CS7301: Advanced Topics in Optimization for Machine Learning

Lecture 3.2: Projected and Proximal Gradient Descent

Rishabh Iyer

Department of Computer Science
University of Texas, Dallas
https://github.com/rishabhk108/AdvancedOptML

February 5, 2021

Outline

- Proximal Gradient Descent
- Constrained Optimization: Lagrange Multipliers
- Projected Gradient Descent

Proximal Gradient Descent

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

Proximal Gradient Descent

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

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

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 $\min_x [f(x) + h(x)]$ where h is a non-differentiable function.

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 $\min_x [f(x) + h(x)]$ where h is a non-differentiable function.
- The update then becomes:

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

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 $\min_x [f(x) + h(x)]$ where h is a non-differentiable function.
- The update then becomes:

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

- After some manipulation, the update becomes:

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

Proximal Gradient Descent

- From the previous slide:

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

- Define $\text{prox}_t(x) = \text{argmin}_z \frac{1}{2t}||x - z||^2 + h(z)$
- 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:
  - $h(x) = a^T x$
  - $h(x) = c$ (a constant)
  - $h(x) = \lambda ||x||^2$
  - $h(x) = -\lambda \log(x)$ (defined only when $x \geq 0$)
  - $h(x) = \frac{1}{2} x^T A x + b^T x + c$, A is positive semi-definite
  - $h(x) = \mu x$ if $x \geq 0$ or ∞ else.
- How to compute the Prox? Solve the optimization problem:

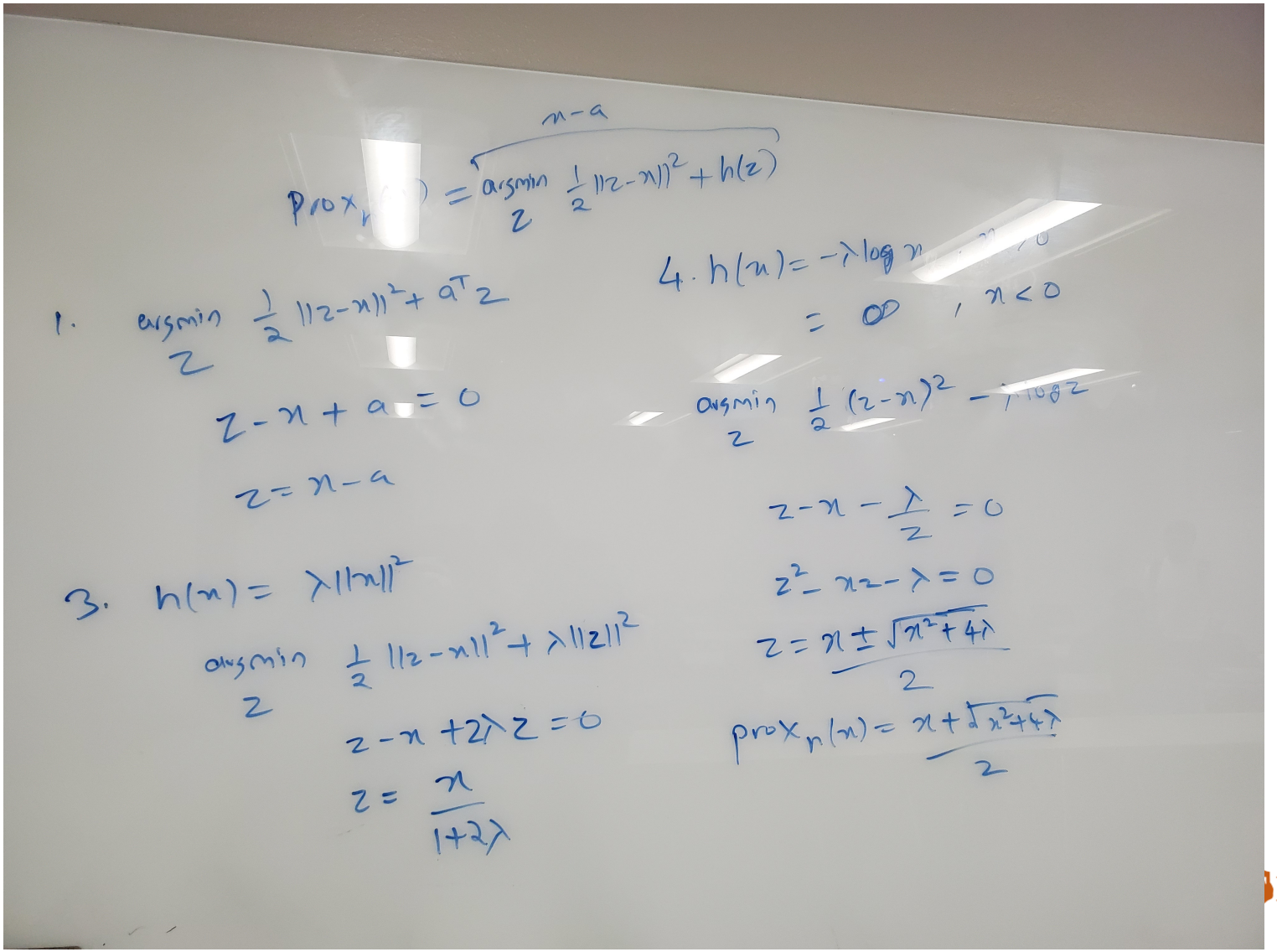
$$\text{prox}_t(x) = \text{argmin}_z \frac{1}{2t}||z - x||^2 + h(z) = \text{argmin}_z \frac{1}{2}||z - x||^2 + th(z)$$

- Using the optimality conditions (since h is convex):

$$z - x + th'(z) = 0$$

- What about if the function is non-differentiable?

