Subgradient Method

2023.03.23

박정현, 전상후

0. Introduction

Why subgradient method?

Q. 지난주 다룬 알고리즘들은 제약 조건이 없으며 1~2번 미분 가능한 convex function을 최적화
그렇다면 제약조건이 있거나 미분이 불가능한 점이 있을 경우 어떻게 목적함수를 최소화할 수 있을까?

A. Subgradient method를 사용할 것
Non-differentiable convex function도 최소화할 수 있는 알고리즘이며, 제약조건이 있는 경우로 확장이 가능

같은 first order method에 속해 gradient method와 매우 유사한 방식으로 작동하나, 다음과 같은 차이점이 있음

- Non-differentiable한 convex function에 바로 적용될 수 있다.
- Step length를 line search를 통해 구하지 않는다. 선제적으로 결정된 값이 사용된다.
- Descent method가 아니다. 즉 함숫값은 증가할 수 있으며 실제로도 때때로 그렇다.

1부 요약: Subgradient는 무엇이며 이것이 수학적으로 어떤 성질을 지니는가? Subgradient method는 어떻게 작동하는가?

1

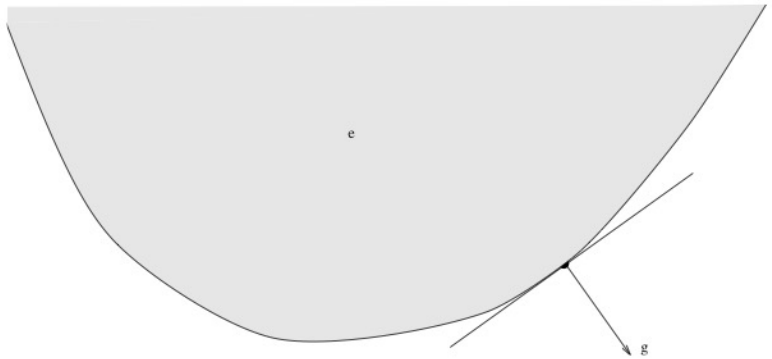
Subgradient

Subgradient의 정의와 성질

1.1 Definition

What is subgradient?

dom f의 x에 대해 모든 z가 다음을 만족할 때 vector g는 **subgradient** $f(z) \geq f(x) + g^T(z - x)$



기하적으로 (g, -1)은 epi f의 supporting hyperplane *note: $a^T z \leq b$

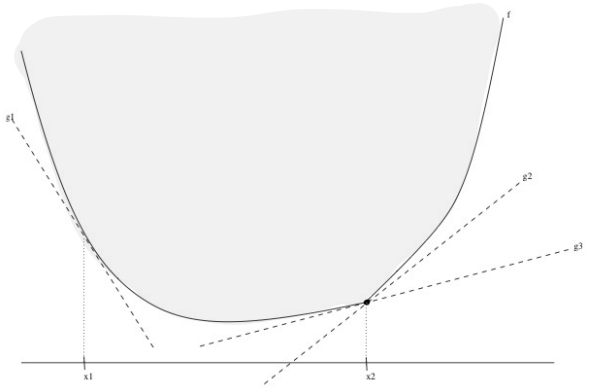
g가 x hat에서의 subgradient일 때, f의 모든 점 (x, f(x))와 epi f의 모든 점 (x, t)에 대해…

(subgradient의 정의) $f(x) \geq f(\hat{x}) + g^T(x - \hat{x})$ (epigraph의 정의) $t \geq f(\hat{x}) + g^T(x - \hat{x})$

$g^T x - t \leq g^T \hat{x} - f(\hat{x}) \rightarrow (g, -1)^T(x, t) \leq -f(\hat{x}) + g^T \hat{x}$ 우변이 상수, (x, t)가 epi f의 점

1.1 Definition

What is subgradient? $f(z) \geq f(x) + g^T(z - x)$



convex

- Differentiable: 유일하게 존재 (gradient와 일치)
- Non-differentiable: 여러 개 존재

non-convex

- 반드시 존재하지는 않음

즉, subgradient는 미분 불가능한 점에 대해서도 정의될 수 있으며
convexity는 subgradient의 존재를 보장해줌

f의 어떤 점 x에서 sub-gradient가 적어도 하나 존재할 경우, f는 x에서 **subdifferentiable**

이 점에서의 모든 subgradient 집합은 **subdifferential of f at x** $\partial f(x)$

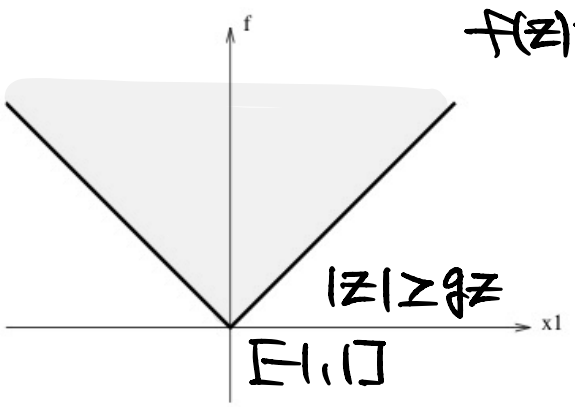
dom f의 모든 x에 대해 f가 subdifferentiable 하다면, **f는 subdifferentiable**

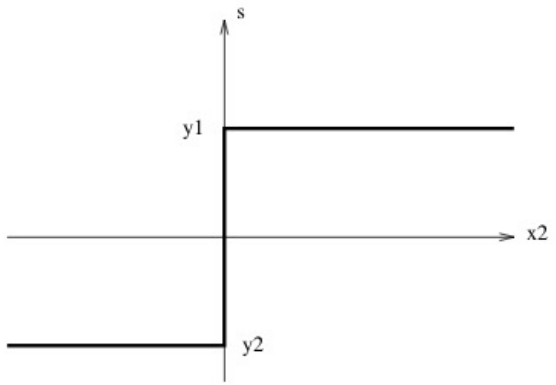
Set 전체를 구하는 것은 strong calculus, 그 중 하나의 원소 g만을 구하는 것은 weak calculus

전자는 이론적인 측면에서 의미를 가지나, practical하게 사용되는 method들은 후자로 충분

1.1 Definition

Definition and basic properties $f(z) \geq f(x) + g^T(z - x)$





Subgradient는 다음의 특성을 지님

① f가 x에서 미분 가능하고 볼록 함수라면 $\partial f(x)$는 $\{\nabla f(x)\}$만을 원소로 갖는다. (gradient가 유일한 원소)

② 만약 $\partial f(x) = \{g\}$ 라면 (원소가 유일하게 존재) f는 x에서 미분 가능하며, $g = \nabla f(x)$이다.

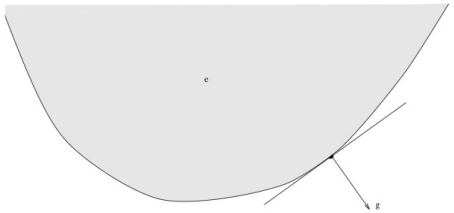
③ 집합 $\partial f(x)$는 f가 convex인지 아닌 지와 무관하게 항상 closed convex set이 된다.

