THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par l'Université Toulouse 1 Capitole

Présentée et soutenue par

Radu-Alexandru DRAGOMIR

Le 14 septembre 2021

Méthodes de gradient de Bregman pour problèmes à régularité relative

Ecole doctorale : **EDMITT - Ecole Doctorale Mathématiques, Informatique et Télécommunications de Toulouse**

Spécialité : **Mathématiques et Applications**

Unité de recherche :
UMR 5314 -TSE-R

Thèse dirigée par
Jérome BOLTE et Alexandre D'ASPREMONT

Jury
M. George LAN, Rapporteur
M. Yoel DRORI, Rapporteur
M. Edouard PAUWELS, Examinateur
M. Cristobal GUZMAN, Examinateur
Mme Jelena DIAKONIKOLAS, Examinatrice
M. Yurii NESTEROV, Examinateur
M. Jérome BOLTE, Directeur de thèse
M. Alexandre D'ASPREMONT, Co-directeur de thèse

Résumé

En apprentissage statistique et traitement du signal, de nombreuses tâches se formulent sous la forme d'un problème d'optimisation de grande taille. Dans ce contexte, les méthodes du premier ordre, qui utilisent uniquement l'information apportée par le gradient de la fonction objectif, sont privilégiées en raison de leur faible coup par itération et de leur simplicité. Nous étudions dans cette thèse les méthodes du premier ordre à distance de Bregman, qui constituent une généralisation de la célèbre méthode de descente de gradient. Cette généralisation consiste à remplacer la distance euclidienne par une distance plus générale, dite de Bregman, générée par une fonction convexe noyau suffisamment simple. La fonction noyau est choisie de manière à être adaptée à la géométrie de la fonction objectif au travers de la condition de *régularité relative*, introduite en 2017 par Bauschke, Bolte et Teboulle. Depuis son apparition, cette condition a fait naître de nouvelles perspectives en optimisation du premier ordre.

Tout d'abord, nous appliquons les méthodes de Bregman aux problèmes d'optimisation sur des espaces de matrices de rang faible. En exploitant la structure matricielle et en utilisant la propriété de régularité relative, nous proposons des noyaux de Bregman qui permettent d'améliorer la performance numérique par rapport aux méthodes euclidiennes.

Ensuite, nous nous penchons sur la complexité théorique de ces algorithmes. Un des problèmes les plus importants est de déterminer s'il existe une version *accélérée* de l'algorithme de gradient de Bregman qui possède un meilleur taux de convergence. Dans le cas général, nous démontrons que la réponse est négative : la complexité de la descente de gradient de Bregman standard ne peux pas être améliorée pour des noyaux génériques. La preuve repose sur un contre-exemple pathologique qui a été découvert au travers de méthodes d'analyses de pire cas par ordinateur. Nous évoquons aussi une tentative pour obtenir des résultats positifs d'accélération en spécialisant cette analyse dans le contexte plus restreint de la géométrie entropique.

Enfin, nous étudions la version stochastique de l'algorithme de Bregman pour minimiser des fonctions sous la forme d'espérance, ainsi que des méthodes de réduction de variance lorsque la fonction objectif est une somme finie.

Abstract

We study large-scale optimization problems with applications to signal processing and machine learning. Such problems are typically solved with first-order methods that perform iterative updates using the gradient of the objective function. We focus on the class of Bregman first-order methods, for which the direction of the gradient step is determined by the Bregman divergence induced by a convex kernel function. The choice of the kernel is guided by the *relative smoothness* condition, which requires the kernel to be compatible with the objective through a descent inequality. This condition was introduced recently by Bauschke, Bolte and Teboulle in 2017 and has opened new perspectives in first-order optimization.

In the first part, we apply Bregman methods to minimization problems on the space of low-rank semidefinite matrices. By leveraging the matrix structure and using the relative smoothness property, we show that well-chosen Bregman kernels allow to improve performance over standard Euclidean methods.

Then, we study the theoretical complexity of these algorithms. An important question is to determine whether there exists an accelerated version of Bregman gradient descent which achieves a better convergence rate in the same setting. In the general case, we show that the answer is negative as the complexity of the standard Bregman gradient method cannot be improved for generic kernels. The proof relies on a pathological example which was discovered by analyzing the worst-case behavior of Bregman methods with a computer-aided technique called performance estimation. We also detail an attempt towards improving the convergence speed in a more restricted setting, by specializing the performance estimation framework to the entropic geometry.

Finally, we study a stochastic variant of Bregman gradient descent for expectation minimization problems, which are pervasive in machine learning, along with variance reduction methods for finite-sum objectives.

Remerciements

Je tiens en premier lieu à remercier chaleureusement mes deux directeurs de thèse pour leur encadrement bienveillant et amical. J'ai été marqué, Jérôme, par ton dévouement, ta sincérité, ta philosophie, ta passion pour les belles mathématiques et ton exigence d'une recherche de qualité. Alexandre, ta culture scientifique, ton sens de l'initiative et du relationnel, ton énergie et ton enthousiasme à toute épreuve ont été un moteur essentiel de cette thèse.

Je tiens à exprimer mes sincères remerciements aux membres du jury : Yoel Drori, George Lan, Yurii Nesterov, Edouard Pauwels, Cristobal Guzman et Jelena Diakonikolas, pour avoir assisté a ma soutenance et pour la discussion enrichissante qui s'en est suivie. Un grand merci en particulier à Yoel Drori pour un rapport extrêmement détaillé et constructif.

Si l'activité de chercheur est parfois un travail solitaire, les collaborations en constituent la partie la plus stimulante. Merci à Adrien Taylor, qui, au-delà d'avoir été un collaborateur essentiel sur une des principales parties de cette thèse, s'est aussi révélé être un formidable mentor et ami. Merci à Dmitrii Ostrovskii, avec qui nos discussions sur la géométrie entropique ont rythmé nos semaines de confinement, et à Hadrien Hendrikx et Mathieu Even, pour ce beau projet où l'on a fusionné nos compétences en fin de thèse.

Durant ces trois années, j'ai pu alterner entre deux environnements recherche de deux côtés différents de la France : merci à mes collègues de Toulouse, Camille, Edouard, Rodolfo, Lilian et tous les autres, ainsi qu'à toute l'équipe Sierra à Paris, sans oublier les équipes Willow et Dyogène, pour tous ces échanges passionnants qui m'ont permis de découvrir de près ou de loin une grande variété de sujets.

J'ai eu la chance d'effectuer une majeure partie de ma thèse dans ce lieu à l'ambiance mondialement renommée que l'on appelle le "quatrième étage de l'Inria" (j'en place une pour le troisième aussi, on ne vous oublie pas). C'est lorsque le bureau a fermé, en mars 2020, que l'on s'est subitement rendu compte de tout ce qui allait nous manquer : les verres du vendredi, les pauses café, l'indispensable pause baby-foot, les soirées sur les Champs-Elysées ou bien sur le Vieux-Port, le difficile dilemme "Sav'heure/Bonne Tradition", les débats intenses autour des concours culinaires télévisés ou bien le débrief des dernières sorties de musique urbaine de la semaine. Un grand merci à Mathieu, Grégoire, Thomas K, Manon, Thomas E, Hadrien, Rémi, Yann, Adrien, Alex N, Robin, Ulysse, Vivien, Francis, Yana, Loucas, Raphaël, Oumayma, Justin, Bruno, Ale, Céline, Hans, Eloise, Loïc, Gautier, Armand, Alberto, Pierre, Pierre-Louis, Wilson, Hélène, Emma et tous les autres: je vous souhaite bonne chance pour la suite et j'espère que nos chemins se recroiseront à l'avenir.

Je n'aurais pas pu faire cette thèse sans le soutien inconditionnel de ma famille, qui a toujours su être là pour me guider dans les moments faciles comme les plus difficiles : merci infiniment.

Enfin, ces remerciements seraient incomplets si je n'évoquais pas mon troisième lieu de travail, celui que j'ai occupé bon gré mal gré durant une grande partie de ces deux dernières années : merci à vous, Artiom et Marius, pour tous ces moments inoubliables passés en collocation, et aux nombreux amis que l'on a reçu rue du Temple au fil des années.

Contents

Contributions and Thesis Outline

Throughout this thesis, we consider the minimization problem

$$\min_{x \in \mathcal{C}} f(x)$$

where $\mathcal{C}$ is a convex set and f a differentiable function. We focus on the setting where f satisfies a relative smoothness condition with respect to some kernel function h:

$$f(x) \leq f(y) + \langle \nabla f(y), x - y \rangle + L D_h(x, y), \quad \forall x, y \in \operatorname{int} \mathcal{C},$$

where D_h is the Bregman divergence induced by h. The basic algorithm for solving this problem is Bregman gradient descent (also known as *NoLips* or *mirror descent*).

Chapter 1: we introduce the framework of Bregman gradient methods for relatively-smooth problems, and review related work on theoretical aspects, algorithmic extensions and applications. We also provide a more general historical overview on the use of Bregman divergences in optimization.

Chapter 2: in this chapter, we apply the Bregman gradient method to nonconvex low-rank problems. We wish to minimize a L-smooth function F on the space of semidefinite matrices of size $n \times n$ and rank at most r with the nonconvex *Burer-Monteiro* formulation, which writes

$$f(X) = F(XX^T) + g(X)$$

where $X \in \mathbb{R}^{n \times r}$, and g is a (possibly nonsmooth) regularization or constraint function. We show that the factorized function f is not globally smooth with respect to the squared Euclidean norm, but it can be proven to be relatively-smooth with respect to well-chosen quartic kernels, allowing to apply Bregman proximal gradient methods. We study two types of quartic kernels. The *norm kernel* is defined as

$$h_n(X) = \frac{\alpha}{4} \|X\|^4 + \frac{\sigma}{2} \|X\|^2$$

and can be used for various functions g, such as the sparsity-inducing ℓ_1 norm or the indicator set of the nonnegative orthant. We also introduce the *Gram kernel*,

$$h_G(X) = \frac{\alpha}{4} \|X\|^4 + \frac{\beta}{4} \|X^T X\|^2 + \frac{\sigma}{2} \|X\|^2,$$

which is more complex to use as the Bregman gradient iterates can only be computed easily for problems without penalization, i.e., with $g \equiv 0$. We show that, by leveraging the low-rank matrix structure, the Gram kernel allows to improve numerical performance for well-conditionned functions F. We provide numerical experiments on two applications: symmetric nonnegative matrix factorization (SymNMF) and Euclidean distance matrix completion.

Chapter 3: we study the theoretical complexity of relatively-smooth convex minimization. We prove a lower bound stating that Bregman gradient descent is an *optimal* method in this setting, among the general class of methods that use Bregman divergences and basic linear operations. This result demonstrates that the $\mathcal{O}(1/k)$ convergence rate of Bregman gradient descent cannot be improved for generic kernel h, and thus that additional regularity assumptions are needed in order to devise a faster algorithm.

The worst-case instances involved in the lower bound were inferred from the numerical solution to a *performance estimation problem*, which we describe in the next chapter.

Chapter 4: in this chapter, we use computer-aided tools for analyzing the worst-case complexity of Bregman methods. This technique, called *performance estimation*, was initially designed for the study of Euclidean gradient methods. We extend it to the setting of relatively-smooth problems, for computing theoretical guarantees of Bregman methods with *generic kernel* h. That is, we search for the worst-case behavior among the class of relatively-smooth problem instances of the form

$$\mathcal{B}_L(\mathcal{C}) = \{(f,h)\,:\, h \text{ is a kernel function on } \mathcal{C} \text{ and } f \text{ is convex and } L\text{-smooth relative to } h\}\,.$$

By establishing *interpolation conditions*, we reduce this task to a finite-dimensional semidefinite program defined on an open set. Then, we use topological arguments to show that this problem is equivalent to a simpler problem on a larger class $\overline{\mathcal{B}_L}(\mathcal{C})$, which can be seen as the closure of $\mathcal{B}_L(\mathcal{C})$ and involves possibly nonsmooth convex functions.

We showcase this approach on several examples. First, solving the dual performance estimation problem allows to infer analytical proofs for two different convergence results of BGD. Then, we are also able to discover the corresponding worst-case functions $\overline{f}$ and $\overline{h}$. Because of the topological fact mentioned above, these worst-case elements are actually nonsmooth functions from $\overline{\mathcal{B}_L}(\mathcal{C})$, and can be approached by a sequence of elements of $\mathcal{B}_L(\mathcal{C})$. This *limiting nonsmooth behavior* is at the core of the proof of the lower bound in Chapter 3, which was inferred from these numerically generated worst-case examples.

Chapter 5: in this chapter, we describe an attempt to perform computer-aided analyses of Bregman methods in the more restricted setting of a *fixed* kernel h. Indeed, results from previous chapters suggest that the class $\mathcal{B}_L(\mathcal{C})$ of general relatively-smooth problems is too large, as functions can approach pathological nonsmooth instances from $\overline{\mathcal{B}_L}(\mathcal{C})$. To consider a more realistic setting, we focus on the particular case of the *entropic kernel*

$$h_e(x) = \sum_{i=1}^{d} x^{(i)} \log x^{(i)} - x^{(i)},$$

which is one of the most common kernels used in Bregman methods and presents favorable regularity properties. Therefore, we now search for the worst-case of Bregman methods on the class

$$\mathcal{F}_L^{h_e} = \{f\,:\, f \text{ is convex and } L\text{-smooth relative to } h_e\}\,.$$

We show that the corresponding performance estimation problem (PEP) can be reduced to a finite-dimensional problem through interpolation conditions, which we establish using the smoothing properties of Bregman-Moreau envelopes. We then formulate this problem as a convex program on the set of matrices of pairwise Kullback-Leibler divergences, the *Kullback-Leibler cone.*

Since no solvers are available for this type of problem, we use simple heuristics for solving small instances of the PEP and report preliminary numerical results.

Chapter 6: we study stochastic Bregman gradient methods for problems where the objective is an expectation of relatively-smooth convex functions. We first prove the convergence of Bregman stochastic gradient descent under a condition on the variance of the gradients at the optimal point.

We then focus on variance reduction techniques for finite-sum problems, which are widely used for accelerating the convergence of stochastic methods in the Euclidean setting. We propose the Bregman counterpart of the SAGA method [Defazio et al., 2014]. The analysis is more tedious and requires additional regularity conditions of the kernel h. We show that, with the right choice of step sizes, the desired fast convergence rate is reached asymptotically.

Finally, we demonstrate the effectiveness of these stochastic variants by reporting numerical experiments on Poisson inverse problems.

References: publications related to this thesis are listed below.

- **Chapter** 2 is based on the article [Dragomir et al., 2021a]: Radu-Alexandru Dragomir, Alexandre d'Aspremont, and Jérôme Bolte. Quartic First-Order Methods for Low-Rank Minimization. *Journal of Optimization Theory and Applications*, 189(2), 2021.

- **Chapters** 3 and 4 are based on the article [Dragomir et al., 2021c]: Radu-Alexandru Dragomir, Adrien Taylor, Alexandre d'Aspremont, and Jérôme Bolte. Optimal Complexity and Certification of Bregman First-Order Methods. *Mathematical Programming*, 2021.

- **Chapter** 6 is based on the article [Dragomir et al., 2021b]: Radu-Alexandru Dragomir, Mathieu Even, Hadrien Hendrikx (2021). Fast Stochastic Bregman Gradient Methods: Sharp Analysis and Variance Reduction. *To appear in International Conference on Machine Learning*, 2021.

Chapter 1

Relative Smoothness and Bregman Optimization Methods

Many tasks in computational science, engineering and operations research can be formulated as constrained optimization problems of the form

$$\min_{x \in \mathcal{C}} f(x) \tag{P}$$

where $\mathcal{C}$ is a convex subset of $\mathbb{R}^d$ and f is a differentiable function. We are particularly interested in large-scale applications from signal processing, data analysis and machine learning, where the dimension d typically ranges between 10^3 and 10^9. Problem (P) is generally solved with iterative algorithms, which successively minimize local approximations of the function f based on its derivatives. As higher-order derivatives are too costly to compute for high dimension d, we consider methods that only rely on the information provided by the gradient ∇f: they constitute the class of *first-order methods*.

The simplest and most known first-order method is projected gradient descent (GD), which amounts to iterate

$$x_{k+1} = \Pi_{\mathcal{C}}\left(x_k - \lambda_k \nabla f(x_k)\right), \tag{1.1}$$

where $k \geq 0$ is the iteration counter, $\Pi_{\mathcal{C}}$ is the Euclidean projection on $\mathcal{C}$ and $\lambda_k > 0$ is the step size. Equivalently, this update can be written as

$$x_{k+1} = \operatorname*{argmin}_{u \in \mathcal{C}} \; f(x_k) + \langle \nabla f(x_k), u - x_k \rangle + \frac{1}{2\lambda_k}\|u - x_k\|^2, \tag{GD}$$

where $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$ denote the Euclidean inner product and norm on $\mathbb{R}^d$. Thus, gradient descent successively minimizes a surrogate of f consisting of the first-order approximation penalized with a quadratic term which quantifies the inexactness of the linear model far from x_k. The efficiency of the method is connected to how well this surrogate approximates f. The classical condition for analyzing this is to assume that the gradient ∇f is Lipschitz continuous. This ensures that, for λ_k small enough, the inner objective in (GD) is an upper bound on f, and therefore that every iteration decreases the value of the objective.

However, in some problems, the quadratic model is a poor approximation of f and a different geometry might be more adapted. This is the purpose of the *Bregman gradient descent* method, which performs the update

$$x_{k+1} = \operatorname*{argmin}_{u \in \mathcal{C}} \; f(x_k) + \langle \nabla f(x_k), u - x_k \rangle + \frac{1}{\lambda_k} D_h(u, x_k), \tag{BGD}$$

where the quadratic term has been replaced by the more general *Bregman divergence*

$$D_h(x,y) = h(x) - h(y) - \langle \nabla h(y), x - y \rangle$$

induced by some convex *kernel* function h. This divergence is nonnegative by convexity of h. Note that taking $h(x) = \frac{1}{2}\|x\|^2$ yields the Euclidean distance $D_h(x,y) = \frac{1}{2}\|x-y\|^2$. However, choosing a different kernel fuction h can result in a more accurate local model than the standard Euclidean one. This choice is guided by the so-called *relative smoothness condition*, introduced in Bauschke et al. [2017], which amounts to assume that the inner objective in (BGD) is an upper bound on f. Naturally, the kernel h also needs to be simple enough so that the subproblem defining x_{k+1} can be solved easily, that is, in closed form or with efficient numerical schemes.

The Bregman gradient method is also known as *Mirror Descent*, and has been originally proposed by Nemirovski and Yudin [1983] for minimizing convex, possibly nonsmooth, functions. In their context, the kernel h is chosen according to the set $\mathcal{C}$, in order to minimize its diameter with respect to the corresponding Bregman divergence (see the discussion in Section 1.5). In this thesis, we study the more recent relatively-smooth setting, where h is chosen to fit the curvature of the differentiable function f, and use the *Bregman gradient descent* terminology, although there is no consensus in the litterature.

1.1 Relative smoothness and the Bregman gradient method

In this section, we review the general setting for relatively-smooth optimization as well as the main theoretical results. We use standard notation and definitions from convex analysis; see e.g., Rockafellar [1970], Bauschke and Combettes [2011].

The first step is the choice of a kernel function h adapted to the optimization set $\mathcal{C}$. While the blanket assumptions on h vary between papers, there is usually a set of minimal requirements.

Definition 1.1 (Kernel function)**.** *A function $h : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is called a kernel function on $\mathcal{C}$ if*

(i) h is closed convex proper,

(ii) h is continuously differentiable and strictly convex on $\operatorname{int}\mathcal{C}$*,*

(iii) the Bregman gradient iterates are well-posed, i.e., for every $p \in \mathbb{R}^d$, the problem

$$\min_{u \in C} \langle p, u \rangle + h(u)$$

has a unique minimizer, which belongs to $\operatorname{int}\mathcal{C}$*.*

The set $\mathcal{C}$ is sometimes called the *zone* of h. The kernel h induces the Bregman divergence

$$D_h(x,y) = h(x) - h(y) - \langle \nabla h(y), x - y \rangle$$

defined for $x \in \operatorname{dom} h, y \in \operatorname{dom} \nabla h$. Because of strict convexity of h, the Bregman divergence $D_h(x,y)$ is nonnegative and equal to zero if and only if $x = y$. However, it is not a proper distance, as it is not symmetric in general. If h is sufficiently regular, then D_h is locally approximated by a quadratic function when x is close to y:

$$D_h(x,y) \approx \frac{1}{2}\langle \nabla^2 h(y)(x-y), x-y \rangle \quad \text{when } x \to y,$$

where $\nabla^2 h$ denotes the Hessian matrix of h. Equiped with the geometry induced by h, we can now define for $x \in \operatorname{int} \mathcal{C}$ the Bregman gradient map with step size $\lambda > 0$:

$$T_\lambda(x) = \operatorname*{argmin}_{u \in \mathcal{C}} \langle \nabla f(x), u - x \rangle + \frac{1}{\lambda} D_h(u, x),$$

which we assume to be easily computable. Due to Definition 1.1(iii), T_λ is well defined on $\operatorname{int} \mathcal{C}$. We now list the most standard examples of kernels.

- The **Euclidean kernel** $h(x) = \frac{1}{2}\|x\|^2$ induces the Euclidean distance $D_h(x, y) = \frac{1}{2}\|x - y\|^2$, and T_λ is the projected gradient step (1.1).

- The **quadratic kernel** is defined as $h(x) = \frac{1}{2}\langle Ax, x \rangle$, where $A \in \mathbb{R}^{d \times d}$ is a positive definite matrix. Using this kernel corresponds to performing linear preconditioning.

- The **entropy kernel** $h(x) = \sum_{i=1}^d x^{(i)} \log x^{(i)}$ defined for $x \in \mathbb{R}^d_+$ with the convention $0 \log 0 = 0$, induces the *Kullback-Leibler* divergence $D_h = D_{\mathrm{KL}}$ with

  $$D_{\mathrm{KL}}(x, y) = \sum_{i=1}^d x^{(i)} \log\left(\frac{x^{(i)}}{y^{(i)}}\right) - x^{(i)} + y^{(i)}.$$

  In the case when $\mathcal{C}$ is the unit simplex $\{x \in \mathbb{R}^d_+ : \sum_{i=1}^d x^{(i)} = 1\}$, the gradient map is

  $$T_\lambda(x)^{(i)} = \frac{x^{(i)} \exp(-\lambda \nabla f(x)^{(i)})}{\sum_{j=1}^d x^{(j)} \exp(-\lambda \nabla f(x)^{(j)})}$$

  for $i = 1 \dots d$, see, e.g., Beck and Teboulle [2003]. This formula is also known as *exponential weight update*, or *exponentiated gradient* [Kivinen and Warmuth, 1997].

- The **log kernel** $h(x) = \sum_{i=1}^d - \log x^{(i)}$ for $x \in \mathbb{R}^d_{++}$ induces the *Itakura-Saito* divergence $D_h = D_{\mathrm{IS}}$ with

  $$D_{\mathrm{IS}}(x, y) = \sum_{i=1}^d \left(- \log\left(\frac{x^{(i)}}{y^{(i)}}\right) + \frac{x^{(i)}}{y^{(i)}} - 1 \right).$$

  defined for $x, y \in \mathbb{R}^d_{++}$. The corresponding Bregman gradient map for $\mathcal{C} = \mathbb{R}^d_{++}$ is

  $$T_\lambda(x)^{(i)} = \frac{x^{(i)}}{1 + \lambda x^{(i)} \nabla f(x^{(i)})}, \quad i = 1 \dots d.$$

See Section 1.2 for more examples. It should be emphasized that, while a kernel function is differentiable on the interior of its domain, it is not required to be differentiable on the boundary. For instance, the entropy is continuous but not differentiable on the boundary of $\mathbb{R}^d_+$. Moreover, the domain of h can be closed, such as for the entropy which is finite on $\mathbb{R}^d_+$, or open, as for the log kernel.

Convex conjugate and mirror formulation. If h is a kernel function, we define its convex conjugate h^* as

$$h^*(y) = \sup_{u\in\mathbb{R}^d} \langle u, y\rangle - h(u).$$

By Assumption (iii), h^* is finite-valued on $\mathbb{R}^d$. Moreover, its gradient satisfies for every $u \in \mathbb{R}^d$ [Rockafellar, 1970, Sect. 26]

$$\nabla h^*(y) = \operatorname*{argmax}_{u\in\mathbb{R}^d} \langle u, y\rangle - h(u).$$

Hence the Bregman gradient map T_λ can be alternatively written as

$$T_\lambda(x) = \nabla h^* \left[\nabla h(x) - \lambda \nabla f(x)\right],$$

which is sometimes called the *mirror descent* formulation (see Section 1.5 for historical comments on this expression).

We now introduce the fundamental notion of relative smoothness, also known as *smooth adaptable property*, or *Lipschitz-like* condition [Bauschke et al., 2017].

Definition 1.2 (Relative smoothness)**.** *We say that the function f is smooth relative to h if it is differentiable on* $\operatorname{int}\mathcal{C}$ *and if there exists a constant* $L > 0$ *such that*

$$f(x) \leq f(y) + \langle \nabla f(y), x - y\rangle + LD_h(x, y) \tag{RelSmooth}$$

for every $x, y \in \operatorname{int}\mathcal{C}$.

An equivalent characterization of relative smoothness is to assume that $Lh - f$ is convex on $\operatorname{int}\mathcal{C}$, or, if f and h are twice differentiable, to the condition

$$\nabla^2 f(x) \preceq L\nabla^2 h(x) \quad \forall x \in \operatorname{int}\mathcal{C},$$

where $\preceq$ denotes the semidefinite order. If h is the Euclidean kernel, then (RelSmooth) corresponds to the standard notion of smoothness in optimization, implied by Lipschitz continuity of ∇f (which we will refer to as L-smoothness to avoid ambiguity).

Similarly, Lu et al. [2018] also defined the notion of relative strong convexity.

Definition 1.3 (Relative strong convexity)**.** *We say that the function f is strongly convex relative to h if it is differentiable on* $\operatorname{int}\mathcal{C}$ *and if there exists a constant* $\mu > 0$ *such that*

$$f(y) + \langle \nabla f(y), x - y\rangle + \mu D_h(x, y) \leq f(x)$$

for every $x, y \in \operatorname{int}\mathcal{C}$.

In the same, way, relative strong convexity is equivalent to the function $f - \mu h$ being convex on $\operatorname{int}\mathcal{C}$, or to the second-order characterization

$$\mu\nabla^2 h(x) \preceq \nabla^2 f(x) \quad \forall x \in \operatorname{int}\mathcal{C}.$$

The ratio $\kappa = L/\mu$ is called the *relative condition number*: a value close to 1 indicates that f is well approximated by the right-hand side of (RelSmooth). Now, note that this right-hand side provides a simple global majorant of f. A natural approach for solving (P) is then to successively minimize this majorant, and this leads to the Bregman Gradient method (Algorithm 1), also known as NoLips [Bauschke et al., 2017].

Algorithm 1 Bregman Gradient Descent (BGD) / NoLips

```
**Input:** Initial point $x_0 \in \operatorname{int} \mathcal{C}$, step size $\lambda$.
**for** k = 0,1,... **do**
    $x_{k+1} = \operatorname*{argmin}_{u \in \mathcal{C}} f(x_k) + \langle \nabla f(x_k), u - x_k \rangle + \frac{1}{\lambda} D_h(u, x_k)$
**end for**
```

We now state the theoretical convergence rate of Bregman gradient: as in the Euclidean case, the convergence is *sublinear* for convex functions and *linear* for relatively-strongly convex functions.

Proposition 1.4 (BGD convergence rate for convex functions)**.** *Assume that f is convex on $\mathcal{C}$ and L-smooth relative to a kernel function h. Let $\{x_k\}_{k\geq 0}$ be the sequence of iterates produced by Algorithm 1 with step size $\lambda \in (0, 1/L]$. Then, for every $u \in \mathcal{C}$, we have*

$$f(x_k) - f(u) \leq \frac{L D_h(u, x_0)}{k}, \tag{1.2}$$

and if f is additionally μ-strongly convex relative to h,

$$f(x_k) - f(u) \leq L \left(1 - \frac{\mu}{L}\right)^k D_h(u, x_0). \tag{1.3}$$

See [Bauschke et al., 2017, Thm. 1] for the proof of the sublinear rate and [Lu et al., 2018, Thm. 3.1] for the linear convergence result.

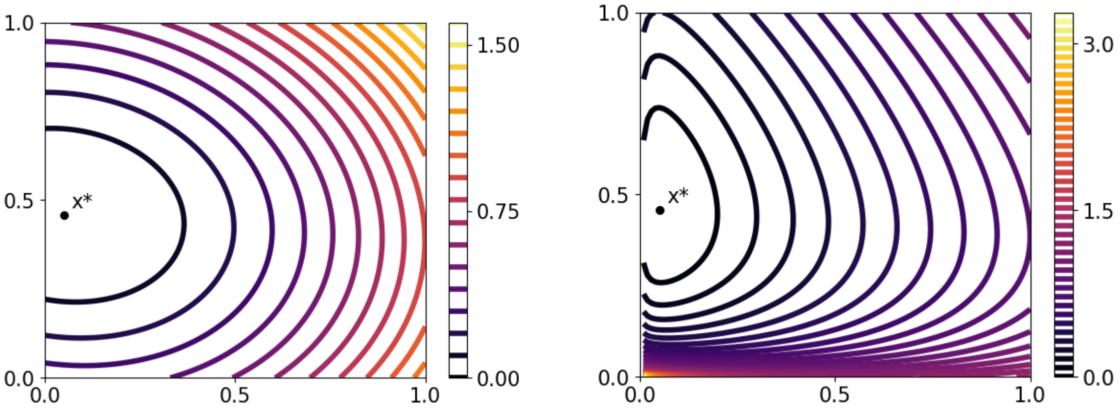
Remark 1. *If the relative smoothness constant L is unknown or too conservative in theory, one can also use Armijo-like line search techniques to determine the step size dynamically (see Chapter 2).*

Remark 2. *Let $x_* \in \operatorname{argmin}_{\mathcal{C}} f$. In order to take $u = x_*$ in Equations (1.2) and (1.3) and obtain a bound on the suboptimality gap $f(x_k) - f(x_*)$, we need x_* to belong to the domain of h. In most cases, this condition is trivially satisfied. However, it can fail if x_* lies on the boundary of $\mathcal{C}$ and $\operatorname{dom} h$ is open, such as for the log kernel.*

Benefits of Bregman gradient descent with relative regularity. The convergence rate of the Bregman Gradient method generalizes that of Euclidean gradient descent on L-smooth and strongly convex functions [Nesterov, 2003, Chap. 2]. Using a different geometry than the Euclidean one can benefit in two types of situations:

1. **Problems with unbounded curvature:** in some problems, the Hessian $\nabla^2 f(x)$ grows unbounded as x reaches some boundary points of $\mathcal{C}$; this is the case for instance with functions involving the Kullback-Leibler divergence (Figure 1.1). Therefore, f is not L-smooth in the standard sense, but relative smoothness can be established with a well chosen kernel that takes into account this singularity. This allows to apply the Bregman gradient method with a fixed step size and ensure global convergence.

2. **Bregman preconditioning:** in other situations, even if f is globally L-smooth, the performance of gradient methods can be improved by choosing a kernel that provides a better approximation of the objective, i.e., such that the gap between $\nabla^2 h$ and $\nabla^2 f$ is as small as possible. There is however a tradeoff, as a tighter approximation usually implies that the subproblem defining the Bregman iterate is harder to solve. This is analogous to the effect of preconditioning for solving linear systems.



(a) $f(x) = \frac{1}{2}\|Ax - b\|^2$

(b) $f(x) = D_{\mathrm{KL}}(b, Ax)$

Figure 1.1: Level lines of the objective function of a linear inverse problem on $\mathbb{R}^2_+$, with the same data $A \in \mathbb{R}^{2\times 2}, b \in \mathbb{R}^2$ and different loss functions. Left: Euclidean loss (Gaussian noise). Right: Kullback-Leibler loss (Poisson noise). Unlike the Euclidean loss, the Poisson loss is not globally smooth as its Hessian diverges near some boundary points of $\mathbb{R}^2_+$.

1.2 Examples of relatively-smooth problems

We now provide a list of examples of relatively-smooth problems with various applications.

1.2.1 L-smooth objective and strongly convex kernel

Prior to the introduction of relative smoothness, most analyses of Bregman gradient descent for differentiable objectives made the assumption that the objective f is L-smooth and that the kernel h is σ-strongly convex with respect to some general norm $\|\cdot\|_E$ [Auslender and Teboulle, 2006, Walid et al., 2015]. This is a particular case of a relatively-smooth problem, as we have for $x, y \in \operatorname{int} \mathcal{C}$,

$$\begin{aligned} f(x) &\leq f(y) + \langle \nabla f(y), x - y\rangle + \frac{L}{2}\|x - y\|_E^2 \\ &\leq f(y) + \langle \nabla f(y), x - y\rangle + \frac{L}{\sigma} D_h(x, y) \end{aligned} \tag{1.4}$$

by definition of smoothness and strong convexity with respect to the norm $\|\cdot\|_E$. However, this is a restrictive setting as many problem of interest are not globally L-smooth. Or, even if they are, tighter relative smoothness constants can be established by eliminating the need for the intermediate bound involving $\|\cdot\|_E$ in the inequality (1.4). We provide such examples in the sequel.

1.2.2 Kernels on nonnegative orthant

This category corresponds to kernels defined on the nonnegative orthant $\mathcal{C} = \mathbb{R}^d_+$. These kernels are mainly designed to tackle problems that have unbounded curvature near the boundary of $\mathcal{C}$.

Log kernel: inverse problems with Poisson noise. Consider the problem

$$\min_{x \in \mathbb{R}^d_+} f(x) = D_{\mathrm{KL}}(b, Ax) \tag{1.5}$$

where $D_{\mathrm{KL}}(x, y) = \sum_{i=1}^m x^{(i)} \log\left(\frac{x^{(i)}}{y^{(i)}}\right) - x^{(i)} + y^{(i)}$ denotes the Kullback-Leibler divergence, $A \in \mathbb{R}^{m \times d}_+$ is a measurement matrix and $b \in \mathbb{R}^m_+$ is a vector of observations. Problem (1.5) models the task of recovering an unknown signal corrupted by Poisson noise, where A encodes the measurement process. It is a fundamental problem in signal processing, with a large range of applications in astronomy, microscopy [Bertero et al., 2009] and medical imaging [Ollinger, 1994, Ben-Tal et al., 2001, Kak and Slaney, 2001]. This is a typical situation of an objective function with unbounded curvature, as $\nabla^2 f(x)$ grows to infinity when $A_j x \to 0$ for some $j \in \{0 \ldots m\}$ (see Figure 1.1). However, when choosing the log-kernel

$$h(x) = \sum_{i=1}^d - \log x^{(i)},$$

we can show that f is smooth relative to h with constant $L = \|b\|_1$ [Bauschke et al., 2017] and thus guarantee that BGD converges with a constant step size $\lambda \leq 1/L$.

Entropy kernel: nonnegative entropy regression. Consider now the problem

$$\min_{x \in \mathbb{R}^d_+} f(x) = D_{\mathrm{KL}}(Ax, b)$$

where $A \in \mathbb{R}^{m \times d}_+, b \in \mathbb{R}^m_+$, which differs from (1.5) as D_{KL} is not symmetric. Then, f is smooth relative to the entropy kernel

$$h(x) = \sum_{i=1}^d x^{(i)} \log x^{(i)}$$

with constant $L = \max_{j=1\ldots d} \sum_{i=1}^m A_{ij}$ [Bauschke et al., 2017]. Such a function appears in entropy-regularized optimal transport, see e.g., Chizat et al. [2018], Mishchenko [2019].

1.2.3 Polynomial kernels

In this situation, the objective function is usually a polynomial of degree higher than 2, whose Hessian then grows unbounded as $\|x\| \to +\infty$. Choosing an appropriate polynomial kernel allows to prove relative smoothness globally, and possibly to improve conditioning.

Quartic polynomials. A large class of polynomial optimization problems is composed of *quadratic inverse problems*, that is, with objectives of the form

$$\min_{x \in \mathbb{R}^d} f(x) = \sum_{j=1}^m (\langle x, A_j x \rangle - b_j)^2$$

where $A_j \in \mathbb{R}^{d\times d}, b_j \in \mathbb{R}, j = 1 \dots m$. Quadratic inverse problems have a broad range of applications, such as phase retrieval [Candès et al., 2015], matrix factorization and unsupervised learning (see Chapter 2 and references therein). As shown by Bolte et al. [2018], such functions are smooth relative to the quartic kernel

$$h(x) = \frac{1}{4}\|x\|^4 + \frac{1}{2}\|x\|^2.$$

In Chapter 2, we specialize the analysis to low-rank matrix problems and leverage this structure to provide tighter kernels.

Another important application is given in Nesterov [2020, 2021], Grapiglia and Nesterov [2021], where the authors propose to use a refined quartic kernel for solving the subproblem that appears in regularized third-order tensor methods.

Higher-order polynomials. Kernels of higher degree have also been considered for deep linear neural networks [Mukkamala et al., 2019] and structure learning on causal models [Romain and D'Aspremont, 2020].

1.2.4 Other applications

Statistical preconditioning for distributed optimization. Consider the finite-sum minimization problem

$$\min_{x\in\mathbb{R}^d} f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$$

where $f_1, \dots f_n$ are differentiable functions with similar structure; e.g., the loss function on different parts of the dataset in the setting of empirical risk minimization. Then, *statistical preconditioning* consists in choosing the kernel h as

$$h(x) = f_{i_0}(x) + \frac{\beta}{2}\|x\|^2$$

for some $i_0 \in \{1 \dots n\}$, where $\beta > 0$ is a given constant.Typically, f_{i_0} is the loss function on a part of the dataset of size n_{prec}. This idea, proposed originally by Shamir et al. [2014], has been developped further through the lens of relative regularity by Hendrikx et al. [2020]. The latter show that, on random datasets and with high probaiblity, f is smooth and strongly convex relative to h with a relative condition number κ_{rel} that satisfies

$$\kappa_{\text{rel}} = 1 + \mathcal{O}\left(\frac{\kappa_{\text{eucl}}}{n_{\text{prec}}}\right)$$

where κ_{eucl} is the standard Euclidean condition number of f. Thus, this kernel allows to improve conditioning over the Euclidean setting. There is however a tradeoff, as computing the Bregman iterates becomes harder when n_{prec} is large. Nevertheless, this method is advantageous in the setting of distributed optimization, where the Bregman subproblem is solved efficiently by a central server, and the goal is to reduce the total number of iterations performed, since computing the gradient of f requires costly communication between different machines.

Algorithmic reinterpretations: EM and Sinkhorn. Some classical iterative methods can also be interpreted as Bregman gradient descent on relatively-smooth functions. Kunstner et al. [2020] show that the celebrated Expectation-Maximization (EM) for maximizing the likelihood of a statistical model with latent variables is an instance of BGD. For exponential families, the objective function of EM is smooth relative to the log-partition function of the distribution. This sheds light on a new theoretical analysis of EM algorithms, as well as improved local convergence rates.

In the same spirit, Léger [2020] studies the Sinkhorn algorithm, which is widely used for solving matrix scaling and entropy-regularized optimal transport problems. He shows that this scheme can also be seen as an instance of BGD with a relatively-smooth objective[1]. The theoretical analysis establishes that Sinkhorn converges at a subblinear rate, as in Proposition 1.4.

1.3 Extensions and variants of Bregman gradient descent

We now review the algorithmic variants of BGD and extensions of relative smoothness for various classes of optimization problems.

Bregman proximal gradient. An important class of problems in signal processing and machine learning is that of *composite optimization problems* of the form

$$\min_{x \in \mathcal{C}} f(x) + g(x)$$

where f is smooth relative to h and g is a (possibly nonsmooth) function that encodes a penalty on the solution, such as sparsity when g is an ℓ_1 norm. This problem can be tackled with the Bregman *proximal* gradient method

$$x_{k+1} = \operatorname*{argmin}_{u \in \mathcal{C}} \; f(x_k) + g(u) + \langle \nabla f(x_k), u - x_k \rangle + \frac{1}{\lambda_k} D_h(u, x_k), \tag{BPG}$$

provided that g is *simple* enough so that the inner subproblem can be solved easily. This method generalizes the Euclidean proximal gradient method, also known as forward-backward splitting [Combettes and Wajs, 2005]. In their work, Bauschke et al. [2017] prove that (BPG) enjoys a sublinear convergence rate of $\mathcal{O}(1/k)$ when f and g are convex; see also Bùi and Combettes [2019] for a more general analysis.

Nonconvex problems. Bolte et al. [2018] extend the Bregman proximal gradient method to relatively-smooth nonconvex objective functions, and establish the following convergence rate of the stationarity measure:

$$\min_{i=1\ldots k} D_h(x_i, x_{i-1}) \leq \mathcal{O}\left(\frac{1}{k}\right).$$

Moreover, they prove convergence to a critical point when the objective satisfies the Kurdyka-Lojasiewicz property [Bolte et al., 2007] as well as some additional local regularity properties. In subsequent work, Bauschke et al. [2019] study different conditions for guaranteeing local linear convergence to a stationary point. Ahookhosh et al. [2021] describe a line-search method for determining the step size of BGD which enjoys favorable local convergence properties.

[1] Interestingly, the author does not make the connection with other work on relative smoothness and seems to have discovered the same notion independently.

Inertial variants of BGD have been proposed for nonconvex problems with similar convergence guarantees and improved numerical performance [Mukkamala and Ochs, 2019, Zhang et al., 2019, Mukkamala et al., 2020]. In these methods, the update of BGD is augmented with an additional momentum term of the form $\alpha_k(x_k - x_{k-1})$, where the coefficient α_k is determined by a line search procedure.

Stochastic gradients. Hanzely and Richtárik [2018], Davis et al. [2018] analyze stochastic variants of the Bregman gradient method, under additional restrictive regularity assumptions. We study stochastic BGD for general relatively-smooth problems in Chapter 6, as well as variance reduction techniques for finite-sum objectives.

Bregman block coordinate methods. In some problems, it can be advantageous to leverage a block coordinate structure of the type

$$x = (x^1, \dots, x^p)$$

where $x^i \in \mathbb{R}^{d_i}$ and $\sum_{i=1}^p d_i = d$, and to assume that f is smooth relative to a different kernel h_i on each coordinate block. This allows the application of Bregman block coordinate methods, which alternate updates of the form

$$x_{k+1} = \operatorname*{argmin}_{u \in \mathbb{R}^{d_i}} \langle \nabla_i f(x_k), u - x_k^i \rangle + \frac{1}{\lambda_k} D_{h_i}(u, x_k^i),$$

for $i \in \{1 \dots p\}$, where $\nabla_i f$ denotes the partial gradient of f on the block i. This approach has been considered in several works [Teboulle and Vaisbourd, 2019, Ahookhosh et al., 2019, Hendrikx et al., Gao et al., 2020]. It is particularly efficient in situations where Bregman block coordinate updates are much easier to compute, or if restricting on separate blocks allows to prove tighter relative smoothness inequalities than on the full coordinates, such as for certain matrix factorization problems.

Relative regularity beyond differentiable functions. The idea of choosing a Bregman geometry that is adapted to the singularities of the objective through *relative regularity* has also been extended to other settings than that of differentiable functions, such as nonsmooth convex functions [Lu, 2019], online learning [Antonakopoulos et al., 2020], variational inequalities and monotone operators [Antonakopoulos et al., 2019, Stonyakin et al., 2020, Cohen et al., 2020].

Variants of relative smoothness. Gutman and Peña [2020] extend the notion to other type of distances than Bregman divergences, and to also take into account the geometry of the optimization set. Maddison et al. [2021] propose a dual variant of relative smoothness, which is invariant to translation (unlike the standard version), with application to ℓ_p norm regression.

1.4 The issue of acceleration and non-homogeneity

Algorithm 2 Accelerated Bregman gradient method in generic form

```
Input: initial point $x_0 \in \operatorname{int} \mathcal{C}$
Set $z_0 = x_0$.
for k = 0,1,... do
    Determine interpolation coefficient $\alpha_k$ and step size $\lambda_k$,
    $y_k = (1-\alpha_k)x_k + \alpha_k z_k$,
    $z_{k+1} = \operatorname{argmin}\{\langle \nabla f(y_k), u - y_k\rangle + \frac{1}{\lambda_k} D_h(u, z_k) \,|\, u \in \mathbb{R}^n\}$,
    $x_{k+1} = (1-\alpha_k)x_k + \alpha_k z_{k+1}$,
end for
```

Acceleration. For convex relatively-smooth functions, Proposition 1.4 states that BGD converges at a rate of $\mathcal{O}(1/k)$ in function values, which is a rather *slow* rate, both in theory and in practice. An important question is whether there exists a first-order Bregman algorithm that achieves a better convergence rate on the same class of problems.

In the Euclidean setting, the answer is positive and was provided in the seminal work of Nesterov [1983][2]. He showed that the accelerated gradient method, obtained by adding a simple and well-chosen linear extrapolation step to gradient descent, achieves a rate of $\mathcal{O}(1/k^2)$ on L-smooth convex functions, and that this worst-case rate is optimal among all first-order Euclidean methods [Nesterov, 2003]. Significant improvement in convergence speed is also observed in practice. This so-called *acceleration technique* has been adapted to improve the efficiency of several other algorithms in continuous optimization; see the monograph by D'Aspremont et al. [2021] for a recent survey.

Naturally, the acceleration technique has been applied to Bregman gradient methods; most instances of accelerated BGD follow the generic form described in Algorithm 2. The choice of the interpolating coefficients α_k and the step size parameters λ_k are crucial for guaranteeing fast convergence. Several choices have been proposed in different settings:

- **Smooth objective and strongly convex kernel:** the first instance of accelerated BGD was studied by Auslender and Teboulle [2006], under the assumption that f is $L_{\|\cdot\|_E}$-smooth and h is $\sigma_{\|\cdot\|_E}$-strongly convex with respect to some norm $\|\cdot\|_E$. In this setting, they show that Algorithm 2 with the same choice of coefficients as Nesterov's accelerated gradient method (that is, $\lambda_k \propto k$ and $\alpha_k \propto 1/k$) yields the improved convergence rate of $\mathcal{O}(1/k^2)$. However, the algorithm's performance is dependent on $\sigma_{\|\cdot\|_E}/L_{\|\cdot\|_E}$, which can be very small or even equal to 0 in the context of problems with unbounded curvature.

- **Relatively-smooth objective:** in this setting, the first tentative of acceleration was proposed by Hanzely et al. [2021]. Their analysis rely on the *triangle scaling exponent* of the Bregman divergence, that is, the largest exponent γ such that

$$D_h\big((1-\theta)\,x + \theta z, (1-\theta)\,x + \theta y\big) \leq \theta^\gamma D_h(x,y), \quad \forall \theta \in [0,1], \qquad \text{(TSE)}$$

[2] There were actually earlier versions of accelerated algorithms proposed by A. Nemirovski which were slightly more complex as they required to perform a two-dimensional minimization subroutine, see e.g., Bubeck [2019].

for every $x, y, z \in \operatorname{int} \mathcal{C}$. Under such property, the authors show that an appropriate choice of coefficients in Algorithm 2 allows to achieve a $\mathcal{O}(1/k^\gamma)$ convergence rate. However, besides quadratic kernels for which $\gamma = 2$ and the entropy where $\gamma = 1$, the inequality (TSE) usually does not hold globally. It is nevertheless valid on small bounded subsets under sufficient regularity of h, as the Bregman divergence locally behaves as a quadratic function. Thus, the authors propose adaptive variants of accelerated BGD to take advantage of this property. Qualitative arguments can then be used to show that the improved rate of $\mathcal{O}(1/k^2)$ is reached asymptotically, and numerical experiments demonstrate improved performance over standard BGD.

Similar results are obtained by Hendrikx et al. [2020] with application to statistical preconditioning for distributed optimization. They propose an adaptive variant of Algorithm 2 which also leverages relative strong convexity. The analysis shows that an improved linear convergence rate of $\mathcal{O}\left((1 - \sqrt{\frac{\mu}{L}})^k\right)$ is reached asymptotically.

Although numerical experiments are encouraging, there is still a gap in our theoretical understanding of accelerated Bregman gradient methods in the relatively-smooth setting. In Chapter 3, we show that in the general case, the $\mathcal{O}(1/k)$ rate of BGD is *optimal* among Bregman first-order methods and therefore that global acceleration is out of reach without additional regularity assumptions.

Non-homogeneity and other difficulties. The main difficulty when trying to transpose convergence results of Euclidean methods to the Bregman setting arises from the lack of homogeneity and translational invariance of the Bregman divergence. Indeed, for non-quadratic kernels, the quantity

$$D_h(x + \lambda v, x)$$

depends on the point x, and is not proportional to λ^2, whereas it reduces to $\lambda^2 \|v\|^2$ in the Euclidean setting. This issue is met when analyzing methods that combine gradients taken at different points and those that rely on step size scaling. This is the case for accelerated methods as well as variance reduction techniques for stochastic optimization (see Chapter 6).

Although it is a different setting, the difficulty of accelerating gradient methods also arises in Riemannian optimization for similar reasons [Zhang and Sra, 2018]; see Hamilton and Moitra [2021] for a negative result on the hyperbolic plane and an interesting intuitive explanation.

Besides acceleration, there is also difficulty in proving the convergence of the iterates of Bregman methods towards the optimal point. Bolte and Pauwels [2020] provide an an instance of BGD where the iterate sequence $\{x_k\}_{k\geq 0}$ does not converge and cycles indefinitely. In their example, the objective function is linear; the issue is created uniquely by the choice of a highly pathological Bregman kernel.

1.5 A brief history of Bregman methods before relative smoothness

Optimization methods with Bregman divergences have also been considered for a long time outside of the relatively-smooth setting. In this section, we provide a historical overview.

Bregman projections. Given a kernel h, the Bregman projection of a point x on a convex set $A \subset \mathbb{R}^d$ is

$$\mathrm{proj}_A^h(x) = \operatorname*{argmin}_{u \in A} D_h(u, x). \tag{1.6}$$

This tool was first introduced by Bregman [1967], along with the corresponding divergence, in order to find a point in the intersection of N convex sets $A_1 \dots A_N$ by alternating such projections. The main interest of choosing a specific kernel h is that it induces a bias that drives the algorithm towards the part of the intersection set that minimizes h[3]. More precisely, with a proper initialization, the method of alternating Bregman projections converges to the solution of the problem

$$\min_{x \in \mathrm{dom}\ h} h(x) \quad \text{s.t.} \quad x \in A_1 \cap \dots \cap A_N.$$

See also Bauschke and Borwein [1997] for a more detailed analysis of Bregman projections.

