Mirror Descent part 2: Introduction to the algorithm

Andersen Ang

ECS, Uni. Southampton, UK
andersen.ang@soton.ac.uk

Homepage angms.science

Version: July 19, 2024
First draft: Sept 6, 2020

Content

- Bregman divergence

$$B_\phi(\boldsymbol{x}, \boldsymbol{y}) \;=\; \phi(\boldsymbol{x}) - \phi(\boldsymbol{y}) - \big\langle \nabla\phi(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \big\rangle,$$

- Bregman projection of $\boldsymbol{y}$ onto the same set $\mathcal{X}$, under the function ϕ

$$P_{\mathcal{X},\phi}(\boldsymbol{y}) = \operatorname*{argmin}_{\boldsymbol{x}\in\mathcal{X}} B_\phi(\boldsymbol{x}, \boldsymbol{y}).$$

- Mirror descent

$$\boldsymbol{x}_{k+1} \;=\; \operatorname*{argmin}_{\boldsymbol{x}} f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k \rangle + \frac{1}{\alpha} B_\phi(\boldsymbol{x}, \boldsymbol{x}_k).$$

Read these first

- Gradient lives in the dual space
  https://angms.science/doc/CVX/Grad_is_dual.pdf

- Mirror Descent part 1: Why mirror descent? Why gradient descent is not enough?
  https://angms.science/doc/CVX/MD_motivation.pdf

Bregman divergence

- A key ingredient of mirror descent is the Bregman divergence.

- Given a norm $\|\cdot\|$ (any norm), the Bregman divergence defined on a function ϕ is

$$B_\phi(\boldsymbol{x}, \boldsymbol{y}) \;=\; \phi(\boldsymbol{x}) - \phi(\boldsymbol{y}) - \big\langle \nabla\phi(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \big\rangle,$$

  where ϕ
  - is differentiable with respect to $\boldsymbol{x}$, i.e. $\nabla_{\boldsymbol{x}}\phi(\boldsymbol{x})\big|_{\boldsymbol{x}=\boldsymbol{x}_0}$ exists for all $\boldsymbol{x}_0$
  - is λ-strongly convex w.r.t. the norm $\|\cdot\|$, i.e., $\phi(\boldsymbol{x}) - \dfrac{\lambda}{2}\|\boldsymbol{x}\|$ is convex.

- Based on the definition of strong-convexity,

$$B_\phi(\boldsymbol{x}, \boldsymbol{y}) \;\geq\; \frac{\lambda}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2.$$

- See here for more on B_ϕ and here for more on strongly convex function.

Some key points about Bregman divergence

$$B_\phi(\boldsymbol{x}, \boldsymbol{y}) \;=\; \phi(\boldsymbol{x}) - \phi(\boldsymbol{y}) - \big\langle \nabla\phi(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \big\rangle,$$

- Bregman divergence is not commutative: $B(\boldsymbol{x}, \boldsymbol{y}) \neq B(\boldsymbol{y}, \boldsymbol{x})$.
  Easy to see it from the definition.

- Bregman divergence is nonnegative.
  This is due to the assumption that $\phi(\boldsymbol{x})$ is strongly-convex

- Bregman divergence is like a generalization of the squared-Euclidean distance.

- See here for more on Bregman divergence.

Bregman projection vs Euclidean projection

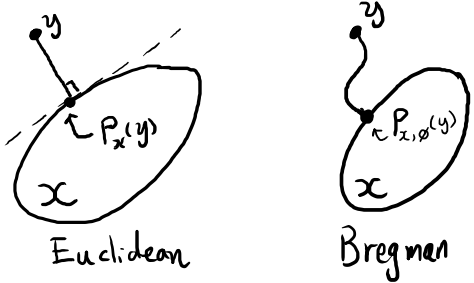
- Given a set $\mathcal{X}$, the Euclidean projection of a point $\boldsymbol{y}$ onto this set is defined as

$$P_{\mathcal{X}}(\boldsymbol{y}) = \underset{\boldsymbol{x}\in\mathcal{X}}{\operatorname{argmin}} \ \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2$$

- The Bregman projection of $\boldsymbol{y}$ onto the same set $\mathcal{X}$, under the function ϕ, is defined as

$$P_{\mathcal{X},\phi}(\boldsymbol{y}) = \underset{\boldsymbol{x}\in\mathcal{X}}{\operatorname{argmin}} \ B_\phi(\boldsymbol{x},\boldsymbol{y}).$$

- Geometrically, Euclidean projection uses Euclidean geometry to give an orthogonal projection, while Bregman projection uses a curved geometry to give the projection.