$f(z) - f(x) \geq g^T(z - x)$ 는 half space이고, half space는 항상 closed convex set
Closed convex set들의 intersection은 마찬가지로 closed convex set

1.2 Basic Properties

Proof of basic properties $f(z) \geq f(x) + g^T(z - x)$

④ f가 convex하다면 $\partial f(x)$는 반드시 하나 이상의 원소를 가진다. (non-convex일 시 공집합이 될 수도 있음)



(remind) f가 convex 하다면 epi f도 convex

(remind) Convex set C의 boundary에 있는 점 x0에 대해, x0에서의 supporting hyperplane이 항상 존재

$a^T(x - x_0) \leq 0$ 을 만족하는 non-zero a가 존재 [*Supporting hyperplane theorem*]

따라서, epi f의 점 (z, t)와 그 boundary인 f의 점 (x, f(x))에 대해 다음 식을 만족하는 non-zero a가 항상 존재하며,

$$\begin{bmatrix} a \\ b \end{bmatrix}^T \left(\begin{bmatrix} z \\ t \end{bmatrix} - \begin{bmatrix} x \\ f(x) \end{bmatrix} \right) = a^T(z - x) + b(t - f(x)) \leq 0$$

f(z)보다는 epi f(z)인 t값이 무조건 크거나 같으므로 t 대신 f(z) 대입해도 부등식 성립

이때 b는 0이 아니므로, 양변을 b로 나누어 정리하면 $f(z) \geq f(x) - (a/b)^T(z - x)$

따라서 subgradient의 정의에 의해 $-a/b \in \partial f(x)$. 즉 적어도 하나의 원소를 가지게 된다.

1.3 Calculus of subgradients

Calculus properties $f(z) \geq f(x) + g^T(z - x)$

⑤ 어떤 x^*가 함수 f의 minimizer가 되기 위한 필요충분조건은 g=0이 x^*에서의 subgradient인 것이다.

$$f(x) \geq f(x^*) \;\leftrightarrow\; f(x) \geq f(x^*) + 0^T(x - x^*)$$

(+) 이는 다변수함수로 확장 가능하다. F(x, y)의 y에 대한 infimum이 x에서 구해질 때, ∂F(x, y)에는 (g, 0)이 있어야 한다.

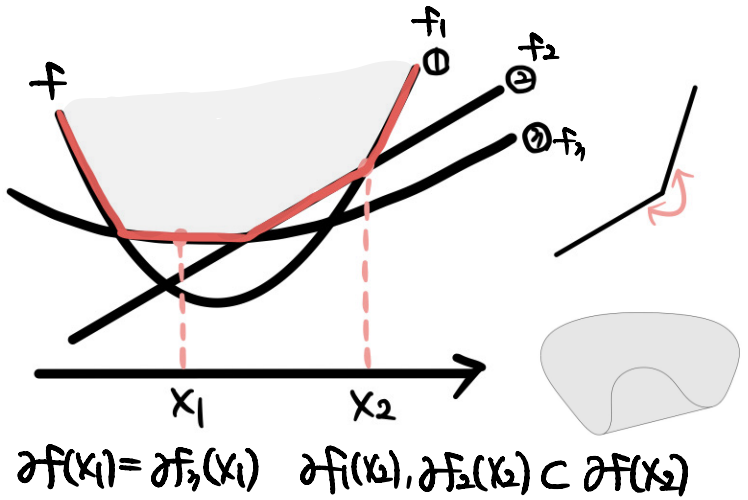
다음은 Subdifferential이 다양한 연산과 관련하여 만족하는 성질들이다.

- Nonnegative Scaling $\text{For } \alpha \geq 0,\ \partial(\alpha f)(x) = \alpha \partial f(x).$
- Addition $\partial f(x) = \partial f_1(x) + \cdots + \partial f_m(x).$
- Affine transformation $h(x) = f(Ax + b).\ \text{Then } \partial h(x) = A^T \partial f(Ax + b).$
- Pointwise Maximum

  $f(x) = \max_{i=1,\ldots,m} f_i(x)$ 에 대해, ∂f_i 들과 ∂f 의 관계는…

  $$\partial f(x) = \mathbf{Co} \cup \{\partial f_i(x) \mid f_i(x) = f(x)\}$$

  즉 점 x에서 active한(=점 x에서 f(x)값을 갖는) 함수들의 subdifferential의 합집합에 대한 convex hull



2

Subgradient method

Subgradient method의 기본 형태와 특징

2.2 Basic Subgradient method

What is Subgradient method?

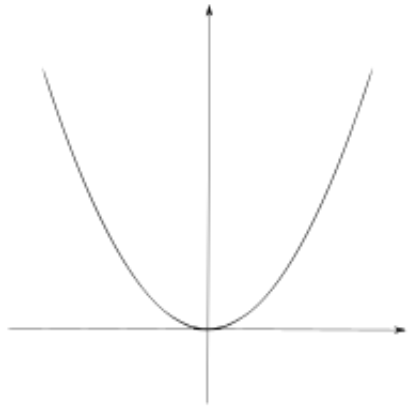
제약조건이 없는 convex function을 최소화하는 가장 기본적인 경우, 다음의 iteration이 사용됨

$$x^{(k+1)} = x^{(k)} - \alpha_k g^{(k)}$$

$x^{(k+1)}$: k번째 iterate, α_k: Positive한 step size, $g^{(k)}$: x(k)에서의 아무 subgradient $g^{(k)} \in \partial f(x^{(k)})$

만약 해당 점에서 f가 미분 가능하다면? g의 선택지가 $\nabla f(x^{(k)})$ 밖에 없어지므로 step size 빼고는 gradient method와 동일

(remind) Gradient method's iteration



$x_{i+1} = x_i -$ (이동거리 x 기울기의 부호)
이때, gradient값은 극솟값에 가까울수록 작아지므로 이동거리를 gradient의 크기와 비례하도록 설정

-> Gradient 직접 이용하되, 이동거리를 사용자가 적절하게 조절할 수 있도록 상수를 추가

$$x^{(k+1)} = x^{(k)} + \alpha_k \nabla f(x_i)$$

2.2 Basic Subgradient method

Subgradient method is not a descent method

Subgradient method, however…

k+1번째 iterate의 함숫값이 k번째 iterate의 함숫값보다 극솟값에 가까움을 보장해주지 못한다! (objective function 증가 가능)

(note) Definition of subgradient $f(z) \geq f(x) + g^T(z - x)$ $f(x^{(k+1)}) \geq f(x^{(k)}) + g^{(k)T}(x^{(k+1)} - x^{(k)})$

(증명) g(k)는 x(k)에서의 subgradient이므로 z 자리에 x(k+1)을, x 자리에 x(k)를 대입하여 식 정리

$$
\begin{aligned}
f(x^{(k+1)}) &= f(x^{(k)} - \alpha_k g^{(k)}) \\
&\geq f(x^{(k)}) + g^{(k)T}(x^{(k)} - \alpha_k g^{(k)} - x^{(k)}) \\
&= f(x^{(k)}) + g^{(k)T}(-\alpha_k g^{(k)}) \\
&= f(x^{(k)}) - \alpha ||g^k||_2^2
\end{aligned}
$$

k+1번째 함숫값은 때때로 k번째 함숫값보다 크거나 같은 값을 가질 수 있음

$f(x^{(k+1)}) \geq f(x^{(k)}) - \alpha \|g^k\|_2^2$

따라서 매 스텝마다 현재의 f 값과 바로 이전의 f 값 중 무엇이 더 작은 지 비교 필요

즉 k번의 iteration을 마친 후에는 k개의 f값 중 무엇이 가장 작은 지 재차 판별하는 과정 필요! 그러므로…

$f_{best}^{(k)} = min\{f(x^{(1)}), ..., f(x^{(k)})\}$ 가 우리가 찾고자 하는 f값이 됨

2.2 Basic Subgradient method

How to choose the step size?

