A Derivative-Free Approximate Gradient Sampling Algorithm for Finite Minimax Problems

Speaker: Julie Nutini

Joint work with Warren Hare

University of British Columbia (Okanagan)

III Latin American Workshop on Optimization and Control
January 10-13, 2012

Table of Contents

Setting

We assume that our problem is of the form

$$\min_x f(x) \quad \text{where } f(x) = \max\{f_i : i = 1, \ldots, N\},$$

where each f_i is continuously differentiable, i.e., $f_i \in \mathcal{C}^1$, **BUT** we cannot compute ∇f_i.

Active Set/Gradients

Definition

We define the **active set** of f at a point $\bar{x}$ to be the set of indices

$$A(\bar{x}) = \{i : f(\bar{x}) = f_i(\bar{x})\}.$$

The set of **active gradients** of f at $\bar{x}$ is denoted by

$$\{\nabla f_i(\bar{x})\}_{i \in A(\bar{x})}.$$

Clarke Subdifferential for Finite Max Function

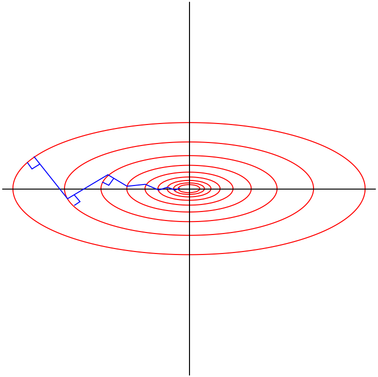
Proposition (Proposition 2.3.12, Clarke '90)

Let $f = \max\{f_i : i = 1, \ldots, N\}$*.*
If $f_i \in \mathcal{C}^1$ *for each* $i \in A(\bar{x})$*, then*

$$\partial f(\bar{x}) = conv\{\nabla f_i(\bar{x})\}_{i \in A(\bar{x})}.$$

Method of Steepest Descent

1. **Initialize:** Set $d^0 = -\text{Proj}(0|\partial f)$.
2. **Step Length:** Find t_k by solving $\min\limits_{t_k>0}\{f(x^k + t_k d^k)\}$.
3. **Update:** Set $x^{k+1} = x^k + t_k d^k$. Increase $k = k + 1$. Loop.



AGS Algorithm - General Idea

- Replace ∂f with an **approximate subdifferential**
- Find an **approximate direction of steepest descent**
- Minimize **along** nondifferentiable ridges

Approximate Gradient Sampling Algorithm
(AGS Algorithm)

AGS Algorithm

1. **Initialize**
2. **Generate Approximate Subdifferential:**
   Sample $Y = [x^k, y^1, \ldots, y^m]$ from around x^k such that
   $$\max_{i=1,\ldots,m} |y^i - x^k| \leq \Delta^k.$$
   Calculate an approximate gradient $\nabla_A f_i$ for each $i \in A(x^k)$ and set
   $$G^k = \text{conv}\{\nabla_A f_i(x^k)\}_{i \in A(x^k)}.$$
3. **Direction:** Set $d^k = -\text{Proj}(0|G^k)$.
4. **Check Stopping Conditions**
5. **(Armijo) Line Search**
6. **Update and Loop**

Convergence

AGS Algorithm

Remark

We define the approximate subdifferential of f at $\bar{x}$ as

$$G(\bar{x}) = \text{conv}\{\nabla_A f_i(\bar{x})\}_{i \in A(\bar{x})},$$

where $\nabla_A f_i(\bar{x})$ is the approximate gradient of f_i at $\bar{x}$.

Lemma

Suppose there exists an $\varepsilon > 0$ *such that* $|\nabla_A f_i(\bar{x}) - \nabla f_i(\bar{x})| \leq \varepsilon$.

Then

1. *for all* $w \in G(\bar{x})$, $\exists\, v \in \partial f(\bar{x})$ *such that* $|w - v| \leq \varepsilon$, *and*
2. *for all* $v \in \partial f(\bar{x})$, $\exists\, w \in G(\bar{x})$ *such that* $|w - v| \leq \varepsilon$.

Proof.

1. By definition, for all $w \in G(\bar{x})$ there exists a set of α_i such that

$$w = \sum_{i \in A(\bar{x})} \alpha_i \nabla_A f_i(\bar{x}), \qquad \text{where } \alpha_i \geq 0, \sum_{i \in A(\bar{x})} \alpha_i = 1.$$

Using the same α_i as above, we see that

$$v = \sum_{i \in A(\bar{x})} \alpha_i \nabla f_i(\bar{x}) \in \partial f(\bar{x}).$$

Then $\qquad |w - v| = |\sum_{i \in A(\bar{x})} \alpha_i \nabla_A f_i(\bar{x})) - \sum_{i \in A(\bar{x})} \alpha_i \nabla f_i(\bar{x})| \leq \varepsilon.$

Hence, for all $w \in G(\bar{x})$, there exists a $v \in \partial f(\bar{x})$ such that

$$|w - v| \leq \varepsilon. \tag{1}$$

□

Convergence

Theorem

Let $\{x^k\}_{k=0}^{\infty}$ be generated by the AGS algorithm.

Suppose there exists a $\bar{K}$ such that given any set Y generated in Step 1 of the AGS algorithm, $\nabla_A f_i(x^k)$ satisfies

$$|\nabla_A f_i(x^k) - \nabla f_i(x^k)| \leq \bar{K}\Delta^k, \text{ where } \Delta^k = \max_{y^i \in Y} |y^i - x^k|.$$

Suppose t^k is bounded away from 0.

Then either

1. $f(x^k) \downarrow -\infty$, *or*
2. $|d^k| \to 0$, $\Delta^k \downarrow 0$ *and every cluster point $\bar{x}$ of the sequence $\{x^k\}_{k=0}^{\infty}$ satisfies $0 \in \partial f(\bar{x})$.*

Convergence - Proof

Proof - Outline.

1. The direction of steepest descent is $-\text{Proj}(0|\partial f(x^k))$.

2. By previous lemma, $G(x^k)$ is a good approximate of $\partial f(x^k)$.

3. So we can show that $-\text{Proj}(0|G(x^k))$ is still a descent direction (approximate direction of steepest descent).

4. Thus, we can show that convergence holds. □

Robust Approximate Gradient Sampling Algorithm

(Robust AGS Algorithm)

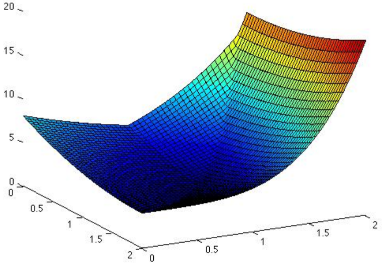
Visual

Consider the function

$$f(x) = \max\{f_1(x), f_2(x), f_3(x)\}$$

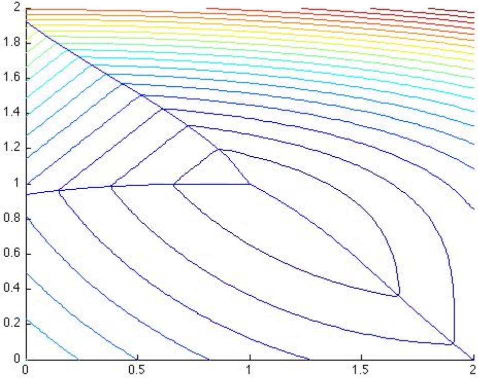
where

$$f_1(x) = x_1^2 + x_2^2$$
$$f_2(x) = (2 - x_1)^2 + (2 - x_2)^2$$
$$f_3(x) = 2 * \exp(x_2 - x_1).$$



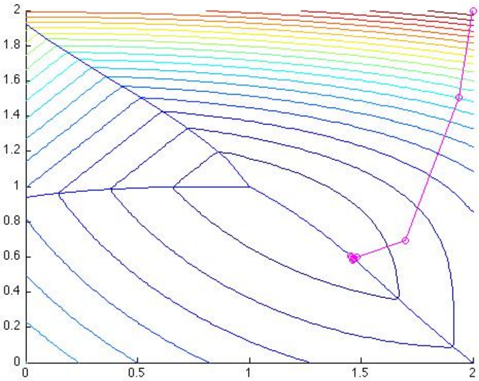
Visual Representation

Contour plot - 'nondifferentiable ridges'



