Incremental Gradient, Subgradient and Proximal Methods for Convex Optimization

Dimitri P. Bertsekas

Akilesh B
Indian Institute of Technology, Hyderabad

October 5, 2016

Recall the classical Subgradient and Proximal Algorithms

Convex Optimization problem

$$\text{minimize } f(x) \text{ subject to } x \in X$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is convex, and X is closed and convex.

Classical subgradient project algorithm : Typical iteration

$$x_{k+1} = P_X(x_k - \alpha_k \nabla f(x_k))$$

where α_k is a positive stepsize and ∇ denotes (any) subgradient.

Classical proximal algorithm : Typical iteration

$$x_{k+1} = \arg\min_{x \in X} \left\{ f(x) + \frac{\|x - x_k\|^2}{2\alpha_k} \right\}$$

where α_k is a positive stepsize.

- Proximal has more solid convergence properties, but requires more overhead.
- The dual problem of proximal algorithm is augmented Lagrangian method.

Problems with Many Additive Cost Components

minimize $\sum_{i=1}^{m} f_i(x)$ subject to $x \in X$

where $f_i : \mathbb{R}^n \to \mathbb{R}$ are convex, and X is closed and convex.

Incremental algorithms (long history, early 90s-present): Typical iteration

- Choose an index $i_k \in \{1, 2, \ldots m\}$,
- Perform a subgradient iteration or a proximal iteration:

$$x_{k+1} = P_X(x_k - \alpha_k \nabla f_{i_k}(x_k))$$

or

$$x_{k+1} = \arg\min_{x \in X} \left\{ f_{i_k}(x) + \frac{\|x - x_k\|^2}{2\alpha_k} \right\}$$

- Motivation is to avoid processing all the cost components at each iteration.

Outline

1. Incremental Algorithms
2. Aggregated Incremental Algorithms
3. Incremental Augmented Lagrangian Algorithms
4. Incremental Treatment of Constraints
5. Convergence Analysis

Incremental Subgradient Methods

Problem: min $\sum_{i=1}^{m} f_i(x)$ subject to $x \in X$, where f_i and X are convex.
Long history: LMS (Widrow-Hoff, 1960, for linear least squares without projection), stochastic approximation literature 1960s, neural network literature 1970s

