Cubic Regularized Quasi-Newton Methods

Dmitry Kamzolov KAMZOLOV.OPT@GMAIL.COM
Klea Ziu ZIUKLEA29@GMAIL.COM
Artem Agafonov AGAFONOV.AD@PHYSTECH.EDU
Martin Takáč TAKAC.MT@GMAIL.COM
Mohamed bin Zayed University of Artificial Intelligence

Abstract

In this paper, we propose a Cubic Regularized L-BFGS. Cubic Regularized Newton outperform classical Newton method in terms of global performance. In classics, L-BFGS approximation is applied for Newton method. We propose a new variant of inexact Cubic Regularized Newton. Then, we use L-BFGS approximation as an inexact Hessian for Cubic Regularized Newton. It allows us to get better theoretical convergence rates and good practical performance, especially from the points where classical Newton is diverging.

1. Introduction

In this paper, we focus on optimization methods that utilize second-order (curvature) information of objective function. Usually, these methods achieve faster convergence than first-order algorithms. But at the same time, the per-iteration cost of second-order methods is significantly higher. For example, a classical Newton method has a quadratic local convergence, but each iteration requires matrix inversion, which is impractical for large-scale optimization problems. Quasi-Newton methods [4, 5, 7, 10, 12, 13, 19, 20] were proposed to reduce the high iteration costs of the Newton method. These methods construct Hessian (inverse) approximations based on first-order (gradient) information or second-order information along random directions (Hessian vector products) [3].

The Cubic regularized Newton method [17] is another approach to using curvature information in optimization algorithms. This algorithm achieves a global convergence and allows for Nesterov acceleration [18]. However, the main drawback of this scheme is an auxiliary subproblem on each iteration. Thus, usually, it is required to run a separate optimization algorithm to solve the subproblem. Cubic Newton algorithm allows for inexact Hessian approximations [11], which makes it applicable to distributed optimization [2, 8, 21]. Moreover, all the results listed above about the Cubic Newton method are also generalizable to higher-order (tensor) methods [1, 9, 15, 16]. In the paper, we propose a Cubic Regularized L-BFGS that uses L-BFGS approximation as an inexact Hessian for Cubic Regularized Newton. It allows us to get better theoretical convergence rates and good practical performance, especially from the points where classical Newton is diverging. Sampled and Greedy L-BFGS approximation theoretically outperform gradient descent. Also, under some special conditions on memory size, we can expect that Cubic L-BFGS will converge with the same rate as Cubic Regularized Newton.

2. Problem Statement and Preliminaries

In this paper, we consider the following optimization problem

$$\min_{x \in \mathbf{E}} f(x), \tag{1}$$

where $\mathbf{E}$ is a finite-dimensional vector space.

Let us introduce some classes of functions $f(x)$ that we are focused on. First, we define *star-convex* function and *μ-strongly star-convex* function. Note, that these functions are non-convex in general.

Definition 1 *Let x^* be a minimizer of the function f. The function f is **star-convex** with respect to x^* if for all $x \in \boldsymbol{E}$*

$$f\left(\alpha x + (1-\alpha)x^*\right) \le \alpha f(x) + (1-\alpha) f(x^*), \quad \forall \alpha \in [0,1]. \tag{2}$$

Definition 2 *Let x^* be a minimizer of the function f. The function f is **μ-strongly star-convex** with respect to x^* if for all $x \in \boldsymbol{E}$*

$$f\left(\alpha x + (1-\alpha)x^*\right) \le \alpha f(x) + (1-\alpha) f(x^*) - \frac{\alpha(1-\alpha)\mu}{2}\|x - x^*\|^2, \quad \forall \alpha \in [0,1]. \tag{3}$$

Note, that convex functions are a subclass of star-convex functions, and mu-strongly convex functions are a subclass of mu-strongly star-convex functions. Also, star-convex function class is much bigger than convex functions. For example, all rational p-norms for vectors are star-convex but not convex, for example $\|x\|_{1/2}$. Also, there are some papers that suggest some evidence that some neural networks are star-convex in large neighbourhoods of its minimizers [14]. To finalise, we introduce smoothness assumptions for the function $f(x)$.

