Parameter Estimation of Generalized Linear Models without Assuming their Link Function

Sreangsu Acharyya
Cloud and Information Services Lab,
Microsoft Research India.

Joydeep Ghosh
Electrical Engineering Dept.,
University of Texas Austin.

Abstract

Canonical generalized linear models (GLM) are specified by a finite dimensional vector and a monotonically increasing function called the link function. Standard parameter estimation techniques hold the link function fixed and optimizes over the parameter vector. We propose a *parameter-recovery* facilitating, jointly-convex, regularized loss functional that is optimized *globally* over the vector as well as the link function, with best rates possible under a first order oracle model. This widens the scope of GLMs to cases where the link function is unknown.

1 Introduction

Generalized linear models are an old workhorse that enjoy widespread use in regression [15]. Although the methods to estimate GLM parameters are now standard, the important problem of learning its link function does not have a compelling solution. This has remained so inspite of GLMs being a heavily used and studied tool. We provide a *convex* formulation for estimating parameters of a canonical GLM when the link function is unknown. To the best of our knowledge this is the first convex formulation to do so.

We begin with a simplistic example whose assumptions are relaxed later. Suppose that a generalized linear relation $\mathbb{R} \supset \mathcal{Y} \ni y_i = g(\langle \boldsymbol{u}, \boldsymbol{x}_i \rangle)$ holds with an unknown, continuous and strictly monotonic function $g(\cdot)$ and an unknown vector $\boldsymbol{u} \in \mathcal{W} \subset \mathbb{R}^n$. Given a data set $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)_{i=1}^N\}$ we would like to *recover* the parameter $\boldsymbol{u}$. For simplicity, we assume X, the $N \times n$ sized matrix obtained by stacking vectors $\boldsymbol{x}_i$, is full rank. If X is rank-deficient we will recover $\boldsymbol{u}$ only up to additions with Nullspace(X) and have to replace our forthcoming claims about *strict* convexity with convexity.

Although we motivate our cost function in terms of a perfect $\boldsymbol{u}$ for pedagogic ease, neither our algorithm nor our analysis require its existence. In fact, if we do know that such an $\boldsymbol{u}$ exists, we show how to achieve exponentially faster convergence.

When $g(\cdot)$ is the identity function, it is sufficient to minimize $\|\boldsymbol{y} - X\boldsymbol{w}\|^2$ with respect to $\boldsymbol{w} \in \mathcal{W}$, to estimate $\boldsymbol{u}$. Let $\mathcal{N}_{\mathcal{W}}(\boldsymbol{w}_*)$ be the normal cone of $\mathcal{W}$ at $\boldsymbol{w}_*$. When $g(\cdot)$ is not identity, the iterative technique of generating $\boldsymbol{w} \to \boldsymbol{w}_*$ that satisfies the KKT condition $\nabla\|\boldsymbol{y} - \boldsymbol{g}(X\boldsymbol{w}_*)\|^2 \in -\mathcal{N}_{\mathcal{W}}(\boldsymbol{w}_*)$ loses its sufficiency because $\|\boldsymbol{y} - \boldsymbol{g}(X\boldsymbol{w})\|^2$ need no longer be convex and may contain exponentially many (in dimensions of $\boldsymbol{x}$) local minima [2]. Without further assumptions, it becomes impossible to restrict $\|\boldsymbol{w}_* - \boldsymbol{u}\|_U^2$ [1] to an arbitrary low value. An effective alternative is to minimize a *matching* Bregman divergence [2] (described in detail in Section 2) that removes the non-convexity. When $g(\cdot)$ is identity, $\frac{1}{2}\|\boldsymbol{y} - X\boldsymbol{w}\|^2$ is indeed such a matching loss.

Recovery of $\boldsymbol{u}$ is clearly affected by our ability to use a matching loss, but we need to know $g(\cdot)$ to do so. Unless one has explicit control over the generative process, $g(\cdot)$ is rarely known. Practitioners typically assume a convenient form or use hypothesis testing to select from a small subset of all possible choices. Our method, in contrast, is to *learn* the recovery-facilitating loss function when $g(\cdot)$ is unknown.

Few things need to be kept in mind about the proposed method. First, in the rare circumstance where the link function is known, methods that exploit this knowledge would enjoy an advantage because they need not search over link functions. Second, since the space of functions considered is vast, using small datasets would overfit. To be useful we quantify how to grow

Appearing in Proceedings of the 18^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2015, San Diego, CA, USA. JMLR: W&CP volume 38.

[1] $U = X(X^\dagger X)^{-1}X^\dagger$ projects on the range space of X.

the search space, with the size of the dataset, using regularization. In our case, recovery of $\boldsymbol{u}$ will automatically imply recovery of $g(\cdot)$ upto a multifunction (i.e. a point to set map) that, when given a new $\boldsymbol{x}$, predicts an interval. Averaging the interval yields a simple and empirically effective point estimate which may be improved exploiting analytic properties of $g(\cdot)$.

2 Background

Notation: Vectors are in bold lower case letters, matrices are capitalized. We decorate a symbol with a star at the bottom (e.g. $\boldsymbol{w}_*$) to indicate optimality and at the top (e.g. ϕ^*) for Legendre conjugation [19]. Bregman divergences and their relation to exponential family densities [12] play a major role in the paper.

Bregman Divergence: Let $\boldsymbol{\phi} : \Theta \mapsto \mathbb{R}$, $\Theta = \operatorname{dom}\boldsymbol{\phi} \subseteq \mathbb{R}^d$ be a strictly convex, closed function, differentiable on $\operatorname{int}\Theta$. The corresponding Bregman divergence $D_\phi\big(\cdot\big|\big|\cdot\big) : \operatorname{dom}(\boldsymbol{\phi}) \times \operatorname{int}(\operatorname{dom}(\boldsymbol{\phi})) \mapsto \mathbb{R}_+$ is defined as $D_\phi\big(\boldsymbol{x}\big|\big|\boldsymbol{y}\big) \triangleq \boldsymbol{\phi}(\boldsymbol{x}) - \boldsymbol{\phi}(\boldsymbol{y}) - \langle \boldsymbol{x}-\boldsymbol{y}, \nabla\boldsymbol{\phi}(\boldsymbol{y})\rangle$. From strict convexity it follows that $D_\phi\big(\boldsymbol{x}\big|\big|\boldsymbol{y}\big) \geq 0$ and $D_\phi\big(\boldsymbol{x}\big|\big|\boldsymbol{y}\big) = 0$ iff $\boldsymbol{x} = \boldsymbol{y}$. Bregman divergences are (strictly) convex in their first argument, but not necessarily in their second. We only consider separable functions of the form $\boldsymbol{\phi}(\cdot) : \mathbb{R}^n \ni \boldsymbol{x} \mapsto \sum_i \phi(x_i)$ that are sums of *identical* scalar convex functions.

The **Legendre conjugate** $\psi(\cdot)$ of a function $\phi(\cdot)$ is $(\phi)^*(\boldsymbol{x}) \triangleq \psi(\boldsymbol{x}) \triangleq \sup_{\boldsymbol{\lambda}}(\langle\boldsymbol{\lambda},\boldsymbol{x}\rangle - \phi(\boldsymbol{\lambda}))$. If $\phi(\cdot)$ is a closed, strictly convex function [19], as assumed in this paper, $\big((\phi)^*\big)^*(\cdot) = \phi(\cdot)$ and $(\nabla\phi)^{-1}(\cdot) = \nabla\psi(\cdot)$ is a one to one mapping, leading to the identity:

$$D^*_\phi\big(\nabla\phi(\boldsymbol{y})\big|\big|\nabla\phi(\boldsymbol{x})\big) = D_\phi\big(\boldsymbol{x}\big|\big|\boldsymbol{y}\big). \tag{1}$$

The **infimal convolution** of $\boldsymbol{\phi}_1(\cdot)$ and $\boldsymbol{\phi}_2(\cdot)$ is denoted in this paper by $\boldsymbol{\phi}_1 \oplus \boldsymbol{\phi}_2$ and is defined as: $[\boldsymbol{\phi}_1 \oplus \boldsymbol{\phi}_2](\boldsymbol{y}) = \inf_{\boldsymbol{x}} \boldsymbol{\phi}_1(\boldsymbol{x}) + \boldsymbol{\phi}_2(\boldsymbol{y}-\boldsymbol{x})$ [19]. The following identities will be useful (in proving Theorem 1) $[\alpha\boldsymbol{\phi}(\cdot)]^* = \alpha\boldsymbol{\phi}^*(\frac{\cdot}{\alpha})$, $\quad[\boldsymbol{\phi}_1+\boldsymbol{\phi}_2]^*(\cdot) = [\boldsymbol{\phi}_1^* \oplus \boldsymbol{\phi}_2^*](\cdot)$.

