Regularized Risk Minimization by Nesterov's Accelerated Gradient Methods: Algorithmic Extensions and Empirical Studies

Xinhua Zhang XINHUA.ZHANG.CS@GMAIL.COM
Department of Computing Science, University of Alberta, Edmonton, AB T6G 2E8, Canada

Ankan Saha ANKANS@CS.UCHICAGO.EDU
Department of Computer Science, University of Chicago, Chicago, IL 60637, USA

S.V.N. Vishwanathan VISHY@STAT.PURDUE.EDU
Department of Statistics and Department of Computer Science, Purdue University, IN 47906, USA

Abstract

Nesterov's accelerated gradient methods (AGM) have been successfully applied in many machine learning areas. However, their empirical performance on training max-margin models has been inferior to existing specialized solvers. In this paper, we first extend AGM to strongly convex and composite objective functions with Bregman style prox-functions. Our unifying framework covers both the ∞-memory and 1-memory styles of AGM, tunes the Lipschiz constant adaptively, and bounds the duality gap. Then we demonstrate various ways to apply this framework of methods to a wide range of machine learning problems. Emphasis will be given on their rate of convergence and how to efficiently compute the gradient and optimize the models. The experimental results show that with our extensions AGM outperforms state-of-the-art solvers on max-margin models.

1. Introduction

There has been an explosion of interest in machine learning over the past decade, much of which has been fueled by the phenomenal success of binary Support Vector Machines (SVMs). Driven by numerous applications, recently, there has been increasing interest in support vector learning with linear models. At the heart of SVMs is the following regularized risk minimization (RRM) problem:

$$\min_{\mathbf{w}} J(\mathbf{w}) := \underbrace{\lambda\Omega(\mathbf{w})}_{\text{regularizer}} + \underbrace{R_{\text{emp}}(\mathbf{w})}_{\text{empirical risk}} \tag{1}$$

$$\text{with} \quad \Omega(\mathbf{w}) := \frac{1}{2}\|\mathbf{w}\|_2^2 \tag{2}$$

$$R_{\text{emp}}(\mathbf{w}) := \frac{1}{n}\max_{b\in\mathbb{R}} \sum_{i=1}^{n} [1 - y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b)]_+, \tag{3}$$

where $[x]_+ = x$ if $x \geq 0$ and 0 otherwise. Here we assume access to a training set of n labeled examples $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ where $\mathbf{x}_i \in \mathbb{R}^p$ and $y_i \in \{-1, +1\}$, and use the half square Euclidean norm $\|\mathbf{w}\|_2^2 = \sum_i w_i^2$ as the regularizer. The parameter λ controls the trade-off between the empirical risk and the regularizer.

There has been significant research devoted to developing specialized optimizers which minimize $J(\mathbf{w})$ efficiently. Zhang et al. [1] proved that cutting plane and bundle methods may require at least $O(np/\epsilon)$ computational efforts to find an ϵ accurate solution to (1), and they suggested using Nesterov's accelerated gradient method (AGM) which provably costs $O(np/\sqrt{\epsilon})$ time complexity. In general, AGM takes $O(1/\sqrt{\epsilon})$ times of gradient query to find an ϵ accurate solution to

$$\min_{\mathbf{x}\in Q} f(\mathbf{x}), \tag{4}$$

where f is convex and has L-Lipschitz continuous gradient (L-*l.c.g*), and Q is a closed convex set in the Euclidean space. AGM is especially suitable for large scale optimization problems because each iteration it only requires the gradient of f.

Unfortunately, despite some successful application of AGM in learning sparse models [2, 3] and game playing

[4], it does not compare favorably to existing specialized optimizers when applied to training large margin models [5]. It turns out that special structures exist in those problems, and to make full use of AGM, one must utilize the computational and statistical properties of the learning problem by properly reformulating the objectives and tailoring the optimizers accordingly.

To this end, our first contribution is to show that in both theory and practice smoothing $R_{\text{emp}}(\mathbf{w})$ as in [6] is advantageous to the primal-dual versions of AGM. The dual of (1) is

$$\max_{\boldsymbol{\alpha}} D(\boldsymbol{\alpha}) = \sum_i \alpha_i - \frac{1}{2\lambda}\boldsymbol{\alpha}^\top Y X^\top X Y \boldsymbol{\alpha}, \tag{5}$$

$$s.t.\ \ \boldsymbol{\alpha} \in Q_2 := \left\{\boldsymbol{\alpha} \in [0, n^{-1}]^n : \sum_i y_i \alpha_i = 0\right\}. \tag{6}$$

Comparing (4) with (1) and (5), it seems more natural to apply AGM to (5) because it is smooth. However in practice, most α_i at the optimum will be on the boundary of $[0, n^{-1}]$. According to [7], such α_i's are easy to identify and so the corresponding entries in the gradient are wasted by AGM. This structure of support vector is unique for max-margin models, which will also be manifested in our experiments (Section 6).

In contrast, smoothing R_{emp} has a lot of advantages. First, it directly optimizes in the primal J, avoiding the indirect translation from the dual solution to the primal. Second, the resulting optimization problem is unconstrained. If Ω is strongly convex, then linear convergence can be achieved. Third, gradient of the smoothed $\tilde{R}_{\text{emp}}$ can often be computed efficiently, and details will be given in Section 5.4. Fourth, the diameter of the dual space Q_2 often grows slowly with n, or even decreases. This allows using a loose smoothing parameter. Fifth, in practice most α_i at the optimum are 0, where $\tilde{R}_{\text{emp}}$ best approximates R_{emp}. Therefore, the approximation is actually much tighter than the worst case theoretical bound, and a good solution for $\tilde{R}_{\text{emp}}$ is more likely to optimize R_{emp} too. Last but most important, the smoothed $\tilde{R}_{\text{emp}}$ themselves are reasonable risk measures [8], which also deliver good generalization performance in statistics. Now that it is much easier to optimize the smoothed objectives, a model which generalizes well can be quickly obtained with the homotopy scheme (*i.e.* anneal the smoothing parameter).

Using the same idea of smoothing R_{emp}, AGM can be applied to a much wider variety of RRM problems by utilizing its composite structure. Given a model ψ of $\tilde{R}$, if $\Omega(\mathbf{w}) + \psi(\mathbf{w})$ can be solved efficiently, then [9] showed that $\Omega(\mathbf{w}) + \tilde{R}(\mathbf{w})$ can be solved in $O(1/\sqrt{\epsilon})$ steps, even if Ω is not differentiable, *e.g.* L_1 norm [10]. Similar approach is applied to the $L_{1,\infty}$ regularizer and the elastic net [11] regularizer by [12]:

$$\Omega(\mathbf{w}) = \frac{\gamma}{2}\|\mathbf{w}\|_2^2 + \|\mathbf{w}\|_1 = \frac{\gamma}{2}\sum_i w_i^2 + \sum_i |w_i|. \tag{7}$$

This Ω is strongly convex with respect to (wrt) the L_2 norm, and similarly in many RRM problems Ω is strongly convex wrt some norm $\|\cdot\|$. For example, the relative entropy regularizer in boosting [13]:

$$\Omega(\mathbf{w}) = \sum_i w_i \log w_i \tag{8}$$

is strongly convex wrt L_1 norm, and the log determinant of a matrix in [14–16]:

$$\Omega(W) = -\log\det W \tag{9}$$

is strongly convex wrt the Frobenius norm. By exploiting the strong convexity, [17] accelerated the convergence rate from $O(1/\sqrt{\epsilon})$ to $O(\log\frac{1}{\epsilon})$. However, the prox-function in this case must be strongly convex wrt $\|\cdot\|$ too. Existing methods either ignore the strong convexity in Ω [9], or restrict the norm to L_2 [10, 17]. As one major contribution of this paper, we extend AGM to exploit this strong convexity in the context of Bregman divergence. In particular, we allow Ω to be strongly convex wrt a Bregman divergence induced by a smooth convex function d (to be formalized later), where d is in turn strongly convex wrt certain norm $\|\cdot\|$. By using d as a prox-function, we manage to achieve linear convergence for a wide range of RRM problems.

There are two types of first order methods that both achieve the optimal rate. The first type is the original AGM pioneered by Nesterov [6, 17–20], which uses a sequence of **e**stimation **f**unctions (hence we call it AGM-EF). In particular, it uses the whole past iterates to progressively build a sequence of estimate functions which approximate the objective function. The second type was developed by a number of other researchers and a unified treatment was given by [9]. Intuitively, it generalizes the idea of gradient descent by **p**roximal **r**egularization (hence we call it AGM-PR), which can be further accelerated by momentum. Therefore, these two types of methods are different in concept. In addition, both AGM-EF and AGM-PR a ∞-memory version which builds a model of the objective by using *all* the past gradients, and a 1-memory version which approximates that model by a *single* Bregman divergence.

We choose to base our extensions on AGM-EF, because compared with AGM-PR it provides much more

		No composite		Composite	
		cvx	sc	cvx	sc
Euclidean	1-memory	[19]	[19]	×	×
	∞-memory	[6]	[17]	[17]	[17]
Bregman	1-memory	[23]	×	×	×
	∞-memory	[6]	×	×	×

Table 1. Summary of AGM-EF. "sc" means strongly convex and "cvx" means just convex. × means novel contribution of this paper. AGM-PR can handle all but sc.

flexibility in adaptively tuning L.[1] This is because the inductive relationship maintained by AGM-EF involves a single iteration, while that for AGM-PR involves two successive steps. The novelty and generality of our method in the context of existing methods are summarized in Table 1. We further provide bounds on the duality gap which amounts to effective termination criteria. As another important contribution, we derive *linear convergence for the duality gap* in the context of strong convexity. Computationally, at each iteration our method requires only one projection and one gradient evaluation within the feasible region.[2]

Outline of the paper. In Section 2, we follow [24, Section 4.1, Definition 3] to extend the concept of strong convexity to the context of Bregman divergence. We show several properties that will play a key role in the subsequent development of the new algorithms. In Section 3 and 4, two novel variants of AGM-EF are developed along the lines of ∞-memory and 1-memory. They both achieve global linear convergence by utilizing the Bregman generalized strong convexity in either Ω or R_{emp}. Section 5 elaborates on how to *effectively* apply our method to solve Bregman regularized risk minimization problems, and many examples of machine learning models are discussed. Also presented is the algorithms which *efficiently* compute the gradient and solve the model. Experimental results are given in Section 6, where we show empirically that by smoothing R_{emp} and exploiting the generalized strong convexity in Ω, the L_2 and entropic regularized risk minimization problems can be solved significantly faster than the state-of-the-art optimizers.

A ready reckoner of the convex analysis concepts used in the paper can be found in Appendix A.

[1] All APM-PR variants with adaptive L, *e.g.* [9, 10, 21, 22], require the estimate of L grow monotonically through iterations. And their technique does not extend to asymmetric Bregman divergence.

[2] Some AGM algorithms require two projections [6] or two gradients [17] per iteration, or evaluate the gradient outside the feasible region [19, Section 2.2.4].

2. Preliminaries

From the optimization perspective, the objectives considered in this paper have the same form as in [9]. Let $\mathbb{R}^p$ be endowed with a norm $\|\cdot\|$. Consider the following nonsmooth convex objective:

$$\min_{\mathbf{x}} J(\mathbf{x}) = f(\mathbf{x}) + \Psi(\mathbf{x}), \tag{10}$$

where $\Psi : \mathbb{R}^p \mapsto \overline{\mathbb{R}} := (-\infty, +\infty]$ and $f : \mathbb{R}^p \mapsto \overline{\mathbb{R}}$ are proper, lower semicontinuous (lsc) and convex. Assume $\operatorname{dom}\Psi$ is closed, f is differentiable on an open set containing $\operatorname{dom}\Psi$, and ∇f is Lipschitz continuous on $\operatorname{dom}\Psi$, *i.e.* there exists $L > 0$ such that

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|^* \leq L \|\mathbf{x} - \mathbf{y}\| \quad \mathbf{x}, \mathbf{y} \in \operatorname{dom}\Psi.$$

Some special cases are in order. The first is constrained smooth optimization, where Ψ is the indicator function for a nonempty closed convex set $Q \subseteq \mathbb{R}^p$:

$$\Psi(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \in Q \\ +\infty & \text{otherwise} \end{cases}.$$

Therefore, in the sequel we will always discuss unconstrained minimization for $J(\mathbf{w})$, although this is just a matter of notation. A second example is the L_1 regularization, where

$$\Psi(\mathbf{x}) = \sum_{i=1}^{p} |x_i| .$$

In fact, many machine learning problems are special cases of (10) and details can be found in Section 5 and [25, Table 5].

Next, we will present in detail two additional assumptions: strong convexity of f and Ψ in the sense of Bregman divergence, and efficiently solvable ground optimization problems.

2.1. Extending strong convexity to Bregman divergence

Let d be a differentiable and σ strongly convex function with respect to some norm $\|\cdot\|$.[3] Then we can define a Bregman divergence:

$$\Delta_d(\mathbf{x}, \mathbf{y}) = d(\mathbf{x}) - d(\mathbf{y}) - \langle \nabla d(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle .$$

By the definition of σ-sc, we have

$$\Delta_d(\mathbf{x}, \mathbf{y}) \geq \frac{\sigma}{2} \|\mathbf{x} - \mathbf{y}\|^2 , \quad \text{for all } \mathbf{x}, \mathbf{y}.$$

Furthermore, Bregman divergence can be used to generalize the concept of strong convexity [24, Definition 3, Chapter 4].

[3] AGM capitalizes on two properties of the norm: convexity and linearity ($\|c \cdot \mathbf{x}\| = |c| \|\mathbf{x}\|$).

Definition 1 (Strong convexity for Bregman divergence)**.** *A convex function f is said to be λ strongly convex with respect to d (λ-sc wrt d) with $\lambda \geq 0$ if for all $\mathbf{x}$ and $\mathbf{y}$ we have*

$$f(\mathbf{x}) \geq f(\mathbf{y}) + \langle \mathbf{g}, \mathbf{x} - \mathbf{y} \rangle + \lambda \Delta_d(\mathbf{x}, \mathbf{y}) \quad \forall\ \mathbf{g} \in \partial f(\mathbf{y}).$$

If $\lambda > 0$, we say f is strictly strongly convex.

For example, with $d(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|^2$ where the norm is Euclidean, we recover the conventional strong convexity. Here we allow λ to be 0 for a unified exposition, and trivially all convex functions are 0-sc wrt any d. It is noteworthy that Definition 1 preserves some important properties of the conventional strong convexity.

Property 1. *If f is λ-sc wrt d, then f must be $\lambda\sigma$-sc wrt $\|\cdot\|$. Hence for any $\alpha \in [0,1]$ and $\mathbf{x}, \mathbf{y}$, we have*

$$f(\alpha\mathbf{x} + (1-\alpha)\mathbf{y}) \leq \alpha f(\mathbf{x}) + (1-\alpha) f(\mathbf{y}) - \frac{\lambda\sigma}{2}\alpha(1-\alpha)\|\mathbf{x} - \mathbf{y}\|^2.$$

Property 2. *If $\alpha_i \geq 0$ and f_i is λ_i-sc wrt d ($\lambda_i \geq 0$), then $\sum_i \alpha_i f_i$ is $\sum_i \alpha_i \lambda_i$-sc wrt d.*

Property 3. *$d(\mathbf{x})$ is 1-sc wrt d. So by Property 2, $\Delta_d(\mathbf{x}, \mathbf{x}_0)$ is also 1-sc wrt d for any fixed $\mathbf{x}_0$.*

Many problems are constrained to a feasible region Q. In the sequel we will always assume that $Q \subseteq \operatorname{dom} d$ and Q is closed and convex.

Property 4. *Suppose $f : \mathbb{R}^n \mapsto \overline{\mathbb{R}}$ is proper, lsc, and λ-sc wrt d and $\mathbf{x}^* = \operatorname{argmin}_{\mathbf{x}} f(\mathbf{x})$. Then*

$$f(\mathbf{x}) - f(\mathbf{x}^*) \geq \lambda \Delta_d(\mathbf{x}, \mathbf{x}^*) \quad \text{for all } \mathbf{x} \in \operatorname{dom} f.$$

The proof simply uses the definition of λ-sc and the optimality condition of $\mathbf{x}^*$: $\langle \mathbf{g}, \mathbf{x} - \mathbf{x}^* \rangle \geq 0$ for all $\mathbf{g} \in \partial f(\mathbf{x}^*)$ and $\mathbf{x} \in \operatorname{dom} f$.

A direct application of Property 2, 3 and 4 gives a very important inequality which is also used extensively in [9, Property 1] and [26, Lemma 6]:

Property 5. *Suppose f is proper, lsc, and convex with range $\overline{\mathbb{R}}$. Let $\mathbf{x}^* = \operatorname{argmax}_{\mathbf{x}} f(\mathbf{x}) + \Delta_d(\mathbf{x}, \mathbf{x}_0)$, then for all $\mathbf{x}$*

$$f(\mathbf{x}) + \Delta_d(\mathbf{x}, \mathbf{x}_0) \geq f(\mathbf{x}^*) + \Delta_d(\mathbf{x}^*, \mathbf{x}_0) + \Delta_d(\mathbf{x}, \mathbf{x}^*).$$

The following property of Bregman divergence plays a key role in keeping a compact expression of our estimation functions.

Property 6. *For all $\alpha_i \geq 0$ and $\mathbf{x}_i$ in the interior of $\operatorname{dom} d$, define*

$$q(\mathbf{x}) := \langle \mathbf{s}, \mathbf{x} \rangle + \sum_i \alpha_i \Delta_d(\mathbf{x}, \mathbf{x}_i).$$

Then $q(\mathbf{x})$ can be equivalent expressed as

$$q(\mathbf{x}) = a\Delta_d(\mathbf{x}, \mathbf{x}^*) + b,$$

where $a = \sum_i \alpha_i$, $\mathbf{x}^* = \operatorname{argmin}_{\mathbf{x}} q(\mathbf{x})$, and $b = q(\mathbf{x}^*)$. Note $\mathbf{x}^*$ is the unconstrained *minimizer of $q(\mathbf{x})$.*

Proof. By the optimality condition of $\mathbf{x}^*$ we have

$$\left\langle \mathbf{s} + \sum_i \alpha_i (\nabla d(\mathbf{x}^*) - \nabla d(\mathbf{x}_i)), \mathbf{x} - \mathbf{x}^* \right\rangle = \mathbf{0} \quad \forall \mathbf{x}. \tag{11}$$

This equality must be changed to $\geq$ if $\mathbf{x}^*$ is the minimizer of $q(\mathbf{x})$ over a constrained set $Q \subsetneq \operatorname{dom} d$. By definition,

$$q(\mathbf{x}^*) = \langle \mathbf{s}, \mathbf{x}^* \rangle + \sum_i \alpha_i \Delta_d(\mathbf{x}^*, \mathbf{x}_i).$$

Subtracting it from the definition of $q(\mathbf{x})$ we get

$$\begin{aligned} q(\mathbf{x}) &= q(\mathbf{x}^*) + \langle \mathbf{s}, \mathbf{x} - \mathbf{x}^* \rangle \\ &\quad + \sum_i \alpha_i (d(\mathbf{x}) - d(\mathbf{x}^*) - \langle \nabla d(\mathbf{x}_i), \mathbf{x} - \mathbf{x}^* \rangle) \\ &= q(\mathbf{x}^*) - \left\langle \sum_i \alpha_i (\nabla d(\mathbf{x}^*) - \nabla d(\mathbf{x}_i)), \mathbf{x} - \mathbf{x}^* \right\rangle \\ &\quad + \sum_i \alpha_i (d(\mathbf{x}) - d(\mathbf{x}^*) - \langle \nabla d(\mathbf{x}_i), \mathbf{x} - \mathbf{x}^* \rangle) \\ &= q(\mathbf{x}^*) + \left(\sum_i \alpha_i \right) \Delta_d(\mathbf{x}, \mathbf{x}^*). \end{aligned}$$

■

Assumption 1. *In the objective (10), we will assume that f is λ_1-sc and Ψ is λ_2-sc wrt a given d ($\lambda_1, \lambda_2 \geq 0$). Then $f + \Psi$ is λ-sc, where*

$$\lambda := \lambda_1 + \lambda_2.$$

2.2. Assumption on the ground optimization problem

We assume it is possible to efficiently solve the following ground problem:

Assumption 2. *Given an arbitrary linear function $\langle \mathbf{u}, \mathbf{x} \rangle$, $\alpha_i \geq 0$ and $\mathbf{x}_i \in \operatorname{dom} \Psi$ ($i \in [k] := \{1, \ldots, k\}$), assume the following optimization problem can be solved efficiently:*

$$\min_{\mathbf{x}} \langle \mathbf{u}, \mathbf{x} \rangle + b + \sum_{i=1}^{k} \alpha_i \Delta_d(\mathbf{x}, \mathbf{x}_i) + \Psi(\mathbf{x}). \tag{12}$$

For different k, we call the assumption BD-k.

In [18] and [19], the 1-memory AGM-EF for general convex objective assumes BD-1. In [6] and [17], BD-∞ is assumed in the sense that for arbitrary $k < \infty$, (12)

is assumed to be efficiently solvable. In our later 1-memory AGM-EF, we will assume BD-2 if $\lambda_1 > 0$. Although most literature assume BD-1, it is actually not hard to see that extension to BD-2 does not cause any real difficulty. In fact, even BD-∞ is feasible as long as $\sum_i \alpha_i \nabla d(\mathbf{x}_i)$ can be aggregated efficiently (which is often true).

As a direct consequence of BD-1, now that the f in (10) is λ_1-sc and L-*l.c.g*, $J(\mathbf{x})$ can be solved in one step if $L = \sigma\lambda_1$. To see this, by definition for all $\mathbf{x}$

$$f(\mathbf{x}) \le f(\mathbf{x}_0) + \langle \nabla f(\mathbf{x}_0), \mathbf{x} - \mathbf{x}_0 \rangle + \frac{L}{2} \|\mathbf{x} - \mathbf{x}_0\|^2,$$

and

$$\begin{aligned} f(\mathbf{x}) &\ge f(\mathbf{x}_0) + \langle \nabla f(\mathbf{x}_0), \mathbf{x} - \mathbf{x}_0 \rangle + \lambda_1 \Delta_d(\mathbf{x}, \mathbf{x}_0) \\ &\ge f(\mathbf{x}_0) + \langle \nabla f(\mathbf{x}_0), \mathbf{x} - \mathbf{x}_0 \rangle + \frac{\lambda_1 \sigma}{2} \|\mathbf{x} - \mathbf{x}_0\|^2. \end{aligned}$$

So clearly $L \ge \sigma\lambda_1$. If $L = \sigma\lambda_1$, then

$$f(\mathbf{x}) \equiv f(\mathbf{x}_0) + \langle \nabla f(\mathbf{x}_0), \mathbf{x} - \mathbf{x}_0 \rangle + \lambda_1 \Delta_d(\mathbf{x}, \mathbf{x}_0).$$

Hence, $f(\mathbf{x}) + \Psi(\mathbf{x})$ exactly satisfies the precondition of BD-1. Therefore, in the sequel we will assume

$$L > \sigma\lambda_1.$$

$c := \frac{L}{\sigma\lambda_1}$ can be viewed as the condition number.

