*

$$\begin{aligned} \boldsymbol{x}_t = \boldsymbol{x}_{t-1} + (1-a_t)(\boldsymbol{x}_{t-2} - \boldsymbol{x}_{t-1}) \\ + b_t \nabla f(\boldsymbol{x}_{t-1})\,, \end{aligned} \tag{14}$$

has the smallest expected error $\mathbb{E}\,\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|$ over the class of oblivious first-order methods. Moreover, its expected error is:

$$\mathbb{E}\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 = R^2 \int_{\mathbb{R}} P_t \,\mathrm{d}\mu\,. \tag{15}$$

Remark 8 *Algorithm (14), although being optimal over the space of all first-order methods, does not require storage of previous gradients. Instead, it has a very convenient momentum-like form and only requires storing two d-dimensional vectors.*

Remark 9 ***(Relationship with Conjugate Gradient).*** *The derivation of the proposed method bears a strong resemblance with the Conjugate Gradient method (Hestenes et al., 1952). One key conceptual difference is that the conjugate gradient constructs the optimal polynomial for the* empirical spectral distribution*, while we construct a polynomial that is optimal only for the expected spectral distribution. A more practical advantage is that the proposed method is more amenable to non-quadratic minimization.*

The previous theorem gives a recipe to construct an optimal algorithm from a sequence of residual orthogonal polynomials. However, in practice we may not have access to the expected spectral distribution μ, let alone its sequence of residual orthogonal polynomials.

The next sections are devoted to solving this problem through different parametric assumption on the spectral distribution: exponential (§4), Marchenko-Pastur (§5), and uniform (§6).

4. Optimal Method under the Exponential Distribution

In this section we assume that the expected spectral distribution follows an exponential distribution:

$$\mathrm{d}\mu = \frac{1}{\lambda_0} e^{-\lambda/\lambda_0}, \quad \lambda \in [0, \infty)\,, \tag{16}$$

where λ_0 is a free parameter of the distribution. In this case the largest eigenvalue ($\equiv$ Lipschitz gradient constant) is unbounded, and so this setting can be identified the convex but non-smooth setting.

A consequence of Theorem 3.1 is that deriving the optimal algorithm is equivalent to finding the sequence of residual orthogonal polynomials with respect to the weight function $\lambda \, \mathrm{d}\mu(\lambda)$. Orthogonal (non-residual) polynomials for this weight function when $\mathrm{d}\mu$ is an exponential distribution have been studied under the name of *generalized Laguerre polynomials* (Abramowitz et al., 1972, p. 781). There is a constant factor between the orthogonal and *residual* orthogonal polynomial, and so the latter can be constructed from the former by finding the appropriate normalization that makes the polynomial residual. This normalization is given by the following lemma:

Lemma 10 *The sequence of scaled Laguerre polynomials*

$$P_0(\lambda) = 1\,, \quad P_1(\lambda) = 1 - \frac{\lambda_0}{2}\lambda\,, \tag{17}$$
$$P_t(\lambda) = \left(\frac{2}{t+1} - \frac{\lambda_0}{t}\lambda\right) P_{t-1}(\lambda) + \left(\frac{t-1}{t+1}\right) P_{t-2}(\lambda)$$

are a family of residual orthogonal polynomials with respect to the measure $\frac{\lambda}{\lambda_0} e^{-\lambda/\lambda_0}$.

From the above Lemma, we can derive the method with best expected error for the decaying exponential distribution. The resulting algorithm is surprisingly simple:

Decaying Exponential Acceleration

Input: Initial guess $\boldsymbol{x}_0$, $\lambda_0 > 0$
Algorithm: Iterate over $t = 1...$

$$\boldsymbol{x}_t = \boldsymbol{x}_{t-1} + \frac{t-1}{t+1}(\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-2}) - \frac{\lambda_0}{t+1}\nabla f(\boldsymbol{x}_{t-1}) \tag{EXP}$$

Remark 11 *In this distribution, and unlike in the MP that we will see in the next section, the largest eigenvalue is not bounded, so this method is more akin to subgradient descent and not gradient descent. Note that because of this, the step-size is decreasing.*