The probability density of a random variable belongs to the **Exponential family** [2] if it has the form $P(Y = \boldsymbol{y} \mid \boldsymbol{\theta}) = \exp^{\langle\boldsymbol{\theta},\boldsymbol{y}\rangle - \phi^*(\boldsymbol{\theta})}$. Then the domain $\Theta = \left\{\boldsymbol{\theta}\,\middle|\,\int_{\mathcal{Y}} \exp^{\langle\boldsymbol{\theta},\boldsymbol{y}\rangle} < \infty\right\}$ is convex and the log partition function $\phi^*(\boldsymbol{\theta})$ is a convex function (strictly so if $\mathcal{Y}$ is affinely independent) [12] from which all cumulants may be recovered, e.g.: $\mathbb{E}[Y] = \nabla_{\boldsymbol{\theta}}\phi^*(\boldsymbol{\theta}) = (\nabla\phi)^{-1}(\boldsymbol{\theta})$. **Maximum likelihood** estimate $\boldsymbol{\theta}_* = \operatorname{Argmax}_{\boldsymbol{\theta}} \log P(\boldsymbol{y} \mid \boldsymbol{\theta})$ is the maximizer of the sample log likelihood. For exponential family we obtain:

$$\begin{aligned}\boldsymbol{\theta}^* &= \operatorname{Argmin}_{\boldsymbol{\theta}} D_{\boldsymbol{\phi}}\big(\boldsymbol{y}\big|\big|(\nabla\phi)^{-1}(\boldsymbol{\theta})\big) \\ &= \operatorname{Argmin}_{\boldsymbol{\theta}} D_\phi\big(\boldsymbol{y}\big|\big|\mathbb{E}[\boldsymbol{y}|\boldsymbol{x}]\big).\end{aligned} \tag{2}$$

[2] w.r.t a base measure omitted for notational simplicity.

3 Formulation

Given a strictly (or strongly) convex regularizer $\mathfrak{R}(\boldsymbol{w})$, a non-negative scalar c_N and $\boldsymbol{y} \in \mathbb{R}^N$

$$\begin{aligned}&\min_{\boldsymbol{w},\phi(\cdot)\in\mathcal{C}\star\subset\mathcal{C}} \frac{1}{N} D_{\boldsymbol{\phi}}\big(\boldsymbol{y}\big|\big|(\nabla\phi)^{-1}(X\boldsymbol{w})\big) + \frac{c_N}{N}\mathfrak{R}(\boldsymbol{w}) \\ &\equiv \frac{1}{N}\min_{\substack{\boldsymbol{w},\boldsymbol{\theta},\phi(\cdot)\in\mathcal{C}\star\subset\mathcal{C}\\ \boldsymbol{\theta}=X\boldsymbol{w}}} D_{\boldsymbol{\phi}}\big(\boldsymbol{y}\big|\big|(\nabla\phi)^{-1}(\boldsymbol{\theta})\big) + \frac{c_N}{N}\mathfrak{R}(\boldsymbol{w})\end{aligned} \tag{3}$$

is our candidate cost functional. $\mathcal{C}$ is the infinite dimensional space of all *separable* convex functions and $\mathcal{C}\star$ is a subset of it used for regularization. As we minimize (3) over $\phi(\cdot)$, the "match" is always maintained. In the absence of simplifying restrictions, that we loathe to make, such as assuming a finite dimensional parameterization of $\phi(\cdot)$, solving (3) is a challenging problem in calculus of variations. **Regularization:** $\boldsymbol{w}$ is regularized using $\mathfrak{R}(\cdot)$ whereas $\phi(\cdot)$ is regularized by restricting it to the subset $\mathcal{C}\star$ described next. We shall later quantify how to choose c_N and $\mathcal{C}\star$ in accordance with the size of the training set.

For regularization, we restrict $\phi(\cdot)$ to a specific subset of $\mathcal{C}$ whose 'size' may be chosen according to the size of the training dataset. For $\boldsymbol{\mu}(\alpha) \triangleq \alpha\boldsymbol{x} + (1-\alpha)\boldsymbol{y}$, the set $\mathcal{C}^{s,\gamma}$ of $\phi(\cdot)$ is defined by the following inequality:

$$\begin{aligned}s\|\boldsymbol{y}-\boldsymbol{x}\|^\gamma &\leq \frac{\alpha\phi(\boldsymbol{x}) + (1-\alpha)\phi(\boldsymbol{y}) - \phi(\boldsymbol{\mu}(\alpha))}{\alpha(1-\alpha)} \quad \forall\alpha\in[0,1] \\ &= \frac{\alpha D_\phi\big(\boldsymbol{x}\big|\big|\boldsymbol{\mu}(\alpha)\big) + (1-\alpha)D_\phi\big(\boldsymbol{y}\big|\big|\boldsymbol{\mu}(\alpha)\big)}{\alpha(1-\alpha)} \\ &\leq \langle\nabla\phi(\boldsymbol{y}) - \nabla\phi(\boldsymbol{x}), \boldsymbol{y}-\boldsymbol{x}\rangle\,.\end{aligned} \tag{4}$$

Inequality (4) generalizes strong convexity which is obtained at $\gamma = 2$. Let $\mathcal{C}_{L,\nu}$ denote the set of convex functions with (L,ν) Hölder continuous gradients. If $\phi(\cdot) \in \mathcal{C}^{s,\gamma}$ then $\phi^*(\cdot) \in \mathcal{C}_{\frac{1}{s},\frac{1}{\gamma-1}}$. In what follows, $\phi(\cdot)$ will be restricted to $C^{s,\gamma}_{L,\nu}$ implying $\phi_*(\cdot) \in \mathcal{C}^{\frac{1}{L},\frac{\nu+1}{\nu}}_{\frac{1}{s},\frac{1}{\gamma-1}}$. For simplicity we will denote these sets by $\mathcal{C}\star$ and refer to them as **Hölder convex** functions. Finally, if A is the adjacent difference matrix and $[\cdot]^p$ a point-wise exponentiation of a vector, we have the following pointwise inequality:

$$\begin{aligned}\frac{1}{L}[A\boldsymbol{\theta}]^{1+1/\nu} \leq [A(\nabla\phi)^{-1}(\boldsymbol{\theta})] \leq \frac{1}{s}[A\boldsymbol{\theta}]^{\frac{1}{\gamma-1}}, \\ L\|\boldsymbol{x}-\boldsymbol{y}\|^\nu \geq D_\phi\big(\boldsymbol{x}\big|\big|\boldsymbol{y}\big) \geq \|\boldsymbol{x}-\boldsymbol{y}\|^\gamma\end{aligned} \tag{5}$$

We shall show that if $c_N = o(N)^{1+1/\nu}$ then $\lim_{N\to\infty}\frac{1}{N}\|\boldsymbol{u}-\boldsymbol{w}_*\|_{X^\dagger X} \to 0$. **Non-degeneracy and Efficiency:** Note that Hölder convexity prevents (3) from being degenerate by keeping $\phi(\cdot)$ bounded away from linearity between any adjacent points $\langle\boldsymbol{x}_i,\boldsymbol{w}\rangle$ and $\langle\boldsymbol{x}_j,\boldsymbol{w}\rangle$. Furthermore this class of convex functions admit fast optimization algorithms that meets the optimal convergence rate $T^{-\frac{\gamma+3}{2\gamma}}$ possible for this class [17]. Remarkably the *accelerated gradient method* (see Section 4, Table 1) can be used for this class to achieve

the optimal convergence rate [8], provided an adjusted, effective Lipschitz constant is used. Setting $\nu = 1$ we obtain the optimal rate for convex functions with Lipschitz continuous gradient.

Although our algorithm to minimize (3) is very simple, listing its updates would not shed much light on their own. Hence we derive the algorithm step by step, removing computational obstacles in our way, the root of which is the optimization over the space of functions $\phi \in \mathcal{C}\star$. Note, $\phi(\cdot)$ not only parameterizes the loss function, its gradient also affects the right argument.

The first piece of our solution is to identify functions in $\mathcal{C}\star$ with members of a specific finite dimensional set $\mathcal{G}$, allowing us to pose (3) as a finite dimensional problem. However, this mapping from $\mathcal{C}\star$ to $\mathcal{G}$ is existential i.e., not constructive and many to one, thus leaving us with a need to optimize a function over $\mathcal{G}$ that we cannot evaluate. Working around this obstacle is the second piece. Properties of separable Bregman divergences play a vital role in the solution. In the interest of space we had to drop the third and final part that addresses how to exploit the fact that $g(\cdot)$ is analytic.

3.1 Uniqueness of the Minimum

Now, we present our first result. In (3) both $\boldsymbol{w}$ and $\phi(\cdot)$ vary, so it is important to know whether the joint optima is unique. We show that $D_{\boldsymbol{\phi}}\left(\boldsymbol{y}\middle|\middle|(\nabla\phi)^{-1}(X\boldsymbol{w})\right)$ is jointly convex in the function $\phi(\cdot)$ and vector $\boldsymbol{w}$. Since $\mathfrak{R}(\boldsymbol{w})$ is strictly convex the optimizer $\boldsymbol{w}_*$ is unique.

