Incremental Gradient, Subgradient, and Proximal Methods for Convex Optimization

Dimitri P. Bertsekas

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology

April 2016

Recall the Classical Subgradient and Proximal Algorithms

Convex Optimization Problem

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

where $f : \Re^n \mapsto \Re$ is convex, and X is closed and convex.

Classical subgradient projection algorithm: Typical iteration

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

where α_k is a positive stepsize and $\tilde{\nabla}$ denotes (any) subgradient.

Classical proximal algorithm: Typical iteration

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

where α_k is a positive parameter.

- Proximal has more solid convergence properties, but requires more overhead.
- Proximal algorithm $\overset{\text{duality}}{\Longleftrightarrow}$ augmented Lagrangian method.

Problems with Many Additive Cost Components

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

where $f_i : \Re^n \mapsto \Re$ are convex, and X is closed and convex.

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

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

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

or

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

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

Separable Convex Optimization Problem

$$\text{minimize} \quad \sum_{i=1}^{m} f_i(x^i) \qquad \text{subject to} \qquad x^i \in X_i,\ i = 1, \ldots, m, \qquad \sum_{i=1}^{m} h_i(x^i) = 0$$

where $f_i : \Re_i^n \mapsto \Re$ are convex, $h_i : \Re_i^n \mapsto \Re^r$ are linear, $X_i \subset \Re_i^n$ are closed and convex.

Dual problem decomposes

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

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

References for this Overview Talk

- Joint and individual works with A. Nedic and M. Wang.
- Focus on convergence, rate of convergence, component formation, and component selection.

- Work on incremental gradient methods and extended Kalman filter for least squares, 1994-1997 (DPB).
- Work on incremental subgradient methods with A. Nedic, 2000-2010.
- Work on incremental proximal methods, 2010-2012 (DPB).
- Work on incremental constraint projection methods with M. Wang, 2012-2014 (following work by A. Nedic in 2011).
- Work on incremental augmented Lagrangian methods 2015 (DPB).

General references:

- Convex Optimization Algorithms book 2015 (DPB).
- Nonlinear Programming: 3rd edition 2016 (DPB).

Outline

Incremental Subgradient Methods

- Problem: $\min_{x \in X} \sum_{i=1}^{m} f_i(x)$, where f_i and X are convex
- Long history: LMS (Widrow-Hoff, 1960, for linear least squares w/out projection), former Soviet Union literature 1960s, stochastic approximation literature 1960s, neural network literature 1970s

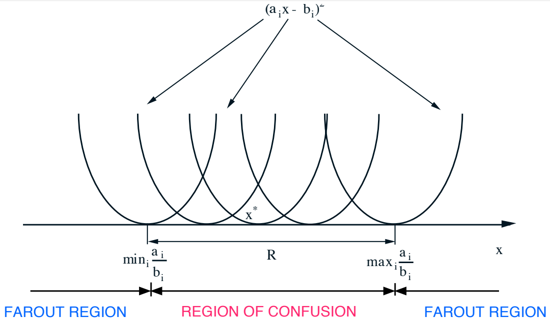
Basic incremental subgradient method

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

- Stepsize selection possibilities:
  - $\sum_{k=0}^{\infty} \alpha_k = \infty$ and $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$
  - α_k: Constant
  - 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: $\min_{x \in \Re} \sum_{i=1}^{m} (c_i x - b_i)^2$



- Conceptually, the 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

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

Many similarities with incremental subgradient

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

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

where $\tilde{\nabla} f_{i_k}(x_{k+1})$ is a special subgradient 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 $i_k \in \{1, \dots, m\}$ and do a subgradient or a proximal iteration

$$x_{k+1} = P_X\big(x_k - \alpha_k \tilde{\nabla} f_{i_k}(x_k)\big) \quad \text{or} \quad x_{k+1} = \arg\min_{x \in X} \left\{ f_{i_k}(x) + \frac{1}{2\alpha_k} \|x - x_k\|^2 \right\}$$

where α_k is a positive stepsize and $\tilde{\nabla}$ denotes (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 a neighborhood of the optimum for constant stepsize.
- Faster convergence for randomized index selection (relative to a worst-case cyclic choice).

Incremental Aggregated Gradient Method

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

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

$$k - b \le \ell_i \le k, \qquad \forall\, i, k.$$

- 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 stepsize by Bertsekas, Nedic, and Borkar (2001)
- Key Work (Blatt, Hero, and Gauchman, 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).
- Intense recent activity by many researchers (Gurbuzbalaban, Ozdaglar, Parrilo, 2015).

Incremental Aggregated Proximal Method

Select index i_k and set

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

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

$$k - b \leq \ell_i \leq k, \qquad \forall\, i, k.$$

Equivalently,

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

where

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

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

Separable Convex Optimization: A Summary

$$\text{minimize} \quad \sum_{i=1}^{m} f_i(x^i) \qquad \text{subject to} \qquad x^i \in X_i,\ i = 1, \ldots, m, \qquad \sum_{i=1}^{m} h_i(x^i) = 0$$

where $f_i : \Re_i^n \mapsto \Re$ are convex, $h_i : \Re_i^n \mapsto \Re^r$ are linear, $X_i \subset \Re_i^n$ are closed and convex.