How easy is it to compute the Prox operator?



Equivalent way of looking at Prox

- Some textbooks define prox as:

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

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

Main ideas for computing Prox

- The main ideas of computing Prox operator are:
  1. If $h'(x) = 0$ for a convex function h, 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.
- Try computing the Prox operator for some more functions (exercise)
  - $h(x) = \lambda x^3, x \geq 0$ and $-\infty$ otherwise.
  - $h(x) = 0$, for $x \neq 0$ and $-\lambda$ if $x = 0$
  - $h(x) = 0$, for $x \neq 0$ and λ if $x = 0$

Computing Prox when h is non-differentiable

- We want to solve: $\text{prox}_f(x) = \text{argmin}_z \frac{1}{2}||z - x||^2 + h(z)$
- If h is differentiable, we have the optimality condition as: $z - x + \nabla h(z) = 0$
- If h is non-differentiable, denote $d_h \in \partial h(z)$ as the sub-gradient. The optimality condition is $0 \in z - x + d_h(z)$
- Consider $h(z) = \lambda||z||_1$
- 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:

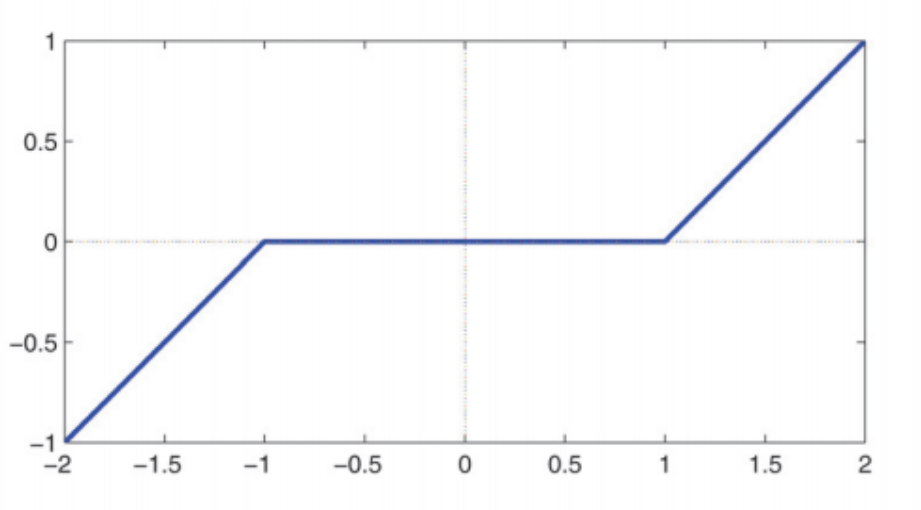
$$z_i = \begin{cases} x_i - \lambda & \text{if } x_i > \lambda \\ 0 & \text{if } -\lambda < x_i < \lambda \\ x_i + \lambda & \text{if } x_i < -\lambda \end{cases}$$

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 $f(x) = Ax - y_2^2$, $c(x) = \|x\|_1$ and $F(x) = f(x) + c(x)$
- **Proximal Subgradient Descent Algorithm:**
  Initialization: Find starting point $x^{(0)}$
  - Let $\widehat{x}^{(k+1)} \equiv z^{(k+1)}$ be a next gradient descent iterate for $f(x^k)$
  - Compute $x^{(k+1)} = \underset{x}{\textbf{argmin}} \frac{1}{2}\|x - z^{(k+1)}\|_2^2 + \lambda t\|x\|_1 = \text{Prox}_{\lambda t}(z^{(k+1)})$ by setting subgradient of this objective to 0. (We already saw that the the Prox operator is the soft-threshold)
  - Set $k = k + 1$, **until** stopping criterion is satisfied (such as no significant changes in x^k w.r.t $x^{(k-1)}$)

Iterative Soft Thresholding Algorithm (Proximal Subgradient Descent) for Lasso

- Let $f(x) = Ax - y_2^2$, $c(x) = \|x\|_1$ and $F(x) = f(x) + c(x)$
- **Proximal Subgradient Descent Algorithm:**
  Initialization: Find starting point $x^{(0)}$
  - Let $z^{(k+1)}$ be a next gradient descent iterate for $f(x^k)$
  - Compute $prox_{\|x\|_1}\left(z^{(k+1)}\right) = x^{(k+1)} =$

Iterative Soft Thresholding Algorithm (Proximal Subgradient Descent) for Lasso

- Let $f(x) = Ax - y_2^2$, $c(x) = \|x\|_1$ and $F(x) = f(x) + c(x)$
- **Proximal Subