Incremental Gradient, Subgradient, and Proximal Methods for Convex Optimization

Dimitri P. Bertsekas

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology

February 2014

Problems with Many Additive Cost and Constraint Components

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

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

Incremental algorithm: Typical iteration

- 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.

Motivation

- Avoid processing all the cost components at each iteration
- Use a simpler constraint to simplify the projection or the proximal minimization

Problems with Many Additive Cost and Constraint Components

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

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

Incremental algorithm: Typical iteration

- 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.

Motivation

- Avoid processing all the cost components at each iteration
- Use a simpler constraint to simplify the projection or the proximal minimization

Problems with Many Additive Cost and Constraint Components

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

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

Incremental algorithm: Typical iteration

- Choose indexes $i_k \in \{1, \dots, m\}$ and $\ell_k \in \{1, \dots, q\}$.
- 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.

Motivation

- Avoid processing all the cost components at each iteration
- Use a simpler constraint to simplify the projection or the proximal minimization

Problems with Many Additive Cost and Constraint Components

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

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

Incremental algorithm: Typical iteration

- 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.

Motivation

- Avoid processing all the cost components at each iteration
- Use a simpler constraint to simplify the projection or the proximal minimization

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).

- See our websites.

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).
- See our websites.

Outline

1. Incremental Algorithms
2. Two Methods for Incremental Treatment of Constraints
3. Convergence Analysis

Outline

1. Incremental Algorithms
2. Two Methods for Incremental Treatment of Constraints
3. Convergence Analysis

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)

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)

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

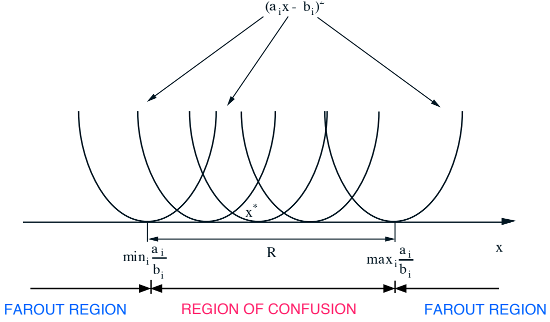
Basic incremental subgradient method

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

- 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

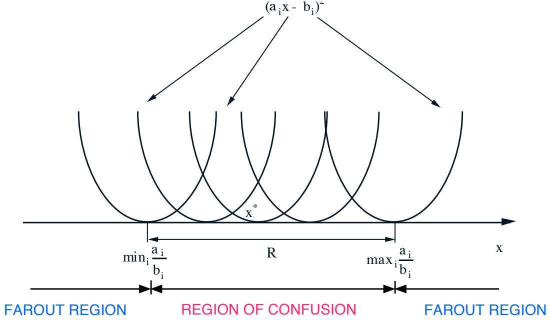
Quadratic One-Dimensional Example: $\min_{x\in\Re} \sum_{i=1}^{m} (a_i x - b_i)^2$



- 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

Convergence Mechanism

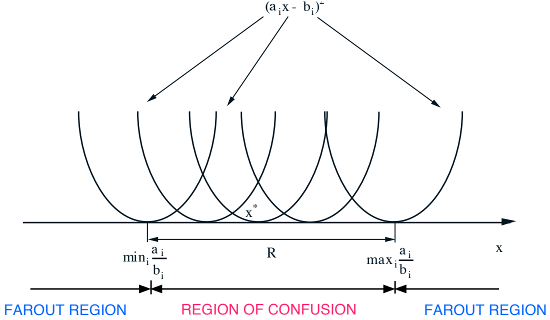
Quadratic One-Dimensional Example: $\min_{x \in \Re} \sum_{i=1}^{m} (a_i x - b_i)^2$



- 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

Convergence Mechanism

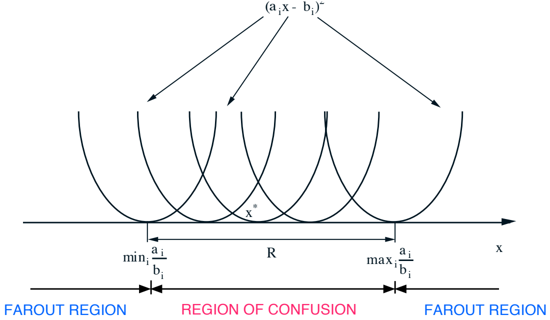
Quadratic One-Dimensional Example: $\min_{x \in \Re} \sum_{i=1}^{m} (a_i x - b_i)^2$



- 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

Convergence Mechanism

Quadratic One-Dimensional Example: $\min_{x \in \Re} \sum_{i=1}^{m} (a_i x - b_i)^2$



- 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

Method with momentum/extrapolation/heavy ball (Polyak 1964): $\beta_k \in [0, 1)$

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

Accelerates in the farout region, decelerates in the confusion region.