Gradient descent에서와는 달리 subgradient method에서는 step size를 미리 설정해야함

- *Constant step size.* $\alpha_k = \alpha$, where α is positive constant independent with k
- *Constant step length.* $\alpha_k = \gamma / ||g^{(k)}||_2, \ where \ \gamma > 0. \ ||x^{(k+1)} - x^{(k)}||_2 = \gamma$
- *Square summable but not summable.* $\alpha_k \geq 0, \ \Sigma_{k=1}^{\infty} \alpha_k^2 < \infty, \ \Sigma_{k=1}^{\infty} \alpha_k = \infty$
- *Nonsummable diminishing.* $\alpha_k \geq 0, \ \lim_{k \to \infty} \alpha_k = 0, \ \Sigma_{k=1}^{\infty} \alpha_k = \infty$
- *Nonsummable diminishing step lengths.*
  $\alpha_k = \gamma / ||g^{(k)}||_2, \ where \ \gamma_k > 0, \ \lim_{k \to \infty} \gamma_k = 0, \ \Sigma_{k=1}^{\infty} \gamma_k = \infty$

교안에는 가장 기본적인 다섯개의 방법이 실려 있으며, 옆의 제약을 만족하는 상수 or 식을 자율적으로 선정하면 됨

(요점) Subgradient method의 step size는 알고리즘이 돌아가기 전 결정되어 있으며
알고리즘이 돌아가는 동안 도출되는 그 어떤 데이터와도 무관하다.

수렴? 1, 2번째의 경우 ϵ-suboptimal point를 찾는 것이 보장되며, $lim_{k \to \infty} f_{best}^{(k)} - f^* < \epsilon$
3, 4, 5번째에서는 알고리즘이 optimal value로 수렴함이 보장된다. $lim_{k \to \infty} f(x^{(k)}) = f^*$

2.3 Convergence Proof

Basic assumption and inequality

Subgradient method의 수렴성을 논하기 위해 필요한 기본 가정

- f에 대한 minimizer인 x^* 가 존재한다.
- Subgradient들의 norm에는 유계가 있다. 즉 모든 k에 대해 $||g^{(k)}||_2 \leq G$ 인 G가 존재한다.
- Initial point와 optimal set 사이의 거리에 대한 상계가 알려져 있다. 즉 $R \geq ||x^{(1)} - x^*||$ 을 안다.

주어진 상수 R, G, alpha를 사용한 다음과 같은 부등식 도출 가능

$$f_{\text{best}}^{(k)} - f^\star \leq \frac{R^2 + G^2 \sum_{i=1}^k \alpha_i^2}{2 \sum_{i=1}^k \alpha_i}.$$

사용할 step size를 대입한 후 수렴성을 보여주면 됨

Constant step size. When $\alpha_k = \alpha$, we have

$$f_{\text{best}}^{(k)} - f^\star \leq \frac{R^2 + G^2\alpha^2 k}{2\alpha k}.$$

The righthand side converges to $G^2\alpha/2$ as $k \to \infty$. Thus, for the subgradient method with fixed step size α, $f_{\text{best}}^{(k)}$ converges to within $G^2\alpha/2$ of optimal. We also find that $f(x^{(k)}) - f^\star \leq G^2\alpha$ within at most $R^2/(G^2\alpha^2)$ steps.

Square summable but not summable. Now suppose

$$\|\alpha\|_2^2 = \sum_{k=1}^\infty \alpha_k^2 < \infty, \qquad \sum_{k=1}^\infty \alpha_k = \infty.$$

Then we have

$$f_{\text{best}}^{(k)} - f^\star \leq \frac{R^2 + G^2\|\alpha\|_2^2}{2\sum_{i=1}^k \alpha_i},$$

which converges to zero as $k \to \infty$, since the numerator converges to $R^2 + G^2\|\alpha\|_2^2$, and the denominator grows without bound. Thus, the subgradient method converges (in the sense $f_{\text{best}}^{(k)} \to f^\star$).

2.3 Convergence Proof

How to get basic inequality

Step ①

$$\begin{aligned}\|x^{(k+1)}-x^\star\|_2^2 &= \|x^{(k)}-\alpha_k g^{(k)}-x^\star\|_2^2 \quad \text{(풀어쓰기)}\\ &\downarrow\\ &= \|x^{(k)}-x^\star\|_2^2-2\alpha_k g^{(k)T}(x^{(k)}-x^\star)+\alpha_k^2\|g^{(k)}\|_2^2\\ &\le \|x^{(k)}-x^\star\|_2^2-2\alpha_k(f(x^{(k)})-f^\star)+\alpha_k^2\|g^{(k)}\|_2^2,\end{aligned}$$

subgradient의 정의 이용

$f(x^\star)\ge f(x^{(k)})+g^{(k)T}(x^\star-x^{(k)})$

$f(x^{(k)})-f^\star\le g^{(k)T}(x^{(k)}-x^\star)$

→ $\|x^{(k+1)}-x^\star\|_2^2\le\|x^{(k)}-x^\star\|_2^2-2\alpha_k(f(x^{(k)})-f^\star)$ ---ⓐ

Step ②

ⓐ의 k 자리에 $1,\dots,n$ 넣어서 전개 → 식들의 양변 더해서 값 얻기

$$\begin{aligned}\|x^{(2)}-x^\star\|_2^2 &\le \|x^{(1)}-x^\star\|_2^2-2\alpha_1(f(x^{(1)})-f^\star)+\alpha_1^2\|g^{(1)}\|_2^2\\ \|x^{(3)}-x^\star\|_2^2 &\le \|x^{(2)}-x^\star\|_2^2-2\alpha_2(f(x^{(2)})-f^\star)+\alpha_2^2\|g^{(2)}\|_2^2\\ \|x^{(4)}-x^\star\|_2^2 &\le \|x^{(3)}-x^\star\|_2^2-2\alpha_2(f(x^{(3)})-f^\star)+\alpha_2^2\|g^{(3)}\|_2^2\\ &\dots\\ \|x^{(n+1)}-x^\star\|_2^2 &\le \|x^{(n)}-x^\star\|_2^2-2\alpha_2(f(x^{(n)})-f^\star)+\alpha_2^2\|g^{(n)}\|_2^2\end{aligned}$$

→ $\|x^{(n+1)}-x^\star\|_2^2\le\|x^{(1)}-x^\star\|_2^2-2\alpha_2\sum_{i=1}^n(f(x^{(i)})-f^\star)+\alpha_2^2\sum_{i=1}^n\|g^{(i)}\|_2^2$ ---ⓑ

Step ③

↳ 처음 가정

Using $\|x^{(k+1)}-x^\star\|_2^2\ge 0$ and $\|x^{(1)}-x^\star\|_2\le R$ we have

$$2\sum_{i=1}^k\alpha_i(f(x^{(i)})-f^\star)\le R^2+\sum_{i=1}^k\alpha_i^2\|g^{(i)}\|_2^2.$$ ---ⓒ

→ $0\le\|x^{(k+1)}-x^\star\|_2^2\le\|x^{(1)}-x^\star\|_2^2-2\sum_{i=1}^k\alpha_i(f(x^{(i)})-f^\star)+\sum_{i=1}^k\alpha_i^2\|g^{(i)}\|_2^2\le R^2-2\sum_{i=1}^k\alpha_i(f(x^{(i)})-f^\star)+\sum_{i=1}^k\alpha_i^2\|g^{(i)}\|_2^2$

$0\le R^2-2\sum_{i=1}^k\alpha_i(f(x^{(i)})-f^\star)+\sum_{i=1}^k\alpha_i^2\|g^{(i)}\|_2^2$ → $2\sum_{i=1}^k\alpha_i(f(x^{(i)})-f^\star)\le R^2+\sum_{i=1}^k\alpha_i^2\|g^{(i)}\|_2^2$

