Rapid DP Convex Optimization via Curvature-Aware (Second-Order) Algorithms

Raheem Hussein Ali

Email: rahem.rahem1984@gmail.com

Abstract

Differentially private gradient-based methods, especially (stochastic) gradient descent, serve as the backbone of privacy-preserving machine learning across convex and non-convex frameworks. In standard (non-private) optimization, second-order approaches—such as Newton-type algorithms—typically achieve faster convergence than first-order counterparts. This paper explores how curvature information from the loss function can be leveraged to accelerate differentially private convex optimization. I introduce a privacy-preserving adaptation of the regularized cubic Newton algorithm originally proposed by Nesterov and Polyak (2006), proving that for strongly convex loss functions, our approach guarantees quadratic convergence and reaches the optimal excess risk bound. Furthermore, we design a practical second-order private optimization procedure tailored for unconstrained logistic regression. Both theoretical and empirical evaluations confirm the superior efficiency of our method. Experimental outcomes demonstrate consistent improvements in excess loss compared to leading baselines, achieving a 10–40 × speedup over DP-GD and DP-SGD on complex benchmark datasets.

1. Introduction

A wide range of machine learning problems can be formulated as convex optimization tasks. Formally, let the dataset be denoted as $S_n = (z_1, z_2, \dots, z_n) \in \mathcal{Z}^n$, where each $z_i \in \mathcal{Z}$. Consider a closed and convex set $\mathcal{W} \subseteq \mathbb{R}^d$ and a loss function $f: \mathcal{W} \times \mathcal{Z} \to \mathbb{R}$ such that, for every $z \in \mathcal{Z}$, the mapping $f(w, z)$ is convex in w. The goal of convex learning is to find an approximate minimizer of

$$\underset{w \in \mathcal{W}}{\operatorname{argmin}} \quad \ell(w, S_n) = \frac{1}{n} \sum_{i=1}^{n} f(w, z_i).$$

In this study, we focus on scenarios where the dataset S_n contains sensitive or confidential information.

Differential Privacy and Convex Optimization. Differential Privacy (DP) [?] provides a principled framework to ensure that the output of a computation does not reveal private information about any individual in the dataset. Within this context, the aim is to design an algorithm $\mathcal{A}: \mathcal{Z}^n \to \mathcal{W}$ that is both differentially private and yields a low excess risk, defined as

$$\ell(\mathcal{A}(S_n), S_n) - \min_{w \in \mathcal{W}} \ell(w, S_n).$$

This problem, widely referred to as *private convex optimization*, lies at the intersection of learning theory and privacy-preserving computation.

The Role and Limitation of DP Gradient Descent. The most prevalent approach for private convex optimization is differentially private (stochastic) gradient descent (DP-GD or DP-SGD). These methods operate iteratively: starting from an initial point w_0, they update as

$$w_{t+1} = w_t - \eta(\nabla_w \ell(w_t, S_n) + \xi_t),$$

where $\eta > 0$ is the learning rate and ξ_t is Gaussian noise injected to ensure privacy guarantees. Due to the composition property of DP, the noise magnitude $\| \xi_t \|$ is proportional to $\sqrt{T}$, where T is the total number of iterations. We assume $\| \nabla_w \ell(w_t, S_n) \| \leq 1$ for all t.

A well-known drawback of DP-(S)GD is its slow convergence. The interplay between the step size η and the iteration count T introduces a tradeoff in excess risk: a small product $\eta \cdot T$ prevents the model from converging optimally, while a large $\eta \cdot \sqrt{T}$ increases the noise impact. As a result, practical implementations tend to use small η and large T, leading to very gradual convergence. This limitation has been theoretically confirmed: for β-smooth convex functions, the optimal DP-GD step size is on the order of $\max\{1/\sqrt{n}, \sqrt{d}/(\varepsilon n)\}$ [1], whereas in the non-private case it is $1/\beta$. Consequently, smaller steps require substantially more iterations. Moreover, the efficiency of DP-SGD is hindered by two additional factors: (1) it demands large batch sizes to perform competitively [2], and (2) hyperparameter tuning for DP algorithms remains challenging [3]. These challenges motivate the question: *Can we develop a private optimization method that adapts the step size dynamically based on the local geometry of the loss?*

Inspiration from Second-Order Optimization. In classical (non-private) optimization, the slow convergence of gradient descent has long been mitigated by algorithms that incorporate curvature information. Such *second-order* or *preconditioned* methods, e.g., Newton's algorithm [2, 3], achieve this by minimizing a local quadratic approximation:

$$w_{t+1} = w_t + \Delta_t, \qquad \text{where} \qquad \Delta_t = \operatorname{argmin}_{\Delta}\{\ell(w_t, S_n) + \langle \nabla \ell(w_t, S_n), \Delta\rangle + \frac{1}{2}\langle H_t \Delta, \Delta\rangle\}.$$

Here, H_t denotes a curvature matrix, such as the Hessian $H_t = \nabla^2 \ell(w_t, S_n)$ in Newton's method. These algorithms adaptively adjust the step direction and magnitude along each dimension, resulting in faster convergence rates.

Research Objective and Contributions. This paper aims to accelerate differentially private convex optimization by exploiting second-order information. We investigate several core questions: Can curvature information be leveraged to improve both the convergence rate and excess loss in private convex optimization? What are effective strategies for privatizing second-order quantities, such as the Hessian matrix? And how does the resulting tradeoff between privacy, utility, and runtime compare to first-order approaches like DP-GD?

We demonstrate that second-order information can indeed enhance private optimization, achieving excess loss comparable to or better than that of DP-GD while substantially reducing computation time. Our contributions include both theoretical guarantees and empirical validations supporting these claims [4, 5]

1.1 Provably Optimal Algorithm for Strongly Convex Functions

Newton's method, a well-established second-order optimization framework, is celebrated for its remarkable convergence rate when applied to smooth and strongly convex objectives in the non-private setting. Specifically, to attain an excess loss of magnitude α, the method requires only $\mathcal{O}(\log\log(1/\alpha))$ iterations—significantly faster than any first-order approach known to date.

An immediate question arises: can one construct a differentially private second-order optimization algorithm that maintains this near-optimal rate and achieves the minimax excess error $\mathrm{err}_{\mathrm{opt}}$ within $\mathcal{O}(\log\log(1/\mathrm{err}_{\mathrm{opt}}))$ iterations? In Section 5, we answer this affirmatively by proposing a differentially private variant of the *cubic-regularized Newton method* introduced by Nesterov and Polyak .

At each iteration t, our method constructs and minimizes a cubic upper approximation of the loss around the current iterate w, expressed as:

$$\ell(w + \Delta, S_n) \le \ell(w, S_n) + \langle \nabla_w \ell(w, S_n), \Delta\rangle + \frac{1}{2}\langle \nabla_w^2 \ell(w, S_n)\ \ \Delta, \Delta\rangle + \mathcal{O}(\| \Delta \|^3).$$

This local cubic model enables controlled use of curvature information while preventing overly aggressive updates that could compromise stability or privacy guarantees. By appropriately privatizing both the gradient and Hessian information, the proposed algorithm retains the quadratic convergence properties of Newton's method in expectation and achieves optimal excess risk bounds under differential privacy constraints.

