Proceedings of Machine Learning Research vol 117:1–23, 2020 31st International Conference on Algorithmic Learning Theory

Exponentiated Gradient ~~vs.~~ Meets Gradient Descent

Udaya Ghai[†] UGHAI@CS.PRINCETON.EDU
Princeton University
Elad Hazan EHAZAN@CS.PRINCETON.EDU
Google AI Princeton & Princeton University
Yoram Singer[†] Y.S@CS.PRINCETON.EDU
Princeton University

Editors: Aryeh Kontorovich and Gergely Neu

Abstract

The (stochastic) gradient descent and the multiplicative update method are probably the most popular algorithms in machine learning. We introduce and study a new regularization which provides a unification of the additive and multiplicative updates. This regularization is derived from an hyperbolic analogue of the entropy function, which we call hypentropy. It is motivated by a natural extension of the multiplicative update to negative numbers. The hypentropy has a natural spectral counterpart which we use to derive a family of matrix-based updates that bridge gradient methods and the multiplicative method for matrices. While the latter is only applicable to positive semidefinite matrices, the spectral hypentropy method can naturally be used with general rectangular matrices. We analyze the new family of updates by deriving tight regret bounds. We study empirically the applicability of the new update for settings such as multiclass learning, in which the parameters constitute a general rectangular matrix.

Keywords: Online Convex Optimization, Gradient Descent, Exponentiated Gradient, Experimentation

1. Introduction

Algorithms for online learning can morally be divided into two camps. On one side is the additive gradient update. Additive gradient-based stochastic methods are the most commonly used approach for learning the parameters of shallow and deep models alike. On the other side stands the multiplicative update method. It is somewhat less glamorous, nonetheless a fundamental primitive in game theory and machine learning, and was rediscovered repeatedly in a variety of algorithmic settings (Arora et al., 2012). Both additive and multiplicative updates can be seen as special cases of a more general technique of learning with *regularization*. General frameworks for regularization were developed in online learning, dubbed Follow-The-Regularized-Leader and in convex optimization as the *Mirrored Decent* algorithms, see more below.

Notable attempts were made to unify different regularization techniques, in particular between the multiplicative and additive update methods (Kivinen and Warmuth, 1997). For example, AdaGrad (Duchi et al., 2011) stemmed from a theoretical study of learning the best regularization in hindsight. As the name implies, the p-norm update (Grove et al., 2001; Gentile, 2003) uses the squared p-norm of the parameters as a regularization. By varying the order of the norm between regret bounds that are characteristic of additive and multiplicative updates.

We study a new, arguably more natural, family of regularization which "interpolates" between additive and multiplicative forms. We analyze its performance both experimentally, and theoretically to obtain tight regret bounds in the online learning paradigm. The motivation for this interpolation stems from the extension of the multiplicative update to negative weights. Instead of using the so called "EG$\pm$ trick", a term coined by Warmuth, which simulates arbitrary weights through duplication to positive and negative components, we use a direct approach. To do so we introduce the hyperbolic regularization with a single temperature-like hyperparameter. Varying the hyperparameter yields regret bounds that translate between those akin to additive, and multiplicative, update rules. This regularization appears briefly in Cesa-Bianchi and Lugosi (2006).

As a natural next step, we investigate the spectral analogue of the hypentropy function. We show that the spectral hypentropy is strongly-convex with respect to the Euclidean or trace norms, again as a function of the single interpolation parameter. The spectral hypentropy yields updates that can be viewed as interpolation between gradient descent rule and matrix multiplicative update (Tsuda et al., 2005; Arora and Kale, 2007).

The standard matrix multiplicative update rule applies only to positive semi-definite matrices. Standard extensions to square and more general matrices increase the dimensionality (Hazan et al., 2012). Moreover, the the regret bounds scale as $O(\sqrt{T\log(m+n)}$ for $m \times n$ matrices. In contrast, the spectral hypentropy regularization is defined for arbitrary, rectangular, matrices. Moreover, the hypentropy-based update in better regret bounds of $O(\sqrt{T \log \min\{m,n\}})$, matching the best known bounds in Kakade et al. (2012).

Related work For background on the multiplicative updates method and its use in machine learning and algorithmic design, see Arora et al. (2012). The matrix version of multiplicative updates method was proposed in Tsuda et al. (2005) and later in Arora and Kale (2007). The study of the interplay between additive and multiplicative updates was initiated in the influential paper of Kivinen and Warmuth (1997). Generalizations of multiplicative updates to negative weights were studied in the context of the Winnow algorithm and mistake bounds in Warmuth (2007); Grove et al. (2001). The latter paper also introduced the p-norm algorithm which was further developed in Gentile (2003). The generalization of the p-norm regularization to matrices was studied in Kakade et al. (2012).

2. Problem Setting

Notation. Vectors are denoted by bold-face letters, e.g. $\mathbf{w}$. The zero vector and the all ones vector are denoted by $\mathbf{0}$ and $\mathbf{1}$ respectively. We denote a ball of radius 1 with respect to the p-norm in $\mathbb{R}^d$ as $B_p = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|_p \leq 1\}$. For simplicity of the presentation in the sequel we assume that weights are confined to the unit ball. Our results generalize straightforwardly to arbitrary radii.

Matrices are denoted by capitalized bold-face letters, e.g. $\mathbf{X}$. We denote the space of real matrices of size $m \times n$ as $\mathbb{R}^{m\times n}$ and symmetric matrices of size $d \times d$ as $\mathbb{S}^d$. For a matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$, we denote the vector of singular values $\sigma(\mathbf{X}) = (\sigma_1, \sigma_2, \ldots, \sigma_l)$ where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_l \geq 0$ and $l = \min\{m, n\}$. Analogously, for $\mathbf{X} \in \mathbb{S}^d$, we denote the vector of its eigenvalues as $\lambda(\mathbf{X}) =$

$(\lambda_1, \lambda_2, \ldots, \lambda_d)$. We use $\|\mathbf{X}\|_p \stackrel{\text{def}}{=} \|\sigma(\mathbf{X})\|_p$ to represent the Schatten p-norm of a matrix, namely, the p-norm of the vector of singular values. We refer to the Schatten norm for $p = 1$ as the trace-norm. Note that the notation of the spectral norm of $\mathbf{X}$ is $\|\mathbf{X}\|_\infty$. We denote the ball of radius τ with respect to the trace-norm as $B_{\text{Tr}}(\tau) = \{\mathbf{X} \in \mathbb{R}^{m\times n} : \|\mathbf{X}\|_1 \leq \tau\}$. We also define the intersection of a ball and the positive orthant as $B_p^+ = B_p \cap \mathbb{R}^d_+$. We use $(x)_+$ to denote $\max(x, 0)$. We denote by $\|\mathbf{x}\|_*$ the dual norm of $\mathbf{x}$, $\|\mathbf{x}\|_* \stackrel{\text{def}}{=} \sup\{\mathbf{z}^\top \mathbf{x} \,|\, \|\mathbf{z}\| \leq 1\}$.

Online Convex Optimization. In online convex optimization (Cesa-Bianchi and Lugosi, 2006; Hazan, 2016; Shalev-Shwartz, 2012), a learner iteratively chooses a vector from a convex set $\mathcal{K} \subset \mathbb{R}^d$. We denote the total number of rounds as T. In each round, the learner commits to a choice $\mathbf{w}_t \in \mathcal{K}$. After committing to this choice, a convex loss function $\ell_t : \mathcal{K} \to \mathbb{R}$ is revealed and the learner incurs a loss $\ell_t(\mathbf{w}_t)$. The most common performance objective of an online learning algorithm $\mathcal{A}$ is regret. Regret is defined to be the total loss incurred by the algorithm with respect to the loss of the best fixed single prediction found in hindsight. Formally, the regret of a learning algorithm $\mathcal{A}$ is defined as,

$$\mathcal{R}_T(\mathcal{A}) \stackrel{\text{def}}{=} \sup_{\ell_1 \ldots \ell_t} \left\{ \sum_{t=1}^T \ell_t(\mathbf{w}_t) - \min_{\mathbf{w}^* \in \mathcal{K}} \sum_{t=1}^T \ell_t(\mathbf{w}^*) \right\}.$$

3. Main Results

We begin by defining the β-hyperbolic entropy, denoted ϕ_β.

Definition 1 (Hyperbolic-Entropy) *For all $\beta > 0$, let $\phi_\beta : \mathbb{R}^d \to \mathbb{R}$ be defined as,*

$$\phi_\beta(\mathbf{x}) = \sum_{i=1}^d \left(x_i \operatorname{arcsinh}\left(\frac{x_i}{\beta}\right) - \sqrt{x_i^2 + \beta^2} \right).$$

For brevity and clarity, we use the shorthand hypentropy for ϕ_β. Given the hypentropy function, we derive its associated Bregman divergence, the relative hypentropy as,

$$\begin{aligned} D_\phi^\beta(\mathbf{x} \,\|\, \mathbf{y}) &= \phi_\beta(\mathbf{x}) - \phi_\beta(\mathbf{y}) - \langle \nabla\phi_\beta(\mathbf{y}), \mathbf{x} - \mathbf{y}\rangle \\ &= \sum_{i=1}^d \left[x_i(\operatorname{arcsinh}\left(\frac{x_i}{\beta}\right) - \operatorname{arcsinh}\left(\frac{y_i}{\beta}\right)) - \sqrt{x_i^2+\beta^2} + \sqrt{y_i^2+\beta^2} \right]. \end{aligned}$$

We next describe an OCO algorithm, Hypentropy Update (HU), which is mirror descent with respect to this divergence.

Algorithm 1: Hypentropy (HU)

Input: $\eta > 0, \beta > 0$, convex subset $\mathcal{K} \subseteq \mathbb{R}^d$
Initialize weight matrix $\mathbf{w}^1 = \mathbf{0}$;
for $i = 1$ **to** T **do**
    (a) Predict $\mathbf{w}^t$ (b) Incur loss $\ell_t(\mathbf{w}_t)$ (c) Calculate $\mathbf{g}^t = \nabla \ell_t(\mathbf{w}^t)$;
    Update: $\mathbf{w}^{t+\frac{1}{2}} = \beta \sinh\left(\operatorname{arcsinh}\left(\frac{\mathbf{w}^t}{\beta}\right) - \eta \mathbf{g}^t\right)$;
    Project onto $\mathcal{K}$: $\mathbf{w}^{t+1} = \arg\min_{\mathbf{v} \in \mathcal{K}} D_\phi^\beta(\mathbf{v} \parallel \mathbf{w}^{t+\frac{1}{2}})$
end

For the specific cases where our constraint set $\mathcal{K}$ is the 1-norm or 2-norm ball, we prove the following regret bounds.