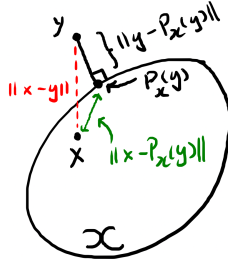


Bregman projection vs Euclidean projection

- For Euclidean projection, for all $\boldsymbol{x} \in \mathcal{X}$, we have

$$\|\boldsymbol{x} - \boldsymbol{y}\|_2^2 \geq \|\boldsymbol{x} - P_{\mathcal{X}}(\boldsymbol{y})\|_2^2 + \|P_{\mathcal{X}}(\boldsymbol{y}) - \boldsymbol{y}\|_2^2.$$

Triangle inequality of Euclidean projection



- Similarly, for Bregman projection,

$$B_\phi(\boldsymbol{x}, \boldsymbol{y}) \geq B_\phi\Big(\boldsymbol{x} - P_{\mathcal{X}}(\boldsymbol{y})\Big) + B_\phi\Big(P_{\mathcal{X}}(\boldsymbol{y}) - \boldsymbol{y}\Big).$$

Re-look at the gradient descent algorithm

- The gradient descent update (with $\alpha > 0$ being the stepsize)

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha \nabla f(\boldsymbol{x}_k), \tag{1}$$

  can be treated as the minimization of a local quadratic model of f at the point $\boldsymbol{x}_k$:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \operatorname*{argmin}_{\boldsymbol{x}} Q(\boldsymbol{x}, \boldsymbol{x}_k) \\ &= \operatorname*{argmin}_{\boldsymbol{x}} f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k \rangle + \frac{1}{2\alpha} \|\boldsymbol{x} - \boldsymbol{x}_k\|_2^2. \end{aligned}$$

  That is, take $\nabla Q(\boldsymbol{x}, \boldsymbol{x}_k) = \boldsymbol{0}$ gives (1).

- See here for more.

The mirror descent algorithm

- Mirror descent = gradient descent with the ℓ_2-norm term $\frac{1}{2}\|\cdot\|_2^2$ replaced by the Bergman distance

$$\begin{aligned}
\boldsymbol{x}_{k+1} &= \operatorname*{argmin}_{\boldsymbol{x}} f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k \rangle + \frac{1}{\alpha} B_\phi(\boldsymbol{x}, \boldsymbol{x}_k) \\
&= \operatorname*{argmin}_{\boldsymbol{x}} f(\boldsymbol{x}_k) + \left\langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k \right\rangle + \frac{\phi(\boldsymbol{x}) - \phi(\boldsymbol{x}_k) - \langle \nabla \phi(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k \rangle}{\alpha} \\
&= \operatorname*{argmin}_{\boldsymbol{x}} \left\langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x} \right\rangle + \frac{\phi(\boldsymbol{x}) - \langle \nabla \phi(\boldsymbol{x}_k), \boldsymbol{x} \rangle}{\alpha} \\
&= \operatorname*{argmin}_{\boldsymbol{x}} \left\langle \nabla f(\boldsymbol{x}_k) - \frac{1}{\alpha} \nabla \phi(\boldsymbol{x}_k), \boldsymbol{x} \right\rangle + \frac{1}{\alpha} \phi(\boldsymbol{x}).
\end{aligned}$$

  Note: there is no an "explicit" formula of MD, because it all depends on the choice of ϕ.

- The whole picture:
  1. You are given a function f to minimize
  2. You pick a norm $\|\cdot\|$ and a function ϕ that is λ-strongly convex w.r.t. the chosen $\|\cdot\|$.
  3. You minimize f using mirror descent.

  You are free to choose any $\|\cdot\|$ and ϕ as you like, and when $\|\cdot\|$ and ϕ are "well-chosen", mirror descent converges faster than gradient descent!

Operator theory of the mirror descent algorithm

- From the compact mirror descent update

$$\boldsymbol{x}_{k+1} \;=\; \operatorname*{argmin}_{\boldsymbol{x}} \underbrace{\Big\langle \nabla f(\boldsymbol{x}_k) - \frac{1}{\alpha}\nabla\phi(\boldsymbol{x}_k), \boldsymbol{x} \Big\rangle + \frac{1}{\alpha}\phi(\boldsymbol{x})}_{Q(\boldsymbol{x},\boldsymbol{x}_k)}, \qquad (*)$$