Definition 3 *The continuously-differentiable function $f(x)$ has L_1-Lipschitz-continuous gradient if for any $x, y \in \boldsymbol{E}$*

$$\|\nabla f(x) - \nabla f(y)\|_* \le L_1 \|x - y\|. \tag{4}$$

Definition 4 *The twicly continuously-differentiable function $f(x)$ has L_2-Lipschitz-continuous Hessian if for any $x, y \in \boldsymbol{E}$*

$$\|\nabla^2 f(x) - \nabla^2 f(y)\| \le L_2 \|x - y\|. \tag{5}$$

Note, these assumptions are the most standard assumptions for the first and second order methods.

3. Inexact Cubic Regularized Newton

In this section, we introduce an Inexact Cubic Regularized Newton (ICN). This method is a main upper-level method of our approach. It guarantees fast convergence and control on inexactness of inner information. This section is mostly inspired by the paper [11] (Section 2) and its generalization from the paper [1] (Section 3,4). For this section, we assume that the function $f(x)$ has L_2-Lipschitz-continuous Hessian.

Let us introduce a generalized definition of inexact Hessian.

Definition 5 *A self-adjoint operator $B_x : \boldsymbol{E} \to \boldsymbol{E}^*$ is an $(\delta_{up}, \delta_{low})$-inexact Hessian for the function $f(x)$ at the point $x \in \boldsymbol{E}$ if*

$$\nabla^2 f(x) \preceq B_x + \delta_{up} D \tag{6}$$

$$B_x \preceq \nabla^2 f(x) + \delta_{low} D \tag{7}$$

Note, that in [11] the authors used only $(\delta_{up}, \delta_{low})$-inexact Hessian with $\delta_{up} = 0$, and in [1] the authors used only $(\delta_{up}, \delta_{low})$-inexact Hessian with $\delta_{low} = \delta_{up} = \delta$.

Now, we can move to the formulation of ICN . Firstly, we introduce *exact Taylor approximation*.

$$\Phi_x(y) \stackrel{\text{def}}{=} f(x) + \langle \nabla f(x), y - x \rangle + \tfrac{1}{2} \left\langle \nabla^2 f(x)(y - x), y - x \right\rangle, \tag{8}$$

and *inexact Taylor approximation*

$$\phi_x(y) \stackrel{\text{def}}{=} f(x) + \langle \nabla f(x), y - x \rangle + \tfrac{1}{2} \langle B_x(y - x), y - x \rangle, \tag{9}$$

Secondly, let us show that regularized inexact Taylor approximation with $(\delta_{up}, \delta_{low})$-inexact Hessian is close to the function $f(x)$ by finding upper and lower bounds.

Lemma 6 *For the function $f(x)$ with L_2-Lipschitz-continuous Hessian and $(\delta_{up}, \delta_{low})$-inexact Hessian B_x, for any $x, y \in \boldsymbol{E}$ we have*

$$f(y) - \phi_x(y) \leq \frac{L_2}{6} \|y - x\|^3 + \frac{\delta_{up}}{2} \tag{10}$$

$$\phi_x(y) \leq f(y) + \frac{L_2}{6} \|y - x\|^3 + \frac{\delta_{low}}{2} \tag{11}$$

Finally, we introduce the ICN operator

$$S_{M,\delta_{up}}(x) = x + \operatorname*{argmin}_{h \in \mathbf{E}} \left\{ f(x) + \langle \nabla f(x), h \rangle + \frac{1}{2} \langle B_x h, h \rangle + \frac{M}{6} \|h\|^3 + \frac{\delta_{up}}{2} \|h\|^2 \right\}, \tag{12}$$

where $M \geq L_2$. Then step of the method is

$$x_{k+1} = S_{M,\delta_{up}}(x_k). \tag{13}$$

Now, we present the convergence theorem of ICN for star-convex and μ-strognly star-convex functions.

Theorem 7 *Let $f(x)$ be a star-convex function (Option A) or μ-strongly star-convex function (Option B) with respect to global minimizer x^*, $f(x)$ has L_2-Lipschitz-continuous Hessian, B_{x_k} is a $(\delta_{up}, \delta_{low})$-inexact Hessian, and $M \geq L_2$, then the total number of iteration $T \geq 1$ of the Inexact Cubic Regularized Newton to find ε-solution x_T such that $f(x_T) - f(x^*) \leq \varepsilon$ is bounded by*

$$\textit{Option A} \qquad T = O(1) \max \left\{ \frac{(\delta_{up} + \delta_{low}) R^2}{\varepsilon}; \sqrt{\frac{M R^3}{\varepsilon}} \right\}, \tag{14}$$

$$\textit{Option B} \qquad T = O(1) \max \left\{ 1; \frac{\delta_{up} + \delta_{low}}{\mu}; \sqrt{\frac{M R}{\mu}} \right\} \log \left(\frac{f(x_0) - f(x^*}{\varepsilon} \right), \tag{15}$$

where $R = \|x_0 - x^*\|$.