Bregman proximal methods. For a convex proper function $g : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ and a kernel h, the Bregman proximal map of g is defined as

$$\mathrm{prox}_g^h(x) = \operatorname*{argmin}_{u \in \mathbb{R}^d} g(u) + D_h(u, x),$$

which generalizes the Moreau proximal map [Moreau, 1965] to the non-Euclidean setting. Note also that taking g as the indicator function of a convex set recovers the Bregman projection (1.6). This operator has been introduced by Censor and Zenios [1992], who study the Bregman proximal point algorithm

$$x_{k+1} = \operatorname*{argmin}_{u \in \mathbb{R}^d} g(u) + \frac{1}{\lambda_k} D_h(u, x_k) \tag{BPP}$$

for minimizing the (possibly nonsmooth) function g on $\mathcal{C}$. Simultaneously, Teboulle [1992] studied the proximal map with Bregman divergence and also an alternative class of divergences called *ϕ-divergences*. He showed that these maps enjoyed a smoothing effect, as does the Moreau proximal map in the Euclidean case. The (BPP) scheme was then extended by Eckstein [1993] to the more general setting of monotone operators, which allows to build nonquadratic versions of augmented Lagrangian methods.

One of the main advantages of using a Bregman divergence adapted to $\mathcal{C}$ in proximal methods is that the kernel acts as a *barrier*, forcing the iterates to belong to the interior of $\mathcal{C}$, thus ensuring favorable regularity properties. For instance, augmented Lagrangian methods with a well-chosen Bregman kernel have a twice continuously differentiable Lagrangian, unlike the Euclidean counterpart, allowing the use of efficient second-order methods for computing the iterates. Further analysis is provided in Chen and Teboulle [1993], Iusem et al. [1994].

Nonsmooth kernels for ill-posed inverse problems. Although Bregman methods are generally used with differentiable kernels, nonsmooth convex kernels can also be used, provided that the subgradient defining D_h is well-chosen. This extension was first studied by Kiwiel [1997] for the Bregman proximal point method.

[3] This idea of *implicit bias* of gradient methods has been re-discovered recently in the context of mirror descent [Gunasekar et al., 2018].

In this situation, the Bregman kernel is chosen so as to exploit the idea of *implicit bias*: in the same way as for Bregman projections (see the paragraph above), the iterates of (BPP) converge towards the part of $\operatorname{argmin} f$ that also minimizes h. This can be useful for solving ill-posed inverse problems: for instance, taking h to be a ℓ_1 norm allows to induce sparsity in the iterates of the algorithm. This idea has had great success in compressed sensing and image denoising, where the Bregman proximal point scheme is also known as *Bregman iteration*, or *Bregman iterative regularization* [Osher et al., 2005, Yin et al., 2008, Goldstein and Osher, 2009], see also Benning and Riis [2021] for a recent survey.

Mirror descent subgradient method. The mirror descent algorithm, designed for minimizing a possibly nonsmooth convex function f over a set $\mathcal{C}$, writes

$$x_{k+1} = \operatorname*{argmin}_{u \in \mathbb{R}^d} \langle f'(x_k), u - x_k \rangle + \frac{1}{\lambda_k} D_h(u, x_k), \tag{MD}$$

where $f'(x_k) \in \partial f(x_k)$ is any subgradient of f at x_k, and the sequence of step sizes $\{\lambda_k\}_{k \geq 0}$ is decreasing to 0. Mirror descent was originally proposed by Nemirovski [1979], Nemirovski and Yudin [1983] in the form of

$$x_{k+1} = \nabla h^* \left[\nabla h(x_k) - \lambda_k f'(x_k) \right] \tag{MD'}$$

and applied by Ben-Tal et al. [2001] to Poisson inverse problems (which were then classified as nonsmooth problems, because of the unbounded curvature). Later, Beck and Teboulle [2003] provided a simplified analysis and showed that, under appropriate assumptions on h, the update (MD') can be rewritten as the subgradient scheme with Bregman divergence desribed in (MD). A central element in the analysis of Mirror Descent is the choice of a (possibly non-Euclidean) norm $\|\cdot\|_E$ for the space E of iterates x_k, along with the corresponding dual norm $\|\cdot\|_{E^*}$ for the space E^* of gradients $\nabla h(x_k), f'(x_k)$. The kernel h is assumed to be strongly convex with respect to $\|\cdot\|_E$, and the subgradients of f are assumed to be bounded with respect to $\|\cdot\|_{E^*}$ by a constant $L_{\|\cdot\|_E}$ on $\mathcal{C}$. The efficiency estimate of (MD) is proportional to

$$L_{\|\cdot\|_E} \left(D_h(x_*, x_0) \right)^{1/2}.$$

Therefore, the goal is to choose the geometry that minimizes this estimate on the set $\mathcal{C}$. The most classical example is the unit simplex $\mathcal{C} = \{x \in \mathbb{R}^d_+ \, : \, \sum_i x_i = 1\}$, for which choosing the ℓ_1 norm along with the entropy kernel allows to improve the efficiency by a factor of $(d/\log d)^{1/2}$ over the Euclidean geometry. This is a considerable theoretical and practical gain for large-scale problems. As an additional advantage, the Bregman projection on the simplex can be computed in closed form, unlike the Euclidean one. A detailed analysis of the computational complexity of mirror descent in various situations is provided in the survey by Juditsky and Nemirovski [2011].

Beyond nonsmooth convex minimization, the mirror descent scheme has been studied in the context of stochastic convex minimization and saddle-point problems [Nemirovski et al., 2009, Duchi et al., 2010] as well as online learning [Hazan, 2011, Bubeck, 2011]. As the simplex models discrete probability measures, entropic mirror descent has had great success in online learning, where it is also known as *multiplicative weight updates*. More recently, mirror descent schemes have also been used for solving reinforcement learning problems [Tomar et al., 2020, Lan, 2021].

From mirror descent to Bregman gradient method and relative smoothness. The mirror descent method (MD) has a similar form to the Bregman gradient scheme. However, mirror descent has mainly been applied in the context of nonsmooth, stochastic and online convex optimization. In that setting, the kernel h is usually chosen accordingly to the constraint set $\mathcal{C}$ and the step size sequence $\{\lambda_k\}_{k\geq 0}$ needs to be decreasing towards 0 to account for nonsmoothness.

The idea of adapting the kernel to the curvature of the objective function through the simple condition

$$\nabla^2 f(x) \preceq L\nabla^2 h(x)$$

has opened new possibilities for applying Bregman methods to differentiable objectives. Interestingly, the relative smoothness assumption was suggested for the first time by Birnbaum et al. [2011] in the context of algorithmic game theory, but remained unnoticed by the optimization community. Later, Bauschke et al. [2017] discovered the same concept independently and popularized the subject among researchers in this field.

Chapter 2

Quartic First-Order Methods for Low-Rank Minimization

Chapter Abstract

We study a general nonconvex formulation for low-rank minimization problems. We use recent results on relatively-smooth optimization to provide efficient and scalable algorithms. Our approach uses the geometry induced by the Bregman divergence of well-chosen kernel functions; for unconstrained problems we introduce a novel family of Gram quartic kernels that improve numerical performance.

Numerical experiments on Euclidean distance matrix completion and symmetric nonnegative matrix factorization show that our algorithms scale well and reach state of the art performance when compared to specialized methods.

Reference: this chapter is based on a publication in Journal of Optimization Theory and Applications [Dragomir et al., 2021a].

2.1 Introduction

Consider a low-rank semidefinite program, written

$$\min F(Y) \quad \text{subject to } Y \succeq 0, \operatorname{rank}(Y) \leq r \tag{SDP-r}$$

in the variable $Y \in \mathbb{R}^{n\times n}$, where $F : \mathbb{R}^{n\times n} \to \mathbb{R}$ is typically a differentiable convex function with Lipschitz continuous gradient and $r \leq n$ is the target rank. Fundamental applications of (SDP-r) arise in various areas including matrix completion [Candès and Recht, 2009, Cai et al., 2010, Jain et al., 2013], matrix sensing [Recht et al., 2007], Euclidean matrix completion [Mishra et al., 2011, Fang and O'Leary, 2012], phase retrieval [Candès et al., 2015], robust principal component analysis [Chen and Wainwright, 2015], to name a few.

A popular approach to solving (SDP-r), known as the Burer-Monteiro formulation [Burer and Monteiro, 2005], consists in representing Y as $Y = XX^T$ to solve

$$\min \Psi(X) \triangleq F(XX^T) + g(X) \tag{P}$$

in the variable $X \in \mathbb{R}^{n\times r}$, where g is a lower semicontinuous *simple* function (in a sense that will be made clear later) that enforces additional constraints or penalties on the factor X. We will write $f : \mathbb{R}^{n\times r} \to \mathbb{R}$ the factorized function with $f(X) = F(XX^T)$.

This reformulation has been thoroughly studied for various low rank minimization problems [Mishra et al., 2014, Tu et al., 2016, Candès et al., 2015, Zheng and Lafferty, 2015, Chen and Wainwright, 2015, Bhojanapalli et al., 2016, Zhao et al., 2015, Sun and Luo, 2016, Park et al., 2016]. It has several benefits: first, the rank constraint is directly enforced by the factorization XX^T. Second, it allows adding penalty functions g to impose additional structural properties on the factor, like nonnegativity or sparsity. More importantly, the problem size is reduced from n^2 to nr variables, which makes it far more efficient on large-scale problems, since r is usually much smaller than n.

This formulation comes however with an important drawback: the objective function of (P) becomes nonconvex, even when F is originally convex. Therefore, local optimization methods can generally only hope to find a stationary point, or at best a local minimum. Nevertheless, recent work shows convergence towards a global optimum for a close enough initialization [Tu et al., 2016, Bhojanapalli et al., 2016, Park et al., 2016], or under additional statistical assumptions about the problem [Chen and Wainwright, 2015, Zheng and Lafferty, 2015, Ge et al., 2016]. Although these global optimality results often impose restrictive assumptions that may not be satisfied in practice, they help to explain why the use of local algorithms for solving (P) usually leads to satisfactory solutions in practice.

The most commonly used algorithm to solve these problem formulations is some variant of the proximal gradient method. However, a critical issue with gradient schemes is the choice of step sizes, which significantly impacts performance. This step size choice is closely related to the smoothness of the objective. In particular, when it has a L-Lipschitz continuous gradient with respect to the Euclidean norm, standard gradient methods can be applied with a step size lying in $]0, 1/L]$. This smoothness assumption is used in the broad majority of theoretical analyses of gradient algorithms, yet there are many cases where it is not satisfied [Bauschke et al., 2017, Bolte et al., 2018]. In particular, it does not hold for the general Burer-Monteiro low-rank problem, as we will show in what follows.

Of course, there is a way to circumvent this issue in classical Euclidean methods, by using an Armijo line search [Lin, 2007]. However, in some cases, this naive line search strategy generates very small step sizes which in turn involve costly subroutines. Other approaches impose a step size that is only proven to be valid in a small neighborhood of the optimum [Bhojanapalli et al., 2016, Park et al., 2016].

Bregman gradient methods. We adopt an original approach based on a recent line of work on non-Euclidean gradient methods [Bauschke et al., 2017, Bolte et al., 2018] and subsequent work [Lu et al., 2018]. Unlike standard gradient descent that uses the uniform Euclidean geometry, the NoLips method, also known as Bregman proximal gradient/Mirror descent, uses the Bregman divergence induced by a well-chosen convex *kernel* function. This allows the algorithm to take gradient steps that are more adapted to the geometry of the problem, advancing faster in directions where the gradient of the objective changes slowly, thus improving convergence speed. The kernel function is chosen so that the objective function satisfies a compatibility condition called *relative smoothness* [Bauschke et al., 2017, Lu et al., 2018], which is a generalization of the usual smoothness assumption mentioned earlier.

In our setting, the objective has a quartic growth, hence choosing the geometry induced by a quartic polynomial will prove to be efficient.

Contributions. In this work, we focus on deriving efficient algorithms to find stationary points of nonconvex low-rank problems. Our main contribution is to identify favorable non-Euclidean geometries for these problems, induced by well-chosen quartic kernels.

We first study a simple quartic *norm* kernel that is compatible with various regularization terms. We then introduce a novel family of quartic kernels that we call *Gram kernels*, which can be applied to unregularized problems. They provide richer geometries which greatly improve convergence speed with little impact on the iteration complexity. We also extend the NoLips/Bregman Gradient scheme to Dyn-NoLips, allowing for adaptive step size strategies.

To highlight the benefits of our approach, we study applications to symmetric nonnegative matrix factorization and Euclidean distance matrix completion and show competitive numerical performance compared to specialized algorithms for these problems.

Notation. For a square matrix M, we denote its trace $\mathbf{Tr}\, M = \sum_{i=1}^n M_{ii}$. For two matrices X and Y of same size, we denote the standard Euclidean inner product and norm by $\langle X, Y\rangle = \mathbf{Tr}(X^TY)$ and $\|X\| = \sqrt{\mathbf{Tr}(X^TX)}$. For a function $f : \mathbb{R}^{n\times r} \to \mathbb{R}$, we denote by $\nabla F(X)$ its gradient matrix $\nabla F(X)_{ij} = \frac{\partial F(X)}{\partial x_{ij}}$ and by $\nabla^2 F(X)[U, V]$ the second derivative at X in the directions $U, V \in \mathbb{R}^{n\times r}$. I_r denotes the identity matrix of size $r \times r$. For two square matrices X, Y, we write $X \preceq Y$ if the matrix $Y - X$ is positive semidefinite. We write $\|\mathcal{A}\|_{\text{op}}$ for the operator norm of a linear application $\mathcal{A}$.

2.2 Quartic geometries for low-rank minimization

2.2.1 Problem Setup

We first state our standing assumptions for Problem (P).

Assumption 2.1. (a) *$F : \mathbb{R}^{n\times n} \to \mathbb{R}$ is a twice continuously differentiable function which is μ_F-strongly convex and L_F-smooth, i.e.,*

$$\begin{aligned} \langle \nabla F(X) - \nabla F(Y), X - Y\rangle &\geq \mu_F\|X - Y\|^2, \\ \|\nabla F(X) - \nabla F(Y)\| &\leq L_F\|X - Y\| \quad \forall X, Y \in \mathbb{R}^{n\times n}, \end{aligned}$$

(b) *$g : \mathbb{R}^{n\times n} \to \mathbb{R} \cup \{+\infty\}$ is a closed convex proper function,*

(c) *$\min_{\mathbb{R}^{n\times r}} \Psi > -\infty$.*

Our analysis will involve the following lemma.

Lemma 2.1. *Let $F : \mathbb{R}^{n\times n} \to \mathbb{R}$ be a twice differentiable μ_F-strongly convex and L_F-smooth function. Then, the function $G := F - \frac{\mu_F}{2}\|\cdot\|^2$ is convex and $(L_F - \mu_F)$-smooth.*

Proof. It suffices to use the second-order characterization [Nesterov, 2003] and notice that, for $Y, U \in \mathbb{R}^{n\times n}$, we have $\nabla^2 G(Y)[U, U] = \nabla^2 F(Y)[U, U] - \mu_F\|U\|^2$ and hence

$$\mu_F\|U\|^2 \leq \nabla^2 F(Y)[U, U] \leq L_F\|U\|^2 \implies 0 \leq \nabla^2 G(Y)[U, U] \leq (L_F - \mu_F)\|U\|^2.$$

■

We write $f : \mathbb{R}^{n\times r} \to \mathbb{R}$ the factorized function defined by

$$f(X) := F(XX^T).$$

Therefore Problem (P) can be written as a standard *composite* optimization problem

$$\min_{X\in\mathbb{R}^{n\times r}} \Psi(X) = f(X) + g(X). \tag{2.1}$$

2.2.2 Relative Smoothness and the Bregman Proximal Gradient Map

In this section, we recall the framework of Bauschke et al. [2017], Bolte et al. [2018] to derive non-Euclidean gradient methods for solving composite problems of the form (2.1).

The first essential step is the choice of a *distance kernel.* In our context we choose a differentiable strictly convex function $h : \mathbb{R}^{n\times r} \to \mathbb{R}$, with $\operatorname{dom} h = \mathbb{R}^{n\times r}$ (although more general distance kernels can be used). The distance kernel h induces in turn a *Bregman divergence*

$$D_h(X, Y) = h(X) - h(Y) - \langle \nabla h(Y), X - Y\rangle.$$

Note that D_h is not a proper distance, as it is not symmetric in general. However D_h enjoys a distance-like separation property: $D_h(X, X) = 0$ and $D_h(X, Y) > 0$ for $X \neq Y$. The choice of a distance kernel suited to the function f is guided by the following relative smoothness condition, also called generalized Lipschitz property.

Definition 2.2 (Relative smoothness)**.** *We say that a differentiable function f on $\mathbb{R}^{n\times r}$ is L-smooth relative to the kernel function h if there exists $L > 0$ such that for every $X, Y \in \mathbb{R}^{n\times r}$,*

$$f(X) \leq f(Y) + \langle \nabla f(Y), X - Y\rangle + L\, D_h(X, Y). \tag{RelSmooth}$$

For twice differentiable functions, relative smoothness has an elementary characterization: f is L-smooth relative to h if and only if

$$\nabla^2 f(X)[U, U] \leq L\, \nabla^2 h(X)[U, U], \quad \forall X, U \in \mathbb{R}^{n\times r}. \tag{2.2}$$

Notice that if $h(X) = \frac{1}{2}\|X\|^2$, then $D_h(X, Y) = \frac{1}{2}\|X - Y\|^2$ and we recover the standard Euclidean descent lemma that would be implied by Lipschitz continuity of the gradient of f.

Bregman proximal gradient map. Now that we are equipped with a non-Euclidean geometry generated by h, we define the Bregman proximal gradient map with step size λ as follows.

$$T_\lambda(X) = \operatorname*{argmin}_{U\in\mathbb{R}^{n\times r}} \left\{ g(U) + f(X) + \langle \nabla f(X), U - X\rangle + \frac{1}{\lambda} D_h(U, X) \right\}, \tag{2.3}$$

which consists in minimizing a surrogate for Ψ where f has been replaced by the upper approximation given by (RelSmooth) and the nonsmooth part g is kept intact, generalizing thus the approach used in the proximal gradient method. The relative smoothness condition ensures that this operation decreases the objective Ψ when $\lambda \in]0, 1/L]$. This map is the basic brick for non-Euclidean methods à la Bregman. The simplest method is the Bregman proximal gradient method, also known as NoLips [Bauschke et al., 2017] and its extension Dyn-NoLips (Algorithm 3), which simply amounts to iterating $X^{k+1} = T_{\lambda_k}(X^k)$, but other possibilities exist using momentum ideas [Auslender and Teboulle, 2006, Hanzely et al., 2021, Mukkamala et al., 2020].

2.2.3 The Quartic Geometry

In order to provide some insight into the quartic geometry of our problem, let us consider the example where F is a *quadratic* function, i.e.,

$$F(Y) = \frac{1}{2}\langle \mathcal{A}Y, Y\rangle + \langle B, Y\rangle \quad \forall Y \in \mathbb{R}^{n\times n}, \tag{2.4}$$

where $B \in \mathbb{R}^{n\times n}$ and $\mathcal{A} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ is some linear map. Then, f writes

$$f(X) = F(XX^T) = \frac{1}{2}\langle \mathcal{A}(XX^T), XX^T\rangle + \langle BX, X\rangle \quad \forall X \in \mathbb{R}^{n\times r}.$$

Clearly, f is a *quartic* function and its gradient is not Lipschitz continuous on $\mathbb{R}^{n\times r}$, as the Hessian "grows" to infinity when $\|X\| \to \infty$. In other words, (RelSmooth) does not hold with the Euclidean kernel $h = \frac{1}{2}\|\cdot\|^2$. We now show that relative smoothness holds with a family of well-chosen quartic kernels, which are more adapted to the geometry of f.

The Quartic Norm Kernel

We begin with a simple quartic kernel, which depends solely on the Frobenius norm of X. Define the *norm kernel* h_N as

$$h_N(X) = \frac{\alpha}{4}\|X\|^4 + \frac{\sigma}{2}\|X\|^2 \quad \forall X \in \mathbb{R}^{n\times r},$$

where $\alpha, \sigma > 0$ are fixed parameters. Note that this kernel is not new by itself, as it has been already studied in Bolte et al. [2018] for vectors in $\mathbb{R}^n$. Our first contribution is to show that it is adapted to every function of our class of problems.

Proposition 2.3 (Norm kernel)**.** *The function f is 1-smooth relative to the norm kernel h_N for $\alpha \geq 6L_F$ and $\sigma \geq 2\|\nabla F(0)\|$.*

Proof. As F is twice differentiable, then so is f and we can use the Hessian characterization (2.2). For $X, U \in \mathbb{R}^{n\times r}$, the second derivative of h_N is written

$$\begin{aligned} \nabla^2 h_N(X)[U,U] &= \alpha\left(\|X\|^2\|U\|^2 + 2\langle X, U\rangle^2\right) + \sigma\|U\|^2 \\ &\geq \alpha\|X\|^2\|U\|^2 + \sigma\|U\|^2. \end{aligned} \tag{2.5}$$

On the other hand, the second derivative of f is

$$\nabla^2 f(X)[U,U] = \nabla^2 F(XX^T)[UX^T + XU^T, UX^T + XU^T] + 2\langle \nabla F(XX^T), UU^T\rangle. \tag{2.6}$$

Since F has a Lipschitz continuous gradient, the standard second derivative inequality yields

$$\nabla^2 F(XX^T)[UX^T + XU^T, UX^T + XU^T] \leq L_F\|UX^T + XU^T\|^2.$$

Now, the second term can be bounded by using the triangle inequality, the Cauchy-Schwarz inequality and the gradient Lipschitz property, to get

$$\begin{aligned} \langle \nabla F(XX^T), UU^T\rangle &= \langle \nabla F(0), UU^T\rangle + \langle \nabla F(XX^T) - \nabla F(0), UU^T\rangle \\ &\leq \|\nabla F(0)\|\,\|U\|^2 + \|\nabla F(XX^T) - \nabla F(0)\|\,\|UU^T\| \\ &\leq \Big(\|\nabla F(0)\| + L_F\|XX^T\|\Big)\|U\|^2 \end{aligned}$$

hence

$$
\begin{aligned}
\nabla^2 f(X)[U,U] &\leq L_F\|UX^T + XU^T\|^2 + 2\left(L_F\|XX^T\| + \|\nabla F(0)\|\right)\|U\|^2 \\
&\leq 2L_F\left(\|UX^T\|^2 + \|XU^T\|^2\right) + 2\left(L_F\|XX^T\| + \|\nabla F(0)\|\right)\|U\|^2 \\
&\leq 6L_F\|X\|^2\|U\|^2 + 2\|\nabla F(0)\|\|U\|^2 \\
&\leq \alpha\|X\|^2\|U\|^2 + \sigma\|U\|^2
\end{aligned} \tag{2.7}
$$

where we used the submultiplicative property of the Frobenius norm, and our choice of parameters α, σ. Combining (2.5) and (2.7) gives that

$$\nabla^2 f(X)[U,U] \leq \nabla^2 h_N(X)[U,U]$$

for all $X, U \in \mathbb{R}^{n\times r}$, hence that f is 1-smooth relative to h [Bauschke et al., 2017]. ■

The Bregman proximal gradient map (2.3) associated with the kernel h_N can be computed easily in closed form. We give its expression in the unconstrained case [Bolte et al., 2018].

Proposition 2.4 (Bregman gradient map for h_N, unconstrained case)**.** *Assume that there is no penalty term, i.e., that $g \equiv 0$. The Bregman gradient map of the norm kernel h_N with step size $\lambda > 0$ is given by*

$$T_\lambda(X) = \frac{1}{\tau_\sigma(\alpha\|U\|^2)}U$$

where

$$U = \nabla h_N(X) - \lambda\nabla f(X) = (\alpha\|X\|^2 + \sigma)X - \lambda\nabla f(X)$$

and $\tau_\sigma(c)$ denotes the unique real solution z to the cubic equation $z^2(z-\sigma) = c$.

Note that $\tau_\sigma(c)$ can be computed in closed form using Cardano's method:

$$\tau_\sigma(c) = \frac{\sigma}{3} + \sqrt[3]{\frac{c+\sqrt{\Delta}}{2} + \frac{\sigma^3}{27}} + \sqrt[3]{\frac{c-\sqrt{\Delta}}{2} + \frac{\sigma^3}{27}} \text{ where } \Delta = c^2 + \frac{4}{27}c\sigma^3.$$

Compared to a standard gradient iteration, the additional operations are elementary and have a minimal impact on the arithmetic complexity.

Constraints and regularization terms. Following the ideas in Bolte et al. [2018], the Bregman proximal gradient map of h_N can also be easily computed in closed form when g is the ℓ_1 norm or the ℓ_0 pseudonorm. As we will show in Section 2.4.1, this is also elementary when g is the indicator function of the nonnegative orthant.

A More Refined Kernel for Unregularized Problems: the Gram Kernel

While the kernel h_N is simple and compatible with many penalties g, a better kernel can be derived for unconstrained instances by considering a richer geometry involving the Gram matrix. Define the *Gram kernel* as

$$h_G(X) = \frac{\alpha}{4}\|X\|^4 + \frac{\beta}{4}\|X^TX\|^2 + \frac{\sigma}{2}\|X\|^2 \quad \forall X \in \mathbb{R}^{n\times r},$$

where $\alpha, \beta \geq 0$, $\sigma > 0$ are given parameters. The Gram kernel is more refined than the previous norm kernel since it incorporates some nonisotropic information with the $\|X^T X\|^2$ term. To show where this term stems from, observe that following Lemma 2.1, F can be decomposed as $F = \frac{\mu_F}{2}\|\cdot\|^2 + \tilde{F}$ where $\tilde{F}$ is $(L_F - \mu_F)$-smooth. Hence f writes

$$f(X) = F(XX^T) = \frac{\mu_F}{2}\|XX^T\|^2 + \tilde{F}(XX^T).$$

Since $\|XX^T\|^2 = \|X^T X\|^2$, the first term can be directly incorporated into the kernel, which allows to prove a tighter relative smoothness inequality.

Proposition 2.5 (Gram kernel)**.** *f is 1-smooth relative to the Gram kernel h_G a choice of constants satisfying $\alpha \geq 2(L_F - \mu_F)$, $\beta \geq 2L_F$ and $\sigma \geq 2\|\nabla F(0)\|$.*

Proof. This amounts to refine the analysis of the proof of Proposition 2.3. Let $X, U \in \mathbb{R}^{n\times r}$. The second derivative of h_G at X in the direction U writes

$$\begin{aligned}\nabla^2 h_G(X)[U,U] &= \alpha\left(\|X\|^2\|U\|^2 + 2\langle X, U\rangle^2\right)\\ &\qquad + \beta\left(\frac{1}{2}\|UX^T + XU^T\|^2 + \|U^T X\|^2\right) + \sigma\|U\|^2\\ &\geq \alpha\|X\|^2\|U\|^2 + \beta\left(\frac{1}{2}\|UX^T + XU^T\|^2 + \|U^T X\|^2\right) + \sigma\|U\|^2.\end{aligned}$$

On the other hand, following (2.6) the second derivative of f satisfies

$$\nabla^2 f(X)[U,U] \leq L_F\|UX^T + XU^T\|^2 + 2\langle\nabla F(XX^T), UU^T\rangle.$$

To bound the second term, we use Lemma 2.1 which states that the function $G(Y) := F(Y) - \mu_F\|Y\|^2/2$ is convex and smooth with constant $L_F - \mu_F$. Using the gradient Lipschitz property of G yields

$$\begin{aligned}\langle\nabla F(XX^T), UU^T\rangle &= \langle\nabla F(0), UU^T\rangle + \mu_F\langle XX^T, UU^T\rangle\\ &\qquad + \langle\nabla F(XX^T) - \nabla F(0) - \mu_F(XX^T - 0), UU^T\rangle\\ &= \langle\nabla F(0), UU^T\rangle + \mu_F\|U^T X\|^2 + \langle\nabla G(XX^T) - \nabla G(0), UU^T\rangle\\ &\leq \|\nabla F(0)\|\,\|U\|^2 + \mu_F\|U^T X\|^2 + (L_F - \mu_F)\|XX^T\|\,\|UU^T\|\\ &\leq \|\nabla F(0)\|\,\|U\|^2 + L_F\|U^T X\|^2 + (L_F - \mu_F)\|X\|^2\|U\|^2,\end{aligned}$$

using that $\mu_F \leq L_F$, and so we have

$$\begin{aligned}\nabla^2 f(X)[U,U] &\leq 2(L_F - \mu_F)\|X\|^2\|U^2\| + L_F\|UX^T + XU^T\|^2 + 2L_F\|U^T X\|^2\\ &\qquad + 2\|\nabla F(0)\|\,\|U\|^2\\ &\leq \alpha\|X\|^2\|U\|^2 + \frac{\beta}{2}\|UX^T + XU^T\|^2 + \beta\|U^T X\|^2 + \sigma\|U\|^2\\ &\leq \nabla^2 h_G(X)[U,U]\end{aligned}$$

which shows that, for the prescribed choice of α, β, σ, the function f is 1-smooth relative to h_G.

■

Approximation quality for well-conditioned F. Let us illustrate the advantage of the Gram kernel when F is well-conditioned. For simplicity, assume here that F is a quadratic function, as in (2.4), i.e., $F(Y) = \frac{1}{2}\langle \mathcal{A}Y, Y\rangle + \langle B, Y\rangle$ where $\mathcal{A}$ is a positive semidefinite linear operator on $\mathbb{R}^{n\times r}$, and hence f has a *quartic* and a *quadratic* term

$$f(X) = \frac{1}{2}\langle \mathcal{A}(XX^T), XX^T\rangle + \langle B, XX^T\rangle.$$

The gap between f and h_G with the choice of coefficients prescribed by Proposition 2.5 writes, for $X \in \mathbb{R}^{n\times r}$,

$$\begin{aligned} h_G(X) - f(X) &= \frac{(L_F - \mu_F)}{2}\|X\|^4 + \frac{L_F}{2}\|X^TX\|^2 + \|\nabla F(0)\|\,\|X\|^2 \\ &\quad - \frac{1}{2}\langle \mathcal{A}(XX^T), XX^T\rangle - \langle BX, X\rangle \\ &= \underbrace{\frac{(L_F - \mu_F)}{2}\|X\|^4 + \frac{1}{2}\langle (L_F I - \mathcal{A})(XX^T), XX^T\rangle}_{d_4(X)} \\ &\quad + \underbrace{\langle (\|\nabla F(0)\| I - B)X, X\rangle}_{d_2(X)} \end{aligned}$$

where we separated the gap into a quartic term d_4 and a quadratic term d_2. It can be seen from (2.2) that the quality of approximation of the kernel is given by the difference of the Hessians. Focusing on the quartic part, the Hessian difference is

$$\begin{aligned} \nabla^2 d_4(X)[U,U] &= 2(L_F - \mu_F)\left(\|X\|^2\|U\|^2 + 2\langle X, U\rangle^2\right) + 2\langle (L_F I - \mathcal{A})(XX^T), UU^T\rangle \\ &\quad + \langle (L_F I - \mathcal{A})(UX^T + XU^T), UX^T + XU^T\rangle \\ &\leq 6(L_F - \mu_F)\|X\|^2\|U\|^2 \\ &\quad + \|L_F I - \mathcal{A}\|_{\text{op}}\left(2\|XX^T\|\,\|UU^T\| + \|UX^T + XU^T\|^2\right) \end{aligned}$$

for $X, U \in \mathbb{R}^{n\times r}$. Recalling that F is L_F-smooth and μ_F-strongly convex, we have that $\|L_F I - \mathcal{A}\|_{\text{op}} \leq (L_F - \mu_F)$, therefore

$$\begin{aligned} \nabla^2 d_4(X)[U,U] &\leq (L_F - \mu_F)\left(6\|X\|^2\|U\|^2 + 2\|XX^T\|\,\|UU^T\| + \|UX^T + XU^T\|^2\right) \\ &\leq 12L_F(1 - \frac{\mu_F}{L_F})\|X\|^2\|U\|^2 \end{aligned}$$

which shows that the quality of approximation of the quartic part of f by the Gram kernel depends on the condition number $\kappa_F := L_F/\mu_F$ of F. Note that one could actually refine the analysis by replacing κ_F with the condition number of F restricted to the set of matrices of rank at most $2r$, which can be much smaller. This is the case when the linear map $\mathcal{A}$ satisfies the *restricted isometry property* (RIP), which occurs with high probability in matrix sensing applications with a sufficiently large number n of samples [Recht et al., 2007, Meka et al., 2010, Jain et al., 2013].

Computing the Bregman gradient map. We now show that, when there is no penalty term g, the Bregman gradient map of h_G can be computed efficiently, as it involves solving an easy quartic minimization subproblem of size r.

Proposition 2.6 (Bregman gradient map, Gram kernel)**.** *Assume that $g \equiv 0$. For $X \in \mathbb{R}^{n\times r}$, the Bregman gradient map of f for the Gram kernel h_G with step size $\lambda > 0$ is given by*

$$T_\lambda(X) = V\left[\alpha\,\mathbf{Tr}(Z)I_r + \beta Z + \sigma I_r\right]^{-1}$$

where the matrices V, Z are computed through the routine:

- *set $V = \nabla h_G(X) - \lambda\nabla f(X)$,*
- *diagonalize V^TV as $V^TV = P^TDP$ where $P \in \mathcal{O}_r$ and $D = \mathbf{diag}(\eta_1^2, \dots, \eta_r^2)$,*
- *let $\mu = (\mu_1, \dots, \mu_r)$ be the unique solution of the convex minimization problem*

$$\min_{x\in\mathbb{R}^r} \phi(x) := \frac{\alpha}{4}\|x\|^4 + \frac{\beta}{4}\sum_{i=1}^r x_i^4 + \frac{\sigma}{2}\|x\|^2 - \sum_{i=1}^r \eta_i x_i,$$

- *finally set $Z = P^T\,\mathbf{diag}\left[\mu_1^2, \dots, \mu_r^2\right]P$.*

Proof. When $g \equiv 0$, The Bregman gradient map of h_G writes, for $X \in \mathbb{R}^{n\times r}$,

$$\begin{aligned} T_\lambda(X) &= \operatorname*{argmin}_{U\in\mathbb{R}^{n\times r}} \left\{\langle\nabla f(X), U - X\rangle + \frac{1}{\lambda}D_{h_G}(U, X)\right\} \\ &= \operatorname*{argmin}_{U\in\mathbb{R}^{n\times r}} \{h_G(U) - \langle V, U\rangle\} \end{aligned} \tag{2.8}$$

where we remove constant terms and defined $V := \nabla h_G(X) - \lambda\nabla f(X)$. Write for the sake of clarity $U^\star := T_\lambda(X)$. The optimization problem (2.8) is strictly convex and the unique solution $U^\star$ satisfies $\nabla h_G(U^\star) = V$, meaning that

$$U^\star\left(\alpha\|U^\star\|^2 I_r + \beta U^{\star T}U^\star + \sigma I_r\right) = V. \tag{2.9}$$

Define $Z := U^{\star T}U^\star \in \mathbb{R}^{r\times r}$. Then, the knowledge of Z determines $U^\star$, since $\|U^\star\|^2 = \mathbf{Tr}(Z)$ and therefore $U^\star = V(\alpha\,\mathbf{Tr}(Z)I_r + \beta Z + \sigma I_r)^{-1}$.

Now, taking (2.9) and multiplying by its transpose implies that

$$(\alpha\|U^\star\|^2 I_r + \beta Z + \sigma I_r)^2 Z = V^TV. \tag{2.10}$$

This shows that V^TV is a polynomial in Z, and therefore that they admit the same eigenvectors. Write the diagonalization

$$\begin{aligned} V^TV &= P^T\,\mathbf{diag}(\eta_1^2, \dots, \eta_r^2)P, \\ Z &= P^T\,\mathbf{diag}(\mu_1^2, \dots, \mu_r^2)P \end{aligned}$$

where $P \in \mathcal{O}_r$ and $\mu_i, \eta_i \geq 0$ for $i = 1 \dots r$. It follows from diagonalizing (2.10) and taking the square root that

$$\left(\alpha\left(\sum_{j=1}^r \mu_j^2\right) + \beta\mu_i^2 + \sigma\right)\mu_i = \eta_i \quad \forall i = 1, \dots, r$$

This is exactly the first-order optimality condition on $\mu = (\mu_1, \dots, \mu_r)$ for the problem

$$\mu = \operatorname*{argmin}_{x\in\mathbb{R}^r} \frac{\alpha}{4}\|x\|^4 + \frac{\beta}{4}\sum_{i=1}^r x_i^4 + \frac{\sigma}{2}\|x\|^2 - \sum_{i=1}^r \eta_i x_i. \tag{2.11}$$

Note that we do not need to enforce the nonnegativity constraint on x, since we chose $\eta_i \geq 0$ it follows that the optimal solution will be nonnegative. Hence, we can reconstruct Z from the diagonalization of $V^T V$ and the solution of Problem (2.11), and thus we get the procedure described in the theorem for computing $U^* = T_\lambda(X)$. ■

Complexity. Note that the order of multiplication is important: we only need to compute the eigendecomposition of $V^T V$, which is of size $r \times r$. We additionally need to solve a small minimization problem of size r, which can be done efficiently using the quartic Bregman gradient algorithm with norm kernel (see Appendix 2.5 for implementation details). Due to this, the complexity of computing the Bregman gradient map of h_G is $O(nr^2 + r^3 + Kr)$, where K is the number of iterations needed to solve the subproblem. Since r is usually much smaller than n by several orders of magnitude, the main computational bottleneck remains in most applications computing the gradient $\nabla f(X)$.

2.2.4 How to choose the most appropriate kernel?

In order to devise efficient methods, one should search for the kernel h such that the upper approximation of f in (RelSmooth) is *as tight as possible*, or, equivalently, such that the Hessian of the residual $Lh - f$ is small. On the other hand, h has to be simple enough so that the Bregman gradient map (2.3) is *easy to compute* (which precludes choosing $h = f$, as the iteration would be as hard to solve as the initial problem). This trade-off is key in choosing the appropriate kernel. Let us review these two conflicting criteria in our situation.

Complexity of the Bregman gradient map. For the norm kernel h_N, one iteration involves computing the gradient of f, then solving a simple scalar equation. The Gram kernel h_G involves solving a subproblem which requires $O(nr^2 + r^3)$ additional operations. This overhead is negligible for the typical regime where $r \ll n$; however, the iterate can be computed easily only for unconstrained problems.

Quality of Hessian approximation. We showed in Section 2.2.3 that the quality of the approximation of the quartic component of f by the Gram kernel is bounded by $O(1 - \mu_F / L_F)$. Therefore, it is expected to show good performance when F is sufficiently well-conditioned. The norm kernel, however, has no such property, as its approximation of f is much coarser. The difference stems from the supplementary $\|X^T X\|^2$ term, which can be much smaller than $\|X\|^4$, especially when the columns of X are nearly orthogonal.

Note that even if F is not globally strongly convex or μ_F is unknown, the Gram kernel can take advantage of local strong convexity through adaptive step sizes, as we show in the sequel.

2.3 Algorithms for quartic low-rank minimization

Now that we are equipped with a non-Euclidean geometry induced by one of the kernels h_N and h_G, we are ready to define the minimization scheme Dyn-NoLips in Algorithm 3. It extends the Bregman proximal gradient, a.k.a NoLips, algorithm from Bolte et al. [2018] to allow step sizes larger than the theoretical value $1/L$.

Algorithm 3 Dyn-NoLips

```
Input: A distance kernel $h$ such that $f$ is smooth relative to $h$ and a maximal step size $\lambda_{\max}$

Initialize $X^0 \in \mathbb{R}^{n\times r}$ such that $\Psi(X^0) < \infty$.
for k = 1,2,... do
    Choose a step size $\lambda_k \leq \lambda_{\max}$ such that the sufficient decrease condition (2.12) holds
    Set $X^k = T_{\lambda_k}(X^{k-1})$
end for
```

Step size choice The step size λ_k is chosen so that the new iterate $X^k = T_{\lambda_k}(X^{k-1})$ satisfies

$$f(X^k) \leq f(X^{k-1}) + \langle \nabla f(X^{k-1}), X^k - X^{k-1}\rangle + \frac{1}{\lambda_k} D_h(X^k, X^{k-1}). \tag{2.12}$$

There are two ways to ensure this condition holds.

- **Fixed step size.** Since f is L-smooth relative to h, (2.12) holds as soon as $0 < \lambda_k \leq 1/L$.
- **Dynamical step size.** In some cases, the relative Lipschitz constant might be too conservative, and better numerical performance can be achieved by taking larger steps. We therefore can use a dynamical strategy for extending the step size, ensuring that (2.12) holds at each iteration. There are many strategies to efficiently adjust the step size; see, e.g., Nesterov [2007]. In our case, we choose a simple strategy similar in spirit to the Armijo line search: at iteration k, start with a tentative step size λ_k, then find the smallest nonnegative integer j such that (2.12) is satisfied with step size $2^{-j}\lambda_k$. Then, set $\lambda_{k+1} = 2^{-j+1}\lambda_k$.

Convergence to a stationary point. We now extend the theoretical convergence results from Bolte et al. [2018] to handle the dynamical step size strategy.

Theorem 2.7 (Convergence results)**.** *Let $\{X^k\}_{k\geq 0}$ be the sequence generated by Algorithm 3. Assume that*

1. *f is L-smooth relative to a distance kernel h such that h is strongly convex and twice continuously differentiable on $\mathbb{R}^{n\times r}$, and the penalty function g is convex,*
2. *$\lambda_{\max} \geq 1/(2L)$,*
3. *the function $\Psi = f + g$ is coercive (meaning that $\Psi(X) \to +\infty$ when $\|X\| \to +\infty$) and semialgebraic.*

Then, the sequence $\{\Psi(X^k)\}_{k\geq 0}$ is nonincreasing, and the sequence $\{X^k\}_{k\geq 0}$ converges towards a critical point X^* of problem Ψ.

Proof. First, the step size λ_k can be bounded for $k \geq 0$ as

$$\frac{1}{2L} \leq \lambda_k \leq \lambda_{\max}.$$

Indeed, the upper bound holds by construction of the algorithm. The lower bound comes from the relative smoothness property: condition (2.12) is true for every $\lambda \in (0, \frac{1}{L}]$, so the inner loop will stop whenever λ gets below $1/L$.

Let us now prove the result. Since Condition (2.12) holds at each iteration k, we can write

$$f(X^{k+1}) \leq f(X^k) + \langle \nabla f(X^k), X^{k+1} - X^k \rangle + \frac{1}{\lambda_k} D_h(X^{k+1}, X^k). \tag{2.13}$$

On the other hand, the optimality condition characterizing $X^{k+1} = T_{\lambda_k}(X^k)$ writes

$$0 \in \lambda_k \left(\partial g(X^{k+1}) + \nabla f(X^k) \right) + \nabla h(X^{k+1}) - \nabla h(X^k), \tag{2.14}$$

where ∂g denotes the subdifferential of the convex function g. Combining (2.14) with the subgradient inequality for g yields

$$\begin{aligned} g(X^{k+1}) \leq g(X^k) &+ \frac{1}{\lambda_k} \langle \nabla h(X^k) - \nabla h(X^{k+1}), X^{k+1} - X^k \rangle \\ &- \langle \nabla f(X^k), X^{k+1} - X^k \rangle. \end{aligned} \tag{2.15}$$

Summing (2.13) and (2.15) gives

$$\Psi(X^{k+1}) \leq \Psi(X^k) + \frac{1}{\lambda_k} \left[D_h(X^{k+1}, X^k) + \langle \nabla h(X^k) - \nabla h(X^{k+1}), X^{k+1} - X^k \rangle \right],$$

which yields

$$\Psi(X^{k+1}) \leq \Psi(X^k) - \frac{1}{\lambda_k} D_h(X^k, X^{k+1}).$$

From this inequality, we can now prove the same convergence properties as for the standard BPG scheme. Indeed, the monotonicity of the sequence $\{\Psi(X^k)\}_{k \geq 0}$ is a direct consequence of the above. Since $\lambda_k \leq \lambda_{\max}$, it follows that at every iteration $k \geq 0$,

$$\Psi(X^k) - \Psi(X^{k+1}) \geq \frac{1}{\lambda_{\max}} D_h(X^k, X^{k+1}).$$

Now, this inequality is the same as the one needed to prove convergence in the case of the fixed step size in Bolte et al. [2018]. Thus, global convergence towards a critical point is a consequence of [Bolte et al., 2018, Th. 4.1], since all the assumptions are met: the kernel h is defined over the entire space $\mathbb{R}^{n \times r}$, it is strongly convex, and ∇h is Lipschitz continuous on bounded subsets of $\mathbb{R}^{n \times r}$ (because we assumed it is C^2). We also need the fact that the sequence $\{X^k\}_{k \geq 0}$ is bounded, which is a consequence of the monotonicity of $\{\Psi(X^k)\}_{k \geq 0}$ and the fact that the function Ψ is coercive. ■

The semialgebraicity assumption is needed to establish the crucial nonsmooth Lojasiewicz property Bolte et al. [2007], required to show convergence to a critical point. It holds for all the applications we cited, since the class of semialgebraic functions includes polynomial functions, ℓ_1 and ℓ_2 norms, the ℓ_0 seminorm and indicators of polynomial sets.

2.4 Applications

We now illustrate applications of our methodology to two different low-rank problems, symmetric nonnegative matrix factorization and Euclidean distance matrix completion. We show that good numerical performance can be reached using the dynamical step strategy, and that, for Euclidean matrix completion, it can be further improved by using the Gram kernel.

2.4.1 Symmetric Nonnegative Matrix Factorization

Symmetric Nonnegative Matrix Factorization (SymNMF) is the task of finding, given a symmetric nonnegative matrix $M \in \mathbb{R}^{n\times n}$, a nonnegative matrix $X \in \mathbb{R}^{n\times r}$ such that $M \approx XX^T$. This is done by solving

$$\min \frac{1}{2}\|M - XX^T\|_F^2 \quad \text{subject to } X \geq 0 \tag{SymNMF}$$

in the variable $X \in \mathbb{R}^{n\times r}$, where the inequality constraint is meant componentwise and $r \leq n$ is the target rank.

(SymNMF) is used as a probabilistic clustering or graph clustering technique [Ding et al., 2005, He et al., 2011]. Numerical experiments by Kuang et al. [2015] have shown that it achieves state-of-the-art clustering accuracy on several text and image datasets.

Solving SymNMF

While (SymNMF) looks similar to the well-known asymmetric NMF problem

$$\min_{X,Y} \frac{1}{2}\|M - XY^T\|,$$

it is actually harder. This is because NMF has a favorable block structure that allows the application of efficient alternating algorithms [Kim and Park, 2013, Cichocki and Phan, 2009]. SymNMF, however, does not enjoy the same block structure. Current solvers fall into two categories:

Direct solvers. There have been several attempts at solving the original problem, including multiplicative update rules [He et al., 2011], projected gradient algorithm quasi-Newton schemes [Kuang et al., 2015], and coordinate descent [Vandaele et al., 2016].

Nonsymmetric relaxations. Another idea is to use a mere penalty method [Kuang et al., 2015, Lu et al., 2017, Zhu et al., 2018], relaxing (SymNMF) to the following penalized nonsymmetric problem

$$\begin{array}{ll} \min & \frac{1}{2}\|M - XY^T\|_F^2 + \mu\|X - Y\|_F^2 \\ \text{subject to} & X, Y \geq 0, \end{array} \tag{P-NMF}$$

in the variables $X, Y \in \mathbb{R}^{n\times r}$, with parameter $\mu \geq 0$. This formulation is very similar to asymmetric NMF and can be solved by the same fast alternating algorithms that exploit the block structure, such as Alternating Nonnegative Least Squares (ANLS) and Hierarchical Alternating Least Squares [Zhu et al., 2018] (HALS), which are arguably the fastest SymNMF solvers.

Applying NoLips. We propose to apply the NoLips/Bregman proximal gradient algorithm for optimizing the original objective function. Problem (SymNMF) falls within our framework with $F(Y) = \frac{1}{2}\|M - Y\|^2$, which has a Lipschitz gradient with constant 1, and $g(X) = i\{X \geq 0\}$ the indicator function of the nonnegative orthant. Therefore, Proposition 2.3 implies that $f(X) := \frac{1}{2}\|M - XX^T\|^2$ is 1-smooth relatively to the kernel h_N with $\alpha = 6$ and $\sigma = 2\|\nabla F(0)\| = 2\|M\|$. Since, in addition, f is polynomial and g is the indicator of a polynomial set, $f + g$ is semialgebraic, and it is also coercive, so Theorem 2.7 guarantees that NoLips will converge towards a stationary point of problem (SymNMF).

In this problem, the Bregman iteration map is solved by simply adding a projection step

$$T_\lambda(X) = \frac{1}{\tau_\sigma(\alpha\|\Pi_+(U)\|^2)}\Pi_+(U),$$

where $U = \nabla h_N(X) - \lambda\nabla f(X)$, τ_σ has been defined in Proposition 2.4 and Π_+ is the projection on the nonnegative orthant: $\Pi_+(U) = \max(U, 0)$ (entrywise).

Computational complexity for NoLips. The computational complexity of an iteration is dominated by gradient computations and objective function evaluations, as all other operations are linear in the size of the variable.

If M is a $n \times n$ **dense** matrix, each gradient and function evaluation uses $O(n^2r + nr^2)$ floating point operations. If M is represented as a **sparse** matrix with $p << n^2$ nonzero elements, then we can take advantage of this structure [Vandaele et al., 2016, Rmk. 2] by using

$$\begin{aligned} f(X) &= \frac{1}{2}\|XX^T - M\|^2 = \frac{1}{2}\|M\|^2 + \frac{1}{2}\|X^TX\|^2 - \langle MX, X\rangle \\ \nabla f(X) &= 2X(X^TX) - 2MX \end{aligned}$$

which yields a much improved $O\left((r^2 + p)n\right)$ complexity per iteration.

