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

The abstract paragraph should be indented ½ inch (3 picas) on both the left- and right-hand margins. Use 10 point type, with a vertical spacing (leading) of 11 points. The word **Abstract** must be centered, bold, and in point size 12. Two line spaces precede the abstract. The abstract must be limited to one paragraph.

1 Introduction

In this paper we consider problems of the form

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

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

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

2 Background

2.1 Loss function and Datasets

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. $x_i \in \mathbb{R}^d$ and labels $y^\top = [y_1, ..., y_n]$

For $\sigma(x) := \frac{\exp(x)}{1+\exp(x)}$ the loss functions w.r.t. weights ω are given by

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

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

which yields the two optimization problems

$$\min_{\omega} L_1(\omega) \tag{4}$$

$$\min_{\omega} L_2(\omega) \tag{5}$$

Remark 1: The 0-1 loss function for logistic regression is given by

$$-\sum_{i=1}^{N}\log\left[\mu_i^{\mathbb{I}(y_i=1)}(1-\mu_i)^{\mathbb{I}(y_i=0)}\right] = -\sum_{i=1}^{N}\left[y_i\log\mu_i + (1-y_i)\log(1-\mu_i)\right]$$

for labels $y_i \in \{0,1\}$ [4, Eq. 8.2–8.3]. If we instead use labels $\tilde{y}_i \in \{-1,+1\}$, the negative log-likelihood becomes

$$\sum_{i=1}^{N}\log\big(1+\exp(-\tilde{y}_i\,\mathbf{w}^T\mathbf{x}_i)\big)$$

[4, Eq. 8.4]. To ensure the loss functions correspond to the correct likelihood, the label encoding must match the loss form [4, Sec. 8.3.1]. Consequently labels were adapted conditioned to meet the loss functions requirements.

The corresponding gradients of L_i are

$$\nabla L_1(x) = \frac{1}{n}X^\top(\hat{y}-y) \tag{6}$$

$$\nabla L_2(x) = -\frac{1}{n}X^\top\big(y\odot\sigma(-y\odot(X\omega))\big) + \nabla r(x) \tag{7}$$

with $\nabla r(\omega)^\top = \lambda\left[\frac{2\alpha\omega_1}{(1+\alpha\omega_1^2)^2}, \dots, \frac{2\alpha\omega_d}{(1+\alpha\omega_d^2)^2}\right]$, where $\sigma(\cdot)$ is applied elementwise, and $\odot$ denotes the entrywise multiplication of vectors.

Differentiating again yields the Hessians

$$\nabla^2 L_1(\omega) = \frac{1}{n}X^\top D(\omega)X \tag{8}$$

$$\nabla^2 L_2(\omega) = \frac{1}{n}X^\top D(\omega)X + \nabla^2 r(\omega), \quad \nabla^2 r(\omega) = \operatorname{diag}\Big(\lambda\frac{2\alpha(1-3\alpha\omega_j^2)}{(1+\alpha\omega_j^2)^3}\Big) \tag{9}$$

where the diagonal matrix $D(\omega)$ has entries

$$D_{ii}(\omega) = \hat{y}_i(1-\hat{y}_i) = \sigma(-y_i x_i^\top\omega)\big(1-\sigma(-y_i x_i^\top\omega)\big), \tag{10}$$

In order to discuss the Algorithms assumptions and conditions in the later sections of this paper we will first state a few properties of the given problems.

2.2 Differentiability

Both L_1 and L_2 satisfy $L_1, L_2 \in C^\infty(\Omega)$ since both functions are compositions of functions from $C^\infty(\Omega)$. 0. Symmetry of Hessian: It is easy to verify that the Hessians of both loss functions are symmetric as

$$(X^\top DX)\top = (X^\top D^\top (X^\top)^\top) = X^\top DX$$

and for $\nabla^2 L_2$ symmetry is even more trivial as we only add a diagonal matrix $\nabla^2 r(\omega)$ onto the first one.