Aggregated incremental gradient method

$$x_{k+1} = P_X\left(x_k - \alpha_k \sum_{j=0}^{m-1} \nabla f_{i_{k-j}}(x_{k-j})\right)$$

- Proposed for differentiable f_i, no constraints, cyclic index selection, and constant stepsize, by Blatt, Hero, and Gauchman (2008).
- Recent work by Schmidt, Le Roux, and Bach (2013), randomized index selection, and constant stepsize.
- A fundamentally different convergence mechanism (relies on differentiability and aims at cost function descent). Works even with a constant stepsize (no region of confusion).

Variants of Incremental Gradient Methods/Differentiable Cost

Method with momentum/extrapolation/heavy ball (Polyak 1964): $\beta_k \in [0, 1)$

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

Accelerates in the farout region, decelerates in the confusion region.

Aggregated incremental gradient method

$$x_{k+1} = P_X\left(x_k - \alpha_k \sum_{j=0}^{m-1} \nabla f_{i_{k-j}}(x_{k-j})\right)$$

- Proposed for differentiable f_i, no constraints, cyclic index selection, and constant stepsize, by Blatt, Hero, and Gauchman (2008).
- Recent work by Schmidt, Le Roux, and Bach (2013), randomized index selection, and constant stepsize.
- A fundamentally different convergence mechanism (relies on differentiability and aims at cost function descent). Works even with a constant stepsize (no region of confusion).

Variants of Incremental Gradient Methods/Differentiable Cost

Method with momentum/extrapolation/heavy ball (Polyak 1964): $\beta_k \in [0, 1)$

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

Accelerates in the farout region, decelerates in the confusion region.

Aggregated incremental gradient method

$$x_{k+1} = P_X\left(x_k - \alpha_k \sum_{j=0}^{m-1} \nabla f_{i_{k-j}}(x_{k-j})\right)$$

- Proposed for differentiable f_i, no constraints, cyclic index selection, and constant stepsize, by Blatt, Hero, and Gauchman (2008).
- Recent work by Schmidt, Le Roux, and Bach (2013), randomized index selection, and constant stepsize.
- A fundamentally different convergence mechanism (relies on differentiability and aims at cost function descent). Works even with a constant stepsize (no region of confusion).

Incremental Proximal Methods (DPB, 2010)

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

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 Proximal Methods (DPB, 2010)

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

Aggregated Incremental Proximal Method

Select index i_k and set

$$x_{k+1} = \arg\min_{x \in X} \left\{ f_{i_k}(x) + \sum_{j=1}^{m-1} \tilde{\nabla} f_{i_{k-\ell}}(x_{k-j+1})'(x - x_k) + \frac{1}{2\alpha_k}\|x - x_k\|^2 \right\}$$

where $\tilde{\nabla} f_{i_{k-j}}(x_{k-j+1})$ is special subgradient at x_{k-m+1} (index advanced by 1)

- Can be written as

$$x_{k+1} = P_X\left(x_k - \alpha_k \sum_{j=0}^{m-1} \tilde{\nabla} f_{i_{k-j}}(x_{k-j+1})\right)$$

- More stable (?) than incremental subgradient or proximal
- May be harder to implement
- Convergence can be shown if $\sum_{k=0}^{\infty} \alpha_k = \infty$ and $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$

Aggregated Incremental Proximal Method

Select index i_k and set

$$x_{k+1} = \arg\min_{x \in X} \left\{ f_{i_k}(x) + \sum_{j=1}^{m-1} \tilde{\nabla} f_{i_{k-\ell}}(x_{k-j+1})'(x - x_k) + \frac{1}{2\alpha_k} \|x - x_k\|^2 \right\}$$

where $\tilde{\nabla} f_{i_{k-j}}(x_{k-j+1})$ is special subgradient at x_{k-m+1} (index advanced by 1)

- Can be written as

$$x_{k+1} = P_X \left(x_k - \alpha_k \sum_{j=0}^{m-1} \tilde{\nabla} f_{i_{k-j}}(x_{k-j+1}) \right)$$

- More stable (?) than incremental subgradient or proximal
- May be harder to implement
- Convergence can be shown if $\sum_{k=0}^{\infty} \alpha_k = \infty$ and $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$

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

Typical iteration

Choose $i_k \in \{1, \ldots, m\}$ 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

Under Lipschitz continuity-type assumptions:

- 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).