Remark 12 *Note the striking similarity with the averaged gradient method of Polyak & Juditsky (1992), which reads*

$$\boldsymbol{\theta}_t = \boldsymbol{y}_{t-1} - \gamma \nabla f(\boldsymbol{\theta})$$
$$\boldsymbol{x}_t = \left(\frac{t-1}{t}\right) \boldsymbol{x}_{t-1} + \frac{1}{t}\boldsymbol{\theta}$$

This algorithm admits the following momentum-based form derived by Flammarion & Bach (2015, §2.2):

$$\boldsymbol{y}_t = t\boldsymbol{x}_{t-1} - (t-1)\boldsymbol{x}_{t-2}$$
$$\boldsymbol{x}_t = \boldsymbol{x}_{t-1} + \frac{t-1}{t+1}(\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-2}) - \frac{\gamma}{t+1}\nabla f(\boldsymbol{y}_t)\,.$$

The difference between EXP and this formula is where the gradient is computed. In EXP, the gradient is evaluated at the current iterate $\boldsymbol{x}_t$, while in Polyak averaging, the gradient is computed at the extrapolated iterate iterate $\boldsymbol{y}_t$.

Parameters Estimation. The exponential distribution has a free parameter λ_0. Since the expected value of this distribution is $1/\lambda_0$, we can estimate the parameter λ_0 from the sample mean, which is $\frac{1}{d}\mathrm{tr}(\boldsymbol{H})$. Hence, we fit this parameter as $\lambda_0 = d/\mathrm{tr}(\boldsymbol{H})$.

4.1. Rate of Convergence

For this algorithm we are able to give a simple expression for the expected error. The next theorem shows that it converges at rate $\mathcal{O}(1/t)$.

Lemma 13 *If we apply Algorithm (EXP) to problem (OPT), where the spectral distribution of $\boldsymbol{H}$ is the decaying exponential $\mathrm{d}\mu(\lambda) = e^{-\lambda/\lambda_0}$, then*

$$\mathbb{E}\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 = \frac{R^2}{\lambda_0(t+1)}\,. \tag{18}$$

In the case of the convex non-smooth functions, optimal algorithm achieves a (worst-case) rate which is $O(1/\sqrt{t})$ (see for instance (Nesterov, 2009)). We thus achieve acceleration, by assuming the function quadratic with the exponential as average spectral density.

5. Optimal Method under the Marchenko-Pastur Distribution

In this section, we will derive the optimal method under the Marchenko-Pastur distribution μ_{MP}, introduced in Example 2. As in the previous section, the first step is to construct a sequence of residual orthogonal polynomials.

Theorem 5.1 *The following sequence polynomials are orthogonal with respect to the weight function $\lambda \, \mathrm{d}\mu_{\mathrm{MP}}(\lambda)$:*

$$P_0(\lambda) = 0\,,\ P_1(\lambda) = 1\ \delta_0 = 0, \rho = \frac{1+r}{\sqrt{r}}$$
$$\delta_t = (-\rho - \delta_{t-1})^{-1} \tag{19}$$
$$P_t = \Big(- \rho\delta_t + \frac{\delta_t}{\sigma^2\sqrt{r}}\lambda\Big) P_{t-1} + (1 - \rho\delta_t)\, P_{t-2}\,.$$

We show in Appendix C.2 that these polynomials are shifted Chebyshev polynomials *of the second kind*. Surprisingly, the Chebyshev of the first kind are used to minimize

Marchenko-Pastur Acceleration

Input: Initial guess $\boldsymbol{x}_0$, MP parameters r, σ^2
Preprocessing: $\rho = \frac{1+r}{\sqrt{r}}$, $\delta_0 = 0$, $\boldsymbol{x}_1 = \boldsymbol{x}_0 - \frac{1}{(1+r)\sigma^2}\nabla f(\boldsymbol{x}_0)$

Algorithm (Optimal): Iterate over $t = 1...$

$$\delta_t = (-\rho - \delta_{t-1})^{-1}, \quad \boldsymbol{x}_t = \boldsymbol{x}_{t-1} + (1 + \rho\delta_t)(\boldsymbol{x}_{t-2} - \boldsymbol{x}_{t-1}) + \delta_t \frac{\nabla f(\boldsymbol{x}_{t-1})}{\sigma^2\sqrt{r}} \qquad \text{(MP-OPT)}$$