BD-2 allows us to inductively apply Property 6 to simplify the expression of the following function

$$q_n(\mathbf{x}) := a_0 \Delta_d(\mathbf{x}, \mathbf{x}_0) + \sum_{i=1}^{n} [b_i + \langle \mathbf{u}_i, \mathbf{x} \rangle + a_i \Delta_d(\mathbf{x}, \mathbf{x}_i)]$$

into

$$q_n(\mathbf{x}) = \left(\sum_{i=0}^{n} a_i \right) \Delta_d(\mathbf{x}, \mathbf{x}_n^*) + q_n(\mathbf{x}_n^*), \quad n \ge 1$$

where $\mathbf{x}_n^* = \operatorname{argmin}_{\mathbf{x}} q_n(\mathbf{x})$. Let $q_0(\mathbf{x}) = a_0 \Delta_d(\mathbf{x}, \mathbf{x}_0)$. Then simplify $q_1(\mathbf{x})$ into the sum of a constant and a Bregman divergence by Property 6:

$$\begin{aligned} q_1(\mathbf{x}) &= (a_0 + a_1) \Delta_d(\mathbf{x}, \mathbf{x}_1^*) + q_1(\mathbf{x}_1^*), \\ \mathbf{x}_1^* &= \operatorname*{argmin}_{\mathbf{x}} q_0(\mathbf{x}) + b_1 + \langle \mathbf{u}_1, \mathbf{x} \rangle + a_1 \Delta_d(\mathbf{x}, \mathbf{x}_1), \end{aligned} \tag{13}$$

since $\mathbf{x}_1^*$ can be computed efficiently according to assumption BD-2. Next, $q_2(\mathbf{x})$ can be simplified by using (13) and Property 6 again:

$$\begin{aligned} q_2(\mathbf{x}) &= (a_0 + a_1) \Delta_d(\mathbf{x}, \mathbf{x}_2^*) + q_2(\mathbf{x}_2^*), \\ \mathbf{x}_2^* &= \operatorname*{argmin}_{\mathbf{x}} q_1(\mathbf{x}) + b_2 + \langle \mathbf{u}_2, \mathbf{x} \rangle + a_2 \Delta_d(\mathbf{x}, \mathbf{x}_2). \end{aligned}$$

This incremental scheme is especially useful when the argmin of all $q_k(\mathbf{x})$ is readily available, [*e.g.* 23, Section 5].

Notations. Lower bold case letters (*e.g.*, $\mathbf{x}$, $\boldsymbol{\alpha}$) denote vectors, x_i denotes the i-th component of $\mathbf{x}$, $\mathbf{0}$ refers to the vector with all zero components, $\mathbf{e}_i$ is the i-th coordinate vector (all 0's except 1 at the i-th coordinate) and $\mathcal{S}_n$ refers to the n dimensional simplex $\{\mathbf{x} \in [0,1]^n : \sum_{i=1}^n x_i = 1\}$. Unless specified otherwise, $\langle \cdot, \cdot \rangle$ denotes the Euclidean dot product $\langle \mathbf{x}, \mathbf{w} \rangle = \sum_i x_i w_i$. We denote $\overline{\mathbb{R}} := \mathbb{R} \cup \{\infty\}$, and $[t] := \{1, \ldots, t\}$. From now on, we will always fix the d in the context and omit the subscript d in Δ_d.

We follow the definition of norms in [6] which we recap here. Suppose a finite dimensional real vector space E (*e.g.* $\mathbb{R}^p$) is endowed with a norm $\|\cdot\|$. The space of linear functions on E is called the dual space which we denote as E^*. The norm of E^* is defined as

$$\|\mathbf{s}\|^* := \max_{\mathbf{x} \in E : \|\mathbf{x}\| = 1} \langle \mathbf{s}, \mathbf{x} \rangle.$$

Suppose A is a linear operator from E_1 to E_2^*, and E_i has norm $\|\cdot\|_i$ for $i = 1, 2$. Then the norm of A is defined as

$$\|A\| := \max_{\mathbf{x} \in E_1, \boldsymbol{\alpha} \in E_2, \|\mathbf{x}\|_1 = \|\boldsymbol{\alpha}\|_2 = 1} \langle A\mathbf{x}, \boldsymbol{\alpha} \rangle. \tag{14}$$

If we define an adjoint operator $A^* : E_2 \mapsto E_1^*$ as

$$\langle A^* \boldsymbol{\alpha}, \mathbf{x} \rangle := \langle A\mathbf{x}, \boldsymbol{\alpha} \rangle, \quad \forall \mathbf{x} \in E_1, \boldsymbol{\alpha} \in E_2.$$

Then it can be shown that

$$\begin{aligned} \|A^*\| &= \max_{\mathbf{x} \in E_1, \boldsymbol{\alpha} \in E_2, \|\mathbf{x}\|_1 = \|\boldsymbol{\alpha}\|_2 = 1} \langle A^* \boldsymbol{\alpha}, \mathbf{x} \rangle \\ &= \max_{\mathbf{x} \in E_1, \boldsymbol{\alpha} \in E_2, \|\mathbf{x}\|_1 = \|\boldsymbol{\alpha}\|_2 = 1} \langle A\mathbf{x}, \boldsymbol{\alpha} \rangle = \|A\|. \end{aligned}$$

The definition of matrix norm in (14) implies that

$$\begin{aligned} \|A\mathbf{x}\|^* &\le \|A\| \, \|\mathbf{x}\| \quad \forall \, \mathbf{x} \in E_1, \\ \|A^* \boldsymbol{\alpha}\|^* &\le \|A^*\| \, \|\boldsymbol{\alpha}\| \quad \forall \, \boldsymbol{\alpha} \in E_2. \end{aligned}$$

To simplify notation we denote

$$\ell_f(\mathbf{x}; \mathbf{y}, \lambda_1) := f(\mathbf{y}) + \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle + \lambda_1 \Delta(\mathbf{x}, \mathbf{y}).$$

If f is λ_1-sc, then $\ell_f(\mathbf{x}; \mathbf{y}, \lambda_1) \le f(\mathbf{x})$ for all $\mathbf{y}$ and $\mathbf{x}$.

3. ∞-memory AGM-EF

The ∞-memory version of AGM-EF refers to the class of algorithms which use in each iteration all the past gradients $\nabla f(\mathbf{u}_1), \ldots, \nabla f(\mathbf{u}_k)$. We present the method in Algorithm 1.

[4] One can verify by simple algebra that $\mathbf{u}_{k+1}$ is a convex combination of $\mathbf{z}_k$ and $\mathbf{x}_k$.

Algorithm 1 ∞-memory AGM-EF (AGM-EF-∞).

1: Arbitrarily initialize $\mathbf{x}_0 \in \operatorname{dom}\Psi$. Set $\mathbf{z}_0 \leftarrow \mathbf{x}_0$.
2: Set $A_0 \leftarrow 0$.
3: $\psi_0(\mathbf{x}) \leftarrow \Delta(\mathbf{x}, \mathbf{x}_0)$.
4: **for** $k = 0, 1, \ldots$ **do**
5: Denote as a_{k+1} the positive root (in a) of $(a + A_k)(\lambda_1 a + \lambda A_k + 1) + a\lambda_2 A_k = L\sigma^{-1}a^2$.
6: $A_{k+1} \leftarrow A_k + a_{k+1}$, $\tau_1 \leftarrow 1 + \lambda A_k$, $\tau_2 \leftarrow \lambda_1 a_{k+1}$, $\tau_3 \leftarrow \lambda_2 a_{k+1}\frac{A_k}{A_{k+1}}$, $\tau \leftarrow \tau_1 + \tau_2 + \tau_3$.
7: $\mathbf{u}_{k+1} \leftarrow \frac{a_{k+1}\tau_1\mathbf{z}_k + (\tau A_k + \tau_3 a_{k+1})\mathbf{x}_k}{\tau A_{k+1} - \tau_2 a_{k+1}}$. [4]
8: $\psi_{k+1}(\mathbf{x}) \leftarrow \psi_k(\mathbf{x}) + a_{k+1}[\Psi(\mathbf{x}) + \ell_f(\mathbf{x}; \mathbf{u}_{k+1}, \lambda_1)]$.
9: Find $\mathbf{z}_{k+1} \leftarrow \operatorname{argmin}_{\mathbf{x}} \psi_{k+1}(\mathbf{x})$.
10: $\mathbf{x}_{k+1} \leftarrow (A_k\mathbf{x}_k + a_{k+1}\mathbf{z}_{k+1})/A_{k+1}$.
11: **end for**

The main idea of the algorithm is to approximate $J(\mathbf{x})$ by a sequence of functions ψ_k that are constructed in Step 8 of Algorithm 1, and then ensure the following relationship at all iterations ($k \geq 0$):

$$A_k J(\mathbf{x}_k) \leq \min_{\mathbf{x}} \psi_k(\mathbf{x}). \tag{15}$$

By construction, for all $k \geq 0$

$$\psi_k(\mathbf{x}) = \Delta(\mathbf{x}, \mathbf{x}_0) + \sum_{i=1}^{k} a_i[\Psi(\mathbf{x}) + \ell_f(\mathbf{x}; \mathbf{u}_i, \lambda_1)]. \tag{16}$$

Summation from 1 to 0 is assumed to be 0. Now it is not hard to see that relationship (15) implies rates of convergence:

Lemma 3. *If (15) holds for all $k \geq 1$, then for any $\mathbf{x} \in \operatorname{dom}\Psi$, we have*

$$J(\mathbf{x}_k) - J(\mathbf{x}) \leq A_k^{-1}\Delta(\mathbf{x}, \mathbf{x}_0). \tag{17}$$

Proof. By (16), we have for all $k \geq 1$

$$\begin{aligned}\psi_k(\mathbf{x}) &= \Delta(\mathbf{x}, \mathbf{x}_0) + \sum_{i=0}^{k} a_i[\Psi(\mathbf{x}) + \ell_f(\mathbf{x}; \mathbf{u}_i, \lambda_1)] \\ &\leq \Delta(\mathbf{x}, \mathbf{x}_0) + \sum_{i=0}^{k} a_i[\Psi(\mathbf{x}) + f(\mathbf{x})] \\ &= \Delta(\mathbf{x}, \mathbf{x}_0) + A_k J(\mathbf{x}).\end{aligned}$$

Combining with (15), we get (17). ∎

Therefore, the rate of convergence totally depends on how fast A_k grows. We will show that Algorithm 1 yields $A_k \sim k^2$ if $\lambda = 0$, or $A_k \sim e^k$ if $\lambda > 0$. All updates are also kept efficient. We next prove (15) and lower bound the growth rate of A_k.

Lemma 4 (Eq (15)). *The sequence $\{\mathbf{x}_k\}$ generated by Algorithm 1 satisfy for all $k \geq 0$*

$$A_k J(\mathbf{x}_k) \leq \min_{\mathbf{x}} \psi_k(\mathbf{x}).$$

Proof. We prove by induction. First check both sides are 0 for $k = 0$. Now suppose (15) holds for some step $k \geq 0$. By (16) and Property 2, ψ_k must be $(\lambda A_k + 1)$-sc wrt d. So by Property 4 and the fact that $\mathbf{z}_k$ minimizes ψ_k, we have

$$\begin{aligned}\psi_k(\mathbf{z}_{k+1}) &\geq \psi_k(\mathbf{z}_k) + (\lambda A_k + 1)\Delta(\mathbf{z}_{k+1}, \mathbf{z}_k) \\ &\geq A_k J(\mathbf{x}_k) + (\lambda A_k + 1)\Delta(\mathbf{z}_{k+1}, \mathbf{z}_k),\end{aligned} \tag{18}$$

where the second inequality is by induction assumption. So

$$\begin{aligned}&\min_{\mathbf{x}} \psi_{k+1}(\mathbf{x}) = \psi_{k+1}(\mathbf{z}_{k+1}) \\ &= \psi_k(\mathbf{z}_{k+1}) + a_{k+1}\ell_f(\mathbf{z}_{k+1}; \mathbf{u}_{k+1}, \lambda_1) + a_{k+1}\Psi(\mathbf{z}_{k+1}) \\ &\overset{(a)}{\geq} A_k f(\mathbf{x}_k) + A_k\Psi(\mathbf{x}_k) + (1 + \lambda A_k)\Delta(\mathbf{z}_{k+1}, \mathbf{z}_k) \\ &\quad + a_{k+1}\ell_f(\mathbf{z}_{k+1}; \mathbf{u}_{k+1}, \lambda_1) + a_{k+1}\Psi(\mathbf{z}_{k+1}) \\ &\overset{(b)}{\geq} A_k\left[f(\mathbf{u}_{k+1}) + \langle\nabla f(\mathbf{u}_{k+1}), \mathbf{x}_k - \mathbf{u}_{k+1}\rangle\right] \\ &\quad + (1 + \lambda A_k)\Delta(\mathbf{z}_{k+1}, \mathbf{z}_k) + A_k\Psi(\mathbf{x}_k) + a_{k+1}\Psi(\mathbf{z}_{k+1}) \\ &\quad + a_{k+1}[f(\mathbf{u}_{k+1}) + \langle\nabla f(\mathbf{u}_{k+1}), \mathbf{z}_{k+1} - \mathbf{u}_{k+1}\rangle \\ &\qquad + \lambda_1\Delta(\mathbf{z}_{k+1}, \mathbf{u}_{k+1})] \\ &= A_{k+1}f(\mathbf{u}_{k+1}) + A_k\Psi(\mathbf{x}_k) + a_{k+1}\Psi(\mathbf{z}_{k+1}) \\ &\quad + \langle\nabla f(\mathbf{u}_{k+1}), A_k\mathbf{x}_k - A_{k+1}\mathbf{u}_{k+1} + a_{k+1}\mathbf{z}_{k+1}\rangle \\ &\quad + \tau_1\Delta(\mathbf{z}_{k+1}, \mathbf{z}_k) + \tau_2\Delta(\mathbf{z}_{k+1}, \mathbf{u}_{k+1}) \\ &\overset{(c)}{\geq} A_{k+1}f(\mathbf{u}_{k+1}) \\ &\quad + A_{k+1}\Psi(\mathbf{x}_{k+1}) + \frac{\sigma}{2}\tau_3\left\|\mathbf{z}_{k+1} - \mathbf{x}_k\right\|^2 \\ &\quad + \langle\nabla f(\mathbf{u}_{k+1}), A_k\mathbf{x}_k - A_{k+1}\mathbf{u}_{k+1} + a_{k+1}\mathbf{z}_{k+1}\rangle \\ &\quad + \frac{\sigma}{2}\tau_1\left\|\mathbf{z}_{k+1} - \mathbf{z}_k\right\|^2 + \frac{\sigma}{2}\tau_2\left\|\mathbf{z}_{k+1} - \mathbf{u}_{k+1}\right\|^2 \\ &\overset{(d)}{\geq} A_{k+1}\Psi(\mathbf{x}_{k+1}) + A_{k+1}\Big[f(\mathbf{u}_{k+1}) \\ &\quad + \frac{a_{k+1}}{A_{k+1}}\left\langle\nabla f(\mathbf{u}_{k+1}), \mathbf{z}_{k+1} - \frac{A_{k+1}\mathbf{u}_{k+1} - A_k\mathbf{x}_k}{a_{k+1}}\right\rangle \\ &\quad + \frac{\sigma}{2}\frac{\tau_1 + \tau_2 + \tau_3}{A_{k+1}}\left\|\mathbf{z}_{k+1} - \frac{\tau_1\mathbf{z}_k + \tau_2\mathbf{u}_{k+1} + \tau_3\mathbf{x}_k}{\tau_1 + \tau_2 + \tau_3}\right\|^2\Big] \\ &\overset{(e)}{=} A_{k+1}\Psi(\mathbf{x}_{k+1}) + A_{k+1}\Big[f(\mathbf{u}_{k+1}) \\ &\quad + \frac{a_{k+1}}{A_{k+1}}\left\langle\nabla f(\mathbf{u}_{k+1}), \mathbf{z}_{k+1} - \frac{A_{k+1}\mathbf{u}_{k+1} - A_k\mathbf{x}_k}{a_{k+1}}\right\rangle \\ &\quad + \frac{L}{2}\left(\frac{a_{k+1}}{A_{k+1}}\right)^2\left\|\mathbf{z}_{k+1} - \frac{\tau_1\mathbf{z}_k + \tau_2\mathbf{u}_{k+1} + \tau_3\mathbf{x}_k}{\tau_1 + \tau_2 + \tau_3}\right\|^2\Big] \\ &\overset{(f)}{=} A_{k+1}\Psi(\mathbf{x}_{k+1}) + A_{k+1}\Big[f(\mathbf{u}_{k+1})\end{aligned}$$

$$+ \langle \nabla f(\mathbf{u}_{k+1}), \mathbf{x}_{k+1} - \mathbf{u}_{k+1} \rangle + \frac{L}{2} \|\mathbf{x}_{k+1} - \mathbf{u}_{k+1}\|^2 \Big]$$
$$\overset{(g)}{\geq} A_{k+1}\Psi(\mathbf{x}_{k+1}) + A_{k+1} f(\mathbf{x}_{k+1}) = A_{k+1} J(\mathbf{x}_{k+1}).$$

Here, step (a) is by (18). (b) is by the convexity of f (at $\mathbf{u}_{k+1}$). (c) is by the λ_2-sc of Ψ and Property 1. (d) is by the convexity and linearity of $\|\cdot\|$. (e) is by the rule of choosing a_{k+1} in Step 5 of Algorithm 1. (f) is by the definition of $\mathbf{x}_{k+1}$ and $\mathbf{u}_{k+1}$. (g) is by *L-l.c.g* of f. ■

Next, we can lower bound the growth rate of A_k.

Lemma 5. *Let $k \geq 1$. Then*

$$A_k \geq \max\left\{ \frac{\sigma}{4L}(k+1)^2, \frac{\sigma}{L - \sigma\lambda_1}\left(1 + \sqrt{\frac{\sigma\lambda}{4L}}\right)^{2k-2} \right\}.$$

Proof. Since $A_0 = 0$, so by solving Step 5 in Algorithm 1, we get $A_1 = \frac{\sigma}{L-\sigma\lambda_1}$. Hence the lemma clearly holds for $k = 1$. For all $k \geq 1$, denote

$$\begin{aligned} M &= (a_{k+1} + A_k)(\lambda_1 a_{k+1} + \lambda A_k + 1) + a_{k+1}\lambda_2 A_k \\ &= A_{k+1} + \lambda A_k A_{k+1} + \lambda_1 a_{k+1} A_{k+1} + \lambda_2 a_{k+1} A_k. \end{aligned}$$

By the choice of a_{k+1} in Step 5 of Algorithm 1, we get

$$\begin{aligned} A_{k+1} \leq M &= \frac{L}{\sigma}(A_{k+1} - A_k)^2 \\ &= \frac{L}{\sigma}\left(\sqrt{A_{k+1}} + \sqrt{A_k}\right)^2 \left(\sqrt{A_{k+1}} - \sqrt{A_k}\right)^2 \\ &\leq \frac{4L}{\sigma} A_{k+1}\left(\sqrt{A_{k+1}} - \sqrt{A_k}\right)^2. \end{aligned} \tag{19}$$

So when $\lambda = 0$ we have

$$A_k \geq \left(\frac{k-1}{2}\sqrt{\frac{\sigma}{L}} + \sqrt{A_1}\right)^2 = \frac{\sigma}{4L}(k+1)^2.$$

When $\lambda > 0$, we have

$$\lambda A_k A_{k+1} \leq M \leq \frac{4L}{\sigma} A_{k+1}\left(\sqrt{A_{k+1}} - \sqrt{A_k}\right)^2$$

where the last step is by (19). So

$$\sqrt{A_{k+1}} \geq \left(1 + \sqrt{\frac{\lambda\sigma}{4L}}\right)\sqrt{A_k},$$

which directly implies the second term in max. ■

Combining Lemma 3, 4 and 5, we derive

Theorem 6. *For all $k \geq 1$ and $\mathbf{x} \in \operatorname{dom}\Psi$,*

$$J(\mathbf{x}_k) - J(\mathbf{x}) \leq \Delta(\mathbf{x}, \mathbf{x}_0) \min\left\{ \frac{4L}{\sigma(k+1)^2}, \frac{L - \sigma\lambda_1}{\sigma}\left(1 + \sqrt{\frac{\sigma\lambda}{4L}}\right)^{-2k+2} \right\}.$$

Therefore, as long as one of λ_1 and λ_2 is strictly positive such that $\lambda = \lambda_1 + \lambda_2 > 0$, $J(\mathbf{x}_k)$ converges linearly. When $\lambda_1 = 0$ and $\lambda_2 > 0$, ψ_k contains only one Bregman divergence making it easier to optimize.

Remark 1. If (18) is replaced by

$$\begin{aligned} \psi_k(\mathbf{z}_{k+1}) &\geq \psi_k(\mathbf{z}_k) + (\lambda A_k + 1)\Delta(\mathbf{z}_{k+1}, \mathbf{z}_k) \\ &\geq A_k J(\mathbf{x}_k) + (\lambda A_k + 1)\frac{\sigma}{2}\|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2, \end{aligned}$$

then it is not hard to see that the proof of Lemma 4 still goes through. So Ψ does not need to be λ_2-sc wrt d, and it suffices to be $\lambda_2\sigma$ strongly convex wrt $\|\cdot\|$. In practice, checking and satisfying the latter condition can be much easier. Similar remark can be made later for AGM-EF-1, and for the ease of exposition we will still assume Ψ is λ_2-sc wrt d.

3.1. Notes on the Computations

The whole algorithm relies on solving $\mathbf{z}_k$ efficiently, and it can be dealt with in two ways. First, by (16), minimizing $\psi_k(\mathbf{x})$ only requires solving the following form of problem:

$$\min_{\mathbf{x}} A_k\Psi(\mathbf{x}) + \Delta(\mathbf{x}, \mathbf{u}_0) + \lambda_1 \sum_{i=0}^{n} a_i \Delta(\mathbf{x}, \mathbf{u}_i) + \left\langle \sum_{i=0}^{n} a_i \nabla f(\mathbf{u}_i), \mathbf{x} \right\rangle$$

This is feasible by Assumption 2, and in practice the gradients of f and d can be aggregated on the fly.

The second method requires making one more assumption, in addition to the usual assumption $\operatorname{dom}\Psi \subseteq \operatorname{dom} d$.

Assumption 7. $\operatorname{dom} d \subseteq \operatorname{dom}\Psi$.

This assumption is often met when d is the entropy and $\operatorname{dom}\Psi$ is the simplex. It ensures that $\mathbf{z}_k := \min_{\mathbf{x}\in\operatorname{dom}\Psi} \psi_k(\mathbf{x})$ is also a solution of the unconstrained optimization $\min_{\mathbf{x}\in\operatorname{dom} d} \psi_k(\mathbf{x})$. Then when Ψ is affine on its domain, we can apply Property 6 and the subsequent discussion on inductively updating $\psi_k(\mathbf{x})$. This scheme is particularly useful in Algorithm 1 because the minimizer $\mathbf{z}_k$ is already available.