Theorem 2 *Let $\mathbf{w} \in B_2$ and assume that for all t, $\|\mathbf{g}_t\|_2 \le G_2$. Setting for $\beta \ge 1$,*

$$\eta = \frac{1}{G_2}\sqrt{\frac{1}{\beta(\beta+1)T}}\,, \;\; \textit{yields} \quad \mathcal{R}_T(\mathrm{HU}) \le 4G_2\sqrt{T}\,.$$

Theorem 3 *Let $\mathbf{w} \in B_1$ and assume that for all t, $\|\mathbf{g}_t\|_\infty \le G_\infty$. Setting for $\beta \le 1$*

$$\eta = \frac{1}{G_\infty}\sqrt{\frac{\log\left(\frac{3}{\beta}\right)}{2T(1+\beta d)}}\,, \;\; \textit{yields} \quad \mathcal{R}_T(\mathrm{HU}) \le 3G_\infty\sqrt{T(1+\beta d)\log\left(\tfrac{3}{\beta}\right)}\,.$$

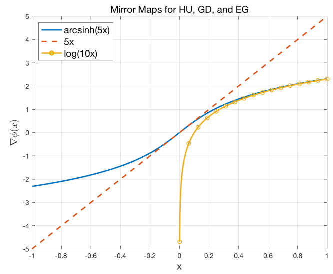
Note that with $\beta > 1$, we have $O(G_2\sqrt{T})$ regret like GD while for $\beta = \Theta(1/d)$, regret is $O(G_\infty\sqrt{T\log d})$ like EG. We also analyze a version of HU, called SHU, for closed convex sets of matrices in $\mathbb{R}^{m\times n}$. The pseudocode of SHU is provided in Algorithm 2. We discuss the spectral hypentropy, $D_\Phi^\beta(\cdot,\cdot)$ in Sec. 5.

Algorithm 2: Spectral Hypentropy (SHU)

Input: $\eta > 0, \beta > 0$, convex domain of matrices $\mathcal{K} \subseteq \mathbb{R}^{m\times n}$
Initialize weight matrix $\mathbf{W}^1 = \mathbf{0}$;
for $i = 1$ **to** T **do**
    (a) Predict $\mathbf{W}^t$ (b) Incur loss $\ell_t(\mathbf{W}_t)$ (c) Calculate $\mathbf{G}^t = \nabla \ell_t(\mathbf{W}^t)$;
    Update: $\mathbf{W}^{t+\frac{1}{2}} = \beta \sinh\left(\operatorname{arcsinh}\left(\frac{\mathbf{W}^t}{\beta}\right) - \eta \mathbf{G}^t\right)$;
    Project onto $\mathcal{K}$: $\mathbf{W}^{t+1} = \arg\min_{\mathbf{V} \in \mathcal{K}} D_\Phi^\beta(\mathbf{V} \parallel \mathbf{W}^{t+\frac{1}{2}})$
end

The update step of SHU requires a spectral decomposition. We define $f(\mathbf{A}) = \mathbf{U} f(\mathrm{diag}\,[\sigma(\mathbf{A})])\mathbf{V}^\top$ where $\mathbf{A} = \mathbf{U}\,\mathrm{diag}(\sigma(\mathbf{A}))\mathbf{V}^\top$ is the singular value decomposition of $\mathbf{A}$. We use this definition twice, once with $f = \operatorname{arcsinh}$, and after subtracting the gradient with $f = \sinh$. We prove the following regret bound for SHU.

	Square	Entropy	Hypentropy
$\phi(x)$	$\frac{1}{2}x^2$	$x\log(x) - x$	$x \operatorname{arcsinh}(\frac{x}{\beta}) - \sqrt{x^2+\beta^2}$
$\nabla\phi(x)$	x	$\log(x)$	$\operatorname{arcsinh}(\frac{x}{\beta})$
$\nabla^2\phi(x)$	1	$\frac{1}{x}$	$\frac{1}{\sqrt{x^2+\beta^2}}$



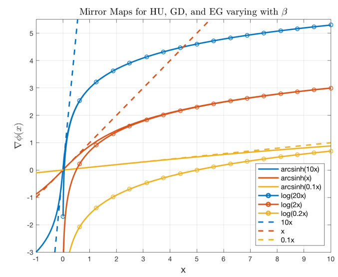


Figure 1: Left:Classical and new divergences. Center: scalar versions of mirror maps used in HU($\beta = \frac{1}{5}$), GD, and EG Algorithms depicted in blue, red, and yellow respectively. EG is only defined for positive values of x. Right: β is varied between $0.1, 1$, and 10 and is similarly depicted along with its linear and logarithmic limits.

Theorem 4 (Trace norm Regret) *Let* $\mathbf{W} \in B_{\mathrm{Tr}}(\tau) \subseteq \mathbb{R}^{m\times n}$ *and let* $\|\mathbf{G}^t\|_\infty \leq G_\infty$ *be a spectral norm bound on the gradients. For* $\gamma = \frac{\beta}{\tau} \leq 1$*, setting,*

$$\eta = \frac{1}{2G_\infty}\sqrt{\frac{\log\left(\frac{3}{\gamma}\right)}{T(1+\gamma\min\{m,n\})}}\,, \quad \textit{yields} \quad \mathcal{R}_T(\mathrm{SHU}) \leq 4\tau G_\infty\sqrt{T(1+\gamma\min\{m,n\})\log\left(\tfrac{3}{\gamma}\right)}\,.$$

HU and SHU are mirror descent algorithms using the hypentropy and spectral hypentropy regularization functions defined in Sec. 4 and Sec. 5 respectively. These sections explore the geometric properties of these new regularization functions. Connections to the EG$\pm$ doubling trick are explored in Sec. 6. The regret theorems above are then proved in App. C. Experimental results which underscore the applicability of HU and SHU are described in App. A. A thorough description of mirror descent is given for completeness in App. B.

4. Hypentropy Divergence

In this section, we explore the β-hyperbolic entropy. We first note that we can view $\phi_\beta(\mathbf{x})$ as the sum of scalar functions, $\phi_\beta(\mathbf{x}) = \sum_{i=1}^d \phi_\beta(x_i)$, each of which satisfies,

$$\phi''_\beta(x) = \frac{1}{\sqrt{x^2+\beta^2}}\,. \tag{1}$$

As we vary β, the relative hypentropy interpolates between the squared Euclidean distance and the relative entropy. The potentials for these divergences are sums of element-wise scalar functions, for simplicity we view them as scalar functions. The interpolation properties of hypentropy can be seen in Figure 4. As β approaches 0, we see that $\nabla^2\phi_\beta$ approaches $\frac{1}{|x|}$. When working only over the positive orthant, as is the case with entropic regularization, the hypentropy second derivative converges to the second derivative of the negative entropy. On the other hand, as β grows much larger than x, we see $\sqrt{\beta^2+x^2} \approx \beta$. Therefore, for larger β, $\nabla^2\phi_\beta$ is essentially a constant. In this regime hypentropy behaves like a scaled squared euclidean distance.

From a mirror descent perspective (see App. B), it makes sense to look at the mirror map, the gradient of the ϕ which defines the dual space where additive updates take place. Weights are mapped into the dual space via the mirror map $\nabla\phi : \mathbb{R}^d \to \mathbb{R}^d$ and mapped back into the primal space via $\nabla\phi^*$. Gradient-Descent (GD) can be framed as mirror descent using the squared euclidean norm potential while Exponentiated-Gradient (EG) amounts to mirror descent using the entropy potential. The Hypentropy Update (HU) uses the mirror map $\nabla\phi_\beta$. As can be seen from Figure 4, for sufficiently large weights, the hypentropy mirror map behaves like $\log(x)$, namely, the EG mirror map. In contrast, $\text{arcsinh}(x/\beta)$ is linear for small weights, and thus behave like GD[1]. Large values of β correspond to a slower transition from the linear regime to the logarithmic regime of the mirror map.

The regret analysis of GD and EG depend on geometric properties of the divergence related to the 2-norm and 1-norm respectively. Given this connection, it is useful to analyze the properties of hypentropy with respect to both the 1-norm and 2-norm. Recall that a function f is α-strongly convex with respect to a norm $\|\cdot\|$ on $\mathcal{K}$ if,

$$\forall \mathbf{x}, \mathbf{y} \in \mathcal{K}, f(\mathbf{x}) - f(\mathbf{y}) - \nabla f(\mathbf{y})(\mathbf{x} - \mathbf{y}) \geq \frac{\alpha}{2}\|\mathbf{x} - \mathbf{y}\|^2.$$

For convenience, we use the following second order characterization of strong-convexity from Thm. 3 in (Yu, 2013).

Lemma 5 *Let $\mathcal{K}$ be a convex subset of some finite vector space $\mathcal{X}$. A twice differentiable function $f : \mathcal{K} \to \mathbb{R}$ is α-strongly convex with respect to a norm $\|\cdot\|$ iff*

$$\inf_{\mathbf{x}\in\mathcal{K}, \mathbf{y}\in\mathcal{X}: \|\mathbf{y}\|=1} \mathbf{y}^\top \nabla^2\phi(\mathbf{x})\, \mathbf{y} \geq \alpha \, .$$

We next prove elementary properties of ϕ_β.