Notes:

- Fundamentally different from the gradient-proximal method, which applies when $m = 2$,

$$\min_{x \in X} \{f_1(x) + f_2(x)\},$$

  and f_1 is differentiable. This is a cost descent method and can use a constant stepsize.
- Aggregated version possible

Convergence Analysis

Under Lipschitz continuity-type assumptions:

- 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).

Notes:

- Fundamentally different from the gradient-proximal method, which applies when $m = 2$,

$$\min_{x \in X} \{ f_1(x) + f_2(x) \},$$

  and f_1 is differentiable. This is a cost descent method and can use a constant stepsize.
- Aggregated version possible

Convergence Analysis

Under Lipschitz continuity-type assumptions:

- 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).

Notes:

- Fundamentally different from the gradient-proximal method, which applies when $m = 2$,

$$\min_{x \in X} \{f_1(x) + f_2(x)\},$$

  and f_1 is differentiable. This is a cost descent method and can use a constant stepsize.
- Aggregated version possible

Outline

Incremental Treatment of Many Constraints by Exact Penalties

Problem

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

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

Equivalent Problem (Assuming f_i are Lipschitz Continuous)

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

where γ is sufficiently large (the two problems have the same set of minima).

Proximal iteration on the dist(x, X_ℓ) function is easy

Project on X_ℓ and interpolate:

$$x_{k+1} = (1 - \beta_k)x_k + \beta_k P_{X_{i_k}}(x_k), \qquad \beta_k = \min\left\{1, (\alpha_k \gamma)/\text{dist}(x_k; X_{i_k})\right\}$$

(since γ is large, usually $\beta_k = 1$).

Incremental Treatment of Many Constraints by Exact Penalties

Problem

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

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

Equivalent Problem (Assuming f_i are Lipschitz Continuous)

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

where γ is sufficiently large (the two problems have the same set of minima).

Proximal iteration on the dist(x, X_ℓ) function is easy

Project on X_ℓ and interpolate:

$$x_{k+1} = (1 - \beta_k)x_k + \beta_k P_{X_{i_k}}(x_k), \qquad \beta_k = \min\left\{1, (\alpha_k\gamma)/\text{dist}(x_k; X_{i_k})\right\}$$

(since γ is large, usually $\beta_k = 1$).

Incremental Treatment of Many Constraints by Exact Penalties

Problem

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

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

Equivalent Problem (Assuming f_i are Lipschitz Continuous)

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

where γ is sufficiently large (the two problems have the same set of minima).

Proximal iteration on the $\text{dist}(x, X_\ell)$ function is easy

Project on X_ℓ and interpolate:

$$x_{k+1} = (1 - \beta_k)x_k + \beta_k P_{X_{i_k}}(x_k), \qquad \beta_k = \min\left\{1,\ (\alpha_k\gamma)/\text{dist}(x_k; X_{i_k})\right\}$$

(since γ is large, usually $\beta_k = 1$).

Constraint Projection Methods (Thesis by M. Wang and Joint Papers)

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

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.

First proposal and analysis of the case where $m = 1$ and some of the constraints are explicit

$$X_\ell = \{x \mid g_\ell(x) \leq 0\}$$

was by A. Nedic (2011). Connection to feasibility/alternating projection methods.

Constraint Projection Methods (Thesis by M. Wang and Joint Papers)

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

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.

First proposal and analysis of the case where $m = 1$ and some of the constraints are explicit

$$X_\ell = \{x \mid g_\ell(x) \leq 0\}$$

was by A. Nedic (2011). Connection to feasibility/alternating projection methods.

Constraint Projection Methods (Thesis by M. Wang and Joint Papers)

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

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.

First proposal and analysis of the case where $m = 1$ and some of the constraints are explicit

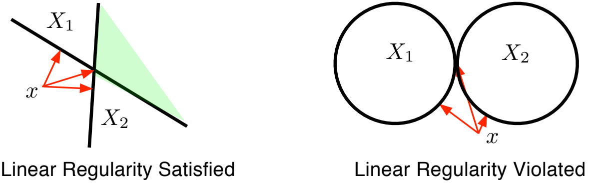
$$X_\ell = \{x \mid g_\ell(x) \leq 0\}$$

was by A. Nedic (2011). Connection to feasibility/alternating projection methods.

Comparison of the two methods

Second method does not require a penalty parameter γ, but needs a linear regularity assumption: For some $\eta > 0$,

$$\left\| x - P_{\cap_{\ell=1}^{q} X_\ell}(x) \right\| \leq \eta \max_{\ell=1,\ldots,q} \| x - P_{X_\ell}(x) \|, \qquad \forall\, x \in \Re^n$$



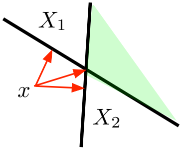
Both methods require diminishing stepsize α_k. Unclear how to construct an aggregated version, or any version that is convergent with a constant stepsize.

The second method involves an interesting two-time scale convergence analysis (the subject of the remainder of this talk).

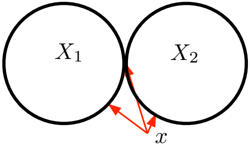
Comparison of the two methods

Second method does not require a penalty parameter γ, but needs a linear regularity assumption: For some $\eta > 0$,

$$\left\| x - P_{\cap_{\ell=1}^{q} X_\ell}(x) \right\| \le \eta \max_{\ell=1,\ldots,q} \|x - P_{X_\ell}(x)\|, \qquad \forall\, x \in \Re^n$$



Linear Regularity Satisfied



Linear Regularity Violated

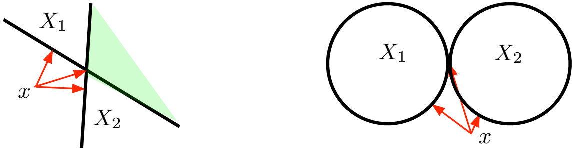
Both methods require diminishing stepsize α_k. Unclear how to construct an aggregated version, or any version that is convergent with a constant stepsize.

The second method involves an interesting two-time scale convergence analysis (the subject of the remainder of this talk).

Comparison of the two methods

Second method does not require a penalty parameter γ, but needs a linear regularity assumption: For some $\eta > 0$,

$$\left\| x - P_{\cap_{\ell=1}^{q} X_\ell}(x) \right\| \leq \eta \max_{\ell=1,\ldots,q} \|x - P_{X_\ell}(x)\|, \qquad \forall\, x \in \Re^n$$



Linear Regularity Satisfied

Linear Regularity Violated

Both methods require diminishing stepsize α_k. Unclear how to construct an aggregated version, or any version that is convergent with a constant stepsize.

The second method involves an interesting two-time scale convergence analysis (the subject of the remainder of this talk).

Outline

Problem

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

Typical iteration

- Choose "randomly" indexes $i_k \in \{1, \ldots, m\}$ and $\ell_k \in \{1, \ldots, q\}$.
- Set

$$x_{k+1} = P_{X_{\ell_k}}\big(x_k - \alpha_k \tilde{\nabla} f_{i_k}(\bar{x}_k)\big)$$

- $\bar{x}_k = x_k$ or $\bar{x}_k = x_{k+1}$.
- $\sum_{k=0}^{\infty} \alpha_k = \infty$ 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" iteration $x_k - \alpha_k \tilde{\nabla} f_{i_k}(\bar{x}_k)$.

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 “randomly" indexes $i_k \in \{1, \ldots, m\}$ and $\ell_k \in \{1, \ldots, q\}$.
- Set
$$x_{k+1} = P_{X_{\ell_k}}\big(x_k - \alpha_k \tilde{\nabla} f_{i_k}(\bar{x}_k)\big)$$
- $\bar{x}_k = x_k$ or $\bar{x}_k = x_{k+1}$.
- $\sum_{k=0}^{\infty} \alpha_k = \infty$ 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" iteration $x_k - \alpha_k \tilde{\nabla} f_{i_k}(\bar{x}_k)$.

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 “randomly" indexes $i_k \in \{1, \ldots, m\}$ and $\ell_k \in \{1, \ldots, q\}$.
- Set

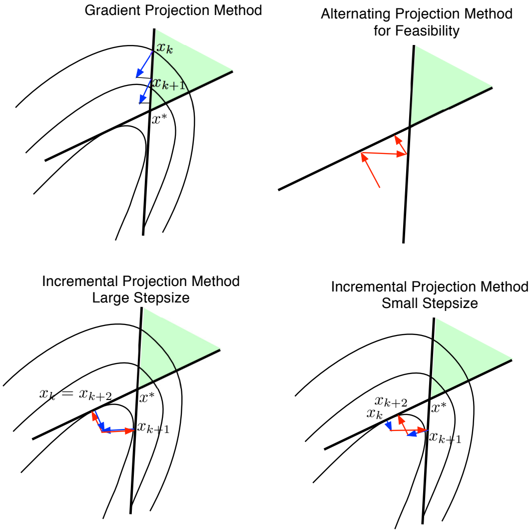
$$x_{k+1} = P_{X_{\ell_k}}\big(x_k - \alpha_k \tilde{\nabla} f_{i_k}(\bar{x}_k)\big)$$

- $\bar{x}_k = x_k$ or $\bar{x}_k = x_{k+1}$.
- $\sum_{k=0}^{\infty} \alpha_k = \infty$ 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" 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.