Step ④

Combining this with

$$\sum_{i=1}^k\alpha_i(f(x^{(i)})-f^\star)\ge\left(\sum_{i=1}^k\alpha_i\right)\min_{i=1,\dots,k}(f(x^{(i)})-f^\star)=\left(\sum_{i=1}^k\alpha_i\right)(f_{\text{best}}^{(k)}-f^\star),$$

→ $\sum_{i=1}^k\alpha_i(f(x^{(i)})-f^\star)\ge\sum_{i=1}^k\alpha_i\min_{i=1,\dots,k}(f(x^{(i)})-f^\star)=\left(\sum_{i=1}^k\alpha_i\right)\min_{i=1,\dots,k}(f(x^{(i)})-f^\star)=\left(\sum_{i=1}^k\alpha_i\right)(f_{\text{best}}^{(k)}-f^\star)$

we have the inequality

$$f_{\text{best}}^{(k)}-f^\star=\min_{i=1,\dots,k}f(x^{(i)})-f^\star\le\frac{R^2+\sum_{i=1}^k\alpha_i^2\|g^{(i)}\|_2^2}{2\sum_{i=1}^k\alpha_i}.$$

→ $2\sum_{i=1}^k\alpha_i(f(x^{(i)})-f^\star)\le R^2+\sum_{i=1}^k\alpha_i^2\|g^{(i)}\|_2^2$

$2\left(\sum_{i=1}^k\alpha_i\right)(f_{\text{best}}^{(k)}-f^\star)\le 2\sum_{i=1}^k\alpha_i(f(x^{(i)})-f^\star)\le R^2+\sum_{i=1}^k\alpha_i^2\|g^{(i)}\|_2^2$

$2\left(\sum_{i=1}^k\alpha_i\right)(f_{\text{best}}^{(k)}-f^\star)\le R^2+\sum_{i=1}^k\alpha_i^2\|g^{(i)}\|_2^2$ → $f_{\text{best}}^{(k)}-f^\star\le\frac{R^2+\sum_{i=1}^k\alpha_i^2\|g^{(i)}\|_2^2}{2\sum_{i=1}^k\alpha_i}$ ---ⓓ

Step ⑤

↳ 처음 가정

Finally, using the assumption $\|g^{(k)}\|_2\le G$, we obtain the basic inequality

$$f_{\text{best}}^{(k)}-f^\star\le\frac{R^2+G^2\sum_{i=1}^k\alpha_i^2}{2\sum_{i=1}^k\alpha_i}.$$

From this inequality we can read off various convergence results.

→ $f_{\text{best}}^{(k)}-f^\star\le\frac{R^2+\sum_{i=1}^k\alpha_i^2\|g^{(i)}\|_2^2}{2\sum_{i=1}^k\alpha_i}\le\frac{R^2+G^2\sum_{i=1}^k\alpha_i^2}{2\sum_{i=1}^k\alpha_i}$, 따라서 $f_{\text{best}}^{(k)}-f^\star\le\frac{R^2+G^2\sum_{i=1}^k\alpha_i^2}{2\sum_{i=1}^k\alpha_i}$

2.3 Convergence Proof

Basic convergence analysis

$$f_{\text{best}}^{(k)} - f^\star \leq \frac{R^2 + G^2 \sum_{i=1}^{k} \alpha_i^2}{2 \sum_{i=1}^{k} \alpha_i}.$$

- *Constant step size.* $\alpha_k = \alpha$, where α is positive constant independent with k

  → converges to $G^2\alpha/2$-suboptimal ($f_{best}^{(k)} - f^* \leq G^2\alpha/2$)
- *Constant step length.* $\alpha_k = \gamma / ||g^{(k)}||_2, \ where \ \gamma > 0. \ ||x^{(k+1)} - x^{(k)}||_2 = \gamma$

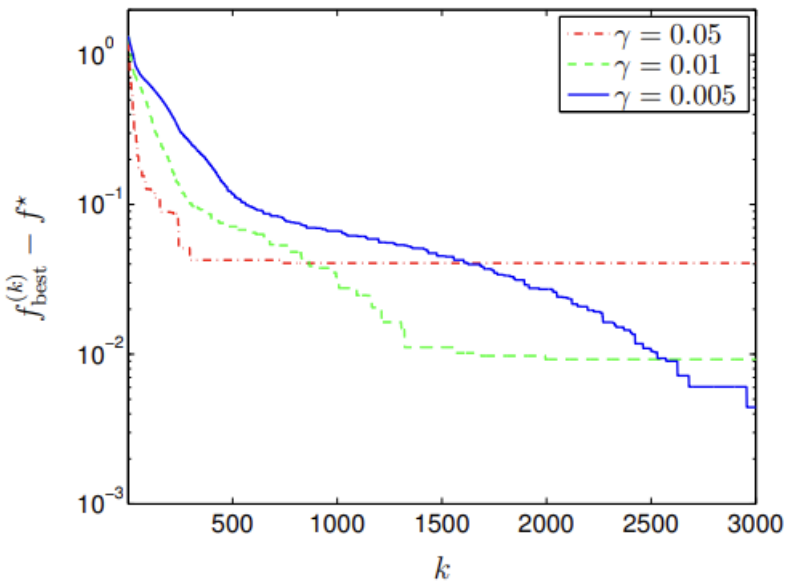
  → converges to $G\gamma/2$-suboptimal ($f_{best}^{(k)} - f^* \leq G\gamma/2$)
- *Square summable but not summable.* $\alpha_k \geq 0, \ \Sigma_{k=1}^{\infty}\alpha_k^2 < \infty, \ \Sigma_{k=1}^{\infty}\alpha_k = \infty$

  → converges. ($lim_{k\to\infty} f(x^{(k)}) = f^*$)
- *Nonsummable diminishing.* $\alpha_k \geq 0, \ \lim_{k\to\infty} \alpha_k = 0, \ \Sigma_{k=1}^{\infty}\alpha_k = \infty$ → converges.
- *Nonsummable diminishing step lengths.* $\alpha_k = \gamma / ||g^{(k)}||_2, \ where \ \gamma_k > 0, \ \lim_{k\to\infty} \gamma_k = 0, \ \Sigma_{k=1}^{\infty}\gamma_k = \infty$ → converges.

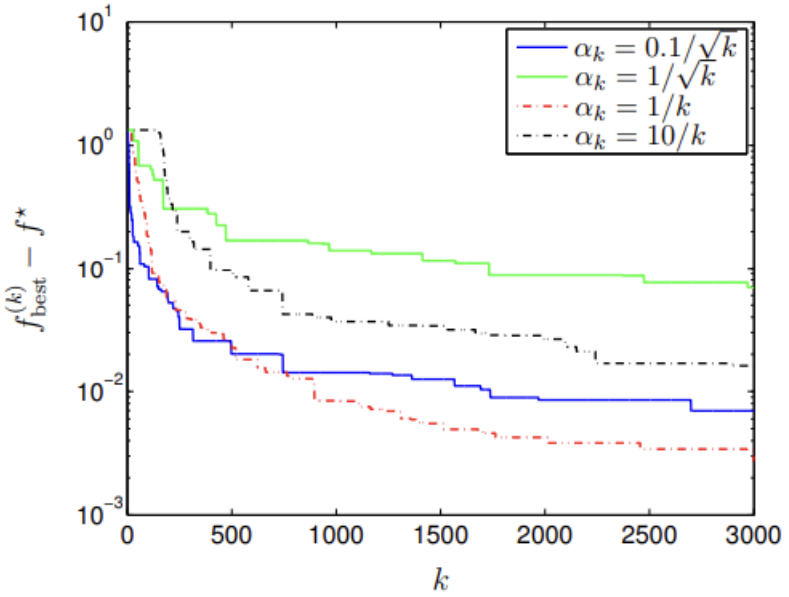
2.3 Convergence Proof

Numeric example

Constant step length rule



Nonsummable rule



수렴하기 위해 굉장히 오래 걸리나, 범용성(미분 불가능한 함수에 적용 가능)의 측면에서 이점이 있음

- 어떻게 제약조건이 있는 문제에도 subgradient method를 적용할 수 있을지? (범용성 증대)
- 어떻게 느린 성능 문제를 완화할 수 있을 지? (단점 보완) -> proximal gradient descent

