Regularization and Randomization of a Convex Optimization Problem

Aymen MERROUCHE

March 13, 2021

Contents

Let $(f, \mathcal{K})$ be a CO. Based on the rate guarantees of GD, we know that the more regular is f, the easier the CO $(f, \mathcal{K})$. In the following, we will look at two methods of smoothing the cost function f.

1 Regularization

Definition 1. *Regularizing the CO problem $(f, \mathcal{K})$ consists in adjoining a regularization function R strongly convex on K and twice continuously differentiable so that $(f + R, \mathcal{K})$ becomes an easier CO problem.*

Example. for example a regularization of the CO $(f, \mathcal{K})$, where f is convex would be $(g, \mathcal{K})$, where $g(x) = f(x) + \alpha\ 2\left\|x - x^1\right\|^2$. The objective function g is now α-strongly convex, the problem is easier and the corresponding GD error $h_g^T = g(x_T) - g(x^*)$ is smaller, and we have :

$$\begin{aligned} f(x_T) - f(x^*) &\leq g(x_T) - g(x^*) + \alpha/2(\|x^* - x_1\| - \|x_T - x_1\|) \\ &\leq \underbrace{g(x_T) - g(x^*) + \alpha/2D^2}_{D \text{ is the diameter}} \leq \underbrace{g(x_T) - g(x_g^*) + \alpha/2D^2}_{g(x_g^*) \leq g(x^*)} \\ &\leq h_T^g + \alpha/2D^2 \end{aligned}$$

Given the rate guarantees of GD, a direct result is the following example :

Example. If f convex, g is α-strongly convex, $h_T^g = O(1/(\alpha T))$ with fixing $\alpha = 1/\sqrt{T}$. Also, if f β-smooth, then g is γ well conditioned with $\gamma = \alpha/(\beta + \alpha)$, $h_T^g = O(e^{-\gamma T})$ with fixing $\alpha = \beta \log(T)/T$.

2 Randomization

Definition 2. *Randomization is done by the introduction of a sample scheme in the optimization problem. Instead of f, a randomized version $\hat{f}_\delta$ is used, given by $\hat{f}_\delta(x) = \mathbb{E}_{U \sim \mathcal{U}}[f(x + \delta U)]$, where $\mathcal{U}(B_1)$ is the uniform distribution on the unit euclidean ball.*

Proposition 3. *$\hat{f}_\delta$ is dG/δ-smooth and a δG uniform approximation of f i.e. $|\hat{f}_\delta(x) - f(x)| \leq \delta G$ $\forall x \in \mathcal{K}$. We then have that :*

$$\begin{aligned} f(x_T) - f(x^*) &\leq \hat{f}_\delta(x_T) - \hat{f}_\delta(x^*) + 2\delta G \\ &\leq \underbrace{\hat{f}_\delta(x_T) - \hat{f}_\delta(x^*_{\hat{f}_\delta}) + 2\delta G}_{\hat{f}_\delta(x^*_{\hat{f}_\delta}) \leq \hat{f}_\delta(x^*)} \\ &\leq h_t^{\hat{f}_\delta} + 2\delta G \end{aligned}$$

Remark. $\hat{f}_\delta(x)$ is a approximation of $f(x)$, it corresponds to a local mean of f around x (where δ is small). Randomization yields a β-smooth function when having no guarantees on f. It allows a faster convergence of algorithms. In practice, we estimate $\hat{f}_\delta(x)$ using a Monte Carlo method by $\frac{1}{M}\sum_{i=1}^M f(x + \delta U_i)$ where U_is are sampled accordingly to uniform distribution on the unit euclidean ball, for a fixed value of δ.

Example 1. For f α-strongly convex, the rate of the CO problem is at most $O(dG^2 logT/(\alpha T))$.

Proof. Using the rate guarantees of GD algorithm for a γ-well conditioned functions with a a learning rate $\eta_t = 1/\beta$ with $\beta = dG/\delta$. $\hat{f}_\delta$ is $\alpha\delta/dG$-strongly convex, this follows from f being α-strongly convex ($\hat{f}_\delta$ is hence α-strongly convex) and proposition 3, combining this with the previous inequality and minimizing the resulting upper bound with respect to δ yields the result. Expressing δ also allows to calibrate this hyper-parameter when implementing the algorithm. □

Example 2. We can similarly show that the rate of any CO problem is at most of the order $O(G/(\alpha D)\sqrt{logT/(dT)})$.

3 Supervised Classification

Let's consider a binary supervised classification task where we observe labels $b_i \in \{\pm 1\}$ along with explanatory variables a_i. The goal is then to predict b_i given a_i.

Case of study. MNIST is a database of 28×28 gray scale images representing handwritten digits. We can consider a binary classification task, with two classes : 0 vs other digits . b_i would then be 1 if the digit is zero and -1 if not and a_i would be the corresponding image with $d = 28 \times 28 = 784$.

Definition 4. *(SVM) Linear Support Vector machines are classifiers of the form $sign(x^T a)$, where x is in $\mathbb{R}^d$.*
We aim at finding, using a train set $\{(a_i, b_i)\}_{i=1}^{n}$, a vector $x \in \mathbb{R}^d$, that minimizes the accuracy approximated over a test set $\{(a_i, b_i)\}_{i=n+1}^{n+n'}$:

$$\mathbb{P}(sign(x^T a) \neq b) \approx \sum_{i=n+1}^{n+n'} \mathbb{1}_{sign(x^T a_i) \neq b_i} \tag{1}$$

Hard Margin Problem. In other words, we want to find $x \in \mathbb{R}^d$ such that :

$$\min_{x \in \mathbb{R}^d} \sum_{i=1}^{n} \mathbb{1}_{sign(\mathbf{x}^T . a_i) \neq b_i} \tag{2}$$

This is known as the hard margin problem.
However this problem is non-polynomial and non-convex due to the indicator function which makes it difficult to resolve. We need to relax it to a CO problem.

Definition 5. *(Hinge Loss) The hinge loss is a convex surrogate of the $0-1$ loss ($\mathbb{1}_{sign(x^T a) \neq b} = \mathbb{1}_{sign(a x^T b) < 0}$), it is given by :*

$$\ell_{a,b} = Hinge(b x^T a) = max(0, 1 - b x^T a) \tag{3}$$

Soft Margin Problem. We relax the hard margin problem to the soft margin problem given by :

$$\min_{x \in \mathbb{R}^d} \frac{\lambda}{2} ||x||^2 + \frac{1}{n} \sum_{i=1}^{n} hinge(b_i x^T a_i) \tag{4}$$

Remark. Thanks to regularization, the objective function of the soft margin problem is λ-strongly convex hence the minimum is unique. However, it is not β-smooth, if we want the objective function to have this property we can use randomization. We can now directly apply GD on this CO problem.

4 ℓ_1-ball constrained CO as dual of the lasso

Stochastic Setting. Considering $f(x) = \sum_{i=1}^{n} \ell_i(x)$ where $(\ell_1, .., \ell_n)$ are assumed to be iid convex functions over $\mathbb{R}^d$, the goal is to minimize the risk $\mathbb{E}[\ell(x)]$.

In this setting, we can face generalization problems, namely, over-fitting (over-fitting happens when the model memorizes the training data rather than learning the underlying distribution hence performing poorly on new unseen data). One way of dealing with this problem is to penalize as follows :

Information criteria (AIC, BIC). a penalization of the form :

$$f(x) + \theta \left\| X \right\|_0 = \theta \sum_{i=1}^{n} \mathbb{1}_{x_i \neq 0}$$

where $\theta > 0$

Remark. The previous penalization, adds θ for each coordinate of x which is not equal to 0. This allows to focus only on the important dimensions and introduces sparsity. For our case of study (binary classification on digit images), it is very convenient since only few pixels on the image, which constitute the digit, are relevant for classification. Only those digits should be taken into account, hence x_is which correspond to an irrelevant area, let's say the upper corner, should all be equal to 0. The previous penalization poses a problem since it is non-polynomial and non-convex. It is relaxed using the convex ℓ_1 norm insted of $\|.\|_0$ as stated below :

LASSO. a penalization of the form :

$$f(x) + \theta \|x\|_1$$

where $\theta > 0$
LASSO penalization is equivalent to the following constrained problem :

$$\min_{\|x\|_1 \leq \tau} f(x) \tag{5}$$

i.e. minimizing f in the ℓ_1 ball of radius τ.

Proof. ($\implies$) is straightforward. ($\impliedby$) using Lagrange duality and KKT conditions. □

We can know add a projection step onto ℓ_1 ball to our gradient descent algorithm after each gradient step to introduce sparsity hence placing ourselves in the right dimension of the problem. In the next section section we will derive using the duality of Lagrange explicitly how to project onto the ℓ_1-ball.

5 Explicit projection on the ℓ_1-ball

5.1 Projection On the Simplex

Let's consider the projection on the simplex $\bigwedge = \{w \in \mathbb{R}^d_+; \sum_{i=1}^d w_i = 1\}$

The Lagrangian of the projection problem is then given by :

$$L(w, \theta, \xi) = \frac{1}{2} \|w - x\|^2 + \theta \sum_{i=1}^{d} w_i - 1 - \sum_{i=1}^{d} \xi_i w_i$$

where $w \in \mathbb{R}^d$ and $\theta \in \mathbb{R}$ and $\xi \in \mathbb{R}^d_+$.
Using KKT conditions we get an explicit algorithm for the projection on the simplex.
We can use this algorithm to project onto the ℓ_1 -ball, since $\forall x \in B_1(z)$ we can project $|x|/z$ onto the simplex. Let w^* be this projection, then $z \times sign(x) \times w^*$ yields the projection onto the ℓ_1-ball. The corresponding algorithm is of complexity $O(d \log(d))$.