Lemma 6 *The function ϕ_β is $(1+\beta)^{-1}$-strongly-convex over B_2 w.r.t the 2-norm.*

Proof To prove the first part, note that from (1) we get that the Hessian is the diagonal matrix,

$$\nabla^2\phi_\beta(\mathbf{x}) = \text{diag}\left[\tfrac{1}{\sqrt{x_1^2+\beta^2}}, \ldots \tfrac{1}{\sqrt{x_d^2+\beta^2}}\right]$$

Strong convexity follows from the diagonal structure of the Hessian, whose smallest eigenvalue is

$$\frac{1}{\sqrt{x^2+\beta^2}} \geq \frac{1}{\sqrt{1+\beta^2}} \geq \frac{1}{1+\beta} \, .$$

■

Lemma 7 *The function ϕ_β is $(1+\beta d)^{-1}$-strongly-convex over B_1 w.r.t. the 1-norm.*

Proof We work with the characterization provided in Lemma 5,

$$\begin{aligned}\inf_{\mathbf{x}\in B_1, \|\mathbf{y}\|_1=1} \mathbf{y}^\top\nabla^2\phi(\mathbf{x})\mathbf{y} &= \inf_{\mathbf{x}\in B_1, \|\mathbf{y}\|_1=1} \sum_{i=1}^d \frac{y_i^2}{\sqrt{\beta^2+x_i^2}} && \big[\text{Equation (1)}\big] \\ &= \inf_{\mathbf{x}\in B_1, \|\mathbf{y}\|_1=1} \frac{1}{\sum_{i=1}^d \sqrt{\beta^2+x_i^2}} \left(\sum_{i=1}^d \frac{y_i^2}{\sqrt{\beta^2+x_i^2}}\right)\left(\sum_{i=1}^d \sqrt{\beta^2+x_i^2}\right)\end{aligned}$$

1. If scaled by an appropriate factor, like $(1+\beta)$, the limits as β tends to 0 and ∞ are exactly EG and GD respectively.

$$\geq \inf_{\mathbf{x}\in B_1, \|\mathbf{y}\|_1=1} \frac{1}{\sum_{i=1}^d \sqrt{\beta^2 + x_i^2}} \left(\sum_{i=1}^d \sqrt{y_i^2} \right)^2 \qquad \text{[Cauchy- Schwarz]}$$

$$= \inf_{\mathbf{x}\in B_1, \|\mathbf{y}\|_1=1} \frac{1}{\sum_{i=1}^d \sqrt{\beta^2 + x_i^2}} \|\mathbf{y}\|_1^2 \geq \frac{1}{\sum_{i=1}^d (\beta + |x_i|)} \geq \frac{1}{1+\beta d} \; .$$

■

We next introduce a generalized notion of diameter with respect to 0 and use it to prove properties of ϕ_β.

Definition 8 *The diameter of a convex set $\mathcal{K}$ that contains 0 with respect to ϕ is,* $\operatorname{diam}_\phi(\mathcal{K}) \stackrel{def}{=} \sup_{\mathbf{x}\in\mathcal{K}} D_\phi(\mathbf{x} \,\|\, \mathbf{0})$.

In App. D, we derive the following diameter bounds:

$$\operatorname{diam}(B_1) \leq \log\left(\frac{3}{\beta}\right), \forall \beta \leq 1 \tag{2}$$

$$\operatorname{diam}(B_2) \leq \frac{2}{\beta} \tag{3}$$

5. Spectral Hyperbolic Divergence

In this section, the focus is on using hypentropy as a spectral regularization function. We show that the matrix version of HU is strongly convex with respect to the trace norm. Our proof technique of strong convexity is a roundabout for the matrix potential. The proof works by showing that the conjugate potential function is smooth with respect to the spectral norm (the dual of the trace norm). The duality of smoothness and strong convexity is then used to show strong convexity.

5.1. Matrix Functions

We are concerned with potential functions that act on the singular values of a matrix. For an even scalar absolutely symmetric function, $f : \mathbb{R} \to \mathbb{R}^+$, consider the trace function,

$$F(\mathbf{X}) = (f \circ \sigma)(\mathbf{X}) = \sum_{i=1}^d f(\sigma_i) = \operatorname{Tr}\left(f\left(\sqrt{\mathbf{X}^\top \mathbf{X}} \right) \right) \;, \tag{4}$$

where we overload the notation for f and denote $f(\mathbf{v}) = \sum_{i=1}^d f(v_i)$. For $\mathbf{X} \in \mathbb{S}^d$ we use

$$F(\mathbf{X}) = (f \circ \lambda)(\mathbf{X}) = \operatorname{Tr}(f(\mathbf{X})) \; . \tag{5}$$

Here $f(\mathbf{X})$ represents the standard lifting of a scalar function to a square matrix, where f acts on the vector of eigenvalues, namely,

$$\mathbf{X} = \mathbf{U} \operatorname{diag}[\lambda(\mathbf{X})] \mathbf{U}^\top \quad \Rightarrow \quad f(\mathbf{X}) = \mathbf{U} \operatorname{diag}[f(\lambda(\mathbf{X}))] \mathbf{U}^\top \; .$$

We also use the gradient of a trace-function in our analysis. The following result from Thm. 14 in (Kakade et al., 2009) shows how to compute a gradient using a singular value decomposition.

Theorem 9 *Let $\mathbf{X} \in \mathbb{R}^{m\times n}$ and $F : \mathbb{R}^{m\times n} \to \mathbb{R}^+$ be defined as above, then*

$$\nabla F(\mathbf{X}) = f'(\mathbf{X}) \,.$$

We also make use of the *Fenchel* conjugate functions. Consider a convex function $f : \mathcal{X} \to \mathbb{R}$ defined on a *finite* vector space $\mathcal{X}$ endowed with an inner product $\langle \cdot, \cdot \rangle$. The conjugate of f is defined as follows.

Definition 10 *The conjugate $f^* : \mathcal{X}^* \to \mathbb{R}$ of a convex function $f : \mathcal{X} \to \mathbb{R}$ is*

$$f^*(\mathbf{z}) = \sup_{\mathbf{x}\in\mathcal{X}} \langle \mathbf{x}, \mathbf{z} \rangle - f(\mathbf{x}) \,.$$

In this section, we use the space of matrices (either $\mathbb{R}^{m\times n}$ or $\mathbb{S}^d$) with the inner product $\langle \mathbf{X}, \mathbf{Y} \rangle = \mathrm{Tr}(\mathbf{X}^\top \mathbf{Y})$. Thus, the dual space of $\mathcal{X}$ is $\mathcal{X}$ itself and f^* is defined over $\mathcal{X}$.

We need to relate the conjugate of a trace function to that of a scalar function. This is achieved by the following result, restated from Thm. 12 (Kakade et al., 2009). The theorem implies that the conjugate of a singular-values function is the singular-values function lifted from the conjugate of the scalar function.

Theorem 11 *Let $F = (f \circ \sigma)$ be defined as in (4), then $F^* = (f^* \circ \sigma)$.*

5.2. Duality of Strong Convexity and Smoothness

Recall that a function f is L-smooth with respect to a norm $\|\cdot\|$ on $\mathcal{K}$ if,

$$\forall \mathbf{x}, \mathbf{y} \in \mathcal{K}, \;\; f(\mathbf{x}) - f(\mathbf{y}) - \nabla f(\mathbf{y})(\mathbf{x} - \mathbf{y}) \leq \frac{L}{2} \|\mathbf{x} - \mathbf{y}\|^2 \,.$$

For convenience, we use the following second order characterization of smoothness which is an analogue of Lemma 5.

Lemma 12 *A twice differentiable function $f : \mathcal{X} \to \mathbb{R}$ is L-smooth with respect to $\|\cdot\|$ at $\mathbf{x}$ iff*

$$\sup_{\mathbf{y}\in\mathcal{X}:\|\mathbf{y}\|=1} \mathbf{y}^\top \nabla^2 \phi(\mathbf{x})\, \mathbf{y} \leq L \,.$$

Strong convexity and smoothness are dual notions in the sense that f is α-strongly convex with respect to a norm $\|\cdot\|$ iff its Fenchel conjugate f^* is α^{-1}-smooth with respect to the dual norm $\|\cdot\|_*$.

For the matrix variant of hypentropy we find it easier to show smoothness of the conjugate rather than strong convexity directly. Unfortunately, as we see in the sequel, the Hessian of the conjugate function is not globally upper bounded. Therefore, we would need a local variant of the duality of strong convexity and smoothness. In the context of mirror descent with mirror map $\nabla\phi$, we show that ϕ is strongly convex over $\mathcal{K}$ if ϕ^* is locally smooth at all points within the image of the mirror map. Formally, we have the following lemma whose proof we provide in App. E. In the following we use the standard notation for image of vector functions, $\nabla f(S) = \{y | \nabla f(x) = y, x \in S\}$,

Lemma 13 (Local duality of smoothness and strong convexity) *Let $\mathcal{K} \subseteq \mathbb{R}^d$ be an open convex set and $\|\cdot\|$ be a norm with dual norm $\|\cdot\|_*$. Let $\phi : \mathbb{R}^d \to \mathbb{R}$ be twice differentiable, closed and convex function. Suppose the Fenchel conjugate $\phi^* : \mathbb{R}^d \to \mathbb{R}$ is L smooth with respect to $\|\cdot\|_*$ over $\mathcal{C} = \nabla\phi(\mathcal{K})$. Then, ϕ is $\frac{1}{L}$ strongly convex with respect to $\|\cdot\|$ over $\mathcal{K}$.*

5.3. Strong Convexity of Spectral Hypentropy

We now analyze the strong convexity of the spectral hypentropy . The spectral function $\Phi_\beta(\mathbf{X})$ is defined for $\mathbf{X} \in \mathbb{R}^{m\times n}$ by (4) and for $\mathbf{X} \in \mathbb{S}^d$ by (5) replacing f with ϕ_β. The main theorem of this subsection is as follows.

Theorem 14 *The trace function $\Phi_\beta : \mathbb{R}^{m\times n} \to \mathbb{R}$ is $(2(\tau + \beta \min\{m, n\}))^{-1}$-strongly convex with respect to the trace norm over $B_{\mathrm{Tr}}(\tau)$.*

We denote the d-dimensional symmetric matrices of trace-ball with maximal radius τ by

$$\mathbb{B}_{\mathrm{Tr}}(\tau) = \left\{\mathbf{X} \in \mathbb{S}^d : \|\mathbf{X}\|_1 \le \tau\right\} .$$

We prove the above theorem by first proving the lemma below for matrices in $\mathbb{B}_{\mathrm{Tr}}(\tau)$. We then extend it to arbitrary matrices using a symmetrization argument, using a technique similar to (Juditsky and Nemirovski, 2008; Warmuth, 2007; Hazan et al., 2012). The following property will be of use:

Definition 15 *Define $\mathcal{K}$ to be (r, τ)-compatible with Φ_β if $\mathcal{K} \subseteq \mathbb{B}_{\mathrm{Tr}}(\tau)$ is a convex subset of matrices that lives in a vector space $\mathcal{X} \subseteq \mathbb{S}^d$ such that $\forall \mathbf{X} \in \mathcal{X}, \mathrm{rank}(\mathbf{X}) \le r$ and $\nabla\Phi_\beta(\mathbf{X}) \in \mathcal{X}$.*

This abstraction will be useful in translating strong convexity of arbitrary matrices to the symmetric case. The bound on the rank is essential to give a modulus of strong convexity result that depends only on $\min\{m, n\}$ rather than $m + n$. The final property is necessary for the low rank structure to be preserved after a primal-dual mapping.