Theorem 1. *If $\phi \in \mathcal{C}$ then the functional $D_{\boldsymbol{\phi}}\left(\boldsymbol{y}\middle|\middle|(\nabla\phi)^{-1}(X\boldsymbol{w})\right)$ is jointly convex in $\phi, \boldsymbol{w}$.*

Proof. Let $\boldsymbol{\theta} = \langle \boldsymbol{x}, \boldsymbol{w}\rangle$ and $\bar{\boldsymbol{\theta}} = \alpha\boldsymbol{\theta}_1 + (1-\alpha)\boldsymbol{\theta}_2$. It will be sufficient to show that $D_{\boldsymbol{\phi}}\left(\boldsymbol{y}\middle|\middle|(\nabla\phi)^{-1}(\boldsymbol{\theta})\right)$ is jointly convex in $\phi(\cdot)$ and $\boldsymbol{\theta}$. Recall $D_{\boldsymbol{\phi}}\left(\boldsymbol{y}\middle|\middle|(\nabla\phi)^{-1}(\boldsymbol{\theta})\right) = \phi(\boldsymbol{y}) + \psi(\boldsymbol{\theta}) - \langle \boldsymbol{y}, \boldsymbol{\theta}\rangle$ is the Fenchel-Young gap: $\phi(\boldsymbol{y}) + \phi^*(\boldsymbol{\theta}) - \langle \boldsymbol{y}, \boldsymbol{\theta}\rangle$ denoted here by $F\begin{pmatrix}\phi\\ \boldsymbol{\theta}\end{pmatrix}$. Showing joint convexity is equivalent to showing $\overbrace{\alpha F\begin{pmatrix}\phi_1\\ \boldsymbol{\theta}_1\end{pmatrix} + (1-\alpha)F\begin{pmatrix}\phi_2\\ \boldsymbol{\theta}_2\end{pmatrix}}^{A} \geq \overbrace{F\left(\alpha\begin{pmatrix}\phi_1\\ \boldsymbol{\theta}_1\end{pmatrix} + \left((1-\alpha)\begin{pmatrix}\phi_2\\ \boldsymbol{\theta}_2\end{pmatrix}\right)\right)}^{B}$. $A = [\alpha\phi_1 + (1-\alpha)\phi_2]\left(\boldsymbol{y}\right) + \alpha\psi(\boldsymbol{\theta}_1) + (1-\alpha)\psi(\theta_2) - \left\langle \boldsymbol{y}, \bar{\boldsymbol{\theta}}\right\rangle$. $B = [\alpha\phi_1 + (1-\alpha)\phi_2]\left(\boldsymbol{y}\right) - \left\langle \boldsymbol{y}, \bar{\boldsymbol{\theta}}\right\rangle + [\alpha\phi_1 + (1-\alpha)\phi_2]^*\left(\bar{\boldsymbol{\theta}}\right)$. We have $A - B = \alpha\phi_1^*(\boldsymbol{\theta}_1) + (1-\alpha)\phi_2^*(\boldsymbol{\theta}_2) - [\alpha\phi_1 + (1-\alpha)\phi_2]^*(\bar{\boldsymbol{\theta}}) = \alpha\phi_1^*(\boldsymbol{\theta}_1) + (1-\alpha)\phi_2^*(\boldsymbol{\theta}_2) - [(\alpha\phi_1)^* \oplus ((1-\alpha)\phi_2)^*](\bar{\boldsymbol{\theta}}) = \alpha\phi_1^*(\boldsymbol{\theta}_1) + (1-\alpha)\phi_2^*(\boldsymbol{\theta}_2) - \left[\min_z (\alpha\phi_1)^*(z) + ((1-\alpha)\phi_2)^*(\alpha\boldsymbol{\theta}_1 + (1-\alpha)\boldsymbol{\theta}_2 - z)\right] \geq 0$ obtained by setting $z = \alpha\boldsymbol{\theta}_1$ □

Corollary 1. *If $\phi(\cdot)$ is convex and $\mathfrak{R}(\cdot)$ is strictly(strongly) convex then the cost function $\inf_\phi \frac{1}{N} D_{\boldsymbol{\phi}}\left(\boldsymbol{y}\middle|\middle|(\nabla\phi)^{-1}(X\boldsymbol{w})\right) + \frac{c_N}{N}\mathfrak{R}(\boldsymbol{w})$ is strictly(strongly) convex in $\boldsymbol{w}$.*

Gradient Descent (GD) [16]	Accelerated GD [16]
Input: $\nabla m\star(\cdot), a, b$ Initialize $\boldsymbol{w}^0, t = 0$. **repeat** $\boldsymbol{w}^{t+1} = w^t - \frac{a}{b+\sqrt{t}}\nabla m\star(\boldsymbol{w}^t)$ **until** Converged	**Input:** $\nabla m\star(\cdot)$, Lipschitz constant l Initialize $\boldsymbol{w}^0, a^0 = 1, t = 0$. **repeat** $\boldsymbol{x}^t = \boldsymbol{w}^t - \frac{1}{l}\nabla m\star(\boldsymbol{w}^t)$ $a^{t+1} = \frac{(1+\sqrt{4(a^t)^2+1})}{2}$ $\boldsymbol{w}^{t+1} = \boldsymbol{x}_i + \frac{a^t-1}{a^{t+1}}(\boldsymbol{x}^t - \boldsymbol{x}^{t-1})$ **until** Converged

Table 1: Accelerated and (un-accelerated) Gradient Descent

Using equation (1) $D_{\boldsymbol{\phi}}\left(\boldsymbol{y}\middle|\middle|(\nabla\phi)^{-1}(X\boldsymbol{w})\right)$ can be represented as $D_{\boldsymbol{\phi}^*}\left(X\boldsymbol{w}\middle|\middle|\nabla\phi(\boldsymbol{y})\right)$. It is also jointly convex in this representation.

Theorem 2. *If $\phi^* \in \mathcal{C}$ then $D_{\boldsymbol{\phi}^*}\left(X\boldsymbol{w}\middle|\middle|\nabla\phi(\boldsymbol{y})\right)$ is jointly convex over ϕ^* and $\boldsymbol{w}$.*

Proof. Follows from essentially same sequence of arguments as used in Theorem 1. □

4 Optimization

Recall that formulation (3) requires optimization over $\boldsymbol{w}$ as well as $\phi(\cdot)$. Block coordinate descent is very popular in machine learning when there are two or more sets of variables that need to be optimized over[3]. However in our setting, naive block coordinate minimization over $\boldsymbol{w}$ and $\boldsymbol{\phi}$ does not readily apply because it is not clear how one may optimize over $\mathcal{C}\star$.

In the forthcoming analysis a prominent role will be played by the **convex marginal functions** $m\star(\boldsymbol{w}) \triangleq \inf_{\phi\in\mathcal{C}\star} D_{\boldsymbol{\phi}}\left(\boldsymbol{y}\middle|\middle|(\nabla\phi)^{-1}(X\boldsymbol{w})\right)$ obtained by taking pointwise infimum[3]. If we can compute the gradient of $m\star(\boldsymbol{w})$ and minimize $\frac{1}{N}m\star(\boldsymbol{w}) + \frac{c_N}{N}\mathfrak{R}(\boldsymbol{w})$ with it, we would achieve our objective (3). The novelty primarily lies in constructing an efficient computational scheme to obtain the gradient that will be referred to as `GradMaPr` . The gradient will be used in an optimization algorithm that is optimal in the black-box first order oracle sense [16] obtaining the convergence rate of $\mathcal{O}(T^{-\frac{\gamma+3}{2\gamma}})$. Since we optimize $m\star(\boldsymbol{w})$ using gradient methods, it is important whether it inherits Hölder smoothness of gradients of (3).

Lemma 1. *If $\phi(\cdot) \in \mathcal{C}^{s,\gamma}$, then the convex marginal function $m(\boldsymbol{w}) = \inf_{\phi\in\mathcal{C}^{s,\gamma}} D_{\boldsymbol{\phi}}\left(\boldsymbol{y}\middle|\middle|(\nabla\phi)^{-1}(X\boldsymbol{w})\right)$ has $(\frac{1}{s}, \frac{1}{\gamma-1})$ Hölder continuous gradient.*

[3]While pointwise supremum preserves convexity, joint convexity (established in Theorem 1) ensures that even the marginal is convex

4.1 GradMaPr : Gradients by Marginalization and Projection

If one can compute a (sub)gradient of $\inf_{\phi\in\mathcal{C}\star} D_{\boldsymbol{\phi}}\big(\boldsymbol{y}\big|\big|(\nabla\phi)^{-1}(X\boldsymbol{w})\big)$, one can optimize functional (3). In spite of the infinite dimensionality, the time complexity of computing the gradient is at most a log factor worse than computing the gradient of a GLM with a known link function which is the linear in N whereas for GradMaPr it is $\mathcal{O}(N\log N)$. Using standard subdifferential calculus [19] we obtain:

$$\partial_{\boldsymbol{\theta}}\min_{\phi\in\mathcal{C}\star} D_{\boldsymbol{\phi}}\big(\boldsymbol{y}\big|\big|(\nabla\phi)^{-1}(\boldsymbol{\theta})\big)\in \operatorname*{ConvHull}_{\phi_*\in\operatorname{Argmin}_{\phi\in\mathcal{C}\star} D_{\boldsymbol{\phi}}\left(\boldsymbol{y}\middle|\middle|(\nabla\phi)^{-1}(\boldsymbol{\theta})\right)} \{(\nabla\phi_*)^{-1}(\boldsymbol{\theta})-\boldsymbol{y}\}. \quad (6)$$

To translate equation (6) word for word into an algorithm would rquire computing the set $\{\phi_*\} = \operatorname{Argmin}_{\phi\in\mathcal{C}\star} D_{\boldsymbol{\phi}}\big(\boldsymbol{y}\big|\big|(\nabla\phi)^{-1}(\boldsymbol{\theta})\big)$ first and then selecting a subgradient from $\operatorname{ConvHull}\{(\nabla\phi_*)^{-1}(\boldsymbol{\theta})-\boldsymbol{y}\}$. The first step is a problem because it is an optimization over the space of functions. The remaining of this section is about how to circumvent this.

