2015 IEEE 54th Annual Conference on Decision and Control (CDC)
December 15-18, 2015. Osaka, Japan

Efficient Bregman Projections onto the Simplex

Walid Krichene Syrine Krichene Alexandre Bayen

***Abstract*— We consider the problem of projecting a vector onto the simplex $\Delta = \{x \in \mathbb{R}^d_+ : \sum_{i=1}^d x_i = 1\}$, using a Bregman projection. This is a common problem in first-order methods for convex optimization and online-learning algorithms, such as mirror descent. We derive the KKT conditions of the projection problem, and show that for Bregman divergences induced by ω-potentials, one can efficiently compute the solution using a bisection method. More precisely, an ϵ-approximate projection can be obtained in $\mathcal{O}(d \log \frac{1}{\epsilon})$. We also consider a class of exponential potentials for which the exact solution can be computed efficiently, and give a $\mathcal{O}(d \log d)$ deterministic algorithm and $\mathcal{O}(d)$ randomized algorithm to compute the projection. In particular, we show that one can generalize the KL divergence to a Bregman divergence which is bounded on the simplex (unlike the KL divergence), strongly convex with respect to the ℓ_1 norm, and for which one can still solve the projection in expected linear time.**

I. Introduction

Many first-order methods for convex optimization and online learning can be formulated as iterative projections of a vector on a feasible set. Consider for example the constrained convex problem, minimize$_{x\in\mathcal{X}} f(x)$, where $\mathcal{X}$ is a convex set and $f : \mathcal{X} \to \mathbb{R}$ is convex. This problem can be solved using the mirror descent algorithm, a first-order method proposed by Nemirovski and Yudin in [21] (see also [4]), which generalizes the projected gradient descent method, by replacing the Euclidean projection step with a generalized Bregman projection. This method can be summarized in Algorithm 1.

Algorithm 1 Mirror descent method with learning rates (η_τ) and Bregman divergence D_ψ.

1: **for** $\tau \in \mathbb{N}$ **do**
2: Query a sub-gradient vector $g^{(\tau)} \in \partial f(x^{(\tau)})$
3: Update

$$x^{(\tau+1)} = \arg\min_{x\in\mathcal{X}} D_\psi(x, (\nabla\psi)^{-1}(\nabla\psi(x^{(\tau)}) - \eta_\tau g^{(\tau)})) \quad (1)$$

4: **end for**

Here, D_ψ is the Bregman divergence induced by a distance generating function ψ. The definition and properties of Bregman divergences will be reviewed in Section II. Some important instances of the mirror descent method include projected gradient descent, obtained by taking the Bregman divergence to be the squared Euclidean distance, and the exponentiated gradient descent [18] (also called Hedge algorithm or multiplicative weights algorithm [1]), obtained by taking the Bregman divergence to be the KL divergence.

In this article, we focus specifically on simplex-constrained convex problems. That is, we suppose that $\mathcal{X}$ is the simplex $\Delta^d = \{x \in \mathbb{R}^d_+ : \sum_{i=1}^d x_i = 1\}$, or more generally, a product of scaled simplexes, $\mathcal{X} = \alpha_1\Delta^{d_1} \times \cdots \times \alpha_I\Delta^{d_I}$. Simplex-constrained problems include non-parametric statistical estimation, see for example Section 7.2 in [8], multi-commodity flow problems, see Chapter 12 in [10], tomography image reconstruction [5] and learning dynamics in repeated games [20]. Other variants of the mirror descent method have been studied as well, such as stochastic mirror descent [17], [19].

Besides its applications to convex optimization, simplex-constrained mirror descent plays an important role in online learning problems [9], in which a decision maker chooses, at each iteration τ, a distribution $x^{(\tau)}$ over a finite action set $\mathcal{A}$ with $|\mathcal{A}| = d$. Then, a bounded loss vector $\ell^{(\tau)} \in [0,1]^d$ is revealed, and the decision maker incurs expected loss $\left\langle \ell^{(\tau)}, x^{(\tau)} \right\rangle = \sum_{i=1}^d x_i^{(\tau)} \ell_i^{(\tau)}$. This sequential decision problem is also called prediction with expert advice [11], and has a long history which dates back to Hannan [15] and Blackwell [6], who studied this problem in the context of repeated games.

In (adversarial) online learning problems, one seeks to design an algorithm which has a guarantee on the worst-case regret, defined as follows: if the algorithm is presented with a sequence of losses $(\ell^{(\tau)})_{1\leq\tau\leq T}$, and it generates a sequence of decisions $(x^{(\tau)})_{1\leq\tau\leq T}$, then the cumulative regret of the algorithm up to iteration T is

$$R((\ell^{(\tau)})_{0\leq\tau\leq T}) = \sum_{\tau=1}^T \left\langle \ell^{(\tau)}, x^{(\tau)} \right\rangle - \min_{x\in\Delta} \sum_{\tau=1}^T \left\langle \ell^{(\tau)}, x \right\rangle,$$

and the worst-case regret is the maximum such regret over admissible sequences of losses $\max_{(\ell^{(\tau)})_{0\leq\tau\leq T}} R((\ell^{(\tau)})_{0\leq\tau\leq T})$. An algorithm is said to have sublinear regret if its worst-case regret grows sub-linearly in T, that is, $\limsup_{T\to\infty} \max_{(\ell^{(\tau)})_{0\leq\tau\leq T}} \frac{R((\ell^{(\tau)})_{0\leq\tau\leq T})}{T} \leq 0$. The online mirror descent method, obtained simply by replacing the subgradient vector $g^{(\tau)}$ in Algorithm 1 with the loss vector $\ell^{(\tau)}$, defines a large class of online learning algorithms with sub-linear regret, see for example the survey of Bubeck and Cesa-Bianchi in [9]. The online

Walid Krichene is with the department of Electrical Engineering and Computer Sciences, University of California, Berkeley, USA. walid@eecs.berkeley.edu
Syrine Krichene is with the ENSIMAG school of Computer Sciences and Applied Mathematics of Grenoble, France. syrine.krichene@ensimag.grenoble-inp.fr
Alexandre Bayen is with the department of Electrical Engineering and Computer Sciences, and the department of Civil and Environmental Engineering, University of California, Berkeley, USA. bayen@berkeley.edu

mirror descent method is summarized in Algorithm 2.

Algorithm 2 Online mirror descent method with learning rates (η_τ) and Bregman divergence D_ψ.

1: **for** $\tau \in \mathbb{N}$ **do**
2: Play action $a^{(\tau)} \sim x^{(\tau)}$
3: Discover loss vector $\ell^{(\tau)} \in [0,1]^d$
4: Incur expected loss $\left\langle \ell^{(\tau)}, x^{(\tau)} \right\rangle$
5: Update

$$x^{(\tau+1)} = \arg\min_{x \in \Delta} D_\psi(x, (\nabla\psi)^{-1}(\nabla\psi(x^{(\tau)}) - \eta_\tau \ell^{(\tau)})) \tag{2}$$

6: **end for**

Online mirror descent, and its stochastic variant, have been applied to several problems including multi-armed bandits [9], [2], machine learning [12] and repeated games [11], to cite a few.