Lemma 16 *The trace function Φ_β is $(2(\tau + \beta r))^{-1}$-strongly convex w.r.t the trace norm over (r, τ)-compatible set $\mathcal{K}$.*

To prove the symmetric variant, we show that Φ_β^* is smooth with respect to the spectral norm, which is the dual norm of the trace norm. The result then follows directly from Lemma 13.

From Theorem 11, Φ_β has Fenchel conjugate

$$\Phi_\beta^*(\mathbf{X}) = \mathrm{Tr}(\phi_\beta^*(\mathbf{X})) .$$

Since the derivative of the conjugate of a function is the inverse of the derivative of the function, we have

$$\frac{d\phi_\beta^*}{dx} = \left(\frac{d\phi_\beta}{dx}\right)^{-1} = \beta \sinh(x) .$$

The indefinite integral of the above yields that up to a constant, $\phi_\beta^*(x) = \beta\cosh(x)$. Clearly, Φ_β is not globally smooth. Nonetheless, we do have smoothness over $\nabla\Phi_\beta(\mathbb{B}_{\mathrm{Tr}}(\tau))$. Before proving this property, we introduce a clever technical lemma of Juditsky and Nemirovski (2008) that allows us to reduce the spectral smoothness for matrices to smoothness of functions in the vector-case.

Lemma 17 *Let $f : \mathbb{R} \to \mathbb{R}$ be a function and $c \in \mathbb{R}_+$ such that that for $a \geq b$,*

$$\frac{f'(a) - f'(b)}{a - b} \leq \frac{c(f''(a) + f''(b))}{2} . \tag{6}$$

Let $F : \mathbb{S}^d \to \mathbb{R}$ be a function defined by $F(\mathbf{X}) = \operatorname{Tr}(f(\mathbf{X}))$. Then, the second directional derivative of F is bounded for any $\mathbf{H} \in \mathbb{S}^d$ as follows,

$$D^2 F(\mathbf{X}) \left[\mathbf{H}, \mathbf{H}\right] \leq c \operatorname{Tr}(\mathbf{H} f''(\mathbf{X}) \mathbf{H}) .$$

We are now prepared to analyze the smoothness of Φ_β^*.

Lemma 18 (Local Smoothness) *If $\mathcal{K}$ is (r, τ)-compatible with Φ_β, then the trace function Φ_β^* is $2(\tau + \beta r)$-smooth with respect to the spectral norm over $\nabla\Phi_\beta(\mathcal{K})$.*

Proof To prove local smoothness, we use the second order conditions from Lemma 12. This requires us to upper bound the second directional derivatives for all directions corresponding to matrices of unit spectral norm. We consider the matrix

$$\mathbf{X} = \nabla\Phi_\beta(\mathbf{Y}) = \phi_\beta'(\mathbf{Y}) = \operatorname{arcsinh}\left(\frac{\mathbf{Y}}{\beta}\right) .$$

Note that $(\phi_\beta^*)''(x) = \beta \cosh(x)$ is positive and convex. Therefore, by the mean value theorem, there exists $c \in [a, b]$ for which,

$$\frac{(\phi_\beta^*)'(b) - (\phi_\beta^*)'(a)}{b - a} = (\phi_\beta^*)''(c) \leq \max\left\{(\phi_\beta^*)''(a)\,,\, (\phi_\beta^*)''(b)\right\} \leq (\phi_\beta^*)''(a) + (\phi_\beta^*)''(b) .$$

We note that by Definition 15, $\nabla\Phi_\beta(\mathcal{X}) \subseteq \mathcal{X}$, so we can restrict ourselves to the vector space $\mathcal{X}$. Therefore, applying Lemma 17, we have

$$\begin{aligned}
\sup_{\mathbf{H}\in\mathcal{X}:\|\mathbf{H}\|_\infty\leq 1} D^2\Phi_\beta^*(\mathbf{X}) \left[\mathbf{H}, \mathbf{H}\right] &\leq \sup_{\mathbf{H}\in\mathcal{X}:\|\mathbf{H}\|_\infty\leq 1} 2 \operatorname{Tr}\left(\mathbf{H}(\phi_\beta^*)''(\mathbf{X})\,\mathbf{H}\right) \\
&= \sup_{\mathbf{H}\in\mathcal{X}:\|\mathbf{H}\|_\infty\leq 1} 2 \operatorname{Tr}\left(\mathbf{H}^2(\phi_\beta^*)''(\mathbf{X})\right) && \text{[Commutativity of trace]} \\
&\leq \sup_{\mathbf{H}\in\mathcal{X}:\|\mathbf{H}\|_\infty\leq 1} 2\langle\sigma^2(\mathbf{H}), \sigma((\phi_\beta^*)''(\mathbf{X}))\rangle && \text{[Von Neumann's trace inequality]}
\end{aligned}$$

where Von Neumann's trace inequality stands for, $\operatorname{Tr}(\mathbf{A}^\top\mathbf{B}) \leq \langle\sigma(\mathbf{A}), \sigma(\mathbf{B})\rangle$. Now, since $\mathbf{H} \in \mathcal{X}$, we know $\operatorname{rank}(\mathbf{H}) \leq r$, and so $\mathbf{H}$ can have at most r nonzero singular values, yielding

$$\sup_{\mathbf{H}\in\mathcal{X}:\|\mathbf{H}\|_\infty\leq 1} D^2\Phi_\beta^*(\mathbf{X}) \left[\mathbf{H}, \mathbf{H}\right] \leq \sup_{\mathbf{H}\in\mathcal{X}:\|\mathbf{H}\|_\infty\leq 1} 2\|\mathbf{H}^2\|_\infty \sum_{i=1}^r \sigma_i((\phi_\beta^*)''(\mathbf{X})) = 2\sum_{i=1}^r (\phi_\beta^*)''(\phi_\beta'(\sigma_i(\mathbf{Y}))) .$$

Now, note that

$$(\phi_\beta^*)''(\phi_\beta'(x)) = \beta\cosh\left(\operatorname{arcsinh}\left(x/\beta\right)\right) = \sqrt{\beta^2 + x^2} \leq \beta + |x| .$$

It then follows that

$$\sum_{i=1}^{r}(\phi_\beta^*)''(\phi_\beta'(\sigma_i(\mathbf{Y}))) \le \beta r + \|\mathbf{Y}\|_1 \le \tau + \beta r\ .$$

Therefore, the second directional derivative in bounded by $2(\tau+\beta r)$ as desired. ■

Proof [Theorem 14] We introduce the symmetrization operator $S:\mathbb{R}^{m\times n}\to\mathbb{S}^{m+n}$ which is a linear function that takes a matrix to a symmetric matrix,

$$S(\mathbf{X}) = \begin{bmatrix} 0 & \mathbf{X} \\ \mathbf{X}^\top & 0 \end{bmatrix}\ .$$

The eigenvalues of $S(\mathbf{X})$ are exactly one copy of singular values and one copy of negative singular values of $\mathbf{X}$. Therefore, we have

$$\Phi_\beta(\mathbf{X}) = \sum_{i=1}^{\min\{m,n\}} \phi_\beta(\sigma_i(\mathbf{X})) = \frac{1}{2}\Phi_\beta(S(\mathbf{X}))\ .$$

Technically, for the above to hold true, we should shift ϕ_β such that its 0 is at 0. Since a constant shift does not affect diameter or convexity properties so this is not an issue.

Let μ be the modulus of strong convexity in the symmetrized space. We bound $\Phi_\beta(\mathbf{X})$ from below as follows,

$$\begin{aligned}
2\Phi_\beta(\mathbf{X}) &= \Phi_\beta(S(\mathbf{X})) \\
&\ge \Phi_\beta(S(\mathbf{Y})) + \langle\nabla\Phi_\beta(S(\mathbf{Y})), S(\mathbf{X})-S(\mathbf{Y})\rangle + \frac{\mu}{2}\|S(\mathbf{X})-S(\mathbf{Y})\|_1^2 \\
&= \Phi_\beta(S(\mathbf{Y})) + \langle\nabla\Phi_\beta(S(\mathbf{Y})), S(\mathbf{X}-\mathbf{Y})\rangle + \frac{\mu}{2}\|S(\mathbf{X}-\mathbf{Y})\|_1^2 \\
&= 2\Phi_\beta(\mathbf{Y}) + 2\langle\nabla\Phi_\beta(\mathbf{Y}), \mathbf{X}-\mathbf{Y}\rangle + 2\mu\|\mathbf{X}-\mathbf{Y}\|_1^2\ .
\end{aligned}$$

Therefore, the modulus of strong convexity over asymmetric matrices is 2μ.

Note that $\mathcal{K}=\{S(\mathbf{X}):\mathbf{X}\in B_{\mathrm{Tr}}(2\tau)\}$ is $(2\min\{m,n\},\tau)$-compatible with Φ_β. In particular, we have a vector space $\mathcal{X}=\{S(\mathbf{X}):\mathbf{X}\in\mathbb{R}^{m\times n}\}$ containing symmetric matrices of rank at most $2\min\{m,n\}$. Furthermore, for any $\mathbf{X}\in\mathcal{X}$, $\mathbf{X}=S(\mathbf{Y})$ for some $\mathbf{Y}\in\mathbb{R}^{m\times n}$ and thus,

$$\nabla\Phi_\beta(\mathbf{X}) = \nabla\Phi_\beta(S(\mathbf{Y})) = S(\nabla\Phi_\beta(\mathbf{Y}))\in\mathcal{X}\ .$$

It follows from Lemma 16 that $\mu\ge(2(2\tau+2\beta\min\{m,n\}))^{-1}$, and so the strong convexity is at most $2\mu\ge(2(\tau+\beta\min\{m,n\}))^{-1}$ as desired. ■

Algorithm 3: EG$\pm$

Input: $\eta > 0, \beta > 0$
Initialize: $\mathbf{u}_i^1 = \mathbf{v}_i^1 = \frac{\beta}{2}$, $\mathbf{G}^0 = \mathbf{0}$;
for $t = 1$ **to** T **do**
  (a) Predict $\mathbf{w}^t = \mathbf{u}^t - \mathbf{v}^t$ (b) Incur loss $\ell_t(\mathbf{w}_t)$ (c) Calculate $\mathbf{G}^t = \mathbf{G}^{t-1} + \underbrace{\nabla \ell_t(\mathbf{w}^t)}_{\mathbf{g}^t}$;
  Update: $\mathbf{u}^{t+\frac{1}{2}} = \mathbf{u}^t \exp(-\eta \mathbf{g}^t)$ and $\mathbf{v}^{t+\frac{1}{2}} = \mathbf{v}^t \exp(\eta \mathbf{g}^t)$;
  Normalize weights: $(\mathbf{u}^{t+1}, \mathbf{v}^{t+1}) = \beta d \left(\sum_{i=1}^d u_i^{t+\frac{1}{2}} + v_i^{t+\frac{1}{2}} \right)^{-1} (\mathbf{u}^{t+\frac{1}{2}}, \mathbf{v}^{t+\frac{1}{2}})$
end

6. Connections to EG$\pm$

The EG$\pm$ algorithm (Kivinen and Warmuth, 1997) maintains two vectors $\mathbf{u}$ and $\mathbf{v}$ such that, $\mathbf{w} = \mathbf{u} - \mathbf{v}$. The two vectors are updated over $\mathbb{R}_+^{2d}$ using the EG algorithm. We consider here a variant where the $2d$ dimensional weights are normalized such that their sum is βd. Typically, we would have $\beta = \frac{1}{d}$, so $(\mathbf{u}, \mathbf{v})$ lie on a unit simplex.

