# Optimization in Machine Learning

## Lecture 19: Proximal, Generalized Gradient Descent and its Acceleration and Projection in GD

Ganesh Ramakrishnan

Department of Computer Science
Dept of CSE, IIT Bombay
https://www.cse.iitb.ac.in/~ganesh

March, 2025

# So far...

- Algorithms for Optimization: First Order and thereafter [Done]
- Accelerated Gradient Descent [Done]
- Stochastic Gradient Descent [Done]
- Accelerated Stochastic [Done]
- Accelerated Stochastic Variants: Hybrid, Adam, Adagrad, RMSProp, *etc.* [Done]
- Generalized/Proximal Gradient Descent
- Constrained Optimization and Projected Gradient Descent

# Outline of next few topics

- Gradient Descent Analysis for Lipschitz Smoothness/Continuity/(Strong) Convexity [Done]
  - Our running Colab Notebook:
- Nestorov's and Polyak's accelerated gradient descent [Done]
- Sub-gradient Descent and Analysis [Done]
- Stochastic Gradient Descent [Done]
- Generalized Gradient Descent [Next]
- Proximal Gradient Descent
- Projected Gradient Descent for Constrained Optimization
- Discrete/Combinatorial Optimization for Subset Selection
- Constrained Optimization, Duality & KKT Conditions
- And so on...more such algorithms

# Generalized Gradient Descent - generalizing (Sub)gradient Descent for Non-Differentiable and Constrained Cases

**Back to representing x as variables (w was used to represent variables only in the case of stochastic algorithms applied to Machine Learning)**

# Generalized Gradient Descent - generalizing (Sub)gradient Descent

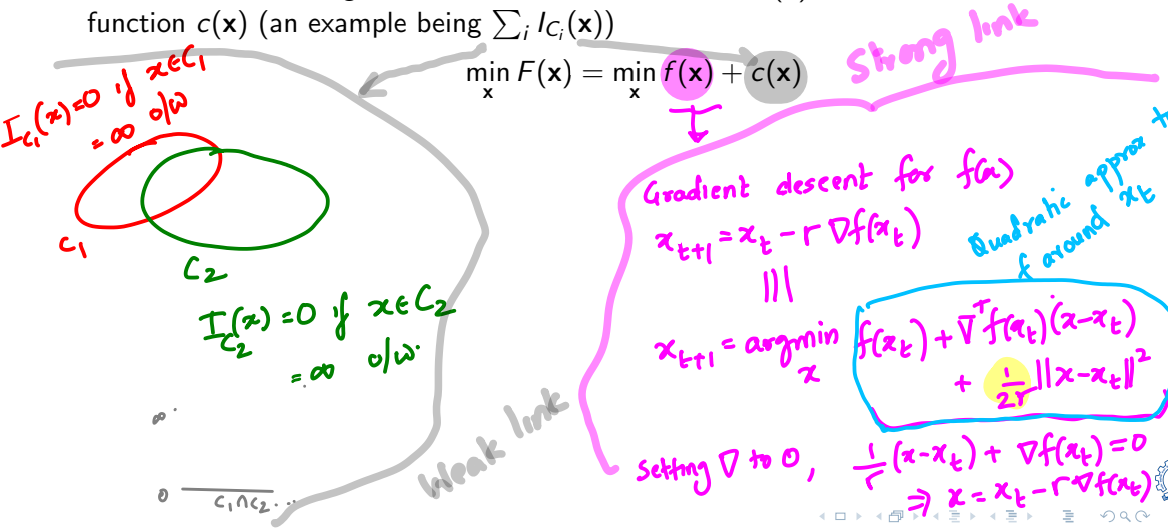
- Consider the following sum of a differentiable function $f(\mathbf{x})$ and a nondifferentiable function $c(\mathbf{x})$ (an example being $\sum_i I_{C_i}(\mathbf{x})$)

$$\min_{\mathbf{x}} F(\mathbf{x}) = \min_{\mathbf{x}} f(\mathbf{x}) + c(\mathbf{x})$$

# Generalized Gradient Descent - generalizing (Sub)gradient Descent

- Consider the following sum of a differentiable function $f(\mathbf{x})$ and a nondifferentiable function $c(\mathbf{x})$ (an example being $\sum_i I_{C_i}(\mathbf{x})$)

$$\min_{\mathbf{x}} F(\mathbf{x}) = \min_{\mathbf{x}} f(\mathbf{x}) + c(\mathbf{x})$$



# Generalized Gradient Descent - generalizing (Sub)gradient Descent

- Consider the following sum of a differentiable function $f(\mathbf{x})$ and a nondifferentiable function $c(\mathbf{x})$ (an example being $\sum_i I_{C_i}(\mathbf{x})$)

$$\min_{\mathbf{x}} F(\mathbf{x}) = \min_{\mathbf{x}} f(\mathbf{x}) + c(\mathbf{x})$$

- Like gradient descent, consider the first order approximation for $f(\mathbf{x})$ around $\mathbf{x}_t$ leaving $c(\mathbf{x})$ alone to obtain the next iterate $\mathbf{x}_{t+1}$:

$$\mathbf{x}_{t+1} = \text{argmin}_{\mathbf{x}} f(\mathbf{x}_t) + \nabla^T f(\mathbf{x}_t)(\mathbf{x} - \mathbf{x}_t) + \frac{1}{2\gamma}||\mathbf{x} - \mathbf{x}_t||^2 + c(\mathbf{x})$$

# Generalized Gradient Descent - generalizing (Sub)gradient Descent

- Consider the following sum of a differentiable function $f(\mathbf{x})$ and a nondifferentiable function $c(\mathbf{x})$ (an example being $\sum_i I_{C_i}(\mathbf{x})$)

$$\min_{\mathbf{x}} F(\mathbf{x}) = \min_{\mathbf{x}} f(\mathbf{x}) + c(\mathbf{x})$$

- Like gradient descent, consider the first order approximation for $f(\mathbf{x})$ around $\mathbf{x}_t$ leaving $c(\mathbf{x})$ alone to obtain the next iterate $\mathbf{x}_{t+1}$:

$$\mathbf{x}_{t+1} = \text{argmin}_{\mathbf{x}} \; f(\mathbf{x}_t) + \nabla^T f(\mathbf{x}_t)(\mathbf{x} - \mathbf{x}_t) + \frac{1}{2\gamma}||\mathbf{x} - \mathbf{x}_t||^2 + c(\mathbf{x})$$

$$\frac{1}{2r}\left[r^2||\nabla f(x_t)||^2 + 2r\nabla^T f(x_t)(x - x_t) + ||x - x_t||^2\right] + c(x)$$

$$= \frac{1}{2r}\left[||x - (x_t - r\nabla f(x_t))||^2\right] + c(x)$$

Say $c(x) = \|x\|_1$

Does this "simplified" formulation based on first order approximation around x_t help?

# Generalized Gradient Descent - generalizing (Sub)gradient Descent

- Consider the following sum of a differentiable function $f(\mathbf{x})$ and a nondifferentiable function $c(\mathbf{x})$ (an example being $\sum_i I_{C_i}(\mathbf{x})$)

$$\min_{\mathbf{x}} F(\mathbf{x}) = \min_{\mathbf{x}} f(\mathbf{x}) + c(\mathbf{x})$$

- Like gradient descent, consider the first order approximation for $f(\mathbf{x})$ around $\mathbf{x}_t$ leaving $c(\mathbf{x})$ alone to obtain the next iterate $\mathbf{x}_{t+1}$:

$$\mathbf{x}_{t+1} = \text{argmin}_{\mathbf{x}} f(\mathbf{x}_t) + \nabla^T f(\mathbf{x}_t)(\mathbf{x} - \mathbf{x}_t) + \frac{1}{2\gamma}||\mathbf{x} - \mathbf{x}_t||^2 + c(\mathbf{x})$$

Grad descent minimizes this approximation exactly as

$$x_{t+1} = x_t - r\nabla f(x_t)$$

# Generalized Gradient Descent - generalizing (Sub)gradient Descent

- Consider the following sum of a differentiable function $f(\mathbf{x})$ and a nondifferentiable function $c(\mathbf{x})$ (an example being $\sum_i I_{C_i}(\mathbf{x})$)

$$\min_{\mathbf{x}} F(\mathbf{x}) = \min_{\mathbf{x}} f(\mathbf{x}) + c(\mathbf{x})$$

- Like gradient descent, consider the first order approximation for $f(\mathbf{x})$ around $\mathbf{x}_t$ leaving $c(\mathbf{x})$ alone to obtain the next iterate $\mathbf{x}_{t+1}$:

$$\mathbf{x}_{t+1} = \text{argmin}_{\mathbf{x}} f(\mathbf{x}_t) + \nabla^T f(\mathbf{x}_t)(\mathbf{x} - \mathbf{x}_t) + \frac{1}{2\gamma}||\mathbf{x} - \mathbf{x}_t||^2 + c(\mathbf{x})$$

- Ignoring $f(\mathbf{x}_t)$ and adding $\frac{\gamma}{2}||\nabla f(\mathbf{x}_t)||^2$ to the objective (without any loss) to complete squares, we obtain $\mathbf{x}_{t+1}$ as:

$$\mathbf{x}_{t+1} = \text{argmin}_{\mathbf{x}} \frac{1}{2\gamma}||\mathbf{x} - (\mathbf{x}_t - \gamma\nabla f(\mathbf{x}_t))||^2 + c(\mathbf{x})$$

- In general, such a step is called a *proximal* step

# Generalized Gradient Descent - generalizing (Sub)gradient Descent

- Consider the following sum of a differentiable function $f(\mathbf{x})$ and a nondifferentiable function $c(\mathbf{x})$ (an example being $\sum_i I_{C_i}(\mathbf{x})$)

$$\min_{\mathbf{x}} F(\mathbf{x}) = \min_{\mathbf{x}} f(\mathbf{x}) + c(\mathbf{x})$$

- Like gradient descent, consider the first order approximation for $f(\mathbf{x})$ around $\mathbf{x}_t$ leaving $c(\mathbf{x})$ alone to obtain the next iterate $\mathbf{x}_{t+1}$:

$$\mathbf{x}_{t+1} = \text{argmin}_{\mathbf{x}} f(\mathbf{x}_t) + \nabla^T f(\mathbf{x}_t)(\mathbf{x} - \mathbf{x}_t) + \frac{1}{2\gamma}||\mathbf{x} - \mathbf{x}_t||^2 + c(\mathbf{x})$$

- Ignoring $f(\mathbf{x}_t)$ and adding $\frac{\gamma}{2}||\nabla f(\mathbf{x}_t)||^2$ to the objective (without any loss) to complete squares, we obtain $\mathbf{x}_{t+1}$ as:

$$\mathbf{x}_{t+1} = \text{argmin}_{\mathbf{x}} \frac{1}{2\gamma}||\mathbf{x} - (\mathbf{x}_t - \gamma\nabla f(\mathbf{x}_t))||^2 + c(\mathbf{x})$$

- In general, such a step is called a *proximal* step

Grad descent minimizes this approximation exactly

# Generalized Gradient Descent - generalizing (Sub)gradient Descent

- Interesting because in many settings, $prox_\gamma(\mathbf{z})$ can be computed efficiently

$$prox_\gamma(\mathbf{z}) = \text{argmin}_{\mathbf{x}} \frac{1}{2\gamma}||\mathbf{x}-\mathbf{z}||^2 + c(\mathbf{x})$$

- Illustration on Lasso: $\mathbf{x}_* = \text{argmin}_{\mathbf{x}} ||\Phi\mathbf{x}-\mathbf{y}||^2 + \lambda\|\mathbf{x}\|_1$. You can successively use $\mathbf{z} = \mathbf{x}_t - \gamma\nabla f(\mathbf{x}_t)$.
-
-
-
-

# Generalized Gradient Descent - generalizing (Sub)gradient Descent

- Interesting because in many settings, $prox_\gamma(\mathbf{z})$ can be computed efficiently

$$prox_\gamma(\mathbf{z}) = \text{argmin}_{\mathbf{x}} \frac{1}{2\gamma}||\mathbf{x}-\mathbf{z}||^2 + c(\mathbf{x})$$

- Illustration on Lasso: $\mathbf{x}_* = \text{argmin}_{\mathbf{x}} ||\Phi\mathbf{x}-\mathbf{y}||^2 + \lambda\|\mathbf{x}\|_1$. You can successively use $\mathbf{z} = \mathbf{x}_t - \gamma\nabla f(\mathbf{x}_t)$.
- 
- 
- 
- 

Step 1 GD for $||\Phi x - y||^2$

$$x_{t+1} = x_t - 2\gamma(\Phi^T\Phi x_t - \Phi^T y)$$

Step 2: Setting $z = x_{t+1}$ solve

$$prox_\gamma(z) = \text{argmin}_x \frac{1}{2\gamma}||x-z||^2 + \lambda\|x\|_1$$

Brings in L1 regularization post-facto

# Generalized Gradient Descent - generalizing (Sub)gradient Descent

- Interesting because in many settings, $prox_\gamma(\mathbf{z})$ can be computed efficiently

$$prox_\gamma(\mathbf{z}) = \text{argmin}_{\mathbf{x}} \frac{1}{2\gamma}||\mathbf{x}-\mathbf{z}||^2 + c(\mathbf{x})$$

- Illustration on Lasso: $\mathbf{x}_* = \text{argmin}_{\mathbf{x}} ||\Phi\mathbf{x}-\mathbf{y}||^2 + \lambda\|\mathbf{x}\|_1$. You can successively use $\mathbf{z} = \mathbf{x}_t - \gamma\nabla f(\mathbf{x}_t)$.

