Efficient Bregman Projections Onto the Simplex

Walid Krichene Syrine Krichene Alexandre Bayen

Electrical Engineering and Computer Sciences, UC Berkeley

ENSIMAG and Criteo Labs, France

December 16, 2015

Outline

Outline

1. Introduction

2. Projection Algorithms

3. Numerical experiments

Bregman Projections onto the simplex

Bregman projections are the building block of mirror descent (Nemirovski and Yudin) and dual averaging (Nesterov).

- Convex optimization: $\min_{x \in \mathcal{X}} f(x)$
- Online learning (regret minimization).

Bregman Projections onto the simplex

Bregman projections are the building block of mirror descent (Nemirovski and Yudin) and dual averaging (Nesterov).

- Convex optimization: $\min_{x \in \mathcal{X}} f(x)$
- Online learning (regret minimization).

Algorithm 2 Mirror descent method

1: **for** $\tau \in \mathbb{N}$ **do**
2: Query a sub-gradient vector $g^{(\tau)} \in \partial f(x^{(\tau)})$ (or loss vector)
3: Update

$$x^{(\tau+1)} = \arg\min_{x \in \mathcal{X}} D_\psi(x, (\nabla\psi)^{-1}(\nabla\psi(x^{(\tau)}) - \eta_\tau g^{(\tau)})) \tag{1}$$

- ψ: strongly convex distance generating function.
- D_ψ: Bregman divergence.

Illustration of Bregman projections

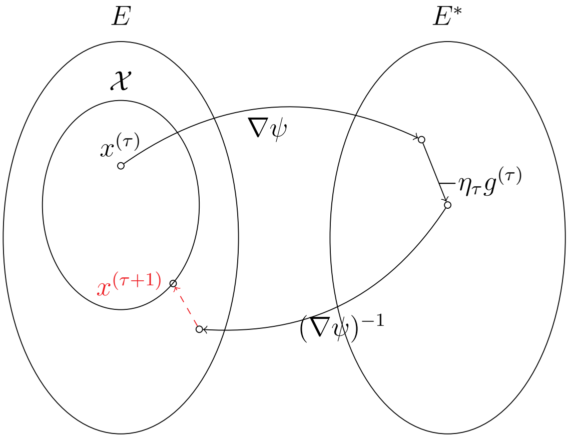


Figure: Illustration of a mirror descent iteration.

$$x^{(\tau+1)} = \arg\min_{x\in\mathcal{X}} D_\psi(x,(\nabla\psi)^{-1}(\nabla\psi(x^{(\tau)}) - \eta_\tau g^{(\tau)}))$$

More precisely

- Feasible set is the simplex (or cartesian product of simplexes)

$$\Delta = \left\{ x \in \mathbb{R}_+^d : \sum_i x_i = 1 \right\}$$

Motivation: online learning, optimization with probability distributions.

More precisely

- Feasible set is the simplex (or cartesian product of simplexes)

$$\Delta = \left\{ x \in \mathbb{R}_+^d : \sum_i x_i = 1 \right\}$$

  Motivation: online learning, optimization with probability distributions.

- DGF is induced by a potential.

$$\psi(x) = \sum_i f(x_i)$$

  $f(x) = \int_1^x \phi^{-1}(u)du$, ϕ increasing, called the potential.
  Consequence: known expression of $\nabla\psi$ and $(\nabla\psi)^{-1}$.

Outline

Projection algorithms

General strategy:

Derive optimality conditions

Design algorithm to satisfy conditions.

Optimality conditions

$$x^\star = \arg\min_{x \in \mathcal{X}} D_\psi\left(x, (\nabla\psi)^{-1}(\nabla\psi(\bar{x}) - \bar{g}\right)$$

Optimality conditions

$x^\star$ is optimal if and only if $\exists \nu^\star \in \mathbb{R}$:

$$\begin{cases} \forall i, \quad x_i^\star = \left(\phi(\phi^{-1}(\bar{x}_i) - \bar{g}_i + \nu^\star)\right)_+, \\ \sum_{i=1}^d x_i^\star = 1, \end{cases}$$

Proof: write KKT conditions, eliminate complementary slackness.