2.3 Invertibility

The matrix $\nabla^2 L_1(\omega) = X^\top D(\omega)X$ is invertible if and only if X has full rank.
Proof: "$\Longrightarrow$" (by contrapositive) Assume that X doesnt have full rank, then by the definition of rank it holds, that

$$\exists y \neq 0 \quad s.t. \quad Xy = 0 \implies X^\top DX \underbrace{Xy}_{=0} = 0$$

$$\implies X^\top DX \quad \text{rank deficient} \implies X^\top DX \text{ not invertible}$$

"$\Longleftarrow$" Assume X has full rank and let $y \neq 0$. Then $Xy \neq 0 \quad \forall y \neq 0$

$$\implies \quad \underbrace{(Xy)^\top}_{\neq 0 \text{ as full rank}} \quad D(Xy) > 0 \implies X^\top DX \quad \text{is pd} \implies X^\top DX \quad \text{invertible because}$$

for positive definite (square) matrices M it holds

$$det(M) = det(U\Lambda U^\top) = det(U)det(M)det(U^T) = \underbrace{det(\underbrace{UU^\top}_{=I})}_{=1} \underbrace{det(M)}_{>0} > 0$$

From the four available datasets we selected `a9a`, `ijcnn1` and `covtype` and using the described criterion we proved, that the Hessian of L_1 is invertible for `ijcnn1` but not for ijcnn1 and `covtype`. Since for singular Hessians Newton's method fails due to the necessary inversion of the Hessian during the update step it is necessary to pick an algorithm with a sufficient regularization for the respective datasets with L_1.
For the matrix $\nabla^2 L_2(\omega) = X^\top D(\omega)X + \nabla^2 r(\omega)$ we will show, that for finite weights there exists a sufficient choice of α s.t. $D \succ 0$ (positive defnite (pd)) holds.
$Proof$: Assume for now, that $\nabla^2 r(\omega)$ is pd (we will show this at the end of this proof) for some sufficient choice of alpha w.r.t. finite weights $|w_j| < \infty$.

We first observe, that $X^\top DX$ is positive semi-definite (psd), i.e. $y^\top X^\top D \overbrace{Xy}^{=\xi} = \xi^\top D\xi \geq 0$ because D is pd which implies that $y^\top Dy > 0 \quad \forall y \neq 0$ and combining this with the observation, that $\xi = Xy$ could be equal

$$y^\top (X^\top DX + \nabla^2 r(\omega))y = \underbrace{y^\top X^\top DXy}_{\geq 0} + \underbrace{y^\top \nabla^2 r(\omega)y}_{>0} > 0 \implies \nabla^2 L_2(\omega) \quad \text{pd} \implies \text{invertible}$$

Inspecting the Hessian of the non convex regularizer $\nabla^2 r(\omega) = \text{diag}\left(\lambda \frac{2\alpha(1-3\alpha\omega_j^2)}{(1+\alpha\omega_j^2)^3}\right)$ of the cross entropy loss function L_2 we directly notice, that since the matrix is psd for $\lambda > 0, \alpha > 0$, it is invertible if and only if it is pd (because if it's not pd it means that one of its eigenvalues must be 0). The matrix is diagonal and inspecting it's entries we directly see that pd holds if and onl yif $1 - 3\alpha \cdot w_j^2 > 0$ is satisfied, which is true for $\alpha < \frac{1}{3w_j^2}$. Thus we can always find a feasible choice for $\alpha > 0$ s.t $\nabla^2 r(\omega)$ is pd for finite weights. In our experiments we are presented with two practical issues, that weaken this statement. First we were given what we understood to be a mandatory parameter choice of $\alpha = 1$ in the project description. The more interesting observation however, is that even when allowed to choose the regularization parameter $\alpha > 0$ freely, the machine precision will treat any weight entries above the machine precision number as infinite and thus even though D is analytically pd we have that numerically the matrix degenerates for large weights. Numerically this can be stabilized by bounding weights heuristically, but since we focused on comparing the performance of different Newton-type methods for practical problems we refrained from doing so to not bias the results. Therefore we cannot guarantee, that the Hessian of our second loss function L_2 is invertible. In the experiments we will see that in fact singular Hessians appear for this Loss functions, making it intractable to solve with non-regularized Newton-type methods.

2.4 Positive semidefiniteness of Hessian

In the previous section we proved, that $\nabla^2 L_1$ is psd while L_2 does not necessarily have this property due to possibly negative eigenvalues of the non-convex regularization term $\nabla^2 r(\omega)$.

2.5 Positive definiteness of Hessian

Since $\nabla^2 L_1(\omega)$ is psd we know, that the Hessian is pd for a dataset, if and only if it is invertible (because psd hessians have non-negative eigenvalues and if they are invertible all eigenvalues are non-zero (meaning they only have positive eigenvalues and thus they're pd). It follows that the Hessian of `ijcnn1` is pd while the Hessians of `a9a` and `covtype` are not pd. Since the Hessian of the regularization term $\nabla^2 r(\omega)$ potentially has negative diagonal entries $\nabla^2 L_2(\omega)$ is not guaranteed to pd. [TODO: I could analyze when exactly that happens, but it feels pretty useless tbh, as the point was made I wanted to make i.e. that we cannot assume pd for AICN].

2.6 Convexity

We know that a twice differentiable function f is convex if and only if its Hessian is psd. After our previous observations we conclude, that L_1 is convex, while L_2 is not.

2.7 Hessian Lipschitz

Both Hessians are Lipschitz, as $\frac{1}{n}X^\top DX =: M$ satisfies

$$\|M(\omega_1) - M(\omega_2)\| = \frac{1}{n}\|X^\top(D(\omega_1) - D(\omega_2))X\| \leq \underbrace{\frac{1}{n}\|X^\top\|\|X\|}_{=:C} \|D(\omega_1) - D(\omega_2)\|$$

Now consider that $d(\sigma) := \sigma(z_k)(1-\sigma(z_k))$ where $z_k = y_i x_i^\top \omega_k$ and observe, that $\frac{d}{d\sigma} = 1 - 2\sigma$ for $\sigma \in (0,1)$ then it follows, that $\sigma'(z) = \sigma(z)(1-\sigma(z)$ (by Remark 2) and by mean value theorem (MVT) we can conclude, that for

$$d'(z) = \sigma(z)(1-\sigma(z))(1-2\sigma(z))$$

we have

$$|d(z_1) - d(z_2)| \leq \sup_z |d'(z)| \cdot |z_1 - z_2| \quad \text{where}$$

$$|z_1 - z_2| = |x_i^\top(\omega_1 - \omega_2)| \leq \underbrace{|y_i|}_{\leq 1 \quad Remark 1} \|x_i\|\|\omega_1 - \omega_2\|$$

$$\implies |D_{ii}(\omega_1) - D_{ii}(\omega_2)| \leq \sup_z |d'(z)| \cdot \|x_i\|\|\omega_1 - \omega_2\|$$

$$\|D(\omega_1) - D(\omega_2)\| \leq \sup_z |d'(z)| \cdot \max_i \|x_i\|\|\omega_1 - \omega_2\|$$

and since we have

$$\sup_z |d'(z)| = \max_{\sigma \in (0,1)} |\sigma(1-\sigma)(1-2\sigma)|$$

which takes its maximum at $\sigma^* = \frac{1}{2} \pm \frac{1}{2\sqrt{3}}$ and yields $d(\sigma^*) = \frac{1}{4}$ we conclude

$$\|D(\omega_1) - D(\omega_2)\| \leq \underbrace{\frac{1}{4}\max_i \|x_i\|}_{=:L'} \|\omega_1 - \omega_2\|$$

$$\implies \|M(\omega_1) - M(\omega_2)\| \leq C\|D(\omega_1) - D(\omega_2)\| \leq \underbrace{CL'}_{=:L} \|\omega_1 - \omega_2\|$$

Since the gradient of the regularization term is also clearly bounded it follows, that L_2 is Lipschitz aswell, as it is the sum of Lipschitz functions.

2.8 L_{semi} semi-strongly self-concordance

Referring to [3] for the respective definitions of the norms and the statement itself (Definition 3 in [3]) we notice, that semi-strongly self-concordance (sssc) implicitly assumes the invertibility of the Hessian. Since we know that L_2 is not convex and not guaranteed to be invertible (because the matrix can become singular for certain choices of weights) L_2 is not sssc. Using the same logic for invertibility of the Hessian for L_1 it follows that this loss function is not sssc for the datasets `a9a` and `covtype`. For `ijcnn1` we can prove that sssc holds. [TODO: My asslong proof for sssc of L1 for that dataset.]

2.9 Classic Newton's Method

The classical origin of Newton's method is as an algorithm for finding the roots of functions. In 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. Newton's method combined with a stepsize η uses the update rule [1]:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (\nabla^2 f(\mathbf{x}_k))^{-1} \nabla f(\mathbf{x}_k) \tag{11}$$

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

2.10 Cubic Newton

AICN gibt sich zwar als regularized method aus, kann in Wirklichkeit aber nicht umgehen die Matrix trotzdem zur Berechnung des Faktors Alpha invertieren zu müssen. Es kämpft deshalb für singulare oder illcondiitoned matrizen mit genau denselben problemen, wie unregularisierte Methoden. Kann man das sicher nicht umgehen, dass man für das Alpha das Skalarprodukt invertieren muss

2.11 Cubic Newton

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

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (\nabla^2 f(\mathbf{x}_k) + H||\mathbf{x}_{k+1} - \mathbf{x}_k||\mathbf{I})^{-1} \nabla f(\mathbf{x}_k) \tag{12}$$

2.12 Levenberg and Marquardt method

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

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

2.13 Regularized Newton

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

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (\nabla^2 f(\mathbf{x}_k) + \sqrt{H||\nabla f(\mathbf{x}_k)||}\mathbf{I})^{-1} \nabla f(\mathbf{x}_k) \tag{14}$$

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

2.14 Appendix

Remark 2:

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

$$\begin{aligned}
L_1(\omega) &= -\frac{1}{n}\sum_{i=1}^{n}\Big[\underbrace{y_i\log\hat{y}_i}_{=:A_i} + \underbrace{(1-y_i)\log(1-\hat{y}_i)}_{=:B_i}\Big] \\
\hat{y}_i &= \sigma(x_i^\top\omega) = \frac{1}{1+e^{-x_i^\top\omega}}
\end{aligned}$$

and applying Remark 2 to $\hat{y}$ we get, that

$$\begin{aligned}
\frac{\partial}{\partial\omega}A_i &= \frac{\partial}{\partial\omega}\big(-y_i\log\hat{y}_i\big) = -y_i\frac{1}{\hat{y}_i}\hat{y}_i(1-\hat{y}_i)x_i = -y_i(1-\hat{y}_i)x_i \\
\frac{\partial}{\partial\omega}B_i &= \frac{\partial}{\partial\omega}\big(-(1-y_i)\log(1-\hat{y}_i)\big) = (1-y_i)\frac{1}{1-\hat{y}_i}\hat{y}_i(1-\hat{y}_i)x_i = (1-y_i)\hat{y}_i x_i \\
\frac{\partial}{\partial\omega}A + \frac{\partial}{\partial\omega}B &= -y_i(1-\hat{y}_i)x_i + (1-y_i)\hat{y}_i x_i = \big(-y_i + y_i\hat{y}_i + \hat{y}_i - y_i\hat{y}_i\big)x_i \\
&= (-y_i+\hat{y}_i)x_i = (\hat{y}_i - y_i)x_i \\
\implies \nabla L_1(\omega) &= \frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial\omega}A_i + \frac{\partial}{\partial\omega}B_i = \frac{1}{n}\sum_{i=1}^{n}\big[\hat{y}_i - y_i\big]x_i = \frac{1}{n}X^\top(\hat{y}-y)
\end{aligned}$$

For the Hessian it then follows

$$\begin{aligned}
\nabla^2_\omega L_1(\omega) &= \nabla_\omega\frac{1}{n}X^\top(\hat{y}-y) = \frac{1}{n}X^\top = \nabla_\omega(\hat{y}-y) = \frac{1}{n}X^\top\nabla_\omega\hat{y} \\
\frac{\partial}{\partial\omega}\big(\hat{y}_i x_i\big) &= \hat{y}_i(1-\hat{y}_i)x_i x_i^\top \\
\implies \frac{\partial\hat{y}}{\partial\omega} &= \operatorname{diag}\big(\sigma(X\omega)\odot(1-\sigma(X\omega))\big)X \\
\implies \nabla^2 L_1(\omega) &= \frac{1}{n}X^\top\operatorname{diag}(\hat{y}\odot(1-\hat{y}))X \\
\implies \nabla^2 L_1(\omega) &= \frac{1}{n}X^\top D X \\
D &= \operatorname{diag}(\hat{y}_i(1-\hat{y}_i)).
\end{aligned}$$

For L_2 we have

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

For the gradient we then get

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

For the Hessians we first observe two remarks:
Remark 3: By chain rule we have

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

From the gradient we have

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

Now notice, that

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

and applying Remark 3 yields

$$\begin{aligned}
\nabla_\omega\sigma(-y_i x_i^\top\omega) &= \sigma(-y_i x_i^\top\omega)\big(1-\sigma(-y_i x_i^\top\omega)\big)(-y_i x_i) \\
\implies \nabla_\omega\big(y_i\,\sigma(-y_i x_i^\top\omega)\big) &= -\underbrace{y_i^2}_{=1 \text{ by Remark 1}}\sigma(-y_i x_i^\top\omega)\big(1-\sigma(-y_i x_i^\top\omega)\big)x_i = -\sigma(-y_i x_i^\top\omega)\big(1-\sigma(-y_i x_i^\top\omega)\big)x_i
\end{aligned}$$