Algorithm (Asymptotic variant): Iterate over $t = 1...$

$$\boldsymbol{x}_t = \boldsymbol{x}_{t-1} - \min\{r^{-1}, r\}(\boldsymbol{x}_{t-2} - \boldsymbol{x}_{t-1}) - \frac{1}{\sigma^2}\min\{1, r^{-1}\}\nabla f(\boldsymbol{x}_{t-1}) \qquad \text{(MP-ASY)}$$

over the worst case. From the optimal polynomial, we can write the optimal algorithm for OPT when the spectral density function of $\boldsymbol{H}$ is the Marchenko-Pastur distribution. By using Theorem 3.1, we obtain the "Marchenko-Pastur accelerated" method, described in Algorithm MP-OPT.

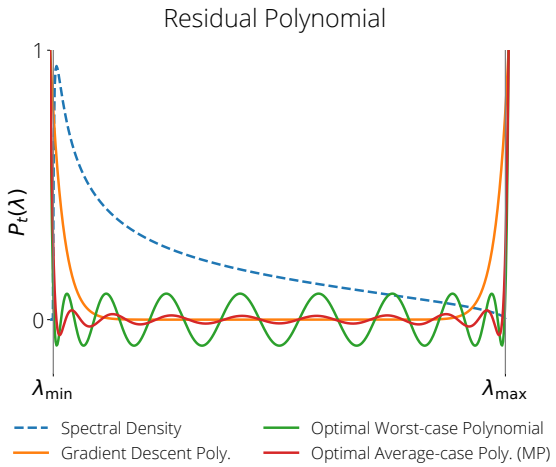

Figure 3. **Residual polynomials** for gradient descent, the minimax optimal polynomial (Chebyshev of the first kind), and the optimal polynomial for the MP distribution, overlayed with a MP distribution in dashed lines. Gradient descent has the highest value at the edges, which is in line with its known suboptimal dependency on the condition number. The MP polynomial is closer to zero than the Chebyshev *except* at the borders, which Chebyshev is optimal by construction.

It's instructive to compare the residual polynomials of different methods to better understand their convergence properties. In Figure 5 we plot the residual polynomials for gradient descent, its classical worst-case analysis accelerated variant (Chebyshev polynomials) and the average-case analysis accelerated (under the MP distribution) variant above. In numbers, the worst-case convergence rate (max value in interval) is $\approx 0.88^t$ for Chebyshev and $\approx 0.93^t$ for MP, so Chebyshev's worst-case rate is 1.7 times smaller. However, the *expected rate* is $\approx 0.67^t$ for MP and $\approx 0.74^t$ for Chebyshev, making MP 1.4 times faster on average.

Asymptotic Behavior. The step-size and momentum terms of the Marchenko-Pastur accelerated method (MP-OPT) depend on the time-varying parameter δ_t. It's possible to compute the limit coefficients as $t \to \infty$. This requires to solve

$$\delta_\infty = (-\rho - \delta_\infty)^{-1}, \quad \text{thus} \quad \delta_\infty = -\sqrt{r} \quad \text{or} \quad \delta_\infty = \frac{-1}{\sqrt{r}}.$$

The sequence δ_t converges to $-\sqrt{r}$ when $r < 1$, otherwise it converges to $-1/\sqrt{r}$. Replacing the value δ_t in Algorithm MP-OPT by its asymptotic value gives the simplified variant MP-ASY.

Algorithm MP-OPT corresponds to a gradient descent with a variable momentum and step-size terms, all converging to a simpler one (Algorithm MP-ASY). However, even if we assume that the spectral density of the Hessian is the Marchenko-Pastur distribution, we still need to estimate the hyper-parameters σ and r. The next section proposes a way to estimate those parameters.