Homework: Recall the Sufficient condition test for simplified lasso and plug it in the proximal operator here to complete our ISTA algorithm

We wrote sufficient condition for min at

$$f(w) = \frac{1}{2}||w-y||^2 + \lambda\|w\|_1$$

$$S_\lambda(y) = \begin{cases} y_i - \lambda & \text{if } y_i \geq \lambda \\ 0 & \text{if } y_i \in [-\lambda, \lambda] \\ y_i + \lambda & \text{if } y_i \leq -\lambda \end{cases}$$

6/80

# Illustration on Lasso

Homework: Recall the Sufficient condition test for simplified lasso and plug it in the proximal operator here to derive the detailed steps of the overall Iterative Soft Thresholding Algorithm (ISTA) for LASSO

Also contemplate the convergence of the ISTA algorithm.

# Illustration on Lasso

**Step 1** GD for $\|\Phi x - y\|^2$

$$x_{t+1} = x_t - 2r(\Phi^T \Phi x_t - \Phi^T y)$$

**Step 2:** Setting $z = x_{t+1}$ solve

$$prox_r(z) = \underset{x}{\text{argmin}} \frac{1}{2r}\|x - z\|^2 + \lambda\|x\|_1 = \underset{x}{\text{argmin}} \frac{1}{2}\|x - z\|^2 + \lambda r\|x\|_1$$

Brings in L1 regularization post-facto

Multiplying entire objective by $r$

$$[prox_r(z)]_i = \begin{cases} z_i - \lambda r & \text{if } z_i \geq \lambda r \\ 0 & \text{if } z_i \in [-\lambda r, \lambda r] \\ z_i + \lambda r & \text{if } z_i \leq -\lambda r \end{cases}$$

[Recap sufficient condition we had derived based on setting a subgradient to 0]

# Iterative Soft Thresholding Algorithm for Solving Lasso

# Proximal Subgradient Descent for Lasso

- Let $\varepsilon(\mathbf{w}) = \|\Phi\mathbf{w} - \mathbf{y}\|_2^2$
- **Proximal Subgradient Descent Algorithm:**
  **Initialization:** Find starting point $\mathbf{w}^{(0)}$ **(Only here, We use superscripts $t$ to represent iterates $\mathbf{w}^t$ so that subscripts $i$ can be used to represent indices $\mathbf{w}_i^t$)**
  - Let $\widehat{\mathbf{w}}^{(\mathbf{t+1})}$ be a next gradient descent iterate for $\mathbf{w}^t$
  - Compute $\mathbf{w}^{(t+1)} = \underset{\mathbf{w}}{\mathbf{argmin}} \frac{1}{2\gamma}||\mathbf{w} - \widehat{\mathbf{w}}^{(\mathbf{t+1})}||_2^2 + \lambda||\mathbf{w}||_1$ by setting subgradient of this objective to $\mathbf{0}$. This results in
    1. ...
    2. ...
    3. ...
  - Set $t = t + 1$, **until** stopping criterion is satisfied (such as no significant changes in $\mathbf{w}^t$ w.r.t $\mathbf{w}^{(t-1)}$)

# Proximal Subgradient Descent for Lasso

- Let $\varepsilon(\mathbf{w}) = \|\Phi\mathbf{w} - \mathbf{y}\|_2^2$
- **Proximal Subgradient Descent Algorithm:**
  **Initialization:** Find starting point $\mathbf{w}^{(0)}$ **(Only here, We use superscripts $t$ to represent iterates $\mathbf{w}^t$ so that subscripts $i$ can be used to represent indices $\mathbf{w}_i^t$)**
  - Let $\widehat{\mathbf{w}}^{(t+1)}$ be a next gradient descent iterate for $\mathbf{w}^t$
  - Compute $\mathbf{w}^{(t+1)} = \underset{\mathbf{w}}{\mathbf{argmin}} \frac{1}{2\gamma}||\mathbf{w} - \widehat{\mathbf{w}}^{(t+1)}||_2^2 + \lambda||\mathbf{w}||_1$ by setting (some) subgradient of this objective to $\mathbf{0}$. This results in
    1. $\cdots w_i^{(t+1)} = \widehat{w}_i^{(t+1)} - \lambda\gamma$ if $\widehat{w}_i^{(t+1)} \geq \lambda\gamma$
    2. $\cdots 0$ if $\widehat{w}_i^{(t+1)} \in [-\lambda\gamma, \lambda\gamma]$
    3. $\cdots w_i^{(t+1)} = \widehat{w}_i^{(t+1)} + \lambda\gamma$ if $\widehat{w}_i^{(t+1)} \leq -\lambda\gamma$
  - Set $t = t + 1$, **until** stopping criterion is satisfied (such as no significant changes in $\mathbf{w}^t$ w.r.t $\mathbf{w}^{(t-1)}$)

# Recall Subgradients in Lasso: Sufficient Condition Test

Recall from lecture 13[1] illustration of the sufficient condition test using the simplified Lasso as an example:

$$f(\mathbf{w}) = \frac{1}{2}||\mathbf{w} - \mathbf{y}||^2 + \lambda||\mathbf{w}||_1$$

Consider subgradients of $f(\mathbf{w})$ : $\mathbf{h} = \mathbf{w} - \mathbf{y} + \lambda\mathbf{s}$, where $s_i = sign(w_i)$ if $w_i \neq 0$ and $s_i \in [-1, 1]$ if $w_i = 0$.
A solution to this problem was found to be $\mathbf{w}^* = S_\lambda(\mathbf{y})$, where $S_\lambda(\mathbf{y})$ is the soft-thresholding operator:

$$S_\lambda(\mathbf{y}) = \begin{cases} y_i - \lambda & \text{if } y_i > \lambda \\ 0 & \text{if } -\lambda \leq y_i \leq \lambda \\ y_i + \lambda & \text{if } y_i < -\lambda \end{cases}$$

Now let $\mathbf{w}^* = S_\lambda(\mathbf{y})$ and we can get $\mathbf{h} = 0$. Why? If $y_i > \lambda$, we have $x_i^* - y_i = -\lambda + \lambda \cdot 1 = 0$. The case of $y_i < \lambda$ is similar. If $-\lambda \leq y_i \leq \lambda$, we have $x_i^* - y_i = -y_i + \lambda(\frac{y_i}{\lambda}) = 0$. Here, $s_i = \frac{y_i}{\lambda}$.

[1] pages 21-22 of https://moodle.iitb.ac.in/pluginfile.php/178765/mod_resource/content/39/CS769_2021__Lecture13-annotated.pdf

# Recall Subgradients in Lasso: Sufficient Condition Test

Recall from lecture 13[1] illustration of the sufficient condition test using the simplified Lasso as an example:

$$f(\mathbf{w}) = \frac{1}{2}||\mathbf{w} - \mathbf{y}||^2 + \lambda||\mathbf{w}||_1$$

Consider subgradients of $f(\mathbf{w})$ : $\mathbf{h} = \mathbf{w} - \mathbf{y} + \lambda\mathbf{s}$, where $s_i = sign(w_i)$ if $w_i \neq 0$ and $s_i \in [-1, 1]$ if $w_i = 0$.

A solution to this problem was found to be $\mathbf{w}^* = S_\lambda(\mathbf{y})$, where $S_\lambda(\mathbf{y})$ is the soft-thresholding operator:

The soft thresholding operator

$$S_\lambda(\mathbf{y}) = \begin{cases} y_i - \lambda & \text{if } y_i > \lambda \\ 0 & \text{if } -\lambda \leq y_i \leq \lambda \\ y_i + \lambda & \text{if } y_i < -\lambda \end{cases}$$

Now let $\mathbf{w}^* = S_\lambda(\mathbf{y})$ and we can get $\mathbf{h} = 0$. Why? If $y_i > \lambda$, we have $x_i^* - y_i = -\lambda + \lambda \cdot 1 = 0$. The case of $y_i < \lambda$ is similar. If $-\lambda \leq y_i \leq \lambda$, we have $x_i^* - y_i = -y_i + \lambda(\frac{y_i}{\lambda}) = 0$. Here, $s_i = \frac{y_i}{\lambda}$.

[1] pages 21-22 of https://moodle.iitb.ac.in/pluginfile.php/178765/mod_resource/content/39/CS769_2021__Lecture13-annotated.pdf

# Iterative Soft Thresholding Algorithm (Proximal Subgradient Descent) for Lasso

- Let $\varepsilon(\mathbf{w}) = \|\phi\mathbf{w} - \mathbf{y}\|_2^2$
- **Iterative Soft Thresholding Algorithm:**
  **Initialization:** Find starting point $\mathbf{w}^{(0)}$ **(Only here, We use superscripts $t$ to represent iterates $\mathbf{w}^t$ so that subscripts $i$ can be used to represent indices $\mathbf{w}_i^t$)**
  - Let $\widehat{\mathbf{w}}^{(\mathbf{t+1})}$ be a next iterate for $\mathbf{w}^t$ computed using using any (gradient) descent algorithm
  - Compute $\mathbf{w}^{(k+1)} = \underset{\mathbf{w}}{\operatorname{argmin}} \frac{1}{2\gamma}||\mathbf{w} - \widehat{\mathbf{w}}^{(\mathbf{t+1})}||_2^2 + \lambda||\mathbf{w}||_1$ by:
    1. If $\widehat{w}_i^{(t+1)} > \lambda\gamma$, then $w_i^{(t+1)} = -\lambda\gamma + \widehat{w}_i^{(t+1)}$
    2. If $\widehat{w}_i^{(t+1)} < -\lambda\gamma$, then $w_i^{(t+1)} = \lambda\gamma + \widehat{w}_i^{(t+1)}$
    3. 0 otherwise.
  - Set $t = t + 1$, **until** stopping criterion is satisfied (such as no significant changes in $\mathbf{w}^t$ w.r.t $\mathbf{w}^{(t-1)}$)

# Generalized Gradient Descent - generalizing (Sub)gradient Descent

- Recall

$$prox_{\gamma}(\mathbf{z}) = \text{argmin}_{\mathbf{x}} \frac{1}{2\gamma}||\mathbf{x} - \mathbf{z}||^2 + c(\mathbf{x})$$

1. Gradient Descent: $c(\mathbf{x}) = 0$
2. Projected Gradient Descent: $c(\mathbf{x}) = \sum_i I_{C_i}(\mathbf{x})$
3. Proximal Minimization: $f(\mathbf{x}) = 0$

We will discuss these specific cases after a short discussion on convergence

[2] Else we just treat this as another minimization problem and obtain an approximate solution. Practical convergence rate can be very slow. Exceptions are partial proximation minimization [Bertsekas and Tseng '94]

# Generalized Gradient Descent - generalizing (Sub)gradient Descent

- Recall

$$prox_\gamma(\mathbf{z}) = \text{argmin}_\mathbf{x} \frac{1}{2\gamma}||\mathbf{x}-\mathbf{z}||^2 + c(\mathbf{x})$$

1. Gradient Descent: $c(\mathbf{x}) = 0$ (or subgradient descent)
2. Projected Gradient Descent: $c(\mathbf{x}) = \sum_i I_{C_i}(\mathbf{x})$
3. Proximal Minimization: $f(\mathbf{x}) = 0$ No differentiable component at all!

$I_{C_1}(x)=0$ if $x \in C_1$, $\infty$ o/w


$I_{C_2}(x)=0$ if $x \in C_2$, $\infty$ o/w

We will discuss these specific cases after a short discussion on convergence

How was the convergence of subgradient descent? $O\left(\frac{1}{\sqrt{T}}\right)$ error

If we apply subgradient descent to f(x) + c(x), we could do in $O\left(\frac{1}{\epsilon^2}\right)$ time or error would decrease as $O\left(\frac{1}{\sqrt{T}}\right)$

Generalized GD availas of the regular gradient descent update and brings c(x) almost as an afterthought..
For this, can we expect this to avail the advatage(s) of gradient descent → At least $O\left(\frac{1}{T}\right)$ ?

[2] Else we just treat this as another minimization problem and obtain an approximate solution. Practical convergence rate can be very slow. Exceptions are partial proximation minimization [Bertsekas and Tseng '94]

# Generalized Gradient Descent - generalizing (Sub)gradient Descent

- Recall

$$prox_{\gamma}(\mathbf{z}) = \text{argmin}_{\mathbf{x}} \frac{1}{2\gamma}||\mathbf{x} - \mathbf{z}||^2 + c(\mathbf{x})$$

1. Gradient Descent: $c(\mathbf{x}) = 0$
2. Projected Gradient Descent: $c(\mathbf{x}) = \sum_i I_{C_i}(\mathbf{x})$
3. Proximal Minimization: $f(\mathbf{x}) = 0$

We will discuss these specific cases after a short discussion on convergence

- Convergence: If $f(\mathbf{x})$ is convex, differentiable, and $\nabla f$ is Lipschitz continuous with constant $L > 0$ AND $c(\mathbf{x})$ **is convex and** $prox_{\gamma}(\mathbf{z})$ **can be solved exactly**[2] then

[2] Else we just treat this as another minimization problem and obtain an approximate solution. Practical convergence rate can be very slow. Exceptions are partial proximation minimization [Bertsekas and Tseng '94]

# Generalized Gradient Descent - generalizing (Sub)gradient Descent

- Recall

$$prox_{\gamma}(\mathbf{z}) = \text{argmin}_{\mathbf{x}} \frac{1}{2\gamma}||\mathbf{x} - \mathbf{z}||^2 + c(\mathbf{x})$$

1. Gradient Descent: $c(\mathbf{x}) = 0$
2. Projected Gradient Descent: $c(\mathbf{x}) = \sum_i I_{C_i}(\mathbf{x})$
3. Proximal Minimization: $f(\mathbf{x}) = 0$

We will discuss these specific cases after a short discussion on convergence

- Convergence: If $f(\mathbf{x})$ is convex, differentiable, and $\nabla f$ is Lipschitz continuous with constant $L > 0$ AND $c(\mathbf{x})$ **is convex and** $prox_{\gamma}(\mathbf{z})$ **can be solved exactly**[2] then

$\approx O\left(\frac{1}{T}\right)$

Eg: $\|\phi x - y\|^2$ is L-smooth

Eg: $\|x\|_1$ is convex

Eg: In ISTA for Lasso we had a closed form sufficient condition using soft thresholding operator $S_\lambda(z)$

[2] Else we just treat this as another minimization problem and obtain an approximate solution. Practical convergence rate can be very slow. Exceptions are partial proximation minimization [Bertsekas and Tseng '94]

# Generalized Gradient Descent - generalizing (Sub)gradient Descent

- Recall

$$prox_{\gamma}(\mathbf{z}) = \text{argmin}_{\mathbf{x}} \frac{1}{2\gamma}||\mathbf{x} - \mathbf{z}||^2 + c(\mathbf{x})$$

1. Gradient Descent: $c(\mathbf{x}) = 0$
2. Projected Gradient Descent: $c(\mathbf{x}) = \sum_i I_{C_i}(\mathbf{x})$
3. Proximal Minimization: $f(\mathbf{x}) = 0$

We will discuss these specific cases after a short discussion on convergence

- Convergence: If $f(\mathbf{x})$ is convex, differentiable, and $\nabla f$ is Lipschitz continuous with constant $L > 0$ AND $c(\mathbf{x})$ **is convex and** $prox_{\gamma}(\mathbf{z})$ **can be solved exactly**[2] then convergence result (and proof) is similar to that for gradient descent

$$f(x_T) - f(x_*) \leq \frac{1}{T}\sum_{t=1}^{T}(f(x_t) - f(x_*)) \leq \frac{\|x_{(0)} - x_*\|^2}{2\gamma T}$$

Say using Subgradient descent

[2] Else we just treat this as another minimization problem and obtain an approximate solution. Practical convergence rate can be very slow. Exceptions are partial proximation minimization [Bertsekas and Tseng '94]

# Convergence Rate: Generalized Gradient Descent vs. Subgradient Descent

- Recap: For Subgradient Descent: The subgradient method has convergence rate $O(1/\sqrt{T})$; to get $f(\mathbf{x}_T^{best}) - f(\mathbf{x}_*) \leq \epsilon$, we need $O(1/\epsilon^2)$ iterations.
  This is actually the best we can do; *i.e.*, we can't do better than $O(1/\sqrt{T})$.

# Convergence Rate: Generalized Gradient Descent vs. Subgradient Descent

- Recap: For Subgradient Descent: The subgradient method has convergence rate $O(1/\sqrt{T})$; to get $f(\mathbf{x}_T^{best}) - f(\mathbf{x}_*) \leq \epsilon$, we need $O(1/\epsilon^2)$ iterations.
  This is actually the best we can do; *i.e.*, we can't do better than $O(1/\sqrt{T})$.

For Generalized Gradient Descent $O\left(\frac{1}{T}\right)$

# Convergence Rate: Generalized Gradient Descent vs. Subgradient Descent

- Recap: For Subgradient Descent: The subgradient method has convergence rate $O(1/\sqrt{T})$; to get $f(\mathbf{x}_T^{best}) - f(\mathbf{x}_*) \leq \epsilon$, we need $O(1/\epsilon^2)$ iterations.
  This is actually the best we can do; *i.e.*, we can't do better than $O(1/\sqrt{T})$.
- For generalized Gradient Descent: If $f(x)$ is convex, differentiable, and $\nabla f$ is Lipschitz continuous with constant $L > 0$ AND $c(x)$ is convex and $prox_\gamma(x)$ can be solved exactly then convergence result (and proof) is similar to that for gradient descent!!

$$f(x_T) - f(x_*) \leq \frac{1}{T}\sum_{t=1}^{T}(f(x_t) - f(x_*)) \leq \frac{\|x_0 - x_*\|^2}{2T\gamma}$$

**Better convergence ($O(1/T)$) because of assuming (i) Differentiability of $f(\mathbf{x})$ and (ii) Lipschitz continuity of $\nabla f(\mathbf{x})$. Further improvements using Accelerated Generalized Gradient Descent!**
**Can we do even better without strong convexity (which is not possible for $c(\mathbf{x})$)? Possibly for specific cases.**

# Convergence Rate: Generalized Gradient Descent vs. Subgradient Descent

- Recap: For Subgradient Descent: The subgradient method has convergence rate $O(1/\sqrt{T})$; to get $f(\mathbf{x}_T^{best}) - f(\mathbf{x}_*) \leq \epsilon$, we need $O(1/\epsilon^2)$ iterations.
  This is actually the best we can do; *i.e.*, we can't do better than $O(1/\sqrt{T})$.
- For generalized Gradient Descent: If $f(x)$ is convex, differentiable, and $\nabla f$ is Lipschitz continuous with constant $L > 0$ AND $c(x)$ is convex and $prox_\gamma(x)$ can be solved exactly then convergence result (and proof) is similar to that for gradient descent!!

In several cases of non-differentiable c(x) as well as in several cases in which the c(x) denotes constraints the prox step can be computed efficiently (and often in closed form!)

$$f(x_T) - f(x_*) \leq \frac{1}{T}\sum_{t=1}^{T}(f(x_t) - f(x_*)) \leq \frac{\|x_0 - x_*\|^2}{2T\gamma}$$

I will provide optional extra convergence analysis for generalized accelerated gradient descent etc

**Better convergence ($O(1/T)$) because of assuming (i) Differentiability of $f(x)$ and (ii) Lipschitz continuity of $\nabla f(x)$. Further improvements using Accelerated Generalized Gradient Descent!** Instead of $prox_f(x_t - r\nabla f(x_t))$ use $prox_f$(accelerated step)

**Can we do even better without strong convexity (which is not possible for $c(x)$)? Possibly for specific cases.**

# Generalized Gradient Descent and its Special Cases

Recall

$$prox_{\gamma}(\mathbf{z}) = \text{argmin}_{\mathbf{x}} \frac{1}{2\gamma}||\mathbf{x} - \mathbf{z}||^2 + c(\mathbf{x})$$

It's special cases are:

1. Gradient Descent: $c(\mathbf{x}) = 0$
2. Projected Gradient Descent: $c(\mathbf{x}) = I_{\mathcal{C}}(\mathbf{x})$ (Example:

# Generalized Gradient Descent and its Special Cases

Recall

$$prox_{\gamma}(\mathbf{z}) = \text{argmin}_{\mathbf{x}} \frac{1}{2\gamma}||\mathbf{x} - \mathbf{z}||^2 + c(\mathbf{x})$$

It's special cases are:

1. Gradient Descent: $c(\mathbf{x}) = 0$
2. Projected Gradient Descent: $c(\mathbf{x}) = I_C(\mathbf{x})$ (Example: $I_C(x) = 0$ if $x \geq 0$, $= \infty$ o/w)

$$x_{t+1} = x_t - \Gamma \nabla f(x_t)$$

$$prox_{\lambda}(x_{t+1}) = \min_x ||x - x_{t+1}||^2 + \lambda I_C(x)$$

$$= \begin{cases} (x_{t+1})_i & \text{if } (x_{t+1})_i \geq 0 \\ 0 & \text{if } (x_{t+1})_i < 0 \end{cases}$$

# Generalized Gradient Descent and its Special Cases

Recall

$$prox_{\gamma}(\mathbf{z}) = \text{argmin}_{\mathbf{x}} \frac{1}{2\gamma}||\mathbf{x} - \mathbf{z}||^2 + c(\mathbf{x})$$

It's special cases are:

1. Gradient Descent: $c(\mathbf{x}) = 0$
2. Projected Gradient Descent: $c(\mathbf{x}) = I_{\mathcal{C}}(\mathbf{x})$ (Example: $= \sum_i I_{g_i}(\mathbf{x})$)
3. Alternating Projection/Proximal Minimization: $f(\mathbf{x}) = 0$
4. Alternating Direction Method of Multipliers
5. Special Cases for Specific Objectives
   - LASSO: (Fast) Iterative Shrinkage Thresholding Algorithm (ISTA/FISTA)

# Generalized Gradient Descent and its Special Cases

Recall

$$prox_{\gamma}(\mathbf{z}) = \text{argmin}_{\mathbf{x}} \frac{1}{2\gamma}||\mathbf{x} - \mathbf{z}||^2 + c(\mathbf{x})$$

It's special cases are:

1. Gradient Descent: $c(\mathbf{x}) = 0$
2. Projected Gradient Descent: $c(\mathbf{x}) = I_{\mathcal{C}}(\mathbf{x})$ (Example: $= \sum_i I_{g_i}(\mathbf{x})$)
3. Alternating Projection/Proximal Minimization: $f(\mathbf{x}) = 0$ } Extra (optional)
4. Alternating Direction Method of Multipliers
5. Special Cases for Specific Objectives
   - LASSO: (Fast) Iterative Shrinkage Thresholding Algorithm (ISTA/FISTA)

# Lipschitz Continuous Functions

Another strong reason favouring proximal/generalized gradient descent

- Plain Lipschitz Continuous (without strong convexity) can be very slow for converergence!
- Many practical ML problems such as L1 Regularized Logistic Regression are Lipschitz Continuous
- Can we do better than the $O(1/\epsilon^2)$ convergence?
- Yes, we can. However we need to make some assumptions! (Recall the lower bound result?)
- The assumption is the function that causes the non-differentiability (for example, L1 norm) should be *simple*. & can be isolated to yield the remaining part that is Lsmooth/ strongly convex etc

# Proximal Gradient Descent

- Recall Gradient Descent as: $x_{t+1} = x_t - \gamma \nabla f(x_t)$.
- This is equivalent to:

$$x_{t+1} = \text{argmin}_x f(x_t) + \nabla f(x_t)^T (x - x_t) + \frac{1}{2\gamma} ||x - x_t||^2$$

- Now consider the optimization problem $\min_x [f(x) + c(x)]$ where $c$ is a non-differentiable function.
- The update then becomes:

$$x_{t+1} = \text{argmin}_x f(x_t) + \nabla f(x_t)^T (x - x_t) + \frac{1}{2\gamma} ||x - x_t||^2 + c(x)$$

- After some manipulation, the update becomes:

$$\text{argmin}_x \frac{1}{2\gamma} (x - [x_t - \gamma \nabla f(x_t)])^2 + c(x)$$

# Proximal Gradient Descent

- From the previous slide:

$$\text{argmin}_x \frac{1}{2\gamma}(x - [x_t - \gamma \nabla f(x_t)])^2 + c(x)$$

- Define $\text{prox}_\gamma(x) = \text{argmin}_z \frac{1}{2\gamma}||x - z||^2 + c(z)$
- Notice that the update rule then is

$$x_{t+1} = \text{prox}_\gamma(x_t - \gamma \nabla f(x_t))$$

- Convergence bound: Assume $f, h$ are convex and the proximal minimization is easy (ideally closed form), then using a step size of $\gamma = 1/L$, we can bound: $f(x_T) - f(x^*) \leq \frac{LR^2}{2T}$
- This is exactly the same convergence rate for smooth functions!

# How easy is it to compute the Prox operator?

- This depends on the specific function at hand.
- Lets consider some examples:
  - $c(x) = a^T x$
  - $c(x) = c$ (a constant)
  - $c(x) = \lambda ||x||^2$
  - $c(x) = -\lambda \log(x)$ (defined only when $x > 0$)
  - $c(x) = \frac{1}{2} x^T A x + b^T x + c$, $A$ is positive semi-definite
  - $c(x) = \mu x$ if $x \geq 0$ or $\infty$ else.

HOMEWORK

While we start with these simple examples there is also a calculus of prox operators which can be used to solve tricker cases of c(x)

- How to compute the Prox? Solve the optimization problem:

$$\text{prox}_{\gamma}(x) = \text{argmin}_z \frac{1}{2\gamma} ||z - x||^2 + c(z) = \text{argmin}_z \frac{1}{2} ||z - x||^2 + \gamma c(z)$$

- Using the optimality conditions (since $c$ is convex):

$$z - x + \gamma c'(z) = 0$$

- What about if the function is non-differentiable?

# Equivalent way of looking at Prox

- Some textbooks define prox as:

$$\text{prox-new}_c(x) = \text{argmin}_z \frac{1}{2}||z - x||^2 + c(z)$$

- Note that with our definition, $\text{prox}_\gamma(x) = \text{prox-new}_{\gamma c}(x)$
- We shall use this definition of prox for the rest of this class.