In all the variants of simplex-constrained mirror descent, one needs to solve, at each iteration τ, the Bregman projection step given in equation (1) or (2). Some instances of Bregman projections are known to have an exact solution which can be computed efficiently. For example, the solution of the KL divergence projection on the simplex is given by the exponential weights update [21], [3], and the Euclidean projection on the simplex can be computed efficiently either by sorting and thresholding in $\mathcal{O}(d \log d)$, or by using a randomized pivot method in $\mathcal{O}(d)$, see [13].

In this article, we start by deriving the KKT conditions of the Bregman projection problem in Section II, then consider, in Section III, a general class of Bregman divergences, induced by ω-potentials, as defined by Audibert et al. [2]. We show that for this class, the solution can be approximated efficiently: an ϵ-approximate solution can be computed in $\mathcal{O}(d \log \frac{1}{\epsilon})$ operations. In Section IV, we consider a class of exponential potentials, and study the resulting Bregman projection, a generalization of the KL-divergence projection. We show that for this class, the exact solution can be computed using a deterministic algorithm with $\mathcal{O}(d \log d)$ complexity, or a randomized algorithm with expected linear complexity. We also study the properties of the resulting Bregman divergence. In particular, we emphasize a tradeoff between strong convexity and boundedness, two properties which affect the convergence rates of the mirror descent method.

II. Bregman projection and optimality conditions

Let $\psi : \mathcal{X} \to \mathbb{R}$ be a convex function defined on a convex set $\mathcal{X}$, and let $\mathring{\mathcal{X}}$ be the subset of $\mathcal{X}$ on which ψ is differentiable. Let $\nabla\psi : \mathring{\mathcal{X}} \to \mathcal{R}$ be the gradient of ψ, and $\mathcal{R}$ its range. The Bregman divergence induced by ψ is defined as follows

$$\begin{aligned} D_\psi : \mathcal{X} \times \mathring{\mathcal{X}} &\to \mathbb{R}_+ \\ (x,y) &\mapsto D_\psi(x,y) = \psi(x) - \psi(y) - \langle \nabla\psi(y), x - y \rangle \end{aligned} \tag{3}$$

By convexity of ψ, the Bregman divergence is non-negative, and $x \mapsto D_\psi(x,y)$ is convex. We will refer to ψ as the distance-generating function. We say that ψ is ℓ_ψ-strongly convex with respect to a reference norm $\|\cdot\|$ if

$$D_\psi(x,y) \geq \frac{\ell_\psi}{2}\|x-y\|^2 \ \forall x, y \in \mathcal{X} \times \mathring{\mathcal{X}}.$$

In order for the Bregman projection (1) to be well-defined, the gradient vector (or loss vector) at iteration τ must satisfy the following consistency condition:

$$\nabla\psi(x^{(\tau)}) - \eta_\tau g^{(\tau)} \in \mathcal{R}. \tag{4}$$

A. Interpretations of the Bregman projection

The Bregman projection, given in equation (1), can be interpreted as projecting on $\mathcal{X}$, the vector $(\nabla\psi)^{-1}(\nabla\psi(x^{(\tau)}) - \eta_\tau g^{(\tau)})$, obtained by mapping the current iterate $x^{(\tau)}$ to the set $\mathcal{R}$ through $\nabla\psi$, taking a step in the opposite direction of the gradient, then mapping the new vector back through $(\nabla\psi)^{-1}$, see Nemirovski and Yudin [21].

A second interpretation can be obtained, as observed by Beck and Teboulle [3], by rewriting the objective function as follows: denoting the vector $(\nabla\psi)^{-1}(\nabla\psi(x^{(\tau)}) - \eta_\tau g^{(\tau)})$ by $\tilde{x}^{(\tau)}$, we have by definition of D_ψ

$$\begin{aligned} x^{(t+1)} &= \arg\min_{x \in \Delta} D_\psi(x, \tilde{x}^{(\tau)}) \\ &= \arg\min_{x \in \Delta} \psi(x) - \psi(\tilde{x}^{(\tau)}) - \left\langle \nabla\psi(\tilde{x}^{(\tau)}), x - \tilde{x}^{(\tau)} \right\rangle \\ &= \arg\min_{x \in \Delta} \psi(x) - \left\langle \nabla\psi(x^{(\tau)}) - \eta_\tau g^{(\tau)}, x \right\rangle, \end{aligned}$$

which is equivalent to minimizing

$$x^{(\tau+1)} = \arg\min_{x \in \Delta} \eta_\tau \left(f(x^{(\tau)}) + \left\langle g^{(\tau)}, x - x^{(\tau)} \right\rangle \right) + D_\psi(x, x^{(\tau)}),$$

which can be interpreted as follows: the first term $f(x^{(\tau)}) + \left\langle g^{(\tau)}, x - x^{(\tau)} \right\rangle$ is the linear approximation of f around the current iterate $x^{(\tau)}$, and the second term $D_\psi(x, x^{(\tau)})$ is a non-negative function which penalizes deviations from $x^{(\tau)}$. The step size (or learning rate) η_τ, controls the relative weight of both terms.

B. Simplex-constrained Bregman projection

In the remainder of the paper, we will assume, to simplify the discussion, that the feasible set is the simplex $\Delta^d = \{x \in \mathbb{R}_+^d : \sum_{i=1}^d x_i = 1\}$.

We observe that all the results can be readily extended to the case in which $\mathcal{X}$ is a product of scaled simplexes, as follows: suppose $\mathcal{X} = \alpha_1 \Delta^{d_1} \times \cdots \times \alpha_K \Delta^{d_K}$, with $\alpha_k > 0$, and let ψ_k be a distance generating function on Δ^{d_k}. Then consider the function

$$\begin{aligned} \psi : \alpha_1 \Delta^{d_1} \times \cdots \times \alpha_K \Delta^{d_K} &\to \mathbb{R} \\ (\alpha_1 x_1, \ldots, \alpha_K x_K) &\mapsto \sum_{k=1}^K \alpha_k \psi_k(x_k). \end{aligned}$$

The gradient of ψ is simply $\nabla\psi : \alpha_1 \mathring{\Delta}^{d_1} \times \cdots \times \alpha_K \mathring{\Delta}^{d_K} \to \mathcal{R}_1 \times \cdots \times \mathcal{R}_K$, $(\alpha_1 x_1, \ldots, \alpha_K x_K) \mapsto (\nabla\psi_1(x_1), \ldots, \nabla\psi_K(x_K))$, and its inverse is given by

$(\nabla\psi)^{-1} : \mathcal{R}_1 \times \dots \mathcal{R}_K \to \alpha_1 \mathring{\Delta}^{d_1} \times \cdots \times \alpha_K \mathring{\Delta}^{d_K}$, $(y_1, \dots, y_K) \mapsto (\alpha_1 \nabla\psi_1^{-1}(y_1), \dots, \alpha_K \nabla\psi_K^{-1}(y_K))$. Finally, the Bregman divergence decomposes as follows

$$\begin{aligned} &D_\psi((\alpha_k x_k)_k, (\alpha_k y_k)_k) \\ &= \sum_k \alpha_k \psi_k(x_k) - \sum_k \alpha_k \psi_k(y_k) - \sum_k \langle \nabla\psi(y_k), \alpha_k(x_k - y_k)\rangle \\ &= \sum_k \alpha_k D_{\psi_k}(x_k, y_k). \end{aligned}$$

Therefore, the projection on $\mathcal{X}$ with Bregman divergence D_ψ can be decomposed into K projections on Δ^{d_k} with Bregman divergence D_{ψ_k}, as follows:

$$\begin{aligned} &\operatorname*{arg\,min}_{x_k \in \Delta^{d_k}} D_\psi(x, (\nabla\psi)^{-1}(\nabla\psi(x^{(\tau)}) - \eta_\tau g^{(\tau)})) \\ &= \operatorname*{arg\,min}_{x_k \in \Delta^{d_k}} \sum_i \alpha_k D_{\psi_k}(x_k, \nabla\psi_k^{-1}(\nabla\psi_k(x_k^{(\tau)}) - \eta_\tau g_k^{(\tau)})), \end{aligned}$$

assuming the consistency condition holds for each k.