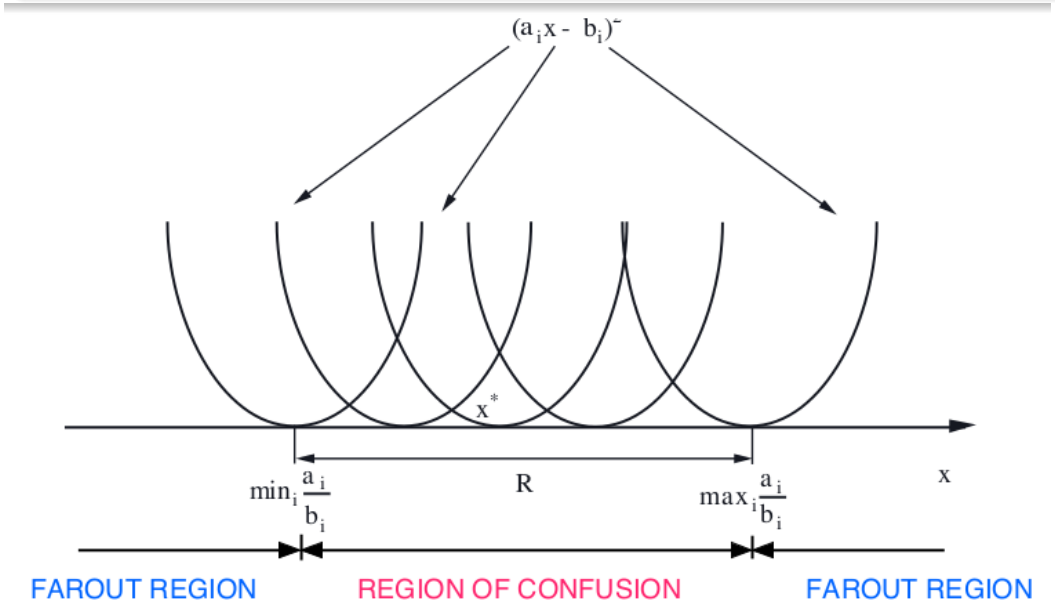
Basic incremental subgradient method

$$x_{k+1} = P_X(x_k - \alpha_k \nabla f_{i_k}(x_k))$$

- Stepsize selection possibilities:
  - $\sum_{k=0}^{\infty} \alpha_k = \infty$ and $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$
  - Constant α_k.
  - Dynamically chosen (based on estimate of optimal cost).
- Index i_k selection possibilities:
  - Cyclically.
  - Fully randomized / equal probability $1/m$.
  - Reshuffling / randomization within a cycle (frequent practical choice).

Convergence Mechanism

Quadratic One-Dimensional Example: min $\sum_{i=1}^{m}(a_i x - b_i)^2$ subject to $x \in \mathbb{R}$,



- Conceptually this idea generalizes to higher dimensions, but is hard to treat/quantify analytically.
- Adapting the stepsize α_k to the farout and confusion regions is an important issue.
- Shaping the confusion region is an important issue.

Incremental Proximal Method

Select index i_k and set

$$x_{k+1} = \arg\min_{x \in X} \left\{ f_{i_k}(x) + \frac{\|x - x_k\|^2}{2\alpha_k} \right\}$$

Many similarities with incremental subgradient

- Similar stepsize choices.
- Similar index selection schemes.
- Can be written as

$$x_{k+1} = P_X(x_k - \alpha_k \nabla f_{i_k}(x_{k+1}))$$

where $\nabla f_{i_k}(x_{k+1})$ is a special gradient at x_{k+1} (index advanced by 1).

Compared to incremental subgradient

- Likely more stable.
- May be harder to implement.

Incremental Subgradient-Proximal methods

Typical iteration

Choose an index $i_k \in \{1, 2, \ldots m\}$ and do a subgradient iteration or a proximal iteration:

$$x_{k+1} = P_X(x_k - \alpha_k \nabla f_{i_k}(x_k))$$

or

$$x_{k+1} = \arg\min_{x \in X} \left\{ f_{i_k}(x) + \frac{\|x - x_k\|^2}{2\alpha_k} \right\}$$

where α_k is a positive stepsize and ∇ denoted any subgradient.

- Idea: Use proximal when easy to implement; use subgradient otherwise.
- A very flexible implementation.
- The proximal iterations still require diminishing α_k for convergence.

Convergence Analysis

Under Lipschitz continuity type assumptions (Nedic and Bertsekas, 2000

- Convergence to the optimum for diminishing stepsize.
- Convergence to the neighborhood of the optimum for constant stepsize.
- Faster convergence for randomized index selection (relative to a worst case cyclic choice).

1 Incremental Algorithms

2 Aggregated Incremental Algorithms

3 Incremental Augmented Lagrangian Algorithms

4 Incremental Treatment of Constraints

5 Convergence Analysis

Incremental Aggregated Gradient Method

$$x_{k+1} = P_X(x_k - \alpha_k \sum_{i=1}^{m} \nabla f_i(x_{l_i}))$$

where $\nabla f_i(x_{l_i})$ is a delayed subgradient of f_i at some earlier iterate x_{l_i} with

$$k - b \leq l_i \leq k \qquad \forall i, k$$

- Key idea : Replace current subgradient components with earlier computed versions.
- Only one component subgradient may be computed per iteration.
- Proposed for nondifferentiable f_i and diminishing step size by Bertsekas et al. (2001).
- **Keywork:**(Blatt et al., 2008) Differentiable strongly convex f_i, no constraints, constant stepsize, and linear convergence.
- This is a gradient method with error proportional to the stepsize.
- A fundamentally different convergence mechanism, relies on differentiability and aims at cost function descent (no region of confusion).