We now show that the EG$\pm$ algorithm can be viewed as an adaptive variant of HU with the update,

$$\begin{aligned} \mathbf{y}^{t+1} &= (\nabla \phi_{\beta_t})^{-1}(\nabla \phi_{\beta_{t-1}}(\mathbf{w}^t) - \eta \mathbf{g}^t) \\ \mathbf{w}^{t+1} &= \arg\min_{\mathbf{v} \in \mathcal{K}} D_\phi^{\beta_t}(\mathbf{v} \parallel \mathbf{y}^{t+1}) , \end{aligned} \tag{7}$$

where $\mathcal{K}$ is the 1-ball of radius βd.

In this adaptive update, $\nabla \phi_{\beta_{t-1}}$ is used to map into the dual space where a gradient update occurs. Afterwards, $(\nabla \phi)_{\beta_t}^{-1}$ maps back to the primal. When used in an OCO setting over the norm-1 ball, β_t can always be chosen to be sufficiently small such that projection step is voided. EG$\pm$ fits into this algorithmic paradigm with a specific choice of β_t that avoids hypentropy projection. In the setting of OCO over the norm-1 ball of radius βd, we have the following result.

Theorem 19 EG$\pm$ *with learning rate* η *is equivalent to the adaptive* HU *algorithm described in* (7) *with the same learning rate and* $\beta_t = \beta d \left(\sum_{i=1}^d \cosh(\eta \mathbf{G}_i^t) \right)^{-1}$*, where* $\mathbf{G}^t = \sum_{s=1}^t \mathbf{g}^t$.

Proof We start with some analysis of EG$\pm$. We have $u_i^{t+1} \propto \exp(-\eta \mathbf{G}_i^t)$ and similarly $v_i^{t+1} \propto \exp(\eta \mathbf{G}_i^t)$. Rescaling such that $\|(\mathbf{u}^{t+1}, \mathbf{v}^{t+1})\|_1 = \beta d$ yields the normalization factor,

$$\frac{\beta d}{\sum_{i=1}^d \exp(-\eta \mathbf{G}_i^t) + \exp(\eta \mathbf{G}_i^t)} = \frac{\beta d}{2 \sum_{i=1}^d \cosh(\eta \mathbf{G}_i^t)}$$

Putting the above all together, we get

$$\begin{aligned} w_i^{t+1} = u_i^{t+1} - v_i^{t+1} &= \frac{\beta d}{2 \sum_{i=1}^d \cosh(\eta \mathbf{G}_i^t)} \big(\exp(-\eta \mathbf{G}_i^t) - \exp(\eta \mathbf{G}_i^t) \big) \\ &= \frac{\beta d \sinh(-\eta \mathbf{G}_i^t)}{\sum_{i=1}^d \cosh(\eta \mathbf{G}_i^t)} = \beta_t \sinh(-\eta \mathbf{G}_i^t) . \end{aligned} \tag{8}$$

Therefore, we have $\mathbf{w}^{t+1} = \nabla\phi_{\beta_t}^{-1}(-\eta\mathbf{G}^t)$. We can now show that the adaptive HU algorithm described in (7) results in the same weights. We prove this property by induction on t. The base case follows because we initialize $\mathbf{w}^0 = 0$ in HU. Now we assume that $\mathbf{w}^t = \nabla\phi_{\beta_{t-1}}^{-1}(-\eta\mathbf{G}^{t-1})$. Applying the hypentropy update, we have

$$\mathbf{y}^{t+1} = \nabla\phi_{\beta_t}^{-1}(\nabla\phi_{\beta_{t-1}}(\nabla\phi_{\beta_{t-1}}^{-1}(-\eta\mathbf{G}^{t-1})) - \eta\mathbf{g}^t) = \nabla\phi_{\beta_t}^{-1}(-\eta\mathbf{G}^{t-1} - \eta\mathbf{g}^t) = \nabla\phi_{\beta_t}^{-1}(-\eta\mathbf{G}^t)\ .$$

Now, note that $\forall x \in \mathbb{R}, |\sinh(x)| \leq \cosh(x)$, so $\|\mathbf{y}^{t+1}\|_1 \leq \beta d$ so projection is not required and $\mathbf{w}^{t+1} = \mathbf{y}^{t+1}$. ■

We also find it useful to consider these updates without normalization or projection. Without any constraint, the regularization parameter β does need to change.

Theorem 20 *Running* HU *with a learning rate* η *and a regularization parameter* β *without projection is equivalent to running* EG$\pm$ *without normalization.*

The proof can be found in App. F.

Discussion While, we can represent EG$\pm$ as an adaptive variant of HU, we still would like to understand how HU with a fixed β relates to EG$\pm$. A brief look into β_t provides some intuition that the two still should be similar updates. Note that for small η, $\cosh(\eta\mathbf{G}_i^t) = 1 + O(\eta^2)$ and $\beta_t \approx \beta$. In this regime, a fixed β should result in a similar update. It should be noted that projection is then an interpolation of weight-scaling and soft-thresholding. The relation to EG$\pm$ also motivates the choice of $\beta \approx \frac{\|\mathbf{w}^\star\|}{d}$ as this provides the right scale for EG$\pm$. For simplicity, let $\beta = \frac{1}{d}$. We see from (8) that the weights follow $\mathbf{w}^t = \nabla\phi^*(-\eta\mathbf{G}^t)$ where $\phi^*(\mathbf{x}) = \log(\sum_{i=1}^d \cosh(x_i))$.

References

Advanced convolutional neural networks. https://goo.gl/8vmt1B.

S. Arora and S. Kale. A combinatorial, primal-dual approach to semidefinite programs. In *Proc. 39th Symposium on Theory Of Computing*, pages 227–236, 2007.

S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

N. Cesa-Bianchi, A. Conconi, and C. Gentile. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50(9):2050–2057, 2004.

J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

C. Gentile. The robustness of the p-norm algorithms. *Machine Learning*, 53(3):265–299, 2003.

A.J. Grove, N. Littlestone, and D. Schuurmans. General convergence results for linear discriminant updates. *Machine Learning*, 43(3):173–210, 2001.

E. Hazan. Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2 (3-4):157–325, 2016.

E. Hazan, S. Kale, and S. Shalev-Shwartz. Near-optimal algorithms for online matrix prediction. In *Conference on Learning Theory*, pages 38–1, 2012.

A. Juditsky and A.S. Nemirovski. Large deviations of vector-valued martingales in 2-smooth normed spaces. *arXiv:0809.0813*, 2008.

Sham M Kakade, Shai Shalev-Shwartz, and Ambuj Tewari. Regularization techniques for learning with matrices. *Journal of Machine Learning Research*, 13:1865–1890, 2012.

S.M. Kakade, S. Shalev-Shwartz, and A. Tewari. Applications of strong convexity–strong smoothness duality to learning with matrices. *arXiv:0910.0610*, 2009.

J. Kivinen and M.K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–63, 1997.

A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, 2009.

A. Krizhevsky, V. Nair, and G. Hinton. CIFAR-10. URL https://goo.gl/xKwPJo.

S. Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

K. Tsuda, G. Rätsch, and M.K. Warmuth. Matrix exponentiated gradient updates for on-line learning and bregman projection. *Journal of Machine Learning Research*, 6:995–1018, 2005.

Manfred Warmuth. Personal communication (circa 2010).

M.K. Warmuth. Winnowing subspaces. In *Proc. 24th International Conference on Machine Learning*, pages 999–1006, 2007.

Yao-Liang Yu. The strong convexity of von neumann's entropy, June 2013.

Appendix A. Experimental Results

Next, we experiment with HU in the context of empirical risk minimization (ERM). In the experiments, $\mathbf{g}_t$ stands for a stochastic estimate of the gradient of the empirical loss. Thus, we can convert the regret analysis to convergence in expectation (Cesa-Bianchi et al., 2004).

Effective Learning Rate For small value w, $\sinh(w) \approx w \approx \operatorname{arcsinh}(w)$. As a result, near 0 the update in HU is morally the additive update, $w_i^{t+1} = w_i^t - \beta\eta g_i^t$. The product $\beta\eta$ can be viewed as the de facto learning rate of the gradient descent portion of the interpolation. As such, we define the *effective learning rate* to be $\beta\eta$. In the sequel, fixing the effective learning rate while changing β is a fruitful lens for comparing HU to with GD.

A.1. Logistic Regression

In this experiment we use the HU algorithm to optimize a logit model. The ambient dimension d is chosen to be 500. A weight $\mathbf{w}$ is drawn uniformly at random from $[-1, 1]^d$. The features are from $\{0, 1\}^d$ and distributed according to the power law, $\Pr[x_i = 1] = 1/5\sqrt{i}$. The label associated with an example $\mathbf{x}_t$ is set to $y_t = \text{sign}(\langle \mathbf{w}, \mathbf{x}_t \rangle)$ with with probability 0.9 and otherwise flipped.

The algorithms are trained with log-loss using batches of size 10. Stochastic gradient descent and the p-norm algorithm (Gentile, 2003) ($p = 2 \ln d$) are used for comparison. Metaparameters were selected via grid search. As can be seen in Fig. 2, the p-norm algorithm performs significantly worse than HU for a large set of values of β, while SGD performs comparably. As expected, for large value of β, SGD and HU are indistinguishable.