Example 1 (Euclidean projection): Consider the function $\psi(x) = \frac{1}{2}\|x\|_2^2$. Then $\nabla\psi(x) = x$, and the Bregman divergence is simply $D_\psi(x, y) = \frac{1}{2}\|x - y\|_2^2$. As a consequence, the Bregman projection step reduces to

$$\begin{aligned} &\operatorname*{arg\,min}_{x \in \Delta^d} D_\psi(x, (\nabla\psi)^{-1}(\nabla\psi(x^{(\tau)}) - \eta_\tau g^{(\tau)})) \\ &= \operatorname*{arg\,min}_{x \in \Delta^d} \frac{1}{2}\|x - (x^{(\tau)} - \eta_\tau g^{(\tau)})\|_2^2, \end{aligned}$$

which corresponds to a projected gradient descent update, with step size η_τ.

C. Optimality conditions

We now derive the KKT conditions for the Bregman projection problem given by

$$\begin{aligned} &\text{minimize}_{x \in \mathbb{R}^d} \quad D_\psi(x, (\nabla\psi)^{-1}(\nabla\psi(\bar{x}) - \bar{g})) \\ &\text{subject to} \qquad x \in \Delta^d \end{aligned} \tag{5}$$

where, $\bar{x} \in \Delta^d$, and $\bar{g} \in \mathbb{R}^d$ are given. Note that we combine $\eta_\tau g^{(\tau)}$ into a single vector $\bar{g}$, to simplify notation. By strong convexity, the solution is unique.

Proposition 1: Consider the Bregman projection problem (5). Then $x^\star \in \mathbb{R}^d$ is optimal if and only if there exist $\lambda^\star \in \mathbb{R}_+^d$ and $\nu^\star \in \mathbb{R}$ such that

$$\begin{cases} x^\star = (\nabla\psi)^{-1}(\nabla\psi(\bar{x}) - \bar{g} + \lambda^\star + \boldsymbol{\nu}^\star), \\ \sum_{i=1}^d x_i^\star = 1, \\ \forall i, \quad x_i^\star \geq 0, \quad \lambda_i^\star x_i^\star = 0, \end{cases}$$

where $\boldsymbol{\nu}^\star$ is the vector whose entries are all equal to $\nu^\star$.

Proof: Define the Lagrangian, for $x \in \mathbb{R}^d$, $\lambda \in \mathbb{R}_+^d$, and $\nu \in \mathbb{R}$,

$$\begin{aligned} \mathcal{L}(x, \lambda, \nu) = D_\psi(x, (\nabla\psi)^{-1}(\nabla\psi(\bar{x}) - \bar{g})) \\ - \langle \lambda, x\rangle + \nu(1 - \sum_{i=1}^d x_i). \end{aligned} \tag{6}$$

For all $x, y \in \mathring{\mathcal{X}}$, the gradient of the Bregman divergence is given by

$$\nabla_x D_\psi(x, y) = \nabla\psi(x) - \nabla\psi(y).$$

Thus the gradient of $\mathcal{L}$ is given by

$$\nabla_x \mathcal{L}(x, \lambda, \nu) = \nabla\psi(x) - \nabla\psi(\bar{x}) + \bar{g} - \lambda - \boldsymbol{\nu}.$$

Writing the KKT conditions of problem (5), we have that $(x^\star, \lambda^\star, \nu^\star)$ is optimal if and only if

$$\begin{cases} \nabla\psi(x^\star) - \nabla\psi(\bar{x}) + \bar{g} - \lambda^\star - \boldsymbol{\nu}^\star = 0, \\ \sum_i x_i^\star = 1, \\ \forall i, \quad x_i^\star \geq 0, \quad \lambda_i^\star \geq 0, \quad \lambda_i^\star x_i^\star = 0, \end{cases}$$

and the first equation can be rearranged as $x^\star = (\nabla\psi)^{-1}(\nabla\psi(\bar{x}) - \bar{g} + \lambda^\star + \boldsymbol{\nu}^\star)$, which proves the claim. ∎

In the next section, we will derive an efficient algorithm to compute an approximate solution for the class of Bregman divergences induced by ω-potentials, by solving the KKT system given in Proposition 1.

III. Efficient approximate projection with ω-potentials

Definition 1: Let $a \in (-\infty, +\infty]$ and $\omega \leq 0$. An increasing, C^1-diffeomorphism $\phi : (-\infty, a) \to (\omega, +\infty)$ is called an ω-potential if

$$\lim_{u \to -\infty} \phi(u) = \omega, \quad \lim_{u \to a} \phi(u) = +\infty, \quad \int_0^1 \phi^{-1}(u)du < \infty.$$

We associate, to an ω-potential ϕ, the distance-generating

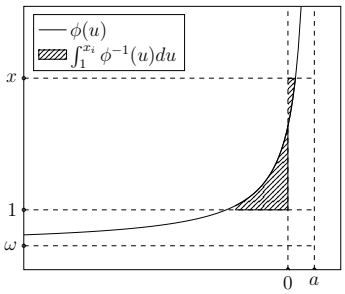

Fig. 1. Illustration of an ω-potential

function ψ defined as follows

$$\begin{aligned} \psi : (\omega, +\infty)^d &\to \mathbb{R} \\ x &\mapsto \sum_{i=1}^d \int_1^{x_i} \phi^{-1}(u)du. \end{aligned}$$

By definition, ψ is finite (in particular, the third condition on the potential ensures that ψ is finite on the boundary of the simplex since $\int_1^0 \phi^{-1}(u)du < \infty$), differentiable on $(\omega, +\infty)^d$, and its gradient is given by

$$\begin{aligned} \nabla\psi : (\omega, \infty)^d &\to \mathcal{R} = (-\infty, a)^d \\ x &\mapsto \nabla\psi(x) = (\phi^{-1}(x_i))_{i=1,\dots,d}, \end{aligned}$$

and since ϕ^{-1} in increasing, ψ is convex. Similarly, the inverse of its gradient is

$$\begin{aligned} (\nabla\psi)^{-1} : (-\infty, a)^d &\to (\omega, \infty)^d \\ y &\mapsto (\phi(y_i))_{i=1,\dots,d}. \end{aligned}$$

Proposition 2: Consider the Bregman projection onto the simplex given in Problem (5), and assume that ψ is induced by an ω-potential ϕ. Then $x^\star$ is a solution if and only if there exists $\nu^\star \in \mathbb{R}$ such that

$$\begin{cases} \forall i, \quad x_i^\star = \left(\phi(\phi^{-1}(\bar{x}_i) - \bar{g}_i + \nu^\star)\right)_+, \\ \sum_{i=1}^d x_i^\star = 1, \end{cases}$$

where x_+ denoted the positive part of x, $x_+ = \max(x, 0)$.