We present the proof in the full version of paper.

To sum up, we propose new ICN under new inexactness assumptions. It opens a new possibilities of choosing approximation B_x and control δ_{up} and δ_{low}. Note, if we want we can create such B_x that $\delta_{up} = 0$, then we don't need to choose this parameter inside the steps of the method. On the other hand, we can choose B_x such that $\delta_{low} = 0$, then we can control level of the errors by δ_{up} and make an Adaptive ICN that can control δ_{up} on desired level. Details on the Adaptive ICN are available in the full version of paper.

4. Quasi-Newton Approximation

In this section, we propose the approach of creating inexact Hessian by quasi-newton approximations. The main idea is simple. We calculate B_x as a quasi-newton approximation and make steps of ICN with such B_x. In this section, we focus on L-BFGS approximation as the most popular one.

To make a step of ICN we need to solve next subproblem:

$$\operatorname*{argmin}_{h \in \mathbf{E}} \left\{ f(x) + \langle \nabla f(x), h \rangle + \frac{1}{2} \langle B_k h, h \rangle + \frac{M}{6} \|h\|^3 + \frac{\delta_{up}}{2} \|h\|^2 \right\}, \tag{16}$$

where $B_x = B_k$ is L-BFGS approximation. The subproblem's first derivative with regard to h:

$$\nabla f(x_k) + (B_k + \delta_{up} I) h^* + \frac{L}{2} \|h^*\| h^* = 0 \tag{17}$$

Then the solution of the subproblem can be formulated as

$$h^* = -\left(B_k + \delta I + \frac{L}{2} \|h^*\| I \right)^{-1} \nabla f(x_k) \tag{18}$$

Note, that to find h^* we have to do a ray-search on $\|h\|$. It takes $O(log(\varepsilon^{-1})$ inversion. It is the same as for Cubic Regularized Newton but the main difference that for low-rank L-BFGS approximation this inversion is much faster to compute. It takes $O(d^3)$ computational operation for the full Hessian, where d is a dimension. For m-memory L-BFGS approximation, the inversion takes $O(m^2 d + m^3)$ computational operation that is much smaller. It makes Cubic Regularized L-BFGS computationally effective. Also, one can show that B_k is (L_1, mL_1)-inexact Hessian for classical history approximation. For greedy or sample approximation, one can show that B_k is $(L_1, 0)$-inexact Hessian.

Theorem 8 *Let $f(x)$ be a star-convex function (Option A) or μ-strongly star-convex function (Option B) with respect to global minimizer x^*, $f(x)$ has L_1-Lipschitz-continuous gradient and L_2-Lipschitz-continuous Hessian, B_k is an m-memory L-BFGS approximation, and $M \geq L_2$, then the total number of iteration $T \geq 1$ of the Cubic L-BFGS to find ε-solution x_T such that $f(x_T) - f(x^*) \leq \varepsilon$ is bounded by*

$$\textit{Option A} \qquad T = O(1) \max \left\{ \frac{mL_1 R^2}{\varepsilon}; \sqrt{\frac{MR^3}{\varepsilon}} \right\}, \tag{19}$$

$$\textit{Option B} \qquad T = O(1) \max \left\{ 1; \frac{mL_1}{\mu}; \sqrt{\frac{MR}{\mu}} \right\} \log \left(\frac{f(x_0) - f(x^*}{\varepsilon} \right), \tag{20}$$

where $R = \|x_0 - x^*\|$.

We share the details in the full paper.

5. Experiments

In this section, we present numerical experiments, which we conducted in order to show the efficiency of our proposed method. We consider l_2-regularized logistic regression problems of the form

$$F(w) = \frac{1}{n}\sum_{i=1}^{n} \log(1 + \exp(-y_i x_i^T w)) + \frac{\mu}{2}\|w\|^2, \tag{21}$$

where $(x_i, y_i)_{i=1}^n$ are the training examples described by features x_i and the class $y_i \in \{-1, 1\}$, and $\mu > 0$ is the regularization parameter. The datasets (a9a,w8a and madelon) used to present the results were taken from LibSVM library [6]. We compared the performance of Cubic L-BFGS with gradient descent (GD), Cubic Newton and classical quasi-Newton method (LBFGS). In Figures 4 and 1, we consider the classification problem on $a9a$ dataset [6]. To get better test results, the regularization $\mu = 10^{-4}$. Memory-size for both variants ofL-BFGS is $m = 10$. In order to show the globalisation properties of the methods, we consider the case when the starting point is $x_0 = 10 \cdot e$, where e is the all-one vector. For results shown in Figure 1, the parameters are fine-tuned and equal to $L_1 = 0.04$, $lr = 0.0123$ and $L_2 = 0.011$. For Figure 4, we use theoretical parameters $L_1 = 0.25$, $lr = \mu/(L_1^2)$, $L_2 = 0.1$ (of GD, L-BFGS, cubic Newton and cubic L-BFGS respectively) for all the methods. One can see that Cubic L-BFGS is very close to classical Cubic L-BFGS but with much less computations with $O(m^2 d + m^3) \sim 10^4$ compared to $O(d^3) \sim 10^6$.

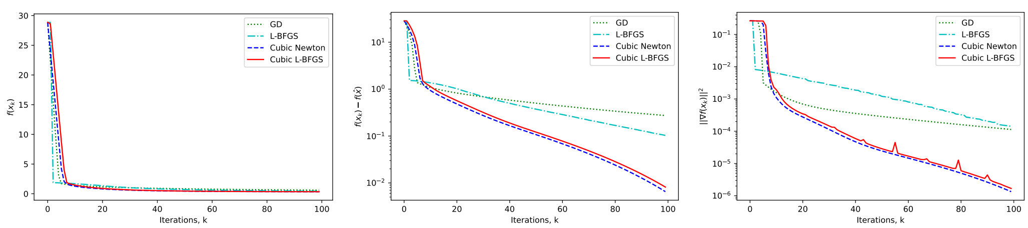

Figure 1: Methods' performance for logistic regression task on $a9a$ dataset for $x_0 = 10 \cdot e$ and fined-tuned parameters.

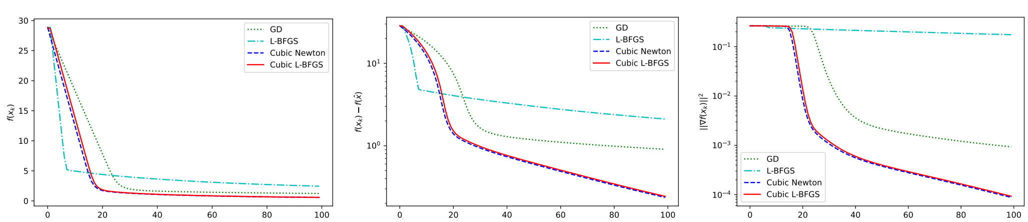

Figure 2: Methods' performance for logistic regression task on $a9a$ dataset for $x_0 = 10 \cdot e$ and theoretical parameters.

References

[1] Artem Agafonov, Dmitry Kamzolov, Pavel Dvurechensky, and Alexander Gasnikov. Inexact tensor methods and their application to stochastic convex optimization. *arXiv preprint arXiv:2012.15636*, 2020.

[2] Artem Agafonov, Pavel Dvurechensky, Gesualdo Scutari, Alexander Gasnikov, Dmitry Kamzolov, Aleksandr Lukashevich, and Amir Daneshmand. An accelerated second-order method for distributed stochastic optimization. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 2407–2413. IEEE, 2021.

[3] Albert S Berahas, Majid Jahani, Peter Richtárik, and Martin Takáč. Quasi-newton methods for machine learning: forget the past, just sample. *Optimization Methods and Software*, pages 1–37, 2021.

[4] Charles G Broyden. Quasi-newton methods and their application to function minimisation. *Mathematics of Computation*, 21(99):368–381, 1967.

[5] Richard H Byrd, Humaid Fayez Khalfan, and Robert B Schnabel. Analysis of a symmetric rank-one trust region method. *SIAM Journal on Optimization*, 6(4):1025–1039, 1996.

[6] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

[7] Andrew R Conn, Nicholas IM Gould, and Ph L Toint. Convergence of quasi-newton matrices generated by the symmetric rank one update. *Mathematical programming*, 50(1):177–195, 1991.

[8] Amir Daneshmand, Gesualdo Scutari, Pavel Dvurechensky, and Alexander Gasnikov. Newton method over networks is fast up to the statistical precision. In *International Conference on Machine Learning*, pages 2398–2409. PMLR, 2021.