1.2 Noise Injection and Hessian Modification Strategies

Our differentially private second-order method introduces two independent sources of randomness at each iteration: $\xi_{t,1}$, used to privatize the gradient computation, and $\xi_{t,2}$, which ensures the privacy of the direction update. To maintain numerical stability and limit the sensitivity of the Hessian, we employ a transformation function Ψ that regularizes its eigenvalues so that none are excessively small. This step is crucial for bounding the ℓ_2-sensitivity and, consequently, controlling the variance of $\xi_{t,2}$.

We explore two distinct regularization schemes—*eigenvalue clipping* and *eigenvalue shifting*. In the clipping approach, each eigenvalue λ_i of $\nabla^2_{w_t}\ell(w_t, S_n)$ is replaced by $\max\{\lambda_i, \lambda_0\}$, where $\lambda_0 > 0$ is a constant threshold that prevents near-singular curvature matrices. Alternatively, under the shifting approach, we modify the Hessian as

$$\Psi(\nabla^2_{w_t}\ell(w_t, S_n)) = \nabla^2_{w_t}\ell(w_t, S_n) + \lambda_0 I.$$

Both strategies achieve a balance between reducing sensitivity and preserving useful curvature information, as the most informative structural components of the Hessian are typically encoded in its larger eigenvalues and corresponding eigenvectors.

Theoretical guarantees for the local convergence of our update rule are presented in Section 5.3, and an extensive experimental analysis is detailed in Section 6. Empirical findings indicate that our algorithm consistently outperforms existing baselines across multiple benchmark datasets.

1.3 Ensuring Global Convergence

A limitation of the update rule introduced in Equation (1) is that, even in the absence of noise, it does not ensure global convergence. If the initialization w_0 is far from the optimal point, the iterates may diverge. To mitigate this issue, we introduce a variant of Newton's update that

utilizes an alternative second-order approximation providing a *Quadratic Upper Bound (QU)* on the logistic loss. This formulation guarantees global convergence similar to the cubic-regularized Newton method while maintaining comparable empirical speed in the private setting.

Experimental comparisons (see Figure 1) illustrate that, despite DP-GD having a smaller per-iteration cost, our QU-based algorithm is approximately $30\times$ faster in real wall-clock time and achieves a lower excess loss for the logistic regression task on the *Covertype* dataset, under $(\varepsilon, \delta) = (1, n^{-2})$-DP.

1.4 Stochastic Minibatch Extension

We further extend our framework to the stochastic minibatch regime, where both gradient and curvature information are computed over randomly sampled subsets of the data. This minibatch variant naturally integrates with our privacy-preserving mechanism and retains the accelerated convergence characteristics. Experimental results demonstrate that this stochastic extension converges substantially faster than DP-SGD while maintaining comparable or improved utility guarantees.

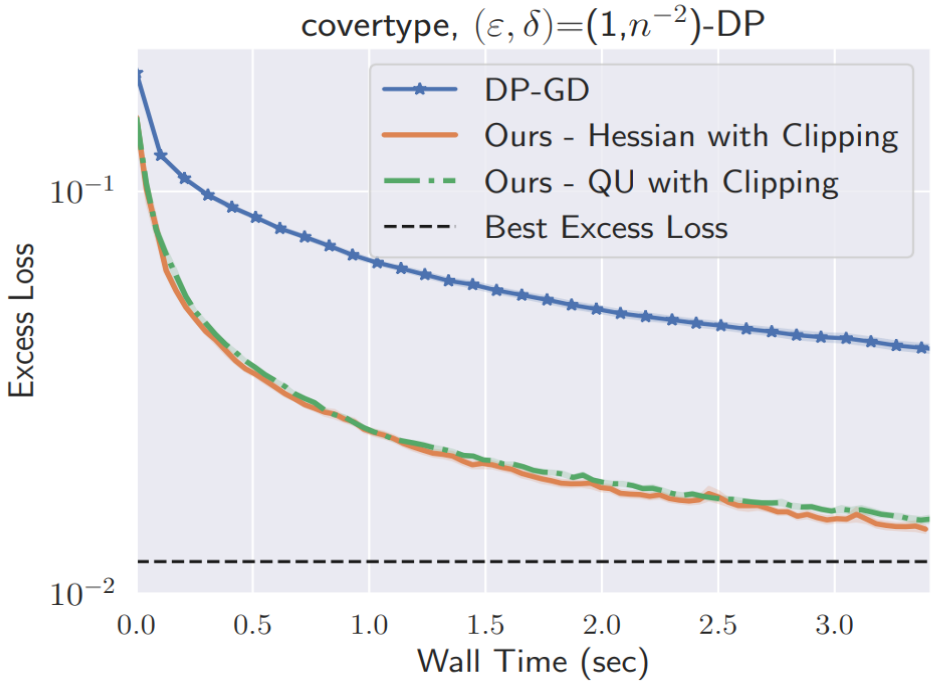

Figure 1: Excess loss versus runtime of DP-GD our algorithms.

2. Related Work

Research on differentially private (DP) optimization is extensive and spans a variety of models and guarantees [11; 12; 13; 14; 15; 16; 17; 18; 19; 20; 21; 22]. Closest to our setting, Avella–Medina, Bradshaw, and Loh [6] investigate the use of second–order information for DP convex optimization. As discussed in Remark 4.5 and Section 6, our approach relaxes several of their more restrictive assumptions and achieves improved excess error for logistic regression.

A rich body of work examines second–order methods in the non–private literature. The algorithmic design typically depends on problem dimensions: when the sample size n is large, various sampling and subsampling strategies have been proposed to efficiently form curvature information [7; 8; 9; 10] When the ambient dimension d is large, a complementary line of research develops scalable approximations to the Hessian (or its inverse/preconditioner) [11; 12; 13; 14; 15]. A related family of techniques estimates curvature from successive gradient differences, leading to quasi–Newton schemes rooted in the classic BFGS methodology 16].

3. Preliminaries

Let $d \in \mathbb{N}$. For a vector $x \in \mathbb{R}^d$, we denote its Euclidean norm by $\| x \|$. For integers $n, m \in \mathbb{N}$ and a matrix $A \in \mathbb{R}^{n\times m}$, the operator norm is defined as $\| A \| = \sup_{x\in\mathbb{R}^m:\|x\|\le 1} \| Ax \|$, and the Frobenius norm is given by $\| A \|_F = \sqrt{\mathrm{trace}(A^\top A)}$, where $\mathrm{trace}(\cdot)$ denotes the matrix trace. The $d \times d$ identity matrix is written as I_d, and $\langle \cdot,\cdot \rangle$ represents the standard inner product on $\mathbb{R}^d$.

For any closed and convex subset $\mathcal{W} \subseteq \mathbb{R}^d$, the Euclidean projection operator $\Pi_{\mathcal{W}}: \mathbb{R}^d \to \mathcal{W}$ is defined by

$$\Pi_{\mathcal{W}}(x) = \operatorname*{argmin}_{y\in\mathcal{W}} \| y - x \|^2.$$

Let $\mathcal{M}_1(\mathbb{R})$ denote the set of all probability measures over $\mathbb{R}$. Throughout, all probabilistic statements are assumed to hold almost surely, and such qualifications are omitted for clarity.

Loss Function Assumptions. Let $\mathcal{Z}$ denote the data domain and $\mathcal{W} \subseteq \mathbb{R}^d$ the parameter space. Consider a loss function $f: \mathcal{W} \times \mathcal{Z} \to \mathbb{R}$ that is continuous in both arguments, convex in w, and where $\mathcal{W}$ is closed and convex. We adopt the following regularity conditions:

1. **L_0-Lipschitz continuity:** There exists $L_0 > 0$ such that for all $z \in \mathcal{Z}$ and $w, v \in \mathcal{W}$,

$$|f(w,z) - f(v,z)| \le L_0 \| w - v \|.$$

2. **L_1-smoothness:** There exists $L_1 > 0$ such that for all $z \in \mathcal{Z}$ and $w, v \in \mathcal{W}$,
$$\| \nabla f(w,z) - \nabla f(v,z) \| \leq L_1 \| w - v \|.$$

3. **L_2-Lipschitz Hessian:** There exists $L_2 > 0$ such that for all $z \in \mathcal{Z}$ and $w, v \in \mathcal{W}$,
$$\| \nabla^2 f(w,z) - \nabla^2 f(v,z) \| \leq L_2 \| w - v \|.$$

4. **μ-strong convexity:** There exists $\mu > 0$ such that for all $z \in \mathcal{Z}$ and $w, v \in \mathcal{W}$,
$$f(v,z) \geq f(w,z) + \langle \nabla f(w,z), v - w \rangle + \frac{\mu}{2} \| v - w \|^2.$$

3.1 Zero-Concentrated Differential Privacy

For our privacy analysis, we employ the *zero-concentrated differential privacy (zCDP)* framework [22, 23], which provides a convenient composition property: the total privacy parameter ρ adds linearly when composing multiple mechanisms.

[zCDP [24], Definition 1.1] A randomized mechanism $\mathcal{A}: \mathcal{Z}^n \to \mathcal{M}_1(\mathbb{R})$ satisfies ρ-zCDP if, for every pair of neighboring datasets $S_n, S_{n'} \in \mathcal{Z}^n$ (differing in one element) and for all $\alpha > 1$,
$$D_\alpha(\mathcal{A}(S_n) \| \mathcal{A}(S_{n'})) \leq \rho\alpha,$$
where $D_\alpha(\cdot\|\cdot)$ denotes the Rényi divergence of order α.

The zCDP parameter ρ can be viewed as approximately ε^2 in standard DP terms. To achieve (ε, δ)-DP, it suffices to set
$$\rho = \frac{\varepsilon^2}{4\log(1/\delta) + 4\varepsilon}$$
as shown in [Lemma 3.5].

[Conversion to (ε, δ)-DP [?], Proposition 1.3] If a mechanism $\mathcal{A}: \mathcal{Z} \to \mathcal{M}_1(\mathbb{R})$ satisfies ρ-zCDP, then for any $\delta > 0$, it is also $(\rho + 2\sqrt{\rho \log(1/\delta)}, \delta)$-DP.

4. Optimal Algorithm for the Class of Strongly Convex Functions

In this section, we introduce a differentially private version of the cubic-regularized Newton method originally proposed by Nesterov and Polyak [25]. To build intuition for our approach, we first revisit the mechanism of DP gradient descent (DP-GD) when applied to the class of L_0-Lipschitz and L_1-smooth convex loss functions.

Let $\{w_t^{\mathrm{GD}}\}_{t\in[T]}$ denote the iterates of DP-GD. By smoothness of the empirical loss ℓ, we can establish a global quadratic upper bound [Theorem 2.1.5], valid for all $w \in \mathcal{W}$ and datasets $S_n \in \mathcal{Z}^n$:

$$\ell(w, S_n) \le q_t(w) := \ell(w_t^{\mathrm{GD}}, S_n) + \langle \nabla\ell(w_t^{\mathrm{GD}}, S_n), w - w_t^{\mathrm{GD}}\rangle + \frac{L_1}{2} \| w - w_t^{\mathrm{GD}} \|^2. \qquad (1)$$

Thus, DP-GD can be viewed as a two-step iterative procedure:

$$\text{(StepI)} \qquad v_{t+1} = \operatorname{argmin}_v q_t(v) = w_t^{\mathrm{GD}} - L_1^{-1}\nabla\ell(w_t^{\mathrm{GD}}, S_n),$$

$$\text{(StepII)} \qquad w_{t+1}^{\mathrm{GD}} = \Pi_{\mathcal{W}}(v_{t+1} + L_1^{-1}\xi_t),$$

where $\xi_t \sim \mathcal{N}(0, \sigma^2 I_d)$ with variance $\sigma^2 = \frac{L_0^2}{2\rho n^2}$, ensuring that each update satisfies ρ-zCDP [Lemma 2.5]. In the unconstrained case where $\mathcal{W} = \mathbb{R}^d$, the projection operator $\Pi_{\mathcal{W}}$ becomes unnecessary. Conceptually, each iteration minimizes the surrogate quadratic function $q_t(w)$ and then projects back to the feasible region.

Cubic Regularization for Second-Order Information. Now consider the broader class of convex loss functions with L_2-Lipschitz continuous Hessians. Nesterov and Polyak [Lemma 1] show that such functions admit a global cubic upper bound: for all $w, w_t \in \mathcal{W}$,

$$\phi_t(w) := \ell(w_t, S_n) + \langle \nabla\ell(w_t, S_n), w - w_t\rangle + \frac{1}{2}\langle \nabla^2\ell(w_t, S_n)(w - w_t), w - w_t\rangle + \frac{L_2}{6} \| w - w_t \|^3, \qquad (2)$$

such that $\ell(w, S_n) \le \phi_t(w)$. Their non-private algorithm updates via $w_{t+1} = \operatorname{argmin}_w \phi_t(w)$, which typically does not admit a closed-form solution—unlike the quadratic case in (1).

Following similar intuition, our proposed differentially private method iteratively minimizes the cubic surrogate $\phi_t(w)$ in a privacy-preserving manner. The meta-algorithm and its private solver are detailed below.

[H]

Meta Algorithm

[1] Training set $S_n \in \mathcal{Z}^n$, total privacy budget ρ-zCDP, initialization $w_0 \in \mathcal{W}$, number of iterations T. $t = 0,1,\dots,T-1$ Query $\ell(w_t, S_n)$, $\nabla\ell(w_t, S_n)$, and $\nabla^2\ell(w_t, S_n)$. Construct $\phi_t(w)$ using Equation (2). $w_{t+1} = \mathrm{DPSolver}(\phi_t(w), \rho/T, w_t)$. w_T.

The DPSolver in Algorithm 1 does not increase the oracle complexity of the overall algorithm, since it operates on the cubic proxy loss $\phi_t(w)$ rather than directly on the original loss $\ell(w, S_n)$.

[H]

DPSolver

[1] Function $\phi: \mathcal{W} \to \mathbb{R}$ defined as

$$\phi(\theta) = \ell + \langle g, \theta - \theta_0 \rangle + \frac{1}{2}\langle H(\theta - \theta_0), \theta - \theta_0 \rangle + \frac{L_2}{6} \| \theta - \theta_0 \|^3,$$

privacy budget $\tilde{\rho}$-zCDP, initialization θ_0. Set

$$N = \frac{2\tilde{\rho}(L_0 + L_1 D + L_2 D^2)^2 n^2}{(L_0 + L_1 D)^2 d}, \qquad \sigma^2 = \frac{N(L_0 + L_1 D)^2}{2\tilde{\rho}}.$$

$i = 0,1, \dots, N-1$ $\eta_i = \frac{2}{\mu(i+2)}$. $\text{grad}_i = g + H(\theta_i - \theta_0) + \frac{L_2}{2} \| \theta_i - \theta_0 \| (\theta_i - \theta_0)$. $\theta_{i+1} = \Pi_{\mathcal{W}}(\theta_i - \eta_i(\text{grad}_i + \mathcal{N}(0, \sigma^2 I_d)))$. $\sum_{i=0}^{N-1} \frac{2i}{N(N+1)} \theta_i$.

Let f be an L_0-Lipschitz, L_1-smooth, L_2-Hessian-Lipschitz, and μ-strongly convex function. Assume $\mathcal{W} \subseteq \mathbb{R}^d$ has finite diameter D, and let $w^* = \text{argmin}_{w \in \mathcal{W}} \ell(w, S_n)$. Then, for every $\rho > 0$, $\beta \in (0,1)$, and sufficiently large n, if the number of iterations in Algorithm 1 is chosen as

$$T = \Theta\left(\sqrt{\frac{L_2}{\mu^{3/4}}} (\ell(w_0, S_n) - \ell(w^*, S_n))^{1/4} + \text{loglog}\frac{1}{\beta}\right),$$

the algorithm achieves the optimal excess risk rate under ρ-zCDP.

5. DP Logistic Regression Using Second-Order Information

The principal drawback of our cubic-regularized Newton method (Algorithm ??) lies in the computational cost of each iteration, which involves solving a nontrivial subproblem despite its low oracle complexity. Additionally, many practical loss functions—such as the logistic loss—are not strongly convex in unconstrained domains. To address both issues, this section develops a more efficient second-order algorithm tailored for unconstrained logistic regression, which serves as a prototypical example of smooth and convex generalized linear models (GLMs). The logistic loss is a widely used convex surrogate for the non-differentiable 0–1 classification loss, and satisfies smoothness and curvature properties that make it suitable for practical optimization algorithms .

5.1 Logistic Loss and Its Properties

Let $d \in \mathbb{N}$, and define the data space $\mathcal{Z} = \mathbb{B}^d(1) \times \{-1,1\}$, where $\mathbb{B}^d(1) = \{x \in \mathbb{R}^d : \| x \| \leq 1\}$ denotes the unit ball. The logistic loss $f_{\mathrm{LL}} : \mathbb{R}^d \times \mathcal{Z} \to \mathbb{R}$ is defined as

$$f_{\mathrm{LL}}(w, (x, y)) = \log(1 + \exp(-y\langle w, x\rangle)). \tag{3}$$

Its gradient and Hessian take the following forms:

$$\nabla_w f_{\mathrm{LL}}(w, (x, y)) = -\frac{yx}{1+\exp(y\langle w,x\rangle)}, \qquad \nabla_w^2 f_{\mathrm{LL}}(w, (x, y)) = xx^\top \frac{\exp(-\langle w,x\rangle/2)+\exp(\langle w,x\rangle/2)}{(\exp(-\langle w,x\rangle/2)+\exp(\langle w,x\rangle/2))^2}. \tag{4}$$

Classical Newton's method [9.5] relies on minimizing a local second-order Taylor expansion. However, in logistic regression, this quadratic approximation may underestimate the true objective, preventing global convergence [26]. To overcome this, we derive a global quadratic upper bound that holds for all (x, y).

For any $v, w, x \in \mathbb{R}^d$ and $y \in \{-1, +1\}$,

$$f_{\mathrm{LL}}(w, (x, y)) \leq f_{\mathrm{LL}}(v, (x, y)) + \langle \nabla f_{\mathrm{LL}}(v, (x, y)), w - v\rangle + \frac{1}{2}\langle H_{\mathrm{qu}}(v, (x, y))(w - v), w - v\rangle,$$

where

$$H_{\mathrm{qu}}(v, (x, y)) = \frac{\tanh(\langle x,v\rangle/2)}{2\langle x,v\rangle} xx^\top \in \mathbb{R}^{d\times d}.$$

Since f_{LL} is $\frac{1}{4}$-smooth, one can also form a simpler quadratic upper bound using [?, Theorem 2.1.5]:

$$f_{\mathrm{LL}}(w, (x, y)) \leq f_{\mathrm{LL}}(v, (x, y)) + \langle \nabla f_{\mathrm{LL}}(v, (x, y)), w - v\rangle + \frac{1}{8} \| w - v \|^2.$$

However, Lemma 5.1 yields a strictly tighter bound because $H_{\mathrm{qu}}(v, (x, y)) \preceq \frac{1}{4} I_d$ (see Appendix B.2).

Both the second-order Taylor approximation $\nabla^2 f_{\mathrm{LL}}$ and the matrix H_{qu} provide quadratic models for the logistic loss. We refer to these collectively as the *second-order information (SOI)*. Note that both $\nabla^2 f_{\mathrm{LL}}(v, (x, y))$ and $H_{\mathrm{qu}}(v, (x, y))$ are positive semidefinite rank-one matrices with largest eigenvalue at most $\frac{1}{4} \| x \|^2 \leq \frac{1}{4}$.

5.2 Algorithm Formulation

Given a dataset $S_n = \{(x_i, y_i)\}_{i=1}^n \subseteq \mathbb{B}^d(1) \times \{-1,1\}$, the empirical logistic loss is

$$\ell_{\mathrm{LL}}(w, S_n) = \frac{1}{n}\sum_{i=1}^{n} f_{\mathrm{LL}}(w, (x_i, y_i)).$$

Our algorithm constructs at each iteration t a quadratic approximation

$$q_t(w) = \ell_{\mathrm{LL}}(w_t, S_n) + \langle \nabla \ell_{\mathrm{LL}}(w_t, S_n), w - w_t \rangle + \frac{1}{2}\langle H(w_t, S_n)(w - w_t), w - w_t \rangle, \qquad (5)$$

where

$$H(w_t, S_n) = \frac{1}{n}\sum_{i=1}^{n} H(w_t, (x_i, y_i)).$$

In the non-private setting, the Newton update is

$$w_{t+1} = w_t - H(w_t, S_n)^{-1} \nabla \ell_{\mathrm{LL}}(w_t, S_n).$$

To introduce differential privacy, we must bound the sensitivity of this update. Because directions corresponding to small eigenvalues of $H(w_t, S_n)$ exhibit greater sensitivity, we modify its spectrum so that all eigenvalues are at least $\lambda_0 > 0$. This guarantees numerical and privacy stability.