Numerical experiments

We implemented the following algorithms: Algorithm 3 with dynamical step size and the norm kernel (`Dyn-NoLips`), the β-`SNMF` scheme from He et al. [2011], where we set $\beta = 0.99$ as advised by the authors, the projected gradient algorithm (`PG`) with Armijo line search from Kuang et al. [2015], where we use the line search parameters $\beta = 0.1$ and $\sigma = 0.01$, the coordinate descent scheme (`CD`) from Vandaele et al. [2016], the `ADMM` algorithm [Lu et al., 2017], and the two fast algorithms from Zhu et al. [2018] for solving the penalized problem (P-NMF): `SymANLS` and `SymHALS`. For the last two, we tuned the μ penalization parameter for best performance. We left out the quasi-Newton algorithm from Kuang et al. [2015] because of its prohibitive $O(n^3)$ complexity for large datasets.

All algorithms were implemented in Julia [Jeff Bezanson, Alan Edelman and Shah, 2017] which is a highly-optimized numerical computing language. Since our algorithms have different complexity per iteration, it is essential to compare them in terms of running time, and Julia provides a fairly accurate way to do so as there is little interpreter overhead in loops. Tests were run on a PC Intel CORE i7-4910MQ CPU @ 2.90 GHz x 8 with 32 Go RAM.

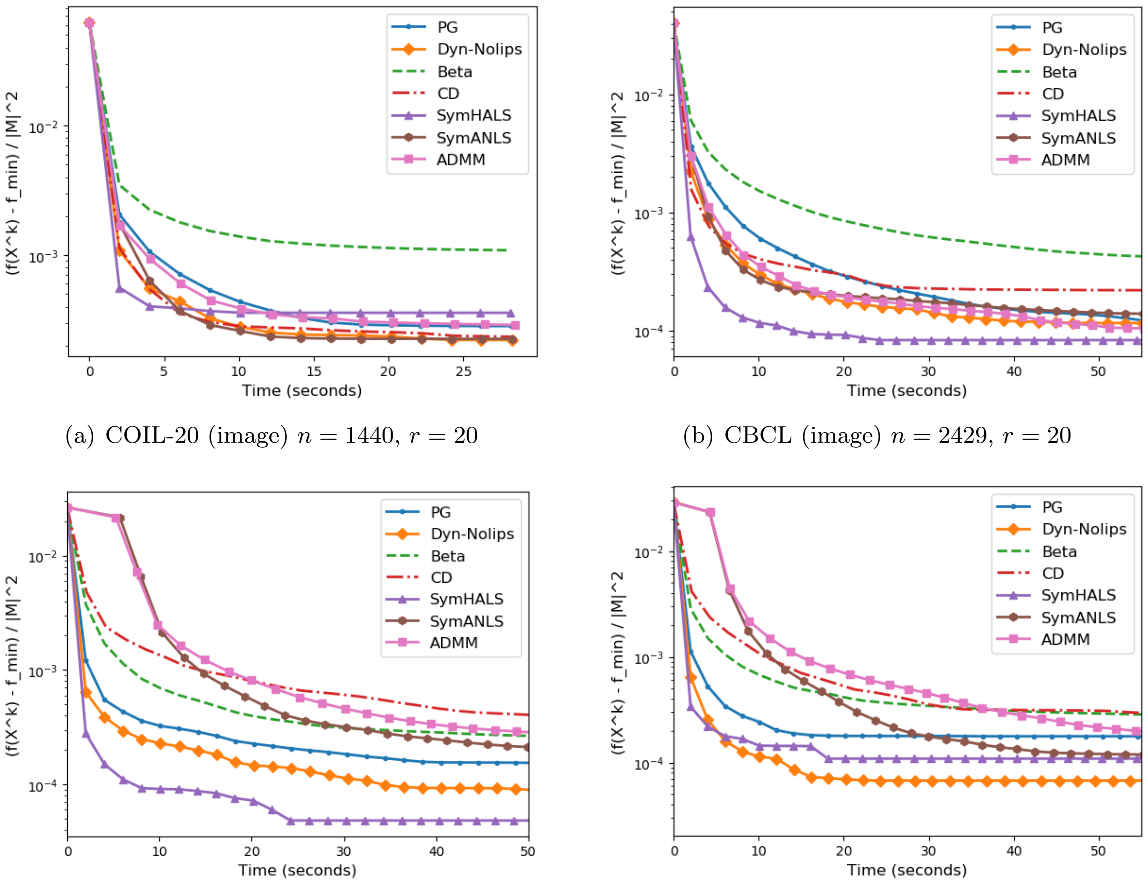
We used two image and two text datasets.

- **Image.**
  - **CBCL**[1]: 2,429 images of faces of size 19 × 19
  - **Coil-20**[2]: 1440 images of size 128×128 representing 20 objects under various angles.
- **Text.**
  - **TDT2**[3]: dataset of 11,201 news articles classified in 96 semantic categories.

[1] http://cbcl.mit.edu/software-datasets/FaceData2.html

[2] http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php

[3] http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html



(a) COIL-20 (image) $n = 1440$, $r = 20$

(b) CBCL (image) $n = 2429$, $r = 20$

(c) TDT2 (text) $n = 9394$, $r = 30$

(d) Reuters (text) $n = 7963$, $r = 30$

Figure 2.1: SymNMF normalized objective gap $\left(f(X^k) - f_{\min}\right)/\|M\|^2$ averaged over 10 random initializations, for various sparse similarity matrices $M \in \mathbb{R}^{n\times n}$. Hyperparameters for SymHALS, SymANLS were tuned for best performance, while Dyn-NoLips is parameter-free.

– **Reuters**[3]: dataset of news articles, which we restricted to the largest 25 categories, leaving a total of 7,963 documents.

For all image and text datasets, we construct a sparse similarity matrix M following the procedure described in [Kuang et al., 2015, Section 7.1]. We begin by computing the similarity graph between data points, using cosine similarity on term frequency vectors for text, and a Gaussian kernel for image (with the self-tuning method for the scale). The graph obtained is *sparsified* by keeping only the edges connecting the k-nearest neighbors, with $k = \lfloor \log_2 n \rfloor + 1$. Then, M is taken as a normalized version of the graph adjacency matrix.

We use the usual convergence criterion for constrained nonconvex problems

$$\frac{\|\nabla^P f(X^k)\|}{\|\nabla^P f(X^0)\|} \leq \epsilon \tag{2.16}$$

Table 2.1: CPU time (in seconds) needed to reach a decrease of $\epsilon = 10^{-3}$ in projected gradient norm (see (2.16) for definition). Results have been averaged over 10 random initializations. Hyperparameters for SymHALS, SymANLS and ADMM have been tuned for best performance. Missing values indicate failure of convergence.

Dataset	r	NoLips	PG	Beta	CD	SymHALS	SymANLS	ADMM
Coil-20	10	24.7	51.4	-	26.2	7.0	32.3	-
	20	23.7	36.8	-	21.3	4.0	18.2	-
	30	20.7	40.8	-	35.4	6.5	20.2	-
	40	21.7	49.5	-	57.6	7.5	28.4	-
CBCL	10	38.2	42.7	44.0	35.6	13.6	35.2	42.8
	20	57.7	88.4	-	93.9	17.8	47.8	-
	30	60.9	134.3	-	135.0	15.1	43.4	-
	40	50.8	126.4	-	90.0	23.7	52.5	-
TDT2	10	35.2	54.2	-	97.5	11.0	-	-
	20	52.4	76.1	-	109.9	20.1	-	-
	30	29.4	45.1	-	-	12.1	-	-
	40	28.0	49.8	-	-	17.7	-	-
Reuters	10	6.5	10.0	-	33.0	3.0	54.2	-
	20	28.7	32.8	-	71.7	9.5	74.7	-
	30	24.3	45.5	-	69.4	6.5	91.0	-
	40	40.2	68.5	-	83.2	10.6	108.3	-

where $\nabla^P f(X)$ is the projected gradient defined as

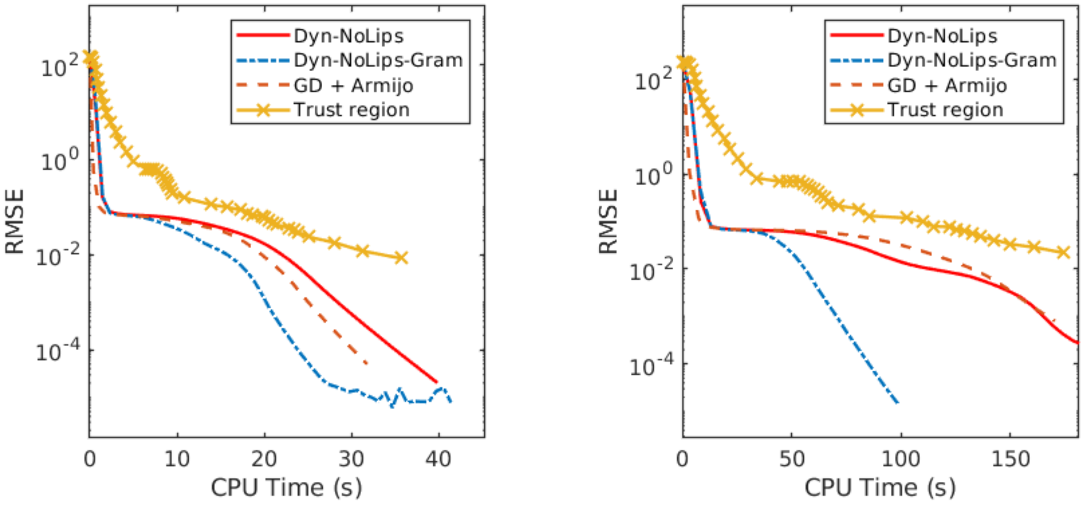
$$(\nabla^P f(X))_{ij} = \begin{cases} \nabla f(X)_{ij} & \text{if } X_{ij} > 0, \\ \min\left(\nabla f(X)_{ij}, 0\right) & \text{if } X_{ij} = 0. \end{cases}$$

Table 2.1 reports the average time needed to reach a convergence criterion of $\epsilon = 10^{-3}$, for 10 random initializations. For each dataset, we test several values for the rank parameter r. In addition, Figure 1 shows the average evolution of the normalized objective gap $\left(f(X^k) - f_{min}\right)/\|M\|^2$, where f_{min} is the minimal objective value encountered in all initializations.

Overall, the algorithm that shows the best convergence speed is `SymHALS`, but it has the disadvantage of needing to tune the penalization parameter μ. In the experiments we report, small values of μ yielded optimal performance, while the convergence theory of Zhu et al. [2018] only holds for large values for which the algorithm is much slower. By contrast, `Dyn-NoLips` is hyperparameter-free and has the second best overall performance. The gap with the other methods is particularly significant on the larger `TDT2` and `Reuters` datasets, showing that the method scales well with problem dimension.

2.4.2 Euclidean Distance Matrix Completion

Euclidean distance matrix completion (EDMC) is the task of recovering the position of n points $x_1^*, \dots, x_n^* \in \mathbb{R}^r$, given the knowledge of a partial set of pairwise distances $d_{ij} = \|x_i^* - x_j^*\|^2$ for $(i, j) \in \Omega$, where $\Omega \subset [1, n] \times [1, n]$. It is a fundamental problem with applications in



(a) $n = 2000, r = 3$

(b) $n = 5000, r = 3$

Figure 2.2: Euclidean matrix completion problems on the Helix dataset, with 10% known distances and two different problem sizes. We present the normalized RMSE over the full distance matrix versus CPU time. The results are averaged over 10 random initializations.

sensor network localization and the study of conformation of molecules; see Fang and O'Leary [2012], Qi and Yuan [2013], Dokmanic et al. [2015] and references therein. The Burer-Monteiro nonconvex formulation for solving this problem writes

$$\min f(X) := \frac{1}{2} \sum_{(i,j)\in\Omega} \left(\|X_i - X_j\|^2 - d_{ij}\right)^2 \tag{EDMC}$$

in the variable $X \in \mathbb{R}^{n\times r}$. It can be rewritten $f(X) = \frac{1}{2}\|\mathcal{P}_\Omega(\kappa(XX^T) - D)\|^2$ where D is the matrix of known distances, $\mathcal{P}_\Omega$ denotes the projection operator such that $\mathcal{P}_\Omega(Y)_{ij} = Y_{ij}$ if $(i,j) \in \Omega$, and $\mathcal{P}_\Omega(Y)_{ij} = 0$ elsewhere, and κ is the linear operator defined for $Y \in \mathbb{R}^{n\times n}$ by

$$\kappa(Y)_{ij} = Y_{ii} + Y_{jj} - 2Y_{ij} \text{ for } 1 \leq i,j \leq n.$$

Applying NoLips with the norm kernel. Problem (EDMC) falls within our framework with $F(Y) = \frac{1}{2}\|\mathcal{P}_\Omega(\kappa(Y) - D)\|^2$, which can be shown to have a Lipschitz gradient with constant $L_{EDM} := 9 \max_{i=1\ldots n} |\{j|(i,j) \in \Omega\}|$. Therefore, as in the case of SymNMF, the norm kernel h_N can be used with a initial step size 1 and parameters $\alpha = 6L_{EDM}$ and $\sigma = \frac{1}{3}\|\nabla F(0)\| = 2\|\mathcal{P}_\Omega(D)\|$.

Using the Gram kernel As the problem is unconstrained, we can also apply minimization using the Gram kernel h_G. We use the parameters $\alpha = 2L_{EDM}$, $\beta = L_{EDM}$ and $\sigma = 2\|\mathcal{P}_\Omega(D)\|$, which ensure that f is 1-smooth relatively to h_G by Proposition 2.5.

Computational complexity for NoLips. As before, the main computational bottleneck for an iteration consists in computing the value and gradient of the objective function. If

$p = |\Omega|$ denotes the number of known distances, then the computational complexity is $O(pr)$. If the Gram kernel is used, each iteration requires an additional $O(nr^2 + r^3)$ flops (See Section 2.2.3), which is negligible compared to the latter in the usual setting where $p >> n$ and r is small.

Numerical experiments We implement the following algorithms: NoLips with a dynamical step size and the norm kernel (`Dyn-NoLips`), NoLips with a dynamical step size and the Gram kernel (`Dyn-NoLips-Gram`), gradient descent with Armijo line search (`GD`), the Riemannian trust region algorithm from Mishra et al. [2011] (`TR`). We leave out semidefinite relaxations because of their memory requirement which is prohibitive on large data. As the implementation for `TR` is provided in Matlab, we run our experiments on Matlab as well, with the same setup as in Section 2.4.1.

We try the algorithms on a standard EDMC problem, the 3-dimensional `Helix` dataset [Mishra et al., 2011] which is generated as $X_i = (\cos(3t_i), \sin(3t_i), 2t_i)$ where $\{t_i\}_{i=1}^n$ are sampled uniformly in $[0, 2\pi]$. We randomly keep only 10 % on the pairwise distances, and test on two different problem sizes: $n = 2000$ and $n = 5000$. Figure 2.2 reports the normalized root mean squared error (RMSE) over *all distances* (known and unknown) averaged on 10 random initializations. All the algorithms manage to recover the ground truth; the `Dyn-NoLips-Gram` algorithm shows the best numerical performance, which demonstrates the advantage of using the Gram geometry.

2.5 Conclusion

We proposed a generic approach for solving Burer-Monteiro formulations of low rank minimization problems using the methodology of Bregman gradient methods and relative smoothness. We studied two quartic kernels, including a new Gram kernel, and demonstrated their benefits on numerical experiments. In future work, performance could be improved further by studying inertial variants [Mukkamala et al., 2020, Hanzely et al., 2021]. New kernels could also be explored beyond the class of quartic functions to tackle other problems with inherent non-Euclidean geometries.

Code

The code for reproducing experiments for SymNMF and Euclidean Distance Matrix Completion can be downloaded from the public repository
https://github.com/RaduAlexandruDragomir/QuarticLowRankOptimization

Appendix: solving the subproblem for computing the Bregman gradient map of the Gram kernel

While it seems that computing the Bregman iteration map of the Gram kernel involves solving another difficult quartic subproblem, it is actually of small size (r is typically not larger than a few dozens) and can be solved efficiently with the NoLips scheme.

Indeed, the objective function ϕ of problem (2.11) is 1-smooth relatively to the norm kernel in $\mathbb{R}^r$ $h_N(x) = \frac{\alpha_u}{4}\|x\|^4 + \frac{\sigma_u}{2}\|x\|^2$ with a choice of parameters $\alpha_u = \alpha + 3\beta$ and $\sigma_u = \sigma$.

Algorithm 4 details the procedure. We initialize μ with the values for the previous iteration of the outer procedure. This proves to be efficient as the values will not vary much from one iteration to another. For the stopping criterion, we use the scaled gradient norm $\|\nabla\phi(v)\|/\|\eta\|$ and a tolerance value $\epsilon = 10^{-6}$.

The subproblem being very well conditionned, it is minimized easily; in numerical experiments, it usually convergences in no more than 20 iterations.

Algorithm 4 Computing the Bregman iteration map of the Gram kernel

```
**Input:** Matrix $X \in \mathbb{R}^{n \times r}$, gradient of the objective $\nabla f(X)$, step size $\lambda > 0$, parameters $\alpha, \beta, \sigma > 0$, subproblem tolerance $\epsilon$, and (optionally), values $\mu^-$ of $\mu$ computed at the previous iteration.
Form $V = \nabla h_G(X) - \lambda \nabla f(X) = \left(\alpha\|X\|^2 I_r + \beta X^T X + \sigma I_r\right) X - \lambda \nabla f(X)$
Compute $V^T V$
Form the eigendecomposition of $V^T V = P^T D P$ where $P \in \mathcal{O}_r$ and $D = \mathbf{diag}(\eta_1^2, \dots, \eta_r^2)$
Initialize $\mu$ as $\mu^-$ if provided, and as $(0, \dots, 0)$ otherwise.
**repeat**
  Compute $\nabla\phi(\mu)$ where $\nabla\phi(\mu)_i = \alpha\|\mu\|^2\mu_i + \beta\mu_i^3 + \sigma\mu_i - \eta_i$
  Compute $\nabla h_N(\mu)$ where $\nabla h_N(\mu)_i = (\alpha + 3\beta)\|\mu\|^2\mu_i + \sigma\mu_i$
  Form $v = \nabla h_N(\mu) - \nabla\phi(u)$
  Set $\mu \leftarrow \left[\tau_\sigma\left((\alpha + 3\beta)\|v\|^2\right)\right]^{-1} v$ where $\tau_\sigma$ has been defined in Proposition 2.4
**until** stopping criterion has been satisfied, i.e., $\|\nabla\phi(v)\|/\|\eta\| < \epsilon$
Form $Z = P^T\, \mathbf{diag}(\mu_1^2, \dots, \mu_r^2) P$
Compute $T_\lambda(X) = V\left[\alpha\, \mathbf{Tr}(Z) I_r + \beta Z + \sigma I_r\right]^{-1}$
**Output**: Bregman gradient iterate $T_\lambda(X)$
```

Chapter 3

A Lower Bound for Relatively-Smooth Convex Mimization

Chapter Abstract

We prove a lower bound showing that the $O(1/k)$ convergence rate of the Bregman gradient descent method (a.k.a. NoLips or mirror descent) is optimal for the class of problems satisfying the relative smoothness assumption, among all first-order methods that use Bregman divergences and linear operations. As a consequence, no algorithm of this class can achieve a better rate than Bregman gradient descent for *generic* kernel functions.

Reference: this chapter is based on a publication in Mathematical Programming [Dragomir et al., 2021c]. Part of this work has been done in collaboration with Adrien Taylor.

3.1 Introduction

We consider the constrained minimization problem

$$\min_{x\in\mathcal{C}} f(x) \tag{P}$$

where f is a convex continuously differentiable function and $\mathcal{C}$ is a nonempty closed convex subset of $\mathbb{R}^d$. In large-scale settings, first-order methods are particularly popular due to their simplicity and their low cost per iteration.

The (projected) gradient descent (PG) is a classical method for solving (P), and consists in successively minimizing quadratic approximations of f, with

$$x_{k+1} = \operatorname*{argmin}_{u\in\mathcal{C}} f(x_k) + \langle \nabla f(x_k), u - x_k\rangle + \frac{1}{2\lambda}\|u - x_k\|^2, \tag{PG}$$

where $\|\cdot\|$ is the Euclidean norm. Although standard, there is often no good reason for making such approximations, beyond our capability of solving this intermediate optimization problem. In other words, this traditional approximation typically does not reflect neither the geometry of f nor that of C. A powerful generalization of PG consists in performing instead a *Bregman gradient step*

$$x_{k+1} = \operatorname*{argmin}_{u \in \mathcal{C}} f(x_k) + \langle \nabla f(x_k), u - x_k \rangle + \frac{1}{\lambda} D_h(u, x_k), \tag{BGD}$$

where the Euclidean distance has been replaced by the *Bregman divergence*

$$D_h(x, y) := h(x) - h(y) - \langle \nabla h(y), x - y \rangle$$

induced by some strictly convex and continuously differentiable *kernel function* h. A well-chosen h allows designing first-order algorithms adapted to the geometry of the constraint set and/or the objective function. Of course, a conflicting goal is to choose h such that each iteration (BGD) can be solved efficiently in practice, discarding choices such as $h = f$ (for which performing an iteration would be as hard as solving the original problem).

Recently, Bauschke et al. [2017] introduced a natural condition for analyzing this scheme, assuming that the inner objective in the iteration (BGD) is an upper bound on f. This ensures that performing an iteration decreases the function values $f(x_k)$. This assumption, known as relative smoothness, generalizes the standard L-smoothness assumption implied by Lipschitz continuity of ∇f. The Bregman gradient algorithm, also called NoLips in the setting of Bauschke et al. [2017], is thus a natural extension of gradient descent (PG) to objective functions whose geometry is better modeled by a non-quadratic kernel h. Practical examples of relative smoothness arise in Poisson inverse problems [Bauschke et al., 2017], quadratic inverse problems [Bolte et al., 2018], rank minimization [Dragomir et al., 2021a] and regularized higher-order tensor methods [Nesterov, 2021].

Can we accelerate Bregman gradient descent? In the Euclidean setting where h is the squared Euclidean norm $\frac{1}{2}\|\cdot\|^2$, accelerated projected gradient methods exhibit faster convergence than the vanilla projected gradient algorithm. These methods, which can be traced back to Nesterov [1983], are proven to be *optimal* for L-smooth functions and have found a number of successful applications, in e.g., imaging [Beck and Teboulle, 2009]. A natural question is therefore to understand whether the (BGD) algorithm can be accelerated in the relatively-smooth setting.

This question has been raised in several works, including [Bauschke et al., 2017, Section 6], [Lu et al., 2018, Section 3.4], and the survey [Teboulle, 2018, Section 6]. Partial answers have already been provided under somewhat strict additional regularity assumptions (see e.g., Auslender and Teboulle [2006], Walid et al. [2015], Hanzely et al. [2021] and discussions in the sequel), while the general case was apparently still open, and relevant in practical applications.

In this work, we establish a lower complexity bound proving that BGD is *optimal* for the general relatively-smooth setting, and therefore that generic acceleration is impossible.

In order to do so, we adopt the standard *black-box model* used for studying complexity of first-order methods [Nemirovski and Yudin, 1983]. We consider that both f and h are described by first-order oracles, so as to obtain generic complexity results, and we look for worst-case *couples* of functions (f, h) satisfying the relative smoothness assumption. A central idea in our approach is the fact that, when studying the worst-case behavior of Bregman methods in the relatively-smooth setting, f and h can get arbitrarily close to some *limiting pathological nonsmooth functions*.

Main inspiration: Performance Estimation Problems. The worst-case instance used for proving our lower bound was inferred from the solution to a Performance Estimation Problem (PEP). The PEP methodology allows to study worst-case behavior of first-order methods by solving appropriate semidefinite programs, and was pioneered by Drori and Teboulle [2014] in the context of smooth convex minimization. In Chapter 4, we show how the PEP technique can be extended to Bregman first-order methods in the relatively-smooth setting, and how it allowed us to discover the key elements for proving our lower bound. However, we emphasize that the proof we give in this chapter is self-contained and can be read independently.

Related work on lower complexity bounds. The first-order black-box model, developed initially in the works of Nemirovski and Yudin [1983] and later Nesterov [2003] has allowed to prove optimal complexity results for several classes of problems in first-order optimization [Drori, 2017]. These results usually rely on well-chosen *worst-case functions* whose structure makes them difficult to minimize for all methods within a given class.

Our worst-case instances are obtained from pointwise maxima of affine functions, reminiscent of lower bounds for nonsmooth convex minimization [Nemirovski and Yudin, 1983, Woodworth and Srebro, 2017]. Our construction then involves smoothing those piecewise affine functions, making them differentiable. This technique is also used in the very related work of Guzmán and Nemirovski [2015], which studies lower bounds for minimization of convex functions that are smooth with respect to ℓ_p norms. To the best of our knowledge, the lower bound obtained in the sequel is not a particular case of those in Guzmán and Nemirovski [2015], as smoothness with respect to a certain norm is different from relative smoothness with respect to the same (squared) norm, beyond the ℓ_2-norm.

Notation, We use $\overline{\mathcal{C}}$ to denote the closure of a set $\mathcal{C}$, $\operatorname{int}\mathcal{C}$ for its interior and $\partial\mathcal{C}$ for its boundary. We denote $(e_1, \ldots, e_n)$ the canonical basis of $\mathbb{R}^d$, and for $p \in \{1, \ldots n\}$ we write $E_p = \operatorname{Span}(e_1, \ldots, e_p)$ the set of vectors supported by the first p coordinates. Subscripts on a vector denote the iteration counter, while a superscript such as $x^{(i)}$ denotes the i-th coordinate.

3.2 Algorithmic setup

In this section, we briefly recall the base ingredients and technical assumptions on f and h that are used within Bregman first-order methods. For a more detailed presentation and examples of applications, we refer the reader to Chapter 1.

3.2.1 Kernel functions

Let $\mathcal{C}$ be a nonempty closed convex subset of $\mathbb{R}^d$. The first step in defining Bregman methods is the choice of a *kernel* (or reference) function h on $\mathcal{C}$.

Definition 3.1 (Kernel function)**.** *A function $h : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is called a kernel function on $\mathcal{C}$ if*

(i) h is closed convex proper (c.c.p.),

(ii) h is continuously differentiable and strictly convex on $\operatorname{int}\mathcal{C}$,

(iii) the Bregman gradient iterates are well-posed, i.e., for every $p \in \mathbb{R}^d$, the problem

$$\min_{u \in C} \langle p, u\rangle + h(u)$$

has a unique minimizer, which belongs to $\operatorname{int} \mathcal{C}$*.*

A kernel function h induces a *Bregman divergence* D_h defined as

$$D_h(x, y) = h(x) - h(y) - \langle \nabla h(y), x - y\rangle \quad \forall x \in \operatorname{dom} h, y \in \operatorname{dom} \nabla h.$$

Note that D_h is not a distance in the classical sense, however it enjoys a separation property; due to the strict convexity of h we have $D_h(x, y) \geq 0\ \forall x \in \operatorname{dom} h, y \in \operatorname{dom} \nabla h$, and $D_h(x, y) = 0$ iff $x = y$.

Examples. We list some of the most classical examples of kernel functions (see Chapter 1 for more):

- the **Euclidean kernel** $h(x) = \frac{1}{2}\|x\|^2$ with domain $\mathbb{R}^d$, and for which $D_h(x, y) = \frac{1}{2}\|x-y\|^2$ is the Euclidean distance,
- the **Boltzmann-Shannon entropy** $h(x) = \sum_i x^{(i)} \log x^{(i)}$ extended to 0 by setting $0 \log 0 = 0$, whose domain is thus $\mathbb{R}^d_+$,
- the **Burg entropy** $h(x) = \sum_i - \log x^{(i)}$ with domain $\mathbb{R}^d_{++}$,
- the **quartic kernel** $h(x) = \frac{1}{4}\|x\|^4 + \frac{1}{2}\|x\|^2$ with domain $\mathbb{R}^d$ [Bolte et al., 2018].

Convex conjugate. If h is a kernel function, we define its convex conjugate h^* as

$$h^*(y) = \sup_{u \in \mathbb{R}^d} \langle u, y\rangle - h(u)$$

If, for every $y \in \mathbb{R}^d$, the supremum in the definition of $h^*(y)$ is attained, then h^* is differentiable and its gradient satisfies for every $u \in \operatorname{dom} \nabla h^*$

$$\nabla h^*(y) = \operatorname*{argmax}_{u \in \mathbb{R}^d} \langle u, y\rangle - h(u).$$

3.2.2 Relatively-smooth optimization problems

We now recall the framework of relatively-smooth optimization [Bauschke et al., 2017, Lu et al., 2018] for solving the minimization problem

$$\min_{x \in \mathcal{C}} f(x)$$

For simplicity, we present the setting without nonsmooth regularization term; our lower bound is a fortiori valid for the Bregman *proximal* gradient algorithm designed for solving composite problems [Bauschke et al., 2017, Eq. (12)].

In addition to these assumptions, the central property we need in order to apply the Bregman gradient method is the so-called relative smoothness property [Bauschke et al., 2017, Lu et al., 2018].

Definition 3.2 (Relative smoothness)**.** *Let h be a kernel function on $\mathcal{C}$, and f a function such that* $\operatorname{dom} h \subset \operatorname{dom} f$*. We say that f is smooth relative to h if it is differentiable on* $\operatorname{int}\mathcal{C}$ *and if there exists a constant $L > 0$ such that*

$$Lh - f \quad \text{is convex on} \operatorname{dom} h. \tag{LC}$$

Relative smoothness allows to build a simple global majorant of f; indeed, (LC) is equivalent to the condition (see, e.g, Bauschke et al. [2017])

$$f(x) \leq f(y) + \langle \nabla f(y), x - y \rangle + LD_h(x, y) \quad \forall x \in \operatorname{dom} h, y \in \operatorname{dom} \nabla h,$$

and the NoLips/Bregman gradient method consists in successively minimizing this upper approximation.

We use the following convenient notation to characterize the class of relatively-smooth problems.

Definition 3.3. *We say that the couple of functions (f, h) is a relatively-smooth instance, and write $(f, h) \in \mathcal{B}_L(\mathcal{C})$ if*

(i) h is a kernel function on $\mathcal{C}$,

(ii) $f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is a closed convex proper function,

(iii) f is L-smooth relative to h on $\mathcal{C}$.

Finally, let us denote by $\mathcal{B}_L$ the union of $\mathcal{B}_L(\mathcal{C})$ for all closed convex sets $\mathcal{C}$:

$$\mathcal{B}_L = \bigcup_{d \geq 1} \bigcup_{\substack{\mathcal{C} \subset \mathbb{R}^d \\ \mathcal{C} \text{ closed convex}}} \mathcal{B}_L(\mathcal{C})$$

3.2.3 The NoLips/Bregman Gradient algorithm

Previous assumptions allow defining the Bregman Gradient (BG)/NoLips algorithm for minimizing f. For simplicity, we consider here the constant step size method.

Algorithm 5 Bregman Gradient (BG) / NoLips [Bauschke et al., 2017]

```
  Input: $(f, h) \in \mathcal{B}_L(\mathcal{C})$, $x_0 \in \operatorname{int}\operatorname{dom} h$, step size $\lambda \in (0, 1/L]$.
  for k = 0,1,... do
      $$x_{k+1} = \operatorname*{argmin}_{u \in \mathbb{R}^d} \langle \nabla f(x_k), u - x_k \rangle + \frac{1}{\lambda} D_h(u, x_k) \tag{3.1}$$
  end for
```

Using first-order optimality conditions, update (3.1) can alternatively be written as

$$x_{k+1} = \nabla h^* \left[\nabla h(x_k) - \lambda \nabla f(x_k)\right]$$

involving the gradient ∇h^* which is usually referred to as the *mirror map.* The three operations $\nabla f, \nabla h$ and ∇h^* are the basic building blocks of Bregman-type methods, which we now define formally.

3.2.4 Defining a class of Bregman first-order methods

For proving a general lower bound for relatively-smooth optimization, we need to specify the oracle model and the class of methods under consideration.

We adopt the first-order black-box model, where information about a function can be gained by calling an *oracle* returning the value and gradient of f at a given point. In the Bregman setting, we assume that we also have access to the first-order oracles of the kernel function h and its conjugate h^*.

Definition 3.4. *An algorithm $\mathcal{A}$ is called a Bregman first-order algorithm if, for a given problem instance $(f, h) \in \mathcal{B}_L$ and number of iterations $T \in \mathbb{N}$, it generates at each time step $t \in \{0, \ldots, T\}$, a set of primal points $\mathcal{X}_t$ and dual points $\mathcal{Y}_t$ from the following process:*

1. *Set $\mathcal{X}_0 = \{x_0\}$, where $x_0 \in \operatorname{int} \operatorname{dom} h$ is some initialization point, and*

$$\mathcal{Y}_0 = \{\nabla f(x_0), \nabla h(x_0)\}.$$

2. *For each $t = 1, \ldots T$, perform one of the two following operations:*

   - *either call the **primal oracle** $(\nabla f, \nabla h)$ at some point x_t chosen such as*

   $$x_t \in \operatorname{Span}(\mathcal{X}_{t-1}) \cap \operatorname{dom} \nabla h$$

   and update the dual set as

   $$\mathcal{Y}_t = \mathcal{Y}_{t-1} \cup \{\nabla f(x_t), \nabla h(x_t)\}.$$

   - *Or call the **mirror oracle** ∇h^* at some dual point y_t chosen such as*

   $$y_t \in \operatorname{Span}(\mathcal{Y}_{t-1})$$

   with

   $$\nabla h^*(y_t) = \operatorname*{argmin}_{u \in C} h(u) - \langle y_t, u \rangle$$

   and update the primal set as

   $$\mathcal{X}_t = \mathcal{X}_{t-1} \cup \{\nabla h^*(y_t)\}.$$

3. *Output some point $x_T \in \operatorname{Span}(\mathcal{X}_T)$.*

Such structural assumptions on the class of algorithms are classical from complexity analyses of Euclidean first-order methods and are used to prove e.g., optimality of accelerated first order methods [Nesterov, 2003]. Definition 3.4 is a natural extension to the Bregman setting, allowing additional uses of the oracles associated with the kernel function h. This model can often be relaxed through the use of more involved information theoretic arguments, see e.g., Nemirovski and Yudin [1983], Guzmán and Nemirovski [2015], Drori [2017], Woodworth and Srebro [2017].

Here, we focus on Definition 3.4 as it is general enough to encompass all Bregman-type methods that only use oracles for $\nabla f, \nabla h$, which we call the *primal oracles*, the map ∇h^*, which we call the *mirror oracle*, as well as linear operations. One can verify that all known Bregman gradient methods, including NoLips and inertial variants such as IGA [Auslender and Teboulle, 2006] or the recent algorithm in Hanzely et al. [2021], fit in this model.

Observe that, as BGD performs one primal oracle call and one mirror call per iteration, an iteration of BGD corresponds actually to *two time steps* of the formal procedure in Definition 3.4. This is why, in order to avoid ambiguity, we state our lower bound as a function of the number of oracle calls.

3.3 Convergence rate and optimality of Bregman gradient descent

In this section, we start by recalling the $O(1/k)$ convergence rate bound for the Bregman gradient/NoLips algorithm in the setting where $(f,h) \in \mathcal{B}_L(\mathcal{C})$. We then proceed to prove that NoLips is an *optimal* algorithm for the class $\mathcal{B}_L(\mathcal{C})$, by showing that this rate is also a *lower bound* for a generic class of Bregman gradient algorithms that we define below. The key elements for proving the lower bound were empirically discovered through the solution to a Performance Estimation Problem (PEP), which is detailed in Chapter 4.

3.3.1 Upper bound

We first recall the $O(1/k)$ convergence rate for Bregman gradient descent/NoLips.

Theorem 3.5 (NoLips convergence rate). *Let $L > 0$, C be a nonempty closed convex subset of $\mathbb{R}^d$ and $(f,h) \in \mathcal{B}_L(C)$ be an relatively-smooth instance. Then the sequence $\{x_k\}_{k\geq 0}$ generated by Algorithm 5 with constant step size $\lambda \in (0, 1/L]$ satisfies for all $k \geq 0$*

$$f(x_k) - f(u) \leq \frac{D_h(u, x_0)}{\lambda\, k} \tag{3.2}$$

for every $u \in \operatorname{dom} h$.

Proof. See [Lu et al., 2018, Thm 3.1.]; we also give an alternative simple proof in Chapter 4 (Section 4.4.1), whose analytical form has been inferred from the solution to a performance estimation problem. ■

Faster algorithms under additional assumptions. It is natural to ask whether an *accelerated* Bregman algorithm can be obtained, with a better convergence rate than $O(1/k)$. This has already been achieved under additional regularity assumptions, as follows:

- in the Euclidean setting, when $h(x) = \frac{1}{2}\|x\|^2$ and f is L-smooth, the seminal accelerated gradient method of Nesterov [1983] enjoys a $O(1/k^2)$ convergence rate, which is optimal for this class of functions [Nesterov, 2003].
- When h is a strongly convex kernel with closed domain and f is L-smooth (which, as discussed in Section 1.2.1, is a particular case of relative smoothness), the Improved Interior Gradient Algorithm (IGA) of Auslender and Teboulle [2006] also admits a $O(1/k^2)$ convergence rate using the same momentum technique as Nesterov-type methods.
- Recently, Hanzely et al. [2021] proposed an accelerated Bregman proximal gradient algorithm with rate $O(1/k^\gamma)$, where $\gamma \in [1,2]$ is determined by some crucial *triangle scaling property* of the Bregman divergence, whose genericity is unclear.

However, the existence of an accelerated algorithm for the general relatively-smooth setting was still an open question prior to this work. Indeed, many applications such as Poisson inverse problems [Bauschke et al., 2017] or D-optimal design [Lu et al., 2018] do not satisfy the supplementary assumptions made in the works mentioned above. In the next section, we prove that, up to a constant factor of 2, the bound (3.2) is not improvable in general for Bregman-type methods, making NoLips an *optimal* algorithm in the black box setting for $(f,h) \in \mathcal{B}_L$.

3.3.2 A lower bound for relatively-smooth Bregman optimization

We show in Theorem 3.12 below that for any $\epsilon \in (0,1)$ and number of oracle calls N, there is a pair of functions $(f,h) \in \mathcal{B}_L(\mathbb{R}^{2N+1})$ and some $x_0 \in \mathbb{R}^{2N+1}$ such that for any *Bregman gradient algorithm* initialized at x_0, the output x_N returned after performing at most N oracle calls satisfies

$$f(x_N) - \min_{\mathbb{R}^{2N+1}} f \geq (1-\epsilon)\frac{LD_h(x_0,x_*)}{2N+1}.$$

Proof intuition. For finding an instance (f,h) which is difficult for all Bregman methods, we use two main ideas. The first is the well-known technique used by Nesterov [2003] for proving that $O(1/k^2)$ is the optimal complexity for L-smooth convex minimization. He defines a "worst function in the world" that allows any gradient method to discover only one dimension per iteration, hence *hiding* the minimizer from the algorithm in the remaining unexplored dimensions.

The second idea is more specific to our setting, and relies on the fact that the set of relatively-smooth problems $\mathcal{B}_L(\mathcal{C})$ is not closed. In particular, a limit of differentiable functions need not be differentiable. Thence, we actually build a worst-case **sequence** of differentiable functions parameterized by some parameter μ, whose limit when $\mu \to 0$ is a nonsmooth pathological function.

Choosing the objective function. Let us fix a dimension $d \geq 1$ and a positive constant $\eta > 0$. Define the convex function $\hat{f}$ for $x \in \mathbb{R}^d$ by

$$\hat{f}(x) = \max_{i=1,\dots,d} |x^{(i)} - 1 - \frac{\eta}{i}| = \|x - x_*\|_\infty$$

which has an optimal value $\hat{f}_* = 0$ attained at $x_* := (1+\eta, 1+\frac{\eta}{2}, \dots, 1+\frac{\eta}{d})$. The behavior of $\hat{f}$ as a *pathological function* comes from the fact that if at least one of the coordinates of x is zero, then $\hat{f}(x) - \hat{f}_* \geq 1$. Let us first prove a technical lemma about the subdifferential of $\hat{f}$.

Lemma 3.6. *Let $x \in \mathbb{R}^d$ and $v \in \partial\hat{f}(x)$ be a subgradient of $\hat{f}$ at x. Then*

(i) $\|v\|_\infty \leq 1$.

(ii) Let $i \in \{1\dots n\}$. If $v^{(i)} \neq 0$ then $|x^{(i)} - x_*^{(i)}| = \|x - x_*\|_\infty$.

Proof. Write $\hat{f}$ as $\hat{f}(x) = \max_{1\leq i\leq d} \hat{f}_i(x)$ with $\hat{f}_i(x) = |x^{(i)} - x_*^{(i)}|$. Then, by [Nesterov, 2003, Lemma 3.1.10], we have

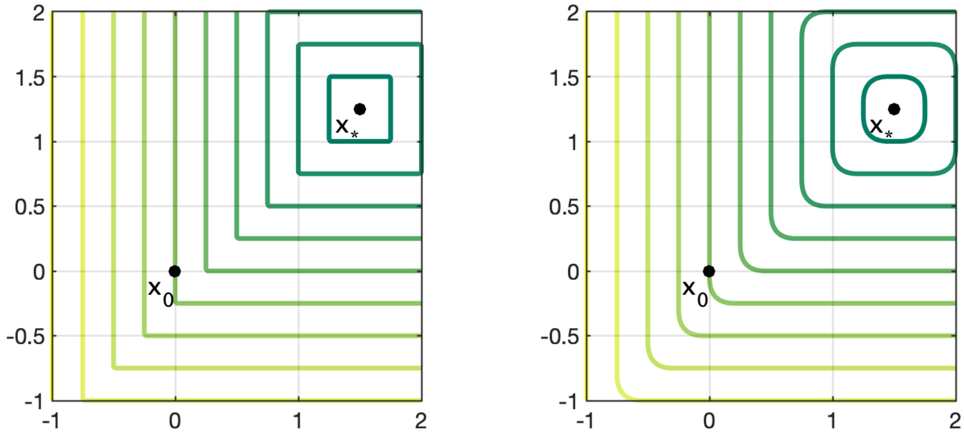
$$\partial\hat{f}(x) = \mathrm{Conv}\,\{\partial\hat{f}_i(x) | i \in I(x)\}$$

where $I(x) = \{i \in \{1\dots d\} \,|\, \hat{f}_i(x) = \hat{f}(x)\}$. Hence, (i) follows immediately from the well-known property that the subgradients of the absolute value lie in $[-1,1]$. (ii) is a consequence of the fact that if $v^{(i)} \neq 0$, then $i \in I(x)$, which means that $|x^{(i)} - x_*^{(i)}| = \|x - x_*\|_\infty$.

■

Note that $\hat{f}$ is nonsmooth hence does not meet our assumptions. We approach it with a differentiable function by considering its Moreau envelope f_μ given by

$$f_\mu(x) = \min_{u\in\mathbb{R}^d} \hat{f}(u) + \frac{1}{2\mu}\|x-u\|^2 \tag{3.3}$$



(a) $\hat{f}$

(b) f_μ

Figure 3.1: Level curves of function $\hat{f}$ (left) and of its smoothed Moreau evelope f_μ (right) for $n = 2, \mu = 0.2$ and $\eta = 1/2$. Lemma 3.8 states that if μ is small enough compared to η, the behaviors of $\hat{f}$ and f_μ at $x_0 = 0$ are the same. Indeed, the size of the smoothed region where the corners are “rounded" decreases when μ goes to 0.

where $\mu \in (0, 1)$ is a small parameter. f_μ is a smoothed version of $\hat{f}$, which behaves similarly to $\hat{f}$ when we choose μ small enough. Figure 3.1 illustrates this phenomenon in two dimensions.

For general properties of the Moreau proximal envelope, we refer to Moreau [1965]. Let us state some properties of f_μ that we need for the analysis.

Lemma 3.7. *f_μ is a differentiable convex function, whose minimizers are the same as those of $\hat{f}$. Its gradient at a point $x \in \mathbb{R}^d$ is given by $\nabla f_\mu(x) = \mu^{-1}\left(x - \mathrm{prox}^{\mu}_{\hat{f}}(x)\right)$ where*

$$\mathrm{prox}^{\mu}_{\hat{f}}(x) = \operatorname*{argmin}_{u \in \mathbb{R}^d} \hat{f}(u) + \frac{1}{2\mu}\|x - u\|^2$$

is the Moreau proximal map. Moreover, ∇f_μ is Lipschitz continuous with constant $1/\mu$.

Let us now prove the central property of f_μ, which states that when the last $d - p$ coordinates of x are small enough, the gradient $\nabla f_\mu(x)$ is supported on the first $p + 1$ coordinates. Recall that we denote $(e_1, \ldots, e_d)$ the canonical basis of $\mathbb{R}^d$ and write, for $p \in \{1 \ldots d\}$, $E_p = \mathrm{Span}(e_1, \ldots, e_p)$ and $E_0 = \{(0, \ldots, 0)\}$.

Lemma 3.8. *Assume that $\mu \in (0, 1)$ and $\eta > 4\mu d^2$. Let $p \in \{0 \ldots d - 1\}$. For any vector $x \in \mathbb{R}^d$ such that*

$$\max_{i=p+1,\ldots,d} |x^{(i)}| \leq \mu$$

we have that $\nabla f_\mu(x) \in E_{p+1}$. In addition, we have $\|\nabla f_\mu(x)\|_\infty \leq 1$.

Proof. Take $x \in \mathbb{R}^d$ such that $\max_{i=p+1,\ldots,n} |x_i| \leq \mu$. By Lemma 3.7, ∇f_μ is given by

$$\nabla f_\mu(x) = \frac{1}{\mu}(x - \mathrm{prox}^{\mu}_{\hat{f}}(x)) \tag{3.4}$$

Write $y = \mathrm{prox}_{\hat{f}}^{\mu}(x)$. The optimality condition defining the proximal map yields

$$y - x + \mu v = 0 \tag{3.5}$$

where $v \in \partial \hat{f}(y)$, and therefore the combination of (3.4) and (3.5) implies

$$\nabla f_\mu(x) = v \in \partial \hat{f}(y). \tag{3.6}$$

Now, let us assume by contradiction that $\nabla f_\mu(x)$ is not in E_{p+1}, meaning that there exists an index
$l \in \{p+2 \dots d\}$ such that $v^{(l)} \neq 0$. It follows from Lemma 3.6 that $|(y - x_*)^{(l)}| = \|y - x_*\|_\infty$. Hence we have in particular that $|y^{(l)} - x_*^{(l)}| \geq |y^{(p+1)} - x_*^{(p+1)}|$. Using Condition (3.5) to replace y we get

$$|x_*^{(l)} + \mu v^{(l)} - x^{(l)}| \geq |x_*^{(p+1)} + \mu v^{(p+1)} - x^{(p+1)}|,$$

and recalling the definition of x_* we have

$$|1 + \frac{\eta}{l} + \mu v^{(l)} - x^{(l)}| \geq |1 + \frac{\eta}{p+1} + \mu v^{(p+1)} - x^{(p+1)}|.$$

By Lemma 3.6, $\|v\|_\infty \leq 1$, so for all i we have $1 + \mu v^{(i)} \geq 1 - \mu\|v\|_\infty \geq 0$. In addition, we assumed that $\max_{i=p+1,\dots,d} |x^{(i)}| \leq \mu < \frac{\eta}{4d^2}$ which implies $\frac{\eta}{i} - x^{(i)} \geq 0$ for all $i \geq p+1$. Therefore, both terms inside the absolute values are nonnegative, it follows that we can drop absolute values and

$$\begin{aligned}
\mu(v^{(l)} - v^{(p+1)}) &\geq \frac{\eta}{p+1} - \frac{\eta}{l} + x^{(l)} - x^{(p+1)} \\
&\geq \eta \cdot \frac{l - (p+1)}{l(p+1)} - 2\mu \\
&\geq \frac{\eta}{l(p+1)} - 2\mu \\
&\geq \frac{\eta}{d^2} - 2\mu,
\end{aligned}$$

and therefore $v^{(l)} - v^{(p+1)} \geq \frac{\eta}{\mu d^2} - 2 > 2$, because we assumed $\eta > 4\mu d^2$. This is a contradiction since $(v^{(l)} - v^{(p+1)}) \leq 2\|v\|_\infty \leq 2$. Finally, the second part of the lemma is a consequence of (3.6) and $\|v\|_\infty \leq 1$.

■

We also need the following lemma for relating the values of $\hat{f}$ and f_μ.

Lemma 3.9. *Let $\mu > 0$ and $x \in \mathbb{R}^d$. Then $f_\mu(x) \geq \hat{f}(x) - \mu$.*

Proof. Write $y = \mathrm{prox}_{\hat{f}}^{\mu}(x)$. By definition of f_μ and the proximal map we have

$$\begin{aligned}
f_\mu(x) &= \hat{f}(y) + \frac{1}{2\mu}\|y - x\|^2 \\
&\geq \hat{f}(y) \\
&= \|y - x_*\|_\infty \\
&\geq \|x - x_*\|_\infty - \|x - y\|_\infty.
\end{aligned}$$

Recall the optimality conditions defining the proximal map can be written as $\mu^{-1}(x-y) \in \partial f(y)$,and, since all subgradients of $\hat{f}$ have coordinates smaller than 1 (Lemma 3.6), we reach $\|x-y\|_\infty \leq \mu$. It follows that $f_\mu(x) \geq \|x-x_*\|_\infty - \|x-y\|_\infty \geq \|x-x_*\|_\infty - \mu = \hat{f}(x) - \mu$, which concludes the proof.

■

Choosing the kernel. As for the objective function f_μ, let us pick a family of kernels h_μ, whose behavior approach those of a nonsmooth function as $\mu \to 0$.

Let us first define a unidimensional convex function $\phi_\mu : \mathbb{R} \to \mathbb{R}$ by

$$\phi_\mu(t) = \begin{cases} t - \mu/2 & \text{if } t \geq \mu, \\ \frac{1}{2\mu}t^2 & \text{elsewhere.} \end{cases}$$

Note that ϕ_μ is sometimes known as the *Huber function*, which is a smooth approximation of the absolute value and also appears as a worst-case function for first-order methods in L-smooth minimization [Taylor et al., 2017].