[9] Pavel Dvurechensky, Dmitry Kamzolov, Aleksandr Lukashevich, Soomin Lee, Erik Ordentlich, César A Uribe, and Alexander Gasnikov. Hyperfast second-order local solvers for efficient statistically preconditioned distributed optimization. *arXiv preprint arXiv:2102.08246*, 2021.

[10] Roger Fletcher. A new approach to variable metric algorithms. *The computer journal*, 13(3): 317–322, 1970.

[11] Saeed Ghadimi, Han Liu, and Tong Zhang. Second-order methods with cubic regularization under inexact information. *arXiv preprint arXiv:1710.05782*, 2017.

[12] Donald Goldfarb. A family of variable-metric methods derived by variational means. *Mathematics of computation*, 24(109):23–26, 1970.

[13] H Fayez Khalfan, Richard H Byrd, and Robert B Schnabel. A theoretical and experimental study of the symmetric rank-one update. *SIAM Journal on Optimization*, 3(1):1–24, 1993.

[14] Bobby Kleinberg, Yuanzhi Li, and Yang Yuan. An alternative view: When does sgd escape local minima? In *International Conference on Machine Learning*, pages 2698–2707. PMLR, 2018.

[15] Dmitry Kovalev and Alexander Gasnikov. The first optimal acceleration of high-order methods in smooth convex optimization. *arXiv preprint arXiv:2205.09647*, 2022.

[16] Yurii Nesterov. Implementable tensor methods in unconstrained convex optimization. *Mathematical Programming*, 186(1):157–183, 2021.

[17] Yurii Nesterov and Boris T Polyak. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

[18] Yurii E Nesterov. A method for solving the convex programming problem with convergence rate o (1/kˆ 2). In *Dokl. akad. nauk Sssr*, volume 269, pages 543–547, 1983.

[19] David F Shanno. Conditioning of quasi-newton methods for function minimization. *Mathematics of computation*, 24(111):647–656, 1970.

[20] Stephen Wright, Jorge Nocedal, et al. Numerical optimization. *Springer Science*, 35(67-68): 7, 1999.

[21] Yuchen Zhang and Xiao Lin. Disco: Distributed optimization for self-concordant empirical loss. In *International conference on machine learning*, pages 362–370. PMLR, 2015.

Appendix A.
Missing parts from Section 2

We denote by $\mathbf{E}^*$ dual space of $\mathbf{E}$. It is the space of all linear functions on $\mathbf{E}$. The value of linear function $g \in \mathbf{E}^*$ at $h \in \mathbf{E}$ is denoted by $\langle g, h \rangle$. We assume that function $f(x)$ is twice continuous differentiable. Then, $\nabla f(x) \in \mathbf{E}^*$ is its gradient, $\nabla^2 f(x) : \mathbf{E} \to \mathbf{E}^*$ is its Hessian, note that $\nabla^2 f(x) h \in \mathbf{E}^*$ for any $h \in E$. Using a self-adjoint positive-definite operator $D : \mathbf{E} \to \mathbf{E}^*$, we can endow spaces $\mathbf{E}$ and $\mathbf{E}^*$ by conjugate Euclidean norms:

$$\|h\| = \langle Dh, h \rangle^{1/2}, \quad \forall h \in \mathbf{E}; \qquad \|g\|_* = \left\langle g, D^{-1} g \right\rangle^{1/2}, \quad \forall g \in \mathbf{E}^*.$$

For self-adjoint linear operator $B : \mathbf{E} \to \mathbf{E}^*$, we define the standard spectral norm

$$\|B\| = \max_{h \in \mathbf{E}} \left\{ |\langle Bh, h \rangle| : \|h\| \leq 1 \right\},$$

note that it corresponds to maximal module of all eigenvalues computed with respect to $D \succ 0$.