4. Polyak Step's Length

2018122037 전상후

Polyak Step

- Recall basic subgradient method

$$x^{(k+1)} = x^{(k)} - \alpha_k g^{(k)}$$

- Polyak k-th step size α_k

$$\|x^{(k+1)} - x^*\|_2^2 \leq \|x^{(k)} - x^*\|_2^2 - 2\alpha_k(f(x^{(k)} - f^*) + a_k^2\|g^{(k)}\|_2^2$$

$$\alpha_k = \frac{f(x^{(k)}) - f^*}{\|g^{(k)}\|_2^2}$$

- Used when optimal value f^* is known (or estimated)

Polyak Step

- Analyze convergence

recall $2\sum_{i=1}^{k}\alpha_i(f(x^{(i)})-f^\star)\le R^2+\sum_{i=1}^{k}\alpha_i^2\|g^{(i)}\|_2^2.$

  - Plug in Polyak step size

$$2\sum_{i=1}^{k}\frac{(f(x^{(i)})-f^\star)^2}{\|g^{(i)}\|_2^2}\le R^2+\sum_{i=1}^{k}\frac{(f(x^{(i)})-f^\star)^2}{\|g^{(i)}\|_2^2},$$

$$\sum_{i=1}^{k}\frac{(f(x^{(i)})-f^\star)^2}{\|g^{(i)}\|_2^2}\le R^2.$$

  - Using Lipschitz constant G $(\| g^{(i)} \|_2 \le G)$

$$\sum_{i=1}^{k}(f(x^{(i)})-f^\star)^2\le R^2G^2. \longrightarrow f(x^{(k)})\to f^*$$

Polyak Step – estimated f^*

- Polyak step size α_k (estimation ver.)

$$\alpha_k = \frac{f(x^{(k)}) - f_{\text{best}}^{(k)} + \gamma_k}{\|g^{(k)}\|_2^2}.$$

Estimate f^* with $f_{best} - \gamma_k$,

$\gamma_k > 0 \; and \; r_k \to 0, \quad \sum_{k=1}^{\infty} \gamma_k = \infty$

γ_k = estimate of suboptimality

$$R^2 \geq \sum_{i=1}^{k} \frac{(f(x^{(i)}) - f_{\text{best}}^{(i)} + \gamma_i)((f(x^{(i)}) - f^*) + (f_{\text{best}}^{(i)} - f^*) - \gamma_i)}{\|g^{(i)}\|_2^2}$$

Polyak Step – estimated f^*

- Convergence Proof
  - Suppose $f_{best}^{(k)} - f^* \geq \epsilon > 0$, then $f\left(x^{(i)}\right) - f^* \geq \epsilon$
  - For $i \geq$ N, $\gamma_i \leq \epsilon$

$$(f(x^{(i)}) - f^*) + (f_{\text{best}}^{(i)} - f^*) - \gamma_i \geq \epsilon. \qquad f(x^{(i)}) - f_{\text{best}}^{(i)} + \gamma_i \geq \gamma_i$$

  - S = sum from i = 1 ~ N-1

$$\sum_{i=N}^{k} \frac{(f(x^{(i)}) - f_{\text{best}}^{(i)} + \gamma_i)((f(x^{(i)}) - f^*) + (f_{\text{best}}^{(i)} - f^*) - \gamma_i)}{\|g^{(i)}\|_2^2} \leq R^2 - S.$$

  - $\left\|g^{(i)}\right\|_2 \leq G$

$$(\epsilon/G^2) \sum_{i=N}^{k} \gamma_i \leq R^2 - S.$$

→ k is finite → converges

Polyak Step – example

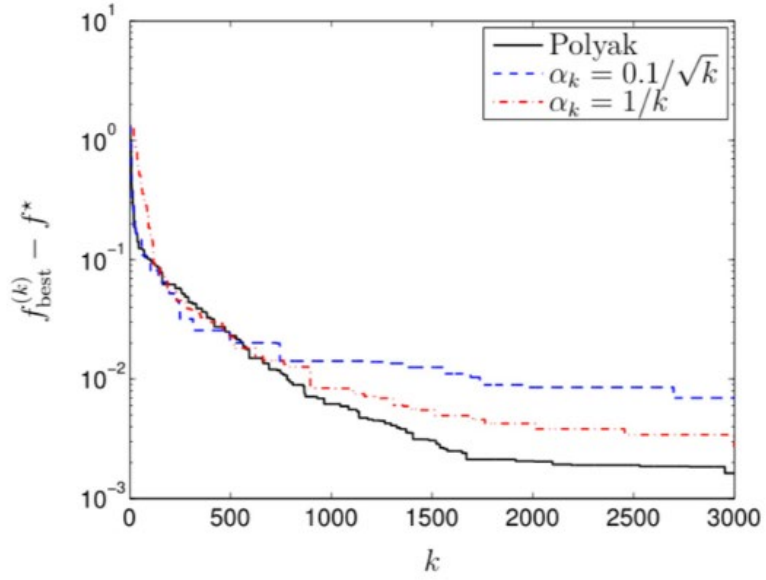


Figure 3: The value of $f_{\text{best}}^{(k)} - f^\star$ versus iteration number k, for the subgradient method with Polyak's step size (solid black line) and the subgradient methods with diminishing step sizes considered in the previous example (dashed lines).