Incremental Aggregated Proximal Method

Select index i_k and set

$$x_{k+1} \in \arg\min_{x \in X} \left\{ f_{i_k}(x) + \sum_{i \neq i_k} \nabla f_i(x_{l_i})(x - x_k) + \frac{\|x - x_k\|^2}{2\alpha_k} \right\}$$

and $\nabla f_i(x_{l_i})$ is a delayed subgradient of f_i at some earlier iterate x_{l_i} with

$$k - b \leq l_i \leq k \qquad \forall i, k$$

Equivalently,

$$x_{k+1} \in \arg\min_{x \in X} \left\{ f_{i_k}(x) + \frac{\|x - z_k\|^2}{2\alpha_k} \right\},$$

where

$$z_k = x_k - \alpha_k \sum_{i \neq i_k} \nabla f_i(x_{l_i})$$

- If f is differentiable and strongly convex, linear convergence can be shown with constant but sufficiently small α_k (DPB 2015).

1 Incremental Algorithms

2 Aggregated Incremental Algorithms

3 Incremental Augmented Lagrangian Algorithms

4 Incremental Treatment of Constraints

5 Convergence Analysis

Separable Convex Optimization problem

minimize $\sum_{i=1}^{m} f_i(x^i)$ subject to $x^i \in X_i, i = \{1, 2, \ldots m\} \sum_{i=1}^{m} h_i x^i = 0$ where $f_i : \mathbb{R}^n \to \mathbb{R}$ are convex, $h_i : \mathbb{R}^n \to \mathbb{R}^r$ are linear and $X_i \subset \mathbb{R}^n$ are closed and convex.

Dual problem decomposes

$$\text{maximize } \sum_{i=1}^{m} q_i(\lambda) \text{ subject to } \lambda \in \mathbb{R}^r$$

where q_i is a "component" dual function:

$$q_i(\lambda) = \inf_{x^i \in X_i} \left\{ f_i(x^i) + \lambda' h_i(x^i) \right\}$$

- The subgradient method exploits the separable structure (Lagrangian relaxation).
- The proximal algorithm yields the augmented Lagrangian method but destroys the separable structure.
- Incremental versions of the proximal algorithm yield incremental augmented Lagrangian methods that exploit the separable structure.

Proximal - Augmented Lagrangian Relation

Proximal Algorithm for the Dual Problem

$$\lambda_{k+1} \in \arg\max_{\lambda \in \mathbb{R}^r} \left\{ \sum_{i=1}^m q_i(\lambda) - \frac{\|\lambda - \lambda_k\|^2}{2\alpha_k} \right\},$$

Dualization using Fenchel duality $\rightarrow$ augmented Lagrangian method

Introduce the augmented Lagrangian function

$$L_\alpha(x, \lambda) = \sum_{i=1}^m f_i(x^i) + \lambda \sum_{i=1}^m h_i(x^i) + \frac{\alpha \left\| \sum_{i=1}^m h_i(x^i) \right\|^2}{2}$$

where $\alpha > 0$ is a parameter. For a sequence $\{\alpha_k\}$ and a starting λ_0, set

$$x_{k+1} \in \arg\min_{x^i \in X^i, i=1,2\ldots m} \quad L_{\alpha_k}(x, \lambda_k)$$

Update λ according to

$$\lambda_{k+1} = \lambda_k + \alpha_k \sum_{i=1}^m h_i(x_{k+1}^i)$$

- A major flaw: min of $L_{\alpha_k}(x, \lambda_k)$ is not separable.

Incremental Augmented Lagrangian Method

Incremental Proximal Algorithm for the Dual Problem

At each iteration k, pick index i_k, and set

$$\lambda_{k+1} \in \arg\max_{\lambda \in \mathbb{R}^r} \left\{ q_{i_k}(\lambda) - \frac{\|\lambda - \lambda_k\|^2}{2\alpha_k} \right\}$$

Dualization using Fenchel duality $\rightarrow$ Incremental augmented Lagrangian method

Pick index i_k, and update the single component x^{i_k} according to

$$x_{k+1}^{i_k} \in \arg\min_{x^{i_k} \in X_{i_k}} \left\{ f_{i_k}(x^{i_k}) + \lambda_k h_{i_k}(x^{i_k}) + \frac{\alpha_k \left\| h_{i_k}(x^{i_k}) \right\|^2}{2} \right\},$$

while keeping the others unchanged, $x_{k+1}^i = x_k^i$ for all $i \neq i_k$.