Optimality conditions

$$x^\star = \underset{x \in \mathcal{X}}{\arg\min}\, D_\psi(x, (\nabla\psi)^{-1}(\nabla\psi(\bar{x}) - \bar{g}))$$

Optimality conditions

$x^\star$ is optimal if and only if $\exists \nu^\star \in \mathbb{R}$:

$$\begin{cases} \forall i, \quad x_i^\star = \left(\phi(\phi^{-1}(\bar{x}_i) - \bar{g}_i + \nu^\star)\right)_+, \\ \sum_{i=1}^d x_i^\star = 1, \end{cases}$$

Proof: write KKT conditions, eliminate complementary slackness.
Comments:

- Reduced a problem in dimension d to a problem in dimension 1.
- The function $c : \nu \mapsto \sum_i \left(\phi(\phi^{-1}(\bar{x}_i) - \bar{g}_i + \nu)\right)_+$ is increasing.
- Can solve for $\nu^\star$ using bisection.

Bisection algorithm for general divergences

Algorithm 3 Bisection method to compute the projection $x^\star$ with precision ϵ.

1: Input: $\bar{x}, \bar{g}, \epsilon$.
2: Initialize

$$\bar{\nu} = \phi^{-1}(1) - \max_i \phi^{-1}(\bar{x}_i) - \bar{g}_i$$

$$\underline{\nu} = \phi^{-1}(1/d) - \max_i \phi^{-1}(\bar{x}_i) - \bar{g}_i$$

3: **while** $c(\bar{\nu}) - c(\underline{\nu}) > \epsilon$ **do**
4: Let $\nu^+ \leftarrow \frac{\bar{\nu}+\underline{\nu}}{2}$
5: **if** $c(\nu^+) > 1$ **then**
6: $\bar{\nu} \leftarrow \nu^+$
7: **else**
8: $\underline{\nu} \leftarrow \nu^+$
9: Return $\tilde{x}(\bar{\nu}) = \left(\phi(\phi^{-1}(\bar{x}_i) - \bar{g}_i + \bar{\nu})\right)_+$

Theorem

The algorithm terminates after $\mathcal{O}(\ln \frac{1}{\epsilon})$ iterations, and outputs $\tilde{x}$ such that

$$\|\tilde{x}(\bar{\nu}) - x^\star\| \leq \epsilon$$

Exact projections for exponential divergences

Special case 1:
$\psi(x) = \|x\|^2$: can compute the solution exactly [1].

[1] J. Duchi, S. Shalev-Schwartz, Y. Singer, T. Chandra, Efficient Projections onto the ℓ_1 Ball for Learning in High Dimensions, ICML 2008.

Exact projections for exponential divergences

Special case 1:
$\psi(x) = \|x\|^2$: can compute the solution exactly [1].
Special case 2:
Exponential divergence:

$$\begin{aligned} \phi_\epsilon : (-\infty, +\infty) &\to (-\epsilon, +\infty) \\ u &\mapsto e^{u-1} - \epsilon, \end{aligned}$$

[1] J. Duchi, S. Shalev-Schwartz, Y. Singer, T. Chandra, Efficient Projections onto the ℓ_1 Ball for Learning in High Dimensions, ICML 2008.

Exact projections for exponential divergences

Special case 1:
$\psi(x) = \|x\|^2$: can compute the solution exactly [1].
Special case 2:
Exponential divergence:

$$\begin{aligned} \phi_\epsilon : (-\infty, +\infty) &\to (-\epsilon, +\infty) \\ u &\mapsto e^{u-1} - \epsilon, \end{aligned}$$

- For $\epsilon = 0$:
  $\psi(x) = H(x) = \sum_i x_i \ln x_i$ (negative entropy).
  $D_\psi(x, y) = D_{KL}(x, y)$.

[1] J. Duchi, S. Shalev-Schwartz, Y. Singer, T. Chandra, Efficient Projections onto the ℓ_1 Ball for Learning in High Dimensions, ICML 2008.

Exact projections for exponential divergences

Special case 1:
$\psi(x) = \|x\|^2$: can compute the solution exactly [1].
Special case 2:
Exponential divergence:

$$\begin{aligned} \phi_\epsilon : (-\infty, +\infty) &\to (-\epsilon, +\infty) \\ u &\mapsto e^{u-1} - \epsilon, \end{aligned}$$

- For $\epsilon = 0$:
  $\psi(x) = H(x) = \sum_i x_i \ln x_i$ (negative entropy).
  $D_\psi(x, y) = D_{KL}(x, y)$.
- For $\epsilon > 0$:
  $\psi(x) = H(x + \epsilon)$
  $D_\psi(x, y) = D_{KL}(x + \epsilon, y + \epsilon)$.

[1] J. Duchi, S. Shalev-Schwartz, Y. Singer, T. Chandra, Efficient Projections onto the ℓ_1 Ball for Learning in High Dimensions, ICML 2008.

Motivation

Bregman projection with KL divergence.