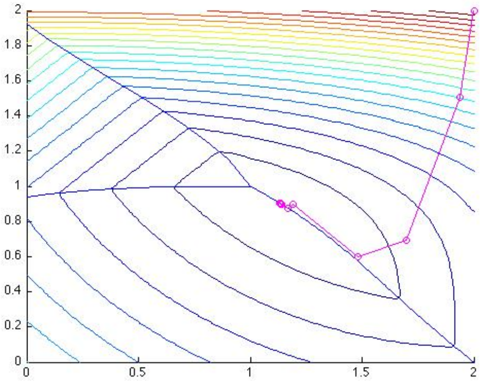
Visual Representation

Regular AGS algorithm:



Visual Representation

Robust AGS algorithm:



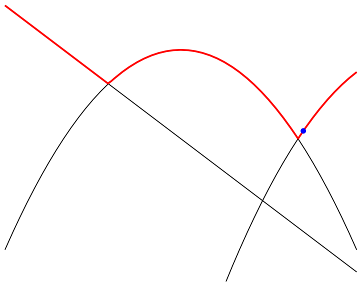
Robust Approximate Gradient Sampling Algorithm

Definition

Let $Y = [x^k, y^1, y^2, \ldots, y^m]$ be a set of sampled points.

The **robust active set** of f on Y is

$$A(Y) = \bigcup_{y^i \in Y} A(y^i).$$



Robust Approximate Gradient Sampling Algorithm

1. **Initialize**
2. **Generate Approximate Subdifferential:**
   Calculate an approximate gradient $\nabla_A f_i$ for each $i \in A(Y)$ and set
   $$G^k = \text{conv}\{\nabla_A f_i(x^k)\}_{i \in A(x^k)},$$
   $$\text{and } G^k_Y = \text{conv}\{\nabla_A f_i(x^k)\}_{i \in A(Y)}.$$
3. **Direction:** Set $d^k = -\text{Proj}(0|G^k)$ and $d^k_Y = -\text{Proj}(0|G^k_Y)$.
4. **Check Stopping Conditions:** Use d^k to check stopping conditions.
5. **(Armijo) Line Search:** Use d^k_Y for the search direction.
6. **Update and Loop**

Convergence

Robust AGS Algorithm

Robust Convergence

Theorem

Let $\{x^k\}_{k=0}^{\infty}$ be generated by the robust AGS algorithm.
Suppose there exists a $\bar{K}$ such that given any set Y generated in Step 1 of the robust AGS algorithm, $\nabla_A f_i(x^k)$ satisfies

$$|\nabla_A f_i(x^k) - \nabla f_i(x^k)| \leq \bar{K}\Delta^k, \text{ where } \Delta^k = \max_{y^i \in Y} |y^i - x^k|.$$

Suppose t^k is bounded away from 0.

Then either

1. $f(x^k) \downarrow -\infty$*, or*
2. $|d^k| \to 0$*,* $\Delta^k \downarrow 0$ *and every cluster point $\bar{x}$ of the sequence $\{x^k\}_{k=0}^{\infty}$ satisfies $0 \in \partial f(\bar{x})$.*

Numerical Results

Approximate Gradients

Requirement

In order for convergence to be guaranteed in the AGS algorithm, $\nabla_A f_i(x^k)$ must satisfy an **error bound** for each of the active f_i:

$$|\nabla_A f_i(x^k) - \nabla f_i(x^k)| \leq \bar{K}\Delta,$$

where $\bar{K} > 0$ and is bounded above.

