

---

# Global Convergence Newton

---

**Raffael Colonnello**  
University of Basel  
Raffael.Colonnello@unibas.ch

**Fynn Gohlke**  
University of Basel  
Fynn.Gohlke@stud.unibas.ch

**Benedikt Heuser**  
University of Basel  
ben.heuser@unibas.ch

## Abstract

1       The abstract paragraph should be indented 1/2 inch (3 picas) on both the left- and  
2       right-hand margins. Use 10 point type, with a vertical spacing (leading) of 11 points.  
3       The word **Abstract** must be centered, bold, and in point size 12. Two line spaces  
4       precede the abstract. The abstract must be limited to one paragraph.

## 5   1   Introduction

6   In this paper we consider problems of the

$$\min_{x \in \mathbb{R}^d} f(\mathbf{x}) \tag{1}$$

7   where  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  is a twice-differentiable function. First-order optimization methods are widely  
8   used for such problems due to their low per-iteration computational cost and their suitability for  
9   parallelization they often suffer from slow convergence for ill-conditioned objective functions [1].  
10   Newton’s method is a popular optimization algorithm that is commonly used to solve optimization  
11   problems. It is a second-order optimization algorithm since it uses second-order information of  
12   the objective function. Newton’s method is known to have fast local convergence guarantees for  
13   convex functions. However, the global convergence properties of Newton’s method are still an active  
14   area of research [2] [3]. In contrast to first-order methods such as gradient descent, second-order  
15   methods such as Newton’s method can achieve much faster convergence when presented with ill  
16   conditioned Hessians by transferring the problem into a more isotropic optimization problem at  
17   the cost of an increase to cubic run time. Newton’s method yields local quadratic convergence if  
18    $f$  is twice differentiable (or we have suitable regularity conditions), which degrades to sublinear  
19   convergence outside of the local regions.

20   In this paper, we explore the theoretical foundations of several Newton-type methods that achieve  
21   different global convergence guarantees, compare their performance in a classification-type problem  
22   for two loss functions on four different datasets. Finally we will propose two modifications of the  
23   algorithms to achieve an increase in runtime, by either coupling the Newton-type method with a  
24   conjugate gradient method for Hessian vector multiplication or Strassen’s algorithm for fast matrix  
25   inversion.

## 26 **2 Background**

### 27 **2.1 Loss function and Datasets**

28 Let  $X = \begin{bmatrix} \dots x_1^\top \dots \\ \vdots \\ \dots x_i^\top \dots \\ \vdots \\ \dots x_n^\top \dots \end{bmatrix} \in \mathbb{R}^{n \times d}$  be the set of data for  $n$  datapoints with  $d$  features, i.e.  $x_i \in \mathbb{R}^d$   
 29 and labels  $y^\top = [y_1, \dots, y_n]$   
 30 For  $\sigma(x) := \frac{\exp(x)}{1+\exp(x)}$  the loss functions w.r.t. weights  $\omega$  are given by

$$L_1(\omega) = -\frac{1}{n} \sum_{i=1}^n \left( y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i) \right), \quad \hat{y}_i = \sigma(x_i^\top \omega) \quad (2)$$

$$L_2(\omega) = \frac{1}{n} \sum_{i=1}^n \log(1 + \exp(-y_i x_i^\top \omega)) + r(\omega), \quad r(\omega) = \lambda \sum_{j=1}^d \frac{\alpha \omega_j^2}{1 + \alpha \omega_j^2} \quad (3)$$

$$(4)$$

31 which yields the two optimization problems

$$\min_{\omega} L_1(\omega) \quad (5)$$

$$\min_{\omega} L_2(\omega) \quad (6)$$

32 The corresponding gradients of  $L_i$  are

$$\nabla L_1(x) = \frac{1}{n} X^\top (\hat{y} - y) \quad (7)$$

$$\nabla L_2(x) = -\frac{1}{n} X^\top (y \odot \sigma(-y \odot (X\omega))) + \nabla r(x) \quad (8)$$

$$\text{with } \nabla r(\omega)^\top = \lambda \cdot \left[ \frac{2\alpha\omega_1}{(1 + \alpha\omega_1^2)^2}, \dots, \frac{2\alpha\omega_d}{(1 + \alpha\omega_d^2)^2} \right] \quad \text{and } \sigma(\cdot) \text{ applied elementwise}$$

and  $\odot$  denotes the entrywise multiplication of vectors.

33 Differentiating again yields the Hessians

$$\nabla^2 L_1(\omega) = \frac{1}{n} X^\top DX \quad (9)$$

$$\nabla^2 L_2(\omega) = \frac{1}{n} X^\top DX + \nabla^2 r(\omega), \quad \nabla^2 r(\omega) = \text{diag} \left( \lambda \frac{2\alpha(1 - 3\alpha\omega_j^2)}{(1 + \alpha\omega_j^2)^3} \right) \quad (10)$$

34 where the diagonal matrix  $D$  has entries

$$D_{ii} = \hat{y}_i(1 - \hat{y}_i) = \sigma(-y_i x_i^\top \omega)(1 - \sigma(-y_i x_i^\top \omega)), \quad (11)$$

35 **Observation:**

36 Since  $\log(\hat{y}_i), \log(1 - \hat{y}_i)$  are concave on  $(0, \infty)$  it follows that  $-\log(\hat{y}_i), -\log(1 - \hat{y}_i)$  are convex  
 37 and thus  $L_1$  is a linear combination of convex functions (which is again convex). Meanwhile  $L_2$  is  
 38 not guaranteed to be convex due to the non-convex regularization term  $r(\omega)$ .

### 39 **2.2 Classic Newton's Method**

40 The classical origin of Newton's method is as an algorithm for finding the roots of functions. In  
 41 this paper it is used to find the roots  $x^*$  of  $\nabla(f(x))$  s.t.  $\nabla(f(x^*)) = 0$  and  $x^*$  a local minimum of  $f$ .  
 42 Newton's method combined with a stepsize  $\eta$  uses the update rule [1]:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta(\nabla^2 f(\mathbf{x}_k))^{-1} \nabla f(\mathbf{x}_k) \quad (12)$$

43 The inverse Hessian can be interpreted as transforming the gradient landscape to be more isotropic,  
 44 thereby improving the conditioning of the problem.

## 45 2.3 Cubic Newton

46 The cubic Newton method was one of the first to achieve a good complexity guarantee globally  
 47 [REFERENCE TO DO: What convergence rate exactly?]. It is based on cubic regularization and uses  
 48 the update rule:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (\nabla^2 f(\mathbf{x}_k) + H \|\mathbf{x}_{k+1} - \mathbf{x}_k\| \mathbf{I})^{-1} \nabla f(\mathbf{x}_k) \quad (13)$$

## 49 2.4 Levenberg and Marquardt method

50 The Levenberg-Marquardt's algorithm [REFERENCE] is an early form of regularized Newton's  
 51 method that modifies the Hessian. For ill conditioned (or singular) H this can increase convergence (or  
 52 make the problem solvable as  $H + \lambda \mathbf{I}$  is always invertible for sufficiently large  $\text{eig}(H) > -\lambda, \lambda >$   
 53 0).he update rule is:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (\nabla^2 f(\mathbf{x}_k) + \lambda_k \mathbf{I})^{-1} \nabla f(\mathbf{x}_k) \quad (14)$$

## 54 2.5 Regularized Newton

55 In their 2023 article Michenko presents a variation of Newton's method that uses the update rule [2]:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (\nabla^2 f(\mathbf{x}_k) + \sqrt{H \|\nabla f(\mathbf{x}_k)\|} \mathbf{I})^{-1} \nabla f(\mathbf{x}_k) \quad (15)$$

56 where  $H > 0$  is a constant. The convergence rate of this algorithm is  $\mathcal{O}(\frac{1}{k^2})$ . This method uses an  
 57 adaptive variant of the Levenberg-Marquardt regularization.

## 58 2.6 Appendix

$$\begin{aligned} L_1(\omega) &= -\frac{1}{n} \sum_{i=1}^n \left[ y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i) \right] \\ \hat{y}_i &= \sigma(x_i^\top \omega) = \frac{1}{1 + e^{-x_i^\top \omega}} \\ \frac{\partial}{\partial \omega} (-y_i \log \hat{y}_i) &= -y_i \frac{1}{\hat{y}_i} \hat{y}_i (1 - \hat{y}_i) x_i = -y_i (1 - \hat{y}_i) x_i \\ \frac{\partial}{\partial \omega} (-(1 - y_i) \log(1 - \hat{y}_i)) &= (1 - y_i) \frac{1}{1 - \hat{y}_i} \hat{y}_i (1 - \hat{y}_i) x_i = (1 - y_i) \hat{y}_i x_i \\ \nabla L_1(\omega) &= \frac{1}{n} \sum_{i=1}^n [\hat{y}_i - y_i] x_i = \frac{1}{n} X^\top (\hat{y} - y) \end{aligned}$$