- Similar to what we did in slide 7, to find the explicit solution of $(*)$, we take $\nabla Q(\boldsymbol{x}, \boldsymbol{x}_k)\big|_{\boldsymbol{x}=\boldsymbol{x}_{k+1}} = \mathbf{0}$, which gives

$$\begin{aligned} & \nabla f(\boldsymbol{x}_k) - \frac{1}{\alpha}\nabla\phi(\boldsymbol{x}_k) + \frac{1}{\alpha}\nabla\phi(\boldsymbol{x}_{k+1}) \;=\; \mathbf{0} \\ \iff \quad & \nabla\phi(\boldsymbol{x}_{k+1}) \;=\; \nabla\phi(\boldsymbol{x}_k) - \alpha\nabla f(\boldsymbol{x}_k). \end{aligned}$$

- View $\nabla\phi$ as an operator, we have

$$\boldsymbol{x}_{k+1} \;=\; \Big(\nabla\phi\Big)^{-1}\Big(\nabla\phi(\boldsymbol{x}_k) - \alpha\nabla f(\boldsymbol{x}_k)\Big).$$

  As a side note, $\Big(\nabla\phi\Big)^{-1} = \nabla\Big(\phi^*\Big)$, where ϕ^* is the conjugate of ϕ.

Why mirror descent is called mirror descent

$$\boldsymbol{x}_{k+1} \;=\; \Big(\nabla\phi\Big)^{-1}\Big(\nabla\phi(\boldsymbol{x}_k) - \alpha\nabla f(\boldsymbol{x}_k)\Big).$$

- $\nabla\phi$ and $(\nabla\phi)^{-1}$ are the mirror maps:

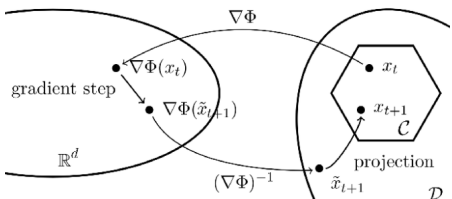


Figure: The "geometry" of mirror descent[1].

[1] Figure from S. Bubeck, Convex optimization: Algorithms and complexity. Foundations and Trends in Machine Learning, 2015.

What about constrained problem

- The mirror descent update $\boldsymbol{x}_{k+1} = \Big(\nabla\phi\Big)^{-1}\Big(\nabla\phi(\boldsymbol{x}_k) - \alpha\nabla f(\boldsymbol{x}_k)\Big)$ we discussed is for unconstrained problem: this update comes from replacing a term in the ordinary gradient descent step.

- For constrained problem $\min_{\boldsymbol{x}\in\mathcal{C}} f(\boldsymbol{x})$, the projected gradient descent adds a Euclidean projection on the gradient step

$$\boldsymbol{x}_{k+1} = P_{\mathcal{C}}\Big(\boldsymbol{x}_k - \alpha\nabla f(\boldsymbol{x}_k)\Big).$$

- For mirror descent, we can do the same: by adding a projection, we arrive at the projected mirror descent

$$\boldsymbol{x}_{k+1} = P_{\mathcal{C},\phi}\bigg(\Big(\nabla\phi\Big)^{-1}\Big(\nabla\phi(\boldsymbol{x}_k) - \alpha\nabla f(\boldsymbol{x}_k)\Big)\bigg).$$

  This corresponds to the $\tilde{\boldsymbol{x}}$ in the figure in the previous slide.

- Note: the projection in projected mirror descent is the Bregman projection, not the Euclidean one.

The deep reasons why we consider mirror descent

- Motivation 1. Primal and dual vector spaces
  - In fact, for a function f, the point $\boldsymbol{x}$ and the gradient $\nabla f(\boldsymbol{x})$ live in different space.
    - $\boldsymbol{x}$ live in the primal space
    - $\nabla f(\boldsymbol{x})$ live in the dual space
  - For optimization in $\mathbb{R}^n$, because $\mathbb{R}^n$ is self-dual, so we take for granted that thinking $\boldsymbol{x}$ and $\nabla f(\boldsymbol{x})$ live in the same space, which in fact technically they live in different space.
  - Gradient descent does not take into account of the primal and dual vector spaces, but mirror descent does $\implies$ why mirror descent is important.

- Motivation 2. Mirror descent using Bregman divergence can handle the case that f is L-smooth with respect to non-Euclidean norm.
  - There are some functions f that the gradient ∇f is L-Lipschitz in, say ℓ_∞ norm.

Last page - summary

- Bregman divergence
- Derivation of the mirror descent: replacing the ℓ_2 norm in gradient descent by Bregman divergence.
- Different expressions of the mirror descent update

Next

- Example of Mirror Descent algorithm
- The convergence theory of mirror descent on convex optimization problem

End of document