Define $d_\mu : \mathbb{R}^d \to \mathbb{R}$ through

$$d_\mu(x) = \frac{\mu}{2}\|x\|^2 + \sum_{i=1}^{n} \phi_\mu(x^{(i)}),\ x \in \mathbb{R}^d. \tag{3.7}$$

d_μ is a differentiable strictly convex function, whose gradient satisfies, for $x \in \mathbb{R}^d$ and $i \in \{1 \dots n\}$,

$$\nabla d_\mu(x)^{(i)} = \mu x^{(i)} + \min(1, x^{(i)}/\mu).$$

From the expression above, we can deduce two crucial properties that we need in the sequel: for $x \in \mathbb{R}^d$ and $i \in \{1 \dots n\}$, we have

$$\nabla d_\mu(x)^{(i)} = 0 \quad \text{if and only if} \quad x^{(i)} = 0, \tag{3.8}$$

$$|\nabla d_\mu(x)^{(i)}| \leq 1 \quad \text{implies} \quad |x^{(i)}| \leq \mu. \tag{3.9}$$

Let $L > 0$. We define the kernel h_μ for $x \in \mathbb{R}^d$ as

$$h_\mu(x) = \frac{1}{L}\left(f_\mu(x) + d_\mu(x)\right). \tag{3.10}$$

By construction, $Lh_\mu - f_\mu$ is convex, so the relative smoothness property holds. It is easy to see that Definition 3.1 is satisfied as h_μ is strongly convex, so we have $(f_\mu, h_\mu) \in \mathcal{B}_L(\mathbb{R}^d)$.

Proving the zero-preserving property of the oracles. Now that the functions are defined, we are ready to prove that all oracles involved in the Bregman algorithm allow to discover *only one dimension per oracle call.*

Proposition 3.10 (Zero-preserving property of $\nabla f_\mu, \nabla h_\mu, \nabla h_\mu^*$)**.** *Assume that $\mu \in (0,1)$ and $\eta > 4\mu d^2$. Let $p \in \{0 \dots d-1\}$, and $x \in \mathbb{R}^d \cap E_p$ a vector supported by the p first coordinates. Then*

$$\nabla f_\mu(x), \nabla h_\mu(x), \nabla h_\mu^*(x) \in E_{p+1}.$$

Proof. Let $x \in E_p$. Then x satisfies the assumption of Lemma 3.8 which proves that $\nabla f_\mu(x) \in E_{p+1}$. By Property (3.8) of d_μ, we also have that $\nabla d_\mu(x) \in E_p$, which allows us to conclude that

$$\nabla h_\mu(x) = L^{-1} \left(\nabla f_\mu(x) + \nabla d_\mu\left(x\right)\right) \in E_{p+1}.$$

It remains to prove the result for $\nabla h_\mu^*(x)$. Write $z = \nabla h_\mu^*(x)$, which amounts to say that $\nabla h_\mu(z) = x$, that is

$$\nabla f_\mu(z) + \nabla d_\mu(z) = Lx$$

using (3.10). Let $l \in \{p+1 \dots d\}$. We have $x \in E_p$, hence the $l-th$ coordinate of x is zero and

$$\nabla f_\mu(z)^{(l)} + \nabla d_\mu(z)^{(l)} = 0.$$

Using the second part of Lemma 3.8, we have that $\|\nabla f_\mu(z)\|_\infty \leq 1$; it follows that $|\nabla d_\mu(z)^{(l)}| \leq 1$, which implies that $|z^{(l)}| \leq \mu$, by property (3.9) of d_μ. Since this holds for any $l \geq p+1$, we have established

$$\max_{l=p+1,\dots,d} |z^{(l)}| \leq \mu.$$

Applying Lemma 3.8 to z, we obtain that $\nabla f_\mu(z) \in E_{p+1}$. Remembering that $\nabla h_\mu(z) = x \in E_p$ by construction, we get

$$\nabla d_\mu(z) = L\nabla h_\mu(z) - \nabla f_\mu(z) \in E_{p+1}.$$

By Property (3.8) of d_μ, it follows that $z \in E_{p+1}$, which concludes the proof.

■

We can now use Proposition 3.10 inductively to state a lower bound on the performance of any Bregman gradient algorithm applied to (f_μ, h_μ).

Proposition 3.11. *Let $N \geq 1$ and choose the dimension $d = 2N+1$. Let $\mu \in (0,1)$ and $\eta > 4\mu d^2$. Consider the functions $f_\mu, h_\mu : \mathbb{R}^d \to \mathbb{R}$ defined in* (3.3) *and* (3.10) *respectively. Then, for any Bregman gradient method satisfying Definition 3.4, applied to (f_μ, h_μ) and initialized at $x_0 = (0, \dots 0)$, the output $\overline{x}$ returned after performing at most N calls to each one of the primal and mirror oracles satisfies*

$$f_\mu(\overline{x}) - \min_{\mathbb{R}^d} f_\mu \geq \frac{LD_{h_\mu}(x_*, x_0)}{2N+1} \cdot \frac{1-\mu}{1+\mu+\eta+\frac{\mu}{2}(1+\eta)^2}.$$

Proof. The zero-preserving property and the structure of Bregman gradient algorithms described in Definition 3.4 implies that the set of primal points $\mathcal{X}_t$ and dual points $\mathcal{Y}_t$ at iteration t are supported by the t first coordinates, i.e.,

$$\mathcal{X}_t, \mathcal{Y}_t \subset E_t.$$

Indeed, since we initialized $\mathcal{X}_0 = \{x_0\} \subset E_0$, this follows by induction. Assume that at time t, we have $\mathcal{X}_t, \mathcal{Y}_t \subset E_t$. If the primal oracle is chosen at iteration $t+1$, since the query point x_{t+1} is taken as a linear combination of points in $\mathcal{X}_t$ it also lies in E_t, and thus Proposition 3.10 states that the new dual vectors $\nabla f_\mu(x_{t+1}), \nabla h_\mu(x_{t+1})$ belong to E_{t+1}. If, on the other hand, the mirror oracle is chosen, then with the same argument we have that $y_{t+1} \in E_t$ an by Proposition 3.10 that $\nabla h_\mu^*(y_{t+1}) \in E_{t+1}$.

Now, because the algorithm has called at most N times each oracle, it has performed at most $2N$ steps and thus the output point satisfies $\overline{x} \in E_{2N}$, which means that $\overline{x}^{(2N+1)} = 0$.

We use Lemma 3.9 to relate $f_\mu(\overline{x})$ and $\hat{f}(\overline{x})$. Recalling that $\min f_\mu = \hat{f}_* = 0$, we get

$$\begin{aligned} f_\mu(\overline{x}) - \min_{\mathbb{R}^d} f_\mu &= f_\mu(\overline{x}) \\ &\geq \hat{f}(\overline{x}) - \mu \\ &\geq |\overline{x}^{(2N+1)} - x_*^{(2N+1)}| - \mu \\ &= 1 + \frac{\eta}{2N+1} - \mu \\ &\geq 1 - \mu, \end{aligned} \tag{3.11}$$

where we used the definition of $\hat{f}$ and the fact that $\overline{x}^{(2N+1)} = 0$.

Let us now upper bound the initial diameter. Remembering that $Lh_\mu = f_\mu + d_\mu$ in (3.10), we have

$$LD_{h_\mu}(x_*, x_0) = D_{f_\mu}(x_*, x_0) + D_{d_\mu}(x_*, x_0).$$

by definition of the Bregman divergence. To deal with the first term, we recall that $f_\mu(x_*) = 0$ and write

$$\begin{aligned} D_{f_\mu}(x_*, x_0) &= f_\mu(x_*) - f_\mu(x_0) - \langle \nabla f_\mu(x_0), x_* - x_0 \rangle \\ &= -f_\mu(x_0) - \langle \nabla f_\mu(x_0), x_* - x_0 \rangle \\ &\leq -\hat{f}(x_0) + \mu - \langle \nabla f_\mu(x_0), x_* - x_0 \rangle \\ &= -1 - \eta + \mu - \langle \nabla f_\mu(x_0), x_* - x_0 \rangle, \end{aligned}$$

where we used again Lemma 3.9 at $x_0 = (0, \ldots, 0)$. Now, Lemma 3.8 applies to x_0 with $p = 0$ and allows to state that $\nabla f_\mu(x_0) \in E_1$ and that $\|\nabla f_\mu(x_0)\|_\infty \leq 1$. Therefore

$$|\langle \nabla f_\mu(x_0), x_* - x_0 \rangle| = |\nabla f_\mu(x_0)^{(1)}\,(x_*^{(1)} - x_0^{(1)})| \leq |x_*^{(1)} - x_0^{(1)}| = 1 + \eta.$$

Hence we have the following upper bound

$$D_{f_\mu}(x_*, x_0) \leq -1 - \eta + \mu + |\langle \nabla f_\mu(x_0), x_* - x_0 \rangle| \leq \mu. \tag{3.12}$$

The second term can be directly computed from Definition (3.7) of d_μ, recalling that $x_*^{(i)} \geq 1 \geq \mu$ for $i \in \{0 \ldots d\}$,

$$\begin{aligned} D_{d_\mu}(x_*, x_0) &= d_\mu(x_*) - d_\mu(x_0) - \langle \nabla d_\mu(x_0), x_* - x_0 \rangle \\ &= d_\mu(x_*) \\ &= \sum_{k=1}^{2N+1} \left[\frac{\mu}{2}(1 + \frac{\eta}{k})^2 + 1 + \frac{\eta}{k} - \frac{\mu}{2} \right] \\ &\leq (2N+1) \left[\frac{\mu}{2}(1+\eta)^2 + \eta + 1 \right]. \end{aligned} \tag{3.13}$$

Combining (3.12) and (3.13) gives

$$\begin{aligned} LD_{h_\mu}(x_*, x_0) &= D_{f_\mu}(x_*, x_0) + D_{d_\mu}(x_*, x_0) \\ &\leq \mu + (2N+1) \left[\frac{\mu}{2}(1+\eta)^2 + \eta + 1 \right] \\ &\leq (2N+1) \left[\mu + \frac{\mu}{2}(1+\eta)^2 + \eta + 1 \right]. \end{aligned}$$

This bound, along with (3.11), yields

$$f_\mu(\overline{x}) - \min_{\mathbb{R}^d} f_\mu \geq 1 - \mu \geq \frac{LD_{h_\mu}(x_*, x_0)}{2N+1} \cdot \frac{1-\mu}{1+\mu+\eta+\frac{\mu}{2}(1+\eta)^2}$$

whence the desired result.

■

Since constants μ, η can be taken arbitrarily small, we now use Proposition 3.10 to show that the bound can be approached to any precision and thus prove our main result.

Theorem 3.12 (Lower complexity bound for $\mathcal{B}_L$)**.** *Let $N \geq 1$, a precision $\epsilon \in (0,1)$ and let $x_0 \in \mathbb{R}^{2N+1}$ be a starting point. Then, there exist functions $(f,h) \in \mathcal{B}_L(\mathbb{R}^{2N+1})$ such that for any Bregman gradient method $\mathcal{A}$ satisfying Definition 3.4 and initialized at x_0, the output $\overline{x}$ returned after performing at most N calls to each one of the primal and mirror oracles satisfies*

$$f(\overline{x}) - \min_{\mathbb{R}^{2N+1}} f \geq \frac{LD_h(x_*, x_0)}{2N+1} \cdot (1-\epsilon).$$

Proof. Consider a number N of oracle calls and a target precision $\epsilon \in (0,1)$. Choose the functions f_μ, h_μ defined respectively in Equations (3.3) and (3.10) on $\mathbb{R}^d$ with $d = 2N+1$. These functions satisfy the conditions in Definition 3.3, since their domain is $\mathbb{R}^d$, they are convex, differentiable, and h_μ is strongly convex. Moreover, relative smoothness holds because $Lh_\mu - f_\mu = d_\mu$ is convex by construction. Hence $(f_\mu, h_\mu) \in \mathcal{B}_L(\mathbb{R}^d)$.

Because the class of problems $\mathcal{B}_L(\mathbb{R}^d)$ is invariant by translation, we can assume without loss of generality that the algorithm is initialized at $x_0 = (0, \ldots 0)$. Recall that the only conditions our analysis imposed on the parameters η, μ are that $\mu \in (0,1)$ and $\eta > 4\mu d^2$.

Let us then choose $\eta = \epsilon/4$ and $\mu = \eta/(5d^2) = \epsilon/(20d^2)$. Under these conditions, Proposition 3.11 applies and gives that for any point $\overline{x}$ returned by a Bregman gradient algorithm that is initialized at x_0 and which performs at most N calls to each oracle we have

$$f_\mu(\overline{x}) - \min_{\mathbb{R}^{2N+1}} f_\mu \geq \frac{LD_{h_\mu}(x_*, x_0)}{2N+1} \cdot \frac{1-\mu}{1+\mu+\eta+\frac{\mu}{2}(1+\eta)^2}.$$

The last term can be bounded from below, using our choice of μ, η, and the fact that $\eta < 1$, as

$$\frac{1-\mu}{1+\eta+\mu+\frac{\mu}{2}(1+\eta)^2} \geq \frac{1-\mu}{1+\eta+3\mu} = \frac{1-\frac{\epsilon}{20d^2}}{1+\frac{\epsilon}{4}+\frac{3\epsilon}{20d^2}} \geq 1-\epsilon$$

yielding the desired result.

■

Remark 3. *One can refine the result above in the case where the primal and mirror oracles are not used the same number of times. Indeed, if the primal oracles are called N_1 times and the mirror oracle is called N_2 times, then the same reasoning shows that the lower bound remains true by replacing $2N$ with $N_1 + N_2$.*

Our lower bound involves the relative smoothness constant L instead of the step size λ in (3.2), but it is equivalent (up to a factor 2) when choosing $\lambda = 1/L$, which is actually the best possible step size choice. This shows the optimality of NoLips within the class of Bregman first-order methods (up to a universal constant).

Connection with Conditional Gradient and the ℓ_∞ setting. The worst-case function used for the lower bound involves the smoothing of an ℓ_∞ norm. As pointed out by one of the referees, there might be a connection between the hardness of the relatively-smooth setting and the lower bound for smooth minimization on the ℓ_∞ ball as done in Guzmán and Nemirovski [2015]. This lower bound, which is also $O(1/k)$, is used by the authors to prove that the rate of the Conditional Gradient algorithm is near-optimal in this setting. It might be insightful to examine connections between these settings in future works, for example by exploiting duality between Bregman gradient methods and Conditional Gradient, as in Bach [2015].

3.4 Conclusion

We proved that, in the general relatively-smooth setting, the Bregman gradient/NoLips algorithm is optimal within a large class of Bregman first-order methods. The fundamental idea used for proving our lower bound is that of *limiting nonsmooth pathological behavior.* Indeed, the class of relatively-smooth problem instances is not closed and the worst case functions are reached as (f, h) approach some non-differentiable functions. This idea, along with the corresponding worst-case functions, have been discovered from the solution to a Performance Estimation Problem, which we detail in the next chapter.

Our result shows that additional assumptions on functions f and h are needed in order to prove better bounds or devise faster algorithms than Bregman gradient descent. If the usual properties of L-smoothness and strong convexity are too restrictive and do not hold in many applications, the future challenge is to find weaker assumptions, that define a larger class of functions where improved rates can be obtained. Another possible approach would to enlarge the oracle model and to find algorithms that do not fit in Definition 3.4, for instance by including second-order oracles of h, in the case when h is simple enough.

Chapter 4

Computer-Aided Analyses of Bregman Gradient Methods with Generic Kernels

Chapter Abstract

We show how worst-case scenarios of Bregman gradient methods with general kernel functions can be computed by solving appropriate semidefinite programs. Additionally, the corresponding convergence proof can be inferred from the solution to the dual program. This technique, called performance estimation, has been pioneered by Drori and Teboulle [2014] in the context of Euclidean smooth convex minimization. We extend the performance estimation framework to Bregman methods for relatively-smooth problems, and use it to provide several complexity results in this setting. In particular, numerically generated worst-case examples were used as a basis for obtaining the general lower bound presented in Chapter 3.

Reference: this chapter is based on a publication in Mathematical Programming [Dragomir et al., 2021c]. Part of this work has been done in collaboration with Adrien Taylor.

4.1 Introduction

We consider the constrained minimization problem

$$\min_{x\in\mathcal{C}} f(x)$$

where f is a continuously differentiable convex function and $\mathcal{C}$ is a nonempty closed convex subset of $\mathbb{R}^d$. We are interested in the setting where f satisfies a *relative smoothness* condition with respect to some convex function h [Bauschke et al., 2017]. The standard method for solving such a problem is the Bregman gradient descent scheme (BGD), which writes

$$x_{k+1} = \operatorname*{argmin}_{u\in\mathcal{C}} f(x_k) + \langle \nabla f(x_k), u - x_k\rangle + \frac{1}{\lambda} D_h(u, x_k), \tag{BGD}$$

where

$$D_h(x,y) = h(x) - h(y) - \langle \nabla h(y), x - y \rangle$$

is the Bregman divergence induced by the kernel function h. If f is L-smooth relative to h, that is, if $Lh - f$ is convex, then the iterates of the (BGD) scheme with step size $\lambda = 1/L$ satisfy the convergence rate

$$f(x_k) - f(u) \leq \frac{LD_h(u, x_0)}{k} \tag{4.1}$$

for every $u \in \operatorname{dom} h$ (see Chapter 1). Despite this result, there are still some open questions regarding the complexity of Bregman methods for relatively-smooth minimization. In particular, is the bound (4.1) optimal? Is there another algorithm that achieves a better rate in the same setting, for generic kernels h?

Performance estimation problems. In this work, we adopt a computer-aided technique for analyzing the worst-case behavior of first-order methods, using *performance estimation problems* (PEPs). PEPs were first introduced by Drori and Teboulle [2014] for analyzing the exact convergence rate of Euclidean gradient descent on L-smooth functions. By solving appropriate semidefinite programs, PEPs allow to:

1. compute (numerically) the *exact* worst-case complexity of an algorithm on a given class of problems after a fixed number of iterations,
2. study the corresponding worst-case functions,
3. infer an analytical worst-case guarantee by obtaining a feasible point to the dual PEP. Such dual feasible points correspond to combinations of inequalities that certify the convergence bound.

The PEP technique was used to analyze the worst-case complexity of gradient methods in several settings, such as smooth convex minimization [Drori and Teboulle, 2014, Taylor et al., 2015, Kim and Fessler, 2016, Drori, 2017, Drori and Taylor, 2019, Barré et al., 2020], nonsmooth convex minimization [Drori and Teboulle, 2016, Drori and Taylor, 2019], stochastic optimization [Taylor and Bach, 2019, Drori and Shamir, 2019] and monotone operators [Ryu et al., 2020, Kim, 2019].

Contributions. We propose to use the PEP methodology for computing the worst-case behavior of Bregman gradient methods on the set $\mathcal{B}_L(\mathcal{C})$ of relatively-smooth problem instances with *general kernels*:

$$\mathcal{B}_L(\mathcal{C}) = \{(f,h) \,:\, h \text{ is a kernel function on } \mathcal{C} \text{ and } f \text{ is convex and } L\text{-smooth relative to } h\}\,.$$

To this end, we adapt the interpolation conditions of Taylor et al. [2017] to handle the class of differentiable and strictly convex functions, which appears in the relatively-smooth setting. We then use topological arguments to show that the resulting performance estimation problem is equivalent to a simpler *limiting* problem on a larger class $\overline{\mathcal{B}_L}(\mathcal{C})$, which can be seen as the closure of $\mathcal{B}_L(\mathcal{C})$ and involves possibly nonsmooth functions:

$$\overline{\mathcal{B}_L}(\mathbb{R}^d) = \{(f,h) : f \text{ and } Lh - f \text{ are convex on } \mathbb{R}^d\}.$$

This simpler problem can then be solved numerically with semidefinite optimization packages.

We showcase this approach on several examples, including the proof that (4.1) is the exact worst-case complexity of BGD, a new result on the convergence rate of the stationarity measure $D_h(x_k, x_{k+1})$ for BGD, and the analysis of the inertial Bregman method from Auslender and Teboulle [2006]. Finally, we show how numerically generated worst-case functions were used to infer the lower bound for general Bregman methods from Chapter 3.

Outline. This chapter is organized as follows. In Section 4.3, we describe the methodology and theoretical guarantees of performance estimation problems for Bregman methods. In Section 4.4, we provide numerical experiments and applications.

Notation. We use $\overline{\mathcal{C}}$ to denote the closure of a set $\mathcal{C}$, $\operatorname{int}\mathcal{C}$ for its interior and $\partial\mathcal{C}$ for its boundary. $\mathbf{S}_n$ denotes the set of symmetric matrices of size $n \times n$. If (P) is an optimization problem, then val(P) stands for its (possibly infinite) value.

Subscripts on a vector denote the iteration counter, while a superscript such as $x^{(i)}$ denotes the i-th coordinate. The set $I = \{0, 1, \dots N, *\}$ is often used to index the first d iterates of an optimization algorithm as well as the optimal point:

$$\{x_i\}_{i\in I} = \{x_0, x_1, \dots, x_N, x_*\}.$$

We use the standard notation $\langle\cdot,\cdot\rangle$ for the Euclidean inner product, and $\|\cdot\|$ for the corresponding Euclidean norm. For a vector $x \in \mathbb{R}^d$, we write $\|x\|_\infty = \max_{i=1\dots n} |x^{(i)}|$ for its ℓ_∞ norm.

4.2 Problem setup

In this section, we briefly recall the blanket assumptions and definitions for relatively-smooth optimization. For a general introduction, we refer the reader to Chapter 1.

Definition 4.1 (Kernel function)**.** *A function $h : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is called a kernel function on $\mathcal{C}$ if*

(i) h is closed convex proper (c.c.p.),

(ii) h is continuously differentiable and strictly convex on $\operatorname{int}\mathcal{C}$,

(iii) the Bregman gradient iterates are well-posed, i.e., for every $p \in \mathbb{R}^d$, the problem

$$\min_{u\in C} \langle p, u\rangle + h(u)$$

has a unique minimizer, which belongs to $\operatorname{int}\mathcal{C}$.

Definition 4.2 (Relative smoothness)**.** *Let h be a kernel function on $\mathcal{C}$, and f a function such that $\operatorname{dom} h \subset \operatorname{dom} f$. We say that f is smooth relative to h if it is differentiable on $\operatorname{int}\mathcal{C}$ and if there exists a constant $L > 0$ such that*

$$Lh - f \quad \text{is convex on} \operatorname{int}\mathcal{C}.$$

We recall the notation for a general relatively-smooth problem instance. Note that such an instance is composed of a couple of functions (f, h), as we seek guarantees that are independent of the chosen kernel.

Definition 4.3. *We say that the couple of functions (f, h) is a relatively-smooth instance, and write $(f, h) \in \mathcal{B}_L(\mathcal{C})$ if*

(i) h is a kernel function on $\mathcal{C}$,

(ii) $f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is a closed convex proper function,

(iii) f is L-smooth relative to h on $\mathcal{C}$.

4.3 Worst-case scenarios of Bregman gradient methods through optimization

4.3.1 Formulation of the performance estimation problem

We formulate the task of finding the worst-case performance of (BGD) as an optimization problem. While we focus on the analysis of the vanilla BGD algorithm for ease of presentation, the same ideas are directly applicable to other Bregman-type algorithms like the inertial variant of Auslender and Teboulle [2006], as we illustrate in the sequel.

For simplicity, we first focus on the unconstrained setting where functions have full domain, i.e., $\mathcal{C} = \mathbb{R}^d$ for some $d \geq 1$. In this setting, the set $\mathcal{B}_L(\mathbb{R}^d)$ can be rewritten as

$$\mathcal{B}_L(\mathbb{R}^d) = \left\{ (f, h) : \mathbb{R}^d \to \mathbb{R} \;\middle|\; \begin{array}{l} f \text{ is convex and differentiable,} \\ h \text{ is strictly convex and differentiable,} \\ Lh - f \text{ is convex,} \\ \forall p \in \mathbb{R}^d, \text{ the function } u \mapsto \langle p, u \rangle + h(u) \\ \quad \text{has a unique minimizer.} \end{array} \right\},$$

The general case when $\mathcal{C}$ is a convex subset of $\mathbb{R}^d$ can be treated along the same approach, as we show in the appendix of this chapter. In fact, from the perspective of performance estimation, every problem in $\mathcal{B}_L(\mathcal{C})$ can be reduced to some problem in $\mathcal{B}_L(\mathbb{R}^d)$ with equivalent convergence rate.

Performance estimation problem. Throughout this section, we fix a number of iterations $N \geq 1$, a relative smoothness parameter $L > 0$, and a step size $\lambda > 0$. In the currently known analyses of Bregman gradient descent, worst-case guarantees have the following form

$$f(x_N) - f(x_*) \leq \theta(N, L, \lambda) D_h(x_*, x_0). \tag{4.2}$$

For instance, Equation (4.1) states this result with $\theta(N, L, 1/L) = L/N$. We then naturally seek the smallest $\theta(N, L, \lambda)$ such that the bound (4.2) holds for any couple $(f, h) \in \mathcal{B}_L(\mathbb{R}^d)$, that is, solve the optimization problem

$$\begin{array}{ll} \text{maximize} & \big(f(x_N) - f(x_*)\big)/D_h(x_*, x_0) \\ \text{subject to} & (f, h) \in \mathcal{B}_L(\mathbb{R}^d), \\ & x_* \text{ is a minimizer of } f, \\ & x_1, \dots, x_N \text{ are generated from } x_0 \text{ by BGD with step size } \lambda, \end{array} \tag{PEP}$$

in the variables $f, h, x_0, \dots, x_N, x_*, d$. We refer to this problem as a performance estimation problem (PEP). We use the convention $0/0 = 0$, so that the objective is well defined when $x_* = x_0$. Optimizing over the dimension d to get dimension-free bounds allows for the problem to admit efficient convex reformulations, as we see in the sequel. Importantly, we look for guarantees that are independent of the kernel h, therefore h is part of the optimization variables.

Let us start by simplifying the problem. First, due to the strict convexity of h, the BGD iteration can be equivalently formulated via the first-order optimality conditions

$$\nabla h(x_{i+1}) = \nabla h(x_i) - \lambda \nabla f(x_i) \quad \forall i \in \{0 \dots N-1\},$$

and, since the domain is $\mathbb{R}^d$, the condition that x_* minimizes f reduces to requiring $\nabla f(x_*) = 0$. Second, the problem is homogeneous in (f, h) (i.e., from a feasible couple (f, h), take any constant $c > 0$ and observe that the couple (cf, ch) is also feasible with the same objective value), hence optimizing the objective function $f(x_N) - f(x_*)$ under the additional constraint $D_h(x_*, x_0) = 1$ produces the same optimal value as the problem above.

Finally, we use the same argument as in Drori and Teboulle [2014], Taylor et al. [2017] and observe that the objective of (PEP) and the algorithmic constraints mentioned above depend solely on the values of the first-order oracles of f and h at the points $x_0, \dots, x_N, x_*$. Denoting $I = \{0, 1, \dots, N, *\}$ the indices associated with the points involved in the problem, we proceed to write these values as

$$\begin{aligned} \{(f_i, g_i)\}_{i \in I} &= \left\{\big(f(x_i), \nabla f(x_i)\big)\right\}_{i \in I}, \\ \{(h_i, s_i)\}_{i \in I} &= \{\big(h(x_i), \nabla h(x_i)\big)\}_{i \in I}. \end{aligned}$$

Using those elements, the iterations of BGD can be expressed as

$$s_{i+1} = s_i - \lambda g_i$$

for $i \in \{0 \dots N-1\}$, and the normalization constraint $D_h(x_*, x_0) = 1$ becomes

$$h_* - h_0 - \langle s_0, x_* - x_0 \rangle = 1.$$

Using those *discrete* representations of f and h, we can reformulate (PEP) as

$$\begin{array}{ll} \text{maximize} & f_N - f_* \\ \text{subject to} & f_i = f(x_i), g_i = \nabla f(x_i), \\ & h_i = h(x_i), s_i = \nabla h(x_i), \quad \text{for all } i \in I \text{ and some } (f, h) \in \mathcal{B}_L(\mathbb{R}^d), \\ & g_* = 0, \\ & s_{i+1} = s_i - \lambda g_i \quad \text{for } i \in \{1 \dots N-1\}, \\ & h_* - h_0 - \langle s_0, x_* - x_0 \rangle = 1, \end{array}$$

in the variables $d, \{(x_i, f_i, g_i, h_i, s_i)\}_{i \in I}$. The equivalence with the initial problem is guaranteed by the first two constraints which are called the *interpolation conditions.*

It turns out that interpolation conditions for the class $\mathcal{B}_L(\mathbb{R}^d)$ are delicate to establish, due to assumptions on h. Fortunately, there exist two classes $\underline{\mathcal{B}_L}(\mathbb{R}^d)$ and $\overline{\mathcal{B}_L}(\mathbb{R}^d)$ for which they can be derived. The first class is a restriction of $\mathcal{B}_L(\mathbb{R}^d)$ where f and $Lh - f$ are both assumed to be strictly convex:

$$\underline{\mathcal{B}_L}(\mathbb{R}^d) = \mathcal{B}_L(\mathbb{R}^d) \cap \{(f, h) : \mathbb{R}^d \to \mathbb{R} \,|\, f \text{ and } Lh - f \text{ are strictly convex}\},$$

whereas the second class consists in considering a relaxation with possibly nonsmooth functions:

$$\overline{\mathcal{B}_L}(\mathbb{R}^d) = \{(f,h) : \mathbb{R}^d \to \mathbb{R} \,|\, f \text{ and } Lh - f \text{ are convex}\}.$$

The following inclusions then directly hold

$$\underline{\mathcal{B}_L}(\mathbb{R}^d) \subset \mathcal{B}_L(\mathbb{R}^d) \subset \overline{\mathcal{B}_L}(\mathbb{R}^d).$$

With theses classes, we can now define two easier problems. The first one is a restriction of (PEP) defined on the class $\underline{\mathcal{B}_L}(\mathbb{R}^d)$, under the additional constraint that all iterates are distinct:

$$\begin{array}{ll} \text{maximize} & f_N - f_* \\ \text{subject to} & f_i = f(x_i), g_i = \nabla f(x_i), \\ & h_i = h(x_i), s_i = \nabla h(x_i), \quad \text{for all } i \in I \text{ and some } (f,h) \in \underline{\mathcal{B}_L}(\mathbb{R}^d), \\ & g_* = 0, \\ & s_{i+1} = s_i - \lambda g_i \quad \text{for } i \in \{1 \dots N-1\}, \\ & h_* - h_0 - \langle s_0, x_* - x_0 \rangle = 1, \\ & x_i \neq x_j \quad \text{for } i \neq j \in I, \end{array} \tag{$\underline{\text{PEP}}$}$$

in the variables $d, \{(x_i, f_i, g_i, h_i, s_i)\}_{i \in I}$. The second problem is a relaxation of (PEP), where $(f,h) \in \overline{\mathcal{B}_L}(\mathbb{R}^d)$ are possibly nonsmooth and g_i, s_i are thus *subgradients*:

$$\begin{array}{ll} \text{maximize} & f_N - f_* \\ \text{subject to} & f_i = f(x_i), g_i \in \partial f(x_i), \\ & h_i = h(x_i), s_i \in \partial h(x_i), \\ & Ls_i - g_i \in \partial(Lh - f)(x_i) \quad \text{for all } i \in I \text{ and some } (f,h) \in \overline{\mathcal{B}_L}(\mathbb{R}^d), \\ & g_* = 0, \\ & s_{i+1} = s_i - \lambda g_i \quad \text{for } i \in \{1 \dots N-1\}, \\ & h_* - h_0 - \langle s_0, x_* - x_0 \rangle = 1, \end{array} \tag{$\overline{\text{PEP}}$}$$

in the variables $d, \{(x_i, f_i, g_i, h_i, s_i)\}_{i \in I}$. We added the technical constraints

$$Ls_i - g_i \in \partial(Lh - f)(x_i),$$

which are redundant for differentiable functions; but that are necessary in order to establish interpolation conditions in the nonsmooth case. Because of the inclusions between the feasible sets of these problems, we naturally have

$$\mathrm{val}(\underline{\text{PEP}}) \leq \mathrm{val}(\text{PEP}) \leq \mathrm{val}(\overline{\text{PEP}}).$$

We prove in the sequel that ($\overline{\text{PEP}}$) can be solved via a semidefinite program and that val($\underline{\text{PEP}}$) = val($\overline{\text{PEP}}$) (Theorem 4.8), allowing to reach our claims.

Note that the relaxed problem ($\overline{\text{PEP}}$) does not correspond to any practical algorithm, as BGD is not properly defined for nonsmooth functions h. However, we see in the sequel that feasible points of this problem correspond to accumulation points of (PEP). In other words, instances of BGD can get arbitrarily close to pathological nonsmooth functions whose behaviors are captured by ($\overline{\text{PEP}}$).

In the following sections, we show that problems ($\underline{\text{PEP}}$) and ($\overline{\text{PEP}}$) can be cast as semidefinite programs (SDP) [Vandenberghe and Boyd, 1996] and solved numerically using standard packages [Lofberg, 2004, Mosek, 2019]. The main ingredient consists in showing that interpolation constraints can actually be expressed using quadratic inequalities, as detailed in the next section.

4.3.2 Interpolation involving differentiability and strict convexity

In this section, we show how to reformulate interpolation constraints for (PEP) and ($\overline{\text{PEP}}$) as quadratic inequalities. We start by recalling interpolation conditions for the class of L-smooth and μ-strongly convex functions.

Theorem 4.4 (Smooth strongly convex interpolation, Taylor et al. [2017])**.** *Let I be a finite index set, $\{(x_i, f_i, g_i)\}_{i\in I} \in (\mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^d)^{|I|}$ and $0 \leq \mu \leq L \leq +\infty$. The following statements are equivalent:*

(i) There exists a proper closed convex function $f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ such that f is μ-strongly convex, has a L-Lipschitz continuous gradient and

$$f_i = f(x_i),\ g_i \in \partial f(x_i) \quad \forall i \in I.$$

(ii) For every $i, j \in I$ we have

$$f_i - f_j - \langle g_j, x_i - x_j\rangle \geq \frac{1}{2L}\|g_i - g_j\|^2 + \frac{\mu}{2(1-\mu/L)}\|x_i - x_j - \frac{1}{L}(g_i - g_j)\|^2.$$

In particular, when $L = +\infty$ (meaning that we require no smoothness) and $\mu = 0$, those conditions reduce to the simpler *convex interpolation* conditions, reminiscent of subgradient inequalities:

$$f_i - f_j - \langle g_j, x_i - x_j\rangle \geq 0. \tag{4.3}$$

In our setting, we want to avoid working with smoothness and strong convexity, so we provide interpolation conditions for the class of differentiable strictly convex functions.

Proposition 4.5 (Differentiable and strictly convex interpolation)**.** *Let I be a finite index set and $\{(x_i, f_i, g_i)\}_{i\in I} \in (\mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^d)^{|I|}$. The following statements are equivalent:*

(i) There exists a convex function $f : \mathbb{R}^d \to \mathbb{R}$ such that f is differentiable, strictly convex and

$$f_i = f(x_i),\ g_i = \nabla f(x_i) \quad \forall i \in I.$$

(ii) For every $i, j \in I$ we have

$$\begin{cases} f_i - f_j - \langle g_j, x_i - x_j\rangle > 0 & \textit{if } x_i \neq x_j, \\ f_i = f_j \textit{ and } g_i = g_j & \textit{otherwise.} \end{cases} \tag{4.4}$$

Proof. (i) $\implies$ (ii). Assume that (i) holds, and choose such a function f. The first inequality of (4.4) follows from strict convexity of f, and the second line is a consequence of the fact that a differentiable convex function has a unique subgradient at each point [Rockafellar, 1970, Thm 25.1].

(ii) $\implies$ (i). Assume (ii). If for all $i, j \in I$, we have $g_i = g_j$ and $x_i = x_j$, then there is only one point and one subgradient to be interpolated, and the result follows immediatly from considering a well-chosen definite quadratic function. In the other case, define

$$\nu = \min_{\substack{i,j\in I \\ x_i \neq x_j}} f_i - f_j - \langle g_j, x_i - x_j\rangle.$$

Because of (4.4) and the finiteness of I, we have that $\nu > 0$. Now, define r as

$$r = \max_{i,j \in I} \|g_i - g_j\|^2 + \|x_i - x_j\|^2$$

so that $r > 0$. Condition (4.4) implies that for all $i, j \in I$ we have

$$f_i - f_j - \langle g_j, x_i - x_j \rangle \geq \frac{\nu}{r} \left(\|g_i - g_j\|^2 + \|x_i - x_j\|^2 \right). \tag{4.5}$$

Indeed, if $x_i \neq x_j$, this follows from the definition of ν and r. If $x_i = x_j$ both sides of the inequality are 0 because of the second line in (4.4). Let us choose two constants $0 < \mu < L < +\infty$ such that

$$\frac{1}{L - \mu} \leq \frac{\nu}{r}, \quad \frac{\mu}{1 - \mu/L} \leq \frac{\nu}{r},$$

which is possible as it suffices to take L large enough and μ small enough. We now proceed to show that the interpolation conditions of Theorem 4.4 hold with the constants μ, L defined above. Using the inequality $\|u - v\|^2 \leq 2\|u\|^2 + 2\|v\|^2$ and (4.5), we get that for all i, j,

$$\begin{aligned}
&\frac{1}{2L}\|g_i - g_j\|^2 + \frac{\mu}{2(1 - \mu/L)}\|x_i - x_j - \frac{1}{L}(g_i - g_j)\|^2 \\
&\leq \left(\frac{1}{2L} + \frac{\mu}{L(L - \mu)} \right) \|g_i - g_j\|^2 + \frac{\mu}{1 - \mu/L}\|x_i - x_j\|^2 \\
&\leq \left(\frac{1}{L} + \frac{\mu}{L(L - \mu)} \right) \|g_i - g_j\|^2 + \frac{\mu}{1 - \mu/L}\|x_i - x_j\|^2 \\
&= \frac{1}{L - \mu}\|g_i - g_j\|^2 + \frac{\mu}{1 - \mu/L}\|x_i - x_j\|^2 \\
&\leq \frac{\nu}{r}\|g_i - g_j\|^2 + \frac{\nu}{r}\|x_i - x_j\|^2 \\
&\leq f_i - f_j - \langle g_j, x_i - x_j \rangle.
\end{aligned}$$

Under those conditions, Theorem 4.4 states that there exists a convex function f that interpolates $\{(x_i, f_i, g_i)\}_{i \in I}$ which is μ-strongly convex and has L-Lipschitz continuous gradients. A fortiori, since $\mu > 0$ and $L < \infty$, f is differentiable and strictly convex. Finally, f is finite on $\mathbb{R}^d$ since it is L-smooth. ■

Using these results, we can now formulate interpolation conditions for the problems ($\overline{\text{PEP}}$) and ($\underline{\text{PEP}}$) involving the classes $\overline{\mathcal{B}_L}(\mathbb{R}^d)$ and $\underline{\mathcal{B}_L}(\mathbb{R}^d)$ that were defined above.

Corollary 4.6 (Interpolation conditions for ($\overline{\text{PEP}}$))**.** *Let I be a finite index set and*

$$\{(x_i, f_i, g_i, h_i, s_i)\}_{i \in I} \in (\mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^d)^{|I|}.$$

The following statements are equivalent.

(i) There exist functions $(f, h) \in \overline{\mathcal{B}_L}(\mathbb{R}^d)$ such that

$$\begin{aligned}
f_i = f(x_i),\ & g_i \in \partial f(x_i), \\
h_i = h(x_i),\ & s_i \in \partial h(x_i), \\
Ls_i - g_i \in\ & \partial(Lh - f)(x_i).
\end{aligned}$$

(ii) For all $i, j \in I$ such that $i \neq j$, we have

$$\begin{aligned} f_i - f_j - \langle g_j, x_i - x_j \rangle &\geq 0, \\ (Lh_i - f_i) - (Lh_j - f_j) - \langle Ls_j - g_j, x_i - x_j \rangle &\geq 0. \end{aligned} \tag{4.6}$$

Proof. (i) $\implies$ (ii) follows immediately from the definition of a subgradient applied to convex functions f and $Lh - f$.
Assume that (ii) holds. By the specialization of (4.3) in Theorem 4.4, conditions (ii) imply that there exist two convex functions $f, d : \mathbb{R}^d \to \mathbb{R}$ such that

$$\begin{aligned} f_i = f(x_i), &\quad g_i \in \partial f(x_i), \\ Lh_i - f_i = d(x_i), &\quad Ls_i - g_i \in \partial d(x_i). \end{aligned}$$

Defining the convex function $h = (f + d)/L$, we have that $d = Lh - f$, hence

$$Ls_i - g_i \in \partial(Lh - f)(x_i)$$

for $i \in I$. We also get

$$h_i = h(x_i),\ s_i \in \partial h(x_i),$$

where we used the fact that $Ls_i \in \partial f(x_i) + \partial d(x_i) \subset \partial(f + d)(x_i) = L\partial h(x_i)$ (see [Rockafellar, 1970, Thm 23.8] for the subdifferential of a sum of convex functions). Hence (i) holds. ∎

Corollary 4.7 (Interpolation conditions for (PEP))**.** *Let I be a finite index set and*

$$\{(x_i, f_i, g_i, h_i, s_i)\}_{i \in I} \in (\mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^d)^{|I|}.$$

Assume that $x_i \neq x_j$ for every $i \neq j \in I$. The following statements are equivalent.

(i) There exist functions $(f, h) \in \underline{\mathcal{B}_L}(\mathbb{R}^d)$ such that

$$\begin{aligned} f_i = f(x_i),\ g_i = \nabla f(x_i), \\ h_i = h(x_i),\ s_i = \nabla h(x_i). \end{aligned}$$

(ii) For all $i, j \in I$ such that $i \neq j$ we have

$$\begin{aligned} f_i - f_j - \langle g_j, x_i - x_j \rangle &> 0, \\ (Lh_i - f_i) - (Lh_j - f_j) - \langle Ls_j - g_j, x_i - x_j \rangle &> 0. \end{aligned} \tag{4.7}$$

Proof. Note that since we assumed $x_i \neq x_j$ for every $i \neq j$, interpolation conditions of Proposition 4.5 reduce to requiring a strict inequality in (4.4) for every $i \neq j$. As before, define d the function

$$d = Lh - f.$$

Then since $(f, h) \in \underline{\mathcal{B}_L}(\mathbb{R}^d)$ the functions f and d are differentiable strictly convex, hence the implication (i) $\implies$ (ii) follows simply from strict convexity of these functions.

Conversely, assume (ii). By using Proposition 4.5 again, we can interpolate differentiable strictly convex functions f and d and recover h with $h = (f + d)/L$, thus we have naturally $Lh - f$ convex. The function h is thus also differentiable and strictly convex. Moreover, it can be seen from the proof of Proposition 4.5 that the interpolating functions can actually be chosen strongly convex, hence with this choice the well-posedness conditions in Definition 4.1 holds, and we can conclude that $(f, h) \in \underline{\mathcal{B}_L}(\mathbb{R}^d)$. ∎

4.3.3 Semidefinite reformulations

Now that we established the interpolation conditions for ($\underline{\text{PEP}}$) and ($\overline{\text{PEP}}$), we may use them to obtain semidefinite performance estimation formulations as in Drori and Teboulle [2014], Taylor et al. [2017]. This is made possible by observing that interpolation conditions (4.6)-(4.7) are quadratic inequalities in the problem variables.

Let $\{(x_i, f_i, g_i, h_i, s_i)\}_{i\in I}$ be a feasible point of one of the PEPs in dimension d. We write $G \in \mathbf{S}_{3(N+2)}$ the Gram matrix that contains all dot products between x_i, g_i, s_i for $i \in I$, with

$$G = \begin{pmatrix} G^{xx} & G^{gx} & G^{sx} \\ G^{gx\top} & G^{gg} & G^{gs} \\ G^{sx\top} & G^{gs\top} & G^{ss} \end{pmatrix} \succeq 0$$

whose size is independent of the dimension d, where the blocks are defined as

$$G^{xx}_{ij} = \langle x_i, x_j\rangle,\; G^{gx}_{ij} = \langle g_i, x_j\rangle,\; G^{gs}_{ij} = \langle g_i, s_j\rangle,\; G^{gg}_{ij} = \langle g_i, g_j\rangle, G^{sx}_{ij} = \langle s_i, x_j\rangle,\; G^{ss}_{ij} = \langle s_i, s_j\rangle,$$

for $i, j \in I$. Denote by

$$F = (f_0, \ldots, f_N, f_*) \in \mathbb{R}^{N+2}, \quad H = (h_0, \ldots, h_N, h_*) \in \mathbb{R}^{N+2},$$

the vectors representing the function values of f, h at the iterates. Finally observe that all the constraints of ($\underline{\text{PEP}}$) and ($\overline{\text{PEP}}$) can be expressed using only G, F and H. For instance, interpolation conditions (4.6) for $\overline{\mathcal{B}}_L(\mathbb{R}^d)$ rewrite for all $i, j \in I$ as

$$\begin{aligned} f_i - f_j - G^{gx}_{ji} + G^{gx}_{jj} &\geq 0, \\ (Lh_i - f_i) - (Lh_j - f_j) - L(G^{sx}_{ji} - G^{sx}_{jj}) + G^{gx}_{ji} - G^{gx}_{jj} &\geq 0. \end{aligned}$$

This allows us to reformulate the relaxation ($\overline{\text{PEP}}$) as a semidefinite program, written

$$\begin{array}{ll} \text{maximize} & f_N - f_* \\ \text{subject to} & f_i - f_j - G^{gx}_{ji} + G^{gx}_{jj} \geq 0, \\ & (Lh_i - f_i) - (Lh_j - f_j) - L(G^{sx}_{ji} - G^{sx}_{jj}) + G^{gx}_{ji} - G^{gx}_{jj} \geq 0 \\ & \qquad \text{for } i, j \in I, \\ & G^{gg}_{**} = 0, \\ & G^{sx}_{i+1,j} = G^{sx}_{ij} - \lambda G^{gx}_{ij} \qquad \text{for } i \in \{0 \ldots N-1\}, j \in I, \\ & h_* - h_0 - G^{sx}_{0*} + G^{sx}_{00} = 1, \\ & G \succeq 0, \end{array} \tag{sdp-$\overline{\text{PEP}}$}$$

in the variables $G \in \mathbf{S}_{3(N+2)}$ and $F, H \in \mathbb{R}^{N+2}$.

Any feasible point of ($\overline{\text{PEP}}$) can be cast into an admissible point of (sdp-$\overline{\text{PEP}}$) by computing the semidefinite Gram matrix G. Conversely, if G, F, H is an admissible point of (sdp-$\overline{\text{PEP}}$), then the vectors $\{(x_i, g_i, s_i)\}_{i\in I}$ can be recovered by performing, for instance, a Cholesky decomposition of G. Note that we expressed the algorithmic constraint $s_{i+1} = s_i - \lambda g_i$ only through scalar products with the x_i's in the SDP, since only the projection of the gradients on $\mathrm{Span}(\{x_i\}_{i\in I})$ is relevant in the PEPs. Because interpolation conditions from Corollary 4.6 are necessary and sufficient, we conclude that the problems are equivalent, that is

$$\mathrm{val}(\text{sdp-}\overline{\text{PEP}}) = \mathrm{val}(\overline{\text{PEP}}).$$

The rank of G determines the dimension of the interpolated problem. If we look instead for a solution that has a given dimension d, this would mean imposing a nonconvex rank constraint on G. Our formulation, on the other hand, is convex and finds the best convergence bound that is dimension-independent, which is a usual requirement for large-scale settings.

In the same way, the value of ($\underline{\text{PEP}}$) can be computed as

$$\begin{array}{ll} \text{maximize} & f_N - f_* \\ \text{subject to} & f_i - f_j - G^{gx}_{ji} + G^{gx}_{jj} > 0, \\ & (Lh_i - f_i) - (Lh_j - f_j) - L(G^{sx}_{ji} - G^{sx}_{jj}) + G^{gx}_{ji} - G^{gx}_{jj} > 0 \\ & \quad \text{for } i \neq j \in I, \\ & G^{gg}_{**} = 0, \\ & G^{sx}_{i+1,j} = G^{sx}_{ij} - \lambda G^{gx}_{ij} \quad \text{for } i \in \{0 \dots N-1\}, j \in I, \\ & h_* - h_0 - G^{sx}_{0*} + G^{sx}_{00} = 1, \\ & G^{xx}_{ii} + G^{xx}_{jj} - 2G^{xx}_{ij} > 0 \quad \text{for } i \neq j \in I, \\ & G \succeq 0, \end{array} \tag{sdp-$\underline{\text{PEP}}$}$$

in the variables $G \in \mathbf{S}_{3(N+2)}$ and $F, H \in \mathbb{R}^{N+2}$, where we used interpolation conditions for $\underline{\mathcal{B}_L}(\mathbb{R}^d)$ from Corollary 4.7, since all points $\{x_i\}_{i \in I}$ are constrained to be distinct. Therefore, as above we infer that

$$\text{val}(\text{sdp-}\underline{\text{PEP}}) = \text{val}(\underline{\text{PEP}}).$$

Recalling the hierarchy between the problems, we thus have

$$\text{val}(\text{sdp-}\underline{\text{PEP}}) \leq \text{val}(\text{PEP}) \leq \text{val}(\text{sdp-}\overline{\text{PEP}}).$$

By comparing the two semidefinite programs stated above, one can notice that the only difference is that (sdp-$\underline{\text{PEP}}$) imposes some inequalities of (sdp-$\overline{\text{PEP}}$) to be strict. In the next section, we use topological arguments to prove that the values of the two problems are actually equal. In fact, strict inequalities have little meaning in numerical optimization (the value of (sdp-$\underline{\text{PEP}}$) is actually a supremum and not a maximum); in our experiments, we focus on (sdp-$\overline{\text{PEP}}$) as solvers usually admit only closed feasible sets.

4.3.4 Tightness of the approach: nonsmooth limit behaviors

We are now ready to prove the main result of this section.