# Main ideas for computing Prox

HOMEWORK

- The main ideas of computing Prox operator are:
  1. If $c'(x) = 0$ for a convex function $c$, the $x$ must be one of its minimizers.
  2. If a minimizer of a convex function exists and is not attained at any point of differentiability, it must be attained at a point of non-differentiability. (through subgradient as in Lasso for $\|x\|_1$)
- Try computing the Prox operator for some more functions (homework)
  - $c(x) = \lambda x^3, x \geq 0$ and $-\infty$ otherwise.
  - $c(x) = 0$, for $x \neq 0$ and $-\lambda$ if $x = 0$
  - $c(x) = 0$, for $x \neq 0$ and $\lambda$ if $x = 0$

$\lambda \geq 0$

H/W

$x = \operatorname{argmin}_x \|x - z\|^2 + \lambda I_c(x)$

$= \frac{\sqrt{1 + 12z\lambda} - 2}{6\lambda}$ if $z \geq 0$ & $x = 0$ o/w

For $x \geq 0$, $\frac{d}{dx} = 0 \Rightarrow 2(x - z) + 3\lambda x^2 = 0 \Rightarrow x =$ .....

# Computing Prox when $c$ is non-differentiable

- We want to solve: $\text{prox}_f(x) = \text{argmin}_z \frac{1}{2}||z - x||^2 + c(z)$
- If $c$ is differentiable, we have the optimality condition as: $z - x + \nabla c(z) = 0$
- If $c$ is non-differentiable, denote $d_c \in \partial c(z)$ as the sub-gradient. The optimality condition is $0 \in z - x + d_c(z)$
- Consider $c(z) = \lambda||z||_1$
- Its easy to see that $x - z = \lambda d$ where $d_i = \{1\}$ if $z_i > 0$, $d_i = \{-1\}$ if $z_i < 0$ and $d_i \in [-1, 1]$ if $z_i = 0$.
- In other words, the sub-gradient optimality conditions are $x_i - z_i = \lambda \text{sign}(z_i)$ if $z_i \neq 0$ and $|x_i - z_i| \leq \lambda$ if $z_i = 0$
- Its easy to see that:

$$z_i = \begin{cases} x_i - \lambda & \text{if } x_i > \lambda \\ 0 & \text{if } -\lambda < x_i < \lambda \\ x_i + \lambda & \text{if } x_i < -\lambda \end{cases}$$

# Computing Prox for L1 Norm

- From the previous slide, the Prox opterator is:

$$z_i = \begin{cases} x_i - \lambda & \text{if } x_i > \lambda \\ 0 & \text{if } -\lambda < x_i < \lambda \\ x_i + \lambda & \text{if } x_i < -\lambda \end{cases}$$

- In other words $\text{prox}^i_{\lambda||x||_1} = [|x_i| - \lambda]_+ \text{sign}(x_i)$ is the soft-thresholding operator!

# Proximal Subgradient Descent for Lasso

- Let $f(\mathbf{x}) = \|\Phi\mathbf{x} - \mathbf{y}\|_2^2$, $c(\mathbf{x}) = \|\mathbf{x}\|_1$ and $F(\mathbf{x}) = f(\mathbf{x}) + c(\mathbf{x})$
- **Proximal Subgradient Descent Algorithm:**
  **Initialization:** Find starting point $\mathbf{x}^{(0)}$
  - Let $\widehat{\mathbf{x}}^{(t+1)} \equiv \mathbf{z}^{(t+1)}$ be a next gradient descent iterate for $f(\mathbf{x}^t)$
  - Compute $\mathbf{x}^{(t+1)} = \underset{\mathbf{x}}{\mathbf{argmin}} \frac{1}{2}\|\mathbf{x} - \mathbf{z}^{(t+1)}\|_2^2 + \lambda\gamma\|\mathbf{x}\|_1 = \text{Prox}_{\lambda\gamma}(\mathbf{z}^{(t+1)})$ by setting subgradient of this objective to $\mathbf{0}$. (We already saw that the Prox operator is the soft-threshold)
  - Set $t = t + 1$, **until** stopping criterion is satisfied (such as no significant changes in $\mathbf{x}^t$ w.r.t $\mathbf{x}^{(t-1)}$)

# Iterative Soft Thresholding Algorithm (Proximal Subgradient Descent) for Lasso

- Let $f(\mathbf{x}) = \|A\mathbf{x} - \mathbf{y}\|_2^2$, $c(\mathbf{x}) = \|\mathbf{x}\|_1$ and $F(\mathbf{x}) = f(\mathbf{x}) + c(\mathbf{x})$
- **Proximal Subgradient Descent Algorithm:**
  **Initialization:** Find starting point $\mathbf{x}^{(0)}$
  - Let $\mathbf{z}^{(t+1)}$ be a next gradient descent iterate for $f(\mathbf{x}^t)$
  - Compute $prox_{\|\mathbf{x}\|_1}\left(\mathbf{z}^{(t+1)}\right) = \mathbf{x}^{(t+1)} =$

# Iterative Soft Thresholding Algorithm (Proximal Subgradient Descent) for Lasso

- Let $f(\mathbf{x}) = \|A\mathbf{x} - \mathbf{y}\|_2^2$, $c(\mathbf{x}) = \|\mathbf{x}\|_1$ and $F(\mathbf{x}) = f(\mathbf{x}) + c(\mathbf{x})$
- **Proximal Subgradient Descent Algorithm:**
  **Initialization:** Find starting point $\mathbf{x}^{(0)}$
  - Let $\mathbf{z}^{(t+1)}$ be a next gradient descent iterate for $f(\mathbf{x}^t)$
  - Compute $prox_{\|\mathbf{x}\|_1}\left(\mathbf{z}^{(t+1)}\right) = \mathbf{x}^{(t+1)} = \underset{\mathbf{x}}{\mathbf{argmin}} \frac{1}{2\gamma}\|\mathbf{x} - \mathbf{z}^{(t+1)}\|_2^2 + \lambda\|\mathbf{x}\|_1$ as follows:

# Iterative Soft Thresholding Algorithm (Proximal Subgradient Descent) for Lasso

- Let $f(\mathbf{x}) = \|A\mathbf{x} - \mathbf{y}\|_2^2$, $c(\mathbf{x}) = \|\mathbf{x}\|_1$ and $F(\mathbf{x}) = f(\mathbf{x}) + c(\mathbf{x})$
- **Proximal Subgradient Descent Algorithm:**
  **Initialization:** Find starting point $\mathbf{x}^{(0)}$
  - Let $\mathbf{z}^{(t+1)}$ be a next gradient descent iterate for $f(\mathbf{x}^t)$
  - Compute $prox_{\|\mathbf{x}\|_1}\left(\mathbf{z}^{(t+1)}\right) = \mathbf{x}^{(t+1)} = \underset{\mathbf{x}}{\mathbf{argmin}} \frac{1}{2\gamma}\|\mathbf{x} - \mathbf{z}^{(t+1)}\|_2^2 + \lambda\|\mathbf{x}\|_1$ as follows:
    1. If $z_i^{(t+1)} > \lambda\gamma$, then $x_i^{(t+1)} = -\lambda\gamma + z_i^{(t+1)}$
    2. If $z_i^{(t+1)} < -\lambda\gamma$, then $x_i^{(t+1)} = \lambda\gamma + z_i^{(t+1)}$
    3. 0 otherwise.
  - Set $t = t + 1$, **until** stopping criterion is satisfied (such as no significant changes in $\mathbf{x}^t$ w.r.t $\mathbf{x}^{(t-1)}$)

# Tables for the Proximal Operator

$$prox_c(\mathbf{z}) = \text{argmin}_{\mathbf{x}} \frac{1}{2\gamma}||\mathbf{x} - \mathbf{z}||^2 + c(\mathbf{x})$$

| For $x \in \Re$, $c(x) =$ | For $z \in \Re$ & $\gamma = 1$, $prox_c(z) =$ |
|---|---|
| Simplified Lasso: $\lambda\|x\|$ | $[\|z\| - \lambda]_+ sign(z)$ |
| $\lambda x \quad x \geq 0$<br>$\infty \quad x < 0$ | $[z - \lambda]_+$ |
| $\lambda x^3 \quad x \geq 0$<br>$\infty \quad x < 0$ | $\frac{-1 + \sqrt{1 + 12\lambda[z]_+}}{6\lambda}$ |
| $-\lambda \log x \quad x > 0$<br>$\infty \quad x \leq 0$ | $\frac{z + \sqrt{z^2 + 4\lambda}}{2}$ |
| $\lambda x \quad 0 \leq x \leq \alpha$<br>$\infty \quad \text{otherwise}$ | $min\{max\{z - \lambda, 0\}, \alpha\}$ |

# Tables for the Proximal Operator

$$prox_c(\mathbf{z}) = \text{argmin}_{\mathbf{x}} \frac{1}{2\gamma}||\mathbf{x}-\mathbf{z}||^2 + c(\mathbf{x})$$

| For $x \in \Re$, $c(x) =$ | For $z \in \Re$ & $\gamma = 1$, $prox_c(z) =$ |
|---|---|
| Simplified Lasso: $\lambda\|x\|$ | $[\|z\| - \lambda]_+ sign(z)$ |
| $\lambda x \quad x \geq 0$<br>$\infty \quad x < 0$ | $[z-\lambda]_+$ |
| $\lambda x^3 \quad x \geq 0$<br>$\infty \quad x < 0$ | $\frac{-1+\sqrt{1+12\lambda[z]_+}}{6\lambda}$ |
| $-\lambda \log x \quad x > 0$<br>$\infty \quad x \leq 0$ | $\frac{z+\sqrt{z^2+4\lambda}}{2}$ |
| $\lambda x \quad 0 \leq x \leq \alpha$<br>$\infty \quad \text{otherwise}$ | $min\{max\{z-\lambda, 0\}, \alpha\}$ |

$[a]_+ = max(0, a)$

We derived this

Intuition:

$\leq \alpha$

$\geq 0$

Serves as some kind of regularizer operating in some interval

# Tables for the Proximal Operator

$$prox_c(\mathbf{z}) = \text{argmin}_{\mathbf{x}} \frac{1}{2\gamma}||\mathbf{x} - \mathbf{z}||^2 + c(\mathbf{x})$$

| For $x \in \Re^n$, $c(\mathbf{x}) =$ | For $z \in \Re$ & $\gamma = 1$, $prox_c(\mathbf{z}) =$ |
|---|---|
| Constant: $c$ | $\mathbf{z}$ |
| Affine: $\mathbf{a}^T\mathbf{x} + b$ | $\mathbf{z} - \mathbf{a}$ |
| Convex quadratic: $\frac{1}{2}\mathbf{x}^T A\mathbf{x} + \mathbf{b}^T\mathbf{x} + c$ (where $A \in S_+^n, \mathbf{b} \in \Re^n$) | $(A + I)^{-1}(\mathbf{z} - \mathbf{b})$ |

# Tables for the Proximal Operator

$$prox_c(\mathbf{z}) = \text{argmin}_{\mathbf{x}} \frac{1}{2\gamma}||\mathbf{x} - \mathbf{z}||^2 + c(\mathbf{x})$$

| For $x \in \Re^n$, $c(\mathbf{x}) =$ | For $z \in \Re$ & $\gamma = 1$, $prox_c(\mathbf{z}) =$ |
|---|---|
| Constant: $c$ | $\mathbf{z}$ |
| Affine: $\mathbf{a}^T\mathbf{x} + b$ | $\mathbf{z} - \mathbf{a}$ |
| Convex quadratic: $\frac{1}{2}\mathbf{x}^T A\mathbf{x} + \mathbf{b}^T\mathbf{x} + c$ (where $A \in S_+^n, \mathbf{b} \in \Re^n$) | $(A + I)^{-1}(\mathbf{z} - \mathbf{b})$ |

Note: in previous slide, the constraints on x translated into post-processing operations on z-a

Preconditioning based on quadratic component

# Calculus for the Proximal Operator: See https://archive.siam.org/books/mo25/mo25_ch6.pdf

$$prox_c(\mathbf{z}) = \text{argmin}_{\mathbf{x}} \frac{1}{2\gamma}||\mathbf{x}-\mathbf{z}||^2 + c(\mathbf{x})$$

| $c(\mathbf{x}) =$ | For $\gamma = 1$, $prox_c(\mathbf{z}) =$ |
|---|---|
| Sum over components: $c(\mathbf{x}) = \sum_{i=1}^{n} c_i(\mathbf{x}_i)$<br>where $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ | Product over components: $prox_c(\mathbf{z}) = \prod_{i=1}^{n} prox_{c_i}(\mathbf{z}_i)$<br>where $\mathbf{z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n]$ |
| $c(\lambda\mathbf{x} + \mathbf{a})$<br>where $\lambda \neq 0$ and $c$ is proper | $\frac{1}{\lambda}\left[prox_{\lambda^2 c}(\lambda\mathbf{z} + \mathbf{a}) - \mathbf{a}\right]$ |
| $\lambda c\left(\frac{1}{\lambda}\mathbf{x}\right)$<br>where $\lambda \neq 0$ and $c$ is proper | $\lambda prox_{c/\lambda}\left(\frac{1}{\lambda}\mathbf{z}\right)$ |
| $c(\mathbf{x}) + \mathbf{a}^T\mathbf{x} + \frac{\beta}{2}\|\mathbf{x}\|^2 + \gamma$<br>where $\beta > 0$, $\gamma \in \Re$, $c$ is proper | $prox_{\frac{1}{\beta+1}c}\left(\frac{\mathbf{z}-\mathbf{a}}{\gamma+1}\right)$ |
| $c(A\mathbf{x} + \mathbf{b})$<br>where $c$ is proper closed and convex, $\mathbf{b} \in \Re^n$, $AA^T = \alpha I$, $\alpha > 0$ | $\mathbf{z} + \frac{1}{\alpha}A^T\left(prox_{\alpha c}\left(A\mathbf{z} + \mathbf{b}\right) - A\mathbf{z} - \mathbf{b}\right)$ |
| $c(\|\mathbf{x}\|)$<br>where $\mathbf{b} \in \Re^n$, $AA^T = \alpha I$, $\alpha > 0$ | $prox_c(\|\mathbf{z}\|)\frac{\mathbf{z}}{\|\mathbf{z}\|}$ $\quad \mathbf{z} \neq 0$<br>$\{\mathbf{u} \| \|\mathbf{u}\| = prox_c(0)\}$ $\quad \mathbf{z} = 0$ |