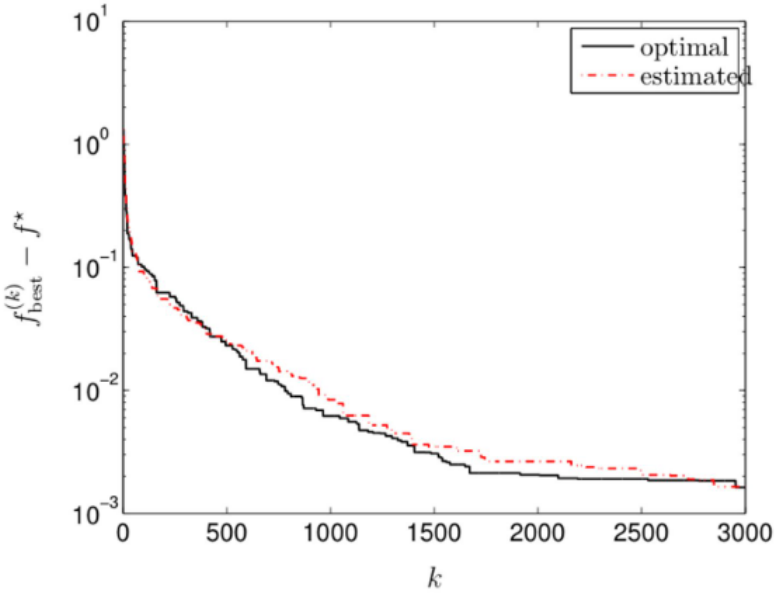
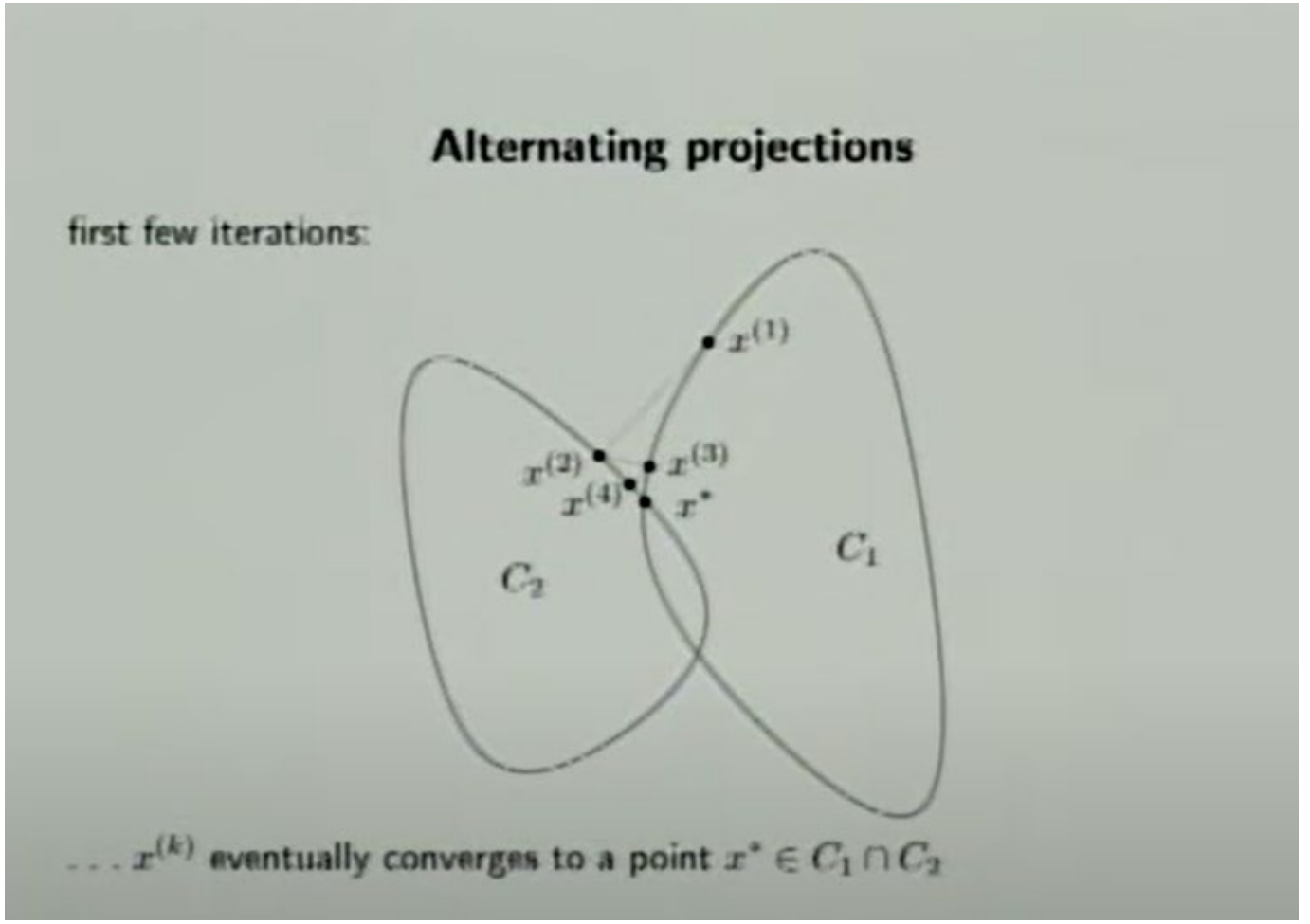


Figure 4: The value of $f_{\text{best}}^{(k)} - f^\star$ versus iteration number k, for the subgradient method with Polyak's step size (solid black line) and the estimated optimal step size (dashed red line).

5. Alternating projections

Alternating projection



Reference: Online lecture of Convex optimization II, Stanford University

Finding intersection point

- We want to find a point in $C = C_1 \cap \cdots \cap C_m$,
- $C_1 \sim C_m$: closed, convex, nonempty

→ minimize $f(x) = max\{\mathbf{dist}(x, C_1), ..., \mathbf{dist}(x, C_m)\}$

- $f^* = 0$ → known optimal → can use Polyak step

$$g = \nabla \mathbf{dist}(x, C_j) = \frac{x - \Pi_{C_j}(x)}{\|x - \Pi_{C_j}(x)\|_2}$$

- $\Pi_{C_j}(x)$ = Euclidean projection onto C_j
- $\|g\|_2 = 1$ (L2 norm of L2 norm = L2 norm)

Subgradient of L2 norm

$$\delta\|x\|_2 = \frac{x}{\|x\|_2} \quad for\ x \neq 0$$

Finding Intersection point

- Subgradient algorithm update

$$x^{(k+1)} = x^{(k)} - \alpha_k g^{(k)}$$

$$\alpha_k = \frac{f(x^{(k)}) - f^*}{\|g^{(k)}\|_2^2}$$

- Using Polyak step length

$$f(x^{(k)}) = \mathbf{dist}(x^{(k)}, C_j) = \|x - \Pi_{C_j}(x)\|_2$$

$$\begin{aligned} x^{(k+1)} &= x^{(k)} - \alpha_k g^{(k)} \\ &= x^{(k)} - f(x^{(k)}) \frac{x^{(k)} - \Pi_{C_j}(x^{(k)})}{\|x^{(k)} - \Pi_{C_j}(x^{(k)})\|_2} \\ &= \Pi_{C_j}(x^{(k)}). \end{aligned}$$

→ Projecting current point to the farthest set

6. Projected subgradient method

Projected subgradient method

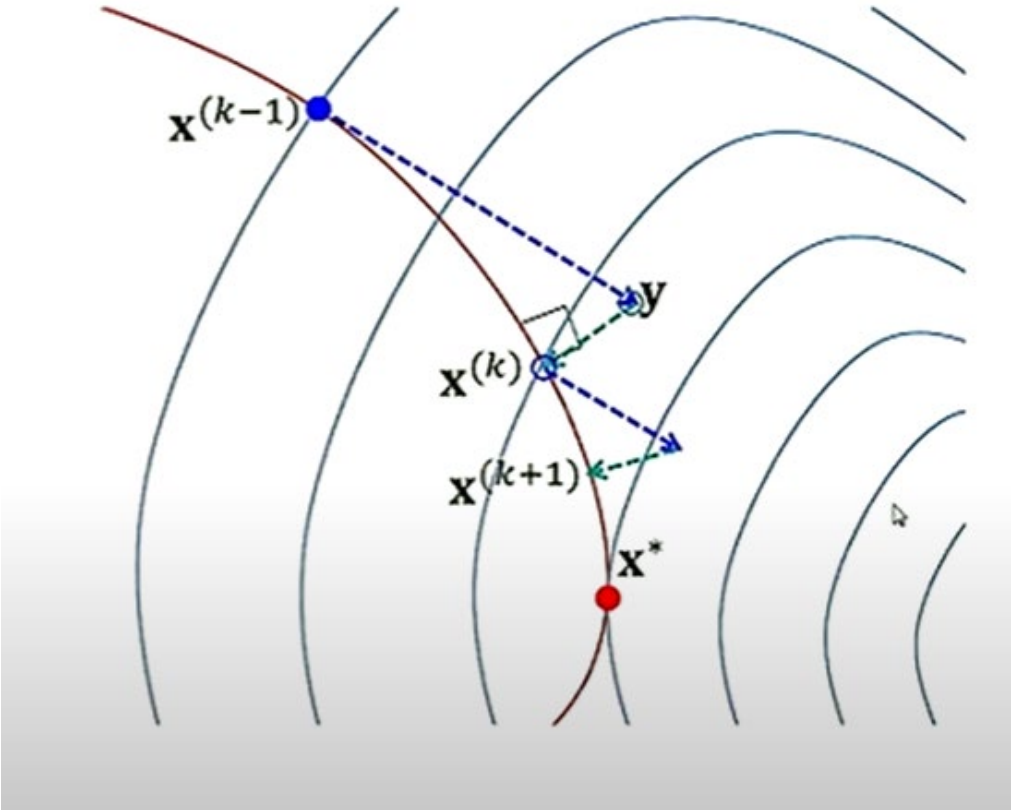
- solving constrained convex optimization problem (C = convex set)