[Spectral Regularization Operator] Let $A \in \mathbb{R}^{d\times d}$ be positive semidefinite with eigendecomposition $A = \sum_{i=1}^{d} \lambda_i u_i u_i^{\top}$. For $\lambda_0 > 0$, define:

$$\Psi_{\lambda_0}(A, \mathrm{clip}) = \sum_{i=1}^{d} \max\{\lambda_0, \lambda_i\} u_i u_i^{\top}, \qquad \Psi_{\lambda_0}(A, \mathrm{add}) = A + \lambda_0 I_d.$$

DP Newton Method with Double Noise

[1] Training set S_n, $\lambda_0 > 0$, $\theta \in (0,1)$, privacy budget ρ-zCDP, initialization w_0, iteration count T, SOI modification $\in \{\mathrm{clip}, \mathrm{add}\}$. Set $\sigma_1 = \sqrt{\frac{T}{n}}\sqrt{2\rho(1-\theta)}$. SOI modification = add $\sigma_2 = \sqrt{T(4n\lambda_0^2 + \lambda_0)}\sqrt{2\rho\theta}$. $\sigma_2 = \sqrt{T(4n\lambda_0^2 - \lambda_0)}\sqrt{2\rho\theta}$. $t = 0,1,\dots,T-1$ Compute $\nabla f_{\mathrm{LL}}(w_t, S_n)$ and $H(w_t, S_n)$. $\tilde{H}_t = \Psi_{\lambda_0}(H(w_t, S_n), \mathrm{SOImodification})$. $\tilde{g}_t = \nabla f_{\mathrm{LL}}(w_t, S_n) + \mathcal{N}(0, \sigma_1^2 I_d)$. $w_{t+1} = w_t - \tilde{H}_t^{-1}\tilde{g}_t + \mathcal{N}(0, \| \tilde{g}_t \|^2 \sigma_2^2 I_d)$. w_T.

If Algorithm 3 uses the *additive* SOI modification, then for any $S_n \in (\mathbb{R}^d \times \{-1,1\})^n$, $w_0 \in \mathcal{W}$, $\lambda_0 > 0$, $T \in \mathbb{N}$, $\rho > 0$, and $\theta \in (0,1)$, the final iterate w_T satisfies ρ-zCDP when $\sigma_1 = \sqrt{\frac{T}{n}}\sqrt{2\rho(1-\theta)}$ and $\sigma_2 = \sqrt{T(4n\lambda_0^2 + \lambda_0)}\sqrt{2\rho\theta}$.

If Algorithm 3 employs the *clipping* SOI modification and $n > \frac{1}{4\lambda_0}$, then under the same setup, w_T satisfies ρ -zCDP provided that $\sigma_1 = \sqrt{\frac{T}{n}}\sqrt{2\rho(1-\theta)}$ and $\sigma_2 = \sqrt{T(4n\lambda_0^2 - \lambda_0)}\sqrt{2\rho\theta}$.

Our DP Newton algorithm differs from the standard (non-private) Newton method in three respects: (1) the gradient is privatized via Gaussian noise; (2) the Hessian is spectrally regularized to prevent excessively small eigenvalues; and (3) an additional noise term is injected into the update based on the privatized gradient and adjusted curvature. This yields four variants of the algorithm—*Hess-clip*, *Hess-add*, *QU-clip*, and *QU-add*—depending on whether the SOI source is the true Hessian or the quadratic upper-bound matrix, and which regularization strategy is used.

[Generalization of Algorithm 3] Although Algorithm 3 is presented for logistic regression, the same privacy guarantees extend to any convex, twice-differentiable, Lipschitz, and smooth loss function without assumptions on Hessian rank. The key technical component lies in bounding the approximate Lipschitz continuity of Ψ in the operator norm (see Lemma B.7). Consequently, our approach generalizes both *objective perturbation* and the *private damped Newton method* , which impose more restrictive low-rank assumptions on the curvature matrix.

5.3 Private and Adaptive Selection of the Minimum Eigenvalue

An important hyperparameter in Algorithm 3 is the lower bound on the eigenvalues, denoted by λ_0. The choice of this parameter introduces a trade-off between preserving curvature information and controlling noise magnitude. A smaller λ_0 keeps the second-order information (SOI) closer to the true Hessian, thus improving approximation fidelity. However, reducing λ_0 increases the variance term σ_2 in the Gaussian mechanism, which results in greater noise injection and potentially degrades convergence stability.

To address this challenge, we propose a *private*, *adaptive*, *and time-varying* strategy for selecting $\lambda_{0,t}$ at each iteration. The goal is to dynamically adjust $\lambda_{0,t}$ to minimize the expected loss of the next iterate under the quadratic approximation of Equation (5). Formally, we define:

$$\lambda_{0,t} = \operatorname{argmin}_{\lambda} \mathbb{E}[q_t(w_t - \Psi_\lambda(H(w_t, S_n), \text{SOImodification})\ \ \tilde{g}_t + \| \tilde{g}_t \| \ \ \sigma_2(\lambda)\ \ \xi)],$$

where q_t is the local quadratic approximation given in Equation (5), Ψ_λ is the spectral modification operator from Definition 5.2, $\tilde{g}_t$ denotes the privatized gradient, and $\xi \sim \mathcal{N}(0, I_d)$ is standard Gaussian noise.

As shown in Appendix B.5, an approximate minimizer of the above expectation satisfies the proportional relationship:

$$\lambda_{0,t} \propto \cdots$$

which yields a principled rule for adaptively tuning the spectral regularization parameter while maintaining differential privacy guarantees. This adaptive mechanism enables Algorithm 3 to balance curvature preservation and noise control more effectively across iterations.

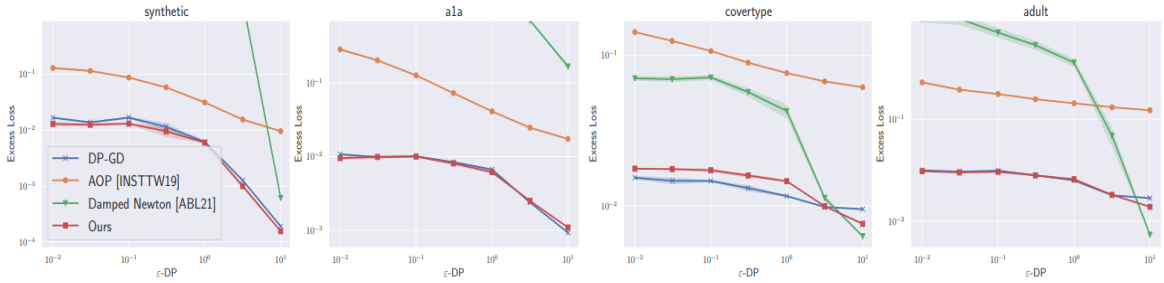

Figure 2: Privacy-Utility tradeoff on different datasets.