Update λ according to

$$\lambda_{k+1} = \lambda_k + \alpha_k h_{i_k}(x_{k+1}^{i_k})$$

Incremental Aggregated Augmented Lagrangian Method (IAAL)

Incremental Aggregated Proximal Algorithm for the Dual Problem

At each iteration k, pick index i_k, and set

$$\lambda_{k+1} \in \arg\max_{\lambda \in \mathbb{R}^r} \left\{ q_{i_k}(\lambda) - \frac{\|\lambda - z_k\|^2}{2\alpha_k} \right\}$$

where

$$z_k = \lambda_k + \alpha_k \sum_{i \neq i_k} \nabla q_i(\lambda_{l_i})$$

Implementation of IAAL

Dualization using Fenchel duality $\rightarrow$ incremental aggregated augmented Lagrangian method

Pick index i_k, and update the single component x^{i_k} according to

$$x_{k+1}^{i_k} \in \arg\min_{x^{i_k} \in X_{i_k}} \left\{ f_{i_k}(x^{i_k}) + \lambda_k h_{i_k}(x^{i_k}) + \frac{\alpha_k \left\| h_{i_k}(x^{i_k}) + \sum_{i \neq i_k} h_i(x_{l_i}^i) \right\|^2}{2} \right\}$$

while keeping the others unchanged, $x_{k+1}^i = x_k^i$ for all $i \neq i_k$.

Update λ according to

$$\lambda_{k+1} = \lambda_k + \alpha_k \left(h_{i_k}(x_{k+1}^{i_k}) + \sum_{i \neq i_k} h_i(x_{l_i}^i) \right)$$

Here $h_i(x_{l_i}^i), i \neq i_k$, come from earlier iterations.

Comparison with Alternating Directions Methods of Multipliers (ADMM)

ADMM Iteration for Separable Problems (DPB 1989)

Perform a separate augmented Lagrangian minimization over x^i, for each $i = 1, 2 \dots m$,

$$x_{k+1}^i \in \arg\min_{x^i \in X_i} \left\{ f_i(x^i) + \lambda_k h_i(x^i) + \frac{\alpha \left\| h_i(x^i) - h_i(x_k^i) + \frac{\sum_{j=1}^m h_j(x_k^j)}{m} \right\|^2}{2} \right\},$$

and then update λ_k according to

$$\lambda_{k+1} = \lambda_k + \frac{\alpha \sum_{i=1}^m h_i(x_{k+1}^i)}{m}$$

Comparison: ADMM vs IALL

- The two methods involve fairly similar operations.
- ADMM has guaranteed convergence for any constant α, and under weaker conditions (dual differentiability and strong convexity are not required).
- IAAL has stepsize restrictions.
- At each iteration, all components x^i are updated in ADMM, but a single component x^i is updated in IAAL (m times greater overhead per iteration)

1 Incremental Algorithms

2 Aggregated Incremental Algorithms

3 Incremental Augmented Lagrangian Algorithms

4 Incremental Treatment of Constraints

5 Convergence Analysis

Incremental Methods with Constraint Projection

minimize $\sum_{i=1}^{m} f_i(x)$ subject to $x \in \cap_{l=1}^{q} X_l$
where $f_i : \mathbb{R}^n \to \mathbb{R}$ are convex, and the sets X_l are closed and convex.

Incremental Constraint Projection Algorithm

- Choose an indexes $i_k \in \{1, 2, \ldots m\}$ and $l_k \in \{1, 2, \ldots q\}$,
- Perform a subgradient iteration or a proximal iteration:

$$x_{k+1} = P_{X_{l_k}}(x_k - \alpha_k \nabla f_{i_k}(x_k))$$

or

$$x_{k+1} = \arg\min_{x \in X_{l_k}} \left\{ f_{i_k}(x) + \frac{\|x - x_k\|^2}{2\alpha_k} \right\}$$

where α_k is a positive stepsize and ∇ denotes (any) subgradient.