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. Some constraints may be sampled faster than others.

Cyclic sampling

Deterministic or random reshuffling every q iterations.

Most distant constraint sampling

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

A variant: Skip constraints that are not violated.

Markov sampling

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

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. Some constraints may be sampled faster than others.

Cyclic sampling

Deterministic or random reshuffling every q iterations.

Most distant constraint sampling

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

A variant: Skip constraints that are not violated.

Markov sampling

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

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. Some constraints may be sampled faster than others.

Cyclic sampling

Deterministic or random reshuffling every q iterations.

Most distant constraint sampling

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

A variant: Skip constraints that are not violated.

Markov sampling

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

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. Some constraints may be sampled faster than others.

Cyclic sampling

Deterministic or random reshuffling every q iterations.

Most distant constraint sampling

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

A variant: Skip constraints that are not violated.

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 $\{1/m, \ldots, 1/m\}$.

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 $\{1/m, \ldots, 1/m\}$.

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

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

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

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

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

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

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

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

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

Concluding Remarks

- Incremental methods exhibit interesting and complicated convergence behavior
- Proximal variants enhance reliability
- Constraint projection variants provide flexibility and enlarge the range of potential applications
- Issues not discussed:
  - Distributed asynchronous implementation. The same mechanism that deals with incrementalism (diminishing stepsize), deals also with asynchronism.
  - Incremental Gauss-Newton methods (Extended Kalman Filter). Often faster in least squares problems. Converges in one cycle for linear least squares.

Concluding Remarks

- Incremental methods exhibit interesting and complicated convergence behavior
- Proximal variants enhance reliability
- Constraint projection variants provide flexibility and enlarge the range of potential applications
- Issues not discussed:
  - Distributed asynchronous implementation. The same mechanism that deals with incrementalism (diminishing stepsize), deals also with asynchronism.
  - Incremental Gauss-Newton methods (Extended Kalman Filter). Often faster in least squares problems. Converges in one cycle for linear least squares.

Concluding Remarks

- Incremental methods exhibit interesting and complicated convergence behavior
- Proximal variants enhance reliability
- Constraint projection variants provide flexibility and enlarge the range of potential applications
- Issues not discussed:
  - Distributed asynchronous implementation. The same mechanism that deals with incrementalism (diminishing stepsize), deals also with asynchronism.
  - Incremental Gauss-Newton methods (Extended Kalman Filter). Often faster in least squares problems. Converges in one cycle for linear least squares.

Concluding Remarks

- Incremental methods exhibit interesting and complicated convergence behavior
- Proximal variants enhance reliability
- Constraint projection variants provide flexibility and enlarge the range of potential applications
- Issues not discussed:
  - Distributed asynchronous implementation. The same mechanism that deals with incrementalism (diminishing stepsize), deals also with asynchronism.
  - Incremental Gauss-Newton methods (Extended Kalman Filter). Often faster in least squares problems. Converges in one cycle for linear least squares.

Concluding Remarks

- Incremental methods exhibit interesting and complicated convergence behavior
- Proximal variants enhance reliability
- Constraint projection variants provide flexibility and enlarge the range of potential applications
- Issues not discussed:
  - Distributed asynchronous implementation. The same mechanism that deals with incrementalism (diminishing stepsize), deals also with asynchronism.
  - Incremental Gauss-Newton methods (Extended Kalman Filter). Often faster in least squares problems. Converges in one cycle for linear least squares.

Concluding Remarks

- Incremental methods exhibit interesting and complicated convergence behavior
- Proximal variants enhance reliability
- Constraint projection variants provide flexibility and enlarge the range of potential applications
- Issues not discussed:
  - Distributed asynchronous implementation. The same mechanism that deals with incrementalism (diminishing stepsize), deals also with asynchronism.
  - Incremental Gauss-Newton methods (Extended Kalman Filter). Often faster in least squares problems. Converges in one cycle for linear least squares.

Thank you!