Hyper-Parameters Estimation. Algorithm MP-OPT and Algorithm MP-ASY require the knowledge of the parameters r and σ. To ensure convergence, it is desirable to scale the parameters such that the largest eigenvalue $\lambda_{\max}$ lies inside the support of the distribution, bounded by $\sigma^2(1+\sqrt{r})^2$. This gives us one equation to set two parameters. We will get another equation by matching r with the first moment of the MP distribution, which can be estimated as the trace of $\boldsymbol{H}$. More formally, the two conditions reads

$$\lambda_{\max}(\boldsymbol{H}) = \sigma^2(1+\sqrt{r})^2\,, \quad \frac{\gamma}{d}\operatorname{tr}(\boldsymbol{H}) = r\,. \qquad (20)$$

With the notation $\tau \overset{\text{def}}{=} \frac{1}{d}\operatorname{tr}(\boldsymbol{H})$, $\overset{\text{def}}{=} \lambda_{\max}(\boldsymbol{H})$ the parameters σ^2 and r are given by

$$\sigma^2 = \left(\frac{\sqrt{L}+\sqrt{\tau}}{L-\tau}\right)^2, \qquad r = \gamma\tau\,. \tag{21}$$

6. Optimal Method under the Uniform Distribution

We now focus on the (unnormalized) uniform distribution over $[\ell, L]$. This weight function is 1 if $\lambda \in [\ell, L]$ and 0 otherwise.

We show in Appendix C.3 that a sequence of orthogonal residual polynomials with respect to this density is a sequence of shifted *Legendre polynomials*. Legendre polynomials are orthogonal w.r.t. the uniform distribution in $[-1, 1]$ and are defined as

$$t\tilde{Q}_t(\lambda) = (2t-1)\lambda\tilde{Q}_{t-1}(\lambda) - (t-1)\tilde{Q}_{t-2}(\lambda)\,. \tag{22}$$

We detail in Appendix C.3 the necessary translation and normalization steps to obtain the sequence of residual orthogonal polynomials.

Uniform Acceleration

Input: Initial guess $\boldsymbol{x}_0 = \boldsymbol{x}_1$, ℓ and L.
Init: $e_{-1} = m_0 = 0$.

Algorithm: Iterate over $t = 1...$

$$\begin{aligned}
d_t &= -\frac{L+\ell}{2} + e_{t-1} \\
e_t &= \frac{-(L-\ell)^2 t^2}{4d_t(4t^2-1)} \qquad \text{(UNIF)} \\
m_t &= (d_t - e_t + m_{t-1}d_t e_{t-1})^{-1} \\
\boldsymbol{x}_t &= \boldsymbol{x}_{t-1} + \Big(1 - m_t(d_t - e_t)\Big)(\boldsymbol{x}_{t-2} - \boldsymbol{x}_{t-1}) \\
&\qquad + m_t \nabla f(\boldsymbol{x}_{t-1})
\end{aligned}$$

Like the Marchenko-Pastur accelerated gradient, the parameters ℓ and L can be estimated through the moment of the uniform distribution.

7. Experiments

We compare the proposed methods and classical accelerated methods on settings with varying degrees of mismatch with our assumptions. We first compare them on quadratics generated from a synthetic dataset, where the empirical spectral density is (approximately) a Marchenko-Pastur distribution. We then compare these methods on another quadratic problem, generated using two non-synthetic datasets, where the MP assumption breaks down. Finally, we compare some the applicable methods in a logistic regression problem. We will see that the proposed methods perform reasonably well in this scenario, although being far from their original quadratic deterministic setting. A full description of datasets and methods can be found in Appendix D.

Methods. We consider worst-case (solid lines) and average-case (dashed lines) accelerated methods. The method "Modified Polyak" runs Polyak momentum in the strongly convex case and defaults to the momentum method of (Ghadimi et al., 2015) in the non-strongly convex regime.

Synthetic Quadratics. We consider the least squares problem with objective function $f(\boldsymbol{x}) = \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2$, where each entry of $\boldsymbol{A} \in \mathbb{R}^{n\times d}$ and $\boldsymbol{b} \in \mathbb{R}^n$ are generated from an i.i.d. random Gaussian distribution. Using different ratios d/n we generate problems that are convex ($\ell = 0$) and strongly convex ($\ell > 0$).

Non-Synthetic Quadratics. The last two columns of Figure 4 compare the same methods, this time on two real (non-synthetic) datasets. We use two UCI datasets: digits[2] ($n = 1797, d = 64$), and breast cancer[3] ($n = 569, d = 32$).

Logistic Regression. Using synthetic i.i.d. data, we compare the proposed methods on a logistic regression problem. We generate two datasets, first one with $n > d$ and second one with $d < n$. Results are shown in Figure 5.