4.2 Circumventing the Computation of ϕ_*

In the forthcoming analysis, an important role will be played by the range set $\mathcal{S}\star(\boldsymbol{\theta}) \triangleq \{\boldsymbol{s}|\boldsymbol{s} = (\nabla\phi)^{-1}(\boldsymbol{\theta}), \phi\in\mathcal{C}\star\}$. Every vector $\boldsymbol{s}\in\mathcal{S}\star(\boldsymbol{\theta})$ corresponds to potentially many $\phi\in\mathcal{C}\star$ that satisfies $\boldsymbol{s} = (\nabla\phi)^{-1}(\boldsymbol{\theta})$ each incurring a different loss $D_{\boldsymbol{\phi}}\big(\boldsymbol{y}\big|\big|(\nabla\phi)^{-1}(\boldsymbol{\theta})\big)$. We define the function $M\star : S\times\boldsymbol{\theta}\mapsto\{\mathbb{R}\cup+\infty\}$ using their minimum as

$$M\star(\boldsymbol{s},\boldsymbol{\theta}) \triangleq \min_{\phi\in\mathcal{C}\star|\boldsymbol{s}=(\nabla\phi)^{-1}(\boldsymbol{\theta})} D_{\phi}\big(\boldsymbol{y}\big|\big|\boldsymbol{s}=(\nabla\phi)^{-1}(\boldsymbol{\theta})\big) = \min_{\phi^*\in\mathcal{C}\star|\boldsymbol{s}=\nabla\phi^*(\boldsymbol{\theta})} D_{\boldsymbol{\phi}^*}\big(\boldsymbol{\theta}\big|\big|\nabla\phi\,(\boldsymbol{y})\big) \text{ using (1)} \quad (7)$$

Note, if $\boldsymbol{s}\notin(\nabla\phi)^{-1}(\boldsymbol{\theta})$ then $M\star(\boldsymbol{s},\boldsymbol{\theta}) = +\infty$. Now, note that objective (3) is equivalent to:

$$\min_{\boldsymbol{w},\boldsymbol{s}\in\mathcal{S}\star(\boldsymbol{\theta}),\boldsymbol{\theta},X\boldsymbol{w}=\boldsymbol{\theta}} \frac{1}{N}M\star(\boldsymbol{s},\boldsymbol{\theta}) + \frac{c_N}{N}\mathfrak{R}(\boldsymbol{w}). \quad (8)$$

Although function $\phi(\cdot)$ does not appear in the cost (8), we still have not circumvented the computation of ϕ_* because it is needed to evaluate the function $M\star(\boldsymbol{s},\boldsymbol{\theta})$. Let us establish some properties of $M\star(\boldsymbol{s},\boldsymbol{\theta})$ that will help us remove this obstacle.

Theorem 3. *Function $M\star(\boldsymbol{s},\boldsymbol{\theta})$ is convex in $\boldsymbol{s}\in\mathcal{S}\star$*

Proof. For a fixed $\boldsymbol{\theta}$, consider two points $\boldsymbol{s}_1,\boldsymbol{s}_2$ that correspond to functions ϕ_1^* and ϕ_2^* that achieves the minimum as indicated in (7), incurring the cost $D_{\boldsymbol{\phi}^*}\big(\boldsymbol{\theta}\big|\big|\nabla\phi\,(\boldsymbol{y})\big)$ with the respective functions. Now consider the point $\alpha\boldsymbol{s}_1+(1-\alpha)\boldsymbol{s}_2 = \alpha\nabla\phi_1^*(\boldsymbol{\theta})+(1-\alpha)\nabla\phi_1^*(\boldsymbol{\theta})$ where $\alpha\in[0,1]$. It is clear that it corresponds to the function $\alpha\phi_1^*+(1-\alpha)\phi_2^*$. The cost function $D_{\boldsymbol{\phi}^*}\big(\boldsymbol{\theta}\big|\big|\nabla\phi\,(\boldsymbol{y})\big)$ has already been proved to be jointly convex in Theorem 2. □

Optimizing $M\star(\boldsymbol{s},\boldsymbol{\theta})$: The (sub)gradient $\partial_{\boldsymbol{s}}M\star(\boldsymbol{s},\boldsymbol{\theta})$ of $M\star(\boldsymbol{s},\boldsymbol{\theta})$ is obtained by differentiating (7) as:

$$= \operatorname*{ConvHull}_{\phi_*^*\in\operatorname*{Argmin}_{\phi^*\in\mathcal{C}\star|\boldsymbol{s}=\nabla\phi^*(\boldsymbol{\theta})} D_{\boldsymbol{\phi}^*}\left(\boldsymbol{\theta}\middle|\middle|\nabla\phi(\boldsymbol{y})\right)} ([\nabla^2\phi_*^*])^{-1}(\nabla\phi_*^*(\boldsymbol{\theta})-\boldsymbol{y})$$

$$= \operatorname*{ConvHull}_{\phi_*\in\operatorname*{Argmin}_{\phi\in\mathcal{C}\star|\boldsymbol{s}=(\nabla\phi)^{-1}(\boldsymbol{\theta})} D_{\phi}\left(\boldsymbol{y}\middle|\middle|\boldsymbol{s}\right)} [\nabla^2\phi_*]_{(\boldsymbol{\theta}=\nabla\phi(\boldsymbol{s}))}(\boldsymbol{s}-\boldsymbol{y}) \quad (9)$$

$$\partial_{\boldsymbol{w}}M\star(\boldsymbol{s},\boldsymbol{\theta}) = X^{\dagger}(\boldsymbol{s}-\boldsymbol{y}). \quad (10)$$

The Hessian $[\nabla^2\phi^*]$ is a positive diagonal matrix since ϕ^* is separable and convex. In (8) we have recast (3) as a optimization featuring $M\star(\boldsymbol{s},\boldsymbol{\theta})$, seemingly amenable to (sub)gradient descent in the joint space $(\boldsymbol{s},\boldsymbol{w})$ using (9) and (10). However, we still do not have a computational scheme to identify $\phi(\cdot)_*^*$ that is required to compute $\partial_{\boldsymbol{s}}M\star(\boldsymbol{s},\boldsymbol{\theta})$ numerically.

Descending along Marginalized $M\star(\boldsymbol{s},\boldsymbol{\theta})$:

An alternative is to use a descent method w.r.t. $\boldsymbol{w}$ on the marginal function $\min_{\boldsymbol{s}} M\star(\boldsymbol{s},\boldsymbol{\theta})$ using its gradient by minimizing $M\star(\boldsymbol{s},\boldsymbol{\theta})$ fully for a given $\boldsymbol{w}$. Recall, $M\star(\boldsymbol{s},\boldsymbol{\theta})$ involves a conceptual optimization over $\phi\in\mathcal{C}\star$, now we have to minimize it further over $\boldsymbol{s}$ to obtain $\boldsymbol{s}_*(\boldsymbol{\theta}) = \operatorname{Argmin}_{\boldsymbol{s}} M\star(\boldsymbol{s},\boldsymbol{\theta})$ and then differentiate with respect to $\boldsymbol{\theta}$. The subgradient of the marginal is:

$$\partial_{\boldsymbol{w}} \inf_{\boldsymbol{s}\in\mathcal{S}\star(\boldsymbol{\theta})} M\star(\boldsymbol{s},\boldsymbol{\theta}) = \operatorname{ConvHull}_{\boldsymbol{s}_*(\boldsymbol{\theta})} X^{\dagger}(\boldsymbol{s}_*(\boldsymbol{\theta})-\boldsymbol{y}). \quad (11)$$

Perhaps surprisingly, as we shall show soon (Theorem 4), not only is $\boldsymbol{s}_*(\boldsymbol{\theta})$ unique, it is independent of ϕ_* but also can be computed very efficiently in $\mathcal{O}(N\log N)$ as

$$\boldsymbol{s}_*(\boldsymbol{\theta}) = \operatorname{Argmin}_{\boldsymbol{s}\in\mathcal{S}\star(\boldsymbol{\theta})} \|\boldsymbol{y}-\boldsymbol{s}\|^2. \quad (12)$$

The key steps of GradMaPr remain the same, it consists of marginalization and projection. Different instances of $\mathcal{S}\star$ only change the set to project on.

We will need the following Lemma proved in [1].

Lemma 2. *[1] If the Bregman divergence $D_{\phi}(\cdot\big|\big|\cdot)$ is separable, and $\mathcal{R}\downarrow$ the set of vectors $\boldsymbol{y}$ in $\mathbb{R}^n$ that are in sorted order, that is, $v_i < v_j$ if $i<j$ then the minimizer $\operatorname{Argmin}_{\boldsymbol{y}\in\mathcal{R}\downarrow} D_{\phi}\big(\boldsymbol{x}\big|\big|\boldsymbol{y}\big)$ is independent of ϕ for all $\boldsymbol{x}\in\operatorname{dom}\phi(\cdot)$.*

Corollary 2. *Let M be a positive diagonal matrix that defines the squared Mahalonobis distance, the minimizer $\operatorname{Argmin}_{\boldsymbol{y}\in\mathcal{R}\downarrow}\|\boldsymbol{x}-\boldsymbol{y}\|_M^2$, is independent of the choice of M.*

Theorem 4. *$\operatorname{Argmin}_{\boldsymbol{s}\in\mathcal{S}\star(\boldsymbol{\theta})} M\star(\boldsymbol{s},\boldsymbol{\theta})$ is unique, independent of the minimizing ϕ_*s defined in (7) and obtained as the Euclidean projection of $\boldsymbol{y}$ on $\mathcal{S}\star(\boldsymbol{\theta})$.*

Proof. The KKT conditions of $\min_{\boldsymbol{s}\in\mathcal{S}\star(\boldsymbol{\theta})} M\star(\boldsymbol{s},\boldsymbol{\theta})$ can be obtained from (9) as $\boldsymbol{s}-\boldsymbol{y}\in\left([\nabla^2\phi_*]\right)^{-1}\mathcal{N}_{\mathcal{S}\star(\boldsymbol{\theta})}(\boldsymbol{s})$ and $\boldsymbol{s}\in\mathcal{S}\star(\boldsymbol{\theta})$. The matrix $\left([\nabla^2\phi_*]\right)^{-1}$ is positive definite and diagonal. Now observe that the KKT conditions are exactly the definition of the projection of $\boldsymbol{y}$ on $\mathcal{S}\star(\boldsymbol{\theta})$ according to the squared Mahalonobis distance defined by the matrix $\left([\nabla^2\phi_*]\right)^{-1}$, which according to Corollary 2 is independent of $\left([\nabla^2\phi_*]\right)^{-1}$ if $\mathcal{S}\star(\boldsymbol{\theta})$ has the conic structure of sorted vectors.(This can be obtained from forthcoming Lemma 3 using simple affine change of variables $A\boldsymbol{t}=\frac{1}{s}[A\boldsymbol{\theta}]^{\frac{1}{\gamma}-1}$). Observe that the matrix $\left([\nabla^2\phi_*]\right)^{-1}$ was the only term that depended on a particular ϕ_*. This concludes the proof. □

Corollary 3. *The subgradient defined in* (6) *is*

$$\begin{aligned}\partial_w m(\boldsymbol{w}) &= \partial_w \inf_{\phi\in\mathcal{C}\star} D_{\boldsymbol{\phi}}\left(\boldsymbol{y}\middle|\middle|(\nabla\phi)^{-1}(X\boldsymbol{w})\right)\\ &= X^\dagger\partial_{\boldsymbol{\theta}} \inf_{\phi\in\mathcal{C}\star} D_{\boldsymbol{\phi}}\left(\boldsymbol{y}\middle|\middle|(\nabla\phi)^{-1}(X\boldsymbol{w})\right)\\ &= \underset{\phi_*\in\underset{\phi\in\mathcal{C}\star}{\mathrm{Argmin}} D_{\boldsymbol{\phi}}\left(\boldsymbol{y}\middle|\middle|(\nabla\phi)^{-1}(\boldsymbol{\theta})\right)}{\mathrm{ConvHull}} X^\dagger\{(\nabla\phi_*)^{-1}(\boldsymbol{\theta})-\boldsymbol{y}\}\\ &= X^\dagger(\boldsymbol{s}_*(\boldsymbol{\theta})-\boldsymbol{y}).\end{aligned}$$

4.3 Representing $\mathcal{S}\star(\boldsymbol{\theta})$ by Linear Inequalities

Central to our efficient computation of $\boldsymbol{s}_*(\boldsymbol{\theta})$ via (12) are two algorithmic devices (i) Bregman's algorithm for solving linearly constrained convex optimization problems [5] and (ii) The pool adjacent violators (PAV) algorithm [4]. Both require the representation of the constraints as a set of linear inequalities, whereas the representation of $\mathcal{S}\star(\boldsymbol{\theta})$ described so far does not have that form. In this section we give an alternative characterizations of the sets $\mathcal{S}\star(\boldsymbol{\theta})$ that will enable the use of PAV and Bregman's algorithm. Let A be the adjacent-difference matrix. We define $\mathcal{G}^{s,\gamma}_{L,\nu}(\boldsymbol{\theta})=\{\boldsymbol{s}|\frac{1}{L}[A\boldsymbol{\theta}]^{1+1/\nu}\leq A\boldsymbol{s},\quad A\boldsymbol{s}\leq\frac{1}{s}[A\boldsymbol{\theta}]^{\frac{1}{\gamma-1}}\}$. We use the shorthand $\mathcal{G}\star(\boldsymbol{\theta})$ when appropriate.

Lemma 3. *The set $\mathcal{S}\star(\boldsymbol{\theta})$ is convex (polyhedral) and given by $\pi_{\boldsymbol{\theta}}\mathcal{G}^{s,\gamma}_{L,\nu}(\boldsymbol{\theta})$. where $\pi_{\boldsymbol{\theta}}$ is the inverse of the permutation operator that* stable *sorts $\boldsymbol{\theta}=X\boldsymbol{w}$ in ascending order.*[4]

Corollary 4. $\boldsymbol{s}_*(\boldsymbol{\theta})=\pi_{\boldsymbol{\theta}}\left(\mathrm{Argmin}_{\boldsymbol{v}\in\mathcal{G}\star(\boldsymbol{\theta})}\|\boldsymbol{v}-(\pi_{\boldsymbol{\theta}})^{-1}(\boldsymbol{y})\|^2\right)$

4.4 Convergence of `GradMaPr` in $\mathcal{O}(N\log N)$

The $\mathcal{O}(N\log N)$ convergence of `GradMaPr` is obtained as long as $\|\boldsymbol{y}\|=\mathcal{O}(N)$ and follows as a result of a sequence of isotonic regressions that need to called at most $\mathcal{O}(\log N)$ times. Isotonic regression is solved in time at most $\mathcal{O}(N)$ using the pool adjacent violators (PAV) algorithm. PAV by itself is not sufficient for $\mathcal{O}(N\log N)$ of `GradMaPr` , it depends on Lemma 2.

We proceed by splitting the variable $\boldsymbol{s}$ (and the corresponding inequalities) to obtain $A\boldsymbol{s}_+\leq\frac{1}{s}[A\boldsymbol{\theta}]^{\frac{1}{\gamma-1}}=A\boldsymbol{t}$, $A\boldsymbol{s}_-\leq\frac{1}{l}[A\boldsymbol{\theta}]^{1+1/\nu}=A\boldsymbol{v}$, and $0=\boldsymbol{s}_++\boldsymbol{s}_-$. We write the constraints in a more suggestive form by concatenating the variables as follows: $\begin{pmatrix}\boldsymbol{s}_+\\ \boldsymbol{s}_-\end{pmatrix}$.

$$\begin{pmatrix}A & \mathbf{0}\\ \mathbf{0} & A\end{pmatrix}\begin{pmatrix}\boldsymbol{s}_+\\ \boldsymbol{s}_-\end{pmatrix}\leq\begin{pmatrix}A & \mathbf{0}\\ \mathbf{0} & A\end{pmatrix}\begin{pmatrix}\boldsymbol{t}\\ \boldsymbol{v}\end{pmatrix},\quad \begin{pmatrix}I & I\end{pmatrix}\begin{pmatrix}\boldsymbol{s}_+\\ \boldsymbol{s}_-\end{pmatrix}=0 \tag{13}$$

This induces an equivalent/conformal split on $\boldsymbol{y}$ as $\boldsymbol{y}_-$, $\boldsymbol{y}_+$ and in the cost function as:

$$\min_{\begin{pmatrix}\boldsymbol{s}_+\\ \boldsymbol{s}_-\end{pmatrix}}\left|\left|\begin{pmatrix}\boldsymbol{s}_+\\ \boldsymbol{s}_-\end{pmatrix}-(\pi_{\boldsymbol{\theta}})^{-1}\begin{pmatrix}-\boldsymbol{y}_+\\ \boldsymbol{y}_-\end{pmatrix}\right|\right|^2. \tag{14}$$

Now we apply augmented method of multipliers algorithm to the cost function (14) subject to the constraints (13). Note that the variables $\boldsymbol{s}_+$ and $\boldsymbol{s}_-$ are decoupled in the constraints (13)(a), as well as in the cost function, hence the ADMM updates can be computed in parallel using PAV in linear time. Next we project on the constraint (13)(b) leading to the update

$$\begin{pmatrix}\boldsymbol{s}_+\\ \boldsymbol{s}_-\end{pmatrix}^{t+1}=\mathrm{Avg}\begin{pmatrix}\boldsymbol{s}_+\\ \boldsymbol{s}_-\end{pmatrix}^{t}. \tag{15}$$

Violation of this constraint can be upper bound by $\mathcal{O}(\|\boldsymbol{y}\|)=\mathcal{O}(N)$ at the start of the iteration. Time complexity of $\mathcal{O}(N\log N)$ follows from geometric rate of convergence of ADMM [14].