# Calculus for the Proximal Operator: See https://archive.siam.org/books/mo25/mo25_ch6.pdf

Author: Amir Beck

$$prox_c(\mathbf{z}) = \text{argmin}_{\mathbf{x}} \frac{1}{2\gamma}||\mathbf{x}-\mathbf{z}||^2 + c(\mathbf{x})$$

| $c(\mathbf{x}) =$ | For $\gamma = 1$, $prox_c(\mathbf{z}) =$ |
|---|---|
| Sum over components: $c(\mathbf{x}) = \sum_{i=1}^{n} c_i(\mathbf{x}_i)$<br>where $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ | Product over components: $prox_c(\mathbf{z}) = \prod_{i=1}^{n} prox_{c_i}(\mathbf{z}_i)$<br>where $\mathbf{z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n]$ |
| $c(\lambda\mathbf{x} + \mathbf{a})$<br>where $\lambda \neq 0$ and $c$ is proper | $\frac{1}{\lambda}\left[prox_{\lambda^2 c}(\lambda\mathbf{z} + \mathbf{a}) - \mathbf{a}\right]$ |
| $\lambda c\left(\frac{1}{\lambda}\mathbf{x}\right)$<br>where $\lambda \neq 0$ and $c$ is proper | $\lambda prox_{c/\lambda^2}\left(\frac{1}{\lambda}\mathbf{z}\right)$ |
| $c(\mathbf{x}) + \mathbf{a}^T\mathbf{x} + \frac{\beta}{2}\|\mathbf{x}\|^2 + \gamma$<br>where $\beta > 0$, $\gamma \in \Re$, $c$ is proper | $prox_{\frac{1}{\beta+1}c}\left(\frac{\mathbf{z}-\mathbf{a}}{\gamma+1}\right)$ |
| $c(A\mathbf{x} + \mathbf{b})$<br>where $c$ is proper closed and convex, $\mathbf{b} \in \Re^n$, $AA^T = \alpha I$, $\alpha > 0$ | $\mathbf{z} + \frac{1}{\alpha}A^T\left(prox_{\alpha c}(A\mathbf{z} + \mathbf{b}) - A\mathbf{z} - \mathbf{b}\right)$ |
| $c(\|\mathbf{x}\|)$<br>where $\mathbf{b} \in \Re^n$, $AA^T = \alpha I$, $\alpha > 0$ | $prox_c(\|\mathbf{z}\|)\frac{\mathbf{z}}{\|\mathbf{z}\|}$ $\quad \mathbf{z} \neq 0$<br>$\{\mathbf{u} \| \|\mathbf{u}\| = prox_c(0)\}$ $\quad \mathbf{z} = 0$ |

Preconditioning which we saw earlier
Only difference is that here we have
$A = \frac{\beta}{2} I$

# Summary of Proximal Gradient Descent

- Proximal Gradient Descent becomes gradient descent if $h = 0$
- If $f$ is 0 (i.e. no smooth function), one can minimize a non-differentiable function $h$ as long as the prox operator is easy to compute: **Proximal Point Algorithm**.
- Key to Proximal GD: Being able to compute Proximal operator.
- What if Prox can be efficiently computed approximately?
- There are papers (Schmidt et al 2011, Inexact Proximal Gradient Methods) where Prox is computed approximately but one can still derive the convergence rates if the errors due to approximation can be controlled!
- Accelerated Proximal GD: Similar to GD, one can accelerate GD to get optimal convergence rates! (Beck and Teboulle 2008)

# Summary of Proximal Gradient Descent

- Proximal Gradient Descent becomes gradient descent if $h = 0$
- If $f$ is 0 (i.e. no smooth function), one can minimize a non-differentiable function $h$ as long as the prox operator is easy to compute: **Proximal Point Algorithm**.
- Key to Proximal GD: Being able to compute Proximal operator.
- What if Prox can be efficiently computed approximately?
- There are papers (Schmidt et al 2011, Inexact Proximal Gradient Methods) where Prox is computed approximately but one can still derive the convergence rates if the errors due to approximation can be controlled!
- Accelerated Proximal GD: Similar to GD, one can accelerate GD to get optimal convergence rates! (Beck and Teboulle 2008)

# Overall: Generalized Gradient Descent - generalizing (Sub)gradient Descent

- $$prox_{\gamma}(\mathbf{z}) = \text{argmin}_{\mathbf{x}} \frac{1}{2\gamma}||\mathbf{x}-\mathbf{z}||^2 + c(\mathbf{x})$$

1. Gradient Descent: $c(\mathbf{x}) = 0$
2. Projected Gradient Descent: $c(\mathbf{x}) = \sum_i I_{C_i}(\mathbf{x})$
3. Proximal Minimization: $f(\mathbf{x}) = 0$

- Convergence: If $f(\mathbf{x})$ is convex, differentiable, and $\nabla f$ is Lipschitz continuous with constant $L > 0$ AND $c(\mathbf{x})$ **is convex and** $prox_{\gamma}(\mathbf{z})$ **can be solved exactly**[3] then convergence result (and proof) is similar to that for gradient descent

$$f(x_t) - f(x_*) \leq \frac{1}{T}\sum_{t=1}^{T}(f(x_t) - f(x_*)) \leq \frac{\|x_{(0)} - x_*\|^2}{2\gamma T}$$

[3] Else we just treat this as another minimization problem and obtain an approximate solution. Practical convergence rate can be very slow. Exceptions are partial proximation minimization [Bertsekas and Tseng '94]

# Overall: Generalized Gradient Descent - generalizing (Sub)gradient Descent

- $$prox_\gamma(\mathbf{z}) = \text{argmin}_\mathbf{x} \frac{1}{2\gamma}||\mathbf{x}-\mathbf{z}||^2 + c(\mathbf{x})$$

1. Gradient Descent: $c(\mathbf{x}) = 0$
2. Projected Gradient Descent: $c(\mathbf{x}) = \sum_i I_{C_i}(\mathbf{x})$
3. Proximal Minimization: $f(\mathbf{x}) = 0$

You can apply acceleration to get z

- Convergence: If $f(\mathbf{x})$ is convex, differentiable, and $\nabla f$ is Lipschitz continuous with constant $L > 0$ AND $c(\mathbf{x})$ **is convex and** $prox_\gamma(\mathbf{z})$ **can be solved exactly**[3] then convergence result (and proof) is similar to that for gradient descent

$$f(x_t) - f(x_*) \leq \frac{1}{T}\sum_{t=1}^{T}(f(x_t) - f(x_*)) \leq \frac{\|x_{(0)} - x_*\|^2}{2\gamma T}$$

[3] Else we just treat this as another minimization problem and obtain an approximate solution. Practical convergence rate can be very slow. Exceptions are partial proximation minimization [Bertsekas and Tseng '94]

# Convergence Rate: Generalized Gradient Descent vs. Subgradient Descent

- Recap: For Subgraident Descent: The subgradient method has convergence rate $O(1/\sqrt{T})$; to get $f(\mathbf{x}_T^{best}) - f(\mathbf{x}_*) \leq \epsilon$, we need $O(1/\epsilon^2)$ iterations.
  This is actually the best we can do; *i.e.*, we can't do better than $O(1/\sqrt{T})$.
- For generalized Gradient Descent: If $f(x)$ is convex, differentiable, and $\nabla f$ is Lipschitz continuous with constant $L > 0$ AND $c(x)$ is convex and $prox_\gamma(x)$ can be solved exactly then convergence result (and proof) is similar to that for gradient descent!! (we present convergence proof directly for the accelerated version as OPTIONAL material)

$$f(x_T) - f(x_*) \leq \frac{1}{T}\sum_{t=1}^{T}(f(x_t) - f(x_*)) \leq \frac{\|x_0 - x_*\|^2}{2T\gamma}$$

**Better convergence ($O(1/T)$) because of assuming (i) Differentiability of $f(\mathbf{x})$ and (ii) Lipschitz continuity of $\nabla f(\mathbf{x})$. Further improvements using Accelerated Generalized Gradient Descent!**
**Can we do even better without strong convexity (which is not possible for $c(\mathbf{x})$)? Possibly for specific cases.**

# Accelerated Generalized Gradient Descent

# (Nesterov) Accelerated Generalized Gradient Descent

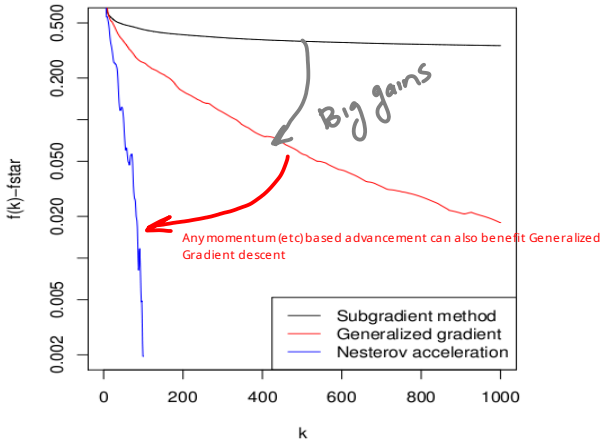



Figure 1: Comparison of different gradient methods

Beginning of EXTRA & OPTIONAL DETAILED SLIDES on
Accelerated Generalized Gradient Descent

# (Nesterov) Accelerated Generalized Gradient Descent

The problem is:

$$\min_{x \in \mathbb{R}^n} \; f(\mathbf{x}) + c(\mathbf{x})$$

where $f(\mathbf{x})$ is convex and differentiable, $c(\mathbf{x})$ is convex and not necessarily differentiable.

- Initialize $\mathbf{x}_u^{(0)} \in \mathbb{R}^n$
- repeat for $t = 1, 2, 3, \ldots$

$$\mathbf{y} = \mathbf{x}^{(t-1)} + \frac{t-2}{t+1}(\mathbf{x}^{(t-1)} - \mathbf{x}^{(t-2)})$$

$$\mathbf{x}^{(t)} = \text{prox}_{\gamma_t}(\mathbf{y} - \gamma_t \nabla f(\mathbf{y}))$$

Or Equivalently, with $\mu_t = 2/(t+1)$.

$$\mathbf{y} = (1 - \mu_t)\mathbf{x}^{(t-1)} + \mu_t \mathbf{x}_u^{(t-1)}$$

$$\mathbf{x}^{t} = \text{prox}_{\gamma_t}(\mathbf{y} - \gamma_t \nabla f(\mathbf{y}))$$

$$\mathbf{x}_u^{(t)} = \mathbf{x}^{(t-1)} + \frac{1}{\mu_t}(\mathbf{x}^{(t)} - \mathbf{x}^{(t-1)})$$

# Algorithm: (Nesterov) Accelerated Generalized Gradient Descent

**Initialize** $\mathbf{x}_u^{(0)}, \mathbf{x}^{(0)} \in \Re^n$
Initialize $t = 1$
**repeat**
1. $\mu_t = 2/(t+1)$
2. $\mathbf{y} = (1-\mu_t)\mathbf{x}^{(t-1)} + \mu_t \mathbf{x}_u^{(t-1)}$.
3. Choose a step size $\gamma_t > 0$ using exact or backtracking ray search.
4. $\mathbf{x}^t = \text{prox}_{\gamma_t}(\mathbf{y} - \gamma_t \nabla f(\mathbf{y}))$
5. $\mathbf{x}_u^{(t)} = \mathbf{x}^{(t-1)} + \frac{1}{\mu_t}(\mathbf{x}^{(t)} - \mathbf{x}^{(t-1)})$
6. Set $t = t + 1$.

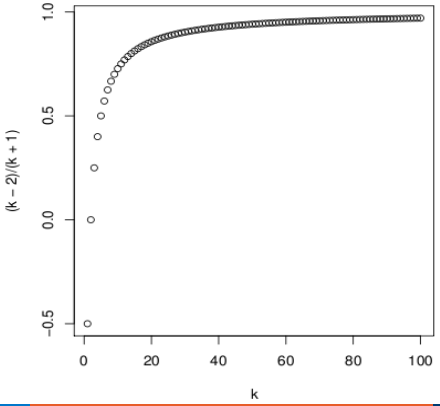
**until** stopping criterion (such as $||\mathbf{x}^t - \mathbf{x}^{t-1}|| \leq \epsilon$ or $f(\mathbf{x}^t) > f(\mathbf{x}^{t-1})$) is satisfied[a]

[a] Better criteria can be found using Lagrange duality theory, etc.

Figure 2: The gradient descent algorithm.

# (Nesterov) Accelerated Generalized Gradient Descent

1. First step $t = 1$ is just usual generalized gradient update: $\mathbf{x}^{(1)} = \text{prox}_{\gamma_1}(\mathbf{x}^{(0)} - \gamma_1 \nabla f(\mathbf{x}^{(0)}))$
2. Thereafter, the method carries some "momentum" from previous iterations
3. $c(\mathbf{x}) = 0$ gives accelerated gradient method
4. The method accelerates more towards the end of iterations



# (Nesterov) Accelerated Generalized Gradient Descent

Examples showing the performance of accelerated gradient descent compared with usual gradient descent.

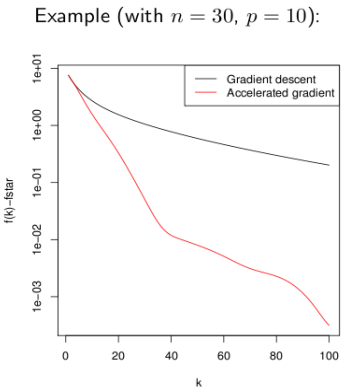



Figure 4: Example 1: Performance of accelerated gradient descent compared with usual gradient descent

# (Nesterov) Accelerated Generalized Gradient Descent: Convergence

Minimize $f(\mathbf{x}) = f(\mathbf{x}) + c(\mathbf{x})$ assuming that:
$f$ is convex, differentiable, $\nabla f$ is Lipschitz with constant $L > 0$, and
$c$ is convex, the prox function can be evaluated.

## Theorem

*Accelerated generalized gradient method with fixed step size $\gamma \leq 1/L$ satisfies:*

$$f(\mathbf{x}^{(T)}) - f(\mathbf{x}^*) \leq \frac{2||x^{(0)} - x^*||^2}{\gamma(T+1)^2}$$

Accelerated generalized gradient method can achieve the optimal $O(1/T^2)$ rate for first-order method, or equivalently, if we want to get $f(\mathbf{x}^{(T)}) - f(\mathbf{x}^*) \leq \epsilon$, we only need $O(1/\epsilon^2)$ iterations. Now we prove this theorem.

# (Nesterov) Accelerated Generalized Gradient Descent: Proof

**Proof:**

First we bound both the convex functions $f(\mathbf{x}^t)$ and $c(\mathbf{x}^t)$.

- Since $\gamma \leq 1/L$ and $\nabla f$ is Lipschitz with constant $L > 0$, we have

$$f(\mathbf{x}^t) \leq f(\mathbf{y}) + \nabla^T f(\mathbf{y})(\mathbf{x}^t - \mathbf{y}) + \frac{L}{2}||\mathbf{x}^t - \mathbf{y}||^2 \leq f(\mathbf{y}) + \nabla f(\mathbf{y})^T(\mathbf{x}^t - \mathbf{y}) + \frac{1}{2\gamma}||\mathbf{x}^t - \mathbf{y}||^2 \quad (1)$$

- In $\mathbf{x}^t = \text{prox}_\gamma(\mathbf{y} - gamma\nabla f(\mathbf{y}))$, let $\mathbf{h} = \mathbf{x}^t$ and $\mathbf{w} = \mathbf{y} - \gamma\nabla f(\mathbf{y})$. Then

$$\mathbf{h} = \text{prox}_\gamma(\mathbf{w}) = \arg\min_{\mathbf{h}} \frac{1}{2\gamma}||\mathbf{w} - \mathbf{h}||^2 + c(\mathbf{h})$$

- For this, we must have

$$0 \in \partial\left(\frac{1}{2\gamma}||\mathbf{w} - \mathbf{h}||^2 + c(\mathbf{h})\right) = -\frac{1}{\gamma}(\mathbf{w} - \mathbf{h}) + \partial c(\mathbf{h}) \quad \Rightarrow \quad -\frac{1}{\gamma}(\mathbf{w} - \mathbf{h}) \in \partial c(\mathbf{h})$$

- According to the definition of subgradient, we have for all $\mathbf{z}$,

# (Nesterov) Accelerated Generalized Gradient Descent: Proof

**Proof:**

First we bound both the convex functions $f(\mathbf{x}^t)$ and $c(\mathbf{x}^t)$.

- Since $\gamma \leq 1/L$ and $\nabla f$ is Lipschitz with constant $L > 0$, we have

$$f(\mathbf{x}^t) \leq f(\mathbf{y}) + \nabla^T f(\mathbf{y})(\mathbf{x}^t - \mathbf{y}) + \frac{L}{2}||\mathbf{x}^t - \mathbf{y}||^2 \leq f(\mathbf{y}) + \nabla f(\mathbf{y})^T(\mathbf{x}^t - \mathbf{y}) + \frac{1}{2\gamma}||\mathbf{x}^t - \mathbf{y}||^2 \quad (1)$$

- In $\mathbf{x}^t = \text{prox}_\gamma(\mathbf{y} - gamma\nabla f(\mathbf{y}))$, let $\mathbf{h} = \mathbf{x}^t$ and $\mathbf{w} = \mathbf{y} - \gamma\nabla f(\mathbf{y})$. Then

$$\mathbf{h} = \text{prox}_\gamma(\mathbf{w}) = \arg\min_{\mathbf{h}} \frac{1}{2\gamma}||\mathbf{w} - \mathbf{h}||^2 + c(\mathbf{h})$$

- For this, we must have

$$0 \in \partial\left(\frac{1}{2\gamma}||\mathbf{w} - \mathbf{h}||^2 + c(\mathbf{h})\right) = -\frac{1}{\gamma}(\mathbf{w} - \mathbf{h}) + \partial c(\mathbf{h}) \quad \Rightarrow \quad -\frac{1}{\gamma}(\mathbf{w} - \mathbf{h}) \in \partial c(\mathbf{h})$$

- According to the definition of subgradient, we have for all $\mathbf{z}$,

$$c(\mathbf{z}) \geq c(\mathbf{h}) - \frac{1}{\gamma}(\mathbf{h} - \mathbf{w})^T(\mathbf{z} - \mathbf{h}) \quad \Rightarrow \quad c(\mathbf{h}) \leq c(\mathbf{z}) + \frac{1}{\gamma}(\mathbf{h} - \mathbf{w})^T(\mathbf{z} - \mathbf{h})$$

# (Nesterov) Accelerated Generalized Gradient Descent: Proof (contd.)

Substituting back for both $\mathbf{h}$ and $\mathbf{w}$ in the above inequality we get for all $\mathbf{z}$,

$$c(\mathbf{x}^t) \leq c(\mathbf{z}) + \frac{1}{\gamma}(\mathbf{x}^t - \mathbf{y} + \gamma \nabla f(\mathbf{y}))^T(\mathbf{z} - \mathbf{x}^t) = c(\mathbf{z}) + \frac{1}{\gamma}(\mathbf{x}^t - \mathbf{y})^T(\mathbf{z} - \mathbf{x}^t) + \nabla f(\mathbf{y})^T(\mathbf{z} - \mathbf{x}^t) \quad (2)$$

Adding inequalities (1) and (2) we get for all $\mathbf{z}$,

$$f(\mathbf{x}^t) \leq f(\mathbf{y}) + c(\mathbf{z}) + \frac{1}{\gamma}(\mathbf{x}^t - \mathbf{y})^T(\mathbf{z} - \mathbf{x}^t) + \frac{1}{2\gamma}||\mathbf{x}^t - \mathbf{y}||^2 + \nabla f(\mathbf{y})^T(\mathbf{z} - \mathbf{y})$$

Since $f$ is convex,

# (Nesterov) Accelerated Generalized Gradient Descent: Proof (contd.)

Substituting back for both $\mathbf{h}$ and $\mathbf{w}$ in the above inequality we get for all $\mathbf{z}$,

$$c(\mathbf{x}^t) \leq c(\mathbf{z}) + \frac{1}{\gamma}(\mathbf{x}^t - \mathbf{y} + \gamma \nabla f(\mathbf{y}))^T(\mathbf{z} - \mathbf{x}^t) = c(\mathbf{z}) + \frac{1}{\gamma}(\mathbf{x}^t - \mathbf{y})^T(\mathbf{z} - \mathbf{x}^t) + \nabla f(\mathbf{y})^T(\mathbf{z} - \mathbf{x}^t) \quad (2)$$

Adding inequalities (1) and (2) we get for all $\mathbf{z}$,

$$f(\mathbf{x}^t) \leq f(\mathbf{y}) + c(\mathbf{z}) + \frac{1}{\gamma}(\mathbf{x}^t - \mathbf{y})^T(\mathbf{z} - \mathbf{x}^t) + \frac{1}{2\gamma}||\mathbf{x}^t - \mathbf{y}||^2 + \nabla f(\mathbf{y})^T(\mathbf{z} - \mathbf{y})$$

Since $f$ is convex, using $f(\mathbf{z}) \geq f(\mathbf{y}) + \nabla f(\mathbf{y})^T(\mathbf{z} - \mathbf{y})$, we further get
$f(\mathbf{x}^t) \leq f(\mathbf{z}) + \frac{1}{\gamma}(\mathbf{x}^t - \mathbf{y})^T(\mathbf{z} - \mathbf{x}^t) + \frac{1}{2\gamma}||\mathbf{x}^t - \mathbf{y}||^2$
Now take $\mathbf{z} = \mathbf{x}^{(t-1)}$, multiply both sides by $(1 - \theta)$ and for $\mathbf{z} = \mathbf{x}^*$ multiply both sides by $\theta$,

$$(1 - \theta)f(\mathbf{x}^t) \leq (1 - \theta)f(\mathbf{x}^{(t-1)}) + \frac{1 - \theta}{\gamma}(\mathbf{x}^t - \mathbf{y})^T(\mathbf{x}^{(t-1)} - \mathbf{x}^t) + \frac{1 - \theta}{2\gamma}||\mathbf{x}^t - \mathbf{y}||^2$$

$$\theta f(\mathbf{x}^t) \leq \theta f(\mathbf{x}^*) + \frac{\theta}{\gamma}(\mathbf{x}^t - \mathbf{y})^T(\mathbf{x}^* - \mathbf{x}^t) + \frac{\theta}{2\gamma}||\mathbf{x}^t - \mathbf{y}||^2$$

# (Nesterov) Accelerated Generalized Gradient Descent: Proof (contd.)

Adding these two inequalities together, we get

$$f(\mathbf{x}^t)-f(\mathbf{x}^*)-(1-\theta)(f(\mathbf{x}^{(t-1)})-f(\mathbf{x}^*)) \leq \frac{1}{\gamma}(\mathbf{x}^t-\mathbf{y})^T((1-\theta)\mathbf{x}^{(t-1)}+\theta\mathbf{x}^*-\mathbf{x}^t)+\frac{1}{2\gamma}||\mathbf{x}^t-\mathbf{y}||^2 \tag{3}$$

- Using $\mathbf{x}_u^t = \mathbf{x}^{(t-1)} + \frac{1}{\theta}(\mathbf{x}^t - \mathbf{x}^{(t-1)})$ and $\mathbf{y} = (1-\theta)\mathbf{x}^{(t-1)} + \theta\mathbf{x}_u^{(t-1)}$, we have $(1-\theta)\mathbf{x}^{(t-1)} + \theta\mathbf{x}^* - \mathbf{x}^t = \theta(\mathbf{x}^* - \mathbf{x}_u^t)$ and using this again in the second equation, $\mathbf{x}^t - \mathbf{y} = \theta(\mathbf{x}_u^t - \mathbf{x}_u^{(t-1)})$
- Substituting these equations into the RHS of inequality (3) we have

$$f(\mathbf{x}^t) - f(\mathbf{x}^*) - (1-\theta)(f(\mathbf{x}^{(t-1)}) - f(\mathbf{x}^*)) \leq \frac{\theta}{2t}\underline{(\mathbf{x}_u^t - \mathbf{x}_u^{(t-1)})}^T[2\theta(\mathbf{x}^* - \mathbf{x}_u^t) + \theta(\mathbf{x}_u^t - \mathbf{x}_u^{(t-1)})]$$

# (Nesterov) Accelerated Generalized Gradient Descent: Proof (contd.)

Adding these two inequalities together, we get

$$f(\mathbf{x}^t)-f(\mathbf{x}^*)-(1-\theta)(f(\mathbf{x}^{(t-1)})-f(\mathbf{x}^*)) \leq \frac{1}{\gamma}(\mathbf{x}^t-\mathbf{y})^T((1-\theta)\mathbf{x}^{(t-1)}+\theta\mathbf{x}^*-\mathbf{x}^t)+\frac{1}{2\gamma}||\mathbf{x}^t-\mathbf{y}||^2 \tag{3}$$

- Using $\mathbf{x}_u^t = \mathbf{x}^{(t-1)} + \frac{1}{\theta}(\mathbf{x}^t - \mathbf{x}^{(t-1)})$ and $\mathbf{y} = (1-\theta)\mathbf{x}^{(t-1)} + \theta\mathbf{x}_u^{(t-1)}$, we have $(1-\theta)\mathbf{x}^{(t-1)} + \theta\mathbf{x}^* - \mathbf{x}^t = \theta(\mathbf{x}^* - \mathbf{x}_u^t)$ and using this again in the second equation, $\mathbf{x}^t - \mathbf{y} = \theta(\mathbf{x}_u^t - \mathbf{x}_u^{(t-1)})$
- Substituting these equations into the RHS of inequality (3) we have

$$f(\mathbf{x}^t) - f(\mathbf{x}^*) - (1-\theta)(f(\mathbf{x}^{(t-1)}) - f(\mathbf{x}^*)) \leq \frac{\theta}{2t}(\mathbf{x}_u^t - \mathbf{x}_u^{(t-1)})^T[2\theta(\mathbf{x}^* - \mathbf{x}_u^t) + \theta(\mathbf{x}_u^t - \mathbf{x}_u^{(t-1)})]$$

$$= \frac{\theta^2}{2\gamma}(\mathbf{x}^* - \mathbf{x}_u^{(t-1)}) - (\mathbf{x}^* - \mathbf{x}_u^{(t-1)})]^T[(\mathbf{x}^* - \mathbf{x}_u^t) + (\mathbf{x}^* - \mathbf{x}_u^{(t-1)})]$$

$$= \frac{\theta^2}{2\gamma}(||\mathbf{x}_u^{(t-1)} - \mathbf{x}^*||^2 - ||\mathbf{x}_u^t - \mathbf{x}^*||^2)$$

# (Nesterov) Accelerated Generalized Gradient Descent: Proof (contd.)

$$\frac{\gamma}{\mu_t^2}(f(\mathbf{x}^{(t)}) - f(\mathbf{x}^*)) + \frac{1}{2}||\mathbf{x}_u^{(t)} - \mathbf{x}^*||^2 \leq \frac{\gamma(1-\mu_t)}{\mu_t^2}(f(\mathbf{x}^{(t-1)}) - f(\mathbf{x}^*)) + \frac{1}{2}||\mathbf{x}_u^{(t-1)} - \mathbf{x}^*||^2$$

Since $\theta = 2/(t+1)$, using $\frac{1-\mu_t}{\theta_t^2} \leq \frac{1}{\theta_{t-1}^2}$, we have

$$\frac{\gamma}{\mu_t^2}(f(\mathbf{x}^{(t)}) - f(\mathbf{x}^*)) + \frac{1}{2}||\mathbf{x}_u^{(t)} - x^*||^2 \leq \frac{\gamma}{\mu_{t-1}^2}(f(\mathbf{x}^{(t-1)}) - f(\mathbf{x}^*)) + \frac{1}{2}||\mathbf{x}_u^{(t-1)} - \mathbf{x}^*||^2$$

Iterating this inequality and using $\mu_1 = 1$ we get

$$\frac{\gamma}{\mu_t^2}(f(\mathbf{x}^{(t)}) - f(\mathbf{x}^*)) + \frac{1}{2}||\mathbf{x}_u^{(t)} - \mathbf{x}^*||^2 \leq \frac{\gamma(1-\mu_1)}{\mu_1^2}(f(\mathbf{x}^{(0)}) - f(\mathbf{x}^*)) + \frac{1}{2}||\mathbf{x}_u^{(0)} - \mathbf{x}^*||^2 \leq \frac{1}{2}||\mathbf{x}^{(0)} - \mathbf{x}^*||^2$$

Hence we conclude

$$f(\mathbf{x}^{(T)}) - f(\mathbf{x}^*) \leq \frac{\mu_T^2}{2\gamma}||\mathbf{x}^{(0)} - \mathbf{x}^*||^2 = \frac{2||\mathbf{x}^{(0)} - \mathbf{x}^*||^2}{\gamma(T+1)^2}$$

# END OF EXTRA AND OPTIONAL DETAILED SLIDES ON Accelerated Generalized Gradient Descent

# Recap: Generalized Gradient Descent and its Special Cases

$$prox_\gamma(\mathbf{z}) = \text{argmin}_\mathbf{x} \frac{1}{2\gamma}||\mathbf{x} - \mathbf{z}||^2 + c(\mathbf{x})$$

It's special cases are:

1. Gradient Descent: $c(\mathbf{x}) = 0$
2. Projected Gradient Descent: $c(\mathbf{x}) = I_\mathcal{C}(\mathbf{x})$ (Example: $= \sum_i I_{g_i}(\mathbf{x})$)
3. Alternating Projection/Proximal Minimization: $f(\mathbf{x}) = 0$
4. Alternating Direction Method of Multipliers
5. Special Cases for Specific Objectives
   - LASSO: (Fast) Iterative Shrinkage Thresholding Algorithm (ISTA/FISTA)

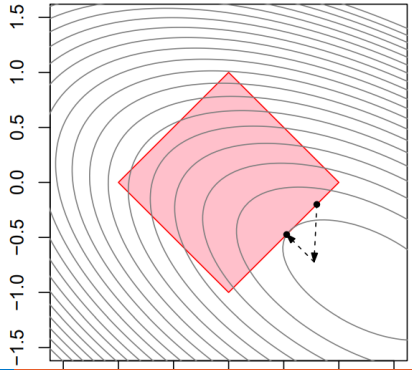
How would you solve for the prox operator in general for projected gradient descent?
(Hint: See following two slides)

# Summary of Proximal Gradient Descent

- Proximal Gradient Descent becomes gradient descent if $c = 0$
- If $f$ is 0 (i.e. no smooth function), one can minimize a non-differentiable function $c$ as long as the prox operator is easy to compute: **Proximal Point Algorithm**.
- Key to Proximal GD: Being able to compute Proximal operator.
- What if Prox can be efficiently computed approximately?
- There are papers (Schmidt et al 2011, Inexact Proximal Gradient Methods) where Prox is computed approximately but one can still derive the convergence rates if the errors due to approximation can be controlled!
- Accelerated Proximal GD: Similar to GD, one can accelerate GD to get optimal convergence rates! (Beck and Teboulle 2008)

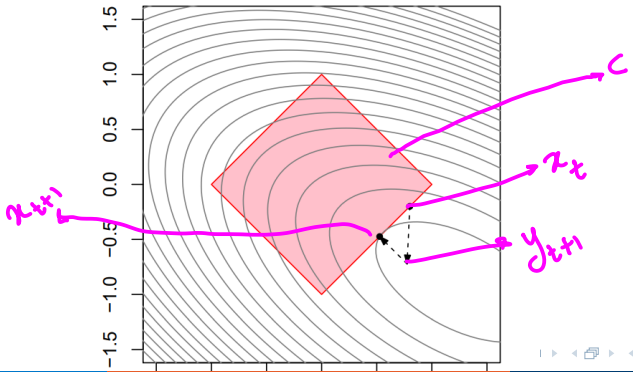
# An important special case: Projected Gradient Descent

- Consider the Problem of Constrained Convex Minimization: $\min_{x \in \mathcal{C}} f(x)$
- A simple modification of the gradient descent procedure is:
  1. At every iteration $t$: (Gradient Step): Compute $y_{t+1} = x_t - \alpha \nabla f(x_t)$
  2. (Projection step) $x_{t+1} = P_{\mathcal{C}}(y_{t+1})$
- Key here is the Projection step. Define $P_{\mathcal{C}}(x) = \text{argmin}_{y \in \mathcal{C}} \frac{1}{2}||x - y||^2$



# An important special case: Projected Gradient Descent

- Consider the Problem of Constrained Convex Minimization: $\min_{x \in \mathcal{C}} f(x)$
- A simple modification of the gradient descent procedure is:
  1. At every iteration $t$: (Gradient Step): Compute $y_{t+1} = x_t - \alpha \nabla f(x_t)$
  2. (Projection step) $x_{t+1} = P_{\mathcal{C}}(y_{t+1})$
- Key here is the Projection step. Define $P_{\mathcal{C}}(x) = \text{argmin}_{y \in \mathcal{C}} \frac{1}{2}||x - y||^2$



# Projected Gradient Descent and Proximal Gradient Descent

- There is a close connection between Proximal and Projected Gradient Descent.
- Define $c(x) = I(x \in \mathcal{C})$ where $I(.)$ is the Indicator function.
- Its easy to see that the $\text{prox}_c(x) = P_{\mathcal{C}}(x)$, i.e. the Prox operator is exactly the same as a projection operator.
- As a result, projected gradient descent becomes a special case of proximal gradient descent.
- Theoretical results of Proj. GD: All results for standard Gradient descent carry over to the projected case as long as the projection operator is easy to compute!

# Projected Gradient Descent and Proximal Gradient Descent

- There is a close connection between Proximal and Projected Gradient Descent.
- Define $c(x) = I(x \in \mathcal{C})$ where $I(.)$ is the Indicator function.
- Its easy to see that the $\text{prox}_c(x) = P_{\mathcal{C}}(x)$, i.e. the Prox operator is exactly the same as a projection operator.
- As a result, projected gradient descent becomes a special case of proximal gradient descent.
- Theoretical results of Proj. GD: All results for standard Gradient descent carry over to the projected case as long as the projection operator is easy to compute!

KKT conditions, Lagrange Dual etc

# Algorithm: Projected Gradient Descent (We use $\mathbf{x}_u^t$ instead of $\mathbf{z}^t$)

**Find** a starting point $\mathbf{x}_p^0 \in \mathcal{C}$.
Set $t = 1$
**repeat**
1. Choose a step size $\gamma_t \propto 1/\sqrt{t}$.
2. Set $\mathbf{x}_u^t = \mathbf{x}_p^{t-1} - \gamma_t \nabla f(\mathbf{x}_p^{t-1})$.
3. Set $\mathbf{x}_p^t = \text{argmin}_{\mathbf{z} \in \mathcal{C}} ||\mathbf{x}_u^t - \mathbf{z}||_2^2$.
4. Set $t = t + 1$.

**until** stopping criterion (such as $||\mathbf{x}_p^t - \mathbf{x}_p^{t-1}|| \leq \epsilon$ or $f(\mathbf{x}_p^t) > f(\mathbf{x}_p^{t-1})$) is satisfied[a]

[a] Better criteria can be found using Lagrange duality theory, such as in the form of duality gap, etc.

Figure 5: The projected gradient descent algorithm.

# Computing the Projection Operator

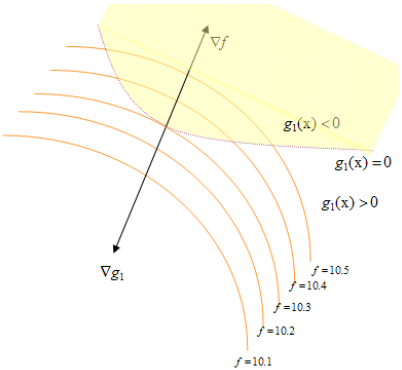



Figure 6: In this Figure $c = 0$ and $f$ is the function being optimized. See optional course material referred to at the end for understanding this figure completely

- Lets assume for simplicity that $\mathcal{C} = \{x | g(x) \leq c\}$ and $f = \frac{1}{2}||z - x||^2$ is the function being minimized
- Computing the projection step involves solving: $\min_z \left\{\frac{1}{2}||z - x||^2, \text{ such that } g(z) \leq c\right\}$.
- Use the idea of Lagrange multipliers and the KKT conditions (see subsequent slides)!
- Define $L(z, \lambda) = \frac{1}{2}||z - x||^2 + \lambda(g(z) - c)$.
- Optimality conditions are: $\nabla_z L = 0$ and $\nabla_\lambda L = 0$!
- There are two options.
  1. Either $x \in \mathcal{C}$, and the constraints are not active ($g(z) < c$), in which case we need $\nabla_z L = 0$ and $\lambda = 0$ (that is $z = x$).
  2. Or $x$ is on the boundary of $\mathcal{C}$, in which case $g(z) = c$ and $z - x + \lambda \nabla g(z) = 0$.
- If both can be solved in closed form, we are done!

# Computing the Projection Operator

From projection perspective this is simply one iteration of projection

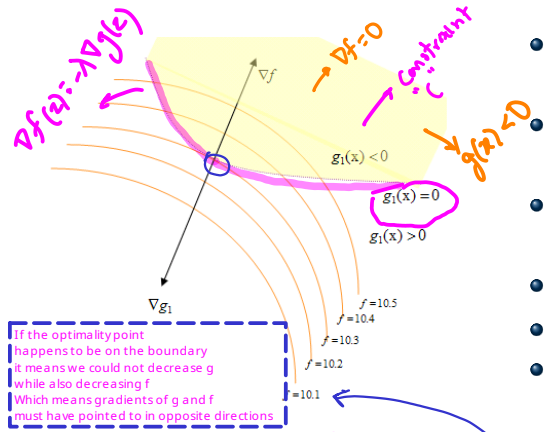



Figure 6: In this Figure $c = 0$ and $f$ is the function being optimized. See optional course material referred to at the end for understanding this figure completely

- Lets assume for simplicity that $\mathcal{C} = \{x | g(x) \leq c\}$ and $f = \frac{1}{2}||z - x||^2$ is the function being minimized
- Computing the projection step involves solving: $\min_z \left\{\frac{1}{2}||z - x||^2, \text{ such that } g(z) \leq c\right\}$.
- Use the idea of Lagrange multipliers and the KKT conditions (see subsequent slides)!
- Define $L(z, \lambda) = \frac{1}{2}||z - x||^2 + \lambda(g(z) - c)$.
- Optimality conditions are: $\nabla_z L = 0$ and $\nabla_\lambda L = 0$!
- There are two options.
  1. Either $x \in \mathcal{C}$, and the constraints are not active ($g(z) < c$), in which case we need $\nabla_z L = 0$ and $\lambda = 0$ (that is $z = x$). if the optimal point is NOt in the boundary we have the regular condition of unconstrained optimization
  2. Or $x$ is on the boundary of $\mathcal{C}$, in which case $g(z) = c$ and $z - x + \lambda \nabla g(z) = 0$. if the optimal point happens to be on the bondary
- If both can be solved in closed form, we are done!