7.1. Discussion

Importance of Spectral Distributions. Average-case accelerated methods exhibit the largest gain when the eigenvalues follow the assumed expected spectral distribution. This is the case for the Marchenko-Pastur acceleration in the first two columns of Figure 4, where its convergence rate is almost identical to that of conjugate gradient, a method that is optimal for the *empirical* (instead of expected) spectral distribution.

Beyond Quadratic Optimization. Results on 4 different logistic regression problems were mixed. On the one hand, as for quadratic objectives, the Uniform acceleration method has a good overall performance. However, contrary to the quadratic optimization results, Marchenko-Pastur acceleration has only mediocre performance in 3 out of 4 ex-

[2] https://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits

[3] https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29

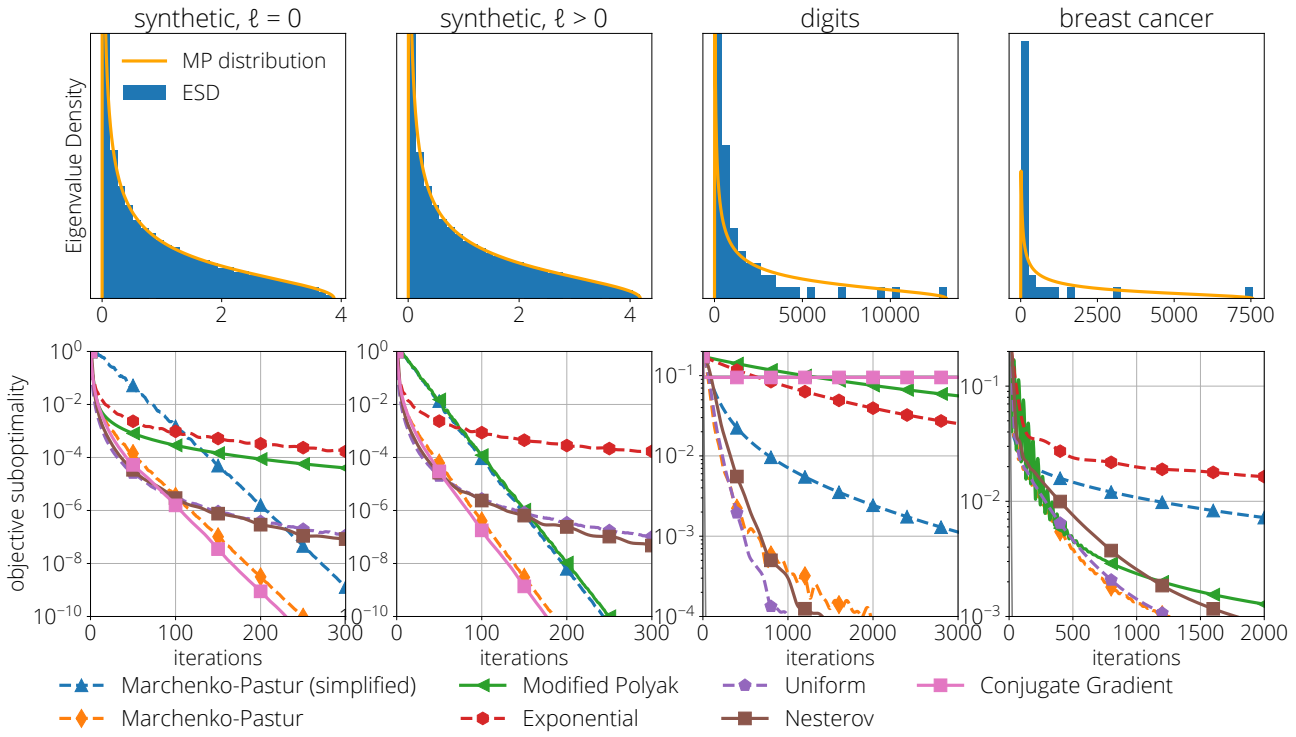

Figure 4. **Benchmark on quadratic objectives**. The top row shows the Hessians spectral density overlaid with the MP distribution, while the bottom row shows the objective suboptimality as a function of the number of iterations. In the first and second row, strongly convex and convex (respectively) synthetic problem with MP distribution. In all plots, the (Accelerated Gradient for) Marchenko-Pastur algorithm performs better than the classical worst-case analysis optimal method (Modified Polyak), and is close in performance with Conjugate Gradient one, which is optimal for the empirical (instead of expected) spectral density. Note from the first plot that the Marchenko-Pastur method maintains a linear convergence rate even in the presence of zero eigenvalues.