Proof: Combining the expression of $\nabla\psi$ and $(\nabla\psi)^{-1}$ with Proposition 1, we have that $x^\star$ is optimal if and only if there exist $\nu^\star \in \mathbb{R}$ and $\lambda^\star \in \mathbb{R}^d_+$ such that

$$\begin{cases} \forall i, \quad x_i^\star = \phi(\phi^{-1}(\bar{x}_i) - \bar{g}_i + \nu^\star + \lambda_i^\star), \\ \sum_{i=1}^d x_i^\star = 1, \\ \forall i, \quad x_i^\star \geq 0, \quad x_i^\star \lambda_i^\star = 0. \end{cases}$$

Let $\mathcal{I} = \{i : x_i^\star > 0\}$ be the support of $x^\star$. Then by the complementary slackness condition, we have for all $i \in \mathcal{I}$, $\lambda_i^\star = 0$, thus $x_i^\star = \phi(\phi^{-1}(\bar{x}_i) - \bar{g}_i + \nu^\star)$, and for all $i \notin \mathcal{I}$,

$$\begin{aligned} &\phi(\phi^{-1}(\bar{x}_i) - \bar{g}_i + \nu^\star) \\ &\quad \leq \phi(\phi^{-1}(\bar{x}_i) - \bar{g}_i + \nu^\star + \lambda_i^\star) \quad \text{since } \phi \text{ is increasing} \\ &\quad = x_i^\star = 0. \end{aligned}$$

Therefore $x_i^\star$ can be simply written $x_i^\star = \left(\phi(\phi^{-1}(\bar{x}_i) - \bar{g}_i + \nu^\star)\right)_+$ which proves the claim. ∎

Next, we make the following observation regarding the support of the solution:

Proposition 3: Let $x^\star$ be the solution to the projection problem (5), and let $\mathcal{I}$ be its support. Then for all i, j, if $i \in \mathcal{I}$ and $\phi^{-1}(\bar{x}_i) - \bar{g}_i \leq \phi^{-1}(\bar{x}_j) - \bar{g}_j$, then $j \in \mathcal{I}$.

Proof: Follows from Proposition 2 and the fact that ϕ is increasing. ∎

As a consequence of the previous propositions, computing the projection reduces to computing the optimal dual variable $\nu^\star$, and since the potential is increasing, one can iteratively approximate $\nu^\star$ using a bisection method, given in Algorithm 3: we start by defining a bound on the optimal $\nu^\star$, $\underline{\nu} \leq \nu^\star \leq \bar{\nu}$, then we iteratively halve the size of the interval by inspecting the value of a carefully defined criterion function.

Theorem 1: Consider the Bregman projection onto the simplex given in Problem (5), and assume that ψ is induced by an ω-potential ϕ. Let $\epsilon > 0$, and consider the bisection method given in Algorithm 3. Then the Algorithm terminates after $T = \mathcal{O}(\log \frac{1}{\epsilon})$ steps, and its output $\tilde{x}(\bar{\nu}^{(T)})$ is such that

$$\|\tilde{x}(\bar{\nu}^{(T)}) - x^\star\|_1 \leq \epsilon.$$

Each step of the algorithm has complexity $\mathcal{O}(d)$, thus the total complexity is $\mathcal{O}\left(d \log \frac{1}{\epsilon}\right)$.

Proof: Define, as in Algorithm 3, the function

$$\tilde{x}(\nu) = \left(\phi(\phi^{-1}(\bar{x}_i) - \bar{g}_i + \nu)_+\right)_{i=1,\dots,d}.$$

Since ϕ is, by assumption, increasing, so is $\nu \mapsto \tilde{x}_i(\nu)$, which is the key fact that allows us to use a bisection.

We will denote by a superscript (t) the value of each variable at iteration t of the loop. To prove the claim, we show the following invariant for t:

Algorithm 3 Bisection method to compute the projection $x^\star$ with precision ϵ.

1: Input: $\bar{x}, \bar{g}, \epsilon$.
2: Initialize

$$\bar{\nu} = \phi^{-1}(1) - \max_i \phi^{-1}(\bar{x}_i) - \bar{g}_i$$
$$\underline{\nu} = \phi^{-1}(1/d) - \max_i \phi^{-1}(\bar{x}_i) - \bar{g}_i$$

3: Define $\tilde{x}(\nu) = \left(\phi(\phi^{-1}(\bar{x}_i) - \bar{g}_i + \nu)_+\right)_{i=1,\dots,d}$
4: **while** $\|\tilde{x}(\bar{\nu}) - \tilde{x}(\underline{\nu})\|_1 > \epsilon$ **do**
5: Let $\nu^+ \leftarrow \frac{\bar{\nu}+\underline{\nu}}{2}$
6: **if** $\sum_i \tilde{x}_i(\nu^+) > 1$ **then**
7: $\bar{\nu} \leftarrow \nu^+$
8: **else**
9: $\underline{\nu} \leftarrow \nu^+$
10: **end if**
11: **end while**
12: Return $\tilde{x}(\bar{\nu})$

(i) $0 \leq \bar{\nu}^{(t)} - \underline{\nu}^{(t)} \leq \frac{\bar{\nu}^{(0)} - \underline{\nu}^{(0)}}{2^t}$,
(ii) $\forall i,\ 0 \leq \tilde{x}_i(\underline{\nu}^{(t)}) \leq \tilde{x}_i(\underline{\nu}^{(t)}) \leq 1$,
(iii) $\sum_{i=1}^d \tilde{x}_i(\underline{\nu}^{(t)}) \leq 1 \leq \sum_{i=1}^d \tilde{x}_i(\bar{\nu}^{(t)})$.

We first prove the invariant for $t = 0$. Let $i_0 = \arg\max_i \phi^{-1}(\bar{x}_i) - \bar{g}_i$. By definition of $\bar{\nu}^{(0)}$ and $\underline{\nu}^{(0)}$, we have

$$\phi^{-1}(1/d) - \underline{\nu} = \phi^{-1}(\bar{x}_{i_0}) - \bar{g}_{i_0} = \phi^{-1}(1) - \bar{\nu}, \quad (7)$$

and it follows that $\tilde{x}_{i_0}(\underline{\nu}^{(0)}) = \frac{1}{d}$ and $\tilde{x}_{i_0}(\bar{\nu}^{(0)}) = 1$. By (7), $\bar{\nu}^{(0)} - \underline{\nu}^{(0)} = \phi^{-1}(1) - \phi^{-1}(1/d) \geq 0$ (since ϕ^{-1} is increasing), which proves (i). Next, since $\nu \mapsto \tilde{x}_i(\nu)$ is increasing, we have

$$0 \leq \tilde{x}_i(\underline{\nu}^{(0)}) \leq \tilde{x}_i(\bar{\nu}^{(0)}) \leq \tilde{x}_{i_0}(\bar{\nu}^{(0)}) = 1,$$

which proves (ii). Finally, we have

$$\sum_{i=1}^d \tilde{x}_i(\underline{\nu}^{(0)}) \leq d\tilde{x}_{i_0}(\underline{\nu}^{(0)}) = 1,$$
$$\sum_{i=1}^d \tilde{x}_i(\bar{\nu}^{(0)}) \geq \tilde{x}_{i_0}(\bar{\nu}^{(0)}) = 1,$$

which proves (iii). This proves the invariant for $t = 0$. Now suppose it holds at iteration t, and let us prove it still holds at $t + 1$. By definition of the bisection (lines 5–10), we immediately have

$$\bar{\nu}^{(t+1)} - \underline{\nu}^{(t+1)} = \frac{\bar{\nu}^{(t)} - \underline{\nu}^{(t)}}{2} = \frac{1}{2}\frac{\bar{\nu}^{(0)} - \underline{\nu}^{(0)}}{2^t},$$

which proves (i). We also have that $\underline{\nu}^{(t)} \leq \underline{\nu}^{(t+1)} \leq \bar{\nu}^{(t+1)} \leq \bar{\nu}^{(t)}$, which proves (ii) since $\nu \mapsto \tilde{x}_i(\nu)$ is increasing. Finally, (iii) follows from the condition of the bisection (line 6).

