PEP in Bregman Method

Zhenghao Xu

Zhejiang University

August 29, 2021

Table of Contents

Progress

Finished:

- (last time) Derived Bound for Bregman Proximal Point
- Complete proof for BPG

Progress

Finished:

- (last time) Derived Bound for Bregman Proximal Point
- Complete proof for BPG

In progress:

- Bregman Halpern iteration for Shannon entropy (numerically unbounded)
- general step size λ_i for BPP's PEP (no time)

Table of Contents

Kernel function

Classical kernel functions:

- $h(x) = \frac{1}{2}\|x\|^2$, $\nabla h(x) = x$
- $h(x) = \sum_{i=1}^{d} x^{(i)} \log x^{(i)}$, $\nabla h(x) = (1 + \log x^{(i)})_i \ (x \geq 0)$
- $h(x) = \sum_{i=1}^{d} - \log x^{(i)}$, $\nabla h(x) = (-\frac{1}{x^{(i)}})_i \ (x \geq 0)$
- $h(x) = \frac{1}{4}\|x\|^4 + \frac{1}{2}\|x\|^2$, $\nabla h(x) = \|x\|^2 x + x$

Kernel function

Last time we discussed the Shannon entropy:

- $h(x) = \sum_{i=1}^{d} x^{(i)} \log x^{(i)}$, $\nabla h(x) = (1 + \log x^{(i)})_i$ $(x \geq 0)$

$\nabla h(x) \leq x$

But this cannot be used in PEP since $\nabla h(x)$ and x are encoded in the Gram matrix. ($G^{s,x}$, $s_i = \nabla h(x_i)$)

Formulation

Problem:

$$\min_{x \in \mathbb{R}^d} \quad \phi(x) \tag{P}$$

Method: Bregman Halpern's iteration

$$x_{k+1} = \nabla h^*[\lambda_k \nabla h(x_0) + (1-\lambda_k)\nabla h(R^h_{\partial\phi}(x_k))] \tag{BH}$$

Bregman resolvent $R^h_T = (\nabla h + \alpha T)^{-1} \circ \nabla h$

Formulation

(relaxed) PEP:

$$\max \quad \min D_h(x_{i+1}, x_i) \qquad \text{(sdp-PEP)}$$

$$s.t. \begin{cases} h_{i-1} - h_i + G^{s,x}_{i-1,i} - G^{s,x}_{i-1,i-1} \le 0 \\ h_{i+1} - h_i + G^{s,x}_{i+1,i} - G^{s,x}_{i+1,i+1} \le 0 \\ h_i - h_* + G^{s,x}_{i,*} - G^{s,x}_{i-1,i-1} \le 0 \\ \phi_{i+1} - \phi_i + G^{w,x}_{i+1,i} - G^{w,x}_{i+1,i+1} \le 0 \\ \phi_i - \phi_* + G^{w,x}_{i,*} - G^{w,x}_{i,i} \le 0 \\ \langle s_k + \alpha w_k - \lambda_{k-1}(s_0 + \alpha w_0) - (1 - \lambda_{k-1}) s_{k-1}, x_j \rangle = 0 \end{cases}$$

(assume $x_* = \phi_* = h_* = 0$)
$P = \big(P_x, P_s, P_w\big) = \big(x_0, \ldots, x_N, s_*, s_0, \ldots, s_N, w_*, w_0, \ldots, w_N\big)$,
$G = P^T P$, $G^{s,x} = P_s^T P_x$, $G^{w,x} = P_w^T P_x$.

Kernel function

We turn to find constraints on $\langle \nabla h(x), x \rangle$:

- $h(x) = \frac{1}{2}\|x\|^2$, $\langle \nabla h(x), x \rangle = 2h(x) = \|x\|^2$
- $h(x) = \sum_{i=1}^{d} x^{(i)} \log x^{(i)}$, $\langle \nabla h(x), x \rangle = \sum_{i=1}^{d} x^{(i)} + h(x)$ $(x \geq 0)$
- $h(x) = \sum_{i=1}^{d} -\log x^{(i)}$, $\langle \nabla h(x), x \rangle = -d \geq -\sum_{i=1}^{d} x^{(i)} - h(x)$ $(x \geq 0)$
- $h(x) = \frac{1}{4}\|x\|^4 + \frac{1}{2}\|x\|^2$, $\langle \nabla h(x), x \rangle = 4h(x) - \|x\|^2$

Bregman Halpern's PEP

For Euclidean case, we still not recover the result given by [Lieder, 2021]. The SDP construction in [Lieder, 2021] is different for it considers the general mapping T.
(Weird)

For the entropy case, we have not found a way to put in the constraints $x > 0$ and relative to $\sum_{i=1}^{d} x^{(i)}$, even we already turn to $\langle \nabla h(x), x \rangle$.
(Turn SDP to QCQP?)

Have tried to use slack variables $s_k + t_k = x_k$, but t_k still appears in inner product.

Have tried to write $x_k^{(i)} = (z_k^{(i)})^2$ to guarantee nonnegative, but inner product is not expressible.

Bregman Halpern's PEP

In Bregman's case, every convexity condition requires inner products with x_i.
But the algorithm iteration can only be expressed in terms of $s_i = \nabla h(x_i)$, so there's no simple way to eliminate x_i from the inner product (unlike in GM Euclidean case in Drori's and Kim's)
So it seems impossible to add constraints on x_i alone. It is always occupied.

Bibliography I

Lieder, F. (2021).
On the convergence rate of the halpern-iteration.
Optimization Letters, pages 1–14.