Theorem 4.8. *The value of the performance estimation problem* (PEP) *for BGD is equal to the value of the nonsmooth relaxation* ($\overline{\text{PEP}}$)*, which can be computed by solving the semidefinite program* (sdp-$\overline{\text{PEP}}$)*.*

Proof. We show that the closure of the feasible set of (sdp-$\underline{\text{PEP}}$) is the feasible set of (sdp-$\overline{\text{PEP}}$). We first need to prove that the strengthened problem ($\underline{\text{PEP}}$) is feasible, by finding an instance of BGD where f and $Lh - f$ are strictly convex and such that all iterates are distinct. It suffices for instance to consider two one-dimensional quadratic functions. Define $f, h : \mathbb{R} \to \mathbb{R}$ with

$$f(x) = \frac{\alpha}{2}x^2, \; h(x) = \frac{1}{2}x^2 \quad \text{where } \alpha = \min\left(\frac{1}{2\lambda}, \frac{L}{2}\right).$$

Then f is strictly convex and so is $Lh - f = \frac{L-\alpha}{2}x^2$ since $L - \alpha \geq \frac{L}{2} > 0$. The optimum is $x_* = 0$. Choose

$$x_0 = \sqrt{2}$$

for which we have $D_h(x_*, x_0) = x_0^2/2 = 1$. Then, BGD is equivalent to gradient descent and the iterates satisfy

$$x_N = (1 - \lambda\alpha)^N x_0.$$

Since $\alpha\lambda \leq 1/2 < 1$, all the iterates are distinct and therefore we constructed a feasible point of ($\underline{\text{PEP}}$). Let us therefore write (G, F, H) a corresponding feasible point of (sdp-$\underline{\text{PEP}}$), and $(\overline{G}, \overline{F}, \overline{H})$ a feasible point of (sdp-$\overline{\text{PEP}}$). Define the sequence $\{(G^k, F^k, H^k)\}_{k\geq 1}$ as

$$\begin{aligned} G^k &= \frac{1}{k}G + (1 - \frac{1}{k})\overline{G}, \\ F^k &= \frac{1}{k}F + (1 - \frac{1}{k})\overline{F}, \\ H^k &= \frac{1}{k}H + (1 - \frac{1}{k})\overline{H}. \end{aligned}$$

Then, for every $k \geq 1$, (G^k, F^k, H^k) is still a feasible point of (sdp-$\underline{\text{PEP}}$), because of convexity of the constraints and the fact that adding a strict inequality to a weak inequality gives a strict inequality. Moreover, the sequence converges to the point $(\overline{G}, \overline{f}, \overline{h})$ when $k \to +\infty$.

Hence we proved that for any feasible point of (sdp-$\overline{\text{PEP}}$), there is a sequence of admissible points of (sdp-$\underline{\text{PEP}}$) that converge to it. Since the objective is linear in the vector F therefore continuous, we deduce that the two problems have the same value:

$$\text{val(sdp-}\underline{\text{PEP}}) = \text{val(sdp-}\overline{\text{PEP}}),$$

which means that val($\underline{\text{PEP}}$) = val($\overline{\text{PEP}}$). As val(PEP) lies in between these two values, we conclude that they are all equal. ■

Theorem 4.8 states that the value of the original problem (PEP) can be computed numerically with a semidefinite solver applied to (sdp-$\overline{\text{PEP}}$). The result itself also helps us gain some theoretical insight: it tells us that the worst-case for BGD on $\mathcal{B}_L(\mathbb{R}^d)$ might be reached as (f, h) approach possibly pathological limiting nonsmooth functions in $\overline{\mathcal{B}_L}(\mathbb{R}^d)$.

4.4 Numerical evidence and computer-assisted proofs

We now provide several applications of the performance estimation framework that we developed for Bregman methods.

4.4.1 Finding the exact worst-case convergence rate of BGD

We first start by the most direct application, that is finding the exact worst-case performance of BGD. Theorem 4.8 states that it can be computed by solving the semidefinite program (sdp-$\overline{\text{PEP}}$). The link to the Matlab implementation is provided in Section 4.5.

To simplify our setting, note that we can assume without loss of generality that the relative smoothness constant L is 1, since we can replace h by a scaled version Lh. Recall that we know from Theorem 3.5, that

$$\text{val(PEP)} \leq \frac{1}{\lambda N}.$$

Table 4.1 shows the result of solving (sdp-$\overline{\text{PEP}}$) for several values of N up to 100, for a step size $\lambda = 1/L$. We observe that with high precision, val(sdp-$\overline{\text{PEP}}$) is equal to the theoretical bound $1/(\lambda N)$.

Table 4.1: Numerical value of the performance estimation problem (PEP) with $\lambda = 1$, $L = 1$. *Rel. error* denotes the relative error between val(PEP) and the theoretical bound of $1/N$ given by Theorem 3.5. *Primal feasibility* corresponds to the maximal absolute value of constraint violation returned by the MOSEK solver.

N	val(PEP)	Rel. error	Primal feasibility
1	1.000	1.8e-11	4.3e-10
2	0.500	1.8e-8	2.8e-9
3	0.333	1.8e-8	2.8e-9
4	0.250	4.9e-8	2.3e-8
5	0.200	1.8e-10	6.4e-11
10	0.100	6.4e-11	1.3e-11
20	0.050	1.1e-8	1.9e-10
50	0.020	6.5e-6	5.0e-7
100	0.01	7.2e-5	1.6e-6

Other values of λ. One can wonder how the numerical value evolves when one varies the step size λ. Our experimental observations are as follows:

- For any $\lambda \in (0, 1/L]$, val(PEP) is exactly equal to the theoretical bound $1/(\lambda N)$.
- For any $\lambda > 1/L$, val(PEP) $= +\infty$, hence BGD does not converge in general with these step size values. This suggests that the maximal step size value allowed for BGD is indeed $1/L$, unlike the Euclidean setting where gradient descent can be applied with a step size that goes up to $2/L$ [Nesterov, 2003].

While results above suggest that $1/(\lambda N)$ is the exact worst-case rate of BGD, they provide only numerical evidence. We can however use them to deduce formal guarantees, both for proving an *upper bound* and a *lower bound*.

Upper bound guarantee through duality. As noticed in previous work on PEPs [Drori and Teboulle, 2014, Taylor et al., 2015], solving the dual of (sdp-$\overline{\text{PEP}}$) can be used to deduce a proof. Indeed, the dual solution gives a combination of the constraints that, when transposed to analytical form, leads to a formal guarantee. This provides the following proof for the $O(1/k)$ convergence rate of Theorem 3.5, which we recall here.

Theorem 4.9 (NoLips convergence rate, recall of Theorem 3.5)**.** *Let $L > 0$, C be a nonempty closed convex subset of $\mathbb{R}^d$ and $(f, h) \in \mathcal{B}_L(C)$ be an relatively-smooth instance. Then the sequence $\{x_k\}_{k\geq 0}$ generated by Algorithm 5 with constant step size $\lambda \in (0, 1/L]$ satisfies for all $k \geq 0$*

$$f(x_k) - f(u) \leq \frac{D_h(u, x_0)}{\lambda k}$$

for every $u \in \operatorname{dom} h$.

Proof. The proof relies on the fact that, since $Lh - f$ is convex we have that $\frac{1}{\lambda}h - f$ is convex for any $\lambda \in (0, \frac{1}{L}]$, and only consists in performing the following weighted sum of inequalities:

- convexity of f, between u and x_i $(i = 0, \ldots, k)$ with weights $\gamma_{*,i} = \frac{1}{k}$:

$$f(u) \geq f(x_i) + \langle \nabla f(x_i), u - x_i \rangle,$$

- convexity of f, between x_i and x_{i+1} $(i = 0, \ldots, k-1)$ with weights $\gamma_{i,i+1} = \frac{i}{k}$:

$$f(x_i) \geq f(x_{i+1}) + \langle \nabla f(x_{i+1}), x_i - x_{i+1} \rangle,$$

- convexity of $\frac{1}{\lambda}h - f$, between u and x_k with weight $\mu_{*,k} = \frac{1}{k}$:

$$\tfrac{1}{\lambda}h(u) - f(u) \geq \tfrac{1}{\lambda}h(x_k) - f(x_k) + \langle \tfrac{1}{\lambda}\nabla h(x_k) - \nabla f(x_k), u - x_k \rangle,$$

- convexity of $\frac{1}{\lambda}h - f$, between x_{i+1} and x_i $(i = 0, \ldots, k-1)$ with weight $\mu_{i+1,i} = \frac{i+1}{k}$

$$\tfrac{1}{\lambda}h(x_{i+1}) - f(x_{i+1}) \geq \tfrac{1}{\lambda}h(x_i) - f(x_i) + \langle \tfrac{1}{\lambda}\nabla h(x_i) - \nabla f(x_i), x_{i+1} - x_i \rangle,$$

- convexity of $\frac{1}{\lambda}h - f$, between x_i and x_{i+1} $(i = 0, \ldots, k-1)$ with weight $\mu_{i,i+1} = \frac{i}{k}$

$$\tfrac{1}{\lambda}h(x_i) - f(x_i) \geq \tfrac{1}{\lambda}h(x_{i+1}) - f(x_{i+1}) + \langle \tfrac{1}{\lambda}\nabla h(x_{i+1}) - \nabla f(x_{i+1}), x_i - x_{i+1} \rangle.$$

The weighted sum is written as

$$\begin{aligned}
0 \geq & \sum_{i=0}^{k} \gamma_{*,i} \left[f(x_i) - f(u) + \langle \nabla f(x_i), u - x_i \rangle \right] \\
& + \sum_{i=0}^{k-1} \gamma_{i,i+1} \left[f(x_{i+1}) - f(x_i) + \langle \nabla f(x_{i+1}), x_i - x_{i+1} \rangle \right] \\
& + \mu_{*,k} \left[\tfrac{1}{\lambda}h(x_k) - f(x_k) - (\tfrac{1}{\lambda}h(u) - f(u)) + \langle \tfrac{1}{\lambda}\nabla h(x_k) - \nabla f(x_k), u - x_k \rangle \right] \\
& + \sum_{i=0}^{k-1} \mu_{i+1,i} \left[\tfrac{1}{\lambda}h(x_i) - f(x_i) - (\tfrac{1}{\lambda}h(x_{i+1}) - f(x_{i+1})) + \langle \tfrac{1}{\lambda}\nabla h(x_i) - \nabla f(x_i), x_{i+1} - x_i \rangle \right] \\
& + \sum_{i=0}^{k-1} \mu_{i,i+1} \left[\tfrac{1}{\lambda}h(x_{i+1}) - f(x_{i+1}) - (\tfrac{1}{\lambda}h(x_i) - f(x_i)) + \langle \tfrac{1}{\lambda}\nabla h(x_{i+1}) - \nabla f(x_{i+1}), x_i - x_{i+1} \rangle \right].
\end{aligned}$$

By substitution of $\nabla h(x_{i+1}) = \nabla h(x_i) - \lambda \nabla f(x_i)$ $(i = 0, \ldots, k-1)$, one can reformulate the weighted sum exactly as (i.e., there is no residual):

$$0 \geq f(x_k) - f(u) - \tfrac{h(u) - h(x_0) - \langle \nabla h(x_0), u - x_0 \rangle}{\lambda k},$$

yielding the desired result. ■

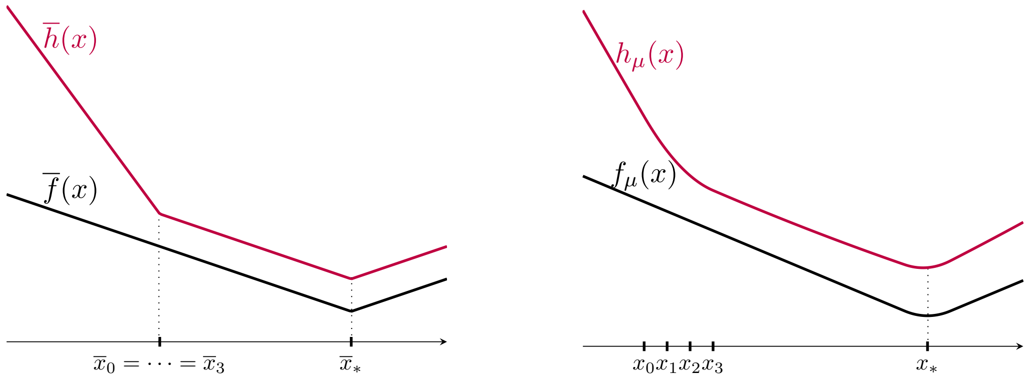


Figure 4.1: Worst-case functions for BGD in dimension 1 with $N = 3$ iterations. The left figure shows the limiting instance $(\overline{f}, \overline{h}) \in \overline{\mathcal{B}_L}(\mathbb{R})$, while the right plot represents the smooth approximation by a valid instance $(f_\mu, h_\mu) \in \mathcal{B}_L(\mathbb{R})$, with smoothing parameter $\mu = 0.1$. As μ goes to 0, functions f_μ, h_μ tend to a pathological behavior where all iterates are equal and for which we have exactly $\overline{f}(\overline{x}_N) - \overline{f}_* = D_{\overline{h}}(\overline{x}_*, \overline{x}_0)/N$.

Lower bound through worst-case functions. As (PEP) computes the *exact* worst-case performance of BGD, experiments above suggest that $1/(\lambda N)$ is also a lower bound, meaning that for every $\epsilon > 0$, there exist functions $(f, h) \in \mathcal{B}_L$ such that the iterates of BGD satisfy

$$f(x_N) - f_* \geq \frac{D_h(x_*, x_0)}{\lambda N} - \epsilon.$$

We detail here how such functions can be constructed from the solution of (sdp-$\overline{\text{PEP}}$). The numerical solver allows us to find a maximizer $\overline{G}, \overline{F}, \overline{H}$ (recall that only the relaxed problem has a maximizer as the feasible set is closed), and by factorizing the matrix G as $P^T P$, we can thus recover the corresponding discrete representation $\{\overline{x}_i, \overline{g}_i, \overline{f}_i, \overline{h}_i, \overline{s}_i\}_{i \in I}$. This discretization can in turn be interpolated to get the corresponding functions $(\overline{f}, \overline{h}) \in \overline{\mathcal{B}_L}$. There are multiple ways to perform this interpolation; see [Taylor et al., 2017, Thm. 1] for a constructive approach.

Recall that since functions $(\overline{f}, \overline{h})$ yield a solution to ($\overline{\text{PEP}}$), they belong to $\overline{\mathcal{B}_L}$ and might thus form a *pathological* nonsmooth limiting worst-case. They can be approached by valid instances $(f_\mu, h_\mu) \in \mathcal{B}_L$ by performing for instance smoothing through Moreau evelopes (as in Chapter 3) and adding a small quadratic to h to make it strictly convex.

There are however many possible maximizers of (sdp-$\overline{\text{PEP}}$). If we seek a low-dimensional example that may be easily interpretable, we can search for a maximizer such that the Gram matrix G has minimal rank. Using rank minimization heuristics, we were able to find one-dimensional worst-case functions. Fix a number of iterations $N \geq 1$, assume $\lambda = 1/L = 1$ and define $\overline{f}, \overline{h} : \mathbb{R} \to \mathbb{R}$ as

$$\begin{aligned} \overline{f}(x) &= |x - 1|, \\ \overline{h}(x) &= \overline{f}(x) + \max(-Nx, 0), \end{aligned}$$

and set $\overline{x}_0 = 0, \overline{x}_* = 1$. Then clearly $(\overline{f}, \overline{h}) \in \overline{\mathcal{B}_L}(\mathbb{R})$. Figure 4.1 shows the functions $\overline{f}, \overline{h}$ as well as their smoothed versions $(f_\mu, h_\mu) \in \mathcal{B}_L(\mathbb{R})$. Note that the pathological behavior also reflects in the iterates: in the limiting instance, all iterates $\overline{x}_0, \ldots, \overline{x}_N$ are equal. In the smoothed version, iterates are distinct (since h_μ is strictly convex), but they get closer and closer as the smoothing parameter μ goes to 0.

The smoothed function f_μ is a Huber function, which is also the worst-case instance for Euclidean gradient descent on L-smooth functions described in Taylor et al. [2017]. This analysis could be formalized to prove the $1/k$ lower bound for BGD; however, this bound is just a particular case of the stronger result for general Bregman gradient methods derived in Chapter 3.

4.4.2 Extension to other criteria

In our performance estimation problem, we focused on studying bounds of the form

$$f(x_N) - f_* \leq \theta(N, L, \lambda) D_h(x_*, x_0).$$

However, we are not limited to this criterion, and different convergence measures might be considered by changing the objective and constraints in (PEP). For instance, another popular criterion is the stationarity measure $D_h(x_k, x_{k+1})$, which boils down to the squared gradient norm in the unconstrained Euclidean case. By adapting (PEP), we get the following new convergence result for BGD.

Proposition 4.10 (BGD convergence rate, take II)**.** *Let $L > 0$, C be a nonempty closed convex subset of $\mathbb{R}^d$ and $(f, h) \in \mathcal{B}_L(C)$ a relatively-smooth problem instance. Then the sequence $\{x_k\}_{k\geq 0}$ generated by Bregman gradient descent with constant step size $\lambda \in (0, 1/L]$ satisfies for $k \geq 2$*

$$\min_{1\leq i\leq k} D_h(x_{i-1}, x_i) \leq \frac{2D_h(x_*, x_0)}{k(k-1)}$$

for every $x_* \in \operatorname{argmin}_C f \cap \operatorname{dom} h$.

Proof. In the same way as before, the formal guarantee has been obtained by examining the dual of the corresponding PEP. The proof relies on the fact that $\frac{1}{\lambda}h - f$ is convex for any $\lambda \in (0, \frac{1}{L}]$, and only consists in performing the following weighted sum of inequalities:

- convexity of f, between x_* and x_i $(i = 0, \ldots, k)$ with weights $\gamma_{*,i} = \frac{2\lambda}{k(k-1)}$:

$$f(x_*) \geq f(x_i) + \langle \nabla f(x_i), x_* - x_i \rangle,$$

- optimality of x_* for each x_k with weight $\gamma_{k,*} = \frac{2\lambda}{k-1}$:

$$f(x_k) \geq f(x_*),$$

- convexity of $\frac{1}{\lambda}h - f$, between x_* and x_k with weight $\mu_{*,k} = \frac{2\lambda}{k(k-1)}$:

$$\tfrac{1}{\lambda}h(x_*) - f(x_*) \geq \tfrac{1}{\lambda}h(x_k) - f(x_k) + \langle \tfrac{1}{\lambda}\nabla h(x_k) - \nabla f(x_k), x_* - x_k \rangle,$$

- convexity of $\frac{1}{\lambda}h - f$, between x_{i+1} and x_i $(i = 0, \ldots, k-1)$ with weight $\mu_{i+1,i} = \frac{2\lambda(i+1)}{k(k-1)}$

$$\tfrac{1}{\lambda}h(x_{i+1}) - f(x_{i+1}) \geq \tfrac{1}{\lambda}h(x_i) - f(x_i) + \langle \tfrac{1}{\lambda}\nabla h(x_i) - \nabla f(x_i), x_{i+1} - x_i \rangle,$$

- definition of smallest residual among the iterates ($i = 1, \dots, k$) with weights $\tau_i = \frac{2(i-1)}{k(k-1)}$:

$$h(x_{i-1}) - h(x_i) - \langle \nabla h(x_i), x_{i-1} - x_i \rangle \geq \min_{1 \leq j \leq k} \{D_h(x_{j-1}, x_j)\}.$$

The weighted sum is written as

$$\begin{aligned}
0 \geq &\sum_{i=0}^{k} \gamma_{*,i}[f(x_i) - f(x_*) + \langle \nabla f(x_i), x_* - x_i \rangle] \\
&+ \gamma_{k,*}[f(x_*) - f(x_k)] \\
&+ \mu_{*,k}[\tfrac{1}{\lambda} h(x_k) - f(x_k) - (\tfrac{1}{\lambda} h(x_*) - f(x_*)) + \langle \tfrac{1}{\lambda} \nabla h(x_k) - \nabla f(x_k), x_* - x_k \rangle] \\
&+ \sum_{i=0}^{k-1} \mu_{i+1,i}[\tfrac{1}{\lambda} h(x_i) - f(x_i) - (\tfrac{1}{\lambda} h(x_{i+1}) - f(x_{i+1})) + \langle \tfrac{1}{\lambda} \nabla h(x_i) - \nabla f(x_i), x_{i+1} - x_i \rangle] \\
&+ \sum_{i=1}^{k} \tau_i [\min_{1 \leq j \leq k} \{D_h(x_{j-1}, x_j)\} - (h(x_{i-1}) - h(x_i) - \langle \nabla h(x_i), x_{i-1} - x_i \rangle)].
\end{aligned}$$

By substitution of $\nabla h(x_{i+1}) = \nabla h(x_i) - \lambda \nabla f(x_i)$ ($i = 0, \dots, k-1$), one can reformulate the weighted sum exactly as (i.e., there is no residual):

$$0 \geq \min_{1 \leq j \leq k} \{D_h(x_{j-1}, x_j)\} - 2 \cdot \frac{h(x_*) - h(x_0) - \langle \nabla h(x_0), x_* - x_0 \rangle}{k(k-1)},$$

yielding the desired result. ■

4.4.3 Beyond BGD: accelerated Bregman algorithms

Our approach is not limited to the vanilla BGD algorithm. For instance, we can also solve the performance estimation problem for the accelerated Bregman algorithm proposed by Auslender and Teboulle [2006], a.k.a. the Improved Interior Gradient Algorithm (IGA). We recall its simplified formulation in Algorithm 6, in the case where there are no affine constraints.

Algorithm 6 Improved Interior Gradient Algorithm (IGA) [Auslender and Teboulle, 2006]

```
Input: Functions $f, h$, initial point $x_0 \in \operatorname{int} \operatorname{dom} h$, step size $\lambda$.
Set $z_0 = x_0$ and $t_0 = 1$.
for k = 0,1,... do
    $y_k = (1 - \frac{1}{t_k}) x_k + \frac{1}{t_k} z_k$
    $z_{k+1} = \operatorname{argmin} \{\langle \nabla f(y_k), u - y_k \rangle + \frac{1}{t_k \lambda} D_h(u, z_k) \,|\, u \in \mathbb{R}^d\}$
    $x_{k+1} = (1 - \frac{1}{t_k}) x_k + \frac{1}{t_k} z_{k+1}$
    $t_{k+1} = (1 + \sqrt{1 + 4t_k^2})/2.$
end for
```

In the setting where f has $\tilde{L}$-Lipschitz continuous gradients and h is a σ-strongly convex kernel function, IGA with step size $\lambda = \sigma / \tilde{L}$ enjoys the following convergence rate [Auslender

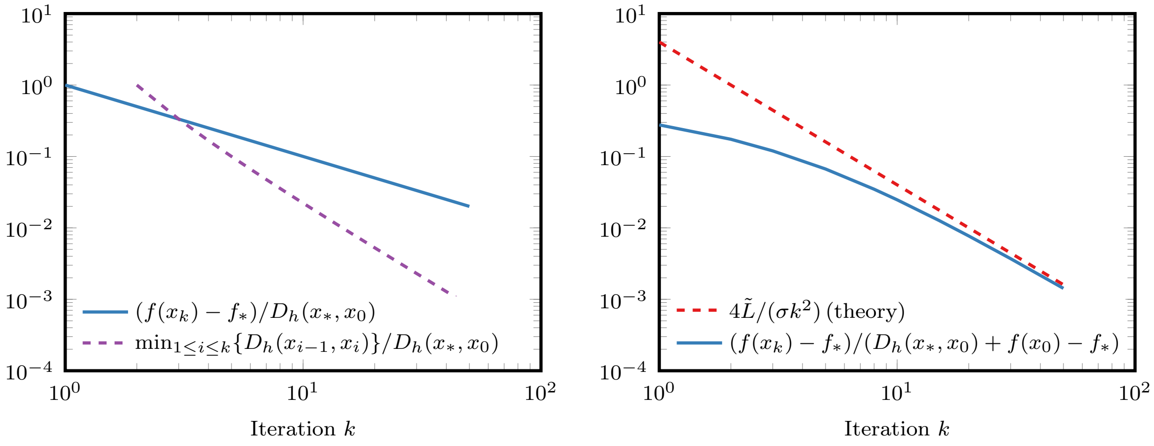


Figure 4.2: Numerical worst-case guarantees obtained from PEPs as functions of the iteration counter k (shown in log scale as rates are sublinear). **Left:** guarantees for Bregman Gradient descent for two different convergence measures. Numerical values confirm exactly the theoretical rates of Theorem 3.5 and Proposition 4.10. **Right:** guarantees for IGA with no affine constraints (Algorithm 6) under the assumption that h is 1-strongly convex and f is 1-smooth, compared to the theoretical bound from Auslender and Teboulle [2006]. Notice that the theoretical bound is not tight in this case, as it is obtained by making some approximations in the proof.

and Teboulle, 2006, Thm. 5.2]:

$$f(x_N) - f_* \leq \frac{4\tilde{L}}{\sigma N^2} \left(D_h(x_*, x_0) + f(x_0) - f_*\right). \tag{4.8}$$

Our PEP framework can also be applied to this algorithm, in order to find the smallest value of $\theta(N, \tilde{L}, \sigma, \lambda)$ which satisfies

$$f(x_N) - f_* \leq \theta(N, \tilde{L}, \sigma, \lambda) \left(D_h(x_*, x_0) + f(x_0) - f_*\right)$$

for every instance of IGA with the supplementary assumptions made above. In this case, we use the standard interpolation conditions of Theorem 4.4 for L-smooth and strongly convex functions. Results are shown in Figure 4.2. The exact numerical worst-case performance of IGA is slightly below the theoretical bound above, since the proof in Auslender and Teboulle [2006] makes some approximations.

IGA in the general relatively-smooth case: failure of acceleration. We pointed out in Section 1.2 that the setting in which f is $\tilde{L}$-smooth and h is σ-strongly convex is a particular case of relative smoothness with constant $L = \tilde{L}/\sigma$. The natural question that was also raised in [Teboulle, 2018, Section 6] is therefore: does IGA converge for the general class $\mathcal{B}_L$? Solving the corresponding PEP yields the following results. For Algorithm 6 with the setting that $(f, h) \in \mathcal{B}_L$ and several choices of step size in $(0, 1/L]$, the solver states that the value of the corresponding performance estimation problem **is unbounded**, i.e., there does not exist any θ such that the bound (4.8) holds for every instance $(f, h) \in \mathcal{B}_L$.

As suggested by the anonymous reviewers, one could legitimately wonder whether there exist other sequences $\{t_k\}_{k\geq 0}$ with $t_k > 1$, perhaps *less aggressive* than the one in Algorithm 6, such that the method converges (note that choosing $t_k = 1$ for all k would yield the standard BGD scheme). After solving the PEP with several choices of such sequences and observing that it is unbounded, we formulate the following conjecture: for *any* sequence $\{t_k\}_{k\geq 0}$, in IGA, such that $t_{k_0} > 1$ for some k_0, it is not possible to bound $f(x_N) - f_*$ in general. Of course, this constitutes numerical evidence and not a formal proof. The conjecture could be proved by constructing worst-case functions in the same spirit as in Chapter 3, with some pathological lack of smoothness that would cause the iterates to diverge when taking a step size larger than $1/L$.

These experiments lead us to believe that accelerated Bregman methods with non-adaptive coefficients fail to converge in the general relatively-smooth setting.

4.4.4 From worst-case functions for BGD to a lower bound for general Bregman methods

We briefly explain how, with the PEP methodology, the worst case functions from Chapter 3 were discovered.

We described in the previous section how a one-dimensional worst-case instance $(\overline{f}, \overline{h})$ for BGD was discovered from low-rank solutions of (sdp-$\overline{\text{PEP}}$). However, this instance may not be difficult enough for a more generic Bregman algorithm that can use abritrary linear combinations of gradients, and thus cannot be used to prove a general lower bound.

Our objective now is to find worst-case instances that are difficult for **any** Bregman gradient algorithm. A desirable property would be that these instances allow to explore only *one dimension* per oracle call, so that the function *hides information* in the unexplored dimensions. This is similar in spirit to the so-called “worst function in the world" of Nesterov [2003]. In order to achieve this goal, we propose to search for functions f for which all gradients $\nabla f(x_i)$ are orthogonal, guaranteeing that one new dimension is explored at each step. Note that a similar approach has been used in some previous work on PEPs to find lower bounds or optimal methods e.g., in Drori [2017], Drori and Taylor [2019]. This amounts to adding some orthogonality constraints to (PEP) and solving

$$\begin{array}{ll} \text{maximize} & \big(f(x_N) - f(x_*)\big)/D_h(x_*, x_0) \\ \text{subject to} & (f,h) \in \mathcal{B}_L(\mathbb{R}^d), \\ & x_* \text{ is a minimizer of } f, \\ & x_1, \ldots, x_N \text{ are generated from } x_0 \text{ by BGD with step size } \lambda, \\ & \langle \nabla f(x_i), \nabla f(x_j) \rangle = 0 \text{ for } i \neq j \in I, \end{array} \tag{PEP-orth}$$

in the variables $f, h, x_0, \ldots, x_N, x_*, d$. In the same spirit as before, we were able to find a dimension-N solution of (PEP-orth). This allows us to interpolate the following worst-case pathological instance:

$$\begin{aligned} \overline{f}(x) &= \|x - (1, \ldots, 1)\|_\infty, \\ \overline{h}(x) &= \overline{f}(x) + \sum_{i=2}^{N} \max(-x^{(i)}, 0). \end{aligned}$$

Again, these are nonsmooth functions and, as such, they do not form valid instances for BGD. However, they can be approached by a sequence of such functions, for instance by applying smoothing with the Moreau evelope, and adding a small quadratic term to make h strictly convex. Along with a few tweaks, this is how we found the worst-case instance that was used to prove the general lower bound for $\mathcal{B}_L$ in Chapter 3.

4.5 Conclusion

In this chapter, we showed how the worst-case complexity of Bregman gradient methods with generic kernel can be computed with the aid of performance estimation problems. To this end, we established interpolation condition for the class of differentiable and strictly convex functions. As we showcased on several examples, PEPs allow to conjecture and prove convergence rates in various situations.

In the setting of Bregman methods, a fundamental concept arising from this work is that of *limiting nonsmooth pathological behavior*. When looking for worst-case guarantees over a class of functions that is open such as the class of differentiable convex functions, the performance estimation problem is a *supremum* and the worst-case maximizing *sequence* might approach some function that is not in this class, e.g., one that is nonsmooth in our case. This idea, observed by analyzing the equivalence between (PEP) and the nonsmooth relaxation ($\overline{\text{PEP}}$), was used in the proof of the lower bound in Chapter 3. Moreover, the worst-case sequence of functions was directly inspired by examining particular solutions of ($\overline{\text{PEP}}$).

Code. Experiments have been run in MATLAB, using the semidefinite solver MOSEK Mosek [2019] as well as the modeling toolbox YALMIP [Lofberg, 2004]. The support for Bregman methods has been added to the Performance Estimation Toolbox (PESTO, Taylor et al. [2018]) for which we provide some examples. The code can be downloaded from the repository https://github.com/RaduAlexandruDragomir/BregmanPerformanceEstimation

Appendix: extension of performance estimation to the case when $\mathcal{C}$ is a general closed convex subset of $\mathbb{R}^d$

For simplicity of the presentation, we left out in Section 4.3 the case when the domain $\mathcal{C}$ is a proper subset of $\mathbb{R}^d$. We show in this section that it actually corresponds to the same minimization problem (sdp-$\overline{\text{PEP}}$).

Let us formulate the performance estimation problem for Bregman gradient descent in the general case. Recall that we denote $\mathcal{B}_L$ the union of $\mathcal{B}_L(\mathcal{C})$ for all closed convex subsets of $\mathbb{R}^d$ and for every $d \geq 1$. The performance estimation problem writes

$$\begin{array}{ll} \text{maximize} & \big(f(x_N) - f(x_*)\big)/D_h(x_*, x_0) \\ \text{subject to} & (f,h) \in \mathcal{B}_L, \\ & x_* \text{ is a minimizer of } f \text{ on } \overline{\operatorname{dom} h} \text{ such that } x \in \operatorname{dom} h, \\ & x_1, \dots, x_N \text{ are generated from } x_0 \text{ by BGD with step size } \lambda, \end{array} \tag{PEP-C}$$

in the variables $f, h, x_0, \dots, x_N, x_*, d$. Now, as (PEP-C) is a problem that includes (PEP) in the special case where $C = \mathbb{R}^d$, its value is larger:

$$\operatorname{val}(\text{PEP}) \leq \operatorname{val}(\text{PEP-C})$$

Let us show that val(PEP-C) is upper bounded by the same relaxation val($\overline{\text{PEP}}$), which allows to conclude that the values are equal. We recall that the problem ($\overline{\text{PEP}}$) can be written, using interpolation conditions of Corollary 4.6, as

$$\begin{array}{ll} \text{maximize} & f_N - f_* \\ \text{subject to} & f_i - f_j - \langle g_j, x_i - x_j \rangle \geq 0, \\ & (Lh_i - f_i) - (Lh_j - f_j) - \langle Ls_j - g_j, x_i - x_j \rangle \geq 0 \quad \text{ for } i,j \in I, \\ & g_* = 0, \\ & s_{i+1} = s_i - \lambda g_i \quad \text{for } i \in \{1, \dots, N-1\}, \\ & h_* - h_0 - \langle s_0, x_* - x_0 \rangle = 1, \end{array}$$

in the variables $n, \{(x_i, f_i, g_i, h_i, s_i)\}_{i \in I}$. We show that every admissible point of (PEP-C) can be cast into an admissible point of (sdp-$\overline{\text{PEP}}$). This actually amounts to show that, from the point of view of performance estimation, an instance $(f,h) \in \mathcal{B}_L(\mathcal{C})$ is actually equivalent to some instance in $\mathcal{B}_L(\mathbb{R}^d)$.

Let $f, h, x_0, \dots, x_N, x_*, d$ be a feasible point of (PEP-C). We distinguish two cases.

Case 1: $x_* \in \operatorname{\mathbf{int\,dom}} h$. This is the simplest case, as the necessary conditions are the same as in the situation where $\mathcal{C} = \mathbb{R}^d$. Indeed, then we have $x_0, \dots, x_N, x_* \in \operatorname{int\,dom} h$, since x_0 is constrained to be in the interior and the next iterates are in $\operatorname{int\,dom} h$ by definition of a kernel function. Since f and h are differentiable on $\operatorname{int\,dom} h$, convexity of f and $Lh - f$ imply that the first two constraints of ($\overline{\text{PEP}}$) hold for all $i,j \in I$. Finally, $g_* = 0$ follows from the fact that x_* minimizes f and that it lies on the interior of the domain. Hence the discrete representation satisfies the constraints of (sdp-$\overline{\text{PEP}}$).

Case 2: $x_* \in \partial\mathbf{dom}\, h$. In this case, f and h are not necessarily differentiable at x_*, but are still differentiable still at $x_0, \dots, x_N$ for the same reasons. But we can still, with a small modification at x_*, derive a discrete representation that fits the constraints of ($\overline{\text{PEP}}$) and whose objective is the same. Indeed, define

$$
\begin{aligned}
(g_i, f_i, s_i, h_i) &= (\nabla f(x_i), f(x_i), \nabla h(x_i), h(x_i)) \text{ for } i = 0, \dots, N, \\
(g_*, f_*, s_*, h_*) &= (0, f(x_*), v, h(x_*)),
\end{aligned}
$$

where $v \in \mathbb{R}^d$ is a vector that are specified later. Then, for $i \in I$ and $j \in \{0 \dots N\}$, convexity of f and $Lh - f$ imply that the constraints

$$
\begin{aligned}
f_i - f_j - \langle g_j, x_i - x_j \rangle &\geq 0 \\
(Lh_i - f_i) - (Lh_j - f_j) - \langle Ls_j - g_j, x_i - x_j \rangle &\geq 0
\end{aligned}
$$

hold. It remains to verify them for $i \in \{0 \dots N\}$ and $j = *$. The first one holds because x_* minimizes f on $\operatorname{dom} h$, so with $g_* = 0$ we have $f_i - f_* \geq 0$. We now show that the second one is satisfied, i.e., that we can choose $v \in \mathbb{R}^d$ so that

$$
(Lh_i - f_i) - (Lh_* - f_*) - \langle Lv, x_i - x_* \rangle \geq 0 \quad \forall i \in \{0 \dots N\}.
$$

To this extent, we use the fact that $x_* \in \partial \operatorname{dom} h$ and that $x_i \in \operatorname{int} \operatorname{dom} h$ for $i = 0 \dots N$. This means that $\{x_*\} \cap \operatorname{int} \operatorname{dom} h = \emptyset$, and therefore by the hyperplane separation theorem [Rockafellar, 1970, Thm 11.3], there exists a hyperplane that separates the convex sets $\{x_*\}$ and $\operatorname{int} \operatorname{dom} h$ *properly*, meaning that there exists a vector $u \in \mathbb{R}^d$ such that

$$
\langle x_i - x_*, u \rangle < 0 \;\; \forall i \in \{0, \dots, N\}.
$$

Set

$$
\begin{aligned}
\alpha &= \min_{i=0\dots N} (Lh_i - f_i) - (Lh_* - f_*), \\
\beta &= \min_{i=0,\dots,N} -\langle x_i - x_*, u \rangle > 0,
\end{aligned}
$$

where $\beta > 0$ because of the separation result. Choose $s_* = v$ as $v = \frac{|\alpha|}{L\beta} u$. Then we have

$$
\begin{aligned}
(Lh_i - f_i) - (Lh_* - f_*) - \langle Ls_*, x_i - x_* \rangle &\geq \alpha + L\frac{|\alpha|}{L\beta}\beta \\
&\geq \alpha + |\alpha| \\
&\geq 0.
\end{aligned}
$$

This eventually provides an instance $\{(x_i, g_i, f_i, h_i, s_i)\}_{i \in I}$ that is admissible for ($\overline{\text{PEP}}$).

To conclude, we proved that in both cases, an admissible point of (PEP-C) can be turned into an admissible point of (sdp-$\overline{\text{PEP}}$) with the same objective value. Hence we have

$$
\operatorname{val}(\text{PEP-C}) \leq \operatorname{val}(\text{sdp-}\overline{\text{PEP}}).
$$

Recalling that val(PEP) $\leq$ val(PEP-C) and that val(sdp-$\overline{\text{PEP}}$) $=$ val(PEP) by Theorem 4.8, we get

$$
\operatorname{val}(\text{PEP-C}) = \operatorname{val}(\text{PEP}).
$$

In other words, solving the performance estimation problem (PEP-C) for functions with any closed convex domain is equivalent to solving the performance estimation problem (PEP) restricted to functions that have full domain.

Chapter 5

Computer-Aided Analyses of Entropic-Smooth Minimization Methods

Chapter Abstract

In the previous chapter, we showed that the problem of finding the worst-case behavior of Bregman methods for *generic kernel* could be formulated as a semidefinite optimization problem. This lead to a lower bound proving the impossibility of acceleration, relying on a pathological worst-case kernel, and demonstrated the necessity of making additional regularity assumptions in order to devise faster algorithms.

In this work, we propose to tackle a more restricted setting, by focusing on the entropic kernel. We show that finding the worst-case behavior of Bregman methods on functions that are convex and smooth relative to the entropy can be formulated as a finite dimensional convex program. This problem involves a new set called the *Kullback-Leibler cone with log-linear constraints*, for which no solver is currently available. However, we manage to solve small instances with heuristic methods and provide some conjectures based on preliminary numerical results.

Collaboration: the content presented in this chapter has not yet been published. Part of this work has been done in collaboration with Dmitrii Ostrovskii.

5.1 Introduction

As in the previous chapter, we consider the minimization problem

$$\min_{x\in\mathcal{C}} f(x)$$

where $\mathcal{C}$ is a convex subset of $\mathbb{R}^d$ and f a continuously differentiable convex function which is smooth relative to some convex kernel h. The standard method for solving such problem is the

Bregman gradient descent scheme (BGD), which writes

$$x_{k+1} = \operatorname*{argmin}_{u \in \mathcal{C}} f(x_k) + \langle \nabla f(x_k), u - x_k \rangle + \frac{1}{\lambda} D_h(u, x_k), \tag{BGD}$$

where

$$D_h(x, y) = h(x) - h(y) - \langle \nabla h(y), x - y \rangle$$

is the Bregman divergence induced by h. An important question, among others, is to determine if there exists an algorithm that achieves a better convergence rate than (BGD) on the same class of functions.

In the previous chapters, we gave a rather negative answer to this question for general kernel h. Indeed, we proved in Chapter 3 a lower bound implying that BGD is optimal among a large class of Bregman-type methods. The worst-case instance involved in this lower bound was discovered numerically by solving a *performance estimation problem* (PEP), which we described in Chapter 4. PEPs allow to formulate the problem of finding the worst-case complexity of gradient methods as numerical optimization problems. They form a great tool for conjecturing and proving new results in first-order optimization [Drori and Teboulle, 2014, Taylor et al., 2015, Kim and Fessler, 2016, Drori and Taylor, 2019, Taylor and Bach, 2019].

A critical limitation of the results from the previous chapters is that we consider generic kernels, that is, we search for the worst possible couple (f, h) such that f is convex and smooth relative to h. By doing so, the worst case kernels that we find are pathological and somewhat unrealistic. For instance, the proof of the lower bound in Chapter 3 involves a *sequence of functions* $\{h_\mu\}_{\mu \geq 0}$ which approaches a nondifferentiable convex function as $\mu \to 0$. In practical applications of relative smoothness, h is more regular, as is it usually a C^∞ function. This is why, in this chapter, we propose to look at the particular case of the *discrete entropy*

$$h_e(x) = \sum_{i=1}^{d} x^{(i)} \log x^{(i)} - x^{(i)},$$

on the set $\mathcal{C} = \mathbb{R}_+^d$, for which the corresponding Bregman divergence is the well-known Kullback-Leibler divergence, and the update of (BGD) writes

$$x_{k+1} = x_k \circ \exp\left(-\lambda \nabla f\left(x_k\right)\right).$$

Therefore, we consider the class of functions f that are convex and smooth relative to the entropy, which we call the class of *entropic-smooth* convex functions. We focus on the entropic kernel for two main reasons. First, it is one of the most common example of non-Euclidean kernels, with applications such as optimal transport and nonnegative entropy regression (see Chapter 1). Secondly, the entropy has some favorable properties which are crucial for our analysis, including the fact that the corresponding Bregman divergence is jointly convex [Bauschke and Borwein, 2001]. Moreover, studying the entropic geometry might also help gain insight about other kernels with similar structure on the nonnegative orthant, such as the log-barrier.

In this chapter, we consider only the unnormalized case where $\mathcal{C} = \mathbb{R}_+^d$; the setting where $\mathcal{C}$ is the unit simplex $\{x \in \mathbb{R}_+^d : \sum_{i=1}^d x^{(i)} = 1\}$ is left for future work as it poses additional difficulties for performance estimation.

Contributions and outline. We propose to apply the performance estimation framework of Drori and Teboulle [2014] for finding the worst-case behavior of Bregman methods on entropic-smooth convex functions.

We start by recalling in Section 5.2 the basic setup and defining the performance estimation problem (PEP). In Section 5.3, we establish the *interpolation conditions* which are needed for formulating the PEP as an equivalent finite-dimensional optimization problem. To do so, we generalize the Euclidean interpolation conditions of Taylor et al. [2017] to the entropic-smooth setting. We rely on the smoothing properties of the Bregman-Moreau envelope [Kan and Song, 2012, Bauschke et al., 2018, Laude et al., 2020], which is the Bregman counterpart of the standard Moreau envelope [Moreau, 1965]. Although part of our analysis can be done for a more general class of kernels, we need additional restrictive assumptions to reach our claims (such as joint convexity of the Bregman divergence), which are satisfied by the entropy.

Then, in Section 5.4, we show that the PEP can be written as a convex problem on the set of matrices of pairwise Kullback-Leibler divergences, which we call the *Kullback-Leibler* cone, with additional log-linear constraints. Unfortunately, we are not aware of an explicit description of such set, or any solvers for this type of problem. We are only able to describe the set in the simple case without log-linear constraints: in that case, we show that the set is *trivial*, as any nonnegative matrix with null diagonal can be approached by a matrix of pairwise Kullback-Leibler divergences.

Finally, we show in Section 5.5 some preliminary numerical experiments, made by solving the PEP with basic heuristics. This leads to the following pessimistic conjecture: among the class of linear-span Bregman methods with *fixed coefficients*, only vanilla BGD is guaranteed to converge in the worst-case.

Notation. Throughout this chapter, we use the notation h for a generic convex kernel function, while h_e denotes the entropic kernel

$$h_e(x) = \sum_{i=1}^{d} x^{(i)} \log x^{(i)} - x^{(i)}$$

defined for $x \in \mathbb{R}^d_+$, with the convention that $0 \log 0 = 0$. The corresponding Bregman divergence is the *Kullback-Leibler divergence*

$$D_{h_e}(x, y) = \sum_{i=1}^{d} x^{(i)} \log \frac{x^{(i)}}{y^{(i)}} - x^{(i)} + y^{(i)}$$

defined for $x \in \mathbb{R}^d_+, y \in \mathbb{R}^d_{++}$. We write $\mathcal{F}^h_L$ the class of convex functions that are L-smooth relative to h.

$\mathbf{1}_m$ denotes the vector $(1, \ldots, 1) \in \mathbb{R}^m$. $\mathrm{Co}(A)$ is the convex cone spanned by the elements of the set A. For $x, y \in \mathbb{R}^d$, $x \circ y$ denotes the componentwise product of x and y. For a scalar function $\phi : \mathbb{R} \to \mathbb{R}$ and a vector $x \in \mathbb{R}^d$, $\phi(x)$ denotes the result of applying ϕ componentwise to the vector x, i.e., $\phi(x) = \big(\phi(x^{(1)}), \ldots, \phi(x^{(d)})\big)$. Let $X \in \mathbb{R}^{d \times m}$ a matrix, and denote $X = [x_1; \ldots; x_m]$ its column vectors. We define the pairwise divergence matrix $D_h[X, X] \in \mathbb{R}^{m \times m}$ as

$$(D_h[X, X])_{ij} = D_h(x_i, x_j), \quad i, j = 1 \ldots m.$$

5.2 Problem setup

5.2.1 Relatively-smooth optimization

We briefly recall the basic ingredients and definitions for relatively-smooth optimization problems. For a general introduction, we refer the reader to Chapter 1.

Definition 5.1 (Kernel function). *A function $h : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is called a kernel function on $\mathcal{C}$ if*

(i) h is closed convex proper (c.c.p.),

(ii) h is continuously differentiable and strictly convex on $\operatorname{int} \mathcal{C}$,

(iii) the Bregman gradient iterates are well-posed, i.e., for every $p \in \mathbb{R}^d$, the problem

$$\min_{u \in C} \langle p, u \rangle + h(u)$$

has a unique minimizer, which belongs to $\operatorname{int} \mathcal{C}$.

The entropy h_e is a particular case of kernel on $\mathcal{C} = \mathbb{R}^d_+$. We also recall the central notion of relative smoothness [Bauschke et al., 2017].

Definition 5.2 (Relative smoothness). *Let h be a kernel function on $\mathcal{C}$, and f a function such that $\operatorname{dom} h \subset \operatorname{dom} f$. We say that f is smooth relative to h if it is differentiable on $\operatorname{int} \mathcal{C}$ and if there exists a constant $L > 0$ such that*

$$Lh - f \quad \text{is convex on} \operatorname{int} \mathcal{C},$$

or equivalently, if

$$f(x) \leq f(y) + \langle \nabla f(y), x - y \rangle + L D_h(x, y) \quad \forall x, y \in \operatorname{int} \mathcal{C}. \tag{5.1}$$

For a given kernel h, we write

$$f \in \mathcal{F}^h_L$$

if the function f is convex and L-smooth relative to h.

Entropic smoothness. When a function f is smooth relative to the entropy h_e, we say that f is *entropic-smooth*. The canonical example of convex entropic-smooth functions is given by functions of the form

$$f_{A,b}(x) = D_{h_e}(Ax, b)$$

for some matrix $A \in \mathbb{R}^{p \times d}_+$ and vector $b \in \mathbb{R}^p_{++}$. The function $f_{A,b}$ is entropic-smooth with constant $L = \max_j \sum_i A_{ij}$ [Bauschke et al., 2017]. Such a function arises in nonnegative entropy regression and entropy-regularized optimal transport [Chizat et al., 2018, Mishchenko, 2019].

Convex conjugate. Recall that if h is a kernel function, we define its convex conjugate h^* as

$$h^*(y) = \sup_{u\in\mathbb{R}^d} \langle u, y\rangle - h(u).$$

By Definition 5.1 (iii), h^* is finite-valued on $\mathbb{R}^d$. Moreover, its gradient satisfies for every $u \in \mathbb{R}^d$ [Rockafellar, 1970, Sect. 26]

$$\nabla h^*(y) = \operatorname*{argmax}_{u\in\mathbb{R}^d} \langle u, y\rangle - h(u).$$

We recall the relationship between the Bregman divergence of h and its conjugate; see [Bauschke and Borwein, 1997, Thm. 3.7] for the proof.

Lemma 5.3 (Bregman divergence of conjugate)**.** *Let h be a kernel function on $\mathcal{C}$. Then for every $x, y \in \operatorname{int}\mathcal{C}$, we have*

$$D_h(x, y) = D_{h^*}\left(\nabla h(y), \nabla h(x)\right).$$

The conjugate of the entropy function is the exponential kernel: for $u \in \mathbb{R}^d$ we have

$$h_e^*(u) = \sum_{i=1}^{d} \exp(u^{(i)}).$$

Let us check that Lemma 5.3 holds for this example. Let $x, y \in \mathbb{R}^d_{++}$ and define

$$\begin{aligned} u &= \nabla h_e(x) = \log(x), \\ v &= \nabla h_e(y) = \log(y). \end{aligned}$$

Then we have

$$\begin{aligned} D_{h_e}(x, y) &= \sum_{i=1}^{d} x^{(i)} \log\frac{x^{(i)}}{y^{(i)}} - x^{(i)} + y^{(i)} \\ &= \sum_{i=1}^{d} \exp(v^{(i)}) - \exp(u^{(i)})\left(1 + v^{(i)} - u^{(i)}\right) \\ &= D_{h_e^*}(v, u). \end{aligned} \tag{5.2}$$

5.2.2 Performance estimation problems for Bregman methods with fixed kernel

In this section, we formulate the problem of finding the worst-case behavior of a Bregman first-order method on the class $\mathcal{F}_L^h$. The fundamental difference with the approach from Chapter 4 is that we now look for the worst case for a *given* kernel h. In the sequel, we will particularly focus on the entropy h_e.