Note that this algorithm can handle all Hölder continuity constrained isotonic regression. Setting $\nu=1$ we obtain the Lipschitz continuity case.

Vector $\boldsymbol{s}_*$ obtained by `GradMaPr` is used in the optimal gradient algorithm (Table 1) achieving $\mathcal{O}(T^{\frac{3+\gamma}{2\gamma}})$ convergence rate, with each iteration being $\mathcal{O}(N\log N)$.

5 Prediction

In this section we describe how our method may be used for prediction. We restrict our description to the case where $g(\cdot)$ is Lipschitz continuous and strongly monotone. The cases for Holder continuity and Holder convexity are similar. The key idea is to find the smallest interval of training data points $\boldsymbol{\theta}_l$ and $\boldsymbol{\theta}_u$ that encloses $\boldsymbol{\theta}_t=\langle\boldsymbol{w},\boldsymbol{x}\rangle$. Any Lipschitz continuous, strongly monotone curve that lies in the bounding box of $(\boldsymbol{\theta}_l,y_l),(\boldsymbol{\theta}_u,y_u)$ denotes the set permissible prediction for $\boldsymbol{\theta}_l\leq\boldsymbol{\theta}\leq\boldsymbol{\theta}_u$. Lipschitz continuity and strong monotonicity can be seen to induce the linear inequalities (nonlinear for the general Holder case) that further

[4] When the components of $\boldsymbol{\theta}$ are not all unique we form $\mathcal{G}\star(\boldsymbol{\theta})$ by considering the unique components of $\boldsymbol{\theta}$ only and then add equality constraint for every replicated value occurring in $\boldsymbol{\theta}$.

restricts the permissible values within the bounding box. Thus given a $\boldsymbol{\theta}_t = \langle \boldsymbol{w}, \boldsymbol{x} \rangle$ we can identify an interval that satisfy the condition of being permissible predictions. We return the average of that interval.

Recall that the prediction is given by $(\nabla)^{-1}\phi(\langle \boldsymbol{x}, \boldsymbol{w} \rangle)$.

Let $\boldsymbol{w}_*$ be the optimal $\boldsymbol{w}$ returned by the algorithm, let $\theta_t = \langle \boldsymbol{w}_*, \boldsymbol{x} \rangle$, $\theta_l = \max_{\langle X(i), \boldsymbol{w} \rangle \leq \theta_t} \langle X(i), \boldsymbol{w} \rangle$, $\theta_u = \min_{\langle X(i), \boldsymbol{w} \rangle \geq \theta_t} \langle X(i), \boldsymbol{w} \rangle$ and the corresponding y's be y_l, y_u. Then the prediction y corresponding to $\boldsymbol{x}$ is:

$$y \in \begin{cases} [y_l, y_u] \\ [\max(y_l, y_u - L(\theta_u - \theta)), \min(y_u, y_l + s(\theta - \theta_l))] \\ [\max(y_l, y_u - l(\theta_u - \theta), \min(y_u, y_l + l(\theta - \theta_l))] \end{cases}$$

when equation (3) is optimized over corresponding $\mathcal{C}\star$.

Continuity: Note that the prediction function is a continuous point-to-set mapping, where continuity of a point-to-set-map is defined in the usual way [19] as: A point-to-set-map $y(\boldsymbol{x})$ is continuous if for all sequences $\boldsymbol{x}_t \to \boldsymbol{x}$ there exists a $y_t \to y$ such that $y_t \in y(\boldsymbol{x}_t)$.

Recovering $\phi(\cdot)$: Although we cannot recover a unique $\phi(\cdot)$ an instance of it can be recovered up to agreement with the training data. To obtain such an estimate, select a continuous function from the point-to-set mapping $\boldsymbol{x} \mapsto \bar{y}(\boldsymbol{x})$, where we use $\bar{y}$ to indicate a selection. Taking the Legendre dual of the integral of the curve $\boldsymbol{x} \mapsto \bar{y}(\boldsymbol{x})$ obtains a desired $\phi(\cdot)$.

Restricted Output Space: If we have incorporated the restriction on the outputs space in the definition of $\mathcal{G} \star (\boldsymbol{\theta})$ there is little that needs to be done at prediction time. If test $\boldsymbol{x}$ is such that $\langle \boldsymbol{w}_*, \boldsymbol{x} \rangle \in [\min_i \langle \boldsymbol{w}_*, \boldsymbol{x}_i \rangle, \max_i \langle \boldsymbol{w}_*, \boldsymbol{x}_i \rangle]$ nothing needs to be done as the prediction function $y(\boldsymbol{x})$ will automatically guarantee the output space interval constraints. On the other hand if $\langle \boldsymbol{w}_*, \boldsymbol{x} \rangle$ lies outside of the range, thresholding will be necessary.

6 Realizable Case

As a pedagogic shortcut we have motivated the cost function (3) using the notion of a vector $\boldsymbol{u}$ that achieves $\boldsymbol{y} = g(X\boldsymbol{w}) = (\nabla\phi)^{-1}(X\boldsymbol{w})$ exactly. The optimization algorithms presented, however, do not require the existence of such a $\boldsymbol{u}$. If, however, such a *perfect* $\boldsymbol{u}$ exists, *exponentially* more efficient techniques may be applied to recover it.

Observe that the *perfect* $\boldsymbol{u}$ assumption implies the following $\{\exists \phi \in \mathcal{C}\star \text{ s.t. } X\boldsymbol{u} \in \nabla\phi(\boldsymbol{y})\} \equiv \{X\boldsymbol{u} \in (\pi_{\boldsymbol{y}})^{-1}\mathbb{G}\star(\pi_{\boldsymbol{y}}(\boldsymbol{y}))\}$. When the regularization on $\boldsymbol{w}$ is specified using a set $\mathcal{W}$ the vector $\boldsymbol{u}$ can be obtained as the following convex feasibility problem

$$\boldsymbol{\theta} \in \{(\pi_{\boldsymbol{y}})^{-1}\mathbb{G}\star(\pi_{\boldsymbol{y}}(\boldsymbol{y})\} \cap \{X\mathcal{W}\}. \tag{16}$$

Any such convex feasibility problem may be solved by both the sequential as well as the parallel Bregman's algorithm [6]. For our framework, two cases are particularly convenient: (i) $\mathcal{W}$ is an ℓ_2 ball and (ii) $\mathcal{W} = \{\boldsymbol{z} | \ \|\boldsymbol{z}\|^2_{X^\dagger X} \leq L\}$. Choosing the Bregman divergence to be squared Euclidean, we obtain the projection on $\{(\pi_{\boldsymbol{y}})^{-1}\mathbb{G}\star(\pi_{\boldsymbol{y}}(\boldsymbol{y})\}$ in linear time by the PAV algorithm and the projection on the set $\mathcal{W}$ reduces to a regularized least squares in case of (i) and is obtained in closed form for case (ii). Both the solutions can be obtained in time linear in the dimension. In this case we obtain an overall geometric convergence rate $\mathcal{O}(e^{-cT})$ [7], as is the case if we apply ADMM to the same problem [14].

7 Performance Guarantees

We have motivated our cost function (3) assuming $\boldsymbol{y}$ equals $g(X\boldsymbol{u})$ *exactly*. From the GLM view point $g(\cdot)$ is the expectation function. In practice we would only have samples drawn from the conditional distribution. Now we answer how well does our algorithm perform in this setting. Let $\boldsymbol{w}_* = \text{Argmin}_{\phi \in \mathcal{C}\star, \boldsymbol{w}} \frac{1}{N} D_{\boldsymbol{\phi}}\big(\boldsymbol{y} = g(X\boldsymbol{u}) \big|\big| (\nabla\phi)^{-1}(X\boldsymbol{w})\big) + \frac{c_N}{N}\mathfrak{R}(\boldsymbol{w})$ and let $\tilde{\boldsymbol{w}}_* = \text{Argmin}_{\phi \in \mathcal{C}\star, \boldsymbol{w}} \frac{1}{N} D_{\boldsymbol{\phi}}\big(\tilde{\boldsymbol{y}} \big|\big| (\nabla\phi)^{-1}(X\boldsymbol{w})\big) + \frac{c_N}{N}\mathfrak{R}(\boldsymbol{w})$ where $\tilde{\boldsymbol{y}}$ is a perturbation of$\boldsymbol{y}$. First we analyze the deterministic case.