Even if Assumption 7 does not hold and $\mathbf{z}_k$ is not an unconstrained minimizer of $\psi_k(\mathbf{x})$, one can still spend extra computations to find the unconstrained minimizer and inductively update $\psi_k(\mathbf{x})$. This idea will be useful if the gradient aggregation in the first method is not viable.

3.2. Adaptively tuning the Lipschitz constant

The Algorithm 1 requires the explicit value of L. This is usually not available, or the global maximum cur-

Algorithm 2 AGM-EF-∞ with adaptive L.

Require: Down scaling factor γ_d and up scaling factor γ_u ($\gamma_d, \gamma_u > 1$). An optimistic estimate $\tilde{L} \leq L$.
1: Arbitrarily initialize $\mathbf{x}_0 \in \operatorname{dom}\Psi$. Set $\mathbf{z}_0 \leftarrow \mathbf{x}_0$.
2: Set $A_0 \leftarrow 0$.
3: $\psi_0(\mathbf{x}) \leftarrow \Delta(\mathbf{x}, \mathbf{x}_0)$.
4: $L_0 \leftarrow \tilde{L} * \gamma_d * \gamma_u$.
5: **for** $k = 0, 1, \ldots$ **do**
6: $\quad L_{k+1} \leftarrow L_k/(\gamma_d * \gamma_u)$.
7: $\quad$ **repeat**
8: $\quad\quad L_{k+1} \leftarrow L_{k+1} * \gamma_u$.
9: $\quad\quad$ Assign to a_{k+1} the positive root (in a) of $(a+A_k)(\lambda_1 a+\lambda A_k+1)+a\lambda_2 A_k = L_{k+1}\sigma^{-1}a^2$.
10: $\quad\quad$ Do step 6 to 10 of Algorithm 1.
11: $\quad$ **until** $A_{k+1}J(\mathbf{x}_{k+1}) \leq \psi_{k+1}(\mathbf{z}_{k+1})$.
12: **end for**

vature is much larger than the local directional curvature. As a result, the steps size $1/L$ becomes too conservative. From the proof of Lemma 4, it is clear that L is used only to ensure (15). So we can probe smaller values of L. The modified algorithm is given in Algorithm 2.

The inner "repeat" loop must terminate in a finite number of steps because L_k grows exponentially and once $L_k \geq L$ the "until" condition must be satisfied. And the number of steps in this inner loop is logarithmic in L, with the final $L_k < \gamma_u L$. Moreover, this L_k is decayed by a factor of γ_d before being used to initialize L_{k+1}. This is in sharp contrast to AGM-PR where the estimates of L must grow monotonically through iterations. Let us formally characterize how adaptively tuning L leads to faster convergence rates through faster growth rate of A_k.

Lemma 8. *For all $k \geq 1$,*

$$A_k \geq \max\left\{\frac{\sigma}{L_1 - \sigma\lambda_1}\prod_{i=2}^{k}\left(1+\sqrt{\frac{\sigma\lambda}{L_i}}\right)^2, \frac{\sigma}{4}\left(\sqrt{\frac{4}{L_1}}+\sum_{i=2}^{k}\sqrt{\frac{1}{L_i}}\right)^2\right\}.$$

Proof. Simply replace the L in (19) by L_{i+1}. ∎

In practice, we observed that the L_k is often only 10 per cent of the real L and therefore by Lemma 8 the convergence rate is 10 times faster than using L. Moreover, the L_k in successive iterations are quite close so the inner loop terminates in only 2-3 steps.

This adaptive scheme relies on the fact that the key relationship (15) is independent of L and involves function values only at two points (rather than globally). In contrast, the algorithm and analysis in [26] keep a global relationship which explicitly involves L, making it hard to accommodate adaptive L.

We also tried to adaptively tune λ, but not successful. This is turns out to be very hard because the proof uses λ as a a global property (recall the fact that ψ_k must be $(\lambda A_k + 1)$-sc wrt d), while L is used only at $\mathbf{u}_{k+1}$ and $\mathbf{x}_{k+1}$ in Step (g) of the proof of Lemma 4.

3.3. Bounding the Duality Gap

Algorithm 1 does not have a termination criterion, and a natural criterion will be based on the duality gap. Furthermore, in some applications like (1) the primal problem is nonsmooth and AGM-EF-∞ is applied only to its dual problem which is *l.c.g.* So it is necessary to convert the dual iterates at each step into the primal, and characterize the convergence rate in the primal. In this subsection, we extend the technique in [2, Section 2] to the case of composite objective. Except the strong convexity, our whole setting and procedure bear much resemblance to [9], [2, Theorem 2.2], [6, Theorem 3] and [17, Section 6]. We are unaware of any existing result which shows *linear convergence of the duality gap* as we will describe below.

Consider a minimax problem

$$\min_{\mathbf{x}}\max_{\boldsymbol{\alpha}\in Q_2}\phi(\mathbf{x},\boldsymbol{\alpha})+\Psi(\mathbf{x}).$$

Here $\Psi : \mathbb{R}^p \mapsto \overline{\mathbb{R}}$ is proper, lower semicontinuous and λ_2-sc wrt d ($\lambda_2 \geq 0$). Let Ψ satisfy Assumption 2. Q_2 is a compact convex set in the Euclidean space. $\phi : \mathbb{R}^p \times Q_2 \mapsto \overline{\mathbb{R}}$ is continuous on $\operatorname{dom}\Psi \times Q_2$. For all fixed $\boldsymbol{\alpha} \in Q_2$, $\phi(\cdot, \boldsymbol{\alpha})$ is λ_1-sc wrt d ($\lambda_1 \geq 0$) and is differentiable on a open set containing $\operatorname{dom}\Psi$. For all fixed $\mathbf{x} \in \operatorname{dom}\Psi$, $\phi(\mathbf{x}, \cdot)$ is strictly concave. Therefore, the $\operatorname{argmax}_{\boldsymbol{\alpha}\in Q_2}\phi(\mathbf{x},\boldsymbol{\alpha})$ is unique and we denote it as $\boldsymbol{\alpha}(\mathbf{x})$.

Let us define

$$f(\mathbf{x}) := \max_{\boldsymbol{\alpha}\in Q_2}\phi(\mathbf{x},\boldsymbol{\alpha}). \tag{20}$$

Then by Denskin's theorem [27, Theorem B.25], f must be convex and differentiable on $\operatorname{dom}\Psi$. We further assume that f is *L-l.c.g* on $\operatorname{dom}\Psi$. A key strong convexity property of f is:

Lemma 9. *Given all the above assumptions on ϕ, $f(\mathbf{x})$ must be λ_1-sc. However, the converse is not necessarily true,* i.e. *$f(\mathbf{x})$ being λ_1-sc does not entail that $\phi(\cdot, \boldsymbol{\alpha})$ is λ_1-sc for all fixed $\boldsymbol{\alpha} \in Q_2$.*

Proof. For any $\mathbf{x}_1, \mathbf{x}_2 \in \operatorname{dom}\Psi$, we have

$$\begin{aligned} f(\mathbf{x}_2) &= \max_{\boldsymbol{\alpha}\in Q_2} \phi(\mathbf{x}_2, \boldsymbol{\alpha}) \geq \phi(\mathbf{x}_2, \boldsymbol{\alpha}(\mathbf{x}_1)) \\ &\geq \phi(\mathbf{x}_1, \boldsymbol{\alpha}(\mathbf{x}_1)) + \langle \nabla_{\mathbf{x}} \phi(\mathbf{x}_1, \boldsymbol{\alpha}(\mathbf{x}_1)), \mathbf{x}_2 - \mathbf{x}_1 \rangle \\ &\quad + \lambda_1 \Delta(\mathbf{x}_2, \mathbf{x}_1) \\ &= f(\mathbf{x}_1) + \langle \nabla f(\mathbf{x}_1), \mathbf{x}_2 - \mathbf{x}_1 \rangle + \lambda_1 \Delta(\mathbf{x}_2, \mathbf{x}_1), \end{aligned}$$

where the last step is by Denskin's theorem. ∎

We also define a dual objective

$$\begin{aligned} J(\mathbf{x}) &:= \Psi(\mathbf{x}) + \max_{\boldsymbol{\alpha}\in Q_2} \phi(\mathbf{x}, \boldsymbol{\alpha}) \\ D(\boldsymbol{\alpha}) &:= \min_{\mathbf{x}} \{\phi(\mathbf{x}, \boldsymbol{\alpha}) + \Psi(\mathbf{x})\} \quad \text{for } \boldsymbol{\alpha} \in Q_2 \end{aligned} \tag{21}$$

where the argmin in (21) may be not unique and $D(\boldsymbol{\alpha})$ may be nonsmooth. Our assumptions above ensure that for any $\boldsymbol{\alpha} \in Q_2$ and any $\mathbf{x}$, the following is true:

$$D(\boldsymbol{\alpha}) \leq J(\mathbf{x}), \quad \text{and} \quad \max_{\boldsymbol{\alpha}\in Q_2} D(\boldsymbol{\alpha}) = \min_{\mathbf{x}} J(\mathbf{x}).$$

When applied to minimize $J(\mathbf{x})$, AGM-EF-∞ (with or without adaptive L) produces a sequence of $\{\mathbf{x}_k, \mathbf{u}_k, \mathbf{z}_k\}$. It is our goal to design a sequence of dual variables $\{\boldsymbol{\alpha}_k\}$ based on $\{\mathbf{x}_i, \mathbf{u}_i, \mathbf{z}_i : i \leq k\}$ such that the duality gap

$$\delta_k := J(\mathbf{x}_k) - D(\boldsymbol{\alpha}_k)$$

goes to 0 fast. Since

$$\delta_k \geq J(\mathbf{x}_k) - \max_{\boldsymbol{\alpha}\in Q_2} D(\boldsymbol{\alpha}) = J(\mathbf{x}_k) - \min_{\mathbf{x}} J(\mathbf{x}),$$

so once δ_k falls below a prescribed tolerance ϵ, $\mathbf{x}_k$ is guaranteed to be an ϵ accurate solution of J. Indeed we will show that the following construction of $\boldsymbol{\alpha}_k$ meets our need:

$$\boldsymbol{\alpha}_k = \frac{1}{A_k} \sum_{i=1}^{k} a_i \boldsymbol{\alpha}(\mathbf{u}_i). \tag{22}$$

where a_i and A_k are also from AGM-EF-∞. (22) can be equivalently reformulated into a recursion which allows efficient update of $\boldsymbol{\alpha}_k$:

$$\boldsymbol{\alpha}_1 = \boldsymbol{\alpha}(\mathbf{u}_1), \text{ and } \boldsymbol{\alpha}_{k+1} = \frac{A_k}{A_{k+1}} \boldsymbol{\alpha}_k + \frac{a_{k+1}}{A_{k+1}} \boldsymbol{\alpha}(\mathbf{u}_{k+1}).$$

Theorem 10. *Suppose a sequence $\{\mathbf{x}_k, \mathbf{u}_k, \mathbf{z}_k\}$ is produced when AGM-EF-∞ is applied to minimize $J(\mathbf{x})$ by treating f as λ_1-sc. Then the $\{\boldsymbol{\alpha}_k\}$ defined by (22) satisfies $\boldsymbol{\alpha}_k \in Q_2$ and*

$$\delta_k = J(\mathbf{x}_k) - D(\boldsymbol{\alpha}_k) \leq \frac{1}{A_k} \max_{\mathbf{x}\in\operatorname{dom}\Psi} \Delta(\mathbf{x}, \mathbf{u}_0). \tag{23}$$

Proof. Since $\mathbf{u}_i \in \operatorname{dom}\Psi$, so $\boldsymbol{\alpha}(\mathbf{u}_i) \in Q_2$. And $\boldsymbol{\alpha}_k$ is a convex combination of $\boldsymbol{\alpha}(\mathbf{u}_i)$ $(i \leq k)$, so $\boldsymbol{\alpha}_k \in Q_2$. Using the fact that $\phi(\mathbf{x}, \boldsymbol{\alpha})$ is λ_1-sc in $\boldsymbol{\alpha}$ for all fixed $\mathbf{x}$, we have

$$\begin{aligned} &\ell_f(\mathbf{x}; \mathbf{u}_i, \lambda_1) \\ &= f(\mathbf{u}_i) + \langle \nabla f(\mathbf{u}_i), \mathbf{x} - \mathbf{u}_i \rangle + \lambda_1 \Delta(\mathbf{x}, \mathbf{u}_i) \\ &= \phi(\mathbf{u}_i, \boldsymbol{\alpha}(\mathbf{u}_i)) + \langle \nabla_{\mathbf{x}} \phi(\mathbf{u}_i, \boldsymbol{\alpha}(\mathbf{u}_i)), \mathbf{x} - \mathbf{u}_i \rangle + \lambda_1 \Delta(\mathbf{x}, \mathbf{u}_i) \\ &\leq \phi(\mathbf{x}, \boldsymbol{\alpha}(\mathbf{u}_i)). \end{aligned} \tag{24}$$

Now by using relationship (15) and (16), we have

$$\begin{aligned} &A_k J(\mathbf{x}_k) \\ &\leq \min_{\mathbf{x}} \left\{ \Delta(\mathbf{x}, \mathbf{u}_0) + A_k \Psi(\mathbf{x}) + \sum_{i=1}^{k} a_i \ell_f(\mathbf{x}; \mathbf{u}_i, \lambda_1) \right\} \\ &\leq \min_{\mathbf{x}} \Delta(\mathbf{x}, \mathbf{u}_0) + A_k \Psi(\mathbf{x}) + \sum_{i=1}^{k} a_i \phi(\mathbf{x}, \boldsymbol{\alpha}(\mathbf{u}_i)) \\ &\leq \min_{\mathbf{x}} \Delta(\mathbf{x}, \mathbf{u}_0) + A_k \Psi(\boldsymbol{\alpha}) + A_k \phi\left(\mathbf{x}, \frac{1}{A_k} \sum_{i=1}^{k} a_i \boldsymbol{\alpha}(\mathbf{u}_i)\right) \\ &\leq \max_{\mathbf{x}\in\operatorname{dom}\Psi} \Delta(\mathbf{x}, \mathbf{u}_0) + A_k \min_{\mathbf{x}} \{\Psi(\mathbf{x}) + \phi(\mathbf{x}, \boldsymbol{\alpha}_k)\} \\ &= \max_{\mathbf{x}\in\operatorname{dom}\Psi} \Delta(\mathbf{x}, \mathbf{u}_0) + A_k D(\boldsymbol{\alpha}_k). \end{aligned}$$ ∎

So δ_k converges linearly as long as $\lambda_1 + \lambda_2 > 0$. If $\operatorname{dom}\Psi$ is unbounded and $\max_{\mathbf{x}\in\operatorname{dom}\Psi} \Delta(\mathbf{x}, \mathbf{u}_0) = \infty$, then the bound in (23) becomes vacuous.

We emphasize that in Theorem 10, AGM-EF-∞ is invoked by treating f as λ_1-sc, although the real strong convexity constant λ_1' of f may be greater than λ_1. In this case, the duality gap will decay at a slower rate than that for the gap of J (by using λ_1' in AGM-EF-∞). However the strong convexity of Ψ is still fully utilized in the duality gap, and in many machine learning problems the strong convexity does come from Ψ rather than f (*i.e.* $\lambda_1 = \lambda_1' = 0$).

4. 1-memory AGM-EF

Note that AGM-EF-∞ keeps a nonparametric form (16) of the model $\psi_k(\mathbf{x})$ whose complexity grows with iteration. In 1-memory AGM-EF, the model is compressed to a simple parametric form in each iteration. Auslender and Teboulle [28] gave a Bregman version for unconstrained optimization. [18] provided an algorithm for constrained problems with Euclidean distance as the prox-function. However, only [26] and [9] accommodate both Bregman divergence and constraints. But their algorithms do not extend to strongly convex objectives and restrict the estimate of L to be nondecreasing through iterations. Therefore, we propose in this section a 1-memory AGM-EF

Algorithm 3 1-memory AGM-EF (AGM-EF-1).

Arbitrarily pick $\mathbf{u}_0 \in \operatorname{dom}\Psi$.
Initialize $c_0 \leftarrow \frac{L}{\sigma} + \lambda_2$.
$q_0(\mathbf{x}) \leftarrow \frac{L}{\sigma}\Delta(\mathbf{x}, \mathbf{u}_0) + \Psi(\mathbf{x}) + \ell_f(\mathbf{x}; \mathbf{u}_0, 0)$.
$\mathbf{x}_0 = \mathbf{z}_0 \leftarrow \operatorname{argmin}_{\mathbf{x}} q_0(\mathbf{x})$.
for $k = 0, 1, \ldots$ **do**
  Assign to a_{k+1} the positive root (in a) of $\sigma(1-a)(c_k + \lambda_2 a) + \sigma\lambda_1 a = La^2$.
  $c_{k+1} \leftarrow (1 - a_{k+1})c_k + (\lambda_1 + \lambda_2)a_{k+1}$.
  $\tau_1 \leftarrow (1 - a_{k+1})c_k$, $\tau_2 \leftarrow \lambda_1 a_{k+1}$,
  $\tau_3 \leftarrow \lambda_2 a_{k+1}(1 - a_{k+1})$, $\tau \leftarrow \tau_1 + \tau_2 + \tau_3$.
  $\mathbf{u}_{k+1} \leftarrow \frac{(\tau - (\tau_1+\tau_2)a_{k+1})\mathbf{x}_k + \tau_1 a_{k+1}\mathbf{z}_k}{\tau - \tau_2 a_{k+1}}$. [5]
  $\psi_{k+1}(\mathbf{x}) \leftarrow (1 - a_{k+1})q_k(\mathbf{x}) + a_{k+1}[\ell_f(\mathbf{x}; \mathbf{u}_{k+1}, \lambda_1) + \Psi(\mathbf{x})]$.
  $\mathbf{z}_{k+1} \leftarrow \operatorname{argmin}_{\mathbf{x}} \psi_{k+1}(\mathbf{x})$.
  $\mathbf{x}_{k+1} \leftarrow (1 - a_{k+1})\mathbf{x}_k + a_{k+1}\mathbf{z}_{k+1}$.
  $q_{k+1}(\mathbf{x}) \leftarrow c_{k+1}\Delta(\mathbf{x}, \mathbf{z}_{k+1}) + \psi_{k+1}(\mathbf{z}_{k+1})$.
end for

which uses Bregman prox-function, and allows constraints and non-monotonic adaptive tuning of L.

Arbitrarily pick $\mathbf{u}_0 \in \operatorname{dom}\Psi$ and initialize by

$$
\begin{aligned}
q_0(\mathbf{x}) &:= \frac{L}{\sigma}\Delta(\mathbf{x}, \mathbf{u}_0) + f(\mathbf{x}_0) + \langle \nabla f(\mathbf{u}_0), \mathbf{x} - \mathbf{u}_0 \rangle + \Psi(\mathbf{x}) \\
\mathbf{x}_0 &= \mathbf{z}_0 = \operatorname*{argmin}_{\mathbf{x}} q_0(\mathbf{x}) \\
c_0 &= \frac{L}{\sigma} + \lambda_2.
\end{aligned}
$$

Then for all $k \geq 0$, define:

$$
\begin{aligned}
\psi_{k+1}(\mathbf{x}) &= (1 - a_{k+1})q_k(\mathbf{x}) + a_{k+1}[\ell_f(\mathbf{x}; \mathbf{u}_{k+1}, \lambda_1) + \Psi(\mathbf{x})] \\
\mathbf{z}_{k+1} &= \operatorname*{argmin}_{\mathbf{x}} \psi_{k+1}(\mathbf{x}) \\
c_{k+1} &= (1 - a_{k+1})c_k + \lambda a_{k+1} \\
q_{k+1}(\mathbf{x}) &= c_{k+1}\Delta(\mathbf{x}, \mathbf{z}_{k+1}) + \psi_{k+1}(\mathbf{z}_{k+1}).
\end{aligned}
$$

By construction for all $k \geq 0$, q_k is c_k-sc and ψ_{k+1} is strongly convex with constant $(1 - a_{k+1})c_k + \lambda a_{k+1}$, *i.e.* c_{k+1}-sc. Clearly, for all $k \geq 1$

$$
\min_{\mathbf{x}} \psi_k(\mathbf{x}) = \psi_k(\mathbf{z}_k) = q_k(\mathbf{z}_k) = \min_{\mathbf{x}} q_k(\mathbf{x}). \tag{25}
$$

But except at $\mathbf{x} = \mathbf{z}_k$, $q_k(\mathbf{x}) \neq \psi_k(\mathbf{x})$ in general. The only case where $q_k(\mathbf{x}) \equiv \psi_k(\mathbf{x})$ is when $\Psi(\mathbf{x})$ is an affine function on $\operatorname{dom} d$ and $\operatorname{dom} d \subseteq \operatorname{dom}\Psi$. Then an inductive application of Property 6 reveals $q_k(\mathbf{x}) \equiv \psi_k(\mathbf{x})$. Lemma 5.2 of [23] is exactly this case with $\Psi(\mathbf{x}) \equiv 0$. However, when $\operatorname{dom} d \nsubseteq \operatorname{dom}\Psi$ then $\mathbf{z}_{k+1}$ actually solves a constrained optimization, and then (11) must be changed to $\geq$ which breaks Property 13.

The proof of rate of convergence for Algorithm 3 relies on the following two relations: for all $k \geq 0$ and $\mathbf{x} \in \operatorname{dom}\Psi$,

$$
\begin{aligned}
q_{k+1}(\mathbf{x}) - J(\mathbf{x}) &\leq (1 - a_{k+1})(q_k(\mathbf{x}) - J(\mathbf{x})) && (26) \\
J(\mathbf{x}_k) &\leq q_k(\mathbf{z}_k). && (27)
\end{aligned}
$$

From these three inequalities, we get for all $\mathbf{x} \in \operatorname{dom}\Psi$,

$$
\begin{aligned}
J(\mathbf{x}_k) &\overset{(27)}{\leq} q_k(\mathbf{z}_k) \overset{(25)}{\leq} q_k(\mathbf{x}) \\
&\overset{(26)}{\leq} J(\mathbf{x}) + (q_0(\mathbf{x}) - J(\mathbf{x})) \cdot \prod_{i=1}^{k}(1 - a_i). && (28)
\end{aligned}
$$

So the gap $J(\mathbf{x}_k) - J(\mathbf{x})$ decays at the same rate as $\prod_{i=1}^{k}(1 - a_i)$.[6] Compared with the ∞-memory AGM-EF, the additional inequality (26) is now needed because the models q_k here are approximations of the ψ_k in (16). Next, we prove the three relations one by one.