- Hedge algorithm in online learning.
- Multiplicative weights algorithm.
- Exponentiated gradient descent.
- Has closed-form solution in $\mathcal{O}(d)$

Motivation

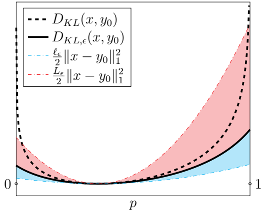
Bregman projection with KL divergence.

- Hedge algorithm in online learning.
- Multiplicative weights algorithm.
- Exponentiated gradient descent.
- Has closed-form solution in $\mathcal{O}(d)$

However:

- $D_{KL}(x, y)$ unbounded on the simplex (problematic for stochastic mirror descent).
- $H(x)$ is not a smooth function (problematic for accelerated mirror descent).

Taking $\epsilon > 0$ solves these issues.



Optimality conditions

Recall general optimality condition: $x_i^\star = \left(\phi(\phi^{-1}(\bar{x}_i) - \bar{g}_i + \nu^\star)\right)_+$.

Optimality conditions with exponential divergence

Let $x^\star$ be the solution and $\mathcal{I} = \{i : x_i^\star > 0\}$ its support. Then

$$\begin{cases} \forall i \in \mathcal{I}, \quad x_i^\star = -\epsilon + \frac{(\bar{x}_i + \epsilon)e^{-\bar{g}_i}}{Z^\star}, \\ Z^\star = \frac{\sum_{i \in \mathcal{I}} (\bar{x}_i + \epsilon)e^{-\bar{g}_i}}{1 + |\mathcal{I}|\epsilon}. \end{cases} \tag{2}$$

Furthermore, if $\bar{y}_i = (\bar{x}_i + \epsilon)e^{-\bar{g}_i}$, then

$$(i \in \mathcal{I} \text{ and } \bar{y}_j > \bar{y}_i) \Rightarrow j \in \mathcal{I}$$

A sorting-based algorithm

Algorithm 4 Sorting method to compute the Bregman projection with D_{ψ_ϵ}

1: Input: $\bar{x}$, $\bar{g}$
2: Output: $x^\star$
3: Form the vector $\bar{y}_i = (\bar{x}_i + \epsilon)e^{-\bar{g}_i}$
4: Sort $\bar{y}$, let $\bar{y}_{\sigma(i)}$ be the i-th smallest element of y.
5: Let $j^\star$ be the smallest index for which

$$(1 + \epsilon(d - j + 1))\bar{y}_{\sigma(j)} - \epsilon \sum_{i \geq j} \bar{y}_{\sigma(i)} > 0$$

6: Set $Z = \frac{\sum_{i \geq j^\star} \bar{y}_{\sigma(i)}}{1+\epsilon(d-j^\star+1)}$
7: Set

$$x_i^\star = \left(-\epsilon + \frac{\bar{y}_i}{Z}\right)_+$$

Complexity: $\mathcal{O}(d \ln d)$

A randomized-pivot algorithm

Adapted from the QuickSelect algorithm: Select i^{th} element of a vector $\bar{y}$.

- Can sort then return i^{th} element: $\mathcal{O}(d \ln d)$.
- QuickSelect: expected $\mathcal{O}(d)$, worst-case $\mathcal{O}(d^2)$.

A randomized-pivot algorithm

$$k = 5 \;\; \boxed{9}\boxed{1}\boxed{4}\boxed{8}\boxed{7}\boxed{2}\boxed{3}\boxed{5}\boxed{6}$$