Lemma 4. *Let $\mathbb{R}^n \ni \boldsymbol{y} = g(X\boldsymbol{u})$ with $g \in \{(\nabla\phi)^{-1} | \phi \in \mathcal{C}^{s,\gamma}_{L,\nu}\}$. If expression (7) is minimized over ϕ in the class $\mathcal{C}^{s,\gamma}_{L,\nu}$ then $\frac{1}{N}\|\boldsymbol{u} - \boldsymbol{w}_*\|_{X^\dagger X} \leq \frac{1}{N}\left(Lc_N\mathfrak{R}(\boldsymbol{u})\right)^{\frac{\gamma}{1+\gamma}}$.*

Proof. Optimality and non-negativity of the regularizer yields $D_{\boldsymbol{\phi}}(\boldsymbol{y} || (\nabla\phi)^{-1}(X\boldsymbol{w}_*)) - D_{\boldsymbol{\phi}}(\boldsymbol{y} || (\nabla\phi)^{-1}(X\boldsymbol{u})) \leq c_N(\mathfrak{R}(\boldsymbol{u}) - \mathfrak{R}(\boldsymbol{w}_*)) \leq c_N\mathfrak{R}(\boldsymbol{u})$. Therefore,

$$D_{\boldsymbol{\phi}}\big((\nabla\phi)^{-1}(X\boldsymbol{u}) \big|\big| (\nabla\phi)^{-1}(X\boldsymbol{w}_*)\big) - \langle \boldsymbol{y} - (\nabla\phi)^{-1}(X\boldsymbol{u}), X(\boldsymbol{w}_* - \boldsymbol{u})\rangle$$

$\leq c_N\mathfrak{R}(\boldsymbol{u})$ or,$D^*_\phi\big(X\boldsymbol{w}_* \big|\big| X\boldsymbol{u}\big) \leq c_N\mathfrak{R}(\boldsymbol{u})$ and from Hölder convexity $\frac{1}{L}\|X\boldsymbol{w}_* - X\boldsymbol{u}\|^{1+1/\nu} \leq c_N\mathfrak{R}(\boldsymbol{u})$. □

The result implies that if $c_N = o(N)^{1+1/\nu}$ then $\lim_{N\to\infty} \frac{1}{N}\|\boldsymbol{u} - \boldsymbol{w}_*\|_{X^\dagger X} \to 0$.

Theorem 5. *Let $\mathbb{R}^N \ni \boldsymbol{y} = g(X\boldsymbol{u})$ with $g \in \{(\nabla\phi)^{-1} | \phi \in \mathcal{C}\star\}$ and $\tilde{\boldsymbol{w}}_* = \text{Argmin}_{\phi \in \mathcal{C}\star, \boldsymbol{w}} \frac{1}{N} D_{\boldsymbol{\phi}}\big(\tilde{\boldsymbol{y}} \big|\big| (\nabla\phi)^{-1}(X\boldsymbol{w})\big) + \frac{c_N}{N}\mathfrak{R}(\boldsymbol{w})$. Let $\mathfrak{R}(\cdot)$ be $s_{\mathfrak{R}}(K)$-strongly convex where K is any positive diagonal matrix, then $\frac{1}{N}\|\tilde{\boldsymbol{w}}_* - \boldsymbol{w}_*\|_K \leq \frac{2||\tilde{\boldsymbol{y}} - \boldsymbol{y}||_{XK^{-1}X^\dagger}}{Nc_N s_{\mathfrak{R}}(K)}$.*

Proof. $\boldsymbol{w}_*$ is the stationary point of $\min_{\boldsymbol{s} \in \mathcal{S}\star} \frac{1}{N}M(\boldsymbol{s}, \boldsymbol{w}) + \frac{c_N}{N}\mathfrak{R}(\boldsymbol{w}) = \frac{1}{N}m(\boldsymbol{w}) + \frac{c_N}{N}\mathfrak{R}(\boldsymbol{w})$. We have $\nabla_{\boldsymbol{w}} m(\boldsymbol{w}) = X^\dagger \text{Proj}_{\mathcal{S}\star}(\boldsymbol{y}) - \boldsymbol{y}$. When $\boldsymbol{y}$

is corrupted into $\tilde{\boldsymbol{y}}$ we obtain the corrupted gradient $\nabla_{\boldsymbol{w}}\tilde{m}(\boldsymbol{w})$ as $X^\dagger \operatorname{Proj}_{\mathcal{S}\star}(\tilde{\boldsymbol{y}}) - \tilde{\boldsymbol{y}}$. Let $\tilde{\boldsymbol{w}}_*$ be the stationary point of $\tilde{m}(\boldsymbol{w}) + \mathfrak{R}(\boldsymbol{w})$.

$$\begin{aligned}\|X^\dagger \boldsymbol{y} - X^\dagger \tilde{\boldsymbol{y}}\| &\geq \|X^\dagger \operatorname{Proj}_{\mathcal{S}\star}(\boldsymbol{y}) - X^\dagger \operatorname{Proj}_{\mathcal{S}\star}(\tilde{\boldsymbol{y}})\| \\ &= \|\nabla_{\boldsymbol{w}} m(\boldsymbol{w}_*) - \nabla_{\boldsymbol{w}}\tilde{m}(\boldsymbol{w}_*) + X^\dagger(\boldsymbol{y} - \tilde{\boldsymbol{y}})\| \\ &= \| - \nabla_{\boldsymbol{w}}\tilde{m}(\boldsymbol{w}_*) + X^\dagger(\boldsymbol{y} - \tilde{\boldsymbol{y}})\|.\end{aligned}$$

Squaring and applying Cauchy-Schwarz inequality we obtain $2||\tilde{\boldsymbol{y}} - \boldsymbol{y}||_{XK^{-1}X^\dagger} \geq ||\nabla_{\boldsymbol{\theta}}\tilde{m}(\boldsymbol{w}_*)||_{K^{-1}} \geq c_N s_{\mathfrak{R}}(K)||\tilde{\boldsymbol{w}}_* - \boldsymbol{w}_*||_K$. □

K **Invariance:** One can tighten the bound by choosing K. We emphasize that the algorithm is oblivious to the choice of K, the bound that holds uniformly over positive diagonal because the projection on $\mathcal{S}\star$ is invariant to such choices.

Equipped with this deterministic bounds one may easily obtain performance guarantees for probabilistic settings using standard large deviation results. For example if $g(\cdot)$ is the expectation function of a canonical GLM [15], equivalently: $P(y|\boldsymbol{x}) = e^{\langle \boldsymbol{x},\boldsymbol{u}\rangle y - \phi^*(\langle \boldsymbol{x},\boldsymbol{u}\rangle)}$.

Theorem 6. *If $\boldsymbol{y}$ has probability density $P(y|\boldsymbol{x}) = e^{\langle \boldsymbol{\theta},\boldsymbol{y}\rangle - \phi^*(\boldsymbol{\theta})}$ with an unknown ϕ^* with a negative entropy function uniformly convex with the modulus function $\delta(\cdot)$ with norm $||\cdot||_{K^{-1}}$. Then $P(\frac{1}{N}\|\tilde{\boldsymbol{w}}_* - \boldsymbol{w}_*\|_K \leq t) \geq 1 - \exp(-N\delta(ts(K)))$.*

8 Discussion and Related Work

The `GradMaPr` part of our algorithm shares remarkable similarity with `Isotron` [11] and its descendant `SlIsotron` [10]. Having developed `GradMaPr` unaware of this family of algorithms, the similarity is pleasantly surprising. Our derivation, which follows an entirely different approach, sheds light on many intriguing and open issues about `Isotron` , `SlIsotron` and allows strengthening the original claims. **Objective:** the objective of the `Isotron` family, and therefore their guarantees are quite different. Whereas our objective is parameter recovery, arguably a more demanding task than prediction, the `Isotron` family is motivated by minimizing expected square loss on prediction: a non-convex problem, making their guarantees surprising and impressive. The technicality, length, tediousness of their analysis points to the difficulty of their undertaking. The respective guarantees speak to the different objectives, `Isotron` do not make any claims about recovery (understandable in light of non-convexity)**Method:** The `Isotron` family of updates are not derived from their motivating cost function (expected square loss) but obtained ad-hoc and analyzed in the stochastic setting. Quite strikingly the updates do not even minimize the empirical square loss and its iterates lack convergence guarantees. This forces the authors to evaluate every intermediate iterate w.r.t a held out test set, incurring either runtime cost or a space cost of keeping the entire history of intermediate values. Although the objective of the `SlIsotron` family is to minimize expected square loss, the updates applied can now be recognized as *constant learning rate* gradient updates of a different cost function: marginalized $M\star(\boldsymbol{s},\boldsymbol{\theta})$ which we have shown to be convex. Although suboptimal in its use of first order information (lacks adaptation of learning rate and acceleration whereas our update does the latter), our analysis shows that those updates are convergent. Not only are the updates of `Isotron` , `SlIsotron` special cases of updates derived here, our loss function is a convex upper bound when $g(\cdot)$ is Lipschitz continuous as assumed in [11], this follows from $D_{\boldsymbol{\phi}}\big(\boldsymbol{y}\big|\big|(\nabla\phi)^{-1}(X\boldsymbol{w})\big) \geq \frac{s}{2}\|\boldsymbol{y} - g(X\boldsymbol{w})\|^2$. In light of their guarantees one can see not only is $M\star(\boldsymbol{s},\boldsymbol{\theta})$ an upper bound it is also Fisher consistent, i.e. sequences that minimize it also achieves Baye's error rate. Our paper places `Isotron` variants firmly on the setting of minimizing convex surrogate losses with gradient descent. **Assumptions:** Our analysis assumes $g(\cdot)$ is Hölder continuous which is strictly more general than Lipschitz continuity assumed in `Isotron` , `SlIsotron` . We make very weak assumptions on the distribution generating the samples (we do not admit to any parametric form just membership in the exponential family with Hölder convex negative entropy functions), whereas `Isotron` , `SlIsotron` guarantees are *distribution free*. **Model Selection:** The `SlIsotron` family lacks the traditional notion of a regularizing function or a regularizing set. It achieves model selection by evaluating all intermediate iterates against a held out test set and returns the function with the best test error. The only hyper-parameter is that of how many iterations to perform. In contrast our setup is more traditional with a regularizing function, the coefficient of which is determined by cross validation. **Guarantees:** The performance guarantees are not comparable across the two approaches as their scopes are different. On the statistical side `Isotron` family comes with distribution free guarantees on expected square loss, whereas our method come with recovery guarantees under very mild statistical assumptions. On the optimization side, for the non-agnostic setting our convergence guarantees are exponentially faster than `Isotron` , `SlIsotron` . For the non-realizable case our update match the best black-box bound possible for a first-order method.