- Connection to feasibility or alternating projection methods.

1 Incremental Algorithms

2 Aggregated Incremental Algorithms

3 Incremental Augmented Lagrangian Algorithms

4 Incremental Treatment of Constraints

5 Convergence Analysis

Incremental Random Projection Method

Problem

minimize $\sum_{i=1}^{m} f_i(x)$ subject to $x \in \cap_{l=1}^{q} X_l$

where $f_i : \mathbb{R}^n \to \mathbb{R}$ are convex, and the sets X_l are closed and convex.

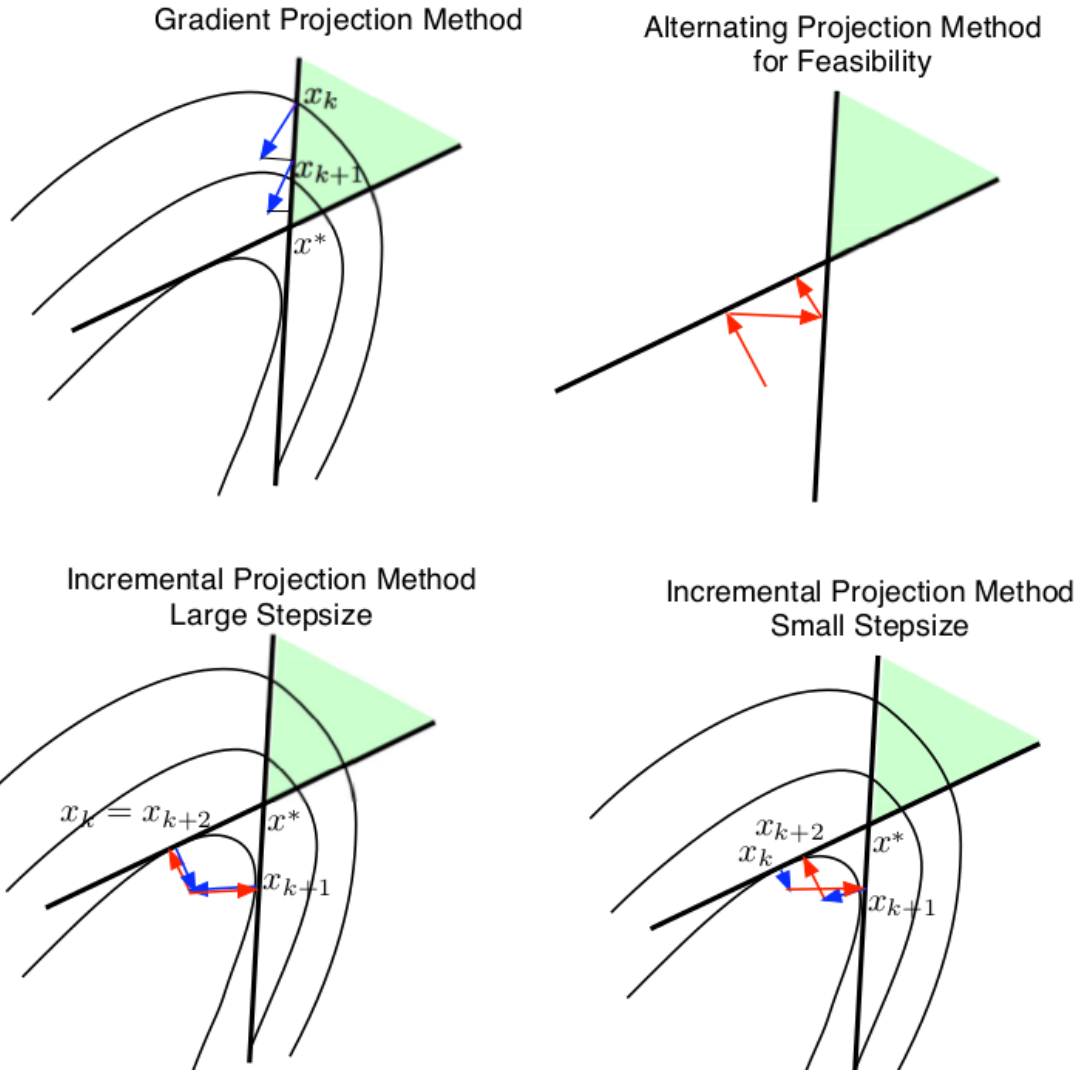
Typical Iteration

- Choose an indexes $i_k \in \{1, 2, \ldots m\}$ and $l_k \in \{1, 2, \ldots q\}$,
- Set
$$x_{k+1} = P_{X_{l_k}}(x_k - \alpha_k \nabla f_{i_k}(\bar{x}_k))$$
- $\bar{x_k} = x_k$ (subgradient iteration) or $\bar{x} = x_{k+1}$ (proximal iteration).
- $\sum_{k=0}^{\infty} \alpha_k = \infty$ and $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$ (diminishing stepsize is essential).

Two way progress

- Progress to feasibility: The projection $P_{l_k}(.)$
- Progress to optimality: The subgradient or proximal iteration $x_k - \alpha_k \nabla f_{i_k}(\bar{x}_k)$

Visualization of Convergence



- Progress to feasibility should be faster than progress to optimality. Gradient stepsizes α_k should be $\ll$ than the feasibility stepsize of 1.

Sampling Schemes for Constraint Index l_k

Nearly independent sampling

$$\inf_{k \geq 0} \text{Prob}\,(l_k = X_l | \mathcal{F}_k > 0), \qquad l = 1, 2 \ldots q$$

where $\mathcal{F}_k$ is the history of the algorithm up to time k.

Cyclic Sampling

Deterministic or random reshuffling every q iterations.

Most distant constraint sampling

$$l_k = \arg\max_{l=1,2\ldots q} \|x_k - P_{X_l}(x_k)\|$$

Markov Sampling

Generate l_k as the state of an ergodic Markov chain with states $1, 2, \ldots q$.

Sampling Schemes for Cost Component Index i_k

Random independent uniform sampling