$$\implies \nabla_\omega\big(y \odot \sigma(-y \odot (X\omega))\big) = -\begin{pmatrix} \overbrace{\sigma(-y_1 x_1^\top \omega)\big(1-\sigma(-y_1 x_1^\top \omega)\big)}^{=D_{1,1}} x_1 \\ \vdots \\ \underbrace{\sigma(-y_n x_n^\top \omega)\big(1-\sigma(-y_n x_n^\top \omega)\big)}_{D_{n,n}} x_n \end{pmatrix}$$

$$= -\begin{bmatrix} D_{1,1} & 0 & \cdots & 0 \\ 0 & D_{2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & D_{n,n} \end{bmatrix} \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,d} \end{bmatrix}$$

$$= -\begin{bmatrix} D_{1,1}\,x_{1,1} & D_{1,1}\,x_{1,2} & \cdots & D_{1,1}\,x_{1,d} \\ D_{2,2}\,x_{2,1} & D_{2,2}\,x_{2,2} & \cdots & D_{2,2}\,x_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ D_{n,n}\,x_{n,1} & D_{n,n}\,x_{n,2} & \cdots & D_{n,n}\,x_{n,d} \end{bmatrix} = -\begin{bmatrix} D_{1,1}\,x_1^\top \\ D_{2,2}\,x_2^\top \\ \vdots \\ D_{n,n}\,x_n^\top \end{bmatrix} = -DX$$

where we factored out the x_i in the last step to rewrite it as matrix-vector product. Deriving the entire expression we conclude:

$$\nabla^2 f(\omega) = -\frac{1}{n} X^\top \nabla_\omega\big(y \odot \sigma(-y \odot (X\omega))\big) = \frac{1}{n} X^\top D X$$
$$D_{ii} = \sigma(-y_i x_i^\top \omega)\big(1-\sigma(-y_i x_i^\top \omega)\big)$$

The hessian of the non-convex regularization term is derived by

$$\nabla^2_\omega r(\omega) = \nabla_\omega \left(2\lambda\alpha \frac{\omega_j}{(1+\alpha\omega_j^2)^2}\right)$$

$$\frac{\partial^2}{\partial \omega_j^2} r(\omega) = 2\lambda\alpha \frac{\partial}{\partial \omega_j}\left(\frac{\omega_j}{(1+\alpha\omega_j^2)^2}\right) = 2\lambda\alpha \frac{(1+\alpha\omega_j^2)^2 - 4\alpha\omega_j^2(1+\alpha\omega_j^2)}{(1+\alpha\omega_j^2)^4} = 2\lambda\alpha \frac{1-3\alpha\omega_j^2}{(1+\alpha\omega_j^2)^3}$$

$$\implies \nabla^2 r(\omega) = \operatorname{diag}\left(2\lambda\alpha \frac{1-3\alpha\omega_j^2}{(1+\alpha\omega_j^2)^3}\right)_{j=1,\dots,d}$$

Combining the steps we derive the Hessian

$$\nabla^2 L_2(\omega) = \nabla^2 f(\omega) + \nabla^2 r(\omega) = \frac{1}{n} X^\top D X + \operatorname{diag}\left(2\lambda\alpha \frac{1-3\alpha\omega^2}{(1+\alpha\omega^2)^3}\right)$$
$$D_{ii} = \sigma(-y_i x_i^\top \omega)\big(1-\sigma(-y_i x_i^\top \omega)\big)$$

2.15 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.

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

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

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

where $\kappa(H) = \frac{\lambda_{max}(H)}{\lambda_{max}(H)}$
Precondition with SSOR to reduce condition number.

References

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

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

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

[4] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, Cambridge, MA, 2012.

Checklist

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

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

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

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? **[TODO]**
   (b) Did you describe the limitations of your work? **[TODO]**
   (c) Did you discuss any potential negative societal impacts of your work? **[TODO]**
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? **[TODO]**
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? **[TODO]**
   (b) Did you include complete proofs of all theoretical results? **[TODO]**
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? **[TODO]**
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? **[TODO]**
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? **[TODO]**
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? **[TODO]**
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? **[TODO]**

(b) Did you mention the license of the assets? **[TODO]**

(c) Did you include any new assets either in the supplemental material or as a URL? **[TODO]**

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? **[TODO]**

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? **[TODO]**

5. If you used crowdsourcing or conducted research with human subjects...

   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? **[TODO]**

   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? **[TODO]**

   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? **[TODO]**

A Appendix

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