5.4 Convergence Analysis Under Privacy Constraints

Let ρ denote the total privacy budget in the zero-concentrated differential privacy (zCDP) framework, and suppose this budget is distributed uniformly across iterations. Under these conditions, the expected squared error of the proposed DP Newton method satisfies:

$$\mathbb{E}_t[\| w_{t+1} - w^* \|_V^2] \leq \nu_{1,t}^2 \| w_t - w^* \|_V^2 + 2\nu_{1,t}\nu_{2,t} \| w_t - w^* \|_V^3 + \nu_{2,t}^2 \| w_t - w^* \|_V^4 + \Delta,$$

where the coefficients are given by

$$\nu_{1,t} = 1 - \frac{\tilde{\lambda}_{\min,t}}{\lambda_0} + \sqrt{\text{rank}}\ (4n\lambda_0^2 - \lambda_0)^{1/2}(2\rho\theta)^{1/2}, \qquad \nu_{2,t} = \frac{0.05}{\tilde{\lambda}_{\min,t}}, \qquad \Delta = \mathcal{O}\left(\frac{\text{rank}\ \rho(1-\theta)n^2}{(\tilde{\lambda}_{\min,t})^2}\right). \tag{6}$$

Here, $\tilde{\lambda}_{\min,t}$ represents the effective smallest eigenvalue after applying the SOI modification:

$$\tilde{\lambda}_{\min,t} = \begin{cases} \min\{\lambda_{\min,t}, \lambda_0\}, & \text{forHess} - \text{clip}, \\ \lambda_{\min,t} + \lambda_0, & \text{forHess} - \text{add}, \end{cases}$$

where $\lambda_{\min,t}$ denotes the smallest non-zero eigenvalue of $\nabla^2 \ell_{\text{LL}}(w_t, S_n)$.

This behavior, combining both linear and quadratic convergence terms, is often referred to as *composite convergence*. Such convergence characteristics have been reported in several analyses

of quasi-Newton and second-order methods (e.g).

The value $\lambda_{\min,t}$ is the smallest non-zero eigenvalue of the Hessian $\nabla^2 \ell_{\mathrm{LL}}(w_t, S_n)$. For sufficiently large n, we have $0 < \nu_{1,t} < 1$, indicating that Algorithm 3, when using the Hessian as its SOI, acts as a local descent method in expectation whenever $\| w_t - w^* \|_V$ is sufficiently larger than the perturbation radius Δ. Intuitively, Theorem 5.8 ensures that the method converges linearly to a neighborhood of the optimal solution, where the radius of this neighborhood is determined by Δ. This phenomenon is empirically verified in Figure 3. Furthermore, the privacy-induced error term Δ in Equation (6) scales with the rank of the data matrix—typically much smaller than d—which contributes to favorable empirical performance. This advantage arises because convergence is measured in the weighted norm $\|\cdot\|_V$.

The coefficients in Equation (6) depend on the current iteration index t, which is theoretically inconvenient. However, in Lemma B.11, we prove the stability relation

$$|\lambda_{\min,t} - \lambda^*_{\min}| \leq 0.1 \quad \| w_t - w^* \|_V,$$

where $\lambda^*_{\min}$ is the smallest non-zero eigenvalue of $\nabla^2 \ell_{\mathrm{LL}}(w^*, S_n)$. Hence, the convergence coefficients can be accurately approximated by their asymptotic values evaluated at the optimum.

5.5 Global Convergence Guarantee for QU-clip and QU-add

We additionally establish global convergence results for the *QU-clip* and *QU-add* variants of our proposed method. In summary, under the standard assumption of *local strong convexity* around the optimum , both QU-clip and QU-add are proven to achieve global convergence. This outcome is intuitive, as these two variants minimize a globally valid quadratic upper bound on the loss function, which ensures descent across the entire domain rather than only within a local neighborhood.

6. Numerical Results

In this section, we empirically assess the effectiveness of our algorithm (Algorithm 3 enhanced with the adaptive minimum eigenvalue selection strategy described in Section 5.2) on binary classification tasks employing logistic regression.

6.1 Experimental Setup

We compare our proposed method against established baselines in differentially private convex optimization. The main experimental configuration is summarized below.

Baseline 1 – DP-(S)GD: This baseline follows the update rule

$$w_{t+1} = w_t - \eta \nabla \ell(w_t, S_n) + \xi,$$

where ξ is Gaussian noise added for differential privacy . Since the logistic loss is 1-Lipschitz, gradient clipping is unnecessary. The Lipschitz constant determines the noise variance in the Gaussian mechanism. To ensure a fair comparison and isolate the benefit of incorporating second-order information, the step size η is selected as the reciprocal of the smoothness parameter.

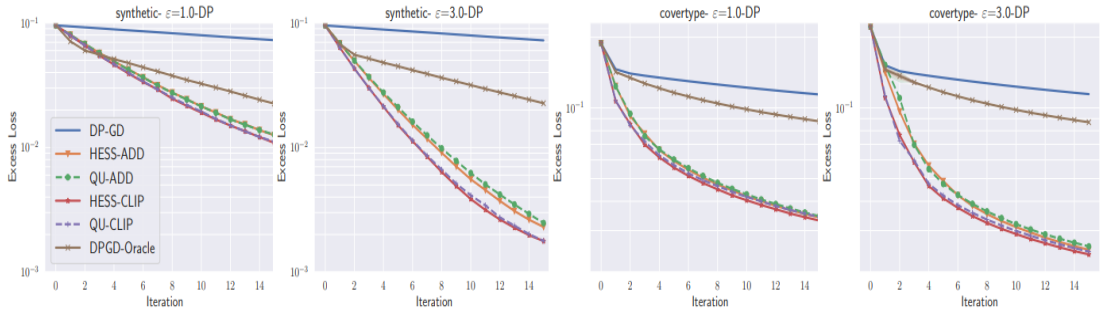

Figure 3: Comparison with DP-GD Oracle where at each iteration the step size tuned non-privately.

6.2 Experimental Baselines and Comparative Evaluation

We benchmark our proposed algorithm against three major baselines in differentially private convex optimization. The experiments assess both accuracy and computational efficiency, focusing on logistic regression tasks with varying privacy budgets $\varepsilon \in \{0.01, 0.1, 1, 10\}$.

Baseline 2 – Approximate Objective Perturbation (AOP): AOP extends the classical *objective perturbation* approach [?, ?], which achieves privacy by (1) injecting a random linear perturbation into the objective function and (2) releasing the minimizer of the perturbed

objective. While the exact minimum ensures differential privacy, computing it can be costly. AOP relaxes this by allowing approximate minimization [?]. Notably, AOP is not an iterative optimization algorithm, and thus, it serves primarily as a non-iterative reference point in our comparison.