59 TODO: Hessian von L1 ausrechnen. (Ist aber gesichert korrekt wegen IBM)

$$L_2(\omega) = \frac{1}{n} \sum_{i=1}^n \underbrace{\log(1 + \exp(-y_i x_i^\top \omega))}_{f_i(\omega)} + \lambda \underbrace{\sum_{j=1}^d \frac{\alpha \omega_j^2}{1 + \alpha \omega_j^2}}_{r(\omega)}$$

60 For the gradient we then get

$$\begin{aligned}
\frac{\partial}{\partial \omega_j} r(\omega) &= 2\lambda\alpha \frac{\omega_j}{(1 + \alpha\omega_j^2)^2} \implies \nabla r(\omega) = 2\lambda\alpha \frac{\omega}{(1 + \alpha\omega^2)^2} \\
\nabla f_i(\omega) &= \frac{\partial}{\partial \omega} \log(1 + e^{-y_i x_i^\top \omega}) \\
&= \frac{1}{\underbrace{1 + e^{y_i x_i^\top \omega}}_{\sigma(-y_i x_i^\top \omega)}} \cdot (-y_i x_i) = \sigma(-y_i x_i^\top \omega) \cdot (-y_i x_i) = -y_i x_i \sigma(-y_i x_i^\top \omega) \\
\nabla f(\omega) &= -\frac{1}{n} \sum_{i=1}^n y_i x_i \sigma(-y_i x_i^\top \omega) = -\frac{1}{n} X^\top (y \odot \sigma(-y \odot (X\omega))) \\
\nabla L_2(\omega) &= \nabla f(\omega) + \nabla r(\omega) \\
&= -\frac{1}{n} X^\top (y \odot \sigma(-y \odot (X\omega))) + 2\lambda\alpha \frac{\omega}{(1 + \alpha\omega^2)^2}
\end{aligned}$$

61 For the Hessians we first observe two remarks:

62 Remark 1:

$$\begin{aligned}
\sigma(z) &= \frac{1}{1 + e^{-z}} \\
\implies \frac{d}{dz} \sigma(z) &= \frac{d}{dz} (1 + e^{-z})^{-1} = -(1 + e^{-z})^{-2} \cdot (-e^{-z}) = \frac{e^{-z}}{(1 + e^{-z})^2} = \frac{1}{1 + e^{-z}} \cdot \frac{e^{-z}}{1 + e^{-z}} = \sigma(z)(1 - \sigma(z))
\end{aligned}$$

63 Remark 2: By chain rule we have

$$\begin{aligned}
z_i(\omega) &:= -y_i x_i^\top \omega \\
\implies \nabla_\omega z_i(\omega) &= -y_i x_i \\
\implies \nabla_\omega \sigma(z_i(\omega)) &= \sigma'(z_i(\omega)) \nabla_\omega z_i(\omega) \\
&= \sigma(-y_i x_i^\top \omega) (1 - \sigma(-y_i x_i^\top \omega)) (-y_i x_i)
\end{aligned}$$

64 From the gradient we have

$$\nabla_\omega^2 f(\omega) = \nabla_\omega \left( -\frac{1}{n} X^\top (y \odot \sigma(-y \odot (X\omega))) \right) = -\frac{1}{n} X^\top \nabla_\omega (y \odot \sigma(-y \odot (X\omega)))$$

65 Now notice, that

$$y \odot \sigma(-y \odot (X\omega)) = \begin{pmatrix} y_1 \sigma(-y_1 x_1^\top \omega) \\ y_2 \sigma(-y_2 x_2^\top \omega) \\ \vdots \\ y_n \sigma(-y_n x_n^\top \omega) \end{pmatrix}$$

66 and applying Remark 2 yields

$$\begin{aligned}
\nabla_\omega \sigma(-y_i x_i^\top \omega) &= \sigma(-y_i x_i^\top \omega) (1 - \sigma(-y_i x_i^\top \omega)) (-y_i x_i) \\
\implies \nabla_\omega (y_i \sigma(-y_i x_i^\top \omega)) &= -y_i^2 \sigma(-y_i x_i^\top \omega) (1 - \sigma(-y_i x_i^\top \omega)) x_i
\end{aligned}$$

$$\begin{aligned}
& \Rightarrow \nabla_{\omega}(y \odot \sigma(-y \odot (X\omega))) = \begin{pmatrix} \overbrace{-y_1^2 \sigma(-y_1 x_1^{\top} \omega) (1 - \sigma(-y_1 x_1^{\top} \omega))}^{=D_{1,1}} x_1 \\ \vdots \\ \underbrace{-y_n^2 \sigma(-y_n x_n^{\top} \omega) (1 - \sigma(-y_n x_n^{\top} \omega))}_{D_{n,n}} x_n \end{pmatrix} \\
& = \begin{bmatrix} D_{1,1} & 0 & \cdots & 0 \\ 0 & D_{2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & D_{n,n} \end{bmatrix} \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,d} \end{bmatrix} = \begin{bmatrix} D_{1,1} x_{1,1} & D_{1,1} x_{1,2} & \cdots & D_{1,1} x_{1,d} \\ D_{2,2} x_{2,1} & D_{2,2} x_{2,2} & \cdots & D_{2,2} x_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ D_{n,n} x_{n,1} & D_{n,n} x_{n,2} & \cdots & D_{n,n} x_{n,d} \end{bmatrix} \\
& = \begin{bmatrix} D_{1,1} x_1^{\top} \\ D_{2,2} x_2^{\top} \\ \vdots \\ D_{n,n} x_n^{\top} \end{bmatrix} = \frac{1}{n} DX
\end{aligned}$$

68 where we factored out the  $x_i$  in the last step to rewrite it as matrix-vector product.

$$\begin{aligned}
& = \frac{1}{n} X^{\top} DX \\
& D_{ii} = y_i^2 \sigma(-y_i x_i^{\top} \omega) (1 - \sigma(-y_i x_i^{\top} \omega)) \nabla_{\omega}^2 r(\omega) = \nabla_{\omega} \left( 2\lambda \alpha \frac{\omega_j}{(1 + \alpha \omega_j^2)^2} \right) \\
& \frac{\partial^2}{\partial \omega_j^2} r(\omega) = 2\lambda \alpha \frac{\partial}{\partial \omega_j} \left( \frac{\omega_j}{(1 + \alpha \omega_j^2)^2} \right) \\
& = 2\lambda \alpha \frac{(1 + \alpha \omega_j^2)^2 - 4\alpha \omega_j^2 (1 + \alpha \omega_j^2)}{(1 + \alpha \omega_j^2)^4} \\
& = 2\lambda \alpha \frac{1 - 3\alpha \omega_j^2}{(1 + \alpha \omega_j^2)^3} \\
& \nabla^2 r(\omega) = \text{diag} \left( 2\lambda \alpha \frac{1 - 3\alpha \omega_j^2}{(1 + \alpha \omega_j^2)^3} \right)_{j=1, \dots, d}
\end{aligned}$$

$$\text{Final Hessian: } \nabla^2 L_2(\omega) = \nabla^2 f(\omega) + \nabla^2 r(\omega) \quad (16)$$

$$= \frac{1}{n} X^{\top} DX + \text{diag} \left( 2\lambda \alpha \frac{1 - 3\alpha \omega^2}{(1 + \alpha \omega^2)^3} \right) \quad (17)$$

$$D_{ii} = \sigma(-y_i x_i^{\top} \omega) (1 - \sigma(-y_i x_i^{\top} \omega)) \quad (18)$$

## 69 2.7 Inexact Newton Method

Given that Newton has cubic complexity we now outline how we aim to reduce the runtime by extending CG and MINRES methods to the Newton-type methods described in our paper. In order for the modified algorithms to inherit the convergence guarantees of the algorithms we want to approximate  $p$  s.t.

$$\|Hp + \nabla f\| \leq \epsilon \quad (\text{absolute tolerance}) < \epsilon = 10^{-8}$$

Since  $H_{1,2} = \nabla^2 L_{1,2}$  are clearly symmetric (as both  $X^{\top} DX$  and  $\nabla^2 r(x)$  are) we can apply the conjugate gradient method if the H is positive definite or have to fall back on MINRES if it is not pd. Positive definiteness depends on the data matrix and the regularizer curvature. [TODO: runtime for MINRES and CG]

Every iteration of Vanilla Newton takes  $O(n^3)$  per iteration because inversion of the Hessian costs

$O(n^3)$ .

for symmetric applying CG to newton drops the effort for conversion down to

$$O(k \cdot n^2) = O(\sqrt{\kappa} \log(1/\epsilon) \cdot n^2)$$

70 where  $\kappa(H) = \frac{\lambda_{max}(H)}{\lambda_{min}(H)}$

71 Precondition with SSOR to reduce condition number.

## 72 References

73 [1] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.