Let h be a kernel function, $N \geq 1$ be a number of iterations, and $\mathcal{A}$ a Bregman first-order method. We wish to compute the worst-case performance of $\mathcal{A}$ on the class $\mathcal{F}_L^h$ after N iterations. Since performance guarantees in the Bregman setting have the form

$$f(x_N) - f(x_*) \leq \theta(N, L) D_h(x_*, x_0),$$

for some function θ, it is natural to seek the worst-case performance as the maximal value of the ratio

$$\big(f(x_N) - f_*\big)/D_h(u, x_0)$$

over the problem class $\mathcal{F}_L^h$. To this extent, we propose to solve the minimization problem, for an initial radius $R > 0$,

$$\begin{aligned} \text{maximize} \quad & f(x_N) - f(x_*) \\ \text{subject to} \quad & D_h(x_*, x_0) = R \\ & f \in \mathcal{F}_L^h, \\ & x_N \text{ is the output of algorithm } \mathcal{A} \text{ initialized at } x_0 \text{ after } N \text{ iterations,} \end{aligned} \tag{PEP}$$

in the variables f, x_0, x_N, x_*. We refer to this problem as a Performance Estimation Problem (PEP). Note that we do not need impose that x_* minimizes f, this will occur naturally in the solutions of (PEP) as to maximize the gap $f(x_N) - f(x_*)$. Although it seems that (PEP) should be solved for every value of R, we will use homogeneity arguments in the sequel to show that it suffices to consider $R = 1$.

Solving (PEP) seems intractable, as it involves the infinite-dimensional variable f. Following Drori and Teboulle [2014], Taylor et al. [2017], we proceed to reformulate it as a finite dimensional problem. Since we assume that the algorithm $\mathcal{A}$ is a first-order method and we adopt the black box model, it can gain information on f only through the first-order oracle $(f, \nabla f)$ at query points $x_0, \ldots, x_N$. Thus we formally write

$$x_N = \mathcal{A}\left(x_0, \{(f(x_i), \nabla f(x_i))\}_{i=0}^{N-1}\right)$$

and, introducing the variables $f_i = f(x_i), g_i = \nabla f(x_i)$ for i in the index set $I = \{0, \ldots, N, *\}$, Problem (PEP) is equivalent to

$$\begin{aligned} \text{maximize} \quad & f_N - f_* \\ \text{subject to} \quad & D_h(x_*, x_0) = R, \\ & f_i = f(x_i), g_i = \nabla f(x_i) \text{ for all } i \in I \text{ and some } f \in \mathcal{F}_L^h, \\ & x_N = \mathcal{A}\left(x_0, \{(f_i, g_i)\}_{i=0}^{N-1}\right), \end{aligned} \tag{PEP-disc}$$

in the variables $\{(x_i, f_i, g_i)\}_{i \in I}$. We now deal with a finite-dimensional optimization problem, the equivalence with the previous PEP being guaranteed by the so-called **interpolation constraints**. We introduce the following convenient definition for interpolable sets [Taylor et al., 2017].

Definition 5.4. *Let I be a finite index set and $S = \{(x_i, f_i, g_i)\}_{i \in I} \in (\mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^d)^{|I|}$ for some dimension d. Let $\mathcal{F}$ be a class of convex functions. We say that the set S is $\mathcal{F}$-interpolable if there exists a function $f \in \mathcal{F}$ such that*

$$f_i = f(x_i),\ g_i \in \partial f(x_i) \quad \forall i \in I.$$

Hence, the second line in (PEP-disc) amounts to requiring that the set $\{(x_i, f_i, g_i)\}_{i \in I}$ is $\mathcal{F}_L^h$-interpolable. We now proceed to establish necessary and sufficient conditions in the case of the entropic kernel.

5.3 Entropic-smooth convex interpolation

In the Euclidean case, Taylor et al. [2017] provide sufficient and necessary conditions for a finite set $\{(x_i, f_i, g_i)\}_{i\in I}$ to be interpolated by a L-smooth strongly convex function (Theorem 4.4). Our aim is to generalize this approach and to provide interpolation conditions for the class $\mathcal{F}_L^h$ of functions that are smooth relative to a given kernel h (recall that, contrary to the previous chapter where we considered interpolation conditions with generic kernels, h is now considered to be fixed).

It turns out that this task requires additional assumptions on h, as we show in the sequel. While necessary interpolation conditions can be derived for any kernel, sufficient conditions require joint convexity of the Bregman divergence $D_h(\cdot,\cdot)$, which is a rather restrictive assumption. Fortunately, it is valid for the important case of the entropy [Bauschke and Borwein, 2001]. Finally, we show that the entropy satisfies a favorable algebraic property which allows to prove the equivalence between the necessary and the sufficient conditions.

5.3.1 Necessary conditions using Bregman co-coercivity

We start by establishing necessary conditions for interpolability. They are based on the following consequence of relative smoothness, which we call *Bregman co-coercivity*. It generalizes a known inequality for L-smooth convex functions [Nesterov, 2003, Eqn. (2.1.7)].

Proposition 5.5. *Let h be a kernel function on $\mathcal{C}$ and f a convex functions such that f is L-smooth relative to h. Then for every $x, y \in \operatorname{int}\mathcal{C}$ we have*

$$D_f(x,y) \geq LD_{h^*}\left(\nabla h(x) - \frac{1}{L}(\nabla f(x) - \nabla f(y)), \nabla h(x)\right).$$

Proof. Let $y \in \operatorname{int}\mathcal{C}$. Consider the function ϕ_y defined for $x \in \operatorname{int}\mathcal{C}$ by

$$\phi_y(x) = D_f(x,y).$$

Since ϕ_y is built by adding a linear function to f, it is also convex and L-smooth relative to h. Thus, we can apply the descent inequality (5.1) which writes for $u, x \in \mathcal{C}$,

$$\phi_y(u) \leq \phi_y(x) + \langle \nabla\phi_y(x), u - x\rangle + LD_h(u,x) := Q_{x,y}(u) \tag{5.3}$$

The right-hand side $Q_{x,y}(u)$ is a strictly convex function which is minimized for $\hat{u}$ such that

$$\frac{1}{L}\nabla\phi_y(x) + \nabla h(\hat{u}) - \nabla h(x) = 0, \tag{5.4}$$

Thus, using (5.4) to substitute $\nabla\phi_y(x)$ in (5.3) yields

$$\begin{aligned}\phi_y(\hat{u}) &\leq Q_{x,y}(\hat{u})\\ &= \phi_y(x) + L\langle \nabla h(x) - \nabla h(\hat{u}), u - x\rangle + LD_h(\hat{u}, x)\\ &= \phi_y(x) - LD_h(x,\hat{u})\end{aligned}$$

Recalling that $0 \leq \phi_x$ by convexity of f and using Lemma 5.3 we get

$$\begin{aligned}0 &\leq \phi_y(x) - LD_h(x,\hat{u})\\ &= D_f(x,y) - LD_{h^*}\left(\nabla h(\hat{u}), \nabla h(x)\right)\\ &= D_f(x,y) - LD_{h^*}\left(\nabla h(x) - \frac{1}{L}\nabla\phi_y(x), \nabla h(x)\right)\end{aligned}$$

which yields the desired result as $\nabla\phi_y(x) = \nabla f(x) - \nabla f(y)$. ■

As a consequence, we deduce necessary conditions for $\mathcal{F}_L^h$-interpolability.

Proposition 5.6 (Necessary interpolation conditions)**.** *Let h be a kernel on $\mathcal{C}$. Let I be a finite index set and $S = \{(x_i, f_i, g_i)\}_{i\in I} \in (\operatorname{int}\mathcal{C} \times \mathbb{R} \times \mathbb{R}^d)^{|I|}$. If the set S is $\mathcal{F}_L^h$-interpolable, then*

$$f_i - f_j - \langle g_j, x_i - x_j\rangle \geq L D_{h^*}\left(\nabla h(x_i) - \frac{1}{L}(g_i - g_j), \nabla h(x_i)\right) \quad \forall i,j \in I.$$

Proof. This simply amounts to choosing an interpolating function $f \in \mathcal{F}_L^h$ and applying the Bregman co-coercivity property (Proposition 5.5) at pairs of points (x_i, x_j). ■

5.3.2 Sufficient conditions with Bregman-Moreau envelopes

To establish sufficient conditions, we need a constructive procedure for building a function in $\mathcal{F}_L^h$ which interpolates the finite set $\{(x_i, f_i, g_i)\}_{i\in I}$.

In the Euclidean setting, Taylor et al. [2017] proceed by first building a nonsmooth convex function as a supremum of affine functions, and then *smoothing* by taking its Moreau envelope [Moreau, 1965][1]. Recall that the (Euclidean) Moreau envelope of a convex function $\hat{f}$ with parameter $\mu > 0$ is given by

$$\operatorname{env}_\mu \hat{f}(x) = \min_{u\in\mathbb{R}^d} \hat{f}(u) + \frac{1}{2\mu}\|u - x\|^2. \tag{5.5}$$

If $\hat{f}$ is closed convex and proper, then $\operatorname{env}_\mu \hat{f}$ is convex and μ^{-1}-smooth [Bauschke and Combettes, 2011, Sect. 12.4]. Thus, the Moreau envelope posses a *smoothing* property, and the level of smoothing is controlled by the coefficient μ.

Bregman-Moreau envelopes. It is natural to wonder if by replacing the squared Euclidean distance with a Bregman divergence in (5.5), relatively-smooth functions can be built based on the same principle. Note that there are two ways to perform this extension, as the Bregman divergence D_h is not symmetric in general. In this work, we show that, in order to reach the desired property, we need to consider the *right* Bregman-Moreau envelope defined as

$$\operatorname{env}_\mu^h \hat{f}(x) = \min_{u\in\operatorname{int}\mathcal{C}} \hat{f}(u) + \frac{1}{\mu} D_h(x, u), \tag{5.6}$$

Historically, the variant that has been considered first is the *left* Bregman-Moreau envelope which writes $\min_{u\in\mathbb{R}^d} \hat{f}(u) + \mu^{-1} D_h(u, x)$, because it is the operation that naturally appears in Bregman proximal methods [Kan and Song, 2012]. The idea of considering the *right* Bregman-Moreau envelope appeared first in Bauschke [2006], and its smoothing properties were studied in Bauschke et al. [2018]; see also Laude et al. [2020] for a recent extension to nonconvex functions $\hat{f}$.

The study of the right Bregman-Moreau envelope requires additional structural assumptions on h; in particular, we need to assume that $D_h(\cdot,\cdot)$ is jointly convex so that $\operatorname{env}_\mu^h \hat{f}$ is well defined.

[1] Actually, the procedure described by Taylor et al. [2017] does not mention explicitly using a Moreau envelope. They rather propose to build a smooth convex function by adding a quadratic to the conjugate of a nonsmooth convex function, and taking the conjugate back. One can show that this process is equivalent to applying the Moreau envelope, by exploiting the relationship between conjugacy and inf-convolution [Bauschke and Combettes, 2011, Prop. 13.21].

Assumption 5.1. *The function $h : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ satisfies the following properties:*

(i) h is convex, proper and closed,

(ii) h is twice continuously differentiable on $\operatorname{int} \mathcal{C}$,

(iii) h is of Legendre type , that is, h is strictly convex on $\operatorname{int} \mathcal{C}$ and such that $\|\nabla h(x_k)\| \to +\infty$ for every sequence $\{x_k\}_{k\geq 0} \subset \operatorname{int} \mathcal{C}$ that converges to a boundary point of $\mathcal{C}$ [Rockafellar, 1970, Sect 26],

(iv) the Bregman divergence $D_h(\cdot, \cdot)$ is jointly convex on $\mathbb{R}^d \times \mathbb{R}^d$,

(v) for every $x \in \operatorname{int} \mathcal{C}$, the function $D_h(x, \cdot)$ is strictly convex on $\mathcal{C}$ and coercive.

Among these conditions, item (iv) is particularly restrictive; fortunately, besides the squared Euclidean norm, it is also verified for the entropic kernel h_e [Bauschke and Borwein, 2001]. All other conditions from Assumption 5.1 also hold for h_e.

Under these assumptions, we can now state the properties of the right Bregman-Moreau envelope studied by Bauschke et al. [2018].

Proposition 5.7. *(Relative smoothness of the right Bregman-Moreau envelope) Assume that $\hat{f}$ is a finite-valued convex function on $\mathbb{R}^d$ which is bounded from below. Let h be a kernel function on $\mathcal{C}$ which satisfies Assumption 5.1. Then, for $\mu > 0$, the right Bregman-Moreau envelope $\operatorname{env}_\mu^h \hat{f}$ defined in (5.6) satisfies the following properties:*

(i) $\operatorname{env}_\mu^h \hat{f}$ is convex on $\mathbb{R}^d$,

(ii) $\operatorname{env}_\mu^h \hat{f}$ is differentiable on $\operatorname{int} \mathcal{C}$ and $1/\mu$-smooth relative to h.

In other words, we have

$$\operatorname{env}_\mu^h \hat{f} \in \mathcal{F}_L^h$$

with $L = 1/\mu$.

Proof. Point (i) is a consequence of [Bauschke et al., 2018, Prop. 2.7 (ii)], and differentiability on $\operatorname{int} \mathcal{C}$ follows from [Bauschke et al., 2018, Prop. 2.19 (ii)]. Finally, by [Bauschke et al., 2018, Prop. 2.4 (ii)] we have

$$\frac{1}{\mu} h - \operatorname{env}_\mu^h \hat{f} = \frac{1}{\mu}\left(h^* + (\mu \hat{f} \circ \nabla h)\right)^*,$$

which shows that $\mu^{-1} h - \operatorname{env}_\mu^h \hat{f}$ is a convex function, since the conjugate of any function is convex [Bauschke and Combettes, 2011, Prop. 13.11], and thus that relative smoothness holds. ■

Let also define the right Bregman-Moreau proximal map for $x \in \mathbb{R}^d$:

$$\operatorname{prox}_\mu^h \hat{f}(x) = \operatorname*{argmin}_{u \in \operatorname{int} \mathcal{C}} \hat{f}(u) + \frac{1}{\mu} D_h(x, u)$$

which we will often write $u(x)$ for shortness when the context is clear. The next proposition caracterizes the Breman-Moreau proximal map and establishes the link with the gradient of the envelope.

Proposition 5.8. *Let $x \in \operatorname{int} \mathcal{C}$. Under the same assumptions as Proposition 5.7, the proximal map $u(x) = \operatorname{prox}^h_\mu \hat{f}(x)$ satisfies*

$$\begin{aligned} \mu\, \partial \hat{f}(u(x)) &\ni \nabla^2 h(u(x))\,(x - u(x))\,, \\ \mu\, \nabla \operatorname{env}^h_\mu \hat{f}(x) &= \nabla h(x) - \nabla h(u(x)). \end{aligned} \tag{5.7}$$

Proof. The first equation follows from the first-order optimality conditions defining $u(x)$, and the second is a consequence of [Bauschke et al., 2018, Prop. 2.19 (ii)]. ■

Using the definition of $\operatorname{env}^h_\mu \hat{f}$ and the proposition above, we can note that for $u(x) = \operatorname{prox}^h_\mu \hat{f}(x)$ we have

$$\begin{aligned} \operatorname{env}^h_\mu \hat{f}(x) &= \hat{f}(u(x)) + \frac{1}{\mu} D_h\,(x, u(x))\,, \\ \mu\, \nabla \operatorname{env}^h_\mu \hat{f}(x) &= \nabla_1 D_h\,(x, u(x))\,, \\ \mu\, \partial \hat{f}(u(x)) &\ni \nabla_2 D_h\,(x, u(x))\,, \end{aligned} \tag{5.8}$$

where ∇_1 and ∇_2 denote the partial gradients in each variable of $D_h(\cdot,\cdot)$.

Double Bregman divergence. As the function D_h is convex on $\mathbb{R}^d \times \mathbb{R}^d$, we can consider its Bregman divergence D_{D_h}, which we call the *double Bregman divergence*. For $x, y, u, v \in \operatorname{int} \mathcal{C}$, it is defined by

$$\begin{aligned} D_{D_h}\big((x,y),(u,v)\big) &= D_h(x,y) - D_h(u,v) - \langle \nabla_1 D_h(u,v), x - u\rangle - \langle \nabla_2 D_h(u,v), y - v\rangle \\ &= D_h(x,y) - D_h(u,v) - \langle \nabla h(u) - \nabla h(v), x - u\rangle \\ &\quad - \langle \nabla^2 h(v)(v - u), y - v\rangle. \end{aligned} \tag{5.9}$$

The double divergence, which was already studied by Bauschke [2006], will play a central role in the sufficient conditions for $\mathcal{F}^h_L$-interpolability, which we are now ready to state. These conditions rely on a two-stage procedure: first, we reduce the problem of interpolating a relatively-smooth convex function to the problem of interpolating a convex function, by the means of the Bregman-Moreau envelope. Then, we use the interpolation result for convex functions from Taylor et al. [2017] (which we recalled in Theorem 4.4).

Proposition 5.9 (Sufficient interpolation conditions)**.** *Consider a kernel h satisfying Asssumption 5.1, with zone $\mathcal{C} \subset \mathbb{R}^d$. Let $L > 0$, I be a finite index set and*

$$S = \{(x_i, f_i, g_i)\}_{i \in I} \in (\mathbb{R}^d_{++} \times \mathbb{R} \times \mathbb{R}^d)^{|I|}.$$

Consider the following statements.

(i) For all $i, j \in I$ we have

$$f_i - f_j - \langle g_j, x_i - x_j\rangle \geq L\, D_{D_h}\big((x_i, u_i), (x_j, u_j)\big)$$

with

$$u_i := \nabla h^* \left[\nabla h(x_i) - \frac{1}{L} g_i\right], \ \forall i \in I. \tag{5.10}$$

(ii) The set

$$\hat{S} := \{(u_i, \hat{f}_i, \hat{g}_i)\}_{i \in I}$$

is $\Gamma(\mathbb{R}^d)$-interpolable, where $\Gamma(\mathbb{R}^d)$ denotes the set of convex closed proper functions on $\mathbb{R}^d$ and the set $\hat{S}$ is defined as

$$\begin{aligned} u_i &= \nabla h^* \left[\nabla h(x_i) - \frac{1}{L} g_i \right], \\ \hat{f}_i &= f_i - L D_h(x_i, u_i), \\ \hat{g}_i &= L \nabla^2 h\left[u_i\right](x_i - u_i), \end{aligned} \tag{5.11}$$

for $i \in I$.

(iii) The set S is $\mathcal{F}_L^h$-interpolable.

Then $(i) \implies (ii) \implies (iii)$.

Proof. $(i) \implies (ii)$. Define $\{u_i, \hat{f}_i, \hat{g}_i\}$ as in (5.11). The definition of u_i (5.10) implies that

$$L(\nabla h(x_i) - \nabla h(u_i)) = g_i, \tag{5.12}$$

and hence we have

$$\begin{aligned} \hat{f}_i - \hat{f}_j - \langle \hat{g}_j, u_i - u_j \rangle &\overset{(5.11)}{=} f_i - L D_h(x_i, u_i) - f_j + L D_h(x_j, u_j) - L\langle \nabla^2 h(u_j)(x_j - u_j), u_i - u_j \rangle \\ &= f_i - L D_h(x_i, u_i) - f_j + L D_h(x_j, u_j) - L\langle \nabla^2 h(u_j)(x_j - u_j), u_i - u_j \rangle \\ &\quad - \langle g_j, x_i - x_j \rangle + L\langle \nabla h(x_j) - \nabla h(u_j), x_i - x_j \rangle \\ &\overset{(5.12)}{=} f_i - f_j - \langle g_j, x_i - x_j \rangle - L\Big(D_h(x_i, u_i) - D_h(x_j, u_j) \\ &\quad - \langle \nabla h(x_j) - \nabla h(u_j), x_i - x_j \rangle - \langle \nabla^2 h(u_j)(u_j - x_j), u_i - u_j \rangle \Big) \\ &\overset{(5.9)}{=} f_i - f_j - \langle g_j, x_i - x_j \rangle - L D_{D_h}\big((x_i, u_i), (x_j, u_j)\big) \\ &\overset{(i)}{\geq} 0 \end{aligned}$$

Hence, the set $\hat{S} = \{u_i, \hat{f}_i, \hat{g}_i\}_{i \in I}$ satisfies the interpolation conditions for convex interpolation [Taylor et al., 2017, Thm. 1], and therefore there exists a convex function $\hat{f} : \mathbb{R}^d \to \mathbb{R}$ that interpolates $\hat{S}$, thus proving (ii).

$(ii) \implies (iii)$. Let $\hat{f}$ be a convex function that interpolates the set $\hat{S}$. Moreover, let us choose $\hat{f}$ such that $\hat{f}$ is finite valued and bounded below on $\mathbb{R}^d$ (this is always possible since the set S is finite). Let $\mu = 1/L$ and define the function f as the Bregman-Moreau envelope

$$f := \mathrm{env}_\mu^h \hat{f}.$$

Since $\hat{f}$ is convex and bounded below, Proposition 5.7 ensures that $f \in \mathcal{F}_L^h$. Let $i \in I$. As $\hat{f}$ interpolates $\hat{S}$, we have $\hat{g}_i \in \partial \hat{f}(u_i)$ which means that

$$L \nabla^2 h(u_i)(x_i - u_i) \in \partial \hat{f}(u_i)$$

by definition of $\hat{g}_i$ (5.11). Hence, we have that u_i satisfies the first-order optimality conditions defining the Bregman-Moreau proximal map of $\hat{f}$ at x_i (Proposition 5.8), therefore

$$u_i = \mathrm{prox}^h_\mu \hat{f}(x_i).$$

As $\hat{f}$ interpolates $\hat{S}$ we have $\hat{f}(u_i) = \hat{f}_i$ and thus

$$f(x_i) \overset{(5.8)}{=} \hat{f}(u_i) + LD_h(x_i, u_i) = \hat{f}_i + LD_h(x_i, u_i) \overset{(5.11)}{=} f_i,$$

and similarly

$$\nabla f(x_i) \overset{(5.7)}{=} \frac{1}{\mu}\left(\nabla h(x_i) - \nabla h\left(\mathrm{prox}^h_\mu \hat{f}(x_i)\right)\right) = L\left(\nabla h(x_i) - \nabla h(u_i)\right) \overset{(5.12)}{=} g_i,$$

which achieves to prove that f interpolates the set S.

■

5.3.3 Equivalence for the entropic kernel

By examining the necessary conditions from Proposition 5.6 and the sufficient conditions form Proposition 5.9, one can notice that they do not seem to coincide in general. However, in the case of the entropic kernel, some favorable algebra allows to show that they are in fact equivalent.

Lemma 5.10. *Let $L > 0$, h_e be the entropic kernel, $x, x' \in \mathbb{R}^d_{++}, g, g' \in \mathbb{R}^d$ and define*

$$u = \nabla h_e^*(\nabla h_e(x) - \frac{1}{L}g),\ u' = \nabla h_e^*(\nabla h_e(x') - \frac{1}{L}g').$$

Then

$$D_{D_{h_e}}\big((x,u),(x',u')\big) = D_{h_e^*}\big(\nabla h_e(x) - \frac{1}{L}(g-g'), \nabla h_e(x)\big)$$

Proof. For the entropy, we have

$$u = \nabla h_e^*(\nabla h_e(x) - \frac{1}{L}g) = x \circ \exp(-g/L).$$

We use the identity for the double Bregman divergence of the entropy [Bauschke, 2006, Example 2.9] which yields

$$D_{D_{h_e}}\big((x,u),(x',u')\big) = D_{h_e}(x, u \circ x'/u'),$$

and thus

$$\begin{aligned} D_{D_{h_e}}\big((x,u),(x',u')\big) &= D_{h_e}(x, x \circ \exp(\frac{g'-g}{L})) \\ &= D_{h_e}\left(x, \nabla h_e^*(\nabla h_e(x) + \frac{1}{L}(g'-g))\right) \\ &= D_{h_e^*}\left(\nabla h_e(x) + \frac{1}{L}(g'-g), \nabla h_e(x)\right). \end{aligned}$$

where we used the duality property of the Bregman divergence (Lemma 5.3). ■

Using this identity, we can show that the necessary and sufficient conditions are equivalent and formulate interpolability conditions for entropic-smooth functions.

Theorem 5.11 (Entropic-smooth interpolability). *Let $L > 0$, $d \geq 1$ a dimension, I be a finite index set and*

$$S = \{x_i, f_i, g_i\}_{i \in I} \in (\mathbb{R}^d_{++} \times \mathbb{R} \times \mathbb{R}^d)^{|I|}.$$

Consider the entropic kernel h_e. The set S is $\mathcal{F}_L^{h_e}$-interpolable if and only if for every $i, j \in I$,

$$f_i - f_j - \langle g_j, x_i - x_j \rangle \geq L D_{h_e^*} \left[\nabla h_e(x_i) - \frac{1}{L}(g_i - g_j), \nabla h_e(x_i) \right]. \tag{5.13}$$

Proof. First, note that the entropy satisfies Assumption 5.1. The implication is a consequence from the necessary condition in Proposition 5.6. Conversely, assume that (5.13) holds. By the identity stated in Lemma 5.10, we have that

$$D_{h_e^*} \left[\nabla h_e(x_i) - \frac{1}{L}(g_i - g_j), \nabla h_e(x_i) \right] = D_{D_{h_e}} \big((x_i, u_i), (x_j, u_j) \big) \quad \forall i, j \in I$$

with $u_i = x_i \circ \exp(-g_i/L)$ for $i \in I$ and therefore, the set S satisfies Condition (i) in Proposition 5.9, which implies that S is $\mathcal{F}_L^{h_e}$-interpolable. ■

Remark 4. *In fact, the "favorable algebra" from Lemma 5.10 also holds for the squared Euclidean norm; therefore, our analysis can be seen as a strict extension of the Euclidean interpolability conditions from Taylor et al. [2017] to the kernels that satisfy this property as well as Assumption 5.1. However, it seems that the quadratic kernels and the entropy are the only kernels with such favorable properties. As stated by Bauschke and Combettes [2011], these two geometries seem to be "limiting cases in a profound sense".*

5.4 Writing the PEP as a convex program on the Kullback-Leibler cone

We now use the interpolation conditions to formulate the problem of finding the worst-case scenario of Bregman methods in entropic-smooth settings as a finite-dimensional problem. Then, we show that by relaxing over the dimension d, this problem becomes a convex program on a new cone that we call the *Kullback-Leibler cone.*

5.4.1 Formulating the PEP on the set of pairwise Kullback-Leibler matrices

Recall that the performance estimation problem (PEP) for entropic-smooth convex minimization writes

$$\begin{array}{ll} \text{maximize} & f_N - f_* \\ \text{subject to} & D_{h_e}(x_*, x_0) = R, \\ & f_i = f(x_i), g_i = \nabla f(x_i) \text{ for all } i \in I \text{ and some } f \in \mathcal{F}_L^{h_e}, \\ & x_N = \mathcal{A}\left(x_0, \{(f_i, g_i)\}_{i=0}^{N-1} \right), \end{array}$$

in the variables d, $\{(x_i, f_i, g_i)\}_{i \in I}$.

Using the interpolations conditions from Theorem 5.11, we can now write explicitly the constraints on the second line, and thus the PEP is equivalent to

$$\begin{array}{ll} \text{maximize} & f_N - f_* \\ \text{subject to} & D_{h_e}(x_*, x_0) = R, \\ & f_i - f_j - \langle g_j, x_i - x_j\rangle \geq L D_{h_e^*}\left[\nabla h_e(x_i) - \frac{1}{L}(g_i - g_j), \nabla h_e(x_i)\right], \quad i, j \in I, \\ & x_N = \mathcal{A}\left(x_0, \{(f_i, g_i)\}_{i=0}^{N-1}\right), \end{array}$$

in the variables d, $\{(x_i, f_i, g_i)\}_{i\in I}$. We now need to express the algorithmic constraints. For simplicity, we first focus on the case when $\mathcal{A}$ is the vanilla Bregman gradient descent (BGD) algorithm with fixed step size $\lambda > 0$, which writes

$$\nabla h_e(x_{i+1}) = \nabla h_e(x_i) - \lambda \nabla f(x_i), \quad i = 0 \ldots N-1.$$

Other Bregman-type algorithms which perform an arbitrary linear combination of past gradients can be treated with the same approach.

It is now convenient to introduce the dual variables $\{s_i\}_{i\in I}$ such that

$$s_i = \nabla h_e(x_i) = \log(x_i)$$

for $i \in I$. We use the three-point identity [Bauschke et al., 2017, Lemma 3] to express the inner product as

$$\begin{aligned} \lambda\langle g_j, x_i - x_j\rangle &= \langle \nabla h_e(x_j) - (\nabla h_e(x_j) - \lambda g_j), x_i - x_j\rangle \\ &= D_{h_e}(x_i, \nabla h_e^*[\nabla h_e(x_j) - \lambda g_j]) - D_{h_e}(x_i, x_j) - D_{h_e}(x_j, \nabla h_e^*[\nabla h_e(x_j) - \lambda g_j]) \\ &= D_{h_e^*}(s_j - \lambda g_j, s_i) - D_{h_e^*}(s_j, s_i) - D_{h_e^*}(s_j - \lambda g_j, s_j) \end{aligned}$$

where the last line uses the result on the Bregman divergence of the conjugate (Lemma 5.3). Hence, using this formulation on the dual space, the PEP rewrites

$$\begin{array}{ll} \text{maximize} & f_N - f_* \\ \text{subject to} & D_{h_e^*}(s_0, s_*) = R, \\ & \lambda(f_i - f_j) - D_{h_e^*}(s_j - \lambda g_j, s_i) + D_{h_e^*}(s_j, s_i) + D_{h_e^*}(s_j - \lambda g_j, s_j) \geq \\ & \qquad \lambda L D_{h_e^*}\left[s_i - \frac{1}{L}(g_i - g_j), s_i\right], \quad i, j \in I, \\ & s_{i+1} = s_i - \lambda g_i, \quad i = 0 \ldots N-1, \end{array} \tag{5.14}$$

in the variables $d, \{(s_i, f_i, g_i)\}_{i\in I}$. We now observe that the constraints depend linearly on the vector of function values $F = (f_0, \ldots, f_N, f_*)$, as well as the values of the Bregman divergence between some dual vectors which are linearly constrained. We make this idea clear in the following proposition, in which we write the PEP in a generic form.

Proposition 5.12. *There exist integers m, p, a constraint matrix $A \in \mathbb{R}^{q\times m}$, vectors $c_0, c_1, \ldots, c_p$ in $\mathbb{R}^{N+2}$ and matrices $B_0, B_1, \ldots, B_p \in \mathbb{R}^{m\times m}$ such that $A\mathbf{1}_m = 0$ and the PEP (5.14) is equivalent to the following optimization problem*

$$\begin{array}{ll} \text{maximize} & \langle c_0, F\rangle \\ \text{subject to} & D \in \mathcal{K}_m(A), \\ & \mathbf{Tr}(B_0 D) = R, \\ & \langle c_i, F\rangle + \mathbf{Tr}(B_i D) \geq 0, \quad i = 1 \ldots q, \end{array} \tag{PEP-$\mathcal{K}$}$$

in the variables $F \in \mathbb{R}^{N+2}$ and $D \in \mathbb{R}^{m\times m}$, where $\mathcal{K}_m(A)$ is defined as the set of constrained pairwise KL-distance matrices as follows

$$\mathcal{K}_m(A) := \left\{ D_{h_e^*}[S,S] \,:\, \textit{there exists } d \in \mathbb{N} \textit{ and } S \in \mathbb{R}^{d\times m} \textit{ such that } AS^T = 0 \right\}. \tag{5.15}$$

Proof. Introducing some auxiliary variables $\{s_i'\}_{i\in I} \in (\mathbb{R}^d)^{|I|}$ and $\{s_{ij}''\}_{i,j\in I} \in (\mathbb{R}^d)^{|I|^2}$, we reformulate (5.14) as

$$\begin{array}{ll} \text{maximize} & f_N - f_* \\ \text{subject to} & D_{h_e^*}(s_0, s_*) = R, \\ & \lambda(f_i - f_j) - D_{h_e^*}(s_j', s_i) + D_{h_e^*}(s_j, s_i) + D_{h_e^*}(s_j', s_j) \geq \\ & \qquad\qquad\qquad\qquad \lambda L D_{h_e^*}\left[s_{ij}'', s_i\right], \quad i,j \in I, \\ & s_{ij}'' = s_i - \frac{1}{L}(g_i - g_j), \quad i,j \in I \\ & s_i' = s_i - \lambda g_i, \quad i \in I, \\ & s_{i+1} = s_i - \lambda g_i, \quad i = 0 \ldots N-1, \end{array} \tag{5.16}$$

in the variables $d, \{(s_i, s_i', f_i, g_i)\}_{i\in I}, \{s_{ij}''\}_{i,j\in I}$. Eliminating the redundant $\{g_i\}$ variables, the last three lines of the constraints rewrite

$$\begin{aligned} s_{ij}'' &= s_i - \frac{1}{\lambda L}(s_i - s_i' - s_j + s_j'), \quad i,j \in I, \\ s_{i+1} &= s_i', \quad i = 0 \ldots N-1. \end{aligned} \tag{5.17}$$

Let us stack the variables $\{s_i\}_{i\in I}, \{s_i'\}_{i\in I}, \{s_{i,j}''\}_{i,j\in I}$ in a matrix $S \in \mathbb{R}^{d\times m}$ with $m = 3(N+2)(N+2)$. Define index reorderings $\sigma, \sigma' : I \to \{1 \ldots m\}$, and $\sigma'' : I \times I \to \{1 \ldots m\}$ such that.

$$\begin{aligned} S[:, \sigma(i)] &= s_i, \quad i \in I, \\ S[:, \sigma'(i)] &= s_i', \quad i \in I, \\ S[:, \sigma''(i,j)] &= s_{ij}'', \quad i,j \in I, \end{aligned}$$

The constraints (5.17) are linear and thus can be written

$$AS^T = 0$$

for some matrix $A \in \mathbb{R}^{q\times m}$ with $q = 2N+2$. Notice that $A\mathbf{1} = 0$ as the constraints (5.17) are invariant by adding a constant term to each variable. Using the index reorderings and inroducing the distance matrix $D = D_{h_e^*}[S,S] \in \mathbb{R}^{m\times m}$, Problem (5.16) becomes equivalent to

$$\begin{array}{ll} \text{maximize} & F_N - F_* \\ \text{subject to} & D_{\sigma(0),\sigma(*)} = R, \\ & \lambda(F_i - F_j) - D_{\sigma'(j),\sigma(i)} + D_{\sigma(j),\sigma(i)} + D_{\sigma'(j),\sigma(j)} \geq \lambda L D_{\sigma''(i,j),\sigma(i)}, \quad i,j \in I, \\ & D = D_{h_e^*}[S,S] \text{ for some } d \in \mathbb{N} \text{ and } S \in \mathbb{R}^{d\times m} \text{ such that } AS^T = 0, \end{array}$$

in the variables $F \in \mathbb{R}^{N+2}, D \in \mathbb{R}^{m\times m}$. Due to the definition (5.15), the last constraint rewrites $D \in \mathcal{K}_m(A)$, and the others are linear, hence the problem is of the form (PEP-$\mathcal{K}$). ■

5.4.2 The Kullback-Leibler cone

Problem (PEP-$\mathcal{K}$) involves linear constraints, and the set $\mathcal{K}_m(A)$ that we call the *Kullback-Leibler cone with log-linear constraints.* Indeed, the matrix A encodes linear constraints on the dual variables $s_i = \log x_i$. We first show that it is indeed a convex cone.

Proposition 5.13. *Let $m, q \geq 1$, and a constraint matrix $A \in \mathbb{R}^{q\times m}$ such that $A\mathbf{1}_m = 0$. The the set $\mathcal{K}_m(A)$ defined in (5.15) is a convex cone.*

Proof. Scalar multiplication. Let $D \in \mathcal{K}_m(A)$ and $\alpha > 0$. Let us pick $S \in \mathbb{R}^{n\times m}$ such that $D = D_{h^*}[S, S]$. Consider the matrix $S' = S + (\log \alpha)\mathbf{1}_{n\times m}$. Then, since we assumed $A\mathbf{1}_m = 0$, we have $AS' = AS$. Moreover, since the dual KL divergence $D_{h_e^*}$ satisfies

$$D_{h_e^*}(u + \log \alpha, v + \log \alpha) = \alpha D_{h_e^*}(u, v) \quad \forall u, v \in \mathbb{R}^d,$$

(see (5.2) for the expression of $D_{h_e^*}$) we have $D_{h_e^*}[S', S'] = \alpha D_{h_e^*}[S, S] = \alpha D$ and hence

$$\alpha D \in \mathcal{K}_m(A).$$

Additivity. Let $D, D' \in \mathcal{K}_m(A)$ and $\alpha > 0$. Choose $S \in \mathbb{R}^{n'\times m}, S' \in \mathbb{R}^{n\times m}$ such that

$$D = D_{h_e^*}[S, S], \quad D' = D_{h_e^*}[S', S'].$$

Consider the matrix $S'' \in \mathbb{R}^{(n+n')\times m}$ obtained by concatenating S and S' as $S'' = [S; S']$. Then $A(S'')^T = AS^T + A(S')^T = 0$, and we have

$$D_{h_e^*}[S'', S''] = D_{h_e^*}[S, S] + D_{h_e^*}[S', S'] = D + D'$$

which proves that $D + D' \in \mathcal{K}_m(A)$ and concludes the proof. ■

The set $\mathcal{K}_m(A)$ is the entropic counterpart of the Euclidean Distance Matrix cone, which can be expressed as a linear transformation of the cone of positive semidefinite matrices [Krislock and Wolkowicz, 2011]. Moreover, in the Euclidean setting, linear constraints on the generating points are equivalent to a set of linear constraints on the distance matrix, which allows formulating the PEP as a semidefinite program [Drori and Teboulle, 2014].

To the best of our knowledge, the Kullback-Leibler cone is novel and no method is known for solving problems of type (PEP-$\mathcal{K}$). However, as a first result, we are able to describe the cone in the case where there are no linear constraints, i.e. $A = 0$. In that case, we simply denote

$$\mathcal{K}_m := \mathcal{K}_m(0)$$

The unconstrained Kullback-Leibler cone is trivial. It is known that the Kullback-Leibler divergence is less regular than the Euclidean distance, for instance, it is not symmetric. Here, we analyze this structure a step further by showing that *the closure of the KL cone $\mathcal{K}_m$ is trivial.* That is, any matrix with nonnegative entries and null diagonal can be approached by a sequence of pairwise KL divergence matrices.

Proposition 5.14 (The closure of $\mathcal{K}_m$ is trivial)**.** *Let $m \geq 3$. The closure of the cone $\mathcal{K}_m$ is*

$$\overline{\mathcal{K}}_m = \left\{ D \in \mathbb{R}_+^{m\times m} \, : \, \mathbf{diag}(D) = (0 \dots 0) \right\}.$$

We start by proving a lemma. Let us denote $\{E_{ij}\}_{1\leq i,j\leq m}$ the canonical basis of the set of matrices of size $m \times m$.

Lemma 5.15. *Let $m \geq 2$. The matrix $E_{1,2}$ belongs to $\overline{\mathcal{K}}_m$, the closure of $\mathcal{K}_m$.*

Proof. Let us first show the result for $m = 3$. Consider the points $x_t, y_t, z_t \in \mathbb{R}$ defined as

$$x_t = t,\ y_t = te^{-1/t},\ z_t = te^{-1/s}$$

where $t, s > 0$ are to be chosen later. Then the pairwise KL divergences of x_t, y_t, z_t are

$$\begin{aligned}
D_{h_e}(x_t, y_t) &= 1 + t(e^{-1/t} - 1)\\
D_{h_e}(y_t, x_t) &= t - e^{-1/t}(1 + t)\\
D_{h_e}(x_t, z_t) &= t(\frac{1}{s} + e^{-1/s} - 1)\\
D_{h_e}(z_t, x_t) &= t(1 - e^{-1/s} - \frac{1}{s}e^{-1/s})\\
D_{h_e}(y_t, z_t) &= e^{-1/t}(\frac{t}{s} - 1) + t(e^{-1/s} - e^{-1/t})\\
D_{h_e}(z_t, y_t) &= e^{-1/s}(1 - \frac{t}{s}) + t(e^{-1/t} - e^{-1/s})
\end{aligned}$$

When $t \to 0$, the pairwise KL matrix $D_{h_e}[X_t, X_t]$ with $X = [x_t; y_t; z_t]$ satisfies

$$\lim_{t\to 0} D_{h_e}[x_t, y_t, z_t] = \lim_{t\to 0} \begin{pmatrix} D_{h_e}(x_t, x_t) & D_{h_e}(x_t, y_t) & D_{h_e}(x_t, z_t)\\ D_{h_e}(y_t, x_t) & D_{h_e}(y_t, y_t) & D_{h_e}(y_t, z_t)\\ D_{h_e}(z_t, x_t) & D_{h_e}(z_t, y_t) & D_{h_e}(z_t, z_t) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0\\ 0 & 0 & 0\\ 0 & e^{-1/s} & 0 \end{pmatrix}$$

which shows, by taking $s \to 0$, that E_{12} belongs to the closure of $\mathcal{K}_3$ (remember that $D_h[X_t, X_t] = D_{h^*}[\log X_t, \log X_t]$). For $m = 2$, it suffices to take the set of points $\{x_t, y_t\}$. For $m \geq 4$, using the same reasoning and by considering the set of points

$$\{x_t, y_t, z_t \dots z_t\},$$

we show that $E_{12} \in \overline{\mathcal{K}}_m$. ■

We are now able to prove the desired result.

Proof of Proposition 5.14. By symmetry of the roles of $x_1, \dots, x_m$, Lemma 5.15 implies that for every $i \neq j \in \{1 \dots m\}$, $E_{ij} \in \overline{\mathcal{K}}_m$. Hence, since it is a convex cone, the cone spanned by these elements also belongs to $\overline{\mathcal{K}}_m$, i.e.,

$$\mathrm{Cone}(\{E_{ij}\}_{1\leq i\neq j\leq m}) \subset \overline{\mathcal{K}}_m,$$

which proves the result since $\overline{\mathcal{K}}_m \subset \left\{D \in \mathbb{R}_+^{m\times m} : \mathbf{diag}(D) = (0 \dots 0)\right\} = \mathrm{Co}(\{E_{ij}\}_{1\leq i\neq j\leq m})$. ■

This result demonstrates the lack of regularity of the Kullback-Leibler divergence, since any combination of nonnegative distances can be approached by a pairwise KL matrix, unlike the Euclidean setting where distance matrices have a nontrivial structure [Krislock and Wolkowicz, 2011].

However, the general case of $\mathcal{K}_m(A)$, where additional log-linear constraints are imposed, cannot be described as easily. Numerical exploration seems to suggest that adding such constraints make the cone non-trivial; but we were not able to provide an explicit description in such setting.

5.5 Preliminary numerical results and conjectures

Although we formulated the performance estimation problem for entropic-smooth minimization as a convex program, there is no known solver for such problem, as we are not able to easily describe the Kullback-Leibler cone $\mathcal{K}_m(A)$ (beyond the setting $A = 0$, where it has a trivial structure).

In this section, we propose a basic heuristic method for solving the PEP, and provide preliminary numerical results.

5.5.1 Heuristic solver for Kullback-Leibler cone programs

Recall the generic form of a K-L cone program

$$\begin{array}{ll} \text{maximize} & \langle c_0, F\rangle \\ \text{subject to} & D \in \mathcal{K}_m(A), \\ & \mathbf{Tr}(B_0 D) = R, \\ & \langle c_i, F\rangle + \mathbf{Tr}(B_i D) \geq 0, \quad i = 1 \ldots q, \end{array}$$

in the variables D, F. We propose here the following basic heuristic:

1. Randomly sample n_S elements $D_1, \ldots, D_{n_S}$ of $\mathcal{K}_m(A)$, and consider the cone spanned by these elements
$$\mathrm{Co}(D_1, \ldots, D_{n_S}) \subset \mathcal{K}_m(A).$$

2. Solve the restricted linear problem on $\mathrm{Co}(D_1, \ldots, D_{n_s})$, writing
$$\begin{array}{ll} \text{maximize} & \langle c_0, F\rangle \\ \text{subject to} & D = \alpha_1 D_1 + \cdots + \alpha_{n_S} D_{n_S}, \\ & \alpha \geq 0, \\ & \mathbf{Tr}(B_0 D) = R, \\ & \langle c_i, F\rangle + \mathbf{Tr}(B_i D) \geq 0, \quad i = 1 \ldots q, \end{array}$$
in the variables $D, F, \alpha \in \mathbb{R}^{n_S}$.

5.5.2 A simple class of linear-span Bregman methods

We propose to analyze the following class of Bregman first-order methods:

$$\nabla h(x_{k+1}) = \nabla h(x_k) - \sum_{i=0}^{k} \gamma_{k,i} \nabla f(x_k), \quad k = 0, 1, \ldots \tag{BFOM}$$

where $\Gamma = \{\gamma_{i,k}\}_{0 \leq i \leq k}$ is a sequence of *fixed* coefficients (i.e., the method is not adaptive). While choosing $\gamma_{k,k} = \lambda$ and $\gamma_{i,k} = 0$ when $k \neq i$ yields the vanilla Bregman Gradient descent method, one can wonder if there exist some other choice of Γ with better worst-case guarantees. To this extend, we propose to use the PEP methodology.

For a given number of iterations N, the worst-case behavior of (BFOM) on the class $\mathcal{F}_L^h$ of entropic-smooth convex functions can be computed by solving the performance estimation problem

$$\begin{array}{ll} \text{maximize} & f(x_N) - f(x_*) \\ \text{subject to} & D_h(x_*, x_0) = R, \\ & f \in \mathcal{F}_L^{h_e}, \\ & \nabla h(x_{k+1}) = \nabla h(x_k) - \sum_{i=0}^k \gamma_{k,i} \nabla f(x_k), \quad k = 0, \ldots, N-1 \end{array} \tag{PEP-BFOM}$$

in the variables $f, x_0, \ldots, x_N, x_*$, and we denote by $\mathcal{E}_N(\Gamma) = \mathrm{val}(\text{PEP-BFOM})$ its optimal value.

In the Euclidean setting with $h = \|\cdot\|^2/2$, the class of algorithms (BFOM) covers a large number of first-order methods, including gradient descent and inertial algorithms such as Nesterov's accelerated gradient method [Nesterov, 1983]. The approach of finding the "best" coefficients Γ through worst-case performance estimation has been explored in Drori and Teboulle [2014], Kim and Fessler [2016], Drori and Taylor [2019], Kim [2019].

However, for Bregman gradient methods on relatively-smooth functions, there is currently no known method with better guarantee than BGD. While, in Chapter 3, we proved that BGD is optimal for generic kernel h, we propose to study if results can be improved with the specific entropic geometry by a method of the class (BFOM).

Resolution for $N = 2$. As a simple problem, we propose to solve (PEP-BFOM) for $N = 2$ iterations and several values of the three parameters $\gamma_{0,0}, \gamma_{1,0}, \gamma_{1,1}$. Hence we study the scheme

$$\begin{aligned} \nabla h_e(x_1) &= \nabla h_e(x_0) - \gamma_{0,0} \nabla f(x_0) \\ \nabla h_e(x_2) &= \nabla h_e(x_1) - \gamma_{1,1} \nabla f(x_1) - \gamma_{1,0} \nabla f(x_0) \end{aligned} \tag{BFOM-2-it}$$

For homogeneity reasons, we can assume w.l.o.g that $L = 1$ and $R = 1$.

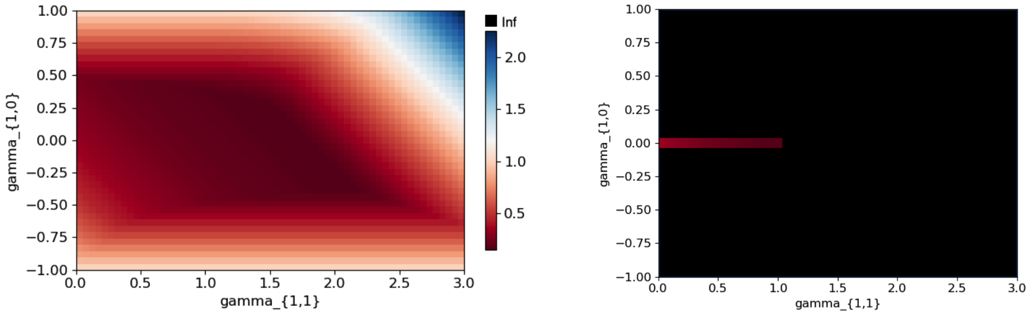
Following the approach detailed in Section 5.4, we write (PEP-BFOM) as a convex program on the Kullback-Leibler cone, and solve it with the heuristic described above.

In Figure 5.1, we show the numerical results for a fixed value of $\gamma_{0,0} = 1/L$, as a function of $\gamma_{1,1}, \gamma_{1,0}$. We compare with the worst-case performance of the same algorithm in the Euclidean setting with $h = \frac{1}{2}\|\cdot\|^2$, obtained with the standard PEP methodology [Drori and Teboulle, 2014, Taylor et al., 2017]. Numerical evidence shows the following:

- in the Euclidean case, there is a non-trivial set of values for $\gamma_{1,0}$ that yields an improved rate, which include well-known momentum-based algorithms.
- In the entropic setting, however, there is no value of $\gamma_{1,0} \neq 0$ that allows the functions value to be bounded, i.e., the value of (PEP-BFOM) is finite only for

  $$\gamma_{1,1}, \gamma_{0,0} \in (0, 1/L] \text{ and } \gamma_{1,0} = 0.$$

  In other words, among the class of methods described by (BFOM-2-it), only Bregman Gradient Descent with step size bounded by $(0, 1/L]$ has guaranteed worst-case convergence.



(a) Euclidean

(b) Entropic

Figure 5.1: Worst-case performance of generic scheme (BFOM-2-it) for $N = 2$ iterations, with $\gamma_{0,0} = 1/L$ and for different values of $(\gamma_{11}, \gamma_{10})$. Black denotes an infinite value. In the Euclidean case (left), the dependence is continous and a large set of values for the parameters is allowed. For the entropic kernel (right), only the choice $\gamma_{11} \in (0, 1/L]$ and $\gamma_{10} = 0$ (corresponding to vanilla BGD) guarantees a finite worst-case objective gap.