# Computing the Projection Operator

- Optimality conditions imply: $g(z) = c$ and $z - x + \lambda \nabla g(z) = 0$. If both these can be solved in closed form, we are done!
- How to compute the Projection operators for constraints $\mathcal{C}_g = \{x \mid g(x) \leq c\}$ when
  - $g(x) = a^T x$
  - $g(x) = ||x||_2^2$
  - $g(x) = ||x||_1$
  - $g(x) = ||x - x_0||^2$
  - $g(x) = x^T A x + bx + c$
  - $g(x) = ||x||_\infty$

# Computing the Projection Operator

- Optimality conditions imply: $g(z) = c$ and $z - x + \lambda \nabla g(z) = 0$. If both these can be solved in closed form, we are done!
- How to compute the Projection operators for constraints $\mathcal{C}_g = \{x \mid g(x) \leq c\}$ when
  - $g(x) = a^T x$
  - $g(x) = ||x||_2^2$
  - $g(x) = ||x||_1$
  - $g(x) = ||x - x_0||^2$
  - $g(x) = x^T A x + bx + c$
  - $g(x) = ||x||_\infty$

$a^T x = c$ & $z - x + \lambda a = 0$ if $x$ is optimal pt on bndy

else $x = z$

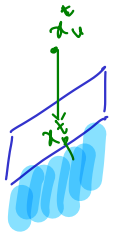
Try out these very simple cases

# Easy to Project Sets $\mathcal{C}$ (with closed form solutions)

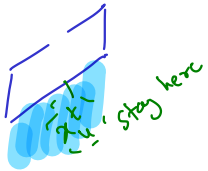
- Solution set of a linear system $\mathcal{C} = \{\mathbf{x} \in \Re^n : A^T\mathbf{x} = \mathbf{b}\}$
- Affine images $\mathcal{C} = \{A\mathbf{x} + \mathbf{b} : \mathbf{x} \in \Re^n\}$
- Nonnegative orthant $\mathcal{C} = \{\mathbf{x} \in \Re^n : \mathbf{x} \succeq 0\}$. It may be hard to project on arbitrary polyhedron.
- Norm balls $\mathcal{C} = \{\mathbf{x} \in \Re^n : \|\mathbf{x}\|_p \leq 1\}$, for $p = 1, 2, \infty$

Homework: For each of the example constrained g's stated here, apply the necessary condition for constrained optimality stated (and motivated) on the previous slide and derive the projection operation.

# Easy to Project Sets $\mathcal{C}$ (with closed form solutions)



- Solution set of a linear system $\mathcal{C} = \{\mathbf{x} \in \Re^n : A^T\mathbf{x} = \mathbf{b}\}$
- Affine images $\mathcal{C} = \{A\mathbf{x} + \mathbf{b} : \mathbf{x} \in \Re^n\}$
- Nonnegative orthant $\mathcal{C} = \{\mathbf{x} \in \Re^n : \mathbf{x} \succeq 0\}$. It may be hard to project on arbitrary polyhedron.
- Norm balls $\mathcal{C} = \{\mathbf{x} \in \Re^n : \|\mathbf{x}\|_p \leq 1\}$, for $p = 1, 2, \infty$



# Table of Orthogonal Projections: See https://archive.siam.org/books/mo25/mo25_ch6.pdf

$$P_C(\mathbf{z}) = prox_{I_C}(\mathbf{z}) = \text{argmin}_{\mathbf{x}} \frac{1}{2\gamma}||\mathbf{x}-\mathbf{z}||^2 + I_C(\mathbf{x}) = \text{argmin}_{\mathbf{x}\in C} \frac{1}{2\gamma}||\mathbf{x}-\mathbf{z}||^2$$

| Set $C =$ | For $\gamma = 1$, $P_C(\mathbf{z}) =$ | Assumptions |
|---|---|---|
| $\Re^n_+$ | $[\mathbf{z}]_+$ | |
| $\text{Box}[\mathbf{l},\mathbf{u}]$ | $P_C(\mathbf{z})_i = min\{max\{z_i, l_i\}, u_i\}$ | $l_i \leq u_i$ |
| $\text{Ball}[\mathbf{c}, r]$ | $\mathbf{c} + \frac{r}{max\{\|\mathbf{z}-\mathbf{c}\|_2, r\}}(\mathbf{z}-\mathbf{c})$ | $\|.\|_2$ ball, centre $\mathbf{c} \in \Re^n$ & radius $r > 0$ |
| $\{\mathbf{x} \| A\mathbf{x} = \mathbf{b}\}$ | $\mathbf{z} - A^T(AA^T)^{-1}(A\mathbf{z}-\mathbf{b})$ | $A \in \Re^{m\times n}$, $\mathbf{b} \in \Re^m$, $A$ is full row rank |
| $\{\mathbf{x} \| \mathbf{a}^T\mathbf{x} \leq b\}$ | $\mathbf{z} - \frac{[\mathbf{a}^T\mathbf{z}-b]_+}{\|\mathbf{a}\|^2}a$ | $0 \neq \mathbf{a} \in \Re^n$ $b \in \Re$ |
| $\Delta_n$ ($n$-simplex) | $[\mathbf{z} - \mu^*\mathbf{e}]_+$ where $\mu^* \in \Re$ satisfies $\mathbf{e}^T[\mathbf{z}-\mu^*\mathbf{e}]_+ = 1$ | |
| $H_{\mathbf{a},b} \cap \text{Box}[\mathbf{l},\mathbf{u}]$ | $P_{\text{Box}[\mathbf{l},\mathbf{u}]}(\mathbf{z}-\mu^*\mathbf{a})$ where $\mu^* \in \Re$ satisfies $\mathbf{a}^T P_{\text{Box}[\mathbf{l},\mathbf{u}]}(\mathbf{z}-\mu^*\mathbf{a}) = b$ | $0 \neq \mathbf{a} \in \Re^n$ $b \in \Re$ |
| $H^-_{\mathbf{a},b} \cap \text{Box}[\mathbf{l},\mathbf{u}]$ | $P_{\text{Box}[\mathbf{l},\mathbf{u}]}(\mathbf{z})$ $\quad \mathbf{a}^T P_{\text{Box}[\mathbf{l},\mathbf{u}]}(\mathbf{z}) \leq b$<br>$P_{\text{Box}[\mathbf{l},\mathbf{u}]}(\mathbf{z}-\lambda^*\mathbf{a})$ $\quad \mathbf{a}^T P_{\text{Box}[\mathbf{l},\mathbf{u}]}(\mathbf{z}) > b$<br>where $\lambda^* \in \Re$ satisfies $\mathbf{a}^T P_{\text{Box}[\mathbf{l},\mathbf{u}]}(\mathbf{z}-\lambda^*\mathbf{a}) = b$ & $\lambda^* > 0$ | $0 \neq \mathbf{a} \in \Re^n$ $b \in \Re$ |
| $B_{\|.\|_1}[0,\alpha]$ | $\mathbf{z}$ $\quad \|z\|_1 \leq \alpha$<br>$[\mathbf{z}-\lambda^*\mathbf{e}]_+ \odot sign(\mathbf{z})$ $\quad \|z\|_1 > \alpha$<br>where $\lambda^* > 0$, & $[\mathbf{z}-\lambda^*\mathbf{e}]_+ \odot sign(\mathbf{z}) = \alpha$ | $\alpha > 0$ |

# Table of Orthogonal Projections: See https://archive.siam.org/books/mo25/mo25_ch6.pdf

$$P_C(\mathbf{z}) = prox_{I_C}(\mathbf{z}) = \text{argmin}_{\mathbf{x}} \frac{1}{2\gamma}||\mathbf{x}-\mathbf{z}||^2 + I_C(\mathbf{x}) = \text{argmin}_{\mathbf{x}\in C} \frac{1}{2\gamma}||\mathbf{x}-\mathbf{z}||^2$$

| Set $C =$ | For $\gamma = 1$, $P_C(\mathbf{z}) =$ | Assumptions |
|---|---|---|
| $\Re^n$ | $[\mathbf{z}]_+$ | |
| $\text{Box}[\mathbf{l},\mathbf{u}]$ | $P_C(\mathbf{z})_i = min\{max\{z_i, l_i\}, u_i\}$ | $l_i \leq u_i$ |
| $\text{Ball}[\mathbf{c}, r]$ | $\mathbf{c} + \frac{r}{max\{\|\mathbf{z}-\mathbf{c}\|_2, r\}}(\mathbf{z}-\mathbf{c})$ | $\|.\|_2$ ball, centre $\mathbf{c} \in \Re^n$ & radius $r > 0$ |
| $\{\mathbf{x} \vert A\mathbf{x} = \mathbf{b}\}$ | $\mathbf{z} - A^T(AA^T)^{-1}(A\mathbf{z}-\mathbf{b})$ | $A \in \Re^{m\times n}$, $\mathbf{b} \in \Re^m$, $A$ is full row rank |
| $\{\mathbf{x} \vert \mathbf{a}^T\mathbf{x} \leq b\}$ | $\mathbf{z} - \frac{[\mathbf{a}^T\mathbf{z}-b]_+}{\|\mathbf{a}\|^2}a$ | $0 \neq \mathbf{a} \in \Re^n$ $b \in \Re$ |
| $\Delta_n$ ($n$-simplex) | $[\mathbf{z}-\mu^*\mathbf{e}]_+$ where $\mu^* \in \Re$ satisfies $\mathbf{e}^T[\mathbf{z}-\mu^*\mathbf{e}]_+ = 1$ | |
| $H_{\mathbf{a},b} \cap \text{Box}[\mathbf{l},\mathbf{u}]$ | $P_{\text{Box}[\mathbf{l},\mathbf{u}]}(\mathbf{z}-\mu^*\mathbf{a})$ where $\mu^* \in \Re$ satisfies $\mathbf{a}^T P_{\text{Box}[\mathbf{l},\mathbf{u}]}(\mathbf{z}-\mu^*\mathbf{a}) = b$ | $0 \neq \mathbf{a} \in \Re^n$ $b \in \Re$ |
| $H^-_{\mathbf{a},b} \cap \text{Box}[\mathbf{l},\mathbf{u}]$ | $P_{\text{Box}[\mathbf{l},\mathbf{u}]}(\mathbf{z})$ if $\mathbf{a}^T P_{\text{Box}[\mathbf{l},\mathbf{u}]}(\mathbf{z}) \leq b$; $P_{\text{Box}[\mathbf{l},\mathbf{u}]}(\mathbf{z}-\lambda^*\mathbf{a})$ if $\mathbf{a}^T P_{\text{Box}[\mathbf{l},\mathbf{u}]}(\mathbf{z}) > b$, where $\lambda^* \in \Re$ satisfies $\mathbf{a}^T P_{\text{Box}[\mathbf{l},\mathbf{u}]}(\mathbf{z}-\lambda^*\mathbf{a}) = b$ & $\lambda^* > 0$ | $0 \neq \mathbf{a} \in \Re^n$ $b \in \Re$ |
| $B_{\|.\|_1}[0,\alpha]$ | $\mathbf{z}$ if $\|z\|_1 \leq \alpha$; $[\mathbf{z}-\lambda^*\mathbf{e}]_+ \odot sign(\mathbf{z})$ if $\|z\|_1 > \alpha$, where $\lambda^* > 0$, & $[\mathbf{z}-\lambda^*\mathbf{e}]_+ \odot sign(\mathbf{z}) = \alpha$ | $\alpha > 0$ |

Radial direct for projection

Geometric intuitions are clear. Question is how these analytical expressions are derived?
Lagrange function $(f + \lambda g)$ and KKT conditions

# Convergence Results for Projected Gradient Descent (PGD)

- Lipschitz continuous function $f$ (C) using PGD: $R^2B^2/\epsilon^2$ iterations
- Lipschitz continuous functions + Strongly Convex $f$ (CS) using PGD: $2B^2/\epsilon - 1$ iterations
- Smooth Function $f$ using PGD: $\frac{R^2L}{\epsilon}$ iterations.
- Smooth Functions using Nestorov's PGD: $\sqrt{\frac{2R^2L}{\epsilon}}$ iterations
- Smooth + Strongly Convex $f$ (SS) using PGD: With $\gamma = 1/L$, achieve an $\epsilon$-approximate solution in $\frac{L}{\mu}\log(\frac{R^2L}{2\epsilon})$ iterations.
- Smooth + Strongly Convex $f$ (SS) using Nestorov's PGD: With $\gamma = 1/L$, achieve an $\epsilon$-approximate solution in $\sqrt{\frac{L}{\mu}}\log(\frac{R^2L}{2\epsilon})$ iterations.
- All results for standard Gradient descent carry over to the projected case **as long as the projection operator is easy to compute!**
- Proofs in monograph by Sebastian Bubeck at `https://moodle.iitb.ac.in/mod/resource/view.php?id=36947`

# How to solve the Projected Gradient Descent in closed form?

# Lagrange Function, KKT Conditions & Duality For Solving Constrained Optimizaton