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in C \end{array}$$

- Projected subgradient method (Π = Euclidean projection on C)

$$x^{(k+1)} = \Pi(x^{(k)} - \alpha_k g^{(k)})$$

Projected subgradient method



Reference: https://www.youtube.com/watch?v=x4IyNENx8L4

Convergence Proof

- Let $z^{(k+1)} = x^{(k)} - \alpha_k g^{(k)}$
  - $$\begin{aligned} \|z^{(k+1)} - x^\star\|_2^2 &= \|x^{(k)} - \alpha_k g^{(k)} - x^\star\|_2^2 \\ &= \|x^{(k)} - x^\star\|_2^2 - 2\alpha_k g^{(k)T}(x^{(k)} - x^\star) + \alpha_k^2 \|g^{(k)}\|_2^2 \\ &\leq \|x^{(k)} - x^\star\|_2^2 - 2\alpha_k (f(x^{(k)}) - f^\star) + \alpha_k^2 \|g^{(k)}\|_2^2. \end{aligned}$$

    Basic inequality (3.2)
  - $\|x^{(k+1)} - x^*\|_2 = \|\Pi(z^{(k+1)}) - x^*\|_2 \leq \|z^{(k+1)} - x^*\|_2$

    Recall the meaning of Π
  - $\|x^{(k+1)} - x^*\|_2^2 \leq \|x^{(k)} - x^*\|_2^2 - 2\alpha_k (f(x^{(k)} - f^*) + a_k^2 \|g^{(k)}\|_2^2$

Alternative Expression

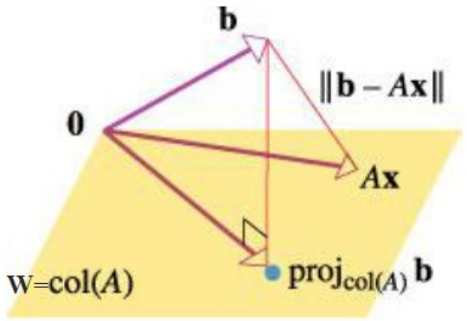
- If C = affine $(C = \{x \mid Ax = b\})$, A = fat, full rank

$$\Pi(z) = z - A^T(AA^T)^{-1}(Az - b).$$

In this case, we can simplify the subgradient update to

$$x^{(k+1)} = x^{(k)} - \alpha_k(I - A^T(AA^T)^{-1}A)g^{(k)},$$

Alternative Expression



‖b – Ax‖는
Ax = proj$_{col(A)}$ b일 때
최소이다.

$$A^T(b - Ax) = 0$$

Ax = proj(b) on col(A)

$$x = (A^T A)^{-1} A^T b$$

$$\hat{b} = A(A^T A)^{-1} A^T b$$

$$b - \hat{b} = (I - A(A^T A)^{-1} A^T) b$$

null(A)로의 projection ↔ col(A^T)로의 orthogonal projection

$$x^{(k+1)} = x^{(k)} - \alpha_k (I - H^T) g^{(k)}$$

Alternative Expression

Projection onto Hyperplane

$$\mathbf{x} = \text{proj}_{\mathbf{Ax=b}}\mathbf{y} = \underset{\mathbf{x}:\,\mathbf{Ax=b}}{\text{argmin}}(\mathbf{y}-\mathbf{x})^{\mathrm{T}}(\mathbf{y}-\mathbf{x})$$

COP

Minimize w.r.t. $\mathbf{x}$:

$$\frac{1}{2}(\mathbf{y}-\mathbf{x})^{\mathrm{T}}(\mathbf{y}-\mathbf{x})$$

Subject to:

$$\mathbf{Ax}-\mathbf{b}=\mathbf{0}$$

$$\mathcal{L}(\mathbf{x},\boldsymbol{\mu}) = \frac{1}{2}\left(\mathbf{y}^{\mathrm{T}}\mathbf{y} - 2\mathbf{x}^{\mathrm{T}}\mathbf{y} + \mathbf{x}^{\mathrm{T}}\mathbf{x}\right) + \boldsymbol{\mu}^{\mathrm{T}}(\mathbf{Ax}-\mathbf{b})$$

$$\nabla\mathcal{L}(\mathbf{x},\boldsymbol{\mu}) = \mathbf{y}-\mathbf{x}+\mathbf{A}^{\mathrm{T}}\boldsymbol{\mu} = \mathbf{0}$$

$$\Rightarrow \mathbf{x} = \mathbf{y}-\mathbf{A}^{\mathrm{T}}\boldsymbol{\mu}$$

$$\Rightarrow \mathbf{Ax} = \mathbf{Ay}-\mathbf{AA}^{\mathrm{T}}\boldsymbol{\mu}$$

$$\Rightarrow \mathbf{b} = \mathbf{Ay}-\mathbf{AA}^{\mathrm{T}}\boldsymbol{\mu}$$

$$\Rightarrow \mathbf{AA}^{\mathrm{T}}\boldsymbol{\mu} = \mathbf{Ay}-\mathbf{b}$$

$$\Rightarrow \boldsymbol{\mu} = \left(\mathbf{AA}^{\mathrm{T}}\right)^{-1}(\mathbf{Ay}-\mathbf{b})$$

$$\Rightarrow \mathbf{x} = \mathbf{y}-\mathbf{A}^{\mathrm{T}}\left(\mathbf{AA}^{\mathrm{T}}\right)^{-1}(\mathbf{Ay}-\mathbf{b})$$

Reference: https://www.youtube.com/watch?v=x4IyNENx8L4

Example 6.1 (using alternative expression)

- Least $l_1 - norm$ problem (Assume A = fat,full rank)

$$\begin{array}{ll} \text{minimize} & \| x \|_1 \\ \text{subject to} & Ax = b \end{array}$$

- Projected subgradient update (subgradient g = **sign**(x))

$$x^{(k+1)} = x^{(k)} - \alpha_k (I - A^T (AA^T)^{-1} A)\, \mathbf{sign}(x^{(k)}).$$

Example 6.1 (using alternative expression)