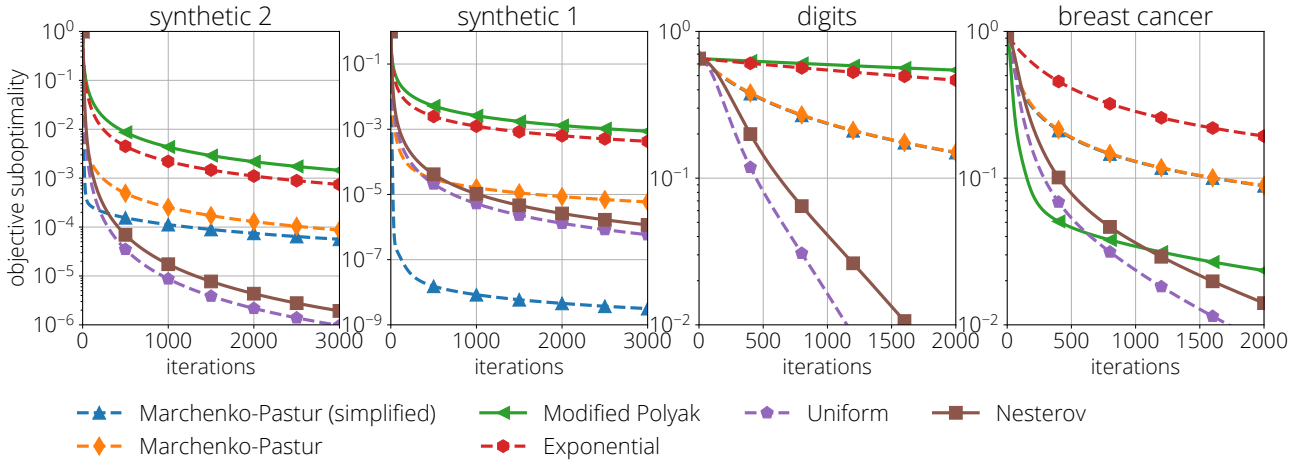

Figure 5. **Benchmark on logistic regression.** Objective suboptimality as a function of the number of iterations on a logistic regression problem. Similarly to the results for quadratic objectives, we observe that average-case acceleration methods have an overall good performance, with the Uniform acceleration being best in 3 out of 4. This provides some empirical evidence that average-case accelerated methods are applicable on problems beyond their original quadratic setting.

periments. These results should also be contrasted with the fact that we did not adapt the methods to the non-quadratic setting. For example, a first step would be to restart the algorithm to accommodate a varying Hessian, or to perform backtracking line-search to choose the magnitude of the update. We leave these adaptations to be the subject of future work.

8. Conclusion and Future Work

In this work, we first developed an average-case analysis of optimization algorithms, and then used it to develop a family of novel algorithms that are optimal under this average-case analysis. We outline potential future work directions.

Modeling outlier eigenvalues. Often the MP distribution fails to model accurately the outlier eigenvalues that arise in real data (e.g., last row of Figure 4). A potential area for improvement is to consider distributions that can model these outlier eigenvalues.

Non-quadratic and stochastic extension. As seen in Figure 5, the methods are applicable and perform well beyond the quadratic setting in which they were conceived. We currently lack the theory to explain this observation.

Convergence rates and asymptotic behavior. We have only been able to derive the average-case rate for the decaying exponential distribution. It would be interesting to derive average-case convergence rates for more methods.

Regarding the asymptotic behavior, we have noted that some average-case optimal methods like Marchenko-Pastur acceleration converge asymptotically to Polyak momentum. After the first version of this work appeared online, (Scieur & Pedregosa, 2020) showed that this is the case for almost all average-case optimal methods.