To conclude the proof, we simply observe that since the distance $|\bar{\nu} - \underline{\nu}|$ decreases exponentially, the algorithm will terminate after a number of steps logarithmic in $1/\epsilon$. Indeed, since ϕ is C^1 on $(-\infty, a)$, it is Lipschitz-continuous on

$[\phi^{-1}(0), \phi^{-1}(1)]$. Let L be its Lipschitz constant, then

$$\begin{aligned}\|\tilde{x}(\underline{\nu}^{(t)}) - \tilde{x}(\bar{\nu}^{(t)})\|_1 &= \sum_{i=1}^{d} |\tilde{x}_i(\underline{\nu}^{(t)}) - \tilde{x}_i(\bar{\nu}^{(t)})| \\ &\leq dL|\underline{\nu}^{(t)} - \bar{\nu}^{(t)}| \\ &= \frac{dL|\underline{\nu}^{(0)} - \bar{\nu}^{(0)}|}{2^t} \qquad \text{by (ii)},\end{aligned}$$

thus the algorithm terminates after $T = \log_2 \frac{|\underline{\nu}^{(0)} - \bar{\nu}^{(0)}|}{\epsilon dL}$ iterations, and the last iterate satisfies

$$\begin{aligned}&\|\tilde{x}(\nu^\star) - \tilde{x}(\bar{\nu}^{(T)})\|_1 \\ &\leq \|\tilde{x}(\underline{\nu}^{(T)}) - \tilde{x}(\bar{\nu}^{(t)})\|_1 \quad \text{by (iii) and since } \tilde{x}_i \text{ are increasing} \\ &\leq \epsilon,\end{aligned}$$

which concludes the proof. ∎

IV. Efficient exact projection with exponential potentials

We now consider a subclass of ω-potentials, for which we derive the exact solution.

Definition 2 (Exponential potential): Let $\epsilon \geq 0$. The function

$$\begin{aligned}\phi_\epsilon : (-\infty, +\infty) &\to (-\epsilon, +\infty) \\ u &\mapsto e^{u-1} - \epsilon,\end{aligned}$$

is called the exponential potential with parameter ϵ. It is a $(-\epsilon)$-potential.

The distance generating function induced by this class of potentials is given by

$$\begin{aligned}\psi_\epsilon(x) &= \sum_{i=1}^{d} \int_1^{x_i} \phi_\epsilon^{-1}(u)du = \sum_{i=1}^{d} \int_1^{x_i} 1 + \ln(u+\epsilon)du \\ &= \sum_{i=1}^{d} (x_i + \epsilon)\ln(x_i + \epsilon) - (1+\epsilon)\ln(1+\epsilon) \\ &= H(x + \boldsymbol{\epsilon}) - H(\mathbf{1} + \boldsymbol{\epsilon}),\end{aligned}$$

where $\boldsymbol{\epsilon}$ is the vector whose entries are all equal to ϵ, and H is the generalized negative entropy function, defined on $\mathbb{R}^d_+$

$$H(x) = \textstyle\sum_{i=1}^d x_i \ln x_i.$$

The corresponding Bregman divergence is

$$\begin{aligned}D_{\psi_\epsilon}(x, y) &= H(x + \boldsymbol{\epsilon}) - H(y + \boldsymbol{\epsilon}) - \langle \nabla H(y + \boldsymbol{\epsilon}), x - y\rangle \\ &= D_{KL}(x + \boldsymbol{\epsilon}, y + \boldsymbol{\epsilon}) \\ &= \sum_{i=1}^{d} (x_i + \epsilon)\ln\frac{x_i + \epsilon}{y_i + \epsilon},\end{aligned}$$

and will be denoted $D_{KL,\epsilon}(x, y)$. In particular, when $\epsilon = 0$, $D_{KL,\epsilon}(x, y)$ is the KL divergence between the distribution vectors x and y. When $\epsilon > 0$, the Bregman divergence is the KL divergence between $x + \boldsymbol{\epsilon}$ and $y + \boldsymbol{\epsilon}$. In particular, as we will see in Proposition 6, $D_{KL,\epsilon}(x, y)$ is bounded whenever $\epsilon > 0$, while the KL divergence ($\epsilon = 0$) can be unbounded.

As mentioned in the introduction, projecting on the simplex with the KL divergence plays a central role in many applications such as online learning. In particular, the projection problem can be solved exactly in $\mathcal{O}(d)$ operations, which makes this projection efficient. However, some variants of mirror descent, such as stochastic mirror descent, require the Bregman divergence to be bounded on the simplex in order to have guarantees on the convergence rate, see for example [14]. In the remainder of this section, we will show that projecting with the generalized KL divergence $D_{KL,\epsilon}$ enjoys many desirable properties (strong convexity with respect to the ℓ_1 norm, boundedness), and the projection can still be computed efficiently.

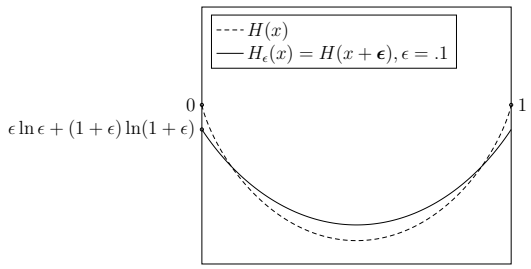

Fig. 2. Illustration of the distance generating function induced by exponential potentials with parameter ϵ, for $d = 2$: $H(x) = x_1 \ln(x_1) + (1 - x_1)\ln(1 - x_1)$.

A. A sorting algorithm to compute the exact projection

We first apply the optimality conditions of Proposition 2 to this special class, and show that the solution is entirely determined by its support.