We used the following 3 approximate gradients:

1. **Simplex Gradient** (see *Lemma 6.2.1*, Kelley '99)
2. **Centered Simplex Gradient** (see *Lemma 6.2.5*, Kelley '99)
3. **Gupal Estimate** (see *Theorem 3.8*, Hare and Nutini '11)

Numerical Results - Overview

- Implementation was done in MATLAB
- 24 nonsmooth test problems (Lukšan-Vlček, '00)
- 25 trials for each problem
- Quality (improvement of digits of accuracy) measured by

$$-\log\left(\frac{|F_{\min} - F^*|}{|F_0 - F^*|}\right)$$

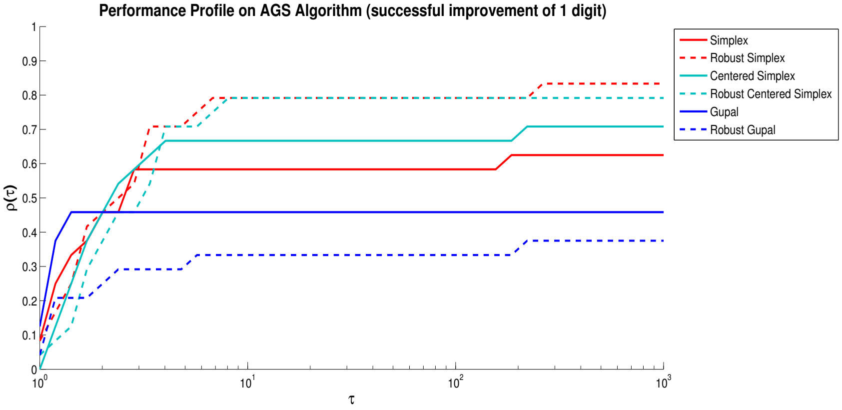
Numerical Results - Goal

Determine any notable numerical differences between:

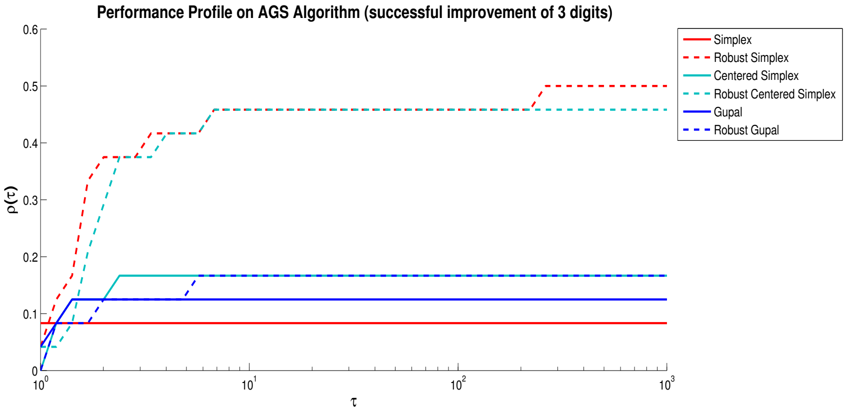
1. Version of the AGS Algorithm
   1. Regular
   2. Robust
2. Approximate Gradient
   1. Simplex Gradient
   2. Centered Simplex Gradient
   3. Gupal Estimate

Thus, 6 different versions of our algorithm were compared.

Performance Profile



Performance Profile



Conclusion

- Approximate gradient sampling algorithm for finite minimax problems
- Robust version
- Convergence ($0 \in \partial f(\bar{x})$)
- Numerical tests

Notes:

- We have numerical results for a robust stopping condition. There is future work in developing the theoretical analysis.
- We are currently working on an application in seismic retrofitting.

Thank you!

References

1. J. V. Burke, A. S. Lewis, and M. L. Overton. *A robust gradient sampling algorithm for nonsmooth, nonconvex optimization*, SIAM J. Optim., 15(2005), pp. 751-779.
2. W. Hare and J. Nutini. *A derivative-free approximate gradient sampling algorithm for finite minimax problems*. Submitted to SIAM J. Optim., 2011.
3. K. C. Kiwiel. *A nonderivative version of the gradient sampling algorithm for nonsmooth nonconvex optimization*, SIAM J. Optim., 20(2010), pp. 1983-1994.
4. C. T. Kelley. Iterative methods for optimization, SIAM, Philadelphia, PA, 1999.
5. L. Lukšan and J. Vlček. *Test Problems for Nonsmooth Unconstrained and Linearly Constrained Optimization*. Technical Report V-798, ICS AS CR, February 2000.