A randomized-pivot algorithm

$k = 5$									
	9	1	4	8	7	2	3	5	6

A randomized-pivot algorithm

$k = 5$	9	1	4	8	7	2	3	5	6
	1	2	9	4	8	7	3	5	6

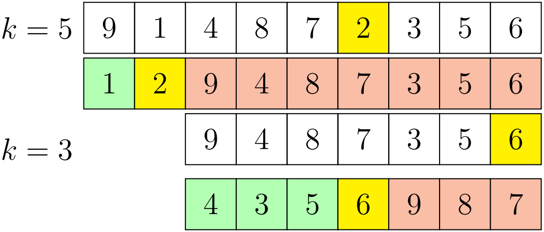
A randomized-pivot algorithm

$k = 5$	9	1	4	8	7	2	3	5	6
	1	2	9	4	8	7	3	5	6
$k = 3$			9	4	8	7	3	5	6

A randomized-pivot algorithm

$k = 5$	9	1	4	8	7	2	3	5	6
	1	2	9	4	8	7	3	5	6
$k = 3$			9	4	8	7	3	5	6

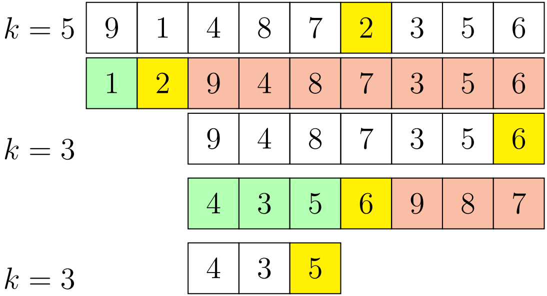
A randomized-pivot algorithm



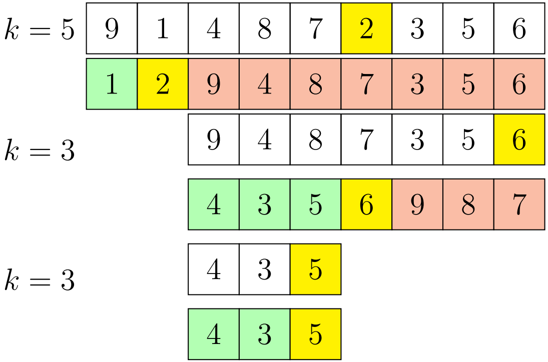
A randomized-pivot algorithm

$k = 5$	9	1	4	8	7	2	3	5	6
	1	2	9	4	8	7	3	5	6
$k = 3$			9	4	8	7	3	5	6
			4	3	5	6	9	8	7
$k = 3$			4	3	5				

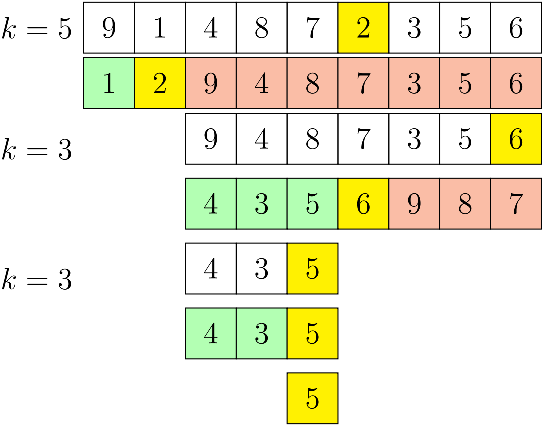
A randomized-pivot algorithm



A randomized-pivot algorithm



A randomized-pivot algorithm



Outline

1. Introduction
2. Projection Algorithms
3. Numerical experiments

Scaling of the SortProject and QuickProject

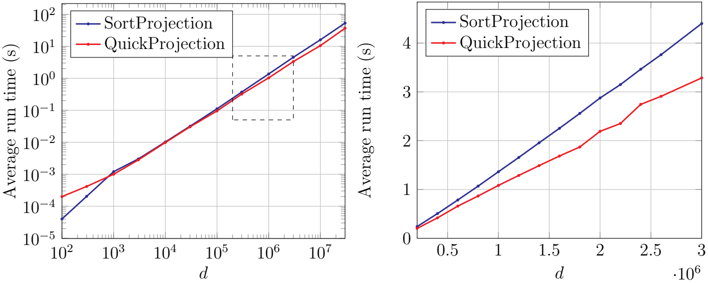


Figure: Execution time of the SortProject and QuickProject algorithms, as a function of problem dimension d

Accelerated entropic descent with and without smoothing

Figure: Entropic descent, with and without smoothing [2]. Offline video

[2] W. Krichene, A. Bayen, P. Bartlett, Accelerated Mirror Descent in Continuous and Discrete Time, NIPS 2015.

Summary

Bregman projection	Method	Complexity
General divergence	Bisection	$\mathcal{O}(\ln \frac{1}{\epsilon})$
Exponential divergence	SortProjection	$\mathcal{O}(d \ln d)$
Exponential divergence	QuickProjection	$\mathcal{O}(d)$ in expection

Used for

- Convex optimization on the simplex.
- Online learning.
- Accelerated entropic descent.
- Code implementation: `github.com/walidk`

Summary

Bregman projection	Method	Complexity
General divergence	Bisection	$\mathcal{O}(\ln \frac{1}{\epsilon})$
Exponential divergence	SortProjection	$\mathcal{O}(d \ln d)$
Exponential divergence	QuickProjection	$\mathcal{O}(d)$ in expection

Used for

- Convex optimization on the simplex.
- Online learning.
- Accelerated entropic descent.
- Code implementation: `github.com/walidk`

Thank you!

`eecs.berkeley.edu/~walid/`

Accelerated entropic descent with and without smoothing

Back

Figure: Entropic descent, with and without smoothing