Proposition 4: Consider the Bregman projection onto the simplex given in Problem (5), with Bregman divergence $D_{KL,\epsilon}$. Let $x^\star$ be the solution and $\mathcal{I} = \{i : x_i^\star > 0\}$ its support. Then

$$\begin{cases}\forall i \in \mathcal{I}, \quad x_i^\star = -\epsilon + \frac{(\bar{x}_i + \epsilon)e^{-\bar{g}_i}}{Z^\star}, \\ Z^\star = \frac{\sum_{i\in\mathcal{I}}(\bar{x}_i + \epsilon)e^{-\bar{g}_i}}{1 + |\mathcal{I}|\epsilon}.\end{cases} \tag{8}$$

Proof: Applying Proposition 2 with the expression $\phi(u) = e^{u-1} + \epsilon$ and $\phi^{-1}(u) = 1 + \ln(u + \epsilon)$, $x^\star$ is a solution if and only if there exists $\nu^\star \in \mathbb{R}$ such that $\forall i$, $x_i^\star = \left(-\epsilon + (\bar{x}_i + \epsilon)e^{-\bar{g}_i}e^{\nu^\star}\right)_+$, and $\sum_i x_i^\star = 1$. Thus, if $\mathcal{I}$ is the support of $x^\star$, then these optimality conditions are equivalent to

$$\begin{cases}\forall i \in \mathcal{I}, \quad x_i^\star = -\epsilon + (\bar{x}_i + \epsilon)e^{-\bar{g}_i}e^{\nu^\star}, \\ \sum_{i\in\mathcal{I}} -\epsilon + (\bar{x}_i + \epsilon)e^{-\bar{g}_i}e^{\nu^\star} = 1,\end{cases}$$

and the second equation can be rewritten as

$$1 + \epsilon|\mathcal{I}| = e^{\nu^\star} \textstyle\sum_{i\in\mathcal{I}} (\bar{x}_i + \epsilon)e^{-\bar{g}_i},$$

which proves the claim, with $Z^\star = e^{-\nu^\star}$. ∎

Proposition 4 shows that solving the Bregman projection with generalized KL divergence reduces to finding the support of the solution. Next, we show that the support has a simple characterization. To this end, we associate to $(\bar{x}, \bar{g})$ the vector $\bar{y}$ defined as follows

$$\forall i, \ \bar{y}_i = (\bar{x}_i + \epsilon)e^{-\bar{g}_i},$$

and we denote by $\bar{y}_{\sigma(i)}$ the i-th largest element of $\bar{y}$.

Algorithm 4 Sorting method to compute the Bregman projection with D_{ψ_ϵ}

Input: $\bar{x}$, $\bar{g}$
Output: $x^\star$
Form the vector $\bar{y}_i = (\bar{x}_i + \epsilon)e^{-\bar{g}_i}$
Sort y, let $\bar{y}_{\sigma(i)}$ be the i-th smallest element of y.
Let $j^\star$ be the smallest index for which

$$c(j) := (1 + \epsilon(d - j + 1))\bar{y}_{\sigma(j)} - \epsilon \sum_{i \geq j} \bar{y}_{\sigma(i)} > 0$$

Set $Z = \frac{\sum_{i \geq j^\star} \bar{y}_{\sigma(i)}}{1+\epsilon(d-j^\star+1)}$
Set

$$x_i^\star = \left(-\epsilon + \frac{\bar{y}_i}{Z(j^\star)} \right)_+$$

Proposition 5: The function

$$c(j) \mapsto (1 + \epsilon(d - j + 1))\bar{y}_{\sigma(j)} - \epsilon \sum_{i \geq j} \bar{y}_{\sigma(i)}$$

is increasing, and the support of $x^\star$ is $\{\sigma(j^\star), \ldots, \sigma(n)\}$, where $j^\star = \min\{j : c(j) > 0\}$.

Proof: First, straightforward algebra shows that

$$c(j+1) - c(j) = (1 + \epsilon(d - j))(\bar{y}_{\sigma(j+1)} - \bar{y}_{\sigma(j)}) \geq 0.$$

Thus c is increasing. To prove the second part of the claim, we know by Proposition 3 that the support is $\{\sigma(i^\star), \ldots, \sigma(n)\}$ for some $i^\star$, and to show that $i^\star = j^\star = \min\{j : c(j) > 0\}$, it suffices to show that $c(i^\star) > 0$ and $c(j) \leq 0$ for all $j < i^\star$. First, by the expression (8) of $x^\star$, we have

$$x^\star_{\sigma(i^\star)} = -\epsilon + \frac{\bar{y}_{\sigma(i^\star)}}{\frac{\sum_{i \geq i^\star} \bar{y}_{\sigma(i)}}{1+\epsilon(d-i^\star+1)}} > 0,$$

which is equivalent to $c(i^\star) > 0$. And if $j < i^\star$ (i.e. $\sigma(j)$ is outside the support), then by the expression (8) again,

$$0 = x^\star_{\sigma(j)} \geq -\epsilon + \frac{\bar{y}_{\sigma(j)}}{\frac{\sum_{i \geq i^\star} \bar{y}_{\sigma(i)}}{1+\epsilon(d-i^\star+1)}}$$

which is equivalent to

$$(1 + \epsilon(d - i^\star - 1))\bar{y}_{\sigma(j)} - \epsilon \sum_{i \geq i^\star} \bar{y}_{\sigma(i)} \leq 0,$$

but $c(j)$ is smaller than the LHS, since

$$c(j) - (1 + \epsilon(d - i^\star - 1))\bar{y}_{\sigma(j)} - \epsilon \sum_{i \geq i^\star} \bar{y}_{\sigma(i)} = \epsilon \sum_{j \leq i < i^\star} \bar{y}_{\sigma(j)} - \bar{y}_{\sigma(i)} \leq 0,$$

which concludes the proof. ∎

Theorem 2: Algorithm 4 solves the Bregman projection problem with exponential potential ϕ_ϵ in $\mathcal{O}(d \log d)$ iterations.

Proof: Correctness of the algorithm follows from the characterization of the support of $x^\star$ in Proposition 5 and the expression of $x^\star$ in Proposition 4. The complexity of the sort operation (step 4) is $\mathcal{O}(d \log d)$, and finding $j^\star$ (step 5) can be done in linear time since the criterion function $c(\cdot)$ is such that $c(j+1) - c(j) = (1+\epsilon(d-j))(\bar{y}_{\sigma(j+1)} - \bar{y}_{\sigma(j)})$, so each criterion evaluation costs $\mathcal{O}(1)$. Therefore, the overall complexity of Algorithm 4 is $\mathcal{O}(d \log d)$. ∎

Algorithm 5 QuickProjection Algorithm to compute the Bregman projection with D_{ψ_ϵ}

Input: $\bar{x}$, $\bar{g}$
Output: $x^\star$
Form the vector $\bar{y}_i = (\bar{x}_i + \epsilon)e^{-\bar{g}_i}$
Initialize $\mathcal{J} = \{1, \ldots, d\}$, $S = 0$, $C = 0$, $s^\star = d + 1$
while $\mathcal{J} \neq \emptyset$ **do**
  Select a random pivot index $j \in \mathcal{J}$
  Partition $\mathcal{J}$

$$\mathcal{J}^+ = \{i \in \mathcal{J} : \bar{y}_i \geq \bar{y}_j\} \quad \mathcal{J}^- = \{i \in \mathcal{J} : \bar{y}_i < \bar{y}_j\}$$

  and compute

$$S^+ = \sum_{i \in \mathcal{J}^+} \bar{y}_i \qquad C^+ = |\mathcal{J}^+|$$

  Let $\gamma = (1 + \epsilon(C + C^+))\bar{y}_j - \epsilon(S + S^+)$
  if $\gamma > 0$ **then**
    $\mathcal{J} \leftarrow \mathcal{J}^-$, $s^\star = j$
    $S \leftarrow S + S^+$, $C \leftarrow C + C^+$
  else
    $\mathcal{J} \leftarrow \mathcal{J}^+$
  end if
end while
Set $Z = \frac{S}{1+\epsilon C}$
Set

$$x_i^\star = \left(-\epsilon + \frac{\bar{y}_i}{Z} \right)_+$$

B. A randomized pivot algorithm to compute the exact solution

We now propose a randomized version of Algorithm 4, which selects a random pivot at each iteration, instead of sorting the full vector. The resulting algorithm, which we call QuickProject, is an extension of the QuickSelect algorithm due to Hoare [16]. A similar idea is used in the randomized version of the ℓ_2 projection on the simplex in [13].