From numerical evidence to formal counter-examples. Numerical results shown above are not formal proofs. However, for negative results, they can lead to rigorous arguments by examining the corresponding worst-case functions, which we deduce from the solutions to the PEP. For instance; let us show that the worst-case performance of (BFOM-2-it) is not bounded for $\gamma_{0,0} = \gamma_{1,1} = 1$ and $\gamma_{1,0} > 0$ in the entropic setting. Let us choose the one-dimensional function f

$$f(x) = D_h(x, x_*)$$

for some $x_* > 0$ to be fixed later. f is convex and entropic-smooth with constant 1. Applying (BFOM-2-it) to f yields

$$x_1 = x_0 \exp\left(-\gamma_{0,0}\nabla f(x_0)\right) = x_*,$$
$$x_2 = x_1 \exp\left(-\gamma_{1,1}\nabla f(x_1) - \gamma_{1,0}\nabla f(x_0)\right) = x_* \left(\frac{x_*}{x_0}\right)^{\gamma_{1,0}}$$

Then, the PEP objective writes

$$\frac{f(x_2) - f_*}{D_h(x_*, x_0)} = \frac{x_* \left(\frac{x_*}{x_0}\right)^{\gamma_{1,0}} \left(\gamma_{1,0} \log \frac{x_*}{x_0} - 1\right) + x_*}{x_* \log \frac{x_*}{x_0} - x_* + x_0}$$

Denoting $u := x_0/x_*$, the quantity rewrites

$$\frac{f(x_2) - f_*}{D_h(x_*, x_0)} = \frac{u^{-\gamma_{1,0}} \left(-\gamma_{1,0} \log u - 1\right) + 1}{-\log u - 1 + u} \rightarrow$$

which shows, for $\gamma_{1,0} > 0$, that

$$\lim_{\frac{x_0}{x_*} \to 0^+} \frac{f(x_2) - f_*}{D_h(x_*, x_0)} = +\infty$$

and hence that the worst-case performance is unbounded. Similar arguments can be used to show that the value is unbounded in the other situations described before.

5.6 Conclusion

In this chapter, we proposed to study computer-aided analyses of Bregman methods for entropic-smooth convex functions. First, we established interpolation conditions that allow to write the PEP as a finite-dimensional problem. These conditions rely on the smoothing property of the Bregman-Moreau envelope, as well as favorable algebraic properties of the entropic kernel (such as joint convexity of the Bregman divergence).

Then, we formulated the PEP as a convex problem on the cone of pairwise Kullback-Leibler divergences with log-linear constraints. Although we showed that in the setting with no constraints, this cone is trivial, we are not aware of a simple description in the general case. Therefore, we had to use heuristics for solving simple versions of the PEP.

In particular, we demonstrated on a small example of two iterations that among all general Bregman first-order methods with *fixed coefficients*, only Bregman gradient descent with step size smaller than $1/L$ is guaranteed to converge in the worst case. This shows that, even if we consider the specific entropic setting instead of generic kernels like in the previous chapters, building accelerated Bregman methods is a tedious task.

There are several important questions that arise for future work:

1. Can we find an algorithm that provably solves convex problems on the Kullback-Leibler cone with log-linear constraints? Note that, since PEPs are usually low-dimensional problems, we do not need efficient solvers, but rather ones with strong guarantees.

2. How to design accelerated Bregman methods in the entropic-smooth setting? An idea would be to consider methods like (BFOM) but with adaptive coefficients, in the spirit of the accelerated Bregman methods proposed in Hanzely et al. [2021], Hendrikx et al. [2020]. An additional difficulty in that case would be that the PEP methodology is not directly applicable to adaptive methods.

Chapter 6

Bregman Stochastic Gradient Descent and Variance Reduction

Chapter Abstract

We study the problem of minimizing an expectation of relatively-smooth functions using Bregman stochastic gradient methods. We first study Bregman stochastic gradient descent, and show convergence towards a region that depends on the magnitude of the gradients at the optimum. Then, we consider variance reduction methods for problems whose objective has a finite-sum structure. In this setting, we show that improved convergence rates can be obtained under additional regularity assumptions on the Bregman kernel. We provide numerical experiments on Poisson inverse problems.

Reference: this chapter is based on a publication in International Conference on Machine Learning [Dragomir et al., 2021b]. This work has been done in collaboration with Hadrien Hendrikx and Mathieu Even, and each student contributed equally.

6.1 Introduction

We consider the minimization problem

$$\min_{x\in\mathcal{C}} f(x), \text{ where } f(x) = \mathbb{E}_\xi\left[f_\xi(x)\right], \tag{6.1}$$

$\mathcal{C}$ is a convex subset of $\mathbb{R}^d$ and f_ξ are differentiable convex functions. These problems typically arise in machine learning when performing (empirical) risk minimization, in which case f_ξ is for instance a loss function for some random sample ξ. Problem (6.1) is also encountered in signal processing applications such as image deblurring or tomographic reconstruction inverse problems, in which the goal is to recover an unknown signal from a large number of noisy observations. First-order methods are often very efficient for solving such problems, but computing a gradient ∇f might be very expensive for large-scale problems (large number of components f_ξ), and even impossible in the case of true risk minimization (infinite number of f_ξ). In this case, stochastic gradient methods have proven to be particularly effective thanks to their low cost per iteration. The simplest one, stochastic gradient descent (SGD), consists in updating x_t as

$$x_{t+1} = \arg\min_{u \in \mathcal{C}} \langle g_t, u - x \rangle + \frac{1}{2\eta_t} \|u - x_t\|^2$$

where g_t is a gradient estimate such that $\mathbb{E}[g_t] = \nabla f(x_t)$. In our case, a natural choice would be $g_t = \nabla f_{\xi_t}(x_t)$ for some ξ_t. The choice of the step size η_t is crucial for obtaining good performances and is typically related to the smoothness of f with respect to the Euclidean norm.

Beyond simply adapting the step size, a powerful generalization of SGD consists in refining the geometry and performing instead Bregman gradient (a.k.a mirror) steps as

$$x_{t+1} = \arg\min_{u \in \mathcal{C}} \langle g_t, u - x \rangle + \frac{1}{2\eta_t} D_h(u, x_t), \tag{BSGD}$$

where the Euclidean distance has been replaced by the Bregman divergence with respect to a kernel function h, which writes:

$$D_h(x, y) = h(x) - h(y) - \langle \nabla h(y), x - y \rangle, \tag{6.2}$$

for all $x, y \in \operatorname{int} \mathcal{C}$.The standard SGD algorithm corresponds to the case where $h = \frac{1}{2}\| \cdot \|^2$. However, a different choice of h might better fit the geometry of the set $\mathcal{C}$ and the curvature of the function, allowing the algorithm to take larger steps in directions where the objective gradient changes slowly. This choice is guided by the notion of relative smoothness and strong convexity, introduced in Bauschke et al. [2017], Lu et al. [2018]. Instead of the squared Euclidean norm for standard smoothness, *relative* regularity is measured with respect to the reference function h. In most situations, the update of BSGD can be computed in closed form. In some other cases, it might require running an optimization subroutine. However, it can be worth doing so to reduce the overall complexity, if the kernel h is chosen so that the inner objective (BSGD) is a good approximation of the objective function.

Contributions. In this work, we study Bregman stochastic gradient methods for relatively-smooth problems. First, we analyze the convergence of vanilla BSGD with fixed step size, relying on a variance condition that depends on the stochastic gradients at the optimum. We show that this condition leads to a tighter result than that of the previous work from Hanzely and Richtárik [2018], especially for the *interpolation setting* where the stochastic gradients are null at the optimum.

Then, we show that variance-reduction techniques, which are widely used to accelerate traditional Euclidean stochastic methods when the objective has a finite-sum structure, can be adapted to the Bregman setting. Although this generally requires stronger regularity assumptions (such as global smoothness of h and Lipschitz continuity of $\nabla^2 h^*$) and an accurate step size schedule, we show that the asymptotical rate of convergence solely depends on relative regularity constants. The same type of results (asymptotic speedup under additional smoothness assumptions) is observed when applying Nesterov-type acceleration to Bregman gradient methods [Hendrikx et al., 2020, Hanzely et al., 2021].

Finally, we illustrate the efficiency of our approach on inverse problems with Poisson noise, which are a typical application of relatively-smooth optimization.

Outline. We start by discussing related work in Section 6.2. Section 6.3 introduces the setup and useful lemmas. We study Bregman stochastic gradient descent in Section 6.4, variance reduction techniques in Section 6.5, and describe numerical experiments in Section 6.6.

6.2 Related work

Euclidean stochastic gradient methods and variance reduction. Stochastic optimization methods, and in particular SGD, are very efficient when the number of samples is high [Bottou, 2012] and are often referred to as "the workhorse of machine learning". The problem with SGD is that, in general, it only converges to a neighbourhood of the optimum unless a diminishing step-size is used. Variance reduction can be used to counter this problem, and many variance-reduced methods have been developed, such as SAG [Schmidt et al., 2013], SDCA [Shalev-Shwartz and Zhang, 2013, Shalev-Shwartz, 2016], SVRG [Johnson and Zhang, 2013] or SAGA [Defazio et al., 2014].

Mirror descent for nonsmooth functions. In the deterministic setting, Bregman gradient descent was originally proposed as the *mirror descent* scheme for nonsmooth convex optimization [Nemirovski and Yudin, 1983, Beck and Teboulle, 2003][1]. Mirror descent was also extended to the stochastic setting [Nemirovski et al., 2009, Lan, 2012, Zhou et al., 2017, Zhang and He, 2018, Antonakopoulos et al., 2020], but remained focused on nonsmooth objectives.

To the best of our knowledge, variance reduction for Bregman stochastic methods was only studied in Shi et al. [2017] in the context of stochastic saddle-point optimization, but without leveraging relative regularity assumptions like we do in this work.

Stochastic Bregman methods for relatively-smooth problems. The introduction of relative smoothness by Bauschke et al. [2017] has opened new perspectives for applying Bregman methods to differentiable objectives. In this more recent context, we are only aware of a few works that study the stochastic variant of Bregman gradient descent.

Hanzely and Richtárik [2018] consider the same setting as us and obtain comparable convergence rates for Bregman SGD, but with a much looser notion of variance, which we discuss more in details in the next section. This is problematic since their bound on the variance is thus proportional to the magnitude of the gradients along the trajectory, and may thus be very large when far from the optimum. In contrast, our definition of variance leverages the stochastic gradients at the optimum, which allows us to obtain significant results without bounded gradients and in the interpolation regime (zero gradients at the optimum). Davis et al. [2018] also analyze a similar setting for Bregman SGD, but again with more restrictive assumptions on the noise and boundedness of the gradients.

When preparing the final version of this work, we became aware of the recent paper by Latafat et al. [2021], who also study variance-reduced Bregman stochastic algorithms for finite-sum minimization. Unlike us, they focus more on nonconvex objective functions. They also

[1] Note that *mirror descent* and *Bregman gradient* refer to the same algorithm, but that *mirror descent* is typically used when f is non-smooth, or in the online optimization community, whereas *Bregman gradient* is generally preferred when using the relative smoothness assumption. However, there is no consensus in the litterature; for instance, Hanzely and Richtárik [2018] use the *mirror descent* terminology although they consider the relatively-smooth setting.

prove a convergence rate on strongly convex objectives under additional regularity assumptions on f and h, but with a weaker dependence on relatively smoothness constants than ours.

6.3 Problem setup and preliminary lemmas

We begin by recalling the base ingredients for relatively-smooth minimization, and prove preliminary lemmas which will be useful for our analysis. For a more general introduction to Bregman methods, we refer the reader to Chapter 1. Let us begin with the blanket assumptions on the kernel function h.

Assumption 6.1. *The function h is a kernel function on $\mathcal{C}$, that is,*

(i) $h : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is a convex function,

(ii) h is finite-valued, strictly convex and differentiable on $\operatorname{int}\mathcal{C}$,

(iii) the Bregman gradient iterates are well-posed, i.e., for every $p \in \mathbb{R}^d$, the problem

$$\min_{u\in\mathcal{C}} \langle p, u\rangle + h(u)$$

has a unique minimizer, which belongs to $\operatorname{int}\mathcal{C}$.

Moreover, we additionally assume that h is twice continuously differentiable on $\operatorname{int}\mathcal{C}$.

We also recall the notions of relative smoothness and strong convexity [Bauschke et al., 2017, Lu et al., 2018].

Definition 6.1. *The function f is said to be L-relatively smooth and μ-relatively strongly convex with respect to h if it is differentiable and for all $x, y \in \operatorname{int}\mathcal{C}$,*

$$\mu D_h(x,y) \le D_f(x,y) \le L D_h(x,y), \tag{6.3}$$

where D_f is defined similarly to (6.2).

Note that if $\mu = 0$, the left-hand side inequality reduces to assuming convexity of f. Similarly, if $h = \frac{1}{2}\|\cdot\|^2$, then $D_h(x,y) = \frac{1}{2}\|x-y\|^2$, and the usual notions of smoothness and strong convexity are recovered. If both functions are two times differentiable, Equation (6.3) can be turned into an equivalent condition on the Hessians:

$$\mu\nabla^2 h(x) \preceq \nabla^2 f(x) \preceq L\nabla^2 h(x).$$

Throughout the chapter, we will generally write $\mu_{f/h}$ and $L_{f/h}$ to insist on the relative aspect.

Recall that the conjugate h^* of h is defined for $y \in \mathbb{R}^d$ as

$$h^*(y) = \sup_{x\in\mathbb{R}^d} \langle x, y\rangle - h(x).$$

Under Assumption 6.1, h^* is convex and differentiable on $\mathbb{R}^d$ [Bauschke and Combettes, 2011, Cor. 18.12], and $\nabla h^*(\nabla h(y)) = y$ for $y \in \operatorname{int} C$. We recall the relation between the Bregman divergence of h and that of its conjugate h^*.

Lemma 6.2 (Duality)**.** *For $x, y \in \operatorname{int}\mathcal{C}$, we have $D_h(x,y) = D_{h^*}(\nabla h(y), \nabla h(x))$.*

See, *e.g.*, Bauschke and Borwein [1997, Thm 3.7.] for the proof. Using duality, we prove the following key lemma.

Lemma 6.3. *Let $x \in \operatorname{int}\mathcal{C}$, and $g_1, g_2 \in \mathbb{R}^d$. Define the points x^+, x_1^+, x_2^+ as the unique points satisfying*

$$\begin{aligned}
\nabla h(x_1^+) &= \nabla h(x) - g_1,\\
\nabla h(x_2^+) &= \nabla h(x) - g_2,\\
\nabla h(x^+) &= \nabla h(x) - \frac{(g_1+g_2)}{2}.
\end{aligned}$$

Then we have

$$D_h(x, x^+) \le \frac{1}{2}\left[D_h(x, x_1^+) + D_h(x, x_2^+)\right].$$

Proof. Using Lemma 6.2 (duality), we have:

$$\begin{aligned}
D_h(x, x^+) &= D_{h^*}\big(\nabla h(x^+), \nabla h(x)\big)\\
&= D_{h^*}\left(\nabla h(x) - \frac{g_1+g_2}{2}, \nabla h(x)\right)\\
&= D_{h^*}\left(\frac{1}{2}[\nabla h(x) - g_1] + \frac{1}{2}[\nabla h(x) - g_2], \nabla h(x)\right)\\
&\le \frac{1}{2} D_{h^*}\left(\nabla h(x) - g_1, \nabla h(x)\right) + \frac{1}{2} D_{h^*}\left(\nabla h(x) - g_2, \nabla h(x)\right)\\
&= \frac{1}{2} D_{h^*}\big(\nabla h(x_1^+), \nabla h(x)\big) + \frac{1}{2} D_{h^*}\big(\nabla h(x_2^+), \nabla h(x)\big)
\end{aligned}$$

where the inequality step is obtained by convexity of the Bregman divergence in its first argument. The final result is obtained by using Lemma 6.2 back. ■

In the Euclidean case $h = \|\cdot\|^2$, we recover $\|\frac{g_1+g_2}{2}\|^2 \le \frac{1}{2}\left(\|g_1\|^2 + \|g_2\|^2\right)$.

We also recall the Bregman counterpart of an inequality linked to co-coercivity of the gradients [Nesterov, 2003, Eq. 2.1.7], which we proved in Chapter 5 (Proposition 5.5).

Lemma 6.4. *If a convex function f is L-smooth relative to the kernel h, then for any $\eta \le \frac{1}{L}$ and $x, y \in \operatorname{int}\mathcal{C}$,*

$$D_f(x,y) \ge \frac{1}{\eta} D_{h^*}(\nabla h(x) - \eta\left(\nabla f(x) - \nabla f(y)\right), \nabla h(x)).$$

6.4 Bregman stochastic gradient descent

In this section, we study the (BSGD) scheme with *fixed* step size $\eta > 0$. Recall that, by writing the first-order optimality conditions, and by strict convexity of h, the iterations can be equivalently defined as

$$\nabla h(x_{t+1}) = \nabla h(x_t) - \eta g_t$$

with $\mathbb{E}[g_t] = \nabla f(x_t)$. We denote by $x^\star$ the minimizer of f, and we assume $x^\star \in \operatorname{int}\mathcal{C}$ so that $\nabla f(x^\star) = 0$; the case where the solution lies on the boundary of $\mathcal{C}$ poses additional technical difficulties and is left for future work. We first establish a lemma satisfied by the updates of Bregman stochastic gradient methods with unbiased gradient estimate.

Lemma 6.5. *Let $x \in \operatorname{int}\mathcal{C}$ and $\eta > 0$. If x^+ satisfies $\nabla h(x^+) = \nabla h(x) - \eta g$ with $\mathbb{E}[g] = \nabla f(x)$, $\nabla f(x^\star) = 0$, then:*

$$\mathbb{E}\left[D_h(x^\star, x^+)\right] = D_h(x^\star, x) - \eta D_f(x^\star, x) - \eta D_f(x, x^\star) + \mathbb{E}\left[D_h(x, x^+)\right],$$

where the expectation is taken with respect to the choice of g.

Proof. By using the simple algebraic *three-point identity* which follows form the definition of Bregman divergence [Bauschke et al., 2017, Lemma 3], we get

$$\begin{aligned} D_h(x^\star, x) - D_h(x^\star, x^+) - D_h(x^+, x) &= \langle \nabla h(x^+) - \nabla h(x), x^\star - x^+ \rangle \\ &= -\eta \langle g, x^\star - x^+ \rangle \\ &= -\eta \langle g, x^\star - x \rangle - \eta \langle g, x - x^+ \rangle \end{aligned} \tag{6.4}$$

For the first term of the right-hand side, we use the fact that $\mathbb{E}[g] = \nabla f(x)$ and $\nabla f(x^\star) = 0$ to obtain

$$\mathbb{E}\left[-\eta \langle g, x^\star - x\rangle\right] = -\eta \langle \nabla f(x) - \nabla f(x^\star), x^\star - x\rangle = \eta D_f(x^\star, x) + \eta D_f(x, x^\star). \tag{6.5}$$

For the second term, we write

$$\mathbb{E}\left[-\eta\langle g, x - x^+\rangle\right] = \mathbb{E}\left[-\langle \nabla h(x) - \nabla h(x^+), x - x^+\rangle\right] = \mathbb{E}\left[-D_h(x, x^+) - D_h(x^+, x)\right]. \tag{6.6}$$

By replacing Equations (6.5) and (6.6) back in the expectation of (6.4), we get

$$D_h(x^\star, x) - \mathbb{E}\left[D_h(x^\star, x^+) + D_h(x^+, x)\right] = \eta D_f(x^\star, x) + \eta D_f(x, x^\star) - \mathbb{E}\left[D_h(x, x^+) + D_h(x^+, x)\right],$$

which yields the desired result after re-arranging terms. ■

Lemma 6.5 holds for any instance of BSGD that uses an unbiased gradient estimate. The methods that we study in the sequel essentially differ in the way the *variance* of this estimate is handled. In the case of vanilla Bregman SGD, we now specify a natural condition for bounding this variance.

Assumption 6.2. *The stochastic gradients g_t are such that*

$$g_t = \nabla f_{\xi_t}(x_t),$$

with $\mathbb{E}_{\xi_t}[f_{\xi_t}] = f$ and f_{ξ_t} is convex and $L_{f/h}$-relatively smooth with respect to h for all ξ_t. Besides, there exists a constant $\sigma^2 \geq 0$ such that for every $t \geq 0$,

$$\frac{1}{2\eta^2}\mathbb{E}_{\xi_t}\left[D_{h^*}\left(\nabla h(x_t) - 2\eta \nabla f_{\xi_t}(x^\star), \nabla h(x_t)\right)\right] \leq \sigma^2.$$

The assumption that the stochastic gradients are actual gradients of stochastic functions which are themselves smooth with respect to h is rather natural, as already discussed in the introduction. It is in particular verified when solving (empirical) risk minimization problems.

Assumption 6.2 is a Bregman adaptation of the usual variance at the optimum definition used for instance in Bach and Moulines [2011], Gower et al. [2019]. Note that if h^* is μ_h-strongly convex with respect to the Euclidean norm, then the assumption is verified for instance when the variance is bounded in ℓ_2 norm, that is if

$$\mathbb{E}_{\xi_t}\left[\|\nabla f_{\xi_t}(x^*)\|\right] \leq \mu_h \sigma^2,$$

since in that case we have

$$\frac{1}{2\eta^2} D_{h^*}\left(\nabla h(x_t) - 2\eta \nabla f_{\xi_t}(x^\star), \nabla h(x_t)\right) \leq \frac{1}{\mu_h}\|\nabla f_{\xi_t}(x^*)\|^2.$$

We used the fact that if h is μ_h-strongly convex, then h^* is $1/\mu_h$-smooth and its Bregman divergence is upper bounded by the squared Euclidean distance, see, e.g., Bauschke and Combettes [2010].

Remark 5. *Hanzely and Richtárik [2018] also consider Bregman SGD for relatively-smooth problems, but with a different assumption on the variance.*

$$\frac{1}{\eta_t}\mathbb{E}_{\xi_t}\left[\langle \nabla f(x_t) - \nabla f_{\xi_t}(x_t), x_{t+1} - \bar{x}_{t+1}\rangle\right] \leq \sigma^2, \tag{6.7}$$

for $t \geq 0$, where g_t is the stochastic gradient estimate and $\bar{x}_{t+1}$ is the output of the (theoretical) Bregman gradient step taken with the true gradient, that is, $\nabla h(\bar{x}_{t+1}) = \nabla h(x_t) - \eta_t \nabla f(x_t)$. Thus, their condition can be written:

$$\frac{1}{\eta_t^2}\mathbb{E}_{\xi_t}\left[D_h(x_{t+1}, \bar{x}_{t+1}) + D_h(\bar{x}_{t+1}, x_{t+1})\right] \leq \sigma^2,$$

so that σ^2 bounds at each step the distance (in the Bregman sense) between x_{t+1} and $\bar{x}_{t+1}$. To illustrate why our assumption is weaker, let us consider the case where h is μ_h-strongly convex. In this setting, a sufficient condition for (6.7) to hold is that

$$\frac{1}{\mu_h}\mathbb{E}_{\xi_t}\left[\|\nabla f(x_t) - \nabla f_{\xi_t}(x_t)\|^2\right] \leq \sigma^2,$$

while a sufficient condition for our variance definition to hold is (using that $\nabla f(x^\star) = 0$):

$$\frac{1}{\mu_h}\mathbb{E}_{\xi_t}\left[\|\nabla f(x^\star) - \nabla f_{\xi_t}(x^\star)\|^2\right] \leq \sigma^2,$$

which only depends on the magnitude of the gradients at the optimum instead of the variance along the full trajectory since x_t is replaced by $x^\star$. In particular, in the interpolation setting where $\nabla f_\xi(x^\star) = 0$ for every ξ, $\sigma^2 = 0$ with our condition.

We now state our convergence result for Bregman SGD. To avoid notation clutter, we generally omit with respect to which variable expectations are taken when clear from the context.

Theorem 6.6. *If f is $L_{f/h}$-smooth and $\mu_{f/h}$-strongly convex relative to h with $\mu_{f/h} > 0$, and Assumptions 6.1 and 6.2 hold, then for a step size $\eta \leq 1/(2L_{f/h})$, the iterates produced by Bregman stochastic gradient (BSGD) satisfy*

$$\mathbb{E}\left[D_h(x^\star, x_t)\right] \leq (1 - \eta\mu_{f/h})^t D_h(x^\star, x_0) + \eta\frac{\sigma^2}{\mu_{f/h}}.$$

Proof. By using Lemma 6.5 with $x = x_t$, we obtain:

$$\begin{aligned}\mathbb{E}_{\xi_t}\left[D_h(x^\star, x_{t+1})\right] &= D_h(x^\star, x_t) - \eta D_f(x^\star, x_t) - \eta D_f(x_t, x^\star) + \mathbb{E}_{\xi_t}\left[D_h(x_t, x_{t+1})\right] \\ &\leq D_h(x^\star, x_t) - \eta D_f(x^\star, x_t) + \mathbb{E}_{\xi_t}\left[D_h(x_t, x_{t+1})\right],\end{aligned} \tag{6.8}$$

since D_f is nonnegative by convexity of f. Using Lemma 6.3 with

$$\begin{aligned}x &= x_t, \\ g_1 &= 2\eta\left[\nabla f_{\xi_t}(x_t) - \nabla f_{\xi_t}(x^\star)\right], \\ g_2 &= 2\eta \nabla f_{\xi_t}(x^\star),\end{aligned}$$

we have that $\nabla h(x_{t+1}) = \nabla h(x_t) - (g_1 + g_2)/2$ and hence the last term can be bounded as

$$D_h(x_t, x_{t+1}) \leq \frac{1}{2} D_{h^*}(\nabla h(x_t) - g_1, \nabla h(x_t)) + \frac{1}{2} D_{h^*}(\nabla h(x_t) - g_2, \nabla h(x_t))$$

We use Lemma 6.4 (Bregman co-coercivity) to bound the first term as :

$$\begin{aligned}D_{h^*}(\nabla h(x_t) - g_1, \nabla h(x_t)) &= D_{h^*}(\nabla h(x_t) - 2\eta\left[\nabla f_{\xi_t}(x_t) - \nabla f_{\xi_t}(x^\star)\right], \nabla h(x_t)) \\ &\leq 2\eta D_{f_{\xi_t}}(x_t, x^\star),\end{aligned}$$

so that

$$\frac{1}{2}\mathbb{E}\left[D_{h^*}(\nabla h(x_t) - g_1, \nabla h(x_t))\right] \leq \eta D_f(x_t, x^\star),$$

by linearity of D_f in f and the fact that $\mathbb{E}_{\xi_t}\left[f_{\xi_t}\right] = f$. To bound the second term, we use the variance condition from Assumption 6.2:

$$\frac{1}{2}\mathbb{E}\left[D_{h^*}\left(\nabla h(x_t) - g_2, \nabla h(x_t)\right)\right] = \frac{1}{2}\mathbb{E}\left[D_{h^*}\left(\nabla h(x_t) - 2\eta \nabla f_{\xi_t}(x^\star), \nabla h(x_t)\right)\right] \leq \eta^2\sigma^2.$$

By relative strong convexity of f we have $-D_f(x^\star, x_t) \leq -\mu_{f/h} D_h(x^\star, x_t)$ and thus inserting these bounds in (6.8) yields

$$\mathbb{E}_{\xi_t}\left[D_h(x^\star, x_{t+1})\right] \leq (1 - \eta\mu_{f/h}) D_h(x^\star, x_t) + \eta^2\sigma^2.$$

■

Remark 6 (Interpolation). *In the interpolation setting (when $\nabla f_{\xi_t}(x^\star) = 0$ for all ξ_t), we have that $\sigma^2 = 0$. Theorem 6.6 thus proves linear convergence in this case. For instance, when solving objectives of the form $D_{\mathrm{KL}}(Ax, b)$ (which has applications in optimal transport [Mishchenko, 2019]) or $D_{\mathrm{KL}}(b, Ax)$ (which has application in deblurring or tomographic reconstruction), then the variance as defined in Hanzely and Richtárik [2018] may be unbounded, whereas the variance as we define it is equal to 0 if there exists z such that $Az = b$.*

When f is convex but not relatively-strongly convex ($\mu_{f/h} = 0$), the analysis of Theorem 6.6 can be adapted to obtain a $1/T$ decrease of the error up to a noise region.

Theorem 6.7 (Convex case). *Under the same assumptions as Theorem 6.6, if $\mu_{f/h} = 0$, then*

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T} D_f(x^\star, x_t)\right] \leq \frac{D_h(x^\star, x_0)}{\eta T} + \eta\sigma^2 \tag{6.9}$$

Proof. We follow the same analysis as the beginning of the proof of Theorem 6.6. Bounding the terms of (6.8) in same way but by keeping $D_f(x^\star, x_t)$, we get

$$\eta D_f(x^\star, x_t) = D_h(x^\star, x_t) - \mathbb{E}_{\xi_t}\left[D_h(x^\star, x_{t+1})\right] + \eta^2\sigma^2.$$

Averaging over t and dividing by η leads to (6.9). ■

Contrary to the Euclidean case, we do not obtain a guarantee on the average iterate in general. This is because the bound is on the average of $D_f(x^\star, x_t)$ instead of $D_f(x_t, x^\star)$, and Bregman divergences are not necessarily convex in their second argument (except for the particular cases of the Euclidean distance and the Kullback-Leibler divergence).

6.5 Variance reduction

We have shown in the previous section that BSGD enjoys guarantees that are similar to that of its Euclidean counterpart, although the notion of variance needs to be adapted. We show in this section that it is also possible to apply variance reduction techniques to accelerate convergence, in the case where the objective is a finite sum of the form

$$\min_{x\in\mathcal{C}} f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x), \tag{6.10}$$

As previously, we also assume that the minimizer $x^\star$ belongs to $\operatorname{int}\mathcal{C}$, so that $\nabla f(x^\star) = 0$. To solve (6.10), we propose in Algorithm 7 a Bregman adaptation of the SAGA algorithm [Defazio et al., 2014], which is one of the most standard variance-reduced stochastic methods. Following its Euclidean counterpart, Bregman-SAGA uses previously computed gradients $\nabla f_j(\phi_j^t)$ in order to reduce the variance of the estimate g_t at the current point x_t. Note that one can check that the gradient estimate is ubiased as $\mathbb{E}_{i_t}\left[g_t\right] = \nabla f(x_t)$.

Algorithm 7 Bregman-SAGA$((\eta_t)_{t\geq 0}, x_0)$

```
$\phi_i = x_0$ for $i = 1, ..., n$
for $t = 0, 1, 2, \dots$ do
    Pick $i_t \in \{1, ..., n\}$ uniformly at random
    $g_t = \nabla f_{i_t}(x_t) - \nabla f_{i_t}(\phi_{i_t}^t) + \frac{1}{n}\sum_{j=1}^n \nabla f_j(\phi_j^t)$
    $x_{t+1} = \arg\min_x \left\{\langle g_t, x - x_t\rangle + \frac{1}{\eta_t} D_h(x, x_t)\right\}$
    $\phi_{i_t}^{t+1} = x_t$, and store $\nabla f_{i_t}(\phi_{i_t}^{t+1})$.
    $\phi_j^{t+1} = \phi_j^t$ for $j \neq i_t$.
end for
```

We now introduce our standing assumptions for analyzing Bregman-SAGA. Compared to the previous section, we need an additional crucial regularity condition on D_h.

Assumption 6.3. *For all $i \in \{1, \cdots, n\}$, f_i is $L_{f/h}$-smooth relative to h, and f is $\mu_{f/h}$-strongly convex relative to h. Moreover, there exists a gain function G such that for any $x, y, v \in \mathbb{R}^d$ and $\lambda \in [-1, 1]$,*

$$D_{h^*}\left(x + \lambda v, x\right) \leq G(x, y, v)\lambda^2 D_{h^*}\left(y + v, y\right).$$

Such structural assumptions appear to be essential for analyzing Bregman-type methods that use information provided by gradients of past iterates. The function G models the fact that the function $D_{h^*}(x+v, x)$ is not generally homogeneous in v nor invariant to translation in x (except for the Euclidean case where it is equal to $\|v\|^2/2$). Note that such difficulties are also encountered for obtaining accelerated rates with inertial variants of Bregman descent, where similar assumptions are needed [Hanzely et al., 2021]. This seems unavoidable, as suggested by the lower bound from Chapter 3.

Although the gain function G is relatively abstract at this point, it plays a key role in defining the step-size, and convergence guarantees similar those of Euclidean SAGA can be obtained provided G can be chosen small enough. We first state the general Theorem 6.8 (convergence proof for Algorithm 7), and then detail how G can be bounded in several interesting cases.

For $t \geq 0$ and step-sizes $\eta_t > 0$, define

$$H_t = \frac{1}{n}\sum_{i=1}^{n} D_{f_i}(\phi_i^t, x^\star),$$

and the potential ψ_t as follows:

$$\psi_t = \frac{1}{\eta_t} D_h(x^\star, x_t) + \frac{n}{2} H_t.$$

First note that by convexity of h and of the f_i, $\psi_t \geq 0$ for all t. Our goal in this section is to show that $\{\psi_t\}_{t\geq 0}$ converges to 0 at a given speed. Indeed, since $D_h(x^\star, x_t) \leq \psi_t$, this implies (as in Section 6.4) that x_t converges to $x^\star$ at the same rate. To ease notations, we define

$$\bar{\alpha}^t = \frac{1}{n}\sum_{j=1}^{n} \nabla f_j(\phi_j^t), \text{ and } \bar{\alpha}_i^t = \nabla f_i(\phi_i^t) - \bar{\alpha}^t.$$

Expectations are taken either on the full past, or on the index i_t conditionally on the past, but we generally omit this dependence to avoid notation clutter. Similarly, we write i instead of i_t when clear from the context.

Theorem 6.8. *Assume that Algorithm 7 is run with a step size sequence $\{\eta_t\}_{t\geq 0}$ satisfying $\eta_t = 1/(8L_{f/h}G_t)$ for every $t \geq 0$, with G_t decreasing in t and such that for all $j \in \{1, \cdots, n\}$:*

$$G_t \geq G\left(\nabla h(x_t), \nabla h(x_t), \frac{1}{L_{f/h}}(\nabla f_j(x_t) - \nabla f_j(x^\star))\right), \tag{6.11}$$

$$G_t \geq G\Big(\nabla h(x_t) - 2\eta_t \bar{\alpha}^t, \nabla h(\phi_j^t), \frac{1}{L_{f/h}}(\nabla f_j(\phi_j^t) - \nabla f_j(x^\star))\Big). \tag{6.12}$$

Then, under Assumptions 6.1 and 6.3, the potential ψ_t satisfies

$$\mathbb{E}_{i_t}\left[\psi_{t+1}\right] \leq \left(1 - \min\left(\eta_t \mu_{f/h}, \frac{1}{2n}\right)\right)\psi_t,$$

In the convex case ($\mu_{f/h} = 0$), we obtain that

$$\mathbb{E}\left[\frac{1}{4T}\sum_{t=1}^{T} D_f(x_t, x^\star) + H_t\right] \leq \frac{\psi_0}{T}. \tag{6.13}$$

Proof. Since Bregman-SAGA also uses an update with un unbiased gradient estimate, we apply Lemma 6.5 which yields

$$\mathbb{E}_{i_t}\left[D_h(x^\star, x_{t+1})\right] = D_h(x^\star, x_t) - \eta_t D_f(x^\star, x_t) - \eta_t D_f(x_t, x^\star) + \mathbb{E}_{i_t}\left[D_h(x_t, x_{t+1})\right]. \qquad (6.14)$$

In the same way as before, 6.3 implies $D_h(x_t, x_{t+1}) \le (D_1 + D_2)/2$, with

$$\begin{aligned} D_1 &= D_{h^*}(\nabla h(x_t) - 2\eta_t\left[\nabla f_i(x_t) - \nabla f_i(x^\star)\right], \nabla h(x_t)),\\ D_2 &= D_{h^*}(\nabla h(x_t) - 2\eta_t(\nabla f_i(x^\star) - \bar{\alpha}_i^t), \nabla h(x_t)). \end{aligned}$$

Using the gain function with the fact that $\eta_t \le 1/L_{f/h}$, we have

$$\begin{aligned} \mathbb{E}_{i_t}\left[D_1\right] &= \mathbb{E}_i\left[D_{h^*}\left(\nabla h(x_t) - 2\eta_t\left(\nabla f_i(x_t) - \nabla f_i(x^\star)\right), \nabla h(x_t)\right)\right]\\ &\le 4L_{f/h}^2\eta_t^2\,\mathbb{E}_i\Big[G\left(x_t, x_t, \frac{1}{L_{f/h}}(\nabla f_i(x_t) - \nabla f_i(x^\star))\right) \times\\ &\qquad D_{h^*}\left(\nabla h(x_t) - \frac{1}{L_{f/h}}\left(\nabla f_i(x_t) - \nabla f_i(x^\star)\right), \nabla h(x_t)\right)\Big]\\ &\le 4L_{f/h}\eta_t^2\mathbb{E}_i\left[G\left(x_t, x_t, \frac{1}{L_{f/h}}(\nabla f_i(x_t) - \nabla f_i(x^\star))\right)D_f(x_t, x^\star)\right]\\ &\le 4L_{f/h}\eta_t^2 G_t D_f(x_t, x^\star). \end{aligned}$$

where we used Lemma 6.4, and the last line is implied by the choice of G_t in Equation (6.11). Note that we can pull the G_t term out of the expectation over the choice of i since G_t holds for all i.

For bounding D_2, we use a Bregman counterpart of the bias-variance inequality [Pfau, 2013], which we prove in Appendix 6.A for completeness (Lemma 6.14). Let us denote with V the (random) vector

$$V = -2\eta_t(\nabla f_i(x^\star) - \nabla f_i(\phi_i^t)).$$

Then, notice that, the expectation of V with respect to the choice of i is

$$\mathbb{E}_i\left[V\right] = -\frac{2\eta_t}{n}\sum_{j=1}^{n}\nabla f_j(\phi_t^t) = -2\eta_t\overline{\alpha}^t,$$

since $\nabla f(x^\star) = 0$, and that

$$D_2 = D_{h^*}\left(\nabla h(x_t) + V - \mathbb{E}_i\left[V\right], \nabla h(x_t)\right).$$

Therefore, Lemma 6.14 leads to

$$\begin{aligned} \mathbb{E}_i\left[D_2\right] &= \mathbb{E}_i\left[D_{h^*}\left(\nabla h(x_t) + V - \mathbb{E}_i\left[V\right], \nabla h(x_t)\right)\right]\\ &\le \mathbb{E}_i\left[D_{h^*}\left(\nabla h(x_t) + V - \mathbb{E}_i\left[V\right], \nabla h(x_t) - \mathbb{E}_i\left[V\right]\right)\right]\\ &\le 4\eta_t^2 L_{f/h}^2\mathbb{E}_i\Bigg[G\left(\nabla h(x_t) - 2\eta_t\overline{\alpha}^t, \nabla h(\phi_j^t), \frac{1}{L_{f/h}}\left(\nabla f_i(\phi_j^t) - \nabla f_i(x^\star)\right)\right) \times\\ &\qquad D_{h^*}\left[\nabla h(\phi_i^t) - \frac{1}{L_{f/h}}\left(\nabla f_i(\phi_i^t) - \nabla f_i(x^\star)\right), \nabla h(\phi_i^t)\right]\Bigg]\\ &\le 4L_{f/h}\eta_t^2 G_t\mathbb{E}_i\left[D_{f_i}(\phi_i^t, x^\star)\right]. \end{aligned}$$

Where we used the gain function for translating D_{h^*}, and the choice of G_t in Equation (6.12). Recall that $H_t = \frac{1}{n}\sum_{j=1}^n D_{f_j}(\phi_j^t, x^\star)$. Plugging the upper bounds of D_1 and D_2 into Equation (6.14), we obtain:

$$\mathbb{E}_{i_t}\left[D_h(x^\star, x_{t+1})\right] - D_h(x^\star, x_t) \leq -\eta_t D_f(x^\star, x_t) - \eta_t D_f(x_t, x^\star) + 2L_{f/h}\eta_t^2 G_t\left[D_f(x_t, x^\star) + H_t\right].$$

Following Hofmann et al. [2015], we write:

$$\mathbb{E}_{i_t}\left[H_{t+1}\right] = \left(1 - \frac{1}{n}\right) H_t + \frac{1}{n} D_f(x_t, x^\star),$$

Indeed, $\phi_j^{t+1} = \phi_j^t$ with probability $1 - 1/n$, and $\phi_i^{t+1} = x_t$ with probability $1/n$. Therefore, we can use the $-H_t/n$ term to control the excess term from bounding $D_h(x_t, x_{t+1})$. In the end, using that G_t is decreasing and so η_t is increasing, we obtain the following recursion:

$$\begin{aligned}
\mathbb{E}_{i_t}\left[\psi_{t+1}\right] - \psi_t &= \frac{1}{\eta_{t+1}} D_h(x^\star, x_{t+1}) + \frac{n}{2} H_{t+1} - \frac{1}{\eta_t} D_h(x^\star, x_t) - \frac{n}{2} H_t \\
&\leq \frac{1}{\eta_t}\left(D_h(x^\star, x_{t+1}) - D_h(x^\star, x_t)\right) + \frac{n}{2}\left(H_{t+1} - H_t\right) \\
&\leq -D_f(x^\star, x_t) - \frac{1}{2}\left(1 - 4\eta_t L_{f/h} G_t\right) H_t - \left(1 - 2\eta_t L_{f/h} G_t - \frac{1}{2}\right) D_f(x_t, x^\star).
\end{aligned}$$

If we choose $\eta_t \leq 1/(8L_{f/h}G_t)$ then the last term is positive and $1 - 4\eta_t L_{f/h} G_t \geq 1/2$, so that using the relative strong convexity of f leads to:

$$\begin{aligned}
\mathbb{E}_{i_t}\left[\psi_{t+1}\right] &\leq (\eta_t^{-1} - \mu_{f/h}) D_h(x^\star, x_t) + \left(1 - \frac{1}{2n}\right) \frac{n}{2} H_t \\
&\leq \left(1 - \min\left(\eta_t \mu_{f/h}, \frac{1}{2n}\right)\right) \psi_t.
\end{aligned}$$

The result can then be obtained by chaining this inequality. If $\mu_{f/h} = 0$ then we start back from Equation (6.5), use that $D_f(x^\star, x_t) \geq 0$ and the same fact that $1 - 4\eta_t L_{f/h} G_t \geq 1/2$ to obtain:

$$\frac{1}{4}\left[D_f(x_t, x^\star) + H_t\right] \leq \psi_t - \mathbb{E}_{i_t}\left[\psi_{t+1}\right].$$

The result is obtained by averaging over T, since the right hand side yields a telescopic sum, leading to the $1/T$ rate of Equation (6.13). ■

After proving the general convergence theorem, we show how the step size η_t can be chosen in several settings. We start with the most favorable situation, the case where h is quadratic.

Corollary 6.9. *If h is a quadratic function, Assumption 6.3 is satisfied with $G = 1$, so that*

$$\mathbb{E}\left[\psi_t\right] \leq \left(1 - \min\left(\frac{1}{8\kappa_{f/h}}, \frac{1}{2n}\right)\right)^t \psi_0,$$

where $\kappa_{f/h} = L_{f/h}/\mu_{f/h}$ is the relative condition number.

Thus, we recover the result of Defazio et al. [2014] in the Euclidean setting. Another case where a simple gain function G can be found is if h is smooth and strongly convex with respect to some norm.

Corollary 6.10. *If h is L_h-smooth and μ_h-strongly convex with respect to the some norm $\|\cdot\|_E$, then the stepsize can be chosen constant as $\eta_t = \frac{\mu_h}{8L_h L_{f/h}}$, and*

$$\mathbb{E}\left[\psi_t\right] \leq \left(1 - \min\left(\frac{1}{8\kappa_h \kappa_{f/h}}, \frac{1}{2n}\right)\right)^t \psi_0.$$

Proof. In this setting, the gain function can be chosen constant equal to $G = L_h/\mu_h$, since we have for $x, y, v \in \mathbb{R}^d$ and $\lambda \in [-1, 1]$

$$D_{h^*}(x + \lambda v, x) \leq \frac{1}{2\mu_h}\|\lambda v\|_E^2 \leq \lambda^2 \frac{L_h}{\mu_h} D_{h^*}(y + v, y)$$

where we used the fact that h^* is $1/\mu_h$-smooth and $1/L_h$-strongly convex [Bauschke and Combettes, 2010]. Then, Theorem 6.8 yields the result. ■

This convergence result comes with an important drawback: the rate depends on the condition number $\kappa_h = L_h/\mu_h$ of h, which can be excessively large or even infinite in the context of relatively-smooth problems.

Asymptotical rates. In the general case, bounding the gain function is a more tedious task. As stated at the beginning of this section, one of the problems is that Bregman divergences lack translation invariance and homogeneity. However, as the algorithm converges, one can expect these conditions to hold locally, as $D_{h^*}(x + v, x)$ is approximated by $\frac{1}{2}\|v\|^2_{\nabla^2 h^*(x^*)}$ for small enough v, and x close enough to x^*. This is indeed what happens under enough regularity assumptions on h. Here, we consider the restricted setting where h is globally smooth and its conjugate h^* has Lipschitz Hessian.

Proposition 6.11. *If h is L_h-smooth and the Hessian $\nabla^2 h^*$ is M-smooth, then the gain function can be chosen as:*

$$G(x, y, v) = 1 + 2ML_h\left(\|y - x\| + \|v\|\right).$$

Proof. Writing the divergence in integral form, we have for $x, y, v \in \mathbb{R}^d$ and $\lambda \in [-1, 1]$

$$\begin{aligned}
D_{h^*}(x + \lambda v, x) &= \lambda^2 \int_0^1 \int_0^t v^\top \nabla^2 h^*(x + s\lambda v) v \, ds \, dt \\
&\leq \lambda^2 \int_0^1 \int_0^t \left(v^\top \nabla^2 h^*(y + sv)v \; + M\|y + sv - x - \lambda sv\|\|v\|^2\right) ds \, dt \\
&\leq \lambda^2 \int_0^1 \int_0^t \left(v^\top \nabla^2 h^*(y + sv)v \; + M\left(\|y - x\| + 2s\|v\|\right)\|v\|^2\right) ds \, dt \\
&= \lambda^2 \left(D_{h^*}(y + v, y) + M(\|y - x\| + \|v\|)\|v\|^2\right).
\end{aligned}$$

Using the fact that is h is L_h-smooth, h^* is $1/L_h$-strongly convex and hence

$$\|v\|^2 \leq 2L_h D_{h^*}(y + v, y),$$

leading to

$$D_{h^*}(x + \lambda v, x) \leq \lambda^2 \left[1 + 2ML_h\left(\|y - x\| + \|v\|\right)\right] D_{h^*}(y + v, y).$$

■

Note that, even if the regularity conditions of Proposition 6.11 do not hold globally (such as for problems with unbounded curvature), they are at least valid on every bounded subset of $\operatorname{int} C$, as soon as h is C^3 on $\operatorname{int} C$, and we show in the sequel that the dependence on these constants at least disappears asymptotically. We now explicit a possible explicit choice for G_t in this setting.

Corollary 6.12. *Assume that h is L_h-smooth, μ_h-strongly convex and that the Hessian $\nabla^2 h^*$ is M-smooth. Then, there exists an explicit constant C such that if Algorithm 7 is run with a step size $\eta_t = 1/(8L_{f/h}G_t)$ with G_t decreasing and satisfying*

$$G_t \geq \min\left(\frac{L_{f/h}L_h}{\mu_h}, 1 + C\Big(\sum_{j=1}^n \|x_t - \phi_j^t\| + \|\sum_{j=1}^n \nabla f_j(\phi_j^t)\|\Big)\right), \tag{6.15}$$

then we have the convergence rate

$$\mathbb{E}\left[\psi_{t+1}\right] \leq \left(1 - \min\left(\frac{1}{8G_t\kappa_{f/h}}, \frac{1}{2n}\right)\right)\psi_t,$$

where $\lim_{t\to\infty} G_t = 1$, or, more precisely,

$$\mathbb{E}\left[G_t\right] \leq 1 + \mathcal{O}\left(1 - \min\left(\frac{1}{8\kappa_h\kappa_{f/h}}, \frac{1}{2n}\right)\right)^t. \tag{6.16}$$

The explicit expression for the constant C is provided in Appendix 6.A.2 along with the proof. Although the result involves smoothness constants of h which can be large in the relatively-smooth setting, this dependence vanishes asymptotically. Hence, after some time t, which we can roughly estimate using Equation (6.16), we obtain that $G_t = O(1)$. Thus, we reach the same kind of convergence rate as in the ideal quadratic case, which depends only on the *relative* condition number $\kappa_{f/h}$, but with more general functions h, and thus possibly much better conditioning.

We note however that the choice of G_t in (6.15) is rather conservative and has a more theoretical value for now; in our experiments, we will simply choose a constant G_t and show that the algorithm behaves well.

6.6 Application to Poisson inverse problems

We consider the minimization problem

$$\min_{x\in\mathbb{R}_+^d} f(x) = \frac{1}{n}D_{\mathrm{KL}}(b, Ax) \tag{6.17}$$

where $D_{\mathrm{KL}}(u,v) = \sum_{i=1}^n u_i \log(u_i/v_i) - u_i + v_i$ is the Kullback-Leibler divergence, and $A \in \mathbb{R}^{n\times d}$ is a typically sparse matrix that models the measurement process. Problem (6.17) models the maximum likelihood estimation problem when assuming the statistical model

$$b \sim \mathrm{Poisson}(Ax^*)$$

where x^* is the true unknown signal. Inverse problems with Poisson noise arise in various signal processing applications such as astronomy or computerized tomography, see Bertero et al. [2009] and references therein.

