FAST MOREAU-YOSIDA APPROXIMATE

YVES LUCET

ABSTRACT. Adapting the same idea from the Fast-Legendre transform, we note that the Moreau-Yosida Approximate can be factored as several one-dimensional transforms. Similarly, the monotonicity of the convex subdifferential implies the monotonicity of the proximal point mapping. Hence the quadratic worst-case time complexity to compute the Moreau-Yosida approximate on a grid can be reduced to log-linear.

In fact, the proximal mapping is Lipschitz for any convex function. Hence we present a linear-time algorithm to compute the Moreau-Yosida approximate of convex functions. It has a linear worst-case computation time without using the Fast Legendre Transform algorithm. Hence it avoids computing parameters in the dual space.

CONTENTS

1. Introduction 1
2. Fast Moreau-Yosida approximate 1
3. Building a linear-time algorithm 2
4. Application to other transforms in convex analysis 3
References 3

1. INTRODUCTION

2. FAST MOREAU-YOSIDA APPROXIMATE

For an extended real-valued function $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ The Moreau-Yosida approximate

$$F_\lambda(x) := \inf_{y \in \mathbb{R}^d} [f(y) + \frac{\|x-y\|^2}{2\lambda}]$$

can be factored as d one-dimensional approximates:

$$F_\lambda(x) := \inf_{y_1 \in \mathbb{R}} [\frac{|x_1 - y_1|^2}{2\lambda} + \cdots + \inf_{y_d \in \mathbb{R}} [\frac{|x_d - y_d|^2}{2\lambda} + f(y)] \dots].$$

Hence a fast one-dimensional algorithm will give a fast d-dimensional algorithm. We now recall that the proximal mapping

$$P_\lambda(x) := \operatorname*{Argmin}_{y \in \mathbb{R}^d} [f(y) + \frac{\|x-y\|^2}{2\lambda}]$$

i.e. the set of points where the infimum is attained, is monotone.

Date: May 19, 1999.
This work was partly supported by the Pacific Institute for the Mathematical Sciences.

Lemma. *If the function f is convex, the proximal mapping is a monotone mapping from $\mathbb{R}^d$ to $\mathbb{R}^d$.*

Proof. Take p a selection of P_λ. By definition of p we have the optimality condition

$$0 \in \frac{p(x) - x}{\lambda} + \partial f(p(x))$$

where ∂f denotes the convex subdifferential:

$$\partial f(x) := \{s \in \mathbb{R}^d \ : \ f(y) \geq f(x) + \langle s, y - x\rangle \text{ for all } y\}.$$

For any two points x and x', we have

$$\frac{x - p(x)}{\lambda} \in \partial f(p(x)),$$
$$\frac{x' - p(x')}{\lambda} \in \partial f(p(x')).$$

The monotonicity of the convex subdifferential implies:

$$\langle \frac{x - p(x)}{\lambda} - \frac{x' - p(x')}{\lambda}, p(x) - p(x')\rangle \geq 0,$$

in other words

$$\langle x - x', p(x) - p(x')\rangle \geq \|p(x) - p(x')\|^2. \tag{1}$$

So p is *strongly monotone* for any function f. In particular, for univariate functions, p is increasing. □

Using the same scheme as in [1, 4, 2] we can build a $O((n+m)\ln(n+m))$ worst-case time algorithm to compute the Moreau-Yosida approximate at m points where n is the number of points at which we sample the function f to approximate the infimum.

3. Building a linear-time algorithm

As in [3], we can build a linear-time algorithm as soon as we can compute the one-dimensional transform in linear-time. We use the smoothness of the proximal mapping coupled with carefully selected grids to build the algorithm

Applying the Cauchy-Swartz inequality to (1) gives

$$\|p(x) - p(x')\| \leq \|x - x'\|,$$

in other words, any selection p of the proximal mapping P_λ is 1-Lipschitz. Take two partitions $y_1 < \cdots < y_n$ and $x_1 < \cdots < x_m$. Assume

$$y_{i+1} - y_i = x_{j+1} - x_j =: h$$

for any integer i, j with $1 \leq i \leq n-1$ and $1 \leq j \leq m-1$. The algorithm schematized in Table 1 computes the Moreau-Yosida approximate at all the point on the grid $(x_j)_j$ by approximating the infimum with the computation of the minimum on the grid $(y_i)_i$. In other words, we compute the discrete Moreau-Yosida approximate:

$$x_j \mapsto \min_{1 \leq j \leq n} [\frac{|x_j - y_i|^2}{2\lambda} + f(y_i)]$$

at all points x_i, $1 \leq i \leq n$.

Lemma. *The algorithm in Table 1 computes the Moreau-Yosida approximate in linear-time when the function f is convex.*

Operation	Complexity
Compute $p(x_1)$ by a linear search	n
$p(x_2)$ is either equal to $y_{i_2} := p(x_1)$ or y_{i_2+1}	1
$\vdots$	$\vdots$
$p(x_m)$ is either equal to $y_{i_m} := p(x_1)$ or y_{i_m+1}	1
Worst-case time complexity	$n + m$ operations

FIGURE 1. A linear-time algorithm for the Moreau-Yosida approximate.

Proof. We only need to prove that the algorithm computes the Moreau-Yosida approximate. Since

$$0 \leq p(x_j) - p(x_{j-1}) \leq x_j - x_{j-1} \leq h$$

the only possibilities for $p(x_j) - p(x_{j-1})$ are either 0 or h. In the first case, $p(x_j) = p(x_{j-1})$ and in the second $p(x_j)$ is the successor of $p(x_{j-1})$ in the grid $y_1 < \cdots < y_n$. So the result is indeed a selection of the proximal mapping. Consequently, the algorithm computes the Moreau-Yosida approximate. □

4. Application to other transforms in convex analysis

The Lasry-Lions double envelope $h_{\mu,\lambda}$ can be computed as several Moreau envelope

$$h_{\mu,\lambda}(x) = -F_\mu(-F_\lambda(x)).$$

It is a smooth function [5].

Similarly the proximal hull[1] g_λ can be written

$$g_\lambda(x) = h_{\lambda,\lambda}(x) = -F_\lambda(-F_\lambda(x)).$$

References

[1] L. Corrias. Fast Legendre–Fenchel transform and applications to Hamilton–Jacobi equations and conservation laws. *SIAM J. Numer. Anal.*, 33(4):1534–1558, Aug 1996.

[2] Yves Lucet. A fast computational algorithm for the Legendre–Fenchel transform. *Computational Optimization and Applications*, 6(1):27–57, Jul 1996.

[3] Yves Lucet. Faster than the fast Legendre transform, the linear-time Legendre transform. *Numer. Algorithms*, 16(2):171–185 (1998), 1997.

[4] A. Noullez and M. Vergassola. A fast Legendre transform algorithm and applications to the adhesion model. *Journal of Scientific Computing*, 9(3):259–281, 1994.

[5] R. Tyrrell Rockafellar and Roger J.-B. Wets. *Variational analysis.* Springer-Verlag, Berlin, 1998.

CECM, Department of Mathematics and Statistics,, Simon Fraser University, Burnaby BC, V5A 1S6, Canada

E-mail address: lucet@na-net.ornl.gov

[1] the proximal hull is different from the proximal mapping.