Lemma 11 (Eq (26))**.** *For all $k \geq 0$ and $\mathbf{x}$, we have*

$$
q_{k+1}(\mathbf{x}) - J(\mathbf{x}) \leq (1 - a_{k+1})(q_k(\mathbf{x}) - J(\mathbf{x})).
$$

Proof. Since $\mathbf{z}_{k+1}$ minimizes $\psi_{k+1}(\mathbf{x})$ and $\psi_{k+1}(\mathbf{x})$ is c_{k+1}-sc, so by Property 5 we have

$$
\psi_{k+1}(\mathbf{x}) \geq \psi_{k+1}(\mathbf{z}_{k+1}) + c_{k+1}\Delta(\mathbf{x}, \mathbf{z}_{k+1}). \tag{29}
$$

So for all $\mathbf{x} \in Q$,

$$
\begin{aligned}
&(1 - a_{k+1})q_k(\mathbf{x}) + a_{k+1}J(\mathbf{x}) \\
&\geq (1 - a_{k+1})q_k(\mathbf{x}) + a_{k+1}[\ell_f(\mathbf{x}; \mathbf{u}_{k+1}, \lambda_1) + \Psi(\mathbf{x})] && (30) \\
&= \psi_{k+1}(\mathbf{x}) \\
&\geq \psi_{k+1}(\mathbf{z}_{k+1}) + c_{k+1}\Delta(\mathbf{x}, \mathbf{z}_{k+1}) \quad \text{(by (29))} \\
&= q_{k+1}(\mathbf{x}). && (31)
\end{aligned}
$$

■

Lemma 12 (Eq (27))**.** *For all $k \geq 0$, $J(\mathbf{x}_k) \leq q_k(\mathbf{z}_k)$.*

Proof. We prove by induction. First, when $k = 0$ $q_0(\mathbf{z}_0) = J(\mathbf{x}_0)$. Now suppose (27) holds for certain $k \geq 0$. Then

$$
\begin{aligned}
&q_{k+1}(\mathbf{z}_{k+1}) = \psi_{k+1}(\mathbf{z}_{k+1}) \\
&= (1 - a_{k+1})q_k(\mathbf{z}_{k+1}) + a_{k+1}[\ell_f(\mathbf{z}_{k+1}; \mathbf{u}_{k+1}, \lambda_1) + \Psi(\mathbf{z}_{k+1})] \\
&\overset{(a)}{\geq} (1 - a_{k+1})[q_k(\mathbf{z}_k) + c_k\Delta(\mathbf{z}_{k+1}, \mathbf{z}_k)]
\end{aligned}
$$

[5] $\mathbf{u}_{k+1}$ is clearly a convex combination of $\mathbf{z}_k$ and $\mathbf{x}_k$.

[6] The last inequality of (28) does not require $q_0(\mathbf{x}) \geq J(\mathbf{x})$. But $q_0(\mathbf{x}) \geq J(\mathbf{x})$ can be easily proved by Lemma 15.

$$
\begin{aligned}
&\quad + a_{k+1}[\ell_f(\mathbf{z}_{k+1}; \mathbf{u}_{k+1}, \lambda_1) + \Psi(\mathbf{z}_{k+1})] \\
&\overset{(b)}{\geq} (1 - a_{k+1})[f(\mathbf{x}_k) + \Psi(\mathbf{x}_k) + c_k \Delta(\mathbf{z}_{k+1}, \mathbf{z}_k)] \\
&\quad + a_{k+1}\ell_f(\mathbf{z}_{k+1}; \mathbf{u}_{k+1}, \lambda_1) + a_{k+1}\Psi(\mathbf{z}_{k+1}) \\
&\overset{(c)}{\geq} (1 - a_{k+1})[f(\mathbf{u}_{k+1}) + \langle \nabla f(\mathbf{u}_{k+1}), \mathbf{x}_k - \mathbf{u}_{k+1} \rangle \\
&\quad + \frac{c_k \sigma}{2} \|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2] + a_{k+1}[f(\mathbf{u}_{k+1}) \\
&\quad + \langle \nabla f(\mathbf{u}_{k+1}), \mathbf{z}_{k+1} - \mathbf{u}_{k+1} \rangle + \frac{\lambda_1 \sigma}{2} \|\mathbf{z}_{k+1} - \mathbf{u}_{k+1}\|^2] \\
&\quad + (1 - a_{k+1})\Psi(\mathbf{x}_k) + a_{k+1}\Psi(\mathbf{z}_{k+1}) \\
&\overset{(d)}{\geq} \Psi(\mathbf{x}_{k+1}) + f(\mathbf{u}_{k+1}) \\
&\quad + \langle \nabla f(\mathbf{u}_{k+1}), (1 - a_{k+1})\mathbf{x}_k + a_{k+1}\mathbf{z}_{k+1} - \mathbf{u}_{k+1} \rangle \\
&\quad + \frac{\sigma}{2} c_k (1 - a_{k+1}) \|\mathbf{z}_{k+1} - \mathbf{z}_k\|^2 \\
&\quad + \frac{\sigma}{2} \lambda_1 a_{k+1} \|\mathbf{z}_{k+1} - \mathbf{u}_{k+1}\|^2 \\
&\quad + \frac{\sigma}{2} \lambda_2 a_{k+1}(1 - a_{k+1}) \|\mathbf{z}_{k+1} - \mathbf{x}_k\|^2 \\
&\overset{(e)}{\geq} \Psi(\mathbf{x}_{k+1}) + f(\mathbf{u}_{k+1}) \\
&\quad + \langle \nabla f(\mathbf{u}_{k+1}), (1 - a_{k+1})\mathbf{x}_k + a_{k+1}\mathbf{z}_{k+1} - \mathbf{u}_{k+1} \rangle \\
&\quad + \frac{\sigma}{2}(\tau_1 + \tau_2 + \tau_3) \left\| \mathbf{z}_{k+1} - \frac{\tau_1 \mathbf{z}_k + \tau_2 \mathbf{u}_{k+1} + \tau_3 \mathbf{x}_k}{\tau_1 + \tau_2 + \tau_3} \right\|^2 \\
&\overset{(f)}{=} \Psi(\mathbf{x}_{k+1}) + f(\mathbf{u}_{k+1}) \\
&\quad + \langle \nabla f(\mathbf{u}_{k+1}), \mathbf{x}_{k+1} - \mathbf{u}_{k+1} \rangle + \frac{L}{2} \|\mathbf{x}_{k+1} - \mathbf{u}_{k+1}\|^2 \\
&\overset{(g)}{\geq} \Psi(\mathbf{x}_{k+1}) + f(\mathbf{x}_{k+1}) = J(\mathbf{x}_{k+1}),
\end{aligned}
$$

where (a) is because $\mathbf{z}_k$ minimizes q_k and q_k is c_k-sc. (b) is by the induction assumption. (c) is by the convexity of f and σ-sc of d. (d) is by the λ_2-sc of Ψ and Property 1. (e) is by the convexity of norm. (f) is by the definition of $\mathbf{x}_{k+1}$ and $\mathbf{u}_{k+1}$, and the choice of a_{k+1}. (g) is by the L-*l.c.g* of f. ■

Noting that $c_0 \geq \lambda$ by definition, we can bound $\prod_{i=1}^k (1 - a_i)$ by invoking Lemma 2.2.4 of [19] with the strong convexity constant being λ and the Lipschitz constant of the gradient being

$$L' := \frac{L}{\sigma} + \lambda_2.$$

It is easy to verify that the condition number L'/λ is monotonically decreasing in λ_2.

Lemma 13. *For all $k \geq 1$, we have*

$$\prod_{i=1}^k (1 - a_i) \leq \min \left\{ \left(1 - \sqrt{\frac{\lambda}{L'}}\right)^k, \frac{4L'}{(2\sqrt{L'} + k\sqrt{c_0})^2} \right\}.$$

Finally we bound $q_0(\mathbf{x}) - J(\mathbf{x})$ by

$$
\begin{aligned}
&q_0(\mathbf{x}) - J(\mathbf{x}) \\
&= \frac{L}{\sigma}\Delta(\mathbf{x}, \mathbf{u}_0) + \langle \nabla f(\mathbf{u}_0), \mathbf{x} - \mathbf{u}_0 \rangle + f(\mathbf{u}_0) - f(\mathbf{x}) \\
&\leq \left(\frac{L}{\sigma} - \lambda_1\right) \Delta(\mathbf{x}, \mathbf{u}_0). \quad \text{(by } \lambda_1\text{-sc of } f\text{)} \qquad (32)
\end{aligned}
$$

By (28) and the definition $c_0 = L'$, we get

Theorem 14. *For all $k \geq 1$ and $\mathbf{x} \in \operatorname{dom} \Psi$,*

$$
\begin{aligned}
&J(\mathbf{x}_k) - J(\mathbf{x}) \\
&\leq (q_0(\mathbf{x}) - J(\mathbf{x})) \min \left\{ \left(1 - \sqrt{\frac{\lambda}{L'}}\right)^k, \frac{4}{(2+k)^2} \right\} \\
&\leq \left(\frac{L}{\sigma} - \lambda_1\right) \Delta(\mathbf{x}, \mathbf{u}_0) \min \left\{ \left(1 - \sqrt{\frac{\lambda}{L'}}\right)^k, \frac{4}{(2+k)^2} \right\}.
\end{aligned}
$$

This rate is completely independent of Ψ (except λ_2). Although not needed by the proof, we can further show that $q_k(\mathbf{x}) \geq J(\mathbf{x})$ for all $k \geq 0$ and $\mathbf{x} \in \operatorname{dom} \Psi$.

Lemma 15. *$q_k(\mathbf{x}) \geq J(\mathbf{x})$ for all $k \geq 0$ and $\mathbf{x} \in \operatorname{dom} \Psi$.*

Proof. When $k = 0$,

$$
\begin{aligned}
q_0(\mathbf{x}) &= \frac{L}{\sigma}\Delta(\mathbf{x}, \mathbf{u}_0) + f(\mathbf{u}_0) + \langle \nabla f(\mathbf{u}_0), \mathbf{x} - \mathbf{u}_0 \rangle + \Psi(\mathbf{x}) \\
&\geq \frac{L}{2} \|\mathbf{x} - \mathbf{u}_0\|^2 + f(\mathbf{u}_0) + \langle \nabla f(\mathbf{u}_0), \mathbf{x} - \mathbf{u}_0 \rangle + \Psi(\mathbf{x}) \\
&\geq f(\mathbf{x}) + \Psi(\mathbf{x}) = J(\mathbf{x}).
\end{aligned}
$$

Suppose $k \geq 1$. By (25), $q_k(\mathbf{x}) \geq q_k(\mathbf{z}_k)$. By Lemma 12, $q_k(\mathbf{z}_k) \geq J(\mathbf{x}_k)$. So

$$q_k(\mathbf{x}) \geq q_k(\mathbf{z}_k) \geq J(\mathbf{x}_k) \geq J(\mathbf{x}).$$ ■

4.1. Adaptive L

It is straightforward to incorporate backtracking of L into the algorithm. We present this variant in Algorithm 4. Suppose at each iteration the inner loop terminates with L_k and define $L'_k = L_k/\sigma + \lambda_2$. Noting $c_0 = L'_0$ and slightly changing the proof, Lemma 13 can be extended as follows:

Lemma 16. *For all $k \geq 1$, we have*

$$\prod_{i=1}^k (1 - a_i) \leq \min \left\{ \prod_{i=1}^k \left(1 - \sqrt{\frac{\lambda}{L'_i}}\right), \frac{4}{L'_0} \left(\frac{2}{\sqrt{L'_0}} + \sum_{i=1}^k \frac{1}{\sqrt{L'_i}} \right)^{-2} \right\}.$$

Obviously, when $L'_i = L'$ we recover Lemma 13.

Algorithm 4 AGM-EF-1 with adaptive L.

Require: Down scaling factor γ_d and up scaling factor γ_u ($\gamma_d, \gamma_u > 1$). An optimistic estimate $\tilde{L} \le L$.

Arbitrarily pick $\mathbf{u}_0 \in \operatorname{dom}\Psi$. $L_0 \leftarrow \tilde{L}/\gamma_u$.
repeat
  $L_0 \leftarrow L_0 * \gamma_u$.
  Initialize $c_0 \leftarrow \frac{L_0}{\sigma} + \lambda_2$.
  $q_0(\mathbf{x}) \leftarrow \frac{L_0}{\sigma}\Delta(\mathbf{x}, \mathbf{u}_0) + \Psi(\mathbf{x}) + \ell_f(\mathbf{x}; \mathbf{u}_0, 0)$.
  $\mathbf{x}_0 = \mathbf{z}_0 \leftarrow \operatorname{argmin}_{\mathbf{x}} q_0(\mathbf{x})$.
until $J(\mathbf{x}_0) \le \min_{\mathbf{x}} q_0(\mathbf{x}_0)$
for $k = 0, 1, \ldots$ **do**
  $L_{k+1} \leftarrow L_k/(\gamma_d * \gamma_u)$.
  repeat
    $L_{k+1} \leftarrow L_{k+1} * \gamma_u$.
    Assign to a_{k+1} the positive root (in a) of $\sigma(1-a)(c_k + \lambda_2 a) + \sigma\lambda_1 a = L_{k+1}a^2$.
    Do step 7 to 12 of Algorithm 3.
  until $J(\mathbf{x}_{k+1}) \le q_{k+1}(\mathbf{z}_{k+1})$
end for

Furthermore, (32) needs to be changed into

$$q_0(\mathbf{x}) - J(\mathbf{x}) \le \left(\frac{L_0}{\sigma} - \lambda_1\right)\Delta(\mathbf{x}, \mathbf{u}_0).$$

So we conclude for all $k \ge 1$ and $\mathbf{x} \in \operatorname{dom}\Psi$,

$$J(\mathbf{x}_k) - J(\mathbf{x}) \le (L_0' - \lambda)\,\Delta(\mathbf{x}, \mathbf{u}_0)$$
$$\cdot \min\left\{\prod_{i=1}^{k}\left(1 - \sqrt{\frac{\lambda}{L_i'}}\right), \frac{4}{L_0'}\left(\frac{2}{\sqrt{L_0'}} + \sum_{i=1}^{k}\frac{1}{\sqrt{L_i'}}\right)^{-2}\right\}.$$

This bound does not involve the true L, and does not depend on Ψ or the function value of f (which could be used to hide L).

4.2. Bounding the duality gap

It is also not hard to extend AGM-EF-1 to the same primal-dual settings as in Section 3.3.

Using (30) and (31), we derive for all $\mathbf{x} \in \operatorname{dom}\Psi$:

$$q_{k+1}(\mathbf{x}) \le (1-a_{k+1})q_k(\mathbf{x}) + a_{k+1}[\ell_f(\mathbf{x}; \mathbf{u}_{k+1}, \lambda_1) + \Psi(\mathbf{x})]. \tag{33}$$

This inequality allows us to express q_k in terms of the linearizations of f at $\mathbf{u}_i$. For notational convenience, define $a_0 = 1$ and

$$b_k(i) := a_i \prod_{j=i+1}^{k}(1 - a_j) \quad \text{for all } 0 \le i \le k,$$

then it is easy to see that $\sum_{i=0}^{k} b_k(i) = 1$ for all $k \ge 1$.

Lemma 17. *For all $\mathbf{x} \in \operatorname{dom}\Psi$ and $k \ge 1$,*

$$q_k(\mathbf{x}) \le b_k(0)q_0(\mathbf{x}) + \sum_{i=1}^{k} b_k(i)[\ell_f(\mathbf{x}; \mathbf{u}_i, \lambda_1) + \Psi(\mathbf{x})]$$
$$= \frac{L}{\sigma}b_k(0)\Delta(\mathbf{x}, \mathbf{u}_0) + \Psi(\mathbf{x}) + \sum_{i=0}^{k} b_k(i)\ell_f(\mathbf{x}; \mathbf{u}_i, \lambda_1). \tag{34}$$

Proof. The inequality is obvious by inductively applying (33). The equality is by the definition of $q_0(\mathbf{x})$ and the fact that $\sum_{i=0}^{k} b_k(i) = 1$. ■

Go back to the settings of Section 3.3. We minimize $J(\mathbf{x})$ by AGM-EF-1 and find some dual iterates $\boldsymbol{\alpha}_k$ such that the duality gap $J(\mathbf{x}_k) - D(\boldsymbol{\alpha}_k)$ goes to 0 fast. Similar to (22), we construct

$$\boldsymbol{\alpha}_k = \sum_{i=0}^{k} b_k(i)\boldsymbol{\alpha}(\mathbf{u}_i). \tag{35}$$

Comparing with (22), we can see that both formulae are convex combinations of all the past $\boldsymbol{\alpha}(\mathbf{u}_i)$ and higher weights are given to the later $\boldsymbol{\alpha}(\mathbf{u}_i)$. Computationally, $\boldsymbol{\alpha}_k$ can be efficiently updated by recursion

$$\boldsymbol{\alpha}_0 = \boldsymbol{\alpha}(\mathbf{u}_0), \text{ and } \boldsymbol{\alpha}_{k+1} = (1-a_{k+1})\boldsymbol{\alpha}_k + a_{k+1}\boldsymbol{\alpha}(\mathbf{u}_{k+1}).$$

To be self-contained, we state and prove the counterpart of Theorem 10 here.

Theorem 18 (Bounds on the duality gap)**.** *Suppose a sequence $\{\mathbf{x}_k, \mathbf{u}_k, \mathbf{z}_k\}$ is produced when AGM-EF-1 is applied to minimize $J(\mathbf{x})$ by treating f as λ_1-sc. Then the $\{\boldsymbol{\alpha}_k\}$ defined by (35) satisfies $\boldsymbol{\alpha}_k \in Q_2$ and*

$$J(\mathbf{x}_k) - D(\boldsymbol{\alpha}_k) \le \frac{L}{\sigma}b_k(0)\max_{\mathbf{x}\in\operatorname{dom}\Psi}\Delta(\mathbf{x}, \mathbf{u}_0). \tag{36}$$

Proof. Since $\boldsymbol{\alpha}(\mathbf{u}_i) \in Q_2$ and $\boldsymbol{\alpha}_k$ is a convex combination of them, so $\boldsymbol{\alpha}_k \in Q_2$. Clearly, (24) still holds. Denote the right-hand side of (36) as M. Now by using relationship (34) and Lemma 12, we have

$$J(\mathbf{x}_k) \le \min_{\mathbf{x}} q_k(\mathbf{x})$$
$$\le \min_{\mathbf{x}}\left\{\frac{L}{\sigma}b_k(0)\Delta(\mathbf{x}, \mathbf{u}_0) + \Psi(\mathbf{x}) + \sum_{i=0}^{k} b_k(i)\ell_f(\mathbf{x}; \mathbf{u}_i, \lambda_1)\right\}$$
$$\le M + \min_{\mathbf{x}}\left\{\Psi(\mathbf{x}) + \sum_{i=0}^{k} b_k(i)\phi(\mathbf{x}, \boldsymbol{\alpha}(\mathbf{u}_i))\right\}$$
$$\le M + \min_{\mathbf{x}}\left\{\Psi(\mathbf{x}) + \phi\left(\mathbf{x}, \sum_{i=0}^{k} b_k(i)\boldsymbol{\alpha}(\mathbf{u}_i)\right)\right\}$$
$$\le M + D(\boldsymbol{\alpha}_k).$$ ■

5. Application to Regularized Risk Minimization

Regularized risk minimization (RRM) is extensively used in machine learning. In this section, we describe and compare in theory many different ways of training these models by APM. The objective of RRM with linear models can be written as

$$\min_{\mathbf{w}\in Q_1} J(\mathbf{w}) = \Omega(\mathbf{w}) + g^\star(A\mathbf{w}), \tag{37}$$

where Q_1 is a closed convex set. Here, $\Omega(\mathbf{w})$ corresponds to the regularizer and is assumed to be λ-sc wrt some prox-function d_1 on Q_1. d_1 is in turn assumed to be σ_1-sc wrt a norm $\|\cdot\|$ on Q_1[7]. $A\mathbf{w}$ stands for the output of a linear model, and $g^\star$ (the Fenchel dual of function g) encodes the empirical risk measuring the discrepancy between the correct labels and the output of the linear model ($A\mathbf{w}$). Let the domain of g be Q_2, which is also assumed to be closed and convex.

Using the definition of Fenchel dual, the primal objective (37) can be rewritten as a minimax problem:

$$\min_{\mathbf{w}\in Q_1}\max_{\boldsymbol{\alpha}\in Q_2} \mathcal{L}(\mathbf{w},\boldsymbol{\alpha}) := \Omega(\mathbf{w}) + \langle A\mathbf{w},\boldsymbol{\alpha}\rangle - g(\boldsymbol{\alpha}), \tag{38}$$

which further leads to the adjoint problem

$$\begin{aligned}&\max_{\boldsymbol{\alpha}\in Q_2}\left\{-g(\boldsymbol{\alpha}) + \min_{\mathbf{w}\in Q_2}\{\langle A\mathbf{w},\boldsymbol{\alpha}\rangle + \Omega(\mathbf{w})\}\right\}\\ \Leftrightarrow &\max_{\boldsymbol{\alpha}\in Q_2} D(\boldsymbol{\alpha}) := -g(\boldsymbol{\alpha}) - \Omega^\star(-A^\top\boldsymbol{\alpha}).\end{aligned} \tag{39}$$

It is well known [*e.g.* 29, Theorem 3.3.5] that under some mild constraint qualifications, the primal form $J(\mathbf{w})$ and the adjoint form $D(\boldsymbol{\alpha})$ satisfy

$$J(\mathbf{w}) \geq D(\boldsymbol{\alpha}) \quad \text{and} \quad \inf_{\mathbf{w}\in Q_1} J(\mathbf{w}) = \sup_{\boldsymbol{\alpha}\in Q_2} D(\boldsymbol{\alpha}).$$

Let us see some examples in machine learning which have the form (37). Assume we have access to a training set of n labeled examples $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ where $\mathbf{x}_i \in \mathbb{R}^p$ and $y_i \in \{-1,+1\}$. Denote $Y := \mathrm{diag}(y_1,\ldots,y_n)$ and $X := (\mathbf{x}_1,\ldots,\mathbf{x}_n)$.

Example 1: binary SVMs with bias. The primal form of the binary linear SVM with bias is:

$$J(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|^2 + \min_{b\in\mathbb{R}}\frac{1}{n}\sum_{i=1}^n [1 - y_i(\langle\mathbf{x}_i,\mathbf{w}\rangle + b)]_+ .$$

This can be posed in our framework by setting $Q_1 := \mathbb{R}^p$, $A := -YX^\top$, $\Omega(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|^2$, $g^\star(\mathbf{u}) = \min_{b\in\mathbb{R}}\frac{1}{n}\sum_{i=1}^n [1+u_i - y_i b]_+$. This $g^\star$ corresponds to

$$g(\boldsymbol{\alpha}) = \begin{cases} -\sum_i \alpha_i & \text{if } \boldsymbol{\alpha}\in Q_2\\ +\infty & \text{otherwise,}\end{cases} \tag{40}$$

where Q_2, the domain of g, is

$$Q_2 = \left\{\boldsymbol{\alpha}\in[0,n^{-1}]^n : \sum_i y_i\alpha_i = 0\right\}.$$

Then the adjoint form turns out to be the well known SVM dual objective:

$$D(\boldsymbol{\alpha}) = \sum_i \alpha_i - \frac{1}{2\lambda}\boldsymbol{\alpha}^\top YX^\top XY\boldsymbol{\alpha},\ s.t.\ \boldsymbol{\alpha}\in Q_2 \tag{41}$$

Example 2: L_1 regularized SVM. The primal form of the L_1 regularized SVM (L_1-SVM, [30]) is:

$$J(\mathbf{w}) = \lambda\|\mathbf{w}\|_1 + \min_{b\in\mathbb{R}}\frac{1}{n}\sum_{i=1}^n [1 - y_i(\langle\mathbf{x}_i,\mathbf{w}\rangle + b)]_+ .$$

This can be posed in our framework by using exactly the same configurations as above, except that now $\Omega(\mathbf{w}) = \lambda\|\mathbf{w}\|_1$. One can show that $\Omega^\star(\mathbf{v}) = 0$ if $\|\mathbf{v}\|_\infty \leq \lambda$, and ∞ otherwise. The adjoint form is:

$$D(\boldsymbol{\alpha}) = \begin{cases}\sum_i \alpha_i & \text{if } \|XY\boldsymbol{\alpha}\|_\infty \leq \lambda\\ -\infty & \text{otherwise.}\end{cases} \quad s.t.\ \boldsymbol{\alpha}\in Q_2. \tag{42}$$

Example 3: multivariate scores. Joachims [31] proposed a max-margin model which directly optimizes the F_1 score. Assume there are n_+ positive examples and n_- negative examples. F_1-score is defined by using the contingency table: $\Delta(\mathbf{y}',\mathbf{y}) := \frac{2a}{2a+b+c}$.

Contingency table.

	$y=1$	$y=-1$
$y'=1$	a	c
$y'=-1$	b	d

b: false negative
c: false positive

$$b = \sum_{i=1}^n \delta(y_i = 1, y_i' = -1)$$
$$c = \sum_{i=1}^n \delta(y_i = -1, y_i' = 1)$$
$$a = n_+ - b$$
$$d = n_- - c. \quad (n = n_+ + n_-)$$
$$\delta(x) = 1 \text{ if } x \text{ is true. Else } 0.$$

The primal objective proposed by Joachims [31] is

$$J(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|^2 + \max_{\mathbf{y}'\in\{-1,1\}^n}\left[\Delta(\mathbf{y}',\mathbf{y}) + \frac{1}{n}\sum_{i=1}^n \langle\mathbf{w},\mathbf{x}_i\rangle(y_i' - y_i)\right]. \tag{43}$$

[7] In the sequel, $\|\cdot\|_p$ will stand for the L_p norm. Since each space has a single prescribed norm and the space that a variable belongs to is clear from the context, we will not use $\|\cdot\|_1$ to represent the norm on Q_1.

This can be recovered by setting $Q_1 = \mathbb{R}^p$, $\Omega(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|^2$, and letting A be a 2^n-by-p matrix where the $\mathbf{y}'$-th row is $\sum_{i=1}^n \mathbf{x}_i^\top (y'_i - y_i)$ for each $\mathbf{y}' \in \{-1,+1\}^n$. Then $g^\star(\mathbf{u}) = \max_{\mathbf{y}'} \left[\Delta(\mathbf{y}', \mathbf{y}) + \frac{1}{n} u_{\mathbf{y}'}\right]$ which is induced by

$$g(\boldsymbol{\alpha}) = \begin{cases} -n \sum_{\mathbf{y}'} \Delta(\mathbf{y}', \mathbf{y}) \alpha_{\mathbf{y}'} & \text{if } \boldsymbol{\alpha} \in Q_2 \\ +\infty & \text{otherwise} \end{cases}. \quad (44)$$

Here Q_2, the domain of g, is

$$Q_2 = \left\{ \boldsymbol{\alpha} \in [0, n^{-1}]^{2^n} : \sum_{\mathbf{y}'} \alpha_{\mathbf{y}'} = \frac{1}{n} \right\}.$$

So we get the adjoint form

$$D(\boldsymbol{\alpha}) = -\frac{1}{2\lambda} \boldsymbol{\alpha}^\top A A^\top \boldsymbol{\alpha} + n \sum_{\mathbf{y}'} \Delta(\mathbf{y}', \mathbf{y}) \alpha_{\mathbf{y}'}, \ \boldsymbol{\alpha} \in Q_2.$$

Example 4: Max-margin Markov Networks. The conditional random fields (CRFs) [32] and max-margin Markov network ($\mathsf{M^3Ns}$), [33] are also instances of RRM. First, they both minimize a regularized risk with a square norm regularizer. Second, they assume that there is a joint feature map $\boldsymbol{\phi}$ which maps $(\mathbf{x}, \mathbf{y})$ to a feature vector in $\mathbb{R}^p$. Third, they assume a label loss $\ell(\mathbf{y}, \mathbf{y}^i; \mathbf{x}^i)$ which quantifies the loss of predicting label $\mathbf{y}$ when the correct label of input $\mathbf{x}^i$ is $\mathbf{y}^i$. Finally, they assume that the space of labels $\mathcal{Y}$ is endowed with a graphical model structure and that $\boldsymbol{\phi}(\mathbf{x}, \mathbf{y})$ and $\ell(\mathbf{y}, \mathbf{y}^i; \mathbf{x}^i)$ factorize according to the cliques of this graphical model. The main difference is in the loss function employed. CRFs minimize the L_2-regularized logistic loss:

$$J(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{n}\sum_{i=1}^n \log \sum_{\mathbf{y} \in \mathcal{Y}} \exp(\ell(\mathbf{y}, \mathbf{y}^i; \mathbf{x}^i) - \left\langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}^i, \mathbf{y}^i) - \boldsymbol{\phi}(\mathbf{x}^i, \mathbf{y}) \right\rangle), \quad (45)$$

while the $\mathsf{M^3Ns}$ minimize the L_2-regularized hinge loss

$$J(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{n}\sum_{i=1}^n \max_{\mathbf{y} \in \mathcal{Y}} \{\ell(\mathbf{y}, \mathbf{y}^i; \mathbf{x}^i) - \left\langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}^i, \mathbf{y}^i) - \boldsymbol{\phi}(\mathbf{x}^i, \mathbf{y}) \right\rangle\}. \quad (46)$$

Clearly, both cases employ $Q_1 = \mathbb{R}^p$ and $\Omega(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|^2$. With shorthand $\boldsymbol{\psi}^i_{\mathbf{y}} := \boldsymbol{\phi}(\mathbf{x}^i, \mathbf{y}^i) - \boldsymbol{\phi}(\mathbf{x}^i, \mathbf{y})$ and $\ell^i_{\mathbf{y}} := \ell(\mathbf{y}, \mathbf{y}^i; \mathbf{x}^i)$, they both use an $(n|\mathcal{Y}|)$-by-p matrix A whose $(i, \mathbf{y})$-th row is $(-\boldsymbol{\psi}^i_{\mathbf{y}})^\top$. For $\mathsf{M^3Ns}$, $g^\star(\mathbf{u}) = \frac{1}{n}\sum_i \max_{\mathbf{y}} \left\{\ell^i_{\mathbf{y}} + u^i_{\mathbf{y}}\right\}$ and it can be verified that the corresponding g is

$$g(\boldsymbol{\alpha}) = \begin{cases} -\sum_i \sum_{\mathbf{y}} \ell^i_{\mathbf{y}} \alpha^i_{\mathbf{y}} & \text{if } \boldsymbol{\alpha} \in Q_2 \\ +\infty & \text{otherwise,} \end{cases} \quad (47)$$

where Q_2, the domain of g, is

$$Q_2 = \mathcal{S}^n := \left\{ \boldsymbol{\alpha} \in [0,1]^{n|\mathcal{Y}|} : \sum_{\mathbf{y}} \alpha^i_{\mathbf{y}} = \frac{1}{n}, \ \forall\ i \right\}.$$

Clearly, Q_2 is convex and compact. Now the adjoint form can be written as

$$D(\boldsymbol{\alpha}) = -\frac{1}{2\lambda} \boldsymbol{\alpha}^\top A A^\top \boldsymbol{\alpha} + \sum_i \sum_{\mathbf{y}} \ell^i_{\mathbf{y}} \alpha^i_{\mathbf{y}}, \ \boldsymbol{\alpha} \in \mathcal{S}^n. \quad (48)$$

For CRFs, $g^\star(\mathbf{u}) = \frac{1}{n}\sum_i \log \sum_{\mathbf{y} \in \mathcal{Y}} \exp(\ell^i_{\mathbf{y}} + u^i_{\mathbf{y}})$, and the corresponding g is

$$g(\boldsymbol{\alpha}) = \begin{cases} \sum_{i=1}^n \sum_{\mathbf{y}} \alpha^i_{\mathbf{y}} (\log \alpha^i_{\mathbf{y}} - \ell^i_{\mathbf{y}}) + \log n & \text{if } \boldsymbol{\alpha} \in Q_2 \\ +\infty & \text{otherwise,} \end{cases} \quad (49)$$

The domain of g is also $Q_2 = \mathcal{S}^n$. Then the adjoint form is

$$D(\boldsymbol{\alpha}) = -\frac{1}{2\lambda} \boldsymbol{\alpha}^\top A A^\top \boldsymbol{\alpha} + \sum_{i=1}^n \sum_{\mathbf{y}} \alpha^i_{\mathbf{y}} (\log \alpha^i_{\mathbf{y}} - \ell^i_{\mathbf{y}}) + \log n, \quad \boldsymbol{\alpha} \in \mathcal{S}^n. \quad (50)$$

Example 5: Entropy regularized LPBoost In [13], the entropy regularized LPBoost needs to minimize

$$J(\mathbf{w}) = \lambda \Delta(\mathbf{w}, \mathbf{w}^0) + \max_{i \in [t]} \langle \mathbf{u}_i, \mathbf{w} \rangle, \quad (51)$$
$$s.t.\ \mathbf{w} \in Q_1 := \left\{ \mathbf{w} \in [0, \nu]^n : \sum_{i=1}^n w_i = 1 \right\}.$$

Here ν is a constant in $[0,1]$, $\mathbf{w}^0 \in Q_1$ is the uniform distribution, and Δ is the Bregman divergence induced by the entropy (*i.e.* Δ is the relative entropy). $\mathbf{u}_i \in \mathbb{R}^n$ is the so called edge vector. This objective corresponds to $\Omega(\mathbf{w}) = \lambda \Delta(\mathbf{w}, \mathbf{w}^0)$, $A = (\mathbf{u}_1, \ldots, \mathbf{u}_t)^\top$, $g^\star(\mathbf{s}) = \max_i s_i$ which is induced by $g(\boldsymbol{\alpha}) = 0$ if $\boldsymbol{\alpha} \in Q_2 := \mathcal{S}_t$, and ∞ otherwise. Since

$$\Omega^\star(\mathbf{s}) = -\min_{\beta_i \geq 0} \left\{ \lambda \log \sum_{i=1}^n w^0_i \exp\left(\frac{s_i - \beta_i}{\lambda}\right) + \nu \sum_{i=1}^n \beta_i \right\},$$

so the adjoint form can be written as

$$D(\boldsymbol{\alpha}) = -\min_{\beta_i \geq 0} \left\{ \lambda \log \sum_{i=1}^n w^0_i \exp\left(\frac{A_{:i}^\top \boldsymbol{\alpha} + \beta_i}{-\lambda}\right) + \nu \sum_{i=1}^n \beta_i \right\}$$

subject to $\boldsymbol{\alpha} \in Q_2 = \mathcal{S}_t$. Here $A_{:i}$ denotes the i-th column of A. Although this form of $D(\boldsymbol{\alpha})$ is obscure, the strong convexity of Ω implies that $D(\boldsymbol{\alpha})$ is *l.c.g.* The ν is introduced by [13] to cap the density, and this cap is removed if $\nu = \infty$. In that case, β_i in the definition of $D(\boldsymbol{\alpha})$ will all be optimized to 0 and we recover the well known log-sum-exp formula of $D(\boldsymbol{\alpha})$.

Example 6: Elastic net Using square loss as an example of the empirical risk, the primal objective of elastic net regularization is

$$J(\mathbf{w}) = \lambda \left(\gamma \|\mathbf{w}\|_1 + \frac{1}{2} \|\mathbf{w}\|_2^2 \right) + \frac{1}{n} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^\top \mathbf{w})^2. \quad (52)$$

Here the L_1 normalizer $\|\mathbf{w}\|_1$ is introduced to promote the sparsity of the solution. In this case, $\Omega(\mathbf{w}) = \lambda \left(\gamma \|\mathbf{w}\|_1 + \frac{1}{2} \|\mathbf{w}\|_2^2 \right)$ and it dual is left as an exercise for the reader. An equivalent formulation of (52) is by moving the regularizer into the constraint:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \bar{J}(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^\top \mathbf{w})^2 \\ s.t. \quad & \gamma \|\mathbf{w}\|_1 + \frac{1}{2} \|\mathbf{w}\|_2^2 \le r. \end{aligned}$$

It can be shown that for any $\lambda > 0$ there exists an $r > 0$ such that $\operatorname{argmin} \bar{J} = \operatorname{argmin} J$ and vice versa.

There are also many regularized risk minimization problems which optimize over the space of positive semi-definite matrices, *e.g.* [2, 14, 34].

Summary From these examples, we can see the following properties of Ω and g which will also be assumed for our general treatment of the objective (37) and (39). Firstly, the function $\Omega(\mathbf{w})$ which serves as a regularizer is strongly convex. In Example 1, 3, 4, 6, $\Omega(\mathbf{w})$ is λ-sc wrt the Euclidean norm. In Example 5, $f(\mathbf{w})$ is λ-sc wrt the L_1 norm. As a result, $\Omega^\star$ must be $\frac{1}{\lambda}$-*l.c.g* on $\mathbb{R}^p$. Secondly, the *l.c.g* constant of $\Omega^\star(-A^\top \boldsymbol{\alpha})$ in $\boldsymbol{\alpha}$ also depends on the matrix norm of A, which in turn depends on the choice of norm on Q_1 and Q_2. Thirdly, the $g^\star$ is not necessarily differentiable (*e.g.*, hinge loss), but g is always *l.c.g* on Q_2. Finally, Q_2 is bounded and its diameter can be well controlled. This is important for translating dual solutions into the primal.

Our goal is to minimize $J(\mathbf{w})$ over Q_1, and we do not really care about solving the dual $D(\boldsymbol{\alpha})$ over Q_2. However, since $D(\boldsymbol{\alpha})$ has favorable smooth properties, we also often work in the dual as a proxy. To solve $J(\mathbf{w})$ (and $D(\boldsymbol{\alpha})$), there are three main approaches.

Smoothing $g^\star$ to a fixed level. To handle the nonsmoothness of $g^\star$, we can smooth it by using the technique introduced by Nesterov [6]. Then the composite form, $\Omega(\mathbf{w})$ plus the smoothed variant of $g^\star(A\mathbf{w})$, fits the form of AGM-EF and can be solved in $\mathbf{w}$ (primal), $\boldsymbol{\alpha}$ (dual) or primal-dual. Given a prescribed accuracy ϵ, $g^\star$ only needs to be smoothed to a fixed extent.

Smoothing $g^\star$ with decreasing smoothness. [20] introduced a primal-dual method where $g^\star$ is smoothed with decreased smoothness (*i.e.* increased closeness to $g^\star$). As a result, it tends to the optimal solution of $D(\boldsymbol{\alpha})$ and $J(\mathbf{w})$, instead of just attaining a prescribed accuracy ϵ.

No smoothing. Given the smoothness of the dual problem $D(\boldsymbol{\alpha})$, AGM can be applied to maximize it and then convert $\boldsymbol{\alpha}_k$ into $\mathbf{w}_k$ by (22) and (35). No smoothing of $g^\star$ is needed in this case.

The next three subsections will describe these schemes in detail, with focus on the rates of convergence and how each iteration can be performed efficiently. Moreover, we provide intuitions on which scheme is more suitable. For brevity, we will only use AGM-EF-∞ with fixed L as an example, while similar results can be straightforwardly derived for AGM-EF-1 and adaptive L. In this version of the paper, we illustrate all these ideas on Example 1 (SVM with bias).

5.1. Smoothing $g^\star$ to a fixed level

A key technique introduced by Nesterov [6] was to tightly approximate the nonsmooth part $g^\star(A\mathbf{w})$ by a smooth surrogate. The idea of the approach originates from the Theorem 21 in Appendix A which connects the strong convexity of a function and *l.c.g* of its Fenchel dual. $g^\star$ is not *l.c.g* because g is not strongly convex, therefore to make $g^\star$ smooth a natural idea is to add to g a strongly convex function d_2 on Q_2 and then dualize it:

$$\begin{aligned} g_\mu^\star(\mathbf{u}) &:= (g + \mu d_2)^\star(\mathbf{u}) \\ &= \sup_{\boldsymbol{\alpha} \in Q_2} \left\{ \langle \boldsymbol{\alpha}, \mathbf{u} \rangle - g(\boldsymbol{\alpha}) - \mu\, d_2(\boldsymbol{\alpha}) \right\}. \end{aligned} \quad (53)$$

Here $\mu \ge 0$ and d_2 is assumed to be σ_2-sc wrt a norm on Q_2.[8] By proper centering, d_2 can be assumed to satisfy

$$\min_{\boldsymbol{\alpha} \in Q_2} d_2(\boldsymbol{\alpha}) = 0.$$

Let us further define

$$\boldsymbol{\alpha}_0 = \operatorname*{argmin}_{\boldsymbol{\alpha} \in Q_2} d_2(\boldsymbol{\alpha}), \quad D := \max_{\boldsymbol{\alpha} \in Q_2} d_2(\boldsymbol{\alpha}).$$

The main restriction of this approach is that D must be well bounded. Using the definition in (53) we can easily characterize the *uniform* tightness of the approximation: for all $\mathbf{u} \in Q_2$

$$g^\star(\mathbf{u}) - \mu D \le g_\mu^\star(\mathbf{u}) \le g^\star(\mathbf{u}). \quad (54)$$

[8] We can also use the more general form of strong convex as in Definition 1. Here we use the conventional definition for simplicity.

Furthermore, the *l.c.g* constant of $g_\mu^\star(A\mathbf{w})$ in $\mathbf{w}$ wrt the norm on Q_1 can be estimated as follows. By Theorem 21, $g_\mu^\star$ is $(\mu\sigma_2)^{-1}$-*l.c.g* wrt the dual norm on Q_2. So we can apply the chain rule:

$$\begin{aligned}&\left\|\frac{\partial}{\partial \mathbf{w}} g_\mu^\star(A\mathbf{w}_1) - \frac{\partial}{\partial \mathbf{w}} g_\mu^\star(A\mathbf{w}_2)\right\|^* \\ &= \left\|A(\nabla g_\mu^\star(A\mathbf{w}_1) - \nabla g_\mu^\star(A\mathbf{w}_2)\right\|^* \\ &\le \|A\| \frac{1}{\mu\sigma_2} \|A\mathbf{w}_1 - A\mathbf{w}_2\|^* \le \frac{\|A\|^2}{\mu\sigma_2} \|\mathbf{w}_1 - \mathbf{w}_2\|.\end{aligned}$$

That is, $g_\mu^\star(A\mathbf{w})$ is *l.c.g* in $\mathbf{w}$ with constant

$$L_g(\mu) \le \frac{\|A\|^2}{\mu\sigma_2}. \tag{55}$$

Example 1: smoothing the hinge loss. The hinge loss $[1-w]_+$ is the dual of $g(\alpha) = \alpha$ for $\alpha \in [-1, 0]$ and ∞ elsewhere. Adding $\frac{\mu}{2}\alpha^2$ to g and dualize it, we get

$$g_\mu^\star(w) = \begin{cases} 0 & \text{if } w \ge 1 \\ \frac{(1-w)^2}{2\mu} & \text{if } w \in [1-\mu, 1] \\ 1 - w - \frac{\mu}{2} & \text{if } w \le 1-\mu \end{cases}.$$

Some smoothed hinge loss $g_\mu^\star(w)$ with various μ are plotted in Figure 1.

Example 2: smoothing max into soft max. In the entropy regularized LPBoost, $g^\star(\mathbf{s}) = \max_i s_i$ and $g(\mathbf{u}) = 0$ if $\mathcal{S}_t$ and ∞ otherwise.. Then adding prox-function $\sum_i s_i \ln s_i$ to g and dualizing it, we get

$$g_\mu^\star(\mathbf{s}) = \mu \ln \sum_i \exp\left(\frac{s_i}{\mu}\right).$$

When $\mu \to 0$, this soft max recovers max.

With the smoothed $g_\mu^\star$ in place, we now discuss how to find an ϵ accurate solution to $J(\mathbf{w})$ by three different schemes: primal ($\mathbf{w}$), dual ($\boldsymbol{\alpha}$), and primal-dual.

5.1.1. Solving in the primal $\mathbf{w}$.

We will use $g_\mu^\star$ to define a new objective function

$$\begin{aligned}J_\mu(\mathbf{w}) &:= \Omega(\mathbf{w}) + g_\mu^\star(A\mathbf{w}) \\ &= \Omega(\mathbf{w}) + \max_{\boldsymbol{\alpha}\in Q_2} \{\langle A\mathbf{w}, \boldsymbol{\alpha}\rangle - g(\boldsymbol{\alpha}) - \mu\, d_2(\boldsymbol{\alpha})\}.\end{aligned} \tag{56}$$

Since $J_\mu(\mathbf{w}) \le J(\mathbf{w})$ for all $\mathbf{w}$, to make sure that an ϵ accurate solution to J_μ is a 2ϵ accurate solution to J, a sufficient condition is that their deviation be upper

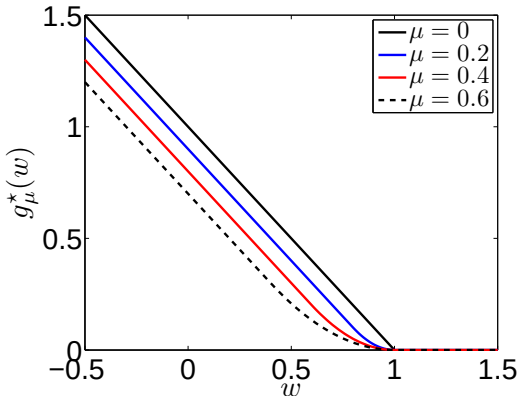


Figure 1. Smoothing hinge loss with different μ.

bounded *everywhere* by ϵ, *i.e.* $\max_{\mathbf{w}} J(\mathbf{w}) - J_\mu(\mathbf{w}) < \epsilon$. By (54), this is guaranteed if μ is small enough

$$\mu \le \frac{\epsilon}{D}. \tag{57}$$

Plugging (57) into (55), we obtain that the *l.c.g* constant of $g_\mu^\star(A\mathbf{w})$ is at most $\frac{\|A\|^2 D}{\sigma_2 \epsilon}$. Let $\mathbf{w}^* = \operatorname{argmin}_{\mathbf{w}} J(\mathbf{w})$. Bearing in mind that Ω is λ-sc, AGM-EF-∞ is readily applicable to $J_\mu(\mathbf{w})$ and the following rate of convergence can be inferred from Theorem 6:

$$\begin{aligned}J_\mu(\mathbf{w}_k) - J_\mu(\mathbf{w}^*) \le \Delta(\mathbf{w}^*, \mathbf{u}_0) \min \Bigg\{ &\frac{4D\|A\|^2}{\sigma_1\sigma_2\epsilon(k+1)^2}, \\ &\frac{4D\|A\|^2}{\sigma_1\sigma_2\epsilon}\left(1 + \sqrt{\frac{\sigma_1\sigma_2\lambda\epsilon}{4D\|A\|^2}}\right)^{-2k+2}\Bigg\}.\end{aligned}$$

Once $J_\mu(\mathbf{w}_k) - J_\mu(\mathbf{w}^*) \le \epsilon$, we must have

$$J(\mathbf{w}_k) - J(\mathbf{w}^*) \le J_\mu(\mathbf{w}_k) + \mu D - J_\mu(\mathbf{w}^*) \le 2\epsilon.$$

Therefore, we obtain the following theorem.