As a motivating application of relative smoothness, Bauschke et al. [2017] prove that the Poisson objective f is relatively smooth with respect to the log-barrier reference function

$$h(x) = -\sum_{i=1}^{d} \log x_i$$

with constant $\sum_{j=1}^{n} b_j/n$. This constant can be quite conservative when A is a sparse matrix, and so we prove a better estimate by leveraging this structure. For $j \in \{1 \dots n\}$, we denote S_j the support of the j-th column of A, that is

$$S_j := \{i \in \{1 \dots n\} \, : A_{ij} \neq 0\}.$$

Proposition 6.13. *The Poisson objective function defined in* (6.17) *is L-smooth relative to the log-barrier for*

$$L \geq \frac{1}{n} \max_{j \in \{1 \dots d\}} \sum_{i \in S_j} b_i.$$

Proof. Let us denote $A_1, \dots A_n$ the row vectors of A. We refine the analysis from Bauschke et al. [2017, Lemma 7] and start by writing for $x \in \mathbb{R}^d_{++}, u \in \mathbb{R}^d$

$$\langle u, \nabla^2 f(x) u \rangle = \frac{1}{n} \sum_{i=1}^{n} b_i \frac{(A_i^\top u)^2}{(A_i^\top x)^2}.$$

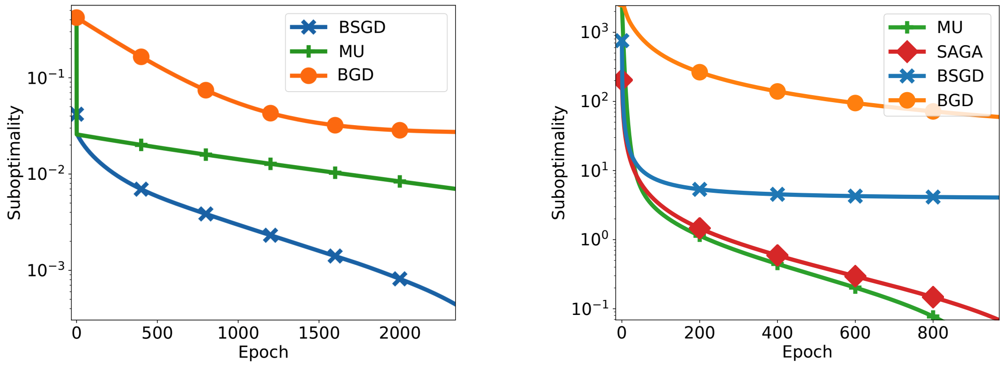
Applying the Jensen inequality to the function $t \mapsto t^2$ and weights $w_{ij} = A_{ij} x_j / (A_i^\top x)$ yields

$$\begin{aligned}
\langle d, \nabla^2 f(x) d \rangle &= \frac{1}{n} \sum_{i=1}^{n} b_i \left(\sum_{j=1}^{d} w_{ij} \frac{u_j}{x_j} \right)^2 \\
&\leq \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{d} b_i w_{ij} \cdot \frac{u_j^2}{x_j^2} \\
&\leq \frac{1}{n} \sum_{j=1}^{d} \sum_{i \in S_j}^{n} b_i \frac{u_j^2}{x_j^2} \\
&\leq L \sum_{j=1}^{d} \frac{u_j^2}{x_j^2} \\
&= L \, \langle u, \nabla^2 h(x) u \rangle,
\end{aligned}$$

where we used the fact that $w_{ij} \in [0,1]$ if $i \in S_j$, and $w_{ij} = 0$ otherwise. ■

The relative Lipschitz constant provided by Proposition 6.13 can be considerably smaller than $\sum_{j=1}^{n} b_j/n$ when A is sparse, which is the case in practical applications.

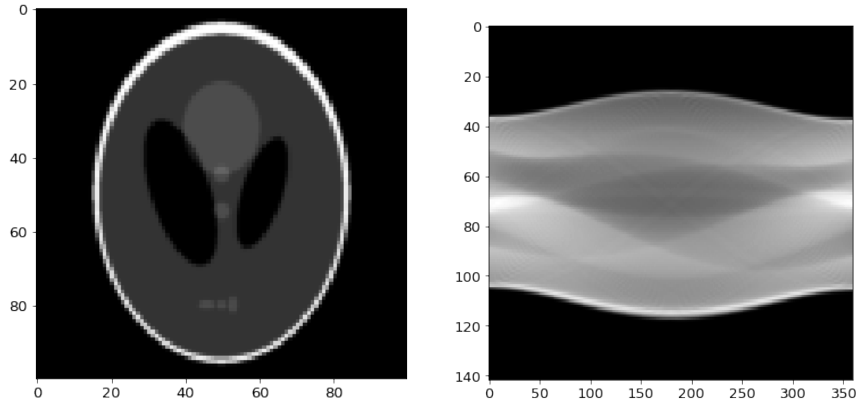
For our numerical experiments, we compare full-batch Bregman gradient descent (BGD), Bregman stochastic gradient descent (BSGD), and the Bregman SAGA scheme described in Algorithm 7. We also implement the Multiplicative Update (MU), also known as Lucy-Richardson or Expectation-Maximization [Shepp and Vardi, 1982], which is the standard baseline for Poisson inverse problems.



(a) Toy problem (interpolation setting).

(b) Tomographic reconstruction.

Figure 6.1: Experiments on Poisson inverse problems. `SAGA` designates Bregman-SAGA (Algorithm 7).



(a) Original image

(b) Sinogram

Figure 6.2: Illustration of the Radon transform on the Shepp-Logan phantom. On the sinogram, each column corresponds to the line integral of the image under different projection angles.

Synthetic problem in the interpolation setting. In Figure 6.1(a), we simulate a synthetic problem the gradients at optimum are zero, by choosing $b = Ax^\star$ for some random $A \in \mathbb{R}^{n \times d}$ and $x^* \in \mathbb{R}^d$ (indices sampled uniformly between 0 and 1), with $n = 10000$ and $d = 1000$. This corresponds to the *interpolation setting*, for which Theorem 6.7 predicts fast convergence towards the optimum of Bregman SGD. We observe that BGD is by far the slowest algorithm, but that BSGD is faster than MU thanks to the stochastic speedup. We also observe that BSGD does not plateau in a noise region and converges to the true solution, which is consistent with Theorem 6.7. The step-size for BGD and BSGD is chosen as $1/L_{f/h}$, whereas MU is parameter-free.

Tomographic reconstruction problem. Computerized tomography [Kak and Slaney, 2001] is the task of reconstructing an object from cross-sectional projections, with fundamental applications to medical imaging. We study a classical synthetic toy problem for this task: the Shepp-Logan phantom (Figure 6.2(a)). In this setting, the observation matrix A corresponds to the discrete *Radon transform*, which is the cross-sectional projection of the original image x along different projection angles $\theta_1, \dots, \theta_n$ (Figure 6.2(b)). That is, the objective writes

$$f(x) = \frac{1}{n} D_{\mathrm{KL}}(b, Ax) = \frac{1}{n} \sum_{i=1}^{n} D_{\mathrm{KL}}(b_{\theta_i}, A_{\theta_i} x) \tag{6.18}$$

where $b_{\theta_i}, A_{\theta_i}$ correspond to the observation and projection matrix along the angle θ_i. For stochastic algorithms, the formulation (6.18) naturally yields a finite-sum structure: we thus take $f_i(x) = D_{\mathrm{KL}}(b_{\theta_i}, A_{\theta_i} x)$ for $i = 1 \dots n$.

We corrupt the sinogram with Poisson noise, and apply our algorithms. We use $n = 360$ projection angles, and the image dimension is $d = 100^2$. As the matrix A has a sparse structure, we use the relative smoothness constant provided by Proposition 6.13 for a better estimate. The step-size given by theory was rather conservative in this case, so we increased it by a factor of 5 for all Bregman algorithms (and even 10 for BGD).

Figure 6.1(b) shows again that stochastic algorithms drastically outperform BGD. Yet, BSGD quickly reaches a plateau because of the noise. On the other hand, BSAGA enjoys variance reduction and fast convergence to the optimum. In this case, BSAGA is on par with MU, the state-of-the-art algorithm for this problem. One possible explanation is that the log barrier allows relative smoothness to hold, but heavily slows down Bregman algorithms when coordinates are close to 0. Yet, these results are encouraging and one may hope for even faster convergence of BSAGA for tomographic reconstruction with a tighter reference function.

6.7 Conclusion

In this work, we have (i) given tight convergence guarantees for Bregman SGD that allow to accurately describe its behaviour in the interpolation setting, and (ii) introduced and analyzed Bregman analogs to the standard variance-reduced algorithm SAGA.

The convergence results on variance reduction require stronger assumptions on the objective than relative smoothness and strong convexity. We show that fast rates can be obtained asymptotically when h is nicely behaved, provided that the step sizes are well chosen. However, in practical experiments, the step size does not need to be taken as conservatively as the theory predicts, and there does not seem to be a slow transient regime. Therefore, a promising extension of our work is to bridge the gap between theory and practice by analyzing this regime and providing more accurate worst-case guarantees.

Appendices

6.A Missing proofs for variance reduction

6.A.1 Bregman variance decomposition

First, we use the following Bregman counterpart of a standard variance identity [Pfau, 2013], which we prove for completeness.

Lemma 6.14 (Bregman variance decomposition)**.** *Let X be a random variable on $\mathbb{R}^d$. Then for any $u \in \mathbb{R}^d$,*

$$\mathbb{E}\left[D_{h^*}(X, u)\right] = D_{h^*}(\mathbb{E}\left[X\right], u) + \mathbb{E}\left[D_{h^*}(X, \mathbb{E}\left[X\right])\right] \tag{6.19}$$

As a consequence, for any random variable V on $\mathbb{R}^d$ and point $y \in \mathbb{R}^d$ we have

$$\mathbb{E}\left[D_{h^*}(y + V - \mathbb{E}\left[V\right], y - \mathbb{E}\left[V\right])\right] \geq \mathbb{E}\left[D_{h^*}(y + V - \mathbb{E}\left[V\right], y)\right]. \tag{6.20}$$

Proof. Denoting $\overline{x} := \mathbb{E}\left[X\right]$, We have for $u \in \mathbb{R}^d$

$$\begin{aligned}
D_{h^*}(\overline{x}, u) + \mathbb{E}\left[D_{h^*}(X, \overline{x})\right] &= h^*(\overline{x}) - h^*(u) - \langle \nabla h^*(u), (\overline{x} - u)\rangle \\
&\quad + \mathbb{E}\left[h^*(X) - h^*(\overline{x}) - \nabla h^*(\overline{x})^\top (X - \overline{x})\right] \\
&= -h^*(u) - \nabla h^*(u)^\top (\overline{x} - u) + \mathbb{E}\left[h^*(X)\right] \\
&= \mathbb{E}\left[h^*(X) - h^*(u) - \nabla h^*(u)^\top (X - u)\right] \\
&= \mathbb{E}\left[D_{h^*}(X, u)\right]
\end{aligned}$$

which proves (6.19). Then, (6.20) follows from applying it to the point $u = y - \mathbb{E}\left[V\right]$ and the random variable $X = y + V - \mathbb{E}\left[V\right]$, along with using the nonnegativity of the Bregman divergence $D_{h^*}(\mathbb{E}\left[X\right], u)$. ■

6.A.2 Lipschitz-Hessian setting: proof of Corollary 6.12

Corollary 6.15 (Recall of Corollary 6.12)**.** *Assume that h is L_h-smooth, μ_h-strongly convex and the Hessian $\nabla^2 h^*$ is M-smooth. Then, there exists an explicit constant C such that if Algorithm 7 is run with a step size $\eta_t = 1/(8L_{f/h}G_t)$ with G_t decreasing in t and satisfying*

$$G_t \geq \min\left(\frac{L_{f/h}L_h}{\mu_h}, 1 + C\Big(\sum_{j=1}^n \|x_t - \phi_j^t\| + \|\sum_{j=1}^n \nabla f_j(\phi_j^t)\|\Big)\right),$$

then we have the convergence rate

$$\mathbb{E}_{i_t}\left[\psi_{t+1}\right] \leq \left(1 - \min\left(\frac{1}{8G_t\kappa_{f/h}}, \frac{1}{2n}\right)\right)\psi_t, \tag{6.21}$$

where $\lim_{t\to\infty} G_t = 1$, or, more precisely,

$$\mathbb{E}\left[G_t\right] \leq 1 + \mathcal{O}\left(1 - \min\left(\frac{1}{8\kappa_h\kappa_{f/h}}, \frac{1}{2n}\right)\right)^t.$$

Proof. Using the gain function from Proposition 6.11, to satisfy the assumptions of Theorem 6.8 it is sufficient to choose G_t such that

$$\begin{aligned}G_t \geq 1 + 2ML_h\Big(&\frac{1}{L_{f/h}}\|\nabla f_{i_t}(x_t) - \nabla f_{i_t}(x^\star)\| + \frac{1}{L_{f/h}}\|\nabla f_{i_t}(\phi^t_{i_t}) - \nabla f_{i_t}(x^\star)\| \\ &+ \|\nabla h(x_t) - \nabla h(\phi^t_{i_t}) - \frac{1}{4nL_{f/h}}\sum_{j=1}^n \nabla f_j(\phi^t_j)\|\Big).\end{aligned} \tag{6.22}$$

As the quantities involving $\nabla f_{i_t}(x^\star)$ are unknown, we provide an upper estimate. We can proceed in the following way, using the fact that, due to relative regularity, f_i is also smooth with constant $L_h L_{f/h}$, and f is strongly convex with constant $\mu_h\mu_{f/h}$:

$$\begin{aligned}\|\nabla f_{i_t}(x_t) - \nabla f_{i_t}(x^\star)\|^2 &\leq 2L_h L_{f/h} D_{f_{i_t}}(x_t, x^\star) \\ &\leq 2L_h L_{f/h}\, n\, D_f(x_t, x^\star) \\ &\leq \frac{L_h L_{f/h}}{\mu_h\mu_{f/h}}\, n\, \|\frac{1}{n}\sum_{j=1}^n \nabla f_j(x_t)\|^2 \\ &\leq \frac{\kappa_f\kappa_{f/h}}{n}\left(\sum_{j=1}^n \|\nabla f_j(x_t) - \nabla f_j(\phi^t_j)\| + \|\sum_{j=1}^n \nabla f_j(\phi^t_j)\|\right)^2 \\ &\leq \frac{\kappa_f\kappa_{f/h}}{n}\left(\sum_{j=1}^n L_h L_{f/h}\|x_t - \phi^t_j\| + \|\sum_{j=1}^n \nabla f_j(\phi^t_j)\|\right)^2.\end{aligned}$$

And similarly, we can estimate the second term from

$$\|\nabla f_{i_t}(\phi^t_{i_t}) - \nabla f_{i_t}(x^\star)\| \leq \|\nabla f_{i_t}(x_t) - \nabla f_{i_t}(x^\star)\| + L_h L_{f/h}\|\phi^t_{i_t} - x_t\|,$$

which leads to the following upper estimate of the RHS of Condition (6.22):

$$\begin{aligned}1{+}2ML_h\Big(&\frac{1}{L_{f/h}}\|\nabla f_{i_t}(x_t) - \nabla f_{i_t}(x^\star)\| + \frac{1}{L_{f/h}}\|\nabla f_i(\phi^t_{i_t}) - \nabla f_i(x^\star)\| \\ &+ \|\nabla h(x_t) - \nabla h(\phi^t_{i_t}) - \frac{1}{4nL_{f/h}}\sum_{j=1}^n \nabla f_j(\phi^t_j)\|\Big) \\ \leq 1 + 2ML_h\Bigg(&\frac{2}{L_{f/h}}\|\nabla f_{i_t}(x_t) - \nabla f_{i_t}(x^\star)\| + L_h\|\phi^t_{i_t} - x_t\| + \|\nabla h(x_t) - \nabla h(\phi^t_{i_t})\| \\ &+ \frac{1}{4nL_{f/h}}\|\sum_{j=1}^n \nabla f_j(\phi^t_j)\|\Bigg) \\ \leq 1 + 2ML_h\Bigg(&2\sqrt{\frac{\kappa_h\kappa_{f/h}}{n}}L_h\sum_{j=1}^n \|x_t - \phi^t_j\| + 2L_h\|\phi^t_{i_t} - x_t\| \\ &+ \left(\frac{1}{4nL_{f/h}} + \frac{2}{L_{f/h}}\sqrt{\frac{\kappa_h\kappa_{f/h}}{n}}\right)\|\sum_{j=1}^n \nabla f_j(\phi^t_j)\|\Bigg)\end{aligned}$$

$$\leq 1 + C\left(\sum_{j=1}^{n} \|x_t - \phi_j^t\| + \|\sum_{j=1}^{n} \nabla f_j(\phi_j^t)\|\right)$$

where C is defined as

$$C = 2ML_h \max\left(4L_h\left(1+\sqrt{\frac{\kappa_h \kappa_{f/h}}{n}}\right), \frac{1}{L_{f/h}}\left(\frac{1}{4n} + 2\sqrt{\frac{\kappa_h \kappa_{f/h}}{n}}\right)\right).$$

Now, with such choice of G_t, Theorem 6.8 applies and the convergence rate (6.21) holds. It remains to prove the estimate for the convergence rate of G_t towards 1. To this end, we show that it is upper bounded by $\mathcal{O}(1+\psi_t^{1/2})$ since

$$\begin{aligned}
1 + C\Bigg(\sum_{j=1}^{n} \|x_t - \phi_j^t\| &+ \|\sum_{j=1}^{n} \nabla f_j(\phi_j^t)\|\Bigg) \\
&\leq 1 + C\left(\sum_{j=1}^{n} \|x_t - \phi_j^t\| + \|\sum_{j=1}^{n} \nabla f_j(x_t)\| + \sum_{j=1}^{n} \|\nabla f_j(\phi_j^t) - \nabla f_j(x_t)\|\right) \\
&\leq 1 + C\left(\sum_{j=1}^{n} (1 + L_h L_{f/h})\|x_t - \phi_j^t\| + n\|\nabla f(x_t)\|\right) \\
&\leq 1 + C\left(\sum_{j=1}^{n} (1 + L_h L_{f/h})\left(\|x_t - x^*\| + \|x^* - \phi_j^t\|\right) + nL_h L_{f/h}\|x_t - x_*\|\right) \\
&\leq 1 + C\left(n(1 + 2L_h L_{f/h}\|x_t - x^*\| + \sum_{j=1}^{n} (1 + L_h L_{f/h})\|x^* - \phi_j^t\|\right) \\
&\leq 1 + C\Bigg(n(1 + 2L_h L_{f/h})\sqrt{\frac{2}{\mu_h} D_h(x^*, x_t)} \\
&\qquad + \sum_{j=1}^{n} (1 + L_h L_{f/h})\sqrt{\frac{2}{\mu_h \mu_{f/h}} D_{f_j}(\phi_j^t, x^*)}\Bigg) \\
&= 1 + \mathcal{O}\left(\sqrt{D_h(x^*, x_t)} + \sum_{j=1}^{n} \sqrt{D_{f_j}(\phi_j^t, x^*)}\right) \\
&= 1 + \mathcal{O}\left(\sqrt{\psi_t}\right)
\end{aligned} \tag{6.23}$$

Since we imposed a safeguard such that $G_t \geq \frac{L_{f/h} L_h}{\mu_h}$, the convergence rate of ψ_t is bounded by

$$\mathbb{E}\left[\psi_t\right] = \mathcal{O}\left(1 - \min\left(\frac{1}{8\kappa_h \kappa_{f/h}}, \frac{1}{2n}\right)\right)^t$$

as stated by Corollary 6.10. Indeed, the assumptions are verified as h is L_h-smooth and μ_h-strongly convex. This worst-case estimate for ψ_t, along with the majorization (6.23), gives the resulting rate for G_t. ∎

Conclusion and Perspectives

In this thesis, we studied several aspects of Bregman methods for relatively-smooth optimization. We focused on practical applications to low-rank problems, extensions to stochastic variants for large-scale problems from machine learning and signal processing, as well as questions of theoretical complexity and acceleration. Yet, there are many open problems and research directions left to explore.

Homogeneity and additional regularity assumptions. As mentioned throughout this thesis, a central issue in the analyses and extensions of Bregman methods is the lack of homogeneity and translational invariance of the Bregman divergence D_h. This problem appears for acceleration [Hanzely et al., 2021], as well as for variance reduced stochastic methods (Chapter 6). In fact, it seems to be an obstacle for every algorithm that has *memory* and combines gradients taken at different points.

The lower bound from Chapter 3 demonstrates that that, in the worst case, it is indeed hopeless to prove accelerated rates for Bregman methods. The corresponding worst-case instance involves pathological functions that a nearly nonsmooth (for which the Bregman divergence is highly non-homogeneous) and that are not quite representative of usual applications. This shows that, in order to study methods with memory, additional regularity assumptions are needed. Several possibilities are available, such as Lipschitz regularity of $\nabla^2 h^*$, or self-concordance ideas [Sun and Tran-Dinh, 2018]. The study on the specific setting of entropy (Chapter 5) can also be pursued.

Despite these theoretical difficulties, numerical experiments show good performance of both accelerated variants [Hendrikx et al., 2020] and variance reduction (Chapter 6). While the current analyses predict that these methods only work in a neighborhood of the optimum, or with asymptotical learning rates, a slow transient regime is not observed in experiments. These results should encourage future work towards bridging the gap between theory and practice.

Convergence on the boundary. A less known problem is that the behavior of Bregman methods is not well understood when the optimum x^* lies on the boundary of the set $\mathcal{C}$. In this setting, the $\mathcal{O}(1/k)$ complexity bound does not hold if h is infinite on the boundary (such as for the log kernel). Additionally, the iterate sequence $\{x^k\}_{k\geq 0}$ is not guaranteed to converge, as demonstrated by the counter-example in Bolte and Pauwels [2020]. This is a key theoretical problem and is relevant in many applications, such as Poisson inverse problems on the nonnegative orthant.

Adaptive variants for improving numerical performance. In this thesis, we mostly focused on theoretical analyses of complexity and studied the convergence rate of the algorithms in the *worst-case*. This approach is fruitful, as it has lead to many important advances in first-order optimization [Nemirovski and Yudin, 1983, Nesterov, 1983].

A complementary approach is to improve practical performance of these methods through adaptiveness. Indeed, in some situations, the relative smoothness inequality is too conservative and Bregman methods exhibit slow convergence speed (see the comparison between BGD and Lucy-Richardson in Chapter 6). In the future, it should be interesting to study methods for automatically adapting the Bregman geometry to the objective, in the spirit of adaptive steps sizes [Duchi et al., 2011] or Quasi-Newton methods [Dennis and Moré, 1974].

Bibliography

Masoud Ahookhosh, Le Thi Khanh Hien, Nicolas Gillis, and Panagiotis Patrinos. Multi-block Bregman proximal alternating linearized minimization and its application to orthogonal nonnegative matrix factorization. *arXiv preprint arXiv:1908.01402*, 2019.

Masoud Ahookhosh, Andreas Themelis, and Panagiotis Patrinos. A Bregman Forward-Backward Linesearch Algorithm for Nonconvex Composite Optimization: Superlinear Convergence to Nonisolated Local Minima. *SIAM Journal on Optimization*, 31(1):653–685, 2021.

Kimon Antonakopoulos, Etis Ensea, and Panayotis Mertikopoulos. An Adaptive Mirror-Prox Algorithm for Variational Inequalities with Singular Operators. In *Advances in Neural Information Processing Systems 32*, 2019.

Kimon Antonakopoulos, E. V. Belmega, and Panayotis Mertikopoulos. Online and Stochastic Optimization beyond Lipschitz Continuity: A Riemannian Approach. In *International Conference on Learning Representations*, 2020.

Alfred Auslender and Marc Teboulle. Interior Gradient and Proximal Methods for Convex and Conic Optimization. *SIAM Journal on Optimization*, 16(3):697–725, 2006.

Francis Bach. Duality Between Subgradient and Conditional Gradient Methods. *SIAM Journal on Imaging Sciences*, 25(1):115–129, 2015.

Francis Bach and Eric Moulines. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Advances in neural information processing systems 24 (NIPS 2011)*, 2011.

Mathieu Barré, Adrien Taylor, and Alexandre D'Aspremont. Complexity Guarantees for Polyak Steps with Momentum. In *Thirty Third Conference on Learning Theory (COLT 33)*, volume 125. PMLR, 2020.

Heinz H. Bauschke. Joint minimization with alternating Bregman proximity operators. *Pacific Journal of Optimization*, 2(3):401–424, 2006.

Heinz H. Bauschke and Jonathan M. Borwein. Legendre functions and the method of random Bregman projections. *Journal of Convex Analysis*, 4(1):27–67, 1997.

Heinz H. Bauschke and Jonathan M. Borwein. Joint and Separate Convexity of the Bregman Distance. *Studies in Computational Mathematics*, 8:23–36, 2001.

Heinz H. Bauschke and Patrick L. Combettes. The Baillon-Haddad Theorem Revisited. *Journal of Convex Analysis*, 17(3-4):781–787, 2010.

Heinz H. Bauschke and Patrick L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces.* Springer Publishing Company, 2011.

Heinz H. Bauschke, Jérôme Bolte, and Marc Teboulle. A Descent Lemma Beyond Lipschitz Gradient Continuity: First-Order Methods Revisited and Applications. *Mathematics of Operations Research*, 42(2):330–348, 2017.

Heinz H. Bauschke, Minh N. Dao, and Scott B. Lindstrom. Regularizing with Bregman–Moreau envelopes. *SIAM Journal on Optimization*, 28(4):3208–3228, 2018.

Heinz H. Bauschke, Jérôme Bolte, Jiawei Chen, Marc Teboulle, and Xianfu Wang. On Linear Convergence of Non-Euclidean Gradient Methods without Strong Convexity and Lipschitz Gradient Continuity. *Journal of Optimization Theory and Applications*, 182(3):1068–1087, 2019.

Amir Beck and Marc Teboulle. Mirror Descent and Nonlinear Projected Subgradient Methods for Convex Optimization. *Operations Research Letters*, 31(3):167–175, 2003.

Amir Beck and Marc Teboulle. A Fast Iterative Shrinkage-Thresholding Algorithm. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

Aharon Ben-Tal, Tamar Margalit, and Arkadi Nemirovski. The Ordered Subsets Mirror Descent Optimization Method with Applications to Tomography. *SIAM Journal on Optimization*, 12(1):79–108, 2001.

Martin Benning and Erlend Skaldehaug Riis. Bregman Methods for Large-Scale Optimisation with Applications in Imaging. In Springer, editor, *Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging: Mathematical Imaging and Vision*, pages 1–42. 2021.

M Bertero, P Boccaci, G Desidera, and G Vicidomini. Image Deblurring with Poisson Data: From Cells to Galaxies. *Inverse Problems*, 2009.

Srinadh Bhojanapalli, Anastasios Kyrillidis, and Sujay Sanghavi. Dropping Convexity for Faster Semidefinite Optimization. *JMLR: Workshop and Conference Proceedings*, 40:1–53, 2016.

Benjamin Birnbaum, Nikhil R. Devanur, and Lin Xiao. Distributed algorithms via gradient descent for fisher markets. In *ACM Conference on Electronic Commerce*, pages 127–136, 2011.

Jérôme Bolte and Edouard Pauwels. Curiosities and counterexamples in smooth convex optimization. *arXiv preprint arXiv:2001.07999*, 2020.

Jérôme Bolte, Aris Daniilidis, and Adrian Lewis. The Łojasiewicz Inequality for Nonsmooth Subanalytic Functions with Applications to Subgradient Dynamical Systems. *SIAM Journal on Optimization*, 17 (4):1205–1223, 2007.

Jérôme Bolte, Shoham Sabach, Marc Teboulle, and Yakov Vaisbourd. First Order Methods Beyond Convexity and Lipschitz Gradient Continuity with Applications to Quadratic Inverse Problems. *SIAM Journal on Optimization*, 28(3):2131–2151, 2018.

Léon Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*. Springer, 2012.

Lev M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200–217, 1967.

Sébastien Bubeck. Introduction to online optimization. *Lecture Notes*, 2011.

Sebastien Bubeck. Nemirovski's acceleration, 2019. URL https://blogs.princeton.edu/imabandit/2019/01/09/nemirovskis-acceleration/.

Minh N. Bùi and Patrick L. Combettes. Bregman Forward-Backward Operator Splitting. *arXiv preprint arXiv:1908.03878*, 2019.

Samuel Burer and Renato D C Monteiro. Local Minima and Convergence in Low-Rank Semidefinite Programming. *Mathematical Programming*, 103(3):427–444, 2005.

Jian-Feng Cai, Emmanuel J. Candès, and Zuowei Shen. A Singular Value Tresholding Algorithm for Matrix Completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

Emmanuel J. Candès and Benjamin Recht. Exact Matrix Completion Via Convex Optimization. *Foundations of Computational Mathematics*, 9(6), 2009.

Emmanuel J. Candès, Xiaodong Li, and Mahdi Soltanolkotabi. Phase retrieval via wirtinger flow: Theory and algorithms. *IEEE Transactions on Information Theory*, 61(4):1985–2007, 2015.

Yair Censor and Stavros A. Zenios. Proximal Minimization Algorithm with D-functions. *Journal of Optimization Theory and Applications*, 73(3):451–464, 1992.

Gong Chen and Marc Teboulle. Convergence Analysis of a Proximal-Like Minimization Algorithm Using Bregman Functions. *SIAM Journal on Optimization*, 3(3):538–543, 1993.

Yudong Chen and Martin J. Wainwright. Fast low-rank estimation by projected gradient descent: General statistical and algorithmic guarantees. *arXiv preprint arXiv:1509.03025*, 2015.

Lénaïc Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Scaling algorithms for unbalanced optimal transport problems. *Mathematics of Computation*, 87(314), 2018.

Andrzej Cichocki and Anh Huy Phan. Fast local algorithms for large scale nonnegative matrix and tensor factorizations. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 2009.

Michael B. Cohen, Aaron Sidford, and Kevin Tian. Relative Lipschitzness in Extragradient Methods and a Direct Recipe for Acceleration. *arXiv preprint arXiv:2011.06572*, 2020.

Patrick L. Combettes and Valérie R. Wajs. Signal recovery by proximal forward-backward splitting. *SIAM Multiscale Modeling and Simululation*, 4:1168–200, 2005.

Alexandre D'Aspremont, Damien Scieur, and Adrien Taylor. Acceleration Methods. *arXiv preprint arXiv:2101.09545*, 2021.

Damek Davis, Dmitriy Drusvyatskiy, and Kellie J. MacPhee. Stochastic model-based minimization under high-order growth. *arXiv preprint arXiv:1807.00255*, 2018.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives. pages 1–15, 2014.

J.e. Jr. Dennis and Jorge J. Moré. Quasi-Newton Methods, Motivation and Theory. *SIAM Review*, 19 (1), 1974.

Chris Ding, Xiaofeng He, and Horst Simon. On the Equivalence of Nonnegative Matrix Factorization and Spectral Clustering. *Proceedings of the 2005 SIAM ICDM*, (4):126–135, 2005.

Ivan Dokmanic, Reza Parhizkar, Juri Ranieri, and Martin Vetterli. Euclidean Distance Matrices: Essential theory, algorithms, and applications. *IEEE Signal Processing Magazine*, 32(6):12–30, 2015.

Radu-Alexandru Dragomir, Alexandre D'Aspremont, and Jérôme Bolte. Quartic First-Order Methods for Low-Rank Minimization. *Journal of Optimization Theory and Applications*, 189(2), 2021a.

Radu-Alexandru Dragomir, Mathieu Even, and Hadrien Hendrikx. Fast Stochastic Bregman Gradient Methods: Sharp Analysis and Variance Reduction. *arXiv preprint arXiv:2104.09813. To appear in International Conference on Machine learning*, 2021b.

Radu-Alexandru Dragomir, Adrien Taylor, Alexandre D'Aspremont, and Jérôme Bolte. Optimal Complexity and Certification of Bregman First-Order Methods. *Mathematical Programming*, 2021c.

Yoel Drori. The Exact Information-Based Complexity of Smooth Convex Minimization. *Journal of Complexity*, 39:1–16, 2017.

Yoel Drori and Ohad Shamir. The Complexity of Finding Stationary Points with Stochastic Gradient Descent. *arXiv preprint arXiv:1910.01845*, 2019.

Yoel Drori and Adrien B. Taylor. Efficient First-Order Methods for Convex Minimization: a Constructive Approach. *Mathematical Programming*, 2019.

Yoel Drori and Marc Teboulle. Performance of First-Order Methods for Smooth Convex Minimization: A Novel Approach. *Mathematical Programming*, 145(1-2):451–482, 2014.

Yoel Drori and Marc Teboulle. An Optimal Variant of Kelley's Cutting-Plane Method. *Mathematical Programming*, 160(1-2):321–351, 2016.

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

John C. Duchi, Shai Shalev-Shwartz, Yoram Singer, and Ambuj Tewari. Composite objective mirror descent. *23rd Conference on Learning Theory*, 2010.

Jonathan Eckstein. Nonlinear Proximal Point Algorithms Using Bregman Functions, with Applications to Convex Programming. *Mathematics of Operations Research*, 18(1):202–226, 1993.

Haw Ren Fang and Dianne P. O'Leary. Euclidean distance matrix completion problems. *Optimization Methods and Software*, 27(4):695–717, 2012.

Tianxiang Gao, Songtao Lu, Jia Liu, and Chris Chu. Randomized bregman coordinate descent methods for Non-Lipschitz optimization. *arXiv preprint arXiv:2001.05202*, 2020.

Rong Ge, Jason D. Lee, and Tengyu Ma. Matrix Completion has No Spurious Local Minimum. In *Advances in Neural Information Processing Systems*, pages 2973–2981, 2016.

Tom Goldstein and Stanley Osher. The split Bregman method for L1-regularized problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009.

Robert Mansel Gower, Nicolas Loizou, Xun Qian, Alibek Sailanbayev, Egor Shulgin, and Peter Richtárik. Sgd: General analysis and improved rates. In *International Conference on Machine Learning*, pages 5200–5209. PMLR, 2019.

G. N. Grapiglia and Yurii Nesterov. On inexact solution of auxiliary problems in tensor methods for convex optimization. *Optimization Methods and Software*, 36(1):145–170, 2021.

Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. In *35th International Conference on Machine Learning*, 2018.

David H. Gutman and Javier F. Peña. The condition number of a function relative to a set. *Mathematical Programming*, 2020.

Cristóbal Guzmán and Arkadi Nemirovski. On Lower Complexity Bounds for Large-Scale Smooth Convex Optimization. *Journal of Complexity*, 31(1):1–14, 2015.

Linus Hamilton and Ankur Moitra. No-go Theorem for Acceleration in the Hyperbolic Plane. *arXiv preprint arXiv:2101.05657*, 2021.

Filip Hanzely and Peter Richtárik. Fastest Rates for Stochastic Mirror Descent Methods. *arXiv preprint arXiv:1803.07374*, 2018.

Filip Hanzely, Peter Richtarik, and Lin Xiao. Accelerated Bregman Proximal Gradient Methods for Relatively Smooth Convex Optimization. *Computational Optimization and Applications*, 2021.

Elad Hazan. The Convex Optimization Approach to Regret Minimization. In Suvrit Sra Wright, Sebastian Nowozin, and Stephen J., editors, *Optimization for Machine Learning*, pages 287–303. MIT Press, 2011.

Zhaoshui He, Shengli Xie, Rafal Zdunek, Guoxu Zhou, and Andrzej Cichocki. Symmetric nonnegative matrix factorization: Algorithms and applications to probabilistic clustering. *IEEE Transactions on Neural Networks*, 22(12):2117–2131, 2011.

Hadrien Hendrikx, Francis Bach, and Laurent Massoulié. Dual-Free Stochastic Decentralized Optimization with Variance Reduction. Number 34th Conference on Neural Information Processing Systems (NeurIPS 2020).

Hadrien Hendrikx, Lin Xiao, Sébastien Bubeck, Francis Bach, and Laurent Massoulié. Statistically preconditioned accelerated gradient method for distributed optimization. In *International Conference on Machine Learning*, number 119, pages 4203—-4227, 2020.

Thomas Hofmann, Aurelien Lucchi, Simon Lacoste-Julien, and Brian Mcwilliams. Variance reduced stochastic gradient descent with neighbors. In *Advances in Neural Information Processing Systems*, 2015.

Alfredo N. Iusem, B. F. Svaiter, and Marc Teboulle. Entropy-Like Proximal Methods in Convex Programming. *Mathematics of Operations Research*, 19(4):790–814, 1994.

Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank Matrix Completion using Alternating Minimization. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, pages 665—-674, 2013.

Stefan Karpinski Jeff Bezanson, Alan Edelman and Viral B. Shah. Julia : A Fresh Approach to Numerical Computing. *SIAM Review*, 59(1):65–98, 2017.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in Neural Information Processing Systems*, 2013.

Anatoli Juditsky and Arkadi Nemirovski. First Order Methods for Nonsmooth Convex Large-Scale Optimization , I : General Purpose Methods. In Suvrit Sra Wright, Sebastian Nowozin, and Stephen J., editors, *Optimization for Machine Learning*, pages 121–147. MIT Press, 2011.

Avinash C. Kak and Malcolm Slaney. *Principles of computerized tomographic imaging*. SIAM, 2001.

Chao Kan and Wen Song. The Moreau envelope function and proximal mapping in the sense of the Bregman distance. *Nonlinear Analysis, Theory, Methods and Applications*, 75(3):1385–1399, 2012.

Donghwan Kim. Accelerated Proximal Point Method and Forward Method for Monotone Inclusions. *arXiv preprint arXiv:1905.05149v2*, 2019.

Donghwan Kim and Jeffrey A. Fessler. Optimized First-Order Methods for Smooth Convex Minimization. *Mathematical Programming*, 159(1-2):81–107, 2016.

Jingu Kim and Haesun Park. Fast Nonnegative Matrix Factorization: An Active-set-like Method and Comparisons. *SIAM Journal on Scientific Computing*, 33(6):3261–3281, 2013.

Jyrki Kivinen and Manfred K. Warmuth. Exponentiated Gradient versus Gradient Descent for Linear Predictors. *Information and Computation*, 132(1), 1997.

Krzysztof C. Kiwiel. Proximal minimization methods with generalized Bregman functions. *SIAM Journal on Control and Optimization*, 35(4):1142–1168, 1997.

N Krislock and H Wolkowicz. Euclidean Distance Matrices and Applications. *Handbook on Semidefinite, Cone and Polynomial Optimization*, (2009-06):879–914, 2011.

Da Kuang, Sangwoon Yun, and Haesun Park. SymNMF: nonnegative low-rank approximation of a similarity matrix for graph clustering. *Journal of Global Optimization*, 62(3):545–574, 2015.

Frederik Kunstner, Raunak Kumar, and Mark Schmidt. Homeomorphic-Invariance of EM: Non-Asymptotic Convergence in KL Divergence for Exponential Families via Mirror Descent. In *International Conference on Machine Learning*, volume 130, 2020.

Guanghui Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1-2):365–397, 2012.

Guanghui Lan. Policy Mirror Descent for Reinforcement Learning: Linear Convergence, New Sampling Complexity, and Generalized Problem Classes. *arXIv preprint arXiv:2102.00135*, 2021.

Puya Latafat, Andreas Themelis, Masoud Ahookhosh, and Panagiotis Patrinos. Bregman finito/miso for nonconvex regularized finite sum minimization without lipschitz gradient continuity. *arXiv preprint arXiv:2102.10312*, 2021.

Emanuel Laude, Peter Ochs, and Daniel Cremers. Bregman Proximal Mappings and Bregman-Moreau Envelopes under Relative Prox-Regularity. *Journal of Optimization Theory and Applications*, 184: 724–761, 2020.

Flavien Léger. A Gradient Descent Perspective on Sinkhorn. *Applied Mathematics and Optimization*, 2020.

Chih-Jen Lin. Projected Gradient Methods for Nonnegative Matrix Factorization. *Neural Computation*, 2007.

J. Lofberg. YALMIP : A Toolbox for Modeling and Optimization in MATLAB. In *In Proceedings of the CACSD Conference*, 2004.

Haihao Lu. “Relative Continuity” for Non-Lipschitz Nonsmooth Convex Optimization Using Stochastic (or Deterministic) Mirror Descent. *INFORMS Journal on Optimization*, 1(4), 2019.

Haihao Lu, Robert M. Freund, and Yurii Nesterov. Relatively-Smooth Convex Optimization by First-Order Methods, and Applications. *SIAM Journal on Optimization*, 28(1):333–354, 2018.

Songtao Lu, Mingyi Hong, and Zhengdao Wang. A Nonconvex Splitting Method for Symmetric Nonnegative Matrix Factorization : Convergence Analysis and Optimality. *IEEE Transactions on Signal Processing*, 65(12):2572–2576, 2017.

Chris J. Maddison, Daniel Paulin, Yee Whye The, and Arnaud Doucet. Dual Space Preconditioning for Gradient Descent. *SIAM Journal on Optimization*, 31(1), 2021.

Raghu Meka, Prateek Jain, and Inderjit S. Dhillon. Guaranteed Rank Minimization via Singular Value Projection. In *Advances in Neural Information Processing Systems 23 (NIPS 2010)*, 2010.

Konstantin Mishchenko. Sinkhorn Algorithm as a Special Case of Stochastic Mirror Descent. *arXiv preprint arXiv:1909.06918*, 2019.

Bamdev Mishra, Gilles Meyer, Silvère Bonnabel, and Rodolphe Sepulchre. Fixed-rank matrix factorizations and Riemannian low-rank optimization. *Computational Statistics*, 29(3-4):591–621, 2014.

Bandev Mishra, Gilles Meyer, and Rodolphe Sepulchre. Low-rank optimization for distance matrix completion. In *Proceedings of the IEEE Conference on Decision and Control*, pages 4455–4460, 2011.

Jean-Jacques Moreau. Proximité et dualité dans un espace hilbertien. *Bull. Soc. Math. Fr.*, 93(2): 273–299, 1965.

APS Mosek. The MOSEK optimization toolbox for MATLAB manual. Version 9.0., 2019. URL http://docs.mosek.com/9.0/toolbox/index.html.

Mahesh Chandra Mukkamala and Peter Ochs. Beyond alternating updates for matrix factorization with inertial Bregman Proximal Gradient algorithms. In *Advances in Neural Information Processing Systems 32*, 2019.

Mahesh Chandra Mukkamala, Felix Westerkamp, Emanuel Laude, Daniel Cremers, and Peter Ochs. Bregman proximal framework for deep linear neural networks. *arXiv preprint arXiv:1910.03638*, 2019.

Mahesh Chandra Mukkamala, Peter Ochs, Thomas Pock, and Shoham Sabach. Convex-Concave Backtracking for Inertial Bregman Proximal Gradient Algorithms in Non-Convex Optimization. *SIAM Journal on Mathematics of Data Science*, 2(3):658–682, 2020.

Arkadi Nemirovski. Efficient methods for large-scale convex optimization problems. *Ekonomika i Matematicheskie Metody*, 15, 1979.

Arkadi Nemirovski and David B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley–Blackwell, 1983.

Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

Yurii Nesterov. A Method of Solving A Convex Programming Problem With Convergence rate O(1/kˆ2). *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Publishing Company, Inc, 1 edition, 2003.

Yurii Nesterov. Gradient methods for minimizing composite objective function. *CORE Report*, 2007.

Yurii Nesterov. Superfast second-order methods for Unconstrained Convex Optimization. *CORE Discussion Paper*, 2020.

Yurii Nesterov. Implementable tensor methods in unconstrained convex optimization. *Mathematical Programming*, 186(1):157–183, 2021.

John Ollinger. Maximum-likelihood reconstruction of transmission images in emission computed tomography via the EM algorithm. *IEEE Transactions on Medical Imaging*, 13, 1994.

Stanley Osher, Martin Burger, Donald Goldfarb, Jinjun Xu, and Wotao Yin. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling and Simulation*, 4(2):460–489, 2005.

Dohyung Park, Anastasios Kyrillidis, Constantine Caramanis, and Sujay Sanghavi. Finding low-rank solutions to matrix problems , efficiently and provably. *arXiv preprint arXiv:1606.03168v1*, 2016.

David Pfau. A generalized bias-variance decomposition for bregman divergences. *Unpublished Manuscript*, 2013. URL http://davidpfau.com/assets/generalized_bvd_proof.pdf.

Hou Duo Qi and Xiaoming Yuan. Computing the nearest Euclidean distance matrix with low embedding dimensions. *Mathematical Programming*, 147(1-2):351–389, 2013.

Benjamin Recht, Maryam Fazel, and Pablo A. Parrilo. Guaranteed Minimum-Rank Solutions of Linear Matrix Equations via Nuclear Norm Minimization. *SIAM Review*, 52(3):471–501, 2007.

R. Tyrrell Rockafellar. *Convex Analysis*. 1970.

Manon Romain and Alexandre D'Aspremont. A Bregman Method for Structure Learning on Sparse Directed Acyclic Graphs. *arXiv preprint arXiv:2011.02764*, 2020.

Ernest K. Ryu, Adrien B. Taylor, Carolina Bergeling, and Pontus Giselsson. Operator Splitting Performance Estimation: Tight Contraction Factors and Optimal parameter Selection. *SIAM Journal on Optimization*, 30(3), 2020.

Mark Schmidt, Nicolas Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162, 09 2013.

Shai Shalev-Shwartz. SDCA without duality, regularization, and individual convexity. In *International Conference on Machine Learning*, 2016.

Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 2013.

Ohad Shamir, Nati Srebro, and Tong Zhang. Communication-efficient distributed optimization using an approximate Newton-type method. In *International Conference on Machine Learning*, 2014.

L. A. Shepp and Y. Vardi. Maximum likelihood reconstruction for emission tomography. *IEEE Transactions on Medical Imaging*, 1(2), 1982.

Zhan Shi, Xinhua Zhang, and Yaoliang Yu. Bregman divergence for stochastic variance reduction: Saddle-point and adversarial prediction. In *Advances in Neural Information Processing Systems*, 2017.

Fedor Stonyakin, Alexander Tyurin, Alexander Gasnikov, Pavel Dvurechensky, Artem Agafonov, Darina Dvinskikh, Dmitry Pasechnyuk, Sergei Artamonov, and Victorya Piskunova. Inexact relative smoothness and strong convexity for optimization and variational inequalities by inexact model. *arXiv preprint arXiv:2001.09013*, 2020.

Ruoyu Sun and Zhi-Quan Luo. Guaranteed Matrix Completion via Nonconvex Factorization. *IEEE Transactions on Information Theory*, 62(11):6535–6579, 2016.

Tianxiao Sun and Quoc Tran-Dinh. Generalized self-concordant functions: a recipe for Newton-type methods. *Mathematical Programming*, 2018.

Adrien Taylor and Francis Bach. Stochastic first-order methods: non-asymptotic and computer-aided analyses via potential functions. In *Thirty-Second Conference on Learning Theory (COLT 32)*, volume 99. PMLR, 2019.

Adrien Taylor, Julien Hendrickx, and François Glineur. Exact Worst-Case Performance of First-Order Methods for Composite Convex Optimization. *SIAM Journal on Optimization*, 27:1600–1621, 2015.

Adrien B. Taylor, Julien M. Hendrickx, and François Glineur. Smooth Strongly Convex Interpolation and Exact Worst-Case Performance of First-Order Methods. *Mathematical Programming*, 161(1-2): 307–345, 2017.

Adrien B. Taylor, Julien M. Hendrickx, and Francois Glineur. Performance Estimation Toolbox (PESTO): Automated Worst-Case Analysis of First-Order Optimization Methods. In *IEEE 56th Annual Conference on Decision and Control (CDC 2017)*, pages 1278–1283, 2018.

Marc Teboulle. Entropic Proximal Mappings with Applications to Nonlinear Programming. *Mathematics of Operations Research*, 17(3):670–690, 1992.

Marc Teboulle. A simplified view of first order methods for optimization. *Mathematical Programming*, 170(1):67–96, 2018.

Marc Teboulle and Yakov Vaisbourd. Novel proximal gradient methods for nonnegative matrix factorization with sparsity constraints. *SIAM Journal on Imaging Sciences*, 13(1):381–421, 2019.

Manan Tomar, Lior Shani, Yonathan Efroni, and Mohammad Ghavamzadeh. Mirror Descent Policy Optimization. *arXiv preprint arXiv:2005.09814*, 2020.

Stephen Tu, Ross Boczar, Max Simchowitz, Mahdi Soltanolkotabi, and Benjamin Recht. Low-rank Solutions of Linear Matrix Equations via Procrustes Flow. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, pages 964–973, 2016.

Arnaud Vandaele, Nicolas Gillis, Qi Lei, Kai Zhong, and Inderjit Dhillon. Efficient and non-convex coordinate descent for symmetric nonnegative matrix factorization. *IEEE Transactions on Signal Processing*, 64(21):5571–5584, 2016.

Lieven Vandenberghe and Stephen Boyd. Semidefinite Programming. *SIAM Review*, 38(1):49–45, 1996.

Krichene Walid, Alexandre Bayen, and Peter L. Bartlett. Accelerated Mirror Descent In Continuous and Discrete Time. In *Advances in Neural Information Processing Systems 28*, pages 2845—-2853, 2015.

Blake Woodworth and Nathan Srebro. Lower Bound for Randomized First Order Convex Optimization. *arXiv preprint arXiv:1709.03594*, 2017.

Wotao Yin, Stanley Osher, Donald Goldfarb, and Jerome Darbon. Bregman iterative algorithms for l1-minimization with applications to compressed sensing. *SIAM Journal on Imaging Sciences*, 1(1): 143–168, 2008.

Hongyi Zhang and Suvrit Sra. An Estimate Sequence for Geodesically Convex Optimization. In *31st Conference On Learning Theory*, 2018.

Siqi Zhang and Niao He. On the convergence rate of stochastic mirror descent for nonsmooth nonconvex optimization. *arXiv preprint arXiv:1806.04781*, 2018.

Xiaoya Zhang, Roberto Barrio, M. Angeles Martínez, Hao Jiang, and Lizhi Cheng. Bregman proximal gradient algorithm with extrapolation for a class of nonconvex nonsmooth minimization problems. *IEEE Access*, 7, 2019.

Tuo Zhao, Zhaoran Wang, and Han Liu. A Nonconvex Optimization Framework for Low Rank Matrix Estimation. In *Advances in Neural Information Processing Systems 28*, pages 559–567, 2015.

Qinqing Zheng and John Lafferty. A Convergent Gradient Descent Algorithm for Rank Minimization and Semidefinite Programming from Random Linear Measurements. In *Advances in Neural Information Processing Systems 28*, 2015.

Zhengyuan Zhou, Panayotis Mertikopoulos, Nicholas Bambos, Stephen Boyd, and Peter Glynn. Stochastic mirror descent in variationally coherent optimization problems. In *Advances in Neural Information Processing Systems*, 2017.

Zhihui Zhu, Xiao Li, Kai Liu, and Qiuwei Li. Dropping Symmetry for Fast Symmetric Nonnegative Matrix Factorization. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, 2018.