Each index $i \in \{1, 2, \ldots m\}$ is chosen with equal probability $1/m$, independently of earlier choices.

Cyclic Sampling

Deterministic or random reshuffling every m iterations.

Markov Sampling

Generate i_k as the state of a Markov chain with states $1, 2, \ldots m$, and steady state distribution $\{1/m, 1/m \ldots, 1/m\}$

Convergence Theorem

Assuming Lipschitz continuity of the cost, linear regularity of the constraint, and nonemptiness of the optimal solution set, $\{x_k\}$ converges to some optimal solution x^* $w.p$ 1, under any combination of the preceding sampling schemes.

Idea of the convergence proof

There are two convergence processes taking place:

- **Progress towards feasibility**, which is fast (geometric because of the linear regularity assumption).
- **Progress towards optimality**, which is slower (because of the diminishing stepsize α_k).
- The two-time scale convergence analysis idea is encoded in a coupled **supermartingale convergence theorem**, which governs the evolution of two measures of progress

  $E[dist^2(x_k, X)]$: Distance to the constraint set, which is fast.

  $E[dist^2(x_k, X^*)]$: Distance to the optimal solution set, which is slow.

Concluding Remarks

- Incremental methods exhibit interesting convergence behavior, and can lead to great efficiencies for large-sum cost functions.
- Incremental proximal methods enhance reliability and can be combined seamlessly with incremental gradient / subgradient methods.
- Incremental proximal methods when dualized yield incremental augmented Lagrangian methods that can take advantage of constrained problem separability.
- Constraint projection variants provide flexibility and enlarge the range of potential applications.
- Incremental methods are amenable to distributed asynchronous implementation.

- **Reference** : Incremental Gradient, Subgradient and Proximal Methods for Convex Optimization : A Survey

Thank you !