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

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

where \( f_i : \mathbb{R}^n \to \mathbb{R} \) are convex, and the sets \( X_\ell \) 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

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

where \( \alpha_k \) is a positive stepsize and \( \mathbf{\tilde{\nabla}} \) 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

\[
\begin{align*}
\text{minimize} \quad & \sum_{i=1}^{m} f_i(x) \\
\text{subject to} \quad & x \in X = \cap_{\ell=1}^{q} X_{\ell},
\end{align*}
\]

where \( f_i : \mathbb{R}^n \rightarrow \mathbb{R} \) are convex, and the sets \( X_{\ell} \) 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

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

where \( \alpha_k \) 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

Bertsekas (M.I.T.)
Problems with Many Additive Cost and Constraint Components

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

where \( f_i : \mathbb{R}^n \rightarrow \mathbb{R} \) are convex, and the sets \( X_\ell \) 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

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

where \( \alpha_k \) is a positive stepsize and \( \nabla \) 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

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

where \( f_i : \mathbb{R}^n \mapsto \mathbb{R} \) are convex, and the sets \( X_{\ell} \) 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

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

where \( \alpha_k \) 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

- Focus on convergence, rate of convergence, component formation, and component selection.

- Work on incremental subgradient methods with A. Nedic, 2000-2010.
- Work on incremental proximal methods, 2010-2012 (DPB).

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

Work on incremental subgradient methods with A. Nedic, 2000-2010.
Work on incremental proximal methods, 2010-2012 (DPB).

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
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 literature 1970s

Basic incremental subgradient method

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

- Stepsize selection possibilities:
  - \( \sum_{k=0}^{\infty} \alpha_k = \infty \) and \( \sum_{k=0}^{\infty} \alpha_k^2 < \infty \)
  - \( \alpha_k \): 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)
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 literature 1970s

Basic incremental subgradient method

\[
x_{k+1} = P_X (x_k - \alpha_k \tilde{\nabla} f_i(x_k))
\]

Stepsize selection possibilities:
- \( \sum_{k=0}^{\infty} \alpha_k = \infty \) and \( \sum_{k=0}^{\infty} \alpha_k^2 < \infty \)
- \( \alpha_k \): 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**

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

- **Stepsize selection possibilities:**
  - \( \sum_{k=0}^{\infty} \alpha_k = \infty \) and \( \sum_{k=0}^{\infty} \alpha_k^2 < \infty \)
  - \( \alpha_k \): 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)
Quadratic One-Dimensional Example: \[ \min_{x \in \mathbb{R}} \sum_{i=1}^{m} (a_i x - b_i)^2 \]

Conceptually, the idea generalizes to higher dimensions, but is hard to treat/quantify analytically.

Adapting the stepsize \( \alpha_k \) to the farout and confusion regions is an important issue.

Shaping the confusion region is an important issue.
Convergence Mechanism

Quadratic One-Dimensional Example:
\[ \min_{x \in \mathbb{R}} \sum_{i=1}^{m} (a_i x - b_i)^2 \]

- Conceptually, the idea generalizes to higher dimensions, but is hard to treat/quantify analytically.
- Adapting the stepsize \( \alpha_k \) to the farout and confusion regions is an important issue.
- Shaping the confusion region is an important issue.
Quadratic One-Dimensional Example:

\[ \min_{x \in \mathbb{R}} \sum_{i=1}^{m} (a_i x - b_i)^2 \]

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

Adapting the stepsize $\alpha_k$ to the farout and confusion regions is an important issue.

Shaping the confusion region is an important issue.
Variants of Incremental Gradient Methods/Differentiable Cost

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

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

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): $\beta_k \in [0, 1)$

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

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): $\beta_k \in [0, 1)$

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

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).
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

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

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

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

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

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

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

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

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 $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$
Select index $i_k$ and set

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

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

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

- 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 $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$
Typical iteration

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

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

where $\alpha_k$ 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 $\alpha_k$ for convergence
Typical iteration

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

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

where $\alpha_k$ 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 $\alpha_k$ 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
  \( m = 2 \),

  \[
  \min_{x \in X} \left\{ f_1(x) + f_2(x) \right\},
  \]

  and \( f_1 \) is differentiable. This is a cost descent method and can use a constant
  stepsize.
- Aggregated version possible
1. Incremental Algorithms
2. Two Methods for Incremental Treatment of Constraints
3. Convergence Analysis
Problem

\[
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{m} f_i(x) \\
\text{subject to} & \quad x \in \cap_{\ell=1}^{q} X_{\ell},
\end{align*}
\]

where \( f_i : \mathbb{R}^n \rightarrow \mathbb{R} \) are convex, and the sets \( X_{\ell} \) are closed and convex.