Theorem 19. *For any given $\epsilon > 0$, setting μ by the equality in (57) and applying AGM-EF-∞ to $J_\mu(\mathbf{w})$, we can guarantee that $\mathbf{w}_k$ is a 2ϵ accurate solution of $J(\mathbf{w})$ as long as*

$$\begin{aligned}k \ge \min \Bigg\{ &\frac{1}{\epsilon}\sqrt{\frac{4D\|A\|^2}{\sigma_1\sigma_2}\Delta(\mathbf{w}^*, \mathbf{u}_0)}, \\ &1 + \frac{1}{2}\ln\left(\frac{4D\|A\|^2}{\sigma_1\sigma_2\epsilon^2}\Delta(\mathbf{w}^*, \mathbf{u}_0)\right) \Big/ \ln\left(1 + \sqrt{\frac{\sigma_1\sigma_2\lambda\epsilon}{4D\|A\|^2}}\right)\Bigg\}.\end{aligned} \tag{58}$$

Note $\ln(1+\epsilon) \approx \epsilon$ when ϵ is close to 0, so the denominator in the second term becomes $O(\sqrt{\epsilon})$ and overall the second term is approximately $O\left(\frac{1}{\sqrt{\epsilon}} \ln \frac{1}{\epsilon}\right)$. The

first term does not depend on λ. Note also that this bound does not explicitly depend on the diameter of Q_1 which is infinity in many cases. A closer look shows that $\Delta(\mathbf{w}^*, \mathbf{u}_0)$ hides the dependence on λ. With a small regularization parameter λ, $\Delta(\mathbf{w}^*, \mathbf{u}_0)$ may be large and could approach infinity when λ tends to 0.

Unfortunately, the bound on the duality gap in (23) does use the diameter of Q_1, and it cannot be replaced by $\Delta(\mathbf{w}^*, \mathbf{u}_0)$ as in Theorem 19. Therefore, we do lose a termination criteria. Fortunately, this problem in duality gap can be avoided if we optimize in $\boldsymbol{\alpha}$. Before describing it in detail, let us illustrate the above procedure on training the SVM with bias.

Here, choose d_1 and d_2 as the Euclidean norm square and the norms on Q_1 and Q_2 are both Euclidean norm. Then $\|A\|^2 = \lambda_{\max}(A^\top A) = \lambda_{\max}(XX^\top)$ where $\lambda_{\max}$ stands for the maximum eigenvalue. $\sigma_1 = \sigma_2 = 1$. The diameter of Q_2 is $D \leq n\frac{1}{n^2} = \frac{1}{n}$. For a given ϵ, set $\mu = n\epsilon$ by (57). Suppose all $\mathbf{x}_i$ lie in the ball with Euclidean radius R. Then $\lambda_{\max}(XX^\top) \leq nR^2$ and the second term in (58) is essentially

$$O\left(\ln\frac{1}{\epsilon}\sqrt{\frac{4\lambda_{\max}(XX^\top)}{\lambda n\epsilon}}\right) \leq O\left(\frac{2R}{\sqrt{\lambda\epsilon}}\ln\frac{1}{\epsilon}\right).$$

Solving in the primal is also advantageous in terms of the condition number. When $g^\star$ is smoothed by small μ or when the regularization parameter λ is small, the condition number $c := L_g(\mu)/\lambda$ becomes very large. According to Theorem 6, the number of iterations to find an ϵ accurate solution is the min of

$$O\left(\frac{\log\frac{1}{\epsilon}}{\log\left(1+\sqrt{\frac{\lambda}{L_g(\mu)}}\right)}\right) \quad \text{and} \quad O\left(\frac{\sqrt{L_g(\mu)}}{\sqrt{\epsilon}}\right).$$

So the linear convergence rate depends on c by $O(\sqrt{c})$, as opposed to $O(c)$ in most linearly converging algorithms, *e.g.* gradient descent. Second, the min in Theorem 6 implies that when λ is very small and the objective is very poorly conditioned, the linear convergence will be automatically superseded by the $1/\sqrt{\epsilon}$ rate which has better "constant". Some class of algorithms require manual rewiring in such a case, *e.g.* [25] and [35].

Finally, it is noteworthy that this method does not require g be *l.c.g.*

5.1.2. Solving in the dual $\boldsymbol{\alpha}$.

Similar to J_μ in (56), we can also define a smoothed version of $D(\boldsymbol{\alpha})$:

$$\begin{aligned} D_\mu(\boldsymbol{\alpha}) &:= -\mu d_2(\boldsymbol{\alpha}) - g(\boldsymbol{\alpha}) + \min_{\mathbf{w}}\{\Omega(\mathbf{w}) + \langle A\mathbf{w}, \boldsymbol{\alpha}\rangle\} \\ &= -\mu d_2(\boldsymbol{\alpha}) - g(\boldsymbol{\alpha}) - \Omega^\star(-A^\top\boldsymbol{\alpha}) \end{aligned} \tag{59}$$

which is to be maximized over $\boldsymbol{\alpha} \in Q_2$. So we can pose $-D_\mu(\boldsymbol{\alpha})$ in the composite form,

$$f(\boldsymbol{\alpha}) = g(\boldsymbol{\alpha}) + \Omega^\star(-A^\top\boldsymbol{\alpha}), \text{ and } \Psi(\boldsymbol{\alpha}) = \mu d_2(\boldsymbol{\alpha}),$$

to which AGM-EF-∞ and AGM-EF-1 can be applied. Since $\Omega^\star$ is $1/\lambda$-*l.c.g*, $f(\boldsymbol{\alpha})$ must be *l.c.g* with constant

$$L_f = \frac{\|A\|^2}{\lambda} + L_g, \tag{60}$$

where L_g is the *l.c.g* constant of g. Ψ is μ-sc. Applying the primal-dual scheme in Section 3.3 with $-D_\mu$ and $-J_\mu$ playing the role of J and D therein respectively, we get

$$J_\mu(\mathbf{w}_k) - D_\mu(\boldsymbol{\alpha}_k) \leq \max_{\boldsymbol{\alpha}\in Q_2}\Delta(\boldsymbol{\alpha}, \mathbf{u}_0)\cdot\min\left\{\frac{4L_f}{\sigma_2(k+1)^2}, \frac{L_f}{\sigma_2}\left(1+\sqrt{\frac{\sigma_2\mu}{4L_f}}\right)^{-2k+2}\right\}.$$

Once $J_\mu(\mathbf{w}_k) - D_\mu(\boldsymbol{\alpha}_k) \leq \epsilon$, it is ensured that

$$\begin{aligned} J(\mathbf{w}_k) - \min_{\mathbf{w}} J(\mathbf{w}) &\leq J_\mu(\mathbf{w}_k) + \mu D - \max_{\boldsymbol{\alpha}\in Q_2} D(\boldsymbol{\alpha}) \\ &\leq J_\mu(\mathbf{w}_k) + \epsilon - D(\boldsymbol{\alpha}_k) \\ &\leq J_\mu(\mathbf{w}_k) + \epsilon - D_\mu(\boldsymbol{\alpha}_k) \leq 2\epsilon. \end{aligned}$$

So we conclude the following theorem.

Theorem 20. *For any given $\epsilon > 0$, setting μ by the equality in (57) and applying the primal-dual scheme in Section 3.3 to $-D_\mu$ and $-J_\mu$, we can guarantee that $\mathbf{w}_k$ is a 2ϵ accurate solution of $J(\mathbf{w})$ as long as*

$$k \geq \min\left\{\frac{1}{\sqrt{\epsilon}}\sqrt{\frac{4M(\|A\|^2+\lambda L_g)}{\lambda\sigma_2}} - 1, \quad 1 + \frac{1}{2}\frac{\ln\left(\frac{4M(\|A\|^2+\lambda L_g)}{\lambda\sigma_2\epsilon}\right)}{\ln\left(1+\sqrt{\frac{\lambda\epsilon\sigma_2}{4D(\|A\|^2+\lambda L_g)}}\right)}\right\}. \tag{61}$$

where $M := \max_{\boldsymbol{\alpha}\in Q_2}\Delta(\boldsymbol{\alpha}, \mathbf{u}_0)$.

It is important to note that this scheme requires g be *l.c.g*, while solving in the primal does not make such a requirement.

Let us apply the scheme to SVM with bias, and use the same choice of norm and prox-function as before. Now $L_g = 0$ and $M = 1/n$. Using the approximation $\ln(1+x) \approx x$ when $|x| \ll 1$, (61) becomes

$$k \geq \min\left\{\frac{2R}{\sqrt{\lambda\epsilon}} - 1, 1 + \frac{R}{\sqrt{\lambda\epsilon}} \ln\left(\frac{4R^2}{\lambda\epsilon}\right)\right\}.$$

As a final note, the way we smooth the empirical risk is different from [36] which changes hinge loss into square hinge loss or higher order. Our method has a smoothing parameter which trades smoothness for the tightness of the approximation. In contrast, the square hinge loss is just a heuristic approximation and no bound is available in optimization for its solution.

5.2. Smoothing $g^\star$ with decreasing smoothness

A typical primal-dual solver for the objectives in (37) and (38) is the excessive gap technique [EGT, 20]. One concrete application is [37] where EGT is used to solve the above Example 4 ($\mathsf{M^3N}$ and CRF). Unfortunately, EGT forces a fixed way to initialize $\mathbf{w}_0$ and $\boldsymbol{\alpha}_0$. This is very inconvenient for homology and other warm-start techniques which utilize the closeness of solutions under small perturbations of the problem parameter (*e.g.* λ).

5.3. No smoothing of $g^\star$

Since we assume g is *l.c.g* and Ω is λ-sc, so the dual (39) is *l.c.g* and AGM-EF-∞ is applicable. Since our ultimate goal is to minimize $J(\mathbf{w})$ we adopt the primal-dual scheme in Section 3.3. The *l.c.g* constant of D is exactly the L_f in (60). Treating $-D$ and $-J$ as the J and D therein respectively, we get

$$J(\mathbf{w}_k) - D(\boldsymbol{\alpha}_k) \leq \frac{4M(\|A\|^2 + \lambda L_g)}{\lambda\sigma_2(k+1)^2}.$$

When applied to SVM with bias where $M = 1/n$ and $L_g = 0$, we get that $J(\mathbf{w}_k) - D(\boldsymbol{\alpha}_k) < \epsilon$ for all

$$k \geq \frac{2R}{\sqrt{\lambda\epsilon}} - 1.$$

When comparing the rates, it is important to bear in mind that machine learning problems usually do not need a high accuracy solution and so $\epsilon = 10^{-2}$ or 10^{-3} might suffice. In many cases, λ will be set to very small such as 10^{-6}. Therefore $\frac{1}{\epsilon}$ can be much smaller than $\frac{1}{\lambda}$. Also, we are currently bounding $\|A\|^2$ by nR^2 which can be very loose in practice. The dependence of $\Delta(\mathbf{w}^*, \mathbf{u}_0)$ on λ is not clear either. Finally in practice when solving in the dual, the box constraints in SVM can cause considerable waste of gradient computation. Therefore the rates above just provide limited guidance and the most appropriate optimization strategy has to be picked empirically.

5.4. Efficient computation of the gradient

So far, we have ignored the computational complexity per iteration which is dominated by two operations: computing the gradient and minimizing the model ψ_k in AGM-EF-∞ (or q_k in AGM-EF-1). We first show in this subsection that the gradient in all the above examples can be computed efficiently. Indeed, the gradients needed are $\frac{\partial}{\partial \mathbf{w}} g_\mu^\star(A\mathbf{w})$ and $\frac{\partial}{\partial \boldsymbol{\alpha}} \Omega^\star(-A^\top\boldsymbol{\alpha})$, with the former always being more challenging. So we focus on calculating $\frac{\partial}{\partial \mathbf{w}} g_\mu^\star(A\mathbf{w})$.

By chain rule, $\frac{\partial}{\partial \mathbf{w}} g_\mu^\star(A\mathbf{w}) = A^\top \nabla g_\mu^\star(A\mathbf{w})$. Using [47, Theorem X.1.4.4], $\nabla g_\mu^\star(\mathbf{u})$ can be computed by

$$\nabla g_\mu^\star(\mathbf{u}) = \operatorname*{argmax}_{\boldsymbol{\alpha}\in Q_2} \langle \mathbf{u}, \boldsymbol{\alpha}\rangle - g(\boldsymbol{\alpha}) - \mu d_2(\boldsymbol{\alpha}). \tag{62}$$

In the case of multivariate score (43) and (44), the dimension of the domain of g is exponentially high in the number of training examples, and therefore it will be intractable to first compute $\nabla g_\mu^\star(A\mathbf{w})$ and then pre-multiply it with $A^\top$ (A has exponentially many rows). Similar tractability issues appear in learning with structured outputs as in $\mathsf{M^3N}$. Below we present a dynamic programming based algorithm, which costs $O(n^2)$ time and space complexity to calculate $A^\top \nabla g_\mu^\star(A\mathbf{w})$ for

$$d_2(\boldsymbol{\alpha}) = \sum_i \alpha_i \ln \alpha_i.$$

In this case, the optimization problem in (62) is

$$\min_{\boldsymbol{\alpha}\in Q_2} \mu \sum_{\mathbf{y}'} \alpha_{\mathbf{y}'} \ln \alpha_{\mathbf{y}'} - n \sum_{\mathbf{y}'} \Delta(\mathbf{y}', \mathbf{y}) \alpha_{\mathbf{y}'} - \sum_{\mathbf{y}'} u_{\mathbf{y}'} \alpha_{\mathbf{y}'}.$$

Noting that the $\mathbf{y}'$-th row of A is $\varphi_{\mathbf{y}'}^\top := \sum_i (y_i' - y_i)\mathbf{x}_i^\top$, we get $u_{\mathbf{y}'} = \varphi_{\mathbf{y}'}^\top \mathbf{w} = \sum_i (y_i' - y_i)\mathbf{x}_i^\top \mathbf{w}$. Following the standard procedures (*e.g.* [37, Lemma 8]), the optimal solution can be written as

$$\alpha_{\mathbf{y}'}^* := \frac{1}{nZ} \exp\left(\frac{1}{\mu}\sum_i y_i' \mathbf{x}_i^\top \mathbf{w} + \frac{n}{\mu}\Delta(\mathbf{y}', \mathbf{y})\right),$$
$$\text{where} \quad Z := \sum_{\mathbf{y}'} \exp\left(\frac{1}{\mu}\sum_i y_i' \mathbf{x}_i^\top \mathbf{w} + \frac{n}{\mu}\Delta(\mathbf{y}', \mathbf{y})\right).$$

So $\alpha_{\mathbf{y}'}^*$ can be interpreted as a distribution over $\mathbf{y}'$

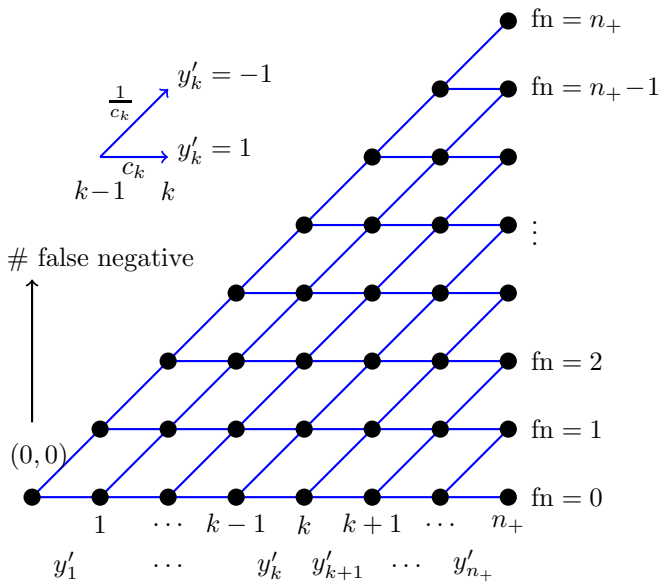


Figure 2. Path weight interpretation of the normalizer Z and the marginal distributions $p(y'_k)$.

(normalized to $\frac{1}{n}$ rather than 1). Then

$$\begin{aligned}\frac{\partial}{\partial \mathbf{w}} g^\star_\mu(A\mathbf{w}) &= \sum_{\mathbf{y}'} \alpha^*_{\mathbf{y}'} \varphi_{\mathbf{y}'} = \sum_{\mathbf{y}'} \alpha^*_{\mathbf{y}'} \sum_i (y'_i - y_i)\mathbf{x}_i \\ &= -2 \sum_i y_i \mathbf{x}_i \sum_{\mathbf{y}' \sim -y_i} \alpha^*_{\mathbf{y}'} \\ &= -2 \sum_i p(y'_i = -y_i) y_i \mathbf{x}_i,\end{aligned}$$

where $\mathbf{y}' \sim -y_i$ means summing up all $\mathbf{y}'$ whose i-th element y'_i equals $-y_i$. So $\sum_{\mathbf{y}' \sim -y_i} \alpha^*_{\mathbf{y}'}$ is exactly the marginal probability $p(y'_i = -y_i)$ under the joint distribution $\alpha^*_{\mathbf{y}'}$. Now we show how to compute the marginal distributions efficiently.

Unlike the inference in graphical models, there is no clique factorization in $\mathbf{y}'$. Fortunately, $\{y'_i\}$ are coupled only through the loss $\Delta(\mathbf{y}', \mathbf{y})$ which in turn depends only on two "sufficient statistics" of $\mathbf{y}'$: false negative b and false positive c. For simplicity, we sometimes also write $\Delta(\mathbf{y}', \mathbf{y})$ as $\Delta(b, c)$. Without loss of generality, assume the positive training examples are the first n_+ ones ($y_1 = \ldots = y_{n_+} = 1$), and the negative examples are the last $n - n_+$ ones ($y_{n_+ + 1} = \ldots = y_n = -1$). Denote $\mathbf{y}'_+ := (y'_1, \ldots, y'_{n_+})^\top$ and $\mathbf{y}'_- := (y'_{n_+ + 1}, \ldots, y'_n)^\top$. $\mathbf{y}'_+ \sim b$ represents that $\mathbf{y}'_+$ commits b false negatives, *i.e.* $\sum_{i=1}^{n_+} \delta(y'_i = -1) = b$. $\mathbf{y}'_- \sim c$ represents that $\mathbf{y}'_-$ commits c false negatives, *i.e.* $\sum_{i=n_+ + 1}^{n} \delta(y'_i = 1) = c$. For simplicity, denote

$$c_k := \exp\left(\frac{1}{\mu} \mathbf{x}_k^\top \mathbf{w}\right).$$

Let us first compute the normalizer Z as follows.

$$\begin{aligned}Z &= \sum_{b=0}^{n_+} \sum_{c=0}^{n_-} \sum_{\mathbf{y}'_+ \sim b} \sum_{\mathbf{y}'_- \sim c} \left(\frac{1}{\mu} \sum_{i=1}^{n} y'_i \mathbf{x}_i^\top \mathbf{w} + \frac{n}{\mu} \Delta(\mathbf{y}', \mathbf{y})\right) \\ &= \sum_{b=0}^{n_+} \sum_{c=0}^{n_-} \exp\left(\frac{n}{\mu} \Delta(b, c)\right) \underbrace{\sum_{\mathbf{y}'_+ \sim b} \exp\left(\frac{1}{\mu} \sum_{i=1}^{n_+} y'_i \mathbf{x}_i^\top \mathbf{w}\right)}_{=: V_+(b)} \\ &\quad \cdot \underbrace{\sum_{\mathbf{y}'_- \sim c} \exp\left(\frac{1}{\mu} \sum_{i=n_+ + 1}^{n} y'_i \mathbf{x}_i^\top \mathbf{w}\right)}_{=: V_-(c)}\end{aligned}$$

Therefore, once we have $V_+(b)$ for all $b \in [n_+]$ and $V_-(c)$ for all $c \in [n_-]$, then Z can be computed in $n_+ n_-$ steps. For simplicity we only show to compute $V_+(b)$, and $V_-(c)$ can be computed in exactly the same way.

For each fixed b, $V_+(b)$ can be equivalently reformulated by Figure 2. Each node (k, f) represents that $\mathbf{y}'$ has committed f false negatives in the first k examples: $\sum_{i=1}^{k} \delta(y'_i = -1) = f$. Each node is connected to two nodes on its right: $(k+1, f+1)$ and $(k+1, f)$. The former corresponds to $y'_{k+1} = -1$, *i.e.* one more false negative is committed. So we attach to the diagonal edge a weight $\exp\left(-\frac{1}{\mu} \mathbf{x}_{k+1}^\top \mathbf{w}\right) = c_{k+1}^{-1}$. The latter means $y'_{k+1} = 1$ and the false negative is not incremented. So the horizontal edge is attached with weight c_{k+1}. A *path* from (k, f) to (k', f') ($k \le k'$ and $f \le f'$) is a sequence of nodes moving from (k, f) to (k', f') along the edges of the graph: $(k, f_0) = (k, f) \to (k+1, f_1) \to \ldots \to (k+s, f_s) = (k', f')$ where $s = k' - k$ and $f_{i+1} - f_i = 0$ or 1. The weight of a path is defined as the product of the weight of all edges on that path.

Clearly $V_+(b)$ is equal to the total weight of all paths from $(0, 0)$ to (n_+, b). To compute it, define $\alpha_k(v)$ as the total weight of all paths from $(0, 0)$ to (k, v). Then it is not hard to see the following recursion for all $k = 1, \ldots, n_+$ and $v = 0, 1, \ldots, k$:

$$\alpha_k(v) = \frac{1}{c_k} \alpha_{k-1}(v - 1) + c_k \alpha_{k-1}(v), \qquad (63)$$

where $\alpha_k(-1) := 0$ and $\alpha_k(k+1) := 0$ for all k. Algorithm 5 computes $V_+(b) = \alpha_{n_+}(b)$ for all $b \in [n_+]$. Clearly the computational cost is $O(n_+^2)$. If we only

Algorithm 5 Forward propagation to compute all $\{V_+(b) : 0 \le b \le n_+\}$.

Initialize $\alpha_0(0) = 1$.
for $k = 1, \ldots, n_+$ **do**
  for $v = 0, 1, \ldots, k$ **do**
    $\alpha_k(v) = \frac{1}{c_k}\alpha_{k-1}(v-1) + c_k\alpha_{k-1}(v)$.
  end for
end for
Return: $V_+(b) = \alpha_{n_+}(b)$ for all $0 \le b \le n_+$.