Appendix B.
Proofs of Section 3

Proof of Lemma 6

One can get the upper-bound (10) from (6)

$$\begin{aligned} f(y) - \phi_x(y) &\leq f(y) - \Phi_x(y) + \Phi_x(y) - \phi_x(y) \leq \frac{L_2}{6}\|y - x\|^3 + \Phi_x(y) - \phi_x(y) \\ &\leq \frac{L_2}{6}\|y - x\|^3 + \tfrac{1}{2}\left\langle (\nabla^2 f(x) - B_x)(y - x), y - x \right\rangle \overset{(6)}{\leq} \frac{L_2}{6}\|y - x\|^3 + \frac{\delta_{up}}{2}\|y - x\|^2. \end{aligned}$$

The lower-bound (11) comes from (7)

$$\begin{aligned} \phi_x(y) - f(y) &\leq \Phi_x(y) - f(y) + \phi_x(y) - \Phi_x(y) \leq \frac{L_2}{6}\|y - x\|^3 + \phi_x(y) - \Phi_x(y) \\ &\leq \frac{L_2}{6}\|y - x\|^3 + \tfrac{1}{2}\left\langle (B_x - \nabla^2 f(x))(y - x), y - x \right\rangle \overset{(7)}{\leq} \frac{L_2}{6}\|y - x\|^3 + \frac{\delta_{low}}{2}\|y - x\|^2. \end{aligned}$$

■

Appendix C.
Extra Experiments

In Figures 3 and 4, we consider the task of classification on the $a9a$ [6] dataset. For every data sample the number of features is $d = 123$ and $n = 20000$. We consider two cases for the starting point x_0. For $x_0 = 0$ and $\mu = 10^{-4}$, as it is shown from the figure 3 the Newton method converges very quick so $x_0 = 0$ is very close to the solution. In order to show the globalisation properties of the methods we consider the case when the starting point is $x_0 = 10 \cdot e$, where e is the all-one vector, as shown in figure 4. We use theoretical parameters $L_1 = 0.25$, $lr = \mu/(L_1^2)$, $L_2 = 0.1$ (of GD, L-BFGS, cubic Newton and cubic L-BFGS respectively) for all the methods.

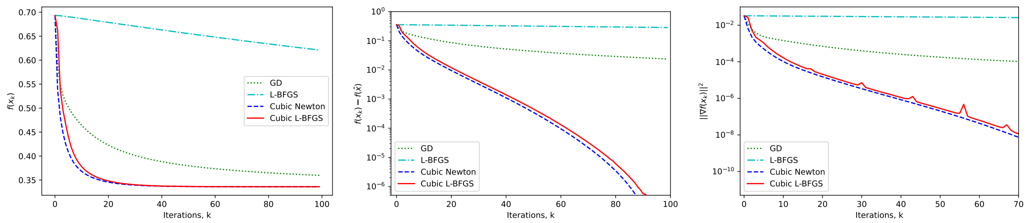

Figure 3: Comparison of Newton methods and gradient descent for logistic regression task on $a9a$ dataset for $x_0 = 0$ and theoretical parameters.

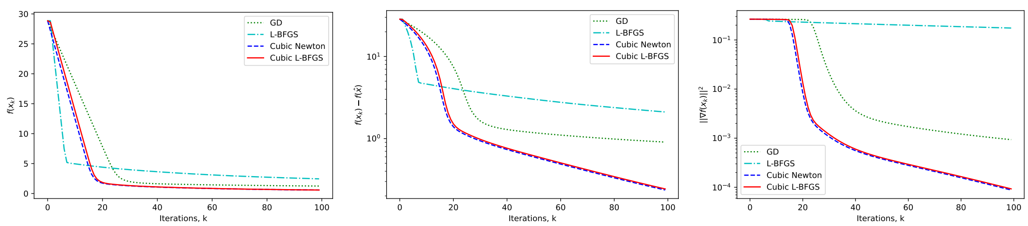

Figure 4: Comparison of Newton methods and gradient descent for logistic regression task on $a9a$ dataset for $x_0 = 10 \cdot e$ and theoretical parameters.

We solve the following minimization problem:

$$\min_{w \in \mathbb{R}^d} F(w) = \frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-y_i x_i^T w)) + \frac{\mu}{2} \|w\|^2, \tag{22}$$