Theorem 3: In expectation, the QuickProject Algorithm terminates after $\mathcal{O}(d)$ operations, and outputs the solution $x^\star$ of the Bregman projection problem 5 with the Bregman divergence $D_{KL,\epsilon}$.

Proof: First, we prove that the algorithm has expected linear complexity. Let $T(n)$ be the expected complexity of the while loop when $|\mathcal{J}| = n$.

The partition and compute step (7) takes $3n$ operations, then we recursively apply the loop to $\mathcal{J}^-$ or $\mathcal{J}^+$, which have sizes $(m, n-m)$ for any $m \in \{1, \ldots, n\}$, with uniform

probability. Thus we can bound $T(n)$ as follows

$$\begin{aligned} T(n) &\le 3n + \frac{1}{n}\sum_{m=1}^{n} T(\max(m, n-m)) \\ &\le 3n + \frac{2}{n}\sum_{m=\frac{n}{2}}^{n} T(m), \end{aligned}$$

and we can show by induction that $T(n) \le 12n$, since $T(0) = 0$ and

$$3n + \frac{2}{n}\sum_{m=\frac{n}{2}}^{n} 12m \le 3n + 12\frac{3n}{4} = 12n.$$

To prove the correctness of the algorithm, we will prove that once the while loop terminates, $s^\star = \sigma(j^\star)$, and S, C are respectively the sum and the cardinality of $\{\bar{y}_{\sigma(i)} : i \ge j^\star\}$, then by Proposition 4, we have the correct expression of $x^\star$. We start by showing the following invariants:

(i) If $\bar{y}_{\sigma(m_t)}$, is the largest element in $\mathcal{J}^{(t)}$, then $\sigma(m_t + 1) = (s^\star)^{(t)}$.
(ii) $\mathcal{J}^{(t)}$ contains $\sigma(j^\star)$ or $\sigma(j^\star - 1)$.
(iii) S and C are the sum and cardinality of $\{i : \sigma(i) \ge s^\star\}$.
(iv) $\gamma^{(t)} = c(j^{(t)})$, where c is the criterion function defined in Proposition 5.

The invariant holds for the first iteration since $\mathcal{J}^{(1)} = \{1, \dots, d\}$, $m_t = d$, and $S^{(1)} = C^{(1)} = 0$. Suppose the invariant is true at iteration t of the loop. Then two cases are possible:

1) If $\gamma^{(t)} \le 0$, then $\mathcal{J}^{(t+1)} = (\mathcal{J}^{(t)})^+$ and $m^{(t+1)} = m^{(t)}$, and the invariant still holds.
2) If $\gamma^{(t)} > 0$, then $\mathcal{J}^{(t+1)} = (\mathcal{J}^{(t)})^-$ and $(s^\star)^{(t+1)} = j^{(t)}$, thus

$$\begin{aligned} &\{i : \sigma(i) \ge (s^\star)^{t+1}\} \\ &= \{i : \sigma(i) \ge (s^\star)^{(t)}\} \cup \{i : (s^\star)^{t+1} \le \sigma(i) \le (s^\star)^{(t)} - 1\} \\ &= \{i : \sigma(i) \ge (s^\star)^{(t)}\} \cup (\mathcal{J}^{(t)})^+, \end{aligned}$$

and by the update step (lines 10–11), the invariant still holds.

To finish the proof, suppose the while loop terminates after T iterations, i.e. $\mathcal{J}^{(T+1)} = \emptyset$. We claim that $(s^\star)^{(T+1)} = \sigma(j^\star)$. During the last update, two cases are possible:

1) If $\gamma^{(T)} > 0$, then $\bar{y}_{j^{(T)}}$ is the smallest element of $\mathcal{J}^{(T)}$. In this case, since $c(i) \le 0$ for $i < j^\star$, and $\mathcal{J}^{(T)}$ contains $\sigma(j^\star)$ or $\sigma(j^\star - 1)$, it must be that $j^{(T)} = \sigma(j^\star)$, thus

$$(s^\star)^{T+1} = j^{(T)} = \sigma(j^\star).$$

2) If $\gamma^{(T)} \le 0$, then $\bar{y}_{j^{(T)}}$ is the largest element of $\mathcal{J}^{(T)}$, in this case, since $c(j^\star) > 0$, it must be that $j^{(T)} = \sigma(j^\star - 1)$, so $m^{(t)} = j^\star - 1$ and

$$(s^\star)^{(T+1)} = (s^\star)^{(T)} = \sigma(m^{(t)} + 1) = \sigma(j^\star).$$

This concludes the proof. ∎

C. Properties of the generalized KL divergence

Algorithms 4 and 5 give efficient methods for computing the projection with generalized KL divergence $D_{KL,\epsilon}$. In this section, we show that this family of Bregman divergences enjoys additional properties, given below.

Proposition 6: For all $\epsilon > 0$, $D_{KL,\epsilon}$ is ℓ_ϵ-strongly convex and L_ϵ-smooth w.r.t. $\|\cdot\|_1$, and bounded by D_ϵ on Δ, with

$$\ell_\epsilon \ge \frac{1}{1+d\epsilon}, \qquad L_\epsilon \le \frac{1}{\epsilon}, \qquad D_\epsilon \le \ln\frac{1+\epsilon}{\epsilon}.$$

Proof: First, we show strong convexity. Let $x, y \in \Delta$. By Taylor's theorem, $\exists z \in (x + \boldsymbol{\epsilon}, y + \boldsymbol{\epsilon})$ such that

$$\begin{aligned} D_{KL,\epsilon}(x, y) &= H(x + \boldsymbol{\epsilon}) - H(y + \boldsymbol{\epsilon}) - \langle \nabla H(y + \boldsymbol{\epsilon}), x - y\rangle \\ &= \frac{1}{2}\left\langle x - y, \nabla^2 H(z)(x - y)\right\rangle \\ &= \frac{1}{2}\sum_i \frac{(x_i - y_i)^2}{z_i}, \end{aligned}$$

where we used the fact that the Hessian of the negative entropy function is $\nabla^2 H(z) = \operatorname{diag}(\frac{1}{z_i})$. And since $\forall i$, $z_i \ge \epsilon$ (z belongs to the segment $(x + \boldsymbol{\epsilon}, y + \boldsymbol{\epsilon})$), it follows that

$$D_{KL,\epsilon}(x, y) \le \frac{1}{2\epsilon}\sum_i (x_i - y_i)^2 \le \frac{1}{2\epsilon}\|x - y\|_1^2.$$

Furthermore, by the Cauchy-Schwartz inequality, $(\sum_i |x_i - y_i|)^2 \le \sum_i \frac{(x_i - y_i)^2}{z_i} \sum_i z_i$, thus

$$D_{KL,\epsilon}(x, y) \ge \frac{1}{2}\frac{\|x - y\|_1^2}{\|z\|_1} = \frac{1}{2}\frac{1}{1 + d\epsilon}\|x - y\|_1^2.$$