Acknowledgements

We would like to thank our colleagues Adrien Taylor for thought-provoking early discussions, Cristóbal Guzmán, Jeffrey Pennington, Francis Bach, Pierre Gaillard and Raphaël Berthier for many pointers and dicussions, and Hossein Mobahi, Nicolas Le Roux, Courtney Paquette, Geoffrey Negiar, Nicolas Loizou, Rémi Le Priol and Reza Babanezhad for fruitful discussion and feedback on the manuscript.

References

Abramowitz, M., Stegun, I. A., et al. Handbook of mathematical functions with formulas, graphs, and mathematical tables, 1972.

Azizian, W., Scieur, D., Mitliagkas, I., Lacoste-Julien, S., and Gidel, G. Accelerating Smooth Games by Manipulating Spectral Shapes. *arXiv preprint arXiv:2001.00602*, 2020.

Bai, Z. and Silverstein, J. W. *Spectral analysis of large dimensional random matrices*. Springer, 2010.

Berthier, R., Bach, F., and Gaillard, P. Accelerated Gossip in Networks of Given Dimension using Jacobi Polynomial Iterations. *arXiv preprint arXiv:1805.08531*, 2018.

Fischer, B. *Polynomial based iteration methods for symmetric linear systems*. Springer, 1996.

Flammarion, N. and Bach, F. From averaging to acceleration, there is only a step-size. In *Conference on Learning Theory*, 2015.

Flanders, D. A. and Shortley, G. Numerical determination of fundamental modes. *Journal of Applied Physics*, 1950.

Gautschi, W. Orthogonal polynomials: applications and computation. *Acta numerica*, 1996.

Ghadimi, E., Feyzmahdavian, H. R., and Johansson, M. Global convergence of the heavy-ball method for convex optimization. In *2015 European Control Conference (ECC)*. IEEE, 2015.

Ghorbani, B., Krishnan, S., and Xiao, Y. An investigation into neural net optimization via hessian eigenvalue density. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

Hestenes, M. R., Stiefel, E., et al. Methods of conjugate gradients for solving linear systems. *Journal of research of the National Bureau of Standards*, 1952.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, 2018.

Jacot, A., Gabriel, F., and Hongler, C. The asymptotic spectrum of the Hessian of DNN throughout training. *arXiv preprint arXiv:1910.02875*, 2019.

Katz, J. and Lindell, Y. *Introduction to modern cryptography*. CRC press, 2014.

Keerthi, S. S. and DeCoste, D. A modified finite Newton method for fast solution of large scale linear SVMs. *Journal of Machine Learning Research*, 2005.

Knuth, D. E. *The art of computer programming*, volume 3. Pearson Education, 1997.

Lacotte, J. and Pilanci, M. Optimal Randomized First-Order Methods for Least-Squares Problems. *arXiv preprint arXiv:2002.09488*, 2020.

Marchenko, V. A. and Pastur, L. A. Distribution of eigenvalues for some sets of random matrices. *Matematicheskii Sbornik*, 1967.

Martin, C. H. and Mahoney, M. W. Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning. *arXiv preprint arXiv:1810.01075*, 2018.

Nemirovski, A. Information-based complexity of convex programming. *Lecture Notes*, 1995.

Nesterov, Y. *Introductory lectures on convex optimization*. Springer, 2004.

Nesterov, Y. Primal-dual subgradient methods for convex problems. *Mathematical programming*, 2009.

Novak, R., Xiao, L., Lee, J., Bahri, Y., Yang, G., Hron, J., Abolafia, D. A., Pennington, J., and Sohl-Dickstein, J. Bayesian deep convolutional networks with many channels are Gaussian processes. *arXiv preprint arXiv:1810.05148*, 2018.

Paquette, C., van Merrinboer, B., and Pedregosa, F. Halting Time is Predictable for Large Models: A Universality Property and Average-case Analysis. *arXiv preprint arXiv:2006.04299*, 2020.

Pennington, J. and Worah, P. Nonlinear random matrix theory for deep learning. In *Advances in Neural Information Processing Systems*, 2017.

Polyak, B. T. and Juditsky, A. B. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 1992.

Sagun, L., Evci, U., Guney, V. U., Dauphin, Y., and Bottou, L. Empirical analysis of the hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454*, 2017.

Scieur, D. and Pedregosa, F. Universal Average-Case Optimality of Polyak Momentum. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

Tao, T. *Topics in random matrix theory*, volume 132. American Mathematical Soc., 2012.