9 Empirical Performance

In this section we compare preliminary empirical performance of our proposed technique with that of

`SlIsotron` on identical prediction tasks. `SlIsotron` is the most advanced algorithm that has comparable capabilities.

First, we discuss some implementation detail. Both our algorithm and `SlIsotron` have $\mathcal{O}(N \log N)$ complexity per iteration to solve the Lipschitz isotonic regression. `SlIsotron` , requires an intricate tree data structure. We use our ADMM variant in Lipschitz isotonic regression in both the methods for simplicity of implementation. This can only affect the running time not the quality of the results. This, however, rules out comparing the runtime of the algorithms and is disadvantageous to our proposed algorithm because we expect it to be faster as it uses accelerated gradient descent whereas `SlIsotron` implicitly uses gradient descent with constant learning rate.

For prediction problems the sources of difference between `SlIsotron` and our algorithm are: (i) different methods for estimating the hyperparameters and (ii) different interpolation strategy used to generate prediction between two training samples. Our algorithm uses the average of the predicted interval as described in Section 5. `SlIsotron` uses linear interpolation. For `SlIsotron` we set the max number of iterations to 100 and set our hyperparameter c_N using 10 fold cross-validation and grid search.

Following [10] we compared RMSE achieved on the UCI datasets: `communities, concrete, housing, parkinsons, winequality`. Note RMSE is not fair to our method because unlike `SlIsotron` it is not designed to minimize RMSE. Although our method performed better on average compared to `SlIsotron` and logistic regression, the difference was not statistically significant at 95% confidence level. One reason that the algorithms are statistically indistinguishable may be because RMSE is a poor choice or this. Lack of dynamic range in RMSE has also been observed in matrix factorization tasks. The empirical result suggests that model selection process used by `SlIsotron` and our method are comparable as are the different interpolation strategies. The different interpolation strategy could yield significantly different predictions only on low density regions of the domain where the nearest neighbor is far enough that our methods returns a large interval. Although no statistical superiority over `SlIsotron` was observed, the fact that our algorithm could match logistic regression in performance should be taken as a positive result.

The next real world prediction task that we pick is learning to rank with pointwise methods. These methods predict the relevance score of an item given its feature vector representation $\boldsymbol{x}_i$ and then orders the items according to the predicted score. Ravikumar et. al. [18] showed that canonical GLM losses with an appropriately scaled relevance scores are the only statistically consistent losses for NDCG [9] a de facto popular metric used to measure rank quality. The predicted score is obtained as $g(\langle \boldsymbol{x}_i, \boldsymbol{w} \rangle$ where g is the inverse link function, hence monotonic. The order among the items, however, is entirely defined by $\langle \boldsymbol{x}_i, \boldsymbol{w} \rangle$, so predicting the relevance score is un-necessary. However for different datasets different GLMS might be better suited. This makes it a appropriate real world test bed for parameter recovery. We evaluated the quality of ranking achieved by (i) linear regression, (ii) logistic regression, `SlIsotron` and our method on the LETOR 4.0 datasets [13], see Figure 1 .

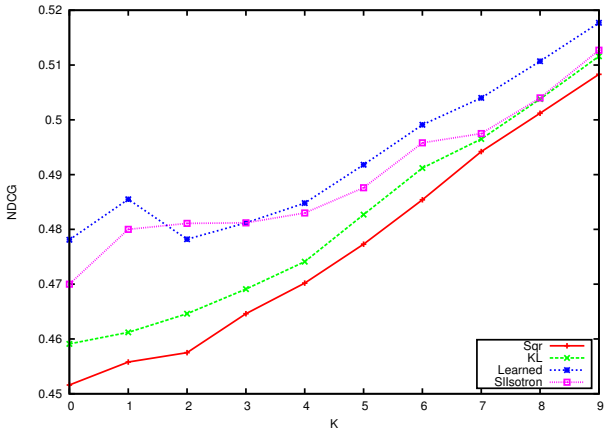

Figure 1: NDCG at different truncations K achieved by (i) linear, (ii) logistic, (iii) `SlIsotron` and (iv) our method on MQ2007

10 Conclusion

This paper proposes a novel method of learning finite dimensional parameters of a generalized linear model whose link function is unknown. We commit neither to a parametric form of conditional expectation function nor to any parametric form of the distribution generating the samples. The parameters are learned by minimizing a matching Bregman divergence simultaneously over all Hölder convex functions as well as the parameters. Remarkably, not only can the global minimum be found, the computational cost per iteration is only a log factor worse in the number of observations as compared to the case where the link function is known. The convergence rates are the best possible for the first order black-box model whose use is justified because we do not know the convex function.

Acknowledgement: Authors acknowledge NSF grant IIS-1421729.

References

[1] S. Acharyya, O. Koyejo, and J. Ghosh. Learning to rank with Bregman divergences and monotone retargeting. In *Uncertainty in Artificial Intelligence, UAI*, 2012.

[2] Peter Auer, Mark Herbster, and Manfred K. Warmuth. Exponentially many local minima for single neurons. In *NIPS*, pages 316–322, 1995.

[3] Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

[4] Michael J. Best and Nilotpal Chakravarti. Active set algorithms for isotonic regression; a unifying framework. *Mathematical Programming*, 47:425–439, 1990.

[5] L. M. Bregman. The relaxation method of finding the common points of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7:200–217, 1967.

[6] Y. Censor and T. Elfving. A multiprojection algorithm using Bregman projections in a product space. *Numerical Algorithms*, 8:221–239, 1994.

[7] Frank Deutsch and Hein Hundal. The rate of convergence for the cyclic projections algorithm I. *Journal of Approximation Theory*, 142:36–55, 2006.

[8] O. Devolder, F. Glineur, and Yu. Nesterov. First order methods of smooth convex optimization with inexact oracle. In *CORE Discussion Paper 2011/2*, 2011.

[9] Kalervo Järvelin and Jaana Kekäläinen. IR evaluation methods for retrieving highly relevant documents. In *23rd ACM SIGIR conference on Research and development in information retrieval*, SIGIR '00, pages 41–48, 2000.

[10] Sham Kakade, Adam Kalai, Varun Kanade, and Ohad Shamir. Efficient learning of generalized and single index models with isotonic regression. In *ARXIV:1104.2018v1 [cs.AI]*, 2011.

[11] Adam Kalai and Ravi Sastry. The isotron algorithm: High-dimensional isotonic regression. In *COLT*, 2009.

[12] E. H. Lehmann. *Theory of Point Estimation*. John Wiley & Sons, 1983.

[13] Tie-Yan Liu, Jun Xu, Tao Qin, Wenying Xiong, and Hang Li. LETOR: Benchmark dataset for research on learning to rank for information retrieval. In *SIGIR 2007 Workshop on Learning to Rank for Information Retrieval*, 2007.

[14] Zhi-Quan Luo. On the linear convergence of the alternatind direction method of multipliers. In *ARXIV:1208.3922v1[math.OC]*, 2012.

[15] C. E. McCulloch and S. R. Searle. *Generalized Linear and Mixed Models*. John Wiley & Sons, 2001.

[16] Arkadi Nemirovski. *Lectures on modern convex optimization*. Society for Industrial and Applied Mathematics, 2001.

[17] Yu. Nesterov. Universal gradient methods for convex optimization problems. In *CORE Discussion Paper 2013/26*, 2013.

[18] Pradeep Ravikumar, Ambuj Tewari, and Eunho Yang. On NDCG consistency of listwise ranking methods. In *Proceedings of 14th International Conference on Artificial Intelligence and Statistics*, AISTATS, 2011.

[19] R T. Rockafellar. *Convex Analysis (Princeton Landmarks in Mathematics and Physics)*. Princeton University Press, December 1996.