We normalise each data point and get $\|x_i\|_2 = 1$ for all $i \in [1, ..., n]$. In Figures 5 and 6, we consider the task of classification on the $w8a$ [6] dataset. For every data sample the number of features is $d = 300$ and $n = 49749$. We consider two cases for the starting point x_0. For $x_0 = 0$ and $\mu = 10^{-4}$ as it is shown from figure 5 the Newton method converges very quick. In order to show the globalisation properties of the methods we consider the case when the starting point is $x_0 = 8 \cdot e$, where e is all-one vector, and $\mu = 10^{-4}$. We use theoretical parameters $L_1 = 0.25$, $lr = \mu/(L_1^2)$, $L_2 = 0.1$ (of GD, L-BFGS, cubic Newton and cubic L-BFGS respectively) for all the methods.
The parameters used for the results represented in figure 7 are $x_0 = 0$, $\mu = 10^{-4}$, $L_1 = 0.03$, $lr = 0.11$ and $L2 = 5 \cdot 10^{-5}$. We use parameters $x_0 = 8 \cdot e$, $\mu = 10^{-4}$, $L_1 = 0.03$, $lr = 0.04$ and $L2 = 5 \cdot 10^{-5}$ for the results shown in figure 8.
In Figures 9 and 10, we consider the task of classification on the $madelon$ [6] dataset. For every data sample the number of features is $d = 500$ and $n = 2000$. We consider two cases for the starting point $x_0 = 0$ and $x_0 = 3 \cdot e$ with $\mu = 10^{-4}$. We use theoretical parameters $L_1 = 0.25$, $lr = \mu/(L_1^2)$, $L_2 = 0.1$ (of GD, L-BFGS, cubic Newton and cubic L-BFGS respectively) for all the methods.
The parameters used for the results presented in figures 11 and 12 are $\mu = 10^{-4}$, $L_1 = 0.2$,

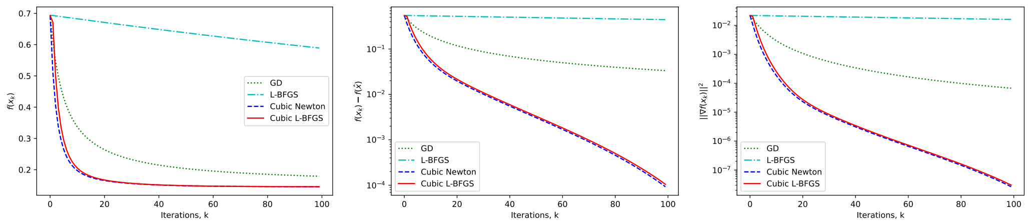

Figure 5: Comparison of Newton methods and gradient descent for logistic regression task on $w8a$ dataset for $x_0 = 0$ and theoretical parameters.

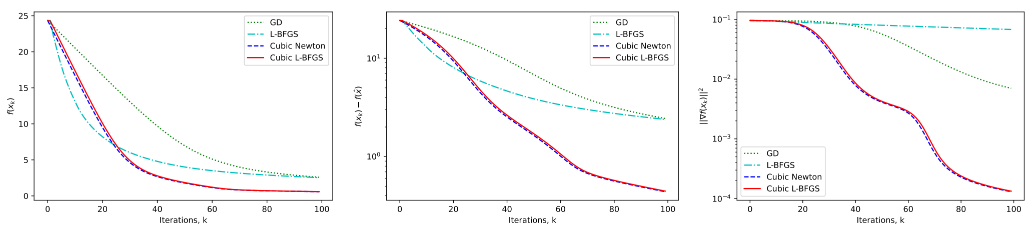

Figure 6: Comparison of Newton methods and gradient descent for logistic regression task on $w8a$ dataset for $x_0 = 8 \cdot e$ and theoretical parameters.

$lr = 0.0025$ and $L2 = 0.02$. The starting point for experiments on figure 11 is $x_0 = 0$, while the starting point for experiments in figure 12 is $x_0 = 3 \cdot e$, where e is the all-one vector.

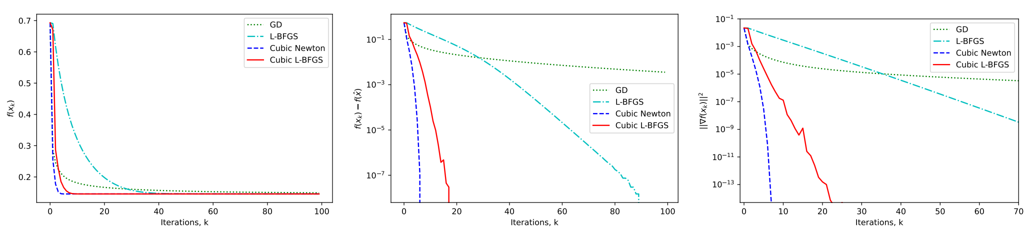

Figure 7: Comparison of Newton methods and gradient descent for logistic regression task on $w8a$ dataset for $x_0 = 0$.