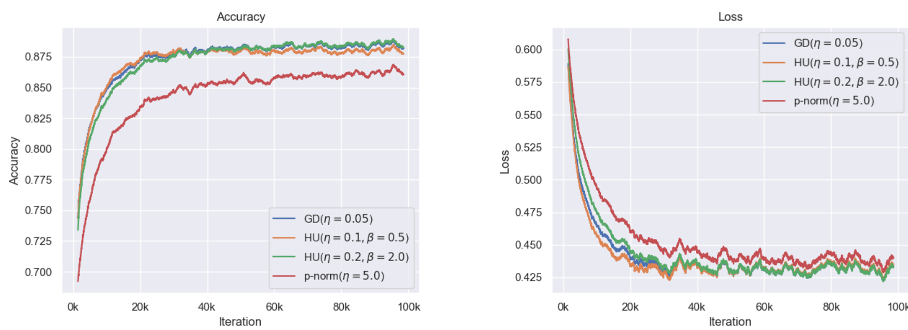


Figure 2: Comparison of accuracy and loss of GD, p-norm, and HU on binary logistic regression.

In the next experiment we use the same logit model with ambient dimension d chosen to be $10,000$. We generate weights, $\mathbf{w} \in \mathbb{R}^d$, with sparsity (fraction of zero weights) $s \in \{0, 0.9\}$. The nonzero weights are chosen uniformly at random from $[-1, 1]$. We run the algorithms for 20,000 iterations. Rather than fixing η, we fix $\eta' = \eta/\sqrt{1+\beta^2}$. This way, as $\beta \to \infty$, $\text{HU}(\eta', \beta)$ behaves like $\text{GD}(\eta)$ while for small β, the update is roughly $\text{EG}\pm(\eta)$. We let $\beta_{\text{EG}} = \|\mathbf{w}^\star\|_1/d$. As discussed in Appendix 6, this choice of β is similar to running EG$\pm$ with an 1-norm bound of $\|\mathbf{w}^\star\|_1$. We then choose $\eta' = 0.1$ and $\beta \in \{0.5, 1, 2, 4, 8\} \times \beta_{\text{EG}}$. In Fig. 3, we show the interpolation between GD and EG$\pm$. The larger β is, the closer the progress of HU resembles that of GD. Intermediate values of β have progress in between the EG$\pm$ and GD. We do not plot EG$\pm$ as it is exactly the same as running HU with β_{EG} as there is no projection in this experiment.

A.2. Multiclass Logistic Regression

In this experiment we use SHU to optimize a multiclass logistic model. We generated 200,000 examples in $\mathbb{R}^{25}$. We set the number of classes to $k = 15$. Labels are generated using a rank 5 matrix $\mathbf{W} \in \mathbb{R}^{k \times d}$. An example $\mathbf{x} \in \mathbb{R}^d$ was labeled according to the prediction rule,

$$y = \arg\max_{i \in [k]} \{(\mathbf{W}\mathbf{x})_i\} \, .$$

With probability 5.0% the label was flipped to a different one at random. The matrix $\mathbf{W}$ and each example $\mathbf{x}_i$ features are determined in a joint process to make the problem poorly conditioned for optimization. Features of each example are first drawn from a standard normal. Weights of $\mathbf{W}$ are

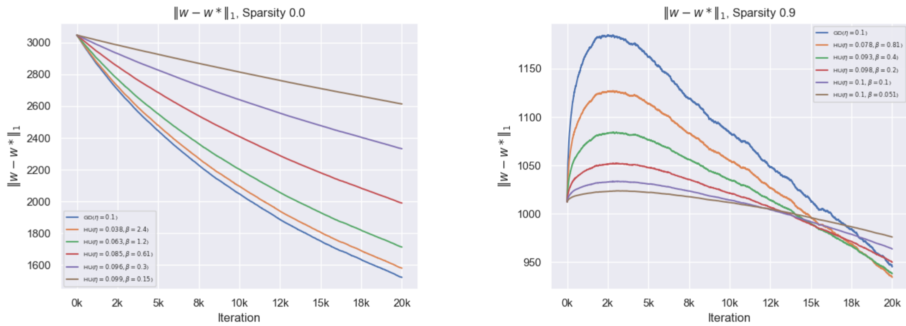


Figure 3: Value of $\|\mathbf{w}^t - \mathbf{w}^\star\|_1$ in dense and sparse settings with $\eta' = 0.1$ and $\beta \in \{0.5, 1, 2, 4, 8\} \times \beta_{\text{EG}}$.

sampled from a standard normal distribution for the first r features and are set to 0 for the remaining $d - r$ features. After labels are computed, features are perturbed by Gaussian noise with standard deviation 0.05. The examples and weights are then scaled and rotated. Coordinate i of the data is scaled by $s_i \propto i^{-1.1}$ where $\sum_{i=1}^{d} s_i = 1$. Then a random rotation $\mathbf{R}$ is applied. The inverse of these transformation is applied to the weights. Therefore, from the original sample $\mathbf{X}_0 \in \mathbb{R}^{n \times d}$ and weights $\mathbf{W}_0 \in \mathbb{R}^{k \times d}$ the new sample and weights are set to be $\mathbf{X} = \mathbf{X}_0 \mathbf{R}$ and $\mathbf{W} = \mathbf{W}_0 \mathbf{R}^{-1}$, where $\mathbf{R} \in \mathbb{R}^{d \times d}$ is the scaling and rotation described above.

Since our ground truth weights are low rank, our goal is to find weights of approximately low rank with low classification error. To do this, we optimize a multiclass logistic regression loss with a trace-norm constraint. We compare SHU, Schatten p-norm algorithm (p-norm algorithm applied to singular values) and gradient descent in the fully stochastic (single example) case. We report results for unconstrained optimization in Fig. 4 and trace-norm constrained optimization in Fig. 5. In these figures only the algorithms with lowest final loss after a grid search are depicted.

Without projection, SHU results in the largest trace norm solution of norm 700 whereas the p-norm algorithm and GD reach solutions with trace norm just above 600. Nevertheless, SHU attains the lowest classification error and loss. Performance is noticeably better than gradient descent. Moreover, the spurious singular values are typically smaller than that of gradient descent. This pattern holds up in both settings.

With our new divergence, the SHU update looks exponential for large singular values and linear for small ones. In this sense, once gradients start accumulating in the directions that correspond to the actual signal, these directions can be exploited exponentially. On the other hand, the spurious directions are morally handled with gradient descent with an effective learning rate $\eta\beta$. Note that in our experiments, the SHU effective learning rate is smaller than the GD learning rate by an order of magnitude. This may explain the smaller magnitude in erroneous singular values. On the other hand, the p-norm algorithm not only increases the magnitude of large singular values, but also shrinks the magnitude of small singular values, resulting in solutions that are closer to being low-rank. To see this, note that the p-norm inverse mirror map has the form

$$f(\boldsymbol{\sigma})_i = \sigma_i^{p-1} / \|\boldsymbol{\sigma}\|_p^{p-2}$$

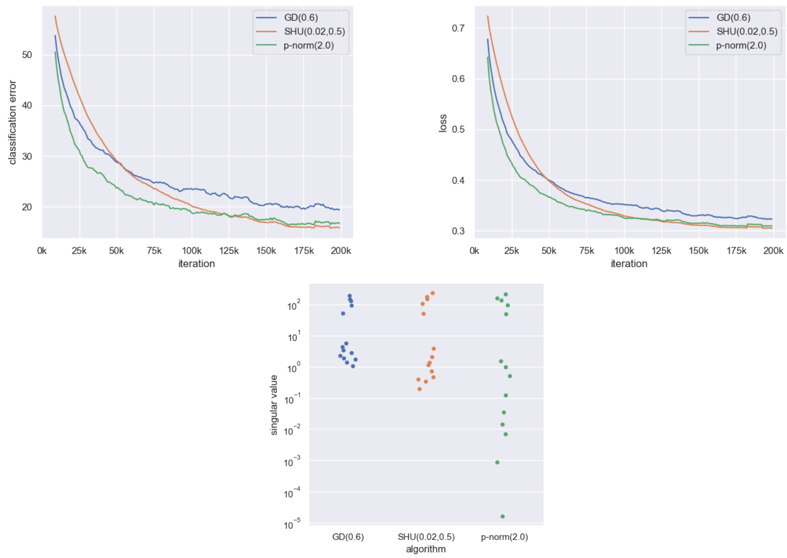


Figure 4: Unconstrained minimization of logistic loss.

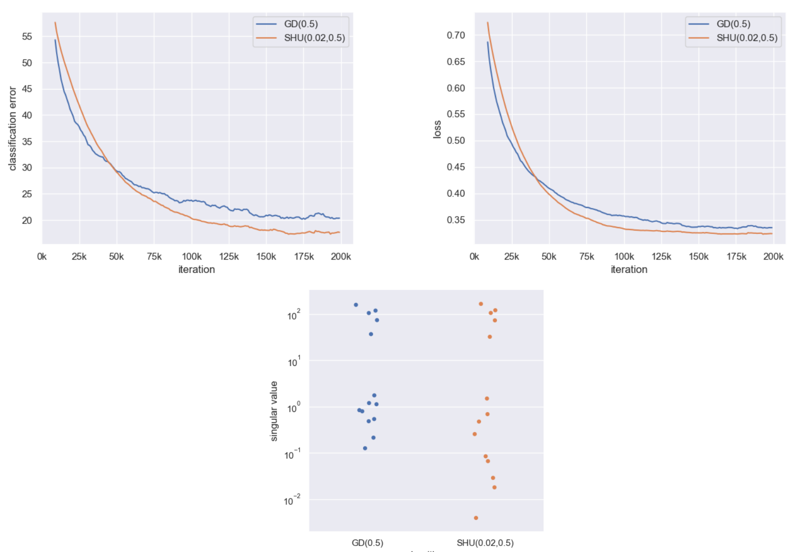


Figure 5: Minimization over the trace-norm ball of radius 500. The p-norm algorithm is not included because the p-norm divergence does not have a closed form projection onto the 1-ball.

for $p = 2\ln(k) \approx 5.4$. Therefore, there is a natural normalization which shrinks small singular values as good directions are exploited. Without projection, this does not happen with SHU. Informally speaking, the following analogy applies to the three methods:

GD: The rich get richer!
SHU: The rich get *much* richer!!
p-norm: The rich get richer and the poor get poorer, oy!