Equivalent Problem (Assuming \( f_i \) are Lipschitz Continuous)

\[
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{m} f_i(x) + \gamma \sum_{\ell=1}^{q} \text{dist}(x, X_{\ell}) \\
\text{subject to} & \quad x \in \mathbb{R}^n,
\end{align*}
\]

where \( \gamma \) 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 \( X_{\ell} \) and interpolate:

\[
x_{k+1} = (1 - \beta_k)x_k + \beta_k P_{X_{ik}}(x_k), \quad \beta_k = \min \left\{ 1, \frac{\alpha_k \gamma}{\text{dist}(x_k; X_{ik})} \right\}
\]

(since \( \gamma \) is large, usually \( \beta_k = 1 \)).
Problem

\[
\text{minimize } \sum_{i=1}^{m} f_i(x) \quad \text{subject to } x \in \bigcap_{\ell=1}^{q} X_\ell,
\]

where \( f_i : \mathbb{R}^n \mapsto \mathbb{R} \) are convex, and the sets \( X_\ell \) are closed and convex.

Equivalent Problem (Assuming \( f_i \) are Lipschitz Continuous)

\[
\text{minimize } \sum_{i=1}^{m} f_i(x) + \gamma \sum_{\ell=1}^{q} \text{dist}(x, X_\ell) \quad \text{subject to } x \in \mathbb{R}^n,
\]

where \( \gamma \) 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 \( X_\ell \) and interpolate:

\[
x_{k+1} = (1 - \beta_k)x_k + \beta_k P_{X_\ell}(x_k), \quad \beta_k = \min \left\{ 1, \frac{(\alpha_k \gamma)}{\text{dist}(x_k; X_\ell)} \right\}
\]

(since \( \gamma \) is large, usually \( \beta_k = 1 \)).
Problem

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

where \( f_i : \mathbb{R}^n \to \mathbb{R} \) are convex, and the sets \( X_\ell \) are closed and convex.

Equivalent Problem (Assuming \( f_i \) are Lipschitz Continuous)

\[
\text{minimize } \sum_{i=1}^{m} f_i(x) + \gamma \sum_{\ell=1}^{q} \text{dist}(x, X_\ell) \quad \text{subject to } x \in \mathbb{R}^n,
\]

where \( \gamma \) 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 \( X_\ell \) and interpolate:

\[
x_{k+1} = (1 - \beta_k)x_k + \beta_k P_{X_{i_k}}(x_k), \quad \beta_k = \min \left\{ 1, \frac{(\alpha_k \gamma)}{\text{dist}(x_k; X_{i_k})} \right\}
\]

(since \( \gamma \) is large, usually \( \beta_k = 1 \)).
minimize $\sum_{i=1}^{m} f_i(x)$ subject to $x \in \bigcap_{\ell=1}^{q} X_\ell$,

where $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ are convex, and the sets $X_\ell$ 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

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

where $\alpha_k$ 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.
minimize $\sum_{i=1}^{m} f_i(x)$ subject to $x \in \bigcap_{\ell=1}^{q} X_\ell$,

where $f_i : \mathbb{R}^n \mapsto \mathbb{R}$ are convex, and the sets $X_\ell$ 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

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

where $\alpha_k$ 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.
minimize $\sum_{i=1}^{m} f_i(x)$ subject to $x \in \cap_{\ell=1}^{q} X_\ell$,

where $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ are convex, and the sets $X_\ell$ 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**

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

where $\alpha_k$ 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 $\gamma$, but needs a **linear regularity assumption**: For some $\eta > 0$,

\[
\|x - P_{\bigcap_{\ell=1}^{q} X_\ell}(x)\| \leq \eta \max_{\ell=1,\ldots,q} \|x - P_{X_\ell}(x)\|, \quad \forall \, x \in \mathbb{R}^n
\]

Both methods require diminishing stepsize $\alpha_k$. **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 $\gamma$, but needs a **linear regularity assumption**: For some $\eta > 0$,

\[
\left\| x - P_{\cap_{\ell=1}^q X_{\ell}}(x) \right\| \leq \eta \max_{\ell=1,...,q} \left\| x - P_{X_{\ell}}(x) \right\|, \quad \forall x \in \mathbb{R}^n
\]