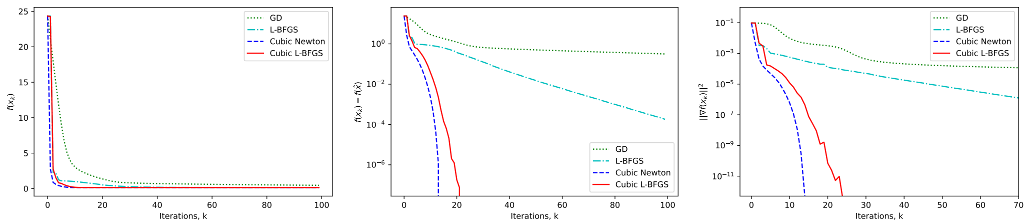

Figure 8: Comparison of Newton methods and gradient descent for logistic regression task on $w8a$ dataset for $x_0 = 8 \cdot e$.

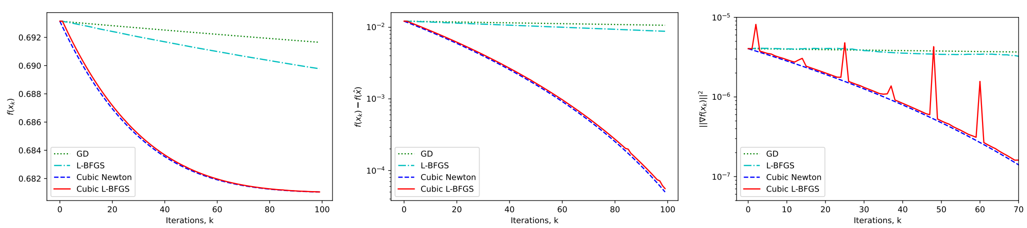

Figure 9: Comparison of Newton methods and gradient descent for logistic regression task on $madelon$ dataset for $x_0 = 0$ and theoretical parameters.

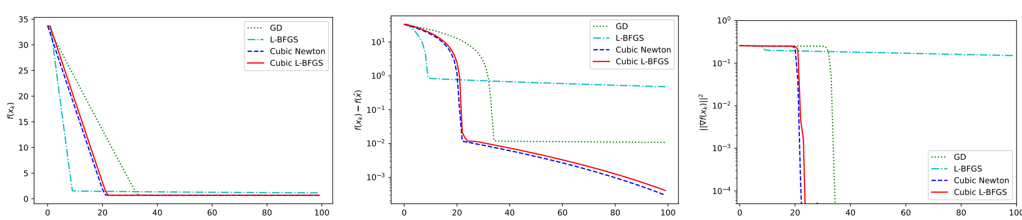

Figure 10: Comparison of Newton methods and gradient descent for logistic regression task on $madelon$ dataset with $x_0 = 3 \cdot e$ and theoretical parameters.

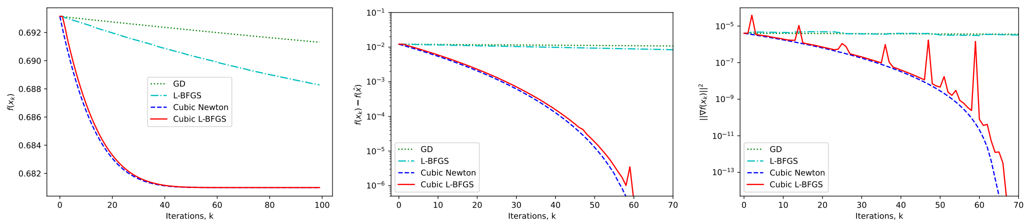

Figure 11: Comparison of Newton methods and gradient descent for logistic regression task on $madelon$ dataset.

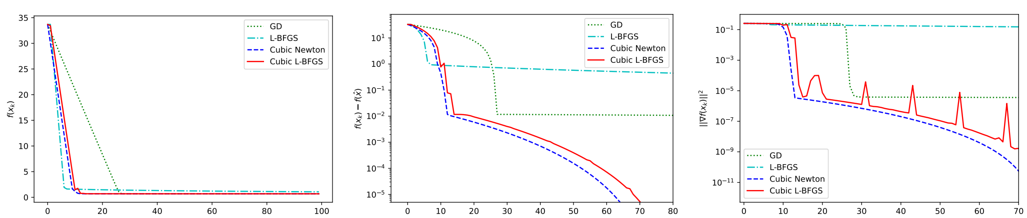

Figure 12: Comparison of Newton methods and gradient descent for logistic regression task on $madelon$ dataset with $x_0 = 3 \cdot e$.