Adding trace norm projection reduces the magnitude of these singular values, but not to the level which the p-norm algorithm can achieve. Overall, it appears that SHU may be slightly more effective at reducing loss but the p-norm algorithm is more effective at producing low rank solutions.

A.3. Image Classification with Neural Networks.

Loss minimization for neural networks is known to be nonconvex, thus the regret bounds from this paper do not apply in this setting. Still, convex optimization algorithms, such as AdaGrad, work well practice for training neural networks. In this section, we use the unconstrained version of the HU to find the weights of a simple neural network for image classification using the popular CIFAR10 dataset (Krizhevsky et al.). SGD was used for comparison. Outside of use of the HU algorithm, the design of the network and code are from the Tensorflow tutorial on convolutional networks for image classification. The network involves 2 convolutional layers, max pooling, and 2 fully connected layers, all using ReLU activations. The loss function is the cross entropy loss plus a 2-norm regularization. For a complete description of the experimental setup see (TFT; Krizhevsky and Hinton, 2009).

Empirically, SGD with learning rate η tended to perform similarly in terms of training error to HU with equivalent effective learning rate $\beta\eta$ for a range of values of β. In order to compare to SGD with learning rate η, β was varied and HU's learning rate was set to be $\frac{\eta}{\beta}$ (in order to keep the effective learning rate invariant). As can be seen from Figure 6, the loss curves for a variety of values of β are very similar for $\beta\eta = 0.005$, although the smallest $\beta = 0.1$ has slightly lower loss. In general, the final loss reached is similar for a fixed effective learning rate. In addition, there is a clear pattern indicating that shrinking β results in sparser weights. This may warrant further investigation.

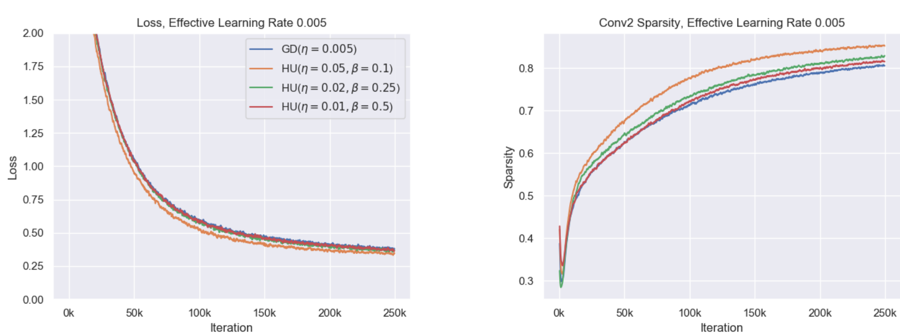


Figure 6: CIFAR10 loss and sparsity level. The effective learning rate was held constant at 0.005 while β is varied. Sparsity is displayed for the second convolutional layer conv2. The loss corresponds to the total loss, which includes regularization.

Algorithm 4: Online Mirror Descent with potential function R

Input: $\eta > 0, \beta > 0$, convex domain $\mathcal{K} \subseteq \mathbb{R}^d$
Let $\mathbf{y}_1$ be such that $\nabla R(\mathbf{y}_1) = 0$ and $\mathbf{w}^1 = \arg\min_{\mathbf{w}\in\mathcal{K}} D_R(\mathbf{w} \parallel \mathbf{y}^1)$;
for $i = 1$ **to** T **do**
  (a) Predict $\mathbf{w}^t$ (b) Incur loss $\ell_t(\mathbf{w}_t)$ (c) Calculate $\mathbf{g}^t = \nabla \ell_t(\mathbf{w}^t)$;
  Update: $\nabla R(\mathbf{y}^{t+1}) = \nabla R(\mathbf{w}^t) - \eta \mathbf{g}^t$;
  Project onto $\mathcal{K}$: $\mathbf{w}^{t+1} = \arg\min_{\mathbf{w}\in\mathcal{K}} D_R(\mathbf{w} \parallel \mathbf{y}^{t+1})$
end

Appendix B. Online Mirrored Descent

Online Mirror Descent (OMD) is an meta-algorithm for online convex optimization. The regularization function, R, is assumed to be strongly convex, smooth, and twice differentiable. Like GD, Mirror Descent is an iterative algorithm involving a simple gradient update. R defines a mapping into a dual space where the updates occur, followed by an inverse mapping to the original space. This step may result in a vector outside of $\mathcal{K}$, so a projection is required. An alternative formulation where the regularization casts a trade-off between moving along the gradient direction and staying close to the current iterate is,

$$\mathbf{w}^{t+1} = \arg\min_{\mathbf{w}\in\mathcal{K}} \left\{ \eta \langle \mathbf{g}^t, \mathbf{w} \rangle + D_R(\mathbf{w} \parallel \mathbf{w}^t) \right\} . \tag{9}$$

For Algorithm 4 with the above assumptions, we have the following regret bound.

Theorem 21 (OMD Regret) *Assume R is μ-strongly convex with respect to a norm $\|\cdot\|$ whose dual is $\|\cdot\|_*$, then running* OMD *with a fixed learning rate η yields the following regret bound,*

$$\mathcal{R}_T \leq \frac{1}{\eta} \sup_{\mathbf{w}\in\mathcal{K}} D_R(\mathbf{w} \parallel \mathbf{w}^1) + \frac{\eta}{2\mu} \sum_{t=1}^{T} \|\mathbf{g}^t\|_*^2 .$$

If we have a bound on the dual norm of a gradient, we can choose a learning rate which minimizes the upper bound, yielding Theorem 24. We next introduce two well known properties of Bregman divergences without proof. The first technical lemma is the Bregman divergence analogue of the law of cosines.

Lemma 22 (Three-point Lemma) *For every three vectors $\mathbf{x}, \mathbf{y}, \mathbf{z}$,*

$$D_R(\mathbf{x} \parallel \mathbf{z}) = D_R(\mathbf{x} \parallel \mathbf{y}) + D_R(\mathbf{y} \parallel \mathbf{z}) - \langle \nabla R(\mathbf{z}) - \nabla R(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle .$$

The next lemma is an analogue of the Pythagorean theorem for Bregman projections.

Lemma 23 (Generalized Pythagorean Theorem) *Let*

$$\mathbf{x}' = \Pi_{\mathcal{K},\phi}(\mathbf{x}) = \arg\min_{\mathbf{y}\in\mathcal{K}} D_R(\mathbf{y} \parallel \mathbf{x}) ,$$

then

$$D_R(\mathbf{z} \parallel \mathbf{x}) \geq D_R(\mathbf{z} \parallel \mathbf{x}') + D_R(\mathbf{x}' \parallel \mathbf{x}) .$$

Proof [Theorem 21] Let $\mathbf{w}^* = \arg\min_{\mathbf{w}\in\mathcal{K}} \sum_{t=1}^T \ell_t(\mathbf{w})$ be the best fixed predictor in hindsight, then

$$\begin{aligned}
\ell_t(\mathbf{w}^t) - \ell_t(\mathbf{w}^*) &\leq \langle \mathbf{g}^t, \mathbf{w}^t - \mathbf{w}^* \rangle && \text{[Convexity]}\\
&= \frac{1}{\eta}\langle \nabla R(\mathbf{w}^t) - \nabla R(\mathbf{y}^{t+1}), \mathbf{w}^t - \mathbf{w}^* \rangle\\
&= \frac{1}{\eta}\langle \nabla R(\mathbf{y}^{t+1}) - \nabla R(\mathbf{w}^t), \mathbf{w}^* - \mathbf{w}^t \rangle\\
&= \frac{1}{\eta}\big(D_R(\mathbf{w}^* \,\|\, \mathbf{w}^t) + D_R(\mathbf{w}^t \,\|\, \mathbf{y}^{t+1}) - D_R(\mathbf{w}^* \,\|\, \mathbf{y}^{t+1})\big) && \text{[Lemma 22]}\\
&= \frac{1}{\eta}\big(D_R(\mathbf{w}^* \,\|\, \mathbf{w}^t) + D_R(\mathbf{w}^t \,\|\, \mathbf{y}^{t+1}) - D_R(\mathbf{w}^* \,\|\, \mathbf{w}^{t+1}) - D_R(\mathbf{w}^{t+1} \,\|\, \mathbf{y}^{t+1})\big) && \text{[Lemma 23]}\\
&= \frac{1}{\eta}\big(D_R(\mathbf{w}^* \,\|\, \mathbf{w}^t) - D_R(\mathbf{w}^* \,\|\, \mathbf{w}^{t+1})\big) + \frac{1}{\eta}\big(D_R(\mathbf{w}^t \,\|\, \mathbf{y}^{t+1}) - D_R(\mathbf{w}^{t+1} \,\|\, \mathbf{y}^{t+1})\big)\,.
\end{aligned}$$

Note that the left hand side term telescopes when summing over t, yielding

$$\sum_{t=1}^T \ell_t(\mathbf{w}^t) - \ell_t(\mathbf{w}^*) \leq \frac{D_R(\mathbf{w}^* \,\|\, \mathbf{w}^1)}{\eta} + \frac{1}{\eta}\sum_{t=1}^T \Big(D_R(\mathbf{w}^t \,\|\, \mathbf{y}^{t+1}) - D_R(\mathbf{w}^{t+1} \,\|\, \mathbf{y}^{t+1})\Big)\,. \quad (10)$$