Algorithm 6 Backward propagation to compute $p(y'_k)$ for all $k \in [n_+]$.

Initialize $\xi_{n_+}(v) = \eta_-(v)$ for all $v = 0, 1, \ldots, n_+$.
$Z_{n_+} = c_{n_+} \sum_{v=0}^{n_+} \alpha_{n_+-1}(v)\xi_{n_+}(v)$.
for $k = n_+ - 1, \ldots, 1$ **do**
  for $v = 0, 1, \ldots, k$ **do**
    $\xi_k(v) = c_{k+1}\xi_{k+1}(v) + \frac{1}{c_{k+1}}\xi_{k+1}(v+1)$.
  end for
  $Z_k = c_k \sum_{v=0}^{k-1} \alpha_{k-1}(v)\xi_k(v)$.
end for
Return: $p(y'_k = 1) = \frac{Z_i}{nZ}$ for all $k \in [n_+]$.

need $V_+(b)$ then the space complexity is $O(n_+)$. But later we will need all $\alpha_k(v)$ so we keep $O(n_+^2)$ memory. Taking into account the similar cost for $V_-(c)$, the total spatial and computational cost is both $O(n^2)$.

To compute the marginal distributions $p(y'_k)$ we need a backward propagation. For example let us consider $p(y'_k = 1)$ for $k \in [n_+]$, and the case of $k > n_+$ (negative examples) can be dealt with similarly. By the definition of $\alpha_{\mathbf{y}'}$, it suffices to compute

$$\begin{aligned}
Z_k &:= \sum_{\mathbf{y}':y'_k=1} \exp\left(\frac{1}{\mu}\sum_i y'_i\mathbf{x}_i^\top\mathbf{w} + \frac{n}{\mu}\Delta(\mathbf{y}',\mathbf{y})\right) \\
&= \sum_{b=0}^{n_+}\sum_{c=0}^{n_-}\exp\left(\frac{n}{\mu}\Delta(b,c)\right)\sum_{\mathbf{y}'_+\sim b, y'_k=1}\exp\left(\frac{1}{\mu}\sum_{i=1}^{n_+} y'_i\mathbf{x}_i^\top\mathbf{w}\right) \\
&\qquad \cdot \sum_{\mathbf{y}'_-\sim c}\exp\left(\frac{1}{\mu}\sum_{i=n_++1}^{n} y'_i\mathbf{x}_i^\top\mathbf{w}\right) \\
&= \sum_{b=0}^{n_+}\underbrace{\sum_{\mathbf{y}'_+\sim b, y'_k=1}\exp\left(\frac{1}{\mu}\sum_{i=1}^{n_+} y'_i\mathbf{x}_i^\top\mathbf{w}\right)}_{=:T_+^k(b)} \\
&\qquad \cdot \underbrace{\sum_{c=0}^{n_-}\exp\left(\frac{n}{\mu}\Delta(b,c)\right)V_-(c)}_{=:\eta_-(b)}.
\end{aligned}$$

Since $V_-(c)$ available from forward propagation, $\{\eta_-(b)\}$ can be computed in $O(n^2)$ time. So the only problem left is to compute $T_+^k(b)$. $T_+^k(b)$ has a very intuitive interpretation in Figure 2: the total weight of all paths from $(0,0)$ to (n_+, b) with the k-th step (*i.e.* between the horizontal coordinate $k-1$ and k) going horizontal (not diagonal). Let $\beta_k^b(v)$ denote the total weight of all paths from (k, v) to (n_+, b). Then

$$T_+^k(b) = \sum_{v=0}^{k-1}\alpha_{k-1}(v)c_k\beta_k^b(v).$$

So

$$\begin{aligned}
Z_k &= \sum_{b=0}^{n_+} T_+^k(b)\eta_-(b) = \sum_{b=0}^{n_+}\sum_{v=0}^{k-1}\alpha_{k-1}(v)c_k\beta_k^b(v)\eta_-(b) \\
&= c_k\sum_{v=0}^{k-1}\alpha_{k-1}(v)\underbrace{\sum_{b=0}^{n_+}\beta_k^b(v)\eta_-(b)}_{=:\xi_k(v)}.
\end{aligned}$$

Therefore as long as $\xi_k(v)$ can be updated efficiently, so is Z_k. Fortunately, $\beta_k^b(v)$ has a recursive form

$$\beta_k^b(v) = c_{k+1}\beta_{k+1}^b(v) + \frac{1}{c_{k+1}}\beta_{k+1}^b(v+1),$$

for all $0 \le k \le n_+ - 1$, $0 \le v \le k$ and $0 \le b \le n_+$. This implies for all $0 \le k \le n_+ - 1$ and $0 \le v \le k$

$$\begin{aligned}
\xi_k(v) &= \sum_{b=0}^{n_+}\beta_k^b(v)\eta_-(b) \\
&= \sum_{b=0}^{n_+}[c_{k+1}\beta_{k+1}^b(v) + \frac{1}{c_{k+1}}\beta_{k+1}^b(v+1)]\eta_-(b) \\
&= c_{k+1}\xi_{k+1}(v) + \frac{1}{c_{k+1}}\xi_{k+1}(v+1).
\end{aligned}$$

The final algorithm is summarized in Algorithm 6. Its time and space cost is both $O(n^2)$. The initialization of ξ_k therein is based on initializing $\beta_{n_+}^b(v) = \delta(v = b)$ for all $b, v = 0, 1, \ldots, n_+$.

The gradient of $g^\star(A\mathbf{w})$ for M^{3}Ns can also be computed efficiently by dynamic programming, but the key structure it exploits is the clique decomposition in graphical models. Details can be found in [37].

5.5. Minimizing the model efficiently

In this section, we show that the model ψ_k can be minimized efficiently.

5.5.1. Diagonal quadratic constrained to a box and a hyperplane

When AGM-EF is applied to solve the dual optimization problem $D(\boldsymbol{\alpha})$ for SVM in (41), each iteration needs to solve the model subject to Q_2. This can be reduced to a box constrained diagonal QP with a single linear equality constraint:

$$\begin{aligned} \min \frac{1}{2}\sum_{i=1}^{n} d_i^2(\alpha_i - m_i)^2 \qquad (64)\\ s.t. \quad l_i \le \alpha_i \le u_i \quad \forall i \in [n];\\ \sum_{i=1}^{n} \sigma_i \alpha_i = z. \end{aligned}$$

Similarly, when solving in the primal with smoothing in (56), the gradient query also involves an optimization in this form. In this section, we focus on the following the QP in (64). The algorithm we describe below stems from [38] and finds the exact optimal solution in $O(n)$ time, faster than the $O(n \log n)$ complexity in [39]. [39] also proposes a median finding based algorithm which has linear time complexity *in expectation*. In contrast, our method is deterministic and linear. Liu and Ye [40] tackle this problem too, but they use the mean bisection and apply Newton's method to find a solution up to an inexact prespecified accuracy δ. The resulting total cost is $O(n \log \frac{1}{\delta})$.

Without loss of generality, we assume $l_i < u_i$ and $d_i \neq 0$ for all i. Also assume $\sigma_i \neq 0$ because otherwise α_i can be solved independently. To make the feasible region nonempty, we also assume

$$\begin{aligned} z \ge \sum_i \sigma_i(\delta(\sigma_i > 0) l_i + \delta(\sigma_i < 0) u_i)\\ \text{and} \quad z \le \sum_i \sigma_i(\delta(\sigma_i > 0) u_i + \delta(\sigma_i < 0) l_i). \end{aligned}$$

With a simple change of variable $\beta_i = \sigma_i(\alpha_i - m_i)$, the problem (64) is simplified as

$$\begin{aligned} \min \quad & \frac{1}{2}\sum_{i=1}^{n} \bar{d}_i^2 \beta_i^2\\ s.t. \quad & l'_i \le \beta_i \le u'_i \quad \forall i \in [n];\\ & \sum_{i=1}^{n} \beta_i = z', \end{aligned}$$

where $\bar{d}_i^2 = \frac{d_i^2}{\sigma_i^2}$, $l'_i = \begin{cases} \sigma_i(l_i - m_i) & \text{if } \sigma_i > 0 \\ \sigma_i(u_i - m_i) & \text{if } \sigma_i < 0 \end{cases}$, $u'_i = \begin{cases} \sigma_i(u_i - m_i) & \text{if } \sigma_i > 0 \\ \sigma_i(l_i - m_i) & \text{if } \sigma_i < 0 \end{cases}$, and $z' = z - \sum_i \sigma_i m_i$.

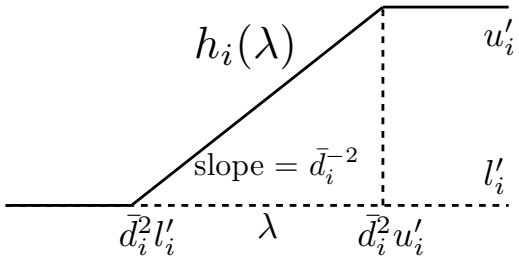


Figure 3. $h_i(\lambda)$

Write out its partial Lagrangian:

$$\min_{\beta_i \in [l'_i, u'_i]} \max_{\lambda \in \mathbb{R}} \sum_{i=1}^{n} \frac{1}{2}\bar{d}_i^2 \beta_i^2 - \lambda\left(\sum_{i=1}^{n} \beta_i - z'\right).$$

Due to strong duality, we can swap the min and max:

$$\begin{aligned} & \max_{\lambda \in \mathbb{R}} \min_{\beta_i \in [l'_i, u'_i]} \sum_{i=1}^{n} \frac{1}{2}\bar{d}_i^2 \beta_i^2 - \lambda\left(\sum_{i=1}^{n} \beta_i - z'\right)\\ &= \max_{\lambda \in \mathbb{R}} \sum_i \min_{\beta_i \in [l'_i, u'_i]} \left(\frac{1}{2}\bar{d}_i^2 \beta_i^2 - \lambda \beta_i\right) + \lambda z'\\ &\Leftrightarrow \min_{\lambda \in \mathbb{R}} \sum_i \underbrace{\max_{\beta_i \in [l'_i, u'_i]} \left(-\frac{1}{2}\bar{d}_i^2 \beta_i^2 + \lambda \beta_i\right)}_{:= H_i(\lambda)} - \lambda z' \qquad (65) \end{aligned}$$

Clearly, the optimal $\beta_i^*(\lambda)$ in the definition of $H_i(\lambda)$ can be solved analytically, and this gives

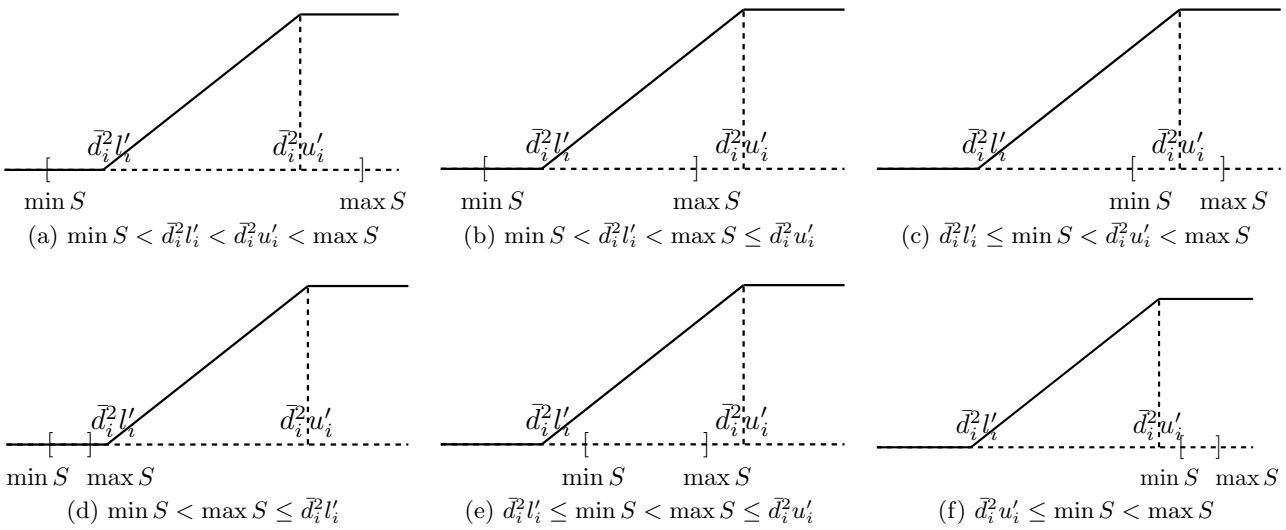
$$H_i(\lambda) = \begin{cases} -\frac{1}{2}\bar{d}_i^2 u_i'^2 + \lambda u'_i & \text{if } \lambda > u'_i \bar{d}_i^2 \\ -\frac{1}{2}\bar{d}_i^2 l_i'^2 + \lambda l'_i & \text{if } \lambda < l'_i \bar{d}_i^2 \\ \frac{\lambda^2}{2\bar{d}_i^2} & \text{if } \lambda \in [l'_i \bar{d}_i^2, u'_i \bar{d}_i^2] \end{cases}$$

$$\text{with} \quad \beta_i^*(\lambda) = \begin{cases} u'_i & \text{if } \lambda > u'_i \bar{d}_i^2 \\ l'_i & \text{if } \lambda < l'_i \bar{d}_i^2 \\ \frac{\lambda}{2\bar{d}_i^2} & \text{if } \lambda \in [l'_i \bar{d}_i^2, u'_i \bar{d}_i^2] \end{cases}.$$

To minimize the objective in (65) as a function of λ, we notice that $H_i(\lambda)$ is convex and differentiable. Thus, the minimizer of (65) is exactly the root of its gradient. Note the gradient of H_i:

$$h_i(\lambda) = \begin{cases} u'_i & \text{if } \lambda > u'_i \bar{d}_i^2 \\ l'_i & \text{if } \lambda < l'_i \bar{d}_i^2 \\ \frac{\lambda}{\bar{d}_i^2} & \text{if } \lambda \in [l'_i \bar{d}_i^2, u'_i \bar{d}_i^2] \end{cases}.$$

See Figure 3 for the plot of $h_i(\lambda)$. So we need to find



(a) $\min S < \bar{d}_i^2 l_i' < \bar{d}_i^2 u_i' < \max S$ (b) $\min S < \bar{d}_i^2 l_i' < \max S \le \bar{d}_i^2 u_i'$ (c) $\bar{d}_i^2 l_i' \le \min S < \bar{d}_i^2 u_i' < \max S$

(d) $\min S < \max S \le \bar{d}_i^2 l_i'$ (e) $\bar{d}_i^2 l_i' \le \min S < \max S \le \bar{d}_i^2 u_i'$ (f) $\bar{d}_i^2 u_i' \le \min S < \max S$

Figure 4. All possible locations of $\min S$ and $\max S$ on $h_i(\lambda)$.

the root of the gradient of (65):

$$f(\lambda) := \sum_{i=1}^{n} h_i(\lambda) - z' = 0. \tag{66}$$

Note that $h_i(\lambda)$ is a monotonically increasing function of λ, so the whole $f(\lambda)$ is monotonically increasing in λ. Since $f(\infty) \ge 0$ by $z' \le \sum_i u_i'$ and $f(-\infty) \le 0$ by $z' \ge \sum_i l_i'$, the root must exist. Considering that f has at most $2n$ kinks (nonsmooth points) and is linear between two adjacent kinks, the simplest idea is to sort $\left\{\bar{d}_i^2 l_i', \bar{d}_i^2 u_i' : i \in [n]\right\}$ into $s^{(1)} \le \ldots \le s^{(2n)}$. If $f(s^{(i)})$ and $f(s^{(i+1)})$ have different signs, then the root must lie between them and can be easily found because f is linear in $[s^{(i)}, s^{(i+1)}]$. This algorithm takes at least $O(n \log n)$ time because of sorting.

However, this cost can be reduced to $O(n)$ by making use of the fact that the median of n (unsorted) elements can be found in $O(n)$ time. Notice that due to the monotonicity of f, f evaluated at the median of a set S is exactly the median of function values, *i.e.*, $f(\text{MED}(S)) = \text{MED}(\{f(x) : x \in S\})$. Algorithm 7 shows the binary search. Let $|S|$ denote the cardinality of S. The while loop must terminate in order $\log_2(2n)$ iterations because in each iteration the cardinality of set S is reduced to at most $\frac{|S|}{2} + 1$ (we will call it "almost halves"). So if $f(m)$ can be evaluated in $O(|S|)$ time, then the time complexity of each iteration is linear in $|S|$, and the total complexity of Algorithm 7 is $O(n)$. Step 7 and 9 ensure that $|S| = 2$ at step 12.

The evaluation of $f(m)$ potentially involves summing up n terms as in (66). However by carefully aggregating the slope and offset, this can be reduced to $O(|S|)$ too. In more detail, let us first consider all the possible locations of $\min S$ and $\max S$ on $h_i(\lambda)$ as illustrated in Figure 4. By halving the set S, the possible transfers of situation are shown in Figure 5. Once the set S gets into the states $(d), (e), (f)$, its state will never change with the shrinking of S, and the contribution of $h_i(\lambda)$ to $f(\lambda)$ will be determined by: l_i' for case (d), u_i' for case (f) and $\frac{\lambda}{\bar{d}_i^2}$ for case (e). So we keep two buffers: $c_g \in \mathbb{R}$ which aggregates the contribution by all the h_i ending in state (d) or (f), and $s_g \in \mathbb{R}$ which aggregates the slope $\frac{1}{\bar{d}_i^2}$ for all h_i ending in state (e). In other words, to evaluate $f(m)$ we only need to visit those h_i which are still in state (a), (b) and (c) (called undetermined states). But how many such i

Algorithm 7 $O(n)$ algorithm to find the root of $f(\lambda)$. Do not allow duplicate points in S.

1: Initialize kink set $S \leftarrow \left\{\bar{d}_i^2 l_i', \bar{d}_i^2 u_i' : i \in [n]\right\}$. Remove duplicates if any.
2: **while** $|S| > 2$ **do**
3: Find the median of S: $m \leftarrow \text{MED}(S)$
4: **if** $f(m) = 0$ **then**
5: **Return** m.
6: **else if** $f(m) > 0$ **then**
7: $S \leftarrow \{x \in S : x \le m\}$.
8: **else**
9: $S \leftarrow \{x \in S : x \ge m\}$.
10: **end if**
11: **end while**
12: **Return** $\frac{lf(u) - uf(l)}{f(u) - f(l)}$ if $S = \{l, u : f(l) \ne f(u)\}$, or any value in $[l, u]$ if $S = \{l < u : f(l) = f(u)\}$.

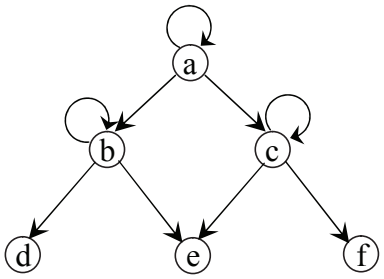


Figure 5. All possible transitions of state.

can there be? By Figure 4, these h_i all contribute at least one kink point in S (state (a) contributes two). If $\left\{\bar{d}_i^2 l_i', \bar{d}_i^2 u_i' : i \in [n]\right\}$ are distinct, then the points in S has one-to-one correspondence to the kink points of h_i. Therefore, the number of h_i in undetermined states must be upper bounded by the size of S. Since the size of S almost halves in each iteration, so is number of h_i in undetermined states. As a result, the cost for computing $f(m)$ halves too. Overall, running Algorithm 7 to completion, the total time spent on evaluating $f(m)$ in step 4 is $O(n)$.

The analysis becomes a bit more complicated when $\left\{\bar{d}_i^2 l_i', \bar{d}_i^2 u_i' : i \in [n]\right\}$ contains duplicate points. In this case, one point in S may correspond to kink points of *multiple* h_i, and so the above argument can no longer be used to upper bound the number of h_i in undetermined states. The simplest patch is to add small perturbations to the duplicate points and make them different. A more principled solution is given in Algorithm 8. The key idea is to allow duplicates in S, and replace $S \leftarrow \{x \in S : x \le m\}$ in step 7 of Algorithm 7 by $S \leftarrow \{x \in S : x < m\}$ (and similarly step 9). An additional level of if-then-else check is introduced so as not to miss out the solution. Clearly, the size of S still halves in Algorithm 8. More importantly, because we do allow the duplicates in S, so the size of S is an upper bound of the number of h_i which is in undetermined states. Therefore, the cost for computing $f(m)$ and $f(y)$ halves through iterations, and the total time spent on evaluating $f(m)$ and $f(y)$ is $O(n)$.

Note that the duplication removal in Algorithm 8 actually cannot be done in $O(n)$ time, and is subject to numerical precision. In our experiment, we used Algorithm 8 which does not remove duplicates. The correctness is easy to prove, and in practice there is almost no duplicates and it works very well.

Algorithm 8 $O(n)$ algorithm to find the root of $f(\lambda)$. Allow duplicate kink points in S.

1: Initialize kink set $S \leftarrow \left\{\bar{d}_i^2 l_i', \bar{d}_i^2 u_i' : i \in [n]\right\}$. Keep duplications and so $|S| = 2n$.
2: **while** $|S| > 2$ **do**
3: Find the median of S: $m \leftarrow \text{MED}(S)$.
4: **if** $f(m) = 0$ **then**
5: **Return** m.
6: **else if** $f(m) > 0$ **then**
7: Find $y := \max\{x \in S : x < m\}$.
// $\{x \in S : x < m\}$ must be nonempty.
8: **if** $f(y) > 0$ **then**
9: $S \leftarrow \{x \in S : x < m\}$.
10: **else**
11: $S \leftarrow \{y, m\}$. // Root lies in $[y, m]$, so exit the while loop immediately.
12: **end if**
13: **else**
14: Find $y := \min\{x \in S : x > m\}$.
// $\{x \in S : x > m\}$ must be nonempty.
15: **if** $f(y) < 0$ **then**
16: $S \leftarrow \{x \in S : x > m\}$.
17: **else**
18: $S \leftarrow \{m, y\}$. // Root lies in $[m, y]$, so exit the while loop immediately.
19: **end if**
20: **end if**
21: **end while**
22: **Return** $\frac{lf(u)-uf(l)}{f(u)-f(l)}$ if $S = \{l, u : f(l) \neq f(u)\}$, or any value in $[l, u]$ if $S = \{l < u : f(l) = f(u)\}$.

5.5.2. Elastic net

For the first type of elastic net (52), the composite optimization is easy thanks to the separability. The second type which uses constraints is much more challenging, and we show in this section how to solve this constrained optimization in linear time. Our approach is similar to the previous Section 5.5.1.