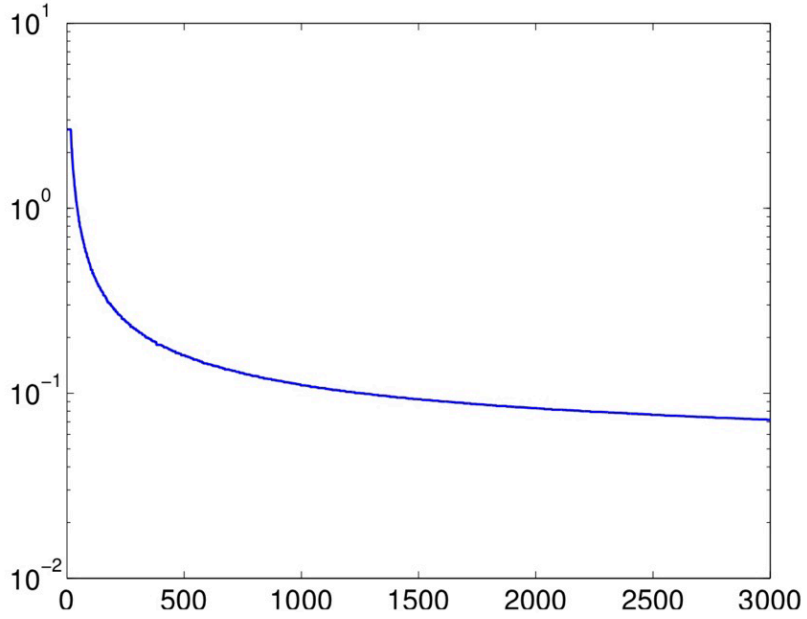


Figure 8: The value of $f_{\text{best}}^{(k)} - f^\star$ versus iteration number k, for the subgradient method with the Polyak estimated step size rule $\gamma_k = 100/k$.

Example 6.2

- Convex primal problem

$$\begin{array}{lll} \text{minimize} & f_0(x) & \\ \text{subject to} & f_i(x) \leq 0, & i = 1, \ldots, m \end{array}$$

- Lagrangian and its dual function
  - Assume Lagrangian has a unique minimizer $x^*(\lambda)$

$$L(x, \lambda) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) \qquad g(\lambda) = \inf_x L(x, \lambda) = f_0(x^\star(\lambda)) + \sum_{i=1}^{m} \lambda_i f_i(x^\star(\lambda))$$

Example 6.2

- Dual problem

$$\begin{aligned} \text{maximize} \quad & g(\lambda) \\ \text{subject to} \quad & \lambda \geq 0 \end{aligned}$$

- Strong duality

Check previous session about Duality.

  - Assume that Slater's condition holds
  - Primal problem = convex → strong duality = zero duality gap
  - Can obtain x^* by finding dual optimal point λ^*

Example 6.2

- Subgradient method

$$\lambda^{(k+1)} = \left(\lambda^{(k)} - \alpha_k h\right)_+, \qquad h \in \partial(-g)(\lambda^{(k)}).$$

- Gradient of -g

$$-f_0(x^\star(\lambda)) - \sum_{i=1}^{m} \lambda_i f_i(x^\star(\lambda)),$$

which has gradient (with respect to λ)

$$h = -(f_1(x^\star(\lambda)), \ldots, f_m(x^\star(\lambda)) \in \partial(-g)(\lambda).$$

Example 6.2

- Subgradient method

The projected subgradient method for the dual has the form

$$\begin{aligned} x^{(k)} &= \operatorname*{argmin}_x \left(f_0(x) + \sum_{i=1}^{m} \lambda_i^{(k)} f_i(x) \right) \\ \lambda_i^{(k+1)} &= \left(\lambda_i^{(k)} + \alpha_k f_i(x^{(k)}) \right)_+ . \end{aligned}$$

Interpretation

- λ_i = Cost ➔ minimizing total cost!

$$
\begin{aligned}
x^{(k)} &= \operatorname*{argmin}_x \left(f_0(x) + \sum_{i=1}^{m} \lambda_i^{(k)} f_i(x) \right) \\
\lambda_i^{(k+1)} &= \left(\lambda_i^{(k)} + \alpha_k f_i(x^{(k)}) \right)_+ .
\end{aligned}
$$

- Meaning of h? Stick to the constraint!

$$h = -(f_1(x^\star(\lambda)), \ldots, f_m(x^\star(\lambda)) \in \partial(-g)(\lambda).$$

positive = ith constraint satisfied → decrease(=or retain) price
negative = ith constraint violated → increase price

7. Subgradient method for constrained optimization

Constrained optimization

- Inequality constrained problem

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \qquad i = 1, \ldots, m \end{array}$$

- Algorithm? Same!

$$x^{(k+1)} = x^{(k)} - \alpha_k g^{(k)}$$

- What's the difference? → selecting $g^{(k)}$ regarding constraint!

$$g^{(k)} \in \begin{cases} \partial f_0(x^{(k)}) & f_i(x^{(k)}) \leq 0, \quad i = 1, \ldots, m, \\ \partial f_j(x^{(k)}) & f_j(x^{(k)}) > 0. \end{cases}$$

Example 7.1

- Linear program

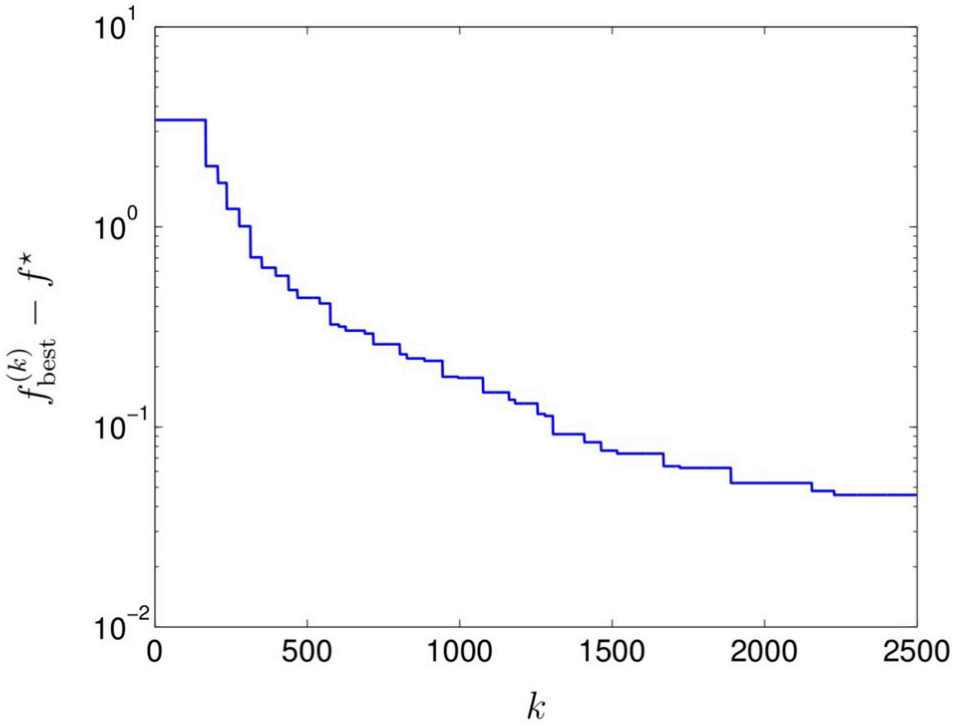
$$\begin{aligned} &\text{minimize} && c^T x \\ &\text{subject to} && \alpha_i^{\mathrm{T}} \mathrm{x} \leq b_i, \qquad i = 1, \ldots, m \end{aligned}$$

- Subgradient g?

  - c in feasible points
  - α_i in infeasible points

→ Subgradients of objective function and ith (violated) constraint

Example 7.1



$$f_{\text{best}}^{(k)} = \min\{f_0(x^{(i)}) \mid x^{(i)} \text{ feasible}, \ i = 1, \ldots, k\}.$$

Figure 10: The value of $f_{\text{best}}^{(k)} - f^\star$ versus the iteration number k. In this case, we use the square summable step size with $\alpha_k = 1/k$ for the optimality update.

Thank you