74 [2] Konstantin Mishchenko. Regularized newton method with global convergence. *SIAM Journal on*  
75 *Optimization*, 33(3):1440–1462, 2023.

76 [3] Slavomír Hanzely, Dmitry Kamzolov, Dmitry Pasechnyuk, Alexander Gasnikov, Peter Richtárik,  
77 and Martin Takáč. A damped newton method achieves global  $(o)(\frac{1}{k^2})$  and local quadratic  
78 convergence rate. *Advances in Neural Information Processing Systems*, 35:25320–25334, 2022.

## 79 Checklist

80 The checklist follows the references. Please read the checklist guidelines carefully for information on  
81 how to answer these questions. For each question, change the default **[TODO]** to **[Yes]**, **[No]**, or  
82 **[N/A]**. You are strongly encouraged to include a **justification to your answer**, either by referencing  
83 the appropriate section of your paper or providing a brief inline description. For example:

- 84 • Did you include the license to the code and datasets? **[Yes]** See Section
- 85 • Did you include the license to the code and datasets? **[No]** The code and the data are  
86 proprietary.
- 87 • Did you include the license to the code and datasets? **[N/A]**

88 Please do not modify the questions and only use the provided macros for your answers. Note that the  
89 Checklist section does not count towards the page limit. In your paper, please delete this instructions  
90 block and only keep the Checklist section heading above along with the questions/answers below.

91 1. For all authors...

- 92 (a) Do the main claims made in the abstract and introduction accurately reflect the paper’s  
93 contributions and scope? **[TODO]**
- 94 (b) Did you describe the limitations of your work? **[TODO]**
- 95 (c) Did you discuss any potential negative societal impacts of your work? **[TODO]**
- 96 (d) Have you read the ethics review guidelines and ensured that your paper conforms to  
97 them? **[TODO]**

98 2. If you are including theoretical results...

- 99 (a) Did you state the full set of assumptions of all theoretical results? **[TODO]**
- 100 (b) Did you include complete proofs of all theoretical results? **[TODO]**

101 3. If you ran experiments...

- 102 (a) Did you include the code, data, and instructions needed to reproduce the main experi-  
103 mental results (either in the supplemental material or as a URL)? **[TODO]**
- 104 (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they  
105 were chosen)? **[TODO]**
- 106 (c) Did you report error bars (e.g., with respect to the random seed after running experi-  
107 ments multiple times)? **[TODO]**
- 108 (d) Did you include the total amount of compute and the type of resources used (e.g., type  
109 of GPUs, internal cluster, or cloud provider)? **[TODO]**

110 4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

- 111 (a) If your work uses existing assets, did you cite the creators? **[TODO]**
- 112 (b) Did you mention the license of the assets? **[TODO]**
- 113 (c) Did you include any new assets either in the supplemental material or as a URL?
- 114 **[TODO]**
- 115 (d) Did you discuss whether and how consent was obtained from people whose data you're
- 116 using/curating? **[TODO]**
- 117 (e) Did you discuss whether the data you are using/curating contains personally identifiable
- 118 information or offensive content? **[TODO]**
- 119 5. If you used crowdsourcing or conducted research with human subjects...
- 120 (a) Did you include the full text of instructions given to participants and screenshots, if
- 121 applicable? **[TODO]**
- 122 (b) Did you describe any potential participant risks, with links to Institutional Review
- 123 Board (IRB) approvals, if applicable? **[TODO]**
- 124 (c) Did you include the estimated hourly wage paid to participants and the total amount
- 125 spent on participant compensation? **[TODO]**

## 126 **A Appendix**

127 Optionally include extra information (complete proofs, additional experiments and plots) in the  
128 appendix. This section will often be part of the supplemental material.