Dual problem decomposes

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

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

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

Dualization using Fenchel duality –> augmented Lagrangian method

Introduce the augmented Lagrangian function

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

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

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

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 $L_{\alpha_k}(x, l_k)$ is not separable.

Incremental Augmented Lagrangian Method

Incremental Proximal Algorithm for the Dual Problem

At iteration k, pick index i_k, and set

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

Dualization using Fenchel duality –> Incremental augmented Lagrangian method

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

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

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 Proximal Algorithm for the Dual Problem

At iteration k, pick index i_k, and set

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

where

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

Dualization using Fenchel duality –> Incremental aggregated augmented Lagrangian method

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

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

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

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

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

Comparison with Alternating Direction Methods of Multipliers (ADMM)

ADMM Iteration for Separable Problems (DPB 1989)

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

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

and then update λ_k according to

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

Comparison with Incremental Aggregated Augmented Lagrangian

- 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)

Incremental Methods with Constraint Projection

$$\text{minimize} \quad \sum_{i=1}^{m} f_i(x) \qquad \text{subject to} \qquad x \in \cap_{\ell=1}^{q} X_\ell,$$

where $f_i : \Re^n \mapsto \Re$ are convex, and the sets X_ℓ are closed and convex.

Incremental constraint projection algorithm

- Choose indexes $i_k \in \{1, \ldots, m\}$ and $\ell_k \in \{1, \ldots, q\}$.
- Perform a subgradient iteration or a proximal iteration

$$x_{k+1} = P_{X_{\ell_k}}\big(x_k - \alpha_k \tilde{\nabla} f_{i_k}(x_k)\big) \quad \text{or} \quad x_{k+1} = \arg\min_{x \in X_{\ell_k}} \left\{ f_{i_k}(x) + \frac{1}{2\alpha_k} \|x - x_k\|^2 \right\}$$

where α_k is a positive stepsize and $\tilde{\nabla}$ denotes (any) subgradient.

Connection to feasibility/alternating projection methods.

Incremental Random Projection Method

Problem

$$\text{minimize} \quad \sum_{i=1}^{m} f_i(x) \qquad \text{subject to} \qquad x \in X = \cap_{\ell=1}^{q} X_\ell,$$

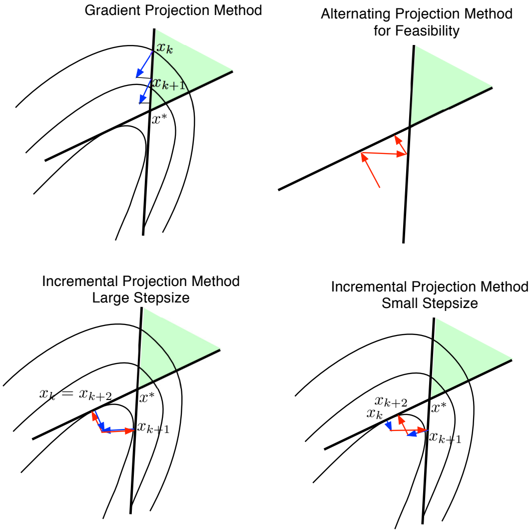
Typical iteration

- Choose indexes $i_k \in \{1, \ldots, m\}$ and $\ell_k \in \{1, \ldots, q\}$.
- Set
$$x_{k+1} = P_{X_{\ell_k}}\big(x_k - \alpha_k \tilde{\nabla} f_{i_k}(\bar{x}_k)\big)$$
- $\bar{x}_k = x_k$ (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 $P_{X_{\ell_k}}(\cdot)$.
- Progress to optimality: The “subgradient/proximal" iteration $x_k - \alpha_k \tilde{\nabla} f_{i_k}(\bar{x}_k)$.

Visualization of Convergence



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

Sampling Schemes for Constraint Index ℓ_k

Nearly independent sampling

$$\inf_{k\geq 0} \text{Prob}(\ell_k = X_\ell \mid \mathcal{F}_k) > 0, \qquad \ell = 1, \ldots, q,$$

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

$$\ell_k = \arg \max_{\ell=1,\ldots,q} \big\| x_k - P_{X_\ell}(x_k) \big\|$$

Markov sampling

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

Sampling Schemes for Cost Component Index i_k

Random independent uniform sampling

Each index $i \in \{1, \ldots, m\}$ 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 $1, \ldots, m$, and steady state distribution $\{1/m, \ldots, 1/m\}$.

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

$$\mathbf{E}[\text{dist}^2(x_k, X)] : \text{ Distance to the constraint set, which is fast}$$

$$\mathbf{E}[\text{dist}^2(x_k, X^*)] : \text{ Distance to the optimal solution set, which is slow}$$

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

Thank you!