Baseline 3 – Damped Newton Method : This algorithm assumes a rank-one Hessian, which holds for logistic regression, and injects Gaussian noise independently into both the gradient and Hessian. Each update takes the form

$$w_{t+1} = w_t - \eta_t H_{\text{noisy},t}^{-1}(w_t, S_n)\tilde{g}_t,$$

where $H_{\text{noisy},t}(w_t, S_n) = \nabla^2 \ell_{\text{LL}}(w_t, S_n) + \Xi_t$ and $\tilde{g}_t = \nabla \ell_{\text{LL}}(w_t, S_n) + \xi_t$. Both Ξ_t and ξ_t are Gaussian noise terms chosen to satisfy privacy. While setting $\eta_t = 1$ prevents convergence, adopting the adaptive rule suggested in [?, Page 22],

$$\eta_t = \frac{\log(1+\beta_t)}{\beta_t}, \qquad \text{where} \beta_t = \| \nabla^2 \ell_{\text{LL}}(w_t, S_n)^{-1} \nabla \ell_{\text{LL}}(w_t, S_n) \|,$$

improves empirical behavior though it no longer ensures privacy. Despite this limitation, the method provides a meaningful performance baseline for comparison.

Datasets: We evaluate on six public datasets: *a1a*, *Adult*, *Covertype*, *Synthetic*, *Fashion-MNIST*, and *Protein*. (Results for Fashion-MNIST and Protein are detailed in Appendix C.) The synthetic dataset is generated as follows: for a fixed $d \in \mathbb{N}$ and $w^* \in \mathbb{R}^d$, feature vectors $\{x_i\}_{i=1}^n$ are sampled uniformly from the unit sphere in $\mathbb{R}^d$, and labels $y_i \in \{-1, +1\}$ are drawn according to

$$\Pr(y_i = +1) = \frac{1}{1+\exp(-\langle x_i, w^* \rangle)}, \qquad \Pr(y_i = -1) = 1 - \Pr(y_i = +1).$$

The privacy level for all experiments is set to $(\varepsilon, \delta = n^{-2})$-DP.

6.3 Privacy–Utility–Runtime Trade-off

We examine how our proposed method balances accuracy, privacy, and computational cost compared to the baselines. For each algorithm, we non-privately tune the total number of iterations and report the lowest achievable excess loss for various ε values in Figure 2. As shown, our algorithm consistently attains the smallest excess loss across a broad range of ε. The private damped Newton method performs competitively only for large ε, while DP-GD and our algorithm yield the strongest overall privacy–utility balance.

Table 1: Runtime comparison between DP-GD and our proposed method. Values denote the ratio of total runtime ($T^*_{\mathrm{DP-GD}}/T^*_{\mathrm{ours}}$) for equivalent accuracy. The final columns indicate the minimum and maximum runtime of our method in seconds.

Dataset	$\varepsilon = 0.01$	$\varepsilon = 0.1$	$\varepsilon = 1$	$\varepsilon = 10$	$\min(T^*_{\mathrm{ours}})$ (s)	$\max(T^*_{\mathrm{ours}})$ (s)
a1a	4.87×	2.95×	5.09×	30.59×	2.45	4.20
synthetic	2.90×	2.90×	5.19×	11.61×	0.18	0.21
adult	12.08×	11.84×	22.17×	38.16×	6.81	8.07
covertype	24.19×	19.85×	35.70×	36.20×	2.93	3.58

Table 1 demonstrates that our algorithm achieves a 10–40 × speedup compared to DP-GD on several challenging datasets. All experiments were conducted on CPU hardware. Importantly, each iteration of Algorithm 3—comprising gradient and second-order information (SOI) computations—is highly parallelizable, suggesting that optimized implementations can yield even greater efficiency. Unless otherwise noted, all reported results correspond to the *Hess-clip* configuration of our method.

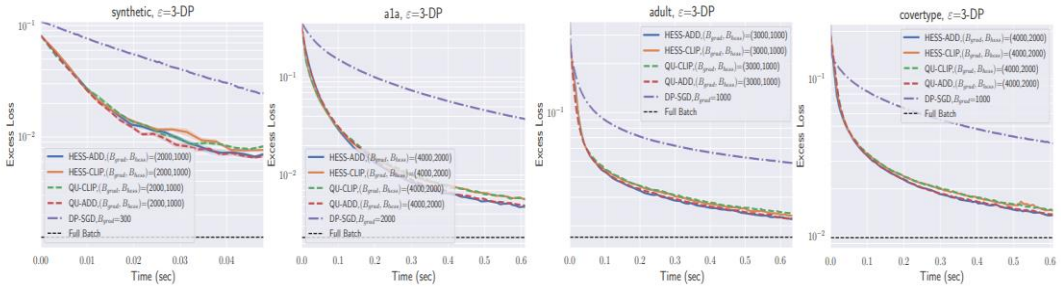

Figure 4: Minibatch Variant of Our Algorithm and Comparison with DP-SGD

6.4 Second-Order Information vs. Optimal Step Size

In non-private convex optimization, the superior performance of second-order algorithms

primarily stems from the fact that second-order information acts as a natural *preconditioner*, reshaping the optimization landscape to enable faster and more stable convergence. This advantage cannot be replicated merely by optimally tuning the step size of a first-order method such as gradient descent (GD). To examine whether this principle extends to our differentially private (DP) framework, we introduce a variant of DP-GD that leverages an optimally chosen step size at each iteration.

Let $\tilde{g}_t$ denote the privatized gradient obtained by adding Gaussian noise to $\nabla \ell_{\mathrm{LL}}(w_t, S_n)$. Instead of using a constant learning rate, we dynamically select the step size η_t according to:

$$\eta_t = \underset{\eta \geq 0}{\operatorname{argmin}} \quad \ell_{\mathrm{LL}}(w_t - \eta \tilde{g}_t, S_n).$$

Note that this method, which we term **DP-GD-Oracle**, is not differentially private, since the adaptive step size depends directly on the unperturbed loss. Nonetheless, it provides a valuable benchmark for assessing whether a single scalar quantity (the step size) can replicate the benefit of full second-order curvature information encoded in a $d \times d$ matrix.

Figure 3 compares the convergence rates of our proposed algorithms with DP-GD-Oracle under both low- and high-privacy regimes. The results clearly indicate that our methods achieve faster convergence than DP-GD-Oracle, even though the latter is non-private. As the privacy budget ε increases, the performance gap widens further, confirming that access to richer curvature information enables more efficient optimization when the added noise becomes smaller.

6.5 Minibatch Variant and Comparison with DP-SGD

Up to this point, our analysis has focused on *full-batch* algorithms, where both the gradient and second-order information (SOI) are computed using the entire dataset. We now extend Algorithm 3 to the *minibatch* regime, in which each iteration relies on a random subsample of the training data. The complete description of this minibatch extension, along with its privacy analysis, is provided in Appendix C.1.