At each iteration of AGM-EF-∞ or AGM-EF-1, we need to solve

$$\min_{\mathbf{w}} \lambda \left(\gamma \|\mathbf{w}\|_1 + \frac{1}{2}\|\mathbf{w}\|_2^2\right) + \frac{L}{2}\|\mathbf{w} - \mathbf{g}_i\|^2.$$

Since all dimensions of $\mathbf{w}$ are decoupled, each w_i can be solved separately as a one dimensional optimization problem. In fact, its solution enjoys a simple closed form [41, p. 384]:

$$\mathbf{w}_i^* = \begin{cases} \frac{Lg_i - \gamma\lambda}{\lambda + L} & \text{if } \lambda < Lg_i/\gamma \\ 0 & \text{if } \lambda \ge Lg_i/\gamma \end{cases}.$$

A more difficult version of elastic net is based on constraints, where in each iteration one needs to solve

$$\min_{\mathbf{w}} \quad \frac{1}{2}\|\mathbf{w}-\mathbf{g}\|^2 \tag{67}$$

$$s.t. \quad \gamma\|\mathbf{w}\|_1 + \frac{1}{2}\|\mathbf{w}\|_2^2 \le \lambda. \tag{68}$$

Clearly the optimal w_i has the same sign as g_i, hence we can assume $g_i \ge 0$ without loss of generality. Next we follow the same idea as in Section 5.5.1 and reformulate (67) into a one dimensional root finding problem. First write out the Lagrangian:

$$\min_{\mathbf{w}}\max_{\lambda\ge 0} \frac{1}{2}\|\mathbf{w}-\mathbf{g}\|^2 + \lambda\left(\gamma\|\mathbf{w}\|_1 + \frac{1}{2}\|\mathbf{w}\|_2^2 - r\right)$$

$$\Leftrightarrow \max_{\lambda\ge 0}\min_{\mathbf{w}} \underbrace{\frac{1}{2}\|\mathbf{w}-\mathbf{g}\|^2 + \lambda\left(\gamma\|\mathbf{w}\|_1 + \frac{1}{2}\|\mathbf{w}\|_2^2 - r\right)}_{=:f_\lambda(\mathbf{w})}$$

where the equivalence is based on a simple check of Slater's condition. For each fixed λ, the optimal $\mathbf{w}$ can be found by setting the subgradient to $\mathbf{0}$.

$$\frac{\partial}{\partial w_i} f_\lambda(\mathbf{w}) = w_i - g_i + \lambda w_i + \lambda\gamma\cdot\begin{cases}1 & \text{if } w_i > 0\\ -1 & \text{if } w_i < 0\\ [-1,1] & \text{if } w_i = 0\end{cases}.$$

Therefore, the optimal solution is

$$w_i^* = \begin{cases}\frac{g_i-\lambda\gamma}{1+\lambda} & \text{if } \lambda \le g_i/\gamma\\ 0 & \text{if } \lambda > g_i/\gamma\end{cases}. \tag{69}$$

Plugging it back to $f_\lambda(\mathbf{w})$ we get the one dimensional optimization problem in λ:

$$H(\lambda) = -r\lambda\gamma + \sum_{i=1}^p \begin{cases}\frac{\lambda(-\lambda\gamma^2+g_i^2+2g_i\gamma)}{2(1+\lambda)} & \text{if } \lambda \le g_i/\gamma\\ \frac{g_i^2}{2} & \text{if } \lambda > g_i/\gamma\end{cases}.$$

It is easy to see that $H(\lambda)$ is concave in $[-1,\infty)$. So its maximizer is 0 or the root of its derivative.

$$H'(\lambda) = -r\gamma + \sum_{i=1}^p \begin{cases}\frac{-\gamma^2\lambda^2-2\gamma^2\lambda+2g_i\lambda+g_i^2}{2(\lambda+1)^2} & \text{if } \lambda \le g_i/\gamma\\ 0 & \text{if } \lambda > g_i/\gamma\end{cases}$$

$$= \frac{1}{2(\lambda+1)^2}h(\lambda)$$

where

$$h(\lambda) = -2\gamma r(\lambda+1)^2 + \sum_{i=1}^p \begin{cases}-\gamma^2(\lambda+1)^2 + (\gamma+g_i)^2 & \text{if } \lambda \le g_i/\gamma\\ 0 & \text{if } \lambda > g_i/\gamma\end{cases}.$$

Clearly $h(\lambda)$ is monotonically decreasing in $[0,\infty)$. So $H(\lambda)$ is maximized at 0 if $h(0) \le 0$, *i.e.*

$$r \ge -\frac{p}{2}\gamma + \frac{1}{2\gamma}\sum_{i=1}^p(\gamma+g_i)^2.$$

Otherwise, $h(\lambda)$ has a root in $[0,\infty)$. Since it monotonically decreases, the binary search trick in Section 5.5.1 can also be applied here. Once it is determined that the optimal λ is less than a set of g_i, these quadratics can be aggregated by summing up the $\gamma\left(g_i+\frac{1}{\gamma}\right)^2$. Finally, $\mathbf{w}$ is recovered by (69).

5.6. Optimizing the Prox-function

When smoothing $g^\star$, we have often used prox-function $d(\mathbf{x}) = \sum_i x_i^2$. However, it is possible to improve the condition number by using an optimized prox-function. This idea was used by [17] where the *l.c.g* constant of a quadratic $\frac{1}{2}\mathbf{x}^\top H\mathbf{x} + \langle\mathbf{b},\mathbf{x}\rangle$ ($\mathbf{x}\in\mathbb{R}^p$) is upper bounded by p when the norm is chosen as $\|\mathbf{x}\|^2 = \sum_i H_{ii}x_i^2$, *i.e.* rescaling all dimensions.

Using this idea, we show in this section that a data dependent optimization of the prox-function can improve the condition number of the smoothed variant of the primal objective as discussed in Section 5.1.

Let us consider the following simple but illustrative example. Suppose $Q_2 = [0,c]^n$, $g(\mathbf{u}) = -\sum_{i=1}^n u_i$. Denote $A = (\mathbf{a}_1,\ldots,\mathbf{a}_n)^\top$. We adopt a prox-function

$$d(\mathbf{u}) = \frac{1}{2}\sum_{i=1}^n b_i^2u_i^2,$$

and we can derive $g_\mu^\star = (g+\mu d)^\star$. The diameter of Q_2 under d is

$$D = \max_{\mathbf{x}\in Q_2} d(\mathbf{u}) = \frac{c^2}{2}\sum_i b_i^2. \tag{70}$$

For any prescribed accuracy $\epsilon > 0$, we first choose μ such that $\mu D \le \epsilon$, *i.e.* $\mu \le \frac{\epsilon}{D}$. Then our goal is to find the b_i which minimizes the Lipschitz constant of the gradient of $g_\mu^\star(A\mathbf{w})$ wrt $\mathbf{w}$.

First compute $g_\mu^\star(A\mathbf{w})$:

$$g_\mu^\star(A\mathbf{w}) = \sup_{\mathbf{u}\in Q_2}\langle A\mathbf{w},\mathbf{u}\rangle + \sum_i u_i - \frac{\mu}{2}\sum_i b_i^2u_i^2.$$

It is easy to see that the optimal $\mathbf{u}^*$ is

$$u_i^* = \mathrm{MED}\left(0, c, \frac{\langle\mathbf{a}_i,\mathbf{w}\rangle+1}{\mu b_i^2}\right).$$

where MED stands for the median. So the gradient of $g_\mu^\star(A\mathbf{w})$ wrt $\mathbf{w}$ can be calculated by

$$\frac{\partial}{\partial \mathbf{w}} g_\mu^\star(A\mathbf{w}) = \sum_{i=1}^n \mathbf{g}_i, \text{ where } \mathbf{g}_i = \begin{cases} 0 & \text{if } u_i^* = 0 \\ \mathbf{a}_i & \text{if } u_i^* = c \\ \frac{\langle \mathbf{a}_i, \mathbf{w}\rangle + 1}{\mu b_i^2}\mathbf{a}_i & \text{else} \end{cases}.$$

So the Hessian of $g_\mu^\star(A\mathbf{w})$ in $\mathbf{w}$ can only take value in

$$H_\delta = \frac{1}{\mu}\sum_i \frac{\delta_i}{b_i^2}\mathbf{a}_i\mathbf{a}_i^\top, \quad \text{where} \quad \delta_i \in \{0,1\}.$$

Now for any $\mathbf{w}_1, \mathbf{w}_2 \in Q_1$, denote $l = \|\mathbf{w}_1 - \mathbf{w}_2\|$ and $\mathbf{v} = (\mathbf{w}_2 - \mathbf{w}_1)/l$ (so $\|\mathbf{v}\| = 1$). Denote $\mathbf{h}(t) = \frac{\partial}{\partial \mathbf{w}} g_\mu^\star(A(\mathbf{w}_1 + t\mathbf{v}))$. So

$$\frac{\left\|\frac{\partial}{\partial \mathbf{w}} g_\mu^\star(A\mathbf{w}_1) - \frac{\partial}{\partial \mathbf{w}} g_\mu^\star(A\mathbf{w}_2)\right\|}{\|\mathbf{w}_1 - \mathbf{w}_2\|} = \frac{\|\mathbf{h}(l) - \mathbf{h}(0)\|}{l}$$
$$\overset{(a)}{\leq} \|\nabla \mathbf{h}(\xi)\| \overset{(b)}{=} \|H_\delta \mathbf{v}\| \overset{(c)}{\leq} \lambda_{\max}(H_\delta).$$

Here, (a) is by the mean value theorem with $\xi \in [0, l]$. (b) is by the chain rule and the δ for H_δ is determined by ξ. (c) is because for any real positive semi-definite matrix H, $\max_{\|\mathbf{v}\|=1} \|H\mathbf{v}\| = \lambda_{\max}(H)$.

Clearly $\lambda_{\max}(H_\delta)$ is maximized when all $\delta_i = 1$ and let us call it $H_{\mathbf{1}}$. In conjunction with (70) and (57), we minimize $\lambda_{\max}(H_{\mathbf{1}})$ wrt b_i:

$$\min_{b_i} \lambda_{\max}(H_{\mathbf{1}}) = \min_{b_i} \frac{D}{\epsilon}\lambda_{\max}\left(\sum_i \frac{1}{b_i^2}\mathbf{a}_i\mathbf{a}_i^\top\right)$$
$$= \frac{c^2}{2\epsilon}\min_{b_i}\left\{\left(\sum_i b_i^2\right)\max_{\mathbf{v}:\|\mathbf{v}\|=1} \mathbf{v}^\top \sum_i b_i^{-2}\mathbf{a}_i\mathbf{a}_i^\top \mathbf{v}\right\}$$
$$= \frac{c^2}{2\epsilon}\max_{\|\mathbf{v}\|=1}\min_{b_i}\left(\sum_i b_i^2\right)\sum_i b_i^{-2}(\mathbf{a}_i^\top \mathbf{v})^2 \quad (71)$$
$$= \frac{c^2}{2\epsilon}\max_{\|\mathbf{v}\|=1}\left(\sum_i \left|\mathbf{a}_i^\top \mathbf{v}\right|\right)^2 \text{ (Cauchy-Schwartz)} \quad (72)$$
$$\Leftrightarrow \max_{\|\mathbf{v}\|=1}\sum_i \left|\mathbf{a}_i^\top \mathbf{v}\right|.$$

However, this last optimization problem is hard so we maximize an approximation of it

$$\max_{\|\mathbf{v}\|=1}\sum_i \left|\mathbf{a}_i^\top \mathbf{v}\right|^2 = \max_{\|\mathbf{v}\|=1} \mathbf{v}^\top\left(\sum_i \mathbf{a}_i\mathbf{a}_i^\top\right)\mathbf{v}.$$

The solution is the eigenvector $\mathbf{v}^*$ corresponding to the maximum eigenvalue of $\sum_i \mathbf{a}_i\mathbf{a}_i^\top$. Then b_i^2 can be recovered by using the optimality condition of Cauchy-Schwartz in (72):

$$b_i^2 = \left|\mathbf{a}_i^\top \mathbf{v}^*\right|.$$

Note [42] used the heuristic that $b_i^2 = \|\mathbf{a}_i\|_\infty$. We can also compare with the isotropic d, *i.e.* $b_i = 1$. Simply plug $b_i = 1$ into (71), and we get

$$\frac{c^2}{2\epsilon} n \sum_i (\mathbf{a}_i^\top \mathbf{v})^2$$

which must be greater than or equal to

$$\frac{c^2}{2\epsilon}\left(\sum_i \left|\mathbf{a}_i^\top \mathbf{v}\right|\right)^2$$

in (72) for all $\mathbf{v}$. Therefore with a fixed ϵ, our approach does possibly reduce the *l.c.g* constant of $g^\star(A\mathbf{w})$ in $\mathbf{w}$. The maximum eigenvector can be found very efficiently by using the power iteration, and usually 5 to 6 iterations is enough.

6. Experimental Results

We will present the experimental result in a later version.

7. Discussion

A lot of efforts (*e.g.*, [43, 44]) have been devoted to making Nesterov's method online, *i.e.* use a stochastic gradient oracle and preserve the $1/\sqrt{\epsilon}$ rate of convergence for the expected gap. This however turns out hopeless as was shown by the lower bounds in [45, 46].

References

[1] Xinhua Zhang, Ankan Saha, and S.V.N. Vishwanathan. Lower bounds on rate of convergence of cutting plane methods. In *Advances in Neural Information Processing Systems 23*, 2011.

[2] Zhaosong Lu. Smooth optimization approach for sparse covariance selection. *SIAM Journal on Optimization*, 19(4):1807–1827, 2009.

[3] Jun Liu, Jianhui Chen, and Jieping Ye. Large-scale sparse logistic regression. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2009.

[4] Andrew Gilpin, Tuomas Sandholm, and Troels Bjerre Sorensen. A heads-up no-limit texas hold'em poker player: Discretized betting models and automatically generated equilibrium-finding programs. In *International Joint Conference on Autonomous Agents and Multiagent Systems*, 2008.

[5] Xinhua Zhang, Ankan Saha, and S.V.N. Vishwanathan. Lower bounds for BMRM and faster rates for training SVMs. Technical report arXiv:0909.1334, 2009. URL http://arxiv.org/abs/0909.1334.

[6] Yurii Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1):127–152, 2005.

[7] John C. Platt. Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods — Support Vector Learning*, pages 185–208. MIT Press, 1999.

[8] Peter L. Bartlett, Michael I. Jordan, and Jon D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

[9] Paul Tseng. On accelerated proximal gradient methods for convex-concave optimization. *submitted to SIAM Journal on Optimization*, 2009.

[10] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[11] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of Royal Statistics Society. B*, 67(2):301–320, 2005.

[12] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, 2010.

[13] Manfred K. Warmuth, Karen A. Glocer, and S. V. N. Vishwanathan. Entropy regularized LPBoost. In Yoav Freund, Yoav Làszlò Györfi, and György Turàn, editors, *Proc. Intl. Conf. Algorithmic Learning Theory*, number 5254 in Lecture Notes in Artificial Intelligence, pages 256 – 271, Budapest, October 2008. Springer-Verlag.

[14] Alexandre d'Aspremont, Onureena Banerjee, and Laurent El Ghaoui. First-order methods for sparse covariance selection. *SIAM Journal on Matrix Analysis and Applications*, 30(1):56–66, 2008.

[15] Prateek Jain, Brian Kulis, Inderjit S. Dhillon, and Kristen Grauman. Online metric learning and fast similarity search. In *Advances in Neural Information Processing Systems*, 2009.

[16] Brian Kulis and Peter L Bartlett. Implicit online learning. In *Proc. Intl. Conf. Machine Learning*, 2010.

[17] Yurii Nesterov. Gradient methods for minimizing composite objective function. Technical Report 76, CORE Discussion Paper, UCL, 2007.

[18] Yurii Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. *Soviet Math. Docl.*, 269:543–547, 1983.

[19] Yurii Nesterov. *Introductory Lectures On Convex Optimization: A Basic Course*. Springer, 2003.

[20] Yurii Nesterov. Excessive gap technique in non-smooth convex minimization. *SIAM J. on Optimization*, 16(1):235–249, 2005. ISSN 1052-6234.

[21] Arkadi Nemirovski. Efficient methods in convex programming. Lecture notes, 1994.

[22] Ting Kei Pong, Paul Tseng, Shuiwang Ji, and Jieping Ye. Trace norm regularization: Reformulations, algorithms, and multi-task learning. *SIAM Journal on Optimization*, 2010.

[23] Alfred Auslender and Marc Teboulle. Interior gradient and proximal methods for convex and conic optimization. *SIAM Journal on Optimization*, 16 (3):697–725, 2006.

[24] Shai Shalev-Shwartz. *Online Learning: Theory, Algorithms, and Applications*. PhD thesis, The Hebrew University of Jerusalem, July 2007.

[25] Choon Hui Teo, S. V. N. Vishwanthan, Alex J. Smola, and Quoc V. Le. Bundle methods for regularized risk minimization. *J. Mach. Learn. Res.*, 11:311–365, January 2010.

[26] Guanghui Lan, Zhaosong Lu, and Renato D. C. Monteiro. Primal-dual first-order methods with $\mathcal{O}(1/\epsilon)$ iteration complexity for cone programming. *Mathematical Programming*, 2009.

[27] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 1995.

[28] Alfred Auslender and Marc Teboulle. Interior gradient and epsilon-subgradient descent methods for constrained convex minimization. *Mathematics of Operations Research*, 29(1):1–26, 2004.

[29] J. M. Borwein and A. S. Lewis. *Convex Analysis and Nonlinear Optimization: Theory and Examples*. CMS books in Mathematics. Canadian Mathematical Society, 2000.

[30] K. P. Bennett and O. L. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optim. Methods Softw.*, 1:23–34, 1992.

[31] T. Joachims. A support vector method for multivariate performance measures. In *Proc. Intl. Conf. Machine Learning*, pages 377–384, San Francisco, California, 2005. Morgan Kaufmann Publishers.

[32] J. D. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic modeling for segmenting and labeling sequence data. In *Proceedings of International Conference on Machine Learning*, volume 18, pages 282–289, San Francisco, CA, 2001. Morgan Kaufmann.

[33] B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 25–32, Cambridge, MA, 2004. MIT Press.

[34] Shuiwang Ji and Jieping Ye. An accelerated gradient method for trace norm minimization. In *Proc. Intl. Conf. Machine Learning*, 2009.

[35] C. Do, Q. Le, and C.S. Foo. Proximal regularization for online and batch learning. In *International Conference on Machine Learning ICML*, 2009.

[36] Olivier Chapelle. Training a support vector machine in the primal. *Neural Computation*, 19(5): 1155–1178, 2007.

[37] Xinhua Zhang, Ankan Saha, and S.V.N. Vishwanathan. Faster rates for training max-margin markov networks. Technical report arXiv:1003.1354, 2010. URL http://arxiv.org/abs/1003.1354.

[38] P. M. Pardalos and N. Kovoor. An algorithm for singly constrained class of quadratic programs subject to upper and lower bounds. *Mathematical Programming*, 46:321–328, 1990.

[39] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandrae. Efficient projections onto the ℓ_1-ball for learning in high dimensions. In *Proc. Intl. Conf. Machine Learning*, 2008.

[40] Jun Liu and Jieping Ye. Efficient euclidean projections in linear time. In *Proc. Intl. Conf. Machine Learning*. Morgan Kaufmann, 2009.

[41] G. B. Passty. Ergodic converence to a zero of the sum of monotone operators in Hilberts space. *Journal of Optimization Theory and Applications*, 72:383–390, 1979.

[42] Tianyi Zhou, Dacheng Tao, and Xindong Wu. NESVM: a fast gradient method for support vector machines. In *Proc. Intl. Conf. Data Mining*, 2010.

[43] Chonghai Hu, James T. Kwok, and Weike Pan. Accelerated gradient methods for stochastic optimization and online learning. In *Neural Information Processing Systems*, 2009.

[44] Lin Xiao. Dual averaging methods for regularized stochastic learning and online optimization. Technical Report MSR-TR-2010-23, Microsoft Research, 2010.

[45] Guanghui Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 2010.

[46] Saeed Ghadimi and Guanghui Lan. "optimal stochastic approximation algorithms for strongly convex stochastic composite optimization. *Submitted*, 2010.

[47] J.B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms, I and II*, volume 305 and 306. Springer-Verlag, 1993.

A. Concepts from Convex Analysis

The following four concepts from convex analysis are used in the paper.

Definition 2. *Suppose a convex function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is finite at $\mathbf{w}$. Then a vector $\mathbf{g} \in \mathbb{R}^n$ is called a subgradient of f at $\mathbf{w}$ if, and only if,*

$$f(\mathbf{w}') \geq f(\mathbf{w}) + \langle \mathbf{w}' - \mathbf{w}, \mathbf{g} \rangle \qquad \text{for all } \mathbf{w}'.$$

The set of all such $\mathbf{g}$ vectors is called the subdifferential of f at $\mathbf{w}$, denoted by $\partial_{\mathbf{w}} f(\mathbf{w})$. For any convex function f, $\partial_{\mathbf{w}} f(\mathbf{w})$ must be nonempty. Furthermore if it is a singleton then f is said to be differentiable *at $\mathbf{w}$, and we use $\nabla f(\mathbf{w})$ to denote the gradient.*

Definition 3. *A convex function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is strongly convex with respect to a norm $\|\cdot\|$ if there exists a constant $\sigma > 0$ such that $f - \frac{\sigma}{2}\|\cdot\|^2$ is convex. σ is called the modulus of strong convexity of f, and for brevity we will call f σ-strongly convex.*

Definition 4. *Suppose a function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is differentiable on $Q \subseteq \mathbb{R}^n$. Then f is said to have Lipschitz continuous gradient (l.c.g) with respect to a norm $\|\cdot\|$ if there exists a constant L such that*

$$\|\nabla f(\mathbf{w}) - \nabla f(\mathbf{w}')\|^* \leq L \|\mathbf{w} - \mathbf{w}'\| \quad \forall\ \mathbf{w}, \mathbf{w}' \in Q.$$

For brevity, we will call f L-l.c.g.

Definition 5. *The Fenchel dual of a function* $f : \mathbb{R}^n \to \overline{\mathbb{R}}$*, is a function* $f^\star : \mathbb{R}^n \to \overline{\mathbb{R}}$ *defined by*

$$f^\star(\mathbf{w}^\star) = \sup_{\mathbf{w} \in \mathbb{R}^n} \{\langle \mathbf{w}, \mathbf{w}^\star \rangle - f(\mathbf{w})\}$$

Strong convexity and *l.c.g* are related by Fenchel duality according to the following lemma:

Theorem 21 ([47, Theorem 4.2.1 and 4.2.2])**.**

1. *If* $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ *is* σ*-strongly convex, then* $f^\star$ *is finite on* $\mathbb{R}^n$ *and* $f^\star$ *is* $\frac{1}{\sigma}$*-l.c.g.*
2. *If* $f : \mathbb{R}^n \to \mathbb{R}$ *is convex, differentiable on* $\mathbb{R}^n$*, and* L*-l.c.g, then* $f^\star$ *is* $\frac{1}{L}$*-strongly convex.*

Finally, the following lemma gives a useful characterization of the minimizer of a convex function.

Lemma 22 ([47, Theorem 2.2.1])**.** *A convex function* f *is minimized at* $\mathbf{w}^*$ *if, and only if,* $\mathbf{0} \in \partial f(\mathbf{w}^*)$*. Furthermore, if* f *is strongly convex, then its minimizer is unique.*