To compute the upper bound on $D_{KL,\epsilon}$, we observe that $D_{KL,\epsilon}(x, y)$ is jointly-convex in (x, y) (by joint-convexity of the KL divergence), therefore, its maximum on $\Delta^d \times \Delta^d$ is attained on a vertex of the feasible set, that is, for $(x, y) = (\delta^{i_0}, \delta^{j_0})$, for some (i_0, j_0), where δ^{i_0} is the Dirac distribution on i_0. Finally, simple calculation shows that

$$D_{KL,\epsilon}(\delta^{i_0}, \delta^{j_0}) = \begin{cases} 0 & \text{if } i_0 = j_0, \\ \ln\frac{1+\epsilon}{\epsilon} & \text{otherwise.} \end{cases}$$

∎

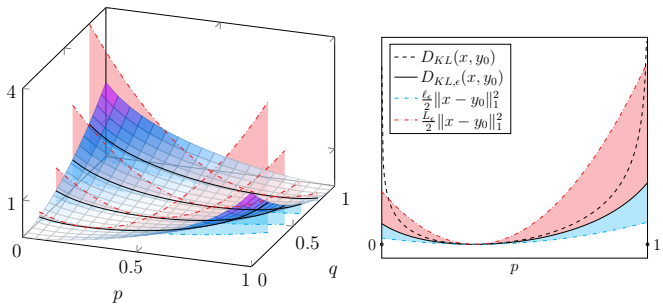

Fig. 3. Illustration of Proposition 6, when $d = 2$. The distributions x and y are parameterized as follows: $x = (p, 1-p)$ and $y = (q, 1-q)$. The surface plot (left) shows the generalized KL divergence for $\epsilon = .1$, with, in dashed lines, the quadratic upper and lower bounds, $\frac{\ell_\epsilon}{2}\|y - x\|_1^2$ and $\frac{L_\epsilon}{2}\|x - y\|_1^2$. The second plot (right) compares $D_{KL,.1}(x, y_0)$ and $D_{KL}(x, y_0)$ for a fixed $y_0 = (.35, .65)$.

D. Numerical experiments

We provide a simple python implementation of the projection algorithms at `github.com/walidk/BregmanProjection`. The implementation of Algorithm 3 is generic and can be instantiated for any ω-potential by providing the function ϕ and its inverse. The implementation of Algorithm 4 and QuickProject are specific to the generalized exponential potential. Finally, we report in Figure 4 the run times of both algorithms as the dimension d grows, averaged over 50 runs, for randomly generated, normally distributed vectors $\bar{x}$ and $\bar{g}$. The numerical simulations are also available on the same repository.

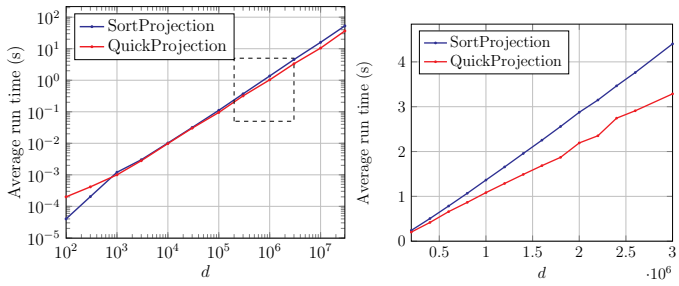

Fig. 4. Execution time as a function of the dimension d, with $\epsilon = .1$, in log-log scale (left). The highlighted region is zoomed-in in linear scale on the right. The simulation confirms that the QuickProject algorithm is, on average, faster than the sorting algorithm, especially for large d.

V. Conclusion

We studied the Bregman projection problem on the simplex with ω-potentials, and derived optimality conditions for the solution, which motivated a simple bisection algorithm to compute ϵ approximate solutions in $\mathcal{O}(d \log(1/\epsilon))$ time. Then we focused on the projection problem with exponential potentials, resulting in a Bregman divergence which generalizes the KL divergence. We showed that in this case, the solution can be computed exactly in $\mathcal{O}(d \log d)$ time using a sorting algorithm, or in expected $\mathcal{O}(d)$ time using a randomized pivot algorithm. This class of divergences is of particular interest because it has a quadratic upper and lower bound (i.e. its distance generating function is both strongly convex and smooth), a property which is essential to obtain convergence guarantees in some settings, such as stochastic mirror descent. A question which remains open is whether one can project in $\mathcal{O}(d)$ time using a deterministic algorithm akin to the "median of medians" algorithm due to Blum et al. [7] which solves the selection problem in deterministic linear time.

The fact that one can efficiently compute the exact solution hinges on the existence of a closed-form solution of the dual variable $\nu^\star$ given the support of the solution (Proposition 4). This is also the case for the Euclidean projection, i.e. when D_ψ is the squared Euclidean norm, see [13]. This suggests that one may derive efficient projection algorithms for other classes of Bregman divergences, which would, in turn, lead to new efficient instances of the mirror descent method.

References

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

[2] Jean-Yves Audibert, Sébastien Bubeck, and Gàbor Lugosi. Regret in online combinatorial optimization. *Mathematics of Operations Research*, 39(1):31–45, 2014.

[3] Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Oper. Res. Lett.*, 31(3):167–175, May 2003.

[4] A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization*. Society for Industrial and Applied Mathematics, 2001.

[5] Aharon Ben-Tal, Tamar Margalit, and Arkadi Nemirovski. The ordered subsets mirror descent optimization method with applications to tomography. *SIAM J. on Optimization*, 12(1):79–108, January 2001.

[6] David Blackwell. An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6(1):1–8, 1956.

[7] Manuel Blum, Robert W. Floyd, Vaughan Pratt, Ronald L. Rivest, and Robert E. Tarjan. Time bounds for selection. *J. Comput. Syst. Sci.*, 7(4):448–461, August 1973.

[8] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*, volume 25. Cambridge University Press, 2010.

[9] Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.

[10] Yair Censor and Stavros Zenios. *Parallel Optimization: Theory, Algorithms and Applications*. Oxford University Press, 1997.

[11] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.

[12] Ofer Dekel, Ran Gilad-Bachrach, Ohad Shamir, and Lin Xiao. Optimal distributed online prediction. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, June 2011.

[13] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l1-ball for learning in high dimensions. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 272–279, New York, NY, USA, 2008. ACM.

[14] John C. Duchi, Alekh Agarwal, Mikael Johansson, and Michael Jordan. Ergodic mirror descent. *SIAM Journal on Optimization (SIOPT)*, 22(4):1549–1578, 2010.

[15] James Hannan. Approximation to Bayes risk in repeated plays. *Contributions to the Theory of Games*, 3:97–139, 1957.

[16] C. A. R. Hoare. Algorithm 65: Find. *Commun. ACM*, 4(7):321–322, July 1961.

[17] Anatoli Juditsky, Arkadi Nemirovski, and Claire Tauvel. Solving variational inequalities with stochastic mirror-prox algorithm. *Stoch. Syst.*, 1(1):17–58, 2011.

[18] Jyrki Kivinen and Manfred K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1 – 63, 1997.

[19] Syrine Krichene, Walid Krichene, Roy Dong, and Alexandre Bayen. Convergence of heterogeneous distributed learning in the stochastic routing game. In *Proceedings of the 53rd Annual Allerton Conference on Communication, Control, and Computing*, 2015.

[20] Walid Krichene, Syrine Krichene, and Alexandre Bayen. Convergence of mirror descent dynamics in the routing game. In *European Control Conference (ECC), accepted*, 2015.

[21] A. S. Nemirovsky and D. B. Yudin. *Problem complexity and method efficiency in optimization*. Wiley-Interscience series in discrete mathematics. Wiley, 1983.