We compare the minibatch version of our algorithm with DP-SGD. Although DP-SGD is typically faster per iteration than DP-GD, achieving strong privacy–utility tradeoffs requires substantially larger batch sizes [, Fig.~2]. This contrasts sharply with the non-private setting, where increasing the batch size offers diminishing returns . In our experiments, we select the DP-SGD batch size to be approximately 2% of the total dataset so that its achievable excess loss

is comparable to that of the full-batch methods. We tune the number of DP-SGD iterations to obtain its best possible outcome.

Figure 4 illustrates the excess loss as a function of wall-clock time. For a fixed runtime, DP-SGD performs more iterations than our second-order methods, but our approaches consistently reach the same (or lower) excess loss with approximately $8-10\times$ less total computation time across all datasets. Variants of our algorithm employing the *additive* SOI modification outperform those based on the *clipping* strategy in the minibatch setting, which can be attributed to the smaller variance parameter σ_2 in the additive formulation (see Algorithm 3). Overall, the privacy–utility–runtime trade-off of the minibatch variant closely mirrors that of its full-batch counterpart.

7. Conclusion and Limitations

We have demonstrated that second-order optimization techniques can be effectively adapted to the differentially private (DP) setting, achieving both improved theoretical convergence rates and superior empirical performance. Our findings suggest several promising directions for future research. A key limitation of our current framework is the computational burden associated with forming and inverting the Hessian when the dimension d is large. In non-private optimization, a growing body of work seeks to mitigate this issue through approximate curvature estimation techniques that preserve essential second-order information while maintaining efficiency . Extending our private second-order framework to incorporate such approximations represents an exciting avenue for further exploration.

Refrencess

[1] N. Agarwal, B. Bullins, and E. Hazan. “Second-order stochastic optimization for machine learning in linear time”. The Journal of Machine Learning Research 18.1 (2017), pp. 4148–4187.

[2] M. Avella-Medina, C. Bradshaw, and P.-L. Loh. “Differentially private inference via noisy optimization”. arXiv preprint arXiv:2103.11003 (2021).

[3] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, et al. “Deep learning with differential privacy”. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. 2016, pp. 308–318.

[4] J. Arbel, O. Marchal, and H. D. Nguyen. “On strict sub-Gaussianity, optimal proxy variance and symmetry for bounded random variables”. ESAIM: Probability and Statistics 24 (2020), pp. 39–55.

[5] F. Bach. “Self-concordant analysis for logistic regression”. Electronic Journal of Statistics 4 (2010), pp. 384–414.

[6] F. Bach. “Adaptivity of averaged stochastic gradient descent to local strong convexity for logistic regression”. The Journal of Machine Learning Research 15.1 (2014), pp. 595–627.

[7] J. A. Blackard and D. J. Dean. “Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables”. Computers and electronics in agriculture 24.3 (1999), pp. 131–151.

[8] R. Bassily, V. Feldman, K. Talwar, and A. Guha Thakurta. “Private stochastic convex optimization with optimal rates”. Advances in neural information processing systems 32 (2019).

[9] M. Bun and T. Steinke. “Concentrated differential privacy: Simplifications, extensions, and lower bounds”. In: Theory of Cryptography Conference. Springer. 2016, pp. 635–658.

[10] R. Bassily, A. Smith, and A. Thakurta. “Private empirical risk minimization: Effi cient algorithms and tight error bounds”. In: 2014 IEEE 55th annual symposium on foundations of computer science. IEEE. 2014, pp. 464–473.

[11] S. P. Boyd and L. Vandenberghe. Convex optimization. Cambridge university press, 2004.

[12] R. Caruana, T. Joachims, and L. Backstrom. “KDD-Cup 2004: results and analysis”. ACM SIGKDD Explorations Newsletter 6.2 (2004), pp. 95–108.

[13] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate. “Differentially Private Empirical Risk Minimization”. Journal of Machine Learning Research 12.29 (2011), pp. 1069–1109.

[14] D. Dua and C. Graff. UCI Machine Learning Repository. 2017.

[15] C. Dwork, F. McSherry, K. Nissim, and A. Smith. “Calibrating noise to sensitivity in private data analysis”. In: Theory of cryptography conference. Springer. 2006, pp. 265–284.

[16] C. Dwork and G. N. Rothblum. “Concentrated differential privacy”. arXiv preprint arXiv:1603.01887 (2016).

[17] M. A. Erdogdu and A. Montanari. “Convergence rates of sub-sampled Newton methods”. Advances in Neural Information Processing Systems 28 (2015).

[18] M. A. Erdogdu. “Newton-Stein method: A second order method for GLMs via Stein’s lemma”. Advances in Neural Information Processing Systems 28 (2015).

[19] R. Gower, D. Kovalev, F. Lieder, and P. Richtárik. “RSN: randomized subspace Newton”. Advances in Neural Information Processing Systems 32 (2019).

[20] S. Ghadimi and G. Lan. “Optimal Stochastic Approximation Algorithms for Strongly Convex Stochastic Composite Optimization I: A Generic Algorithmic Framework”. SIAM Journal on Optimization 22.4 (2012), pp. 1469–1492. eprint: https://doi. org/10.1137/110848864.

[21] S. Gopi, Y. T. Lee, and D. Liu. “Private Convex Optimization via Exponential Mechanism”. In: Proceedings of Thirty Fifth Conference on Learning Theory. Ed. by P.-L. Loh and M. Raginsky. Vol. 178. Proceedings of Machine Learning Research. PMLR, Feb. 2022, pp. 1948–1989.

[22] A. Ganesh, A. Thakurta, and J. Upadhyay. “Langevin diffusion: An almost uni versal algorithm for private Euclidean (convex) optimization”. arXiv preprint arXiv:2204.01585 (2022).

[23] G. H. Golub and C. F. Van Loan. Matrix computations. JHU press, 2013.

[24] N. J. Harvey, C. Liaw, and S. Randhawa. “Simple and optimal high probability bounds for strongly-convex stochastic gradient descent”. arXiv preprint arXiv:1909.00843 (2019).

[25] R. Iyengar, J. P. Near, D. Song, O. Thakkar, et al. Differentially Private Convex Optimization Benchmark. URL: https : / / github . com / sunblaze - ucb / dpml - benchmark.

[26] R. Iyengar, J. P. Near, D. Song, O. Thakkar, et al. “Towards practical differentially private convex optimization”. In: 2019 IEEE Symposium on Security and Privacy (SP). 2019.

[27] Q. Jin and A. Mokhtari. “Non-asymptotic superlinear convergence of standard quasi Newton methods”. Mathematical Programming 200.1 (2023), pp. 425–473.

[28] C. Jin, P. Netrapalli, R. Ge, S. M. Kakade, and M. I. Jordan. “A short note on concentration inequalities for random vectors with subgaussian norm”. arXiv preprint arXiv:1902.03736 (2019).

[29] F. Jarre and P. L. Toint. “Simple examples for the failure of Newton’s method with line search for strictly convex minimization”. Mathematical Programming 158.1 (2016), pp. 23–34.