Both methods require diminishing stepsize $\alpha_k$. **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 $\gamma$, but needs a **linear regularity assumption**: For some $\eta > 0$,

$$\left\| x - P_{\bigcap_{\ell=1}^{q} X_{\ell}}(x) \right\| \leq \eta \max_{\ell=1,\ldots,q} \left\| x - P_{X_{\ell}}(x) \right\|, \quad \forall x \in \mathbb{R}^n$$

Both methods require diminishing stepsize $\alpha_k$. **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

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

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

Typical iteration

- Choose “randomly” indexes \(i_k \in \{1, \ldots, m\}\) and \(\ell_k \in \{1, \ldots, q\}\).
- Set
  \[
x_{k+1} = P_{X_{\ell_k}} (x_k - \alpha_k \tilde{\nabla} f_{i_k}(\bar{x}_k))
  \]
- \(\bar{x}_k = x_k\) 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**

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

**Typical iteration**

- Choose “randomly" indexes \( i_k \in \{1, \ldots, m\} \) and \( \ell_k \in \{1, \ldots, q\} \).
- Set
  \[
  x_{k+1} = P_{X_{\ell_k}}(x_k - \alpha_k \tilde{\nabla} f_{i_k}(\bar{x}_k))
  \]
- \( \bar{x}_k = x_k \) 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

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

## Typical iteration

- Choose "randomly" indexes \( i_k \in \{1, \ldots, m\} \) and \( \ell_k \in \{1, \ldots, q\} \).
- Set
  \[
x_{k+1} = P_{X_{\ell_k}} (x_k - \alpha_k \tilde{\nabla} f_{i_k}(\bar{x}_k))
\]
- \( \bar{x}_k = x_k \) 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 \( x_k - \alpha_k \tilde{\nabla} f_{i_k}(\bar{x}_k) \).

---

**Bertsekas (M.I.T.)**

Incremental Gradient

18 / 24
Progress to feasibility should be faster than progress to optimality. Gradient stepsizes $\alpha_k$ should be $<<$ than the feasibility stepsize of 1.
Sampling Schemes for Constraint Index $\ell_k$

**Nearly independent sampling**

$$\inf_{k \geq 0} \text{Prob}(\ell_k = X_\ell | F_k) > 0, \quad \ell = 1, \ldots, q,$$

where $F_k$ 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 $\ell_k$ as the state of an ergodic Markov chain with states 1, \ldots, $q$. 
Nearly independent sampling

\[ \inf_{k \geq 0} \text{Prob}(\ell_k = X_\ell | \mathcal{F}_k) > 0, \quad \ell = 1, \ldots, q, \]

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 \( \ell_k \) as the state of an ergodic Markov chain with states 1, \ldots, \( q \).
### Sampling Schemes for Constraint Index $\ell_k$

#### Nearly independent sampling

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

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 $\ell_k$ as the state of an ergodic Markov chain with states $1, \ldots, q$. 
Sampling Schemes for Constraint Index $\ell_k$

**Nearly independent sampling**

\[ \inf_{k \geq 0} \text{Prob}(\ell_k = X_\ell | F_k) > 0, \quad \ell = 1, \ldots, q, \]

where $F_k$ 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 $\ell_k$ 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 $\{1/m, \ldots, 1/m\}$. 

Bertsekas (M.I.T.)
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 $\{1/m, \ldots, 1/m\}$. 
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 $\{1/m, \ldots, 1/m\}$. 
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 \( \alpha_k \)).
- This two-time scale convergence analysis idea is encoded in a coupled supermartingale convergence theorem, which governs the evolution of two measures of progress:
  
  \[
  E[\text{dist}^2(x_k, X)] : \text{Distance to the constraint set, which is fast} \\
  E[\text{dist}^2(x_k, X^*)] : \text{Distance to the optimal solution set, which is slow}
  \]
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 \( \alpha_k \)).

This two-time scale convergence analysis idea is encoded in a coupled supermartingale convergence theorem, which governs the evolution of two measures of progress:

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

This two-time scale convergence analysis idea is encoded in a **coupled supermartingale convergence theorem**, which governs the evolution of two measures of progress

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

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

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 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.

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 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.

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.
Thank you!