If suffices to upper bound $D_R(\mathbf{w}^t \,\|\, \mathbf{y}^{t+1}) - D_R(\mathbf{w}^{t+1} \,\|\, \mathbf{y}^{t+1})$. We start by substituting the definition of the Bregman divergence in D_R,

$$\begin{aligned}
&D_R(\mathbf{w}^t \,\|\, \mathbf{y}^{t+1}) - D_R(\mathbf{w}^{t+1} \,\|\, \mathbf{y}^{t+1})\\
&\quad= R(\mathbf{w}^t) - R(\mathbf{w}^{t+1}) - \langle \nabla R(\mathbf{y}^{t+1}), \mathbf{w}^t - \mathbf{w}^{t+1} \rangle\\
&\quad\leq \langle \nabla R(\mathbf{w}^t), \mathbf{w}^t - \mathbf{w}^{t+1} \rangle - \frac{\mu}{2}\|\mathbf{w}^t - \mathbf{w}^{t+1}\|^2 - \langle \nabla R(\mathbf{y}^{t+1}), \mathbf{w}^t - \mathbf{w}^{t+1} \rangle && \text{[μ-strong convexity]}\\
&\quad= \langle \nabla R(\mathbf{w}^t) - \nabla R(\mathbf{y}^{t+1}), \mathbf{w}^t - \mathbf{w}^{t+1} \rangle - \frac{\mu}{2}\|\mathbf{w}^t - \mathbf{w}^{t+1}\|^2\\
&\quad= \eta\,\langle \mathbf{g}^t, \mathbf{w}^t - \mathbf{w}^{t+1} \rangle - \frac{\mu}{2}\|\mathbf{w}^t - \mathbf{w}^{t+1}\|^2 && \text{[Update rule]}\\
&\quad\leq \eta\,\|\mathbf{g}^t\|_*\,\|\mathbf{w}^t - \mathbf{w}^{t+1}\| - \frac{\mu}{2}\|\mathbf{w}^t - \mathbf{w}^{t+1}\|^2 && \text{[Cauchy-Schwarz]}\\
&\quad\leq \frac{\eta^2\|\mathbf{g}^t\|_*^2}{2\mu}\,.
\end{aligned}$$

The last step follows from maximizing the quadratic function in $\|\mathbf{w}^t - \mathbf{w}^{t+1}\|$. Using the above bound (10) completes the proof. ■

Appendix C. Regret Analysis

The HU and SHU are instances of Online Mirror Descent with $D^\beta_\phi(\cdot \,\|\, \cdot)$ and $D^\beta_\Phi(\cdot \,\|\, \cdot)$ divergences, respectively. We provide a simple regret analysis that follows directly from the geometric properties derived in Sec. 4 and Sec.5. The following theorem allows us to bound the regret of an OMD algorithm in terms of the diameter and strong convexity.

Theorem 24 *Assume that $R : \mathcal{K} \to \mathbb{R}$ is μ-strongly convex in respect to a norm $\| \cdot \|$ whose dual norm is $\| \cdot \|_*$. Assume that the diameter of $\mathcal{K}$ is bounded, $\mathrm{diam}_R(\mathcal{K}) \leq D$. Last, assume that $\forall t$, $\|\mathbf{g}^t\|_* \leq G$, then the regret bound of* HU *and learning rate $\eta = \sqrt{2\mu D/(TG^2)}$ satisfies*

$$\mathcal{R}_T \leq 2\sqrt{2\mu^{-1}DTG^2} \, .$$

We next provide regret bounds for HU over B_1 and B_2 and a regret bound for SHU over $B_{\mathrm{Tr}}(\tau)$.

Proof [Theorem 2] Applying Lemma 6 and the diameter bounds from (3) to Theorem 24 yields,

$$\mathcal{R}_T \leq 2\sqrt{2(1+\frac{1}{\beta})TG_2^2} \leq 4G_2\sqrt{T} \, .$$

The final inequality follows from the condition that $\beta \geq 1$. ■

Proof [Theorem 3] Applying Lemma 7 and the diameter bounds from (2) to Theorem 24 yields,

$$\mathcal{R}_T \leq 2G_\infty\sqrt{2T(1+\beta d)\log\left(\tfrac{3}{\beta}\right)} \leq 3G_\infty\sqrt{T(1+\beta d)\log\left(\tfrac{3}{\beta}\right)} \, .$$

■

Proof [Theorem 4] Applying (2) on the vector singular values, we have

$$\mathrm{diam}_{\Phi_\beta}(B_{\mathrm{Tr}}(\tau)) \leq \tau \log\left(\frac{3\tau}{\beta}\right)$$

where $\beta \leq \tau$. Furthermore, from Theorem 14, we can see that Φ_β is $(2(\tau + \beta\min\{m,n\}))^{-1}$ strongly convex with respect to the trace-norm over $B_{\mathrm{Tr}}(\tau)$. Letting $\gamma = \frac{\beta}{\tau}$, the result follows. ■

Appendix D. Diameter Calculation

In this appendix, we derive the diameter bounds from Sec .4. Whenever implied by the context we omit the potential ϕ from the diameter. Before we consider two specific diameters below, we bound the diameter in general as follows,

$$\begin{aligned}
D_\phi^\beta(\mathbf{x} \,\|\, \mathbf{0}) &= \phi_\beta(\mathbf{x}) - \phi_\beta(\mathbf{0}) \\
&= \sum_{i=1}^d \left(x_i \operatorname{arcsinh}(x_i/\beta) - \sqrt{x_i^2+\beta^2}\right) + \beta d \\
&\leq \sum_{i=1}^d x_i \operatorname{arcsinh}(x_i/\beta) \\
&= \sum_{i=1}^d |x_i| \log\left(\frac{1}{\beta}\left(\sqrt{x_i^2+\beta^2} + |x_i|\right)\right)
\end{aligned}$$

Thus, without loss of generality, we can assume that $\mathbf{x}$ lies in the positive orthant. We next bound the diameter of B_2 as follows,

$$\begin{aligned}\operatorname{diam}(B_2) &\leq \sum_{i=1}^{d} x_i \log\left(\frac{1}{\beta}\left(\sqrt{x_i^2+\beta^2}+x_i\right)\right)\\ &\leq \sum_{i=1}^{d} x_i \log\left(1+\frac{2x_i}{\beta}\right)\\ &\leq \sum_{i=1}^{d} \frac{2x_i^2}{\beta} = \frac{2\|\mathbf{x}\|_2^2}{\beta} \leq \frac{2}{\beta}\,. \end{aligned} \tag{11}$$

For $\beta \leq 1$ and $\mathbf{x} \in B_1$ it holds that, $\sqrt{x_i^2+\beta^2}+x_i \leq \sqrt{2}+1$. Hence, for $\beta \leq 1$, we have

$$\operatorname{diam}(B_1) \leq \sum_{i=1}^{d} x_i \log\left(\frac{1+\sqrt{2}}{\beta}\right) = \|\mathbf{x}\|_1 \log\left(\frac{1+\sqrt{2}}{\beta}\right) \leq \log\left(\frac{3}{\beta}\right)\,. \tag{12}$$

Appendix E. Local Duality of Smoothness and Strong-Convexity

Proof [Lemma 13] It suffices to show that for any $\mathbf{x} \in \mathcal{K}$, ϕ is locally $\frac{1}{L}$-strongly convex with respect to $\|\cdot\|$ at $\mathbf{x}$ if ϕ^* is locally L-smooth at $\mathbf{x}^* = \nabla\phi(\mathbf{x})$ with respect to $\|\cdot\|_*$.

From local smoothness at $\mathbf{x}^*$, we have for any $\mathbf{y} \in \mathbb{R}^d$,

$$f(\mathbf{y}) = \frac{1}{2}\mathbf{y}^\top \nabla^2\phi^*(\mathbf{x}^*)\mathbf{y} \leq \frac{L}{2}\|\mathbf{y}\|_*^2\,.$$

Taking the dual, which is order reversing, we have for any $\mathbf{z} \in \mathbb{R}^d$,

$$f^*(\mathbf{z}) = \frac{1}{2}\mathbf{z}^\top [\nabla^2\phi^*(\mathbf{x}^*)]^{-1}\mathbf{z} \geq \frac{1}{2L}\|\mathbf{z}\|^2\,. \tag{13}$$

Since $\nabla\phi^* = (\nabla\phi)^{-1}$, then from the inverse function theorem, we have that

$$\nabla^2\phi^*(\mathbf{x}^*) = [\nabla^2\phi(\mathbf{x})]^{-1}\,. \tag{14}$$

Using the above equality in (13), we have for any $\mathbf{z} \in \mathbb{R}^d$,

$$\frac{1}{2}\mathbf{z}^\top \nabla^2\phi(\mathbf{x})\mathbf{z} \geq \frac{1}{2L}\|\mathbf{z}\|^2\,.$$

■

Appendix F. Equivalence of HU and EG± without projection

Proof [Theorem 20] Note that in EG± without normalization we get,

$$u_i^{t+1} v_i^{t+1} = u_i^t \exp(-\eta g_i^t) v_i^t \exp(\eta g_i^t) = u_i^t v_i^t \, .$$

Therefore, $u_i^t v_i^t$ remains fixed and due to the initialization $\forall i, u_i^0 v_i^0 = \frac{\beta^2}{4}$. This inverse relationship between $\mathbf{u}$ and $\mathbf{v}$ can be used to find a simple closed form solution for $\mathbf{w}$ by solving a quadratic equation. In particular, we know $u > 0$, so we have

$$w = u - v = u - \frac{\beta^2}{4u} \;\Rightarrow\; u = \frac{w + \sqrt{w^2 + \beta^2}}{2} \, .$$

The resulting final update is

$$\begin{aligned} w_i^{t+1} &= \frac{\sqrt{(w_i^t)^2 + \beta^2} + w_i^t}{2} \exp(-\eta g_i^t) - \frac{\sqrt{(w_i^t)^2 + \beta^2} - w_i^t}{2} \exp(\eta g_i^t) \\ &= \sqrt{(w_i^t)^2 + \beta^2} \frac{\exp(-\eta g_i^t) - \exp(\eta g_i^t)}{2} + w_i^t \frac{\exp(-\eta g_i^t) + \exp(\eta g_i^t)}{2} \\ &= \sinh(-\eta g_i^t) \sqrt{(w_i^t)^2 + \beta^2} + \cosh(-\eta g_i^t) w_i^t \, . \end{aligned}$$

Now we consider HU algorithm with the same parameters,

$$\begin{aligned} \mathbf{w}^{t+1} &= \nabla\phi_\beta^{-1}(\nabla\phi_\beta(\mathbf{w}^t) - \eta \mathbf{g}^t) \quad \big[\text{HU update}\big] \\ \Rightarrow w_i^{t+1} &= \beta \sinh\left(\operatorname{arcsinh}\left(\frac{w_i^t}{\beta}\right) - \eta g_i^t\right) \\ &= \beta\Big[\sinh\left(\operatorname{arcsinh}\left(\frac{w_i^t}{\beta}\right)\right)\cosh(-\eta g_i^t) + \cosh\left(\operatorname{arcsinh}\left(\frac{w_i^t}{\beta}\right)\right)\sinh(-\eta g_i^t)\Big] \\ &= \sinh(-\eta g_i^t) \sqrt{(w_i^t)^2 + \beta^2} + \cosh(-\eta g_i^t) w_i^t \, . \end{aligned}$$

Thus indeed the two updates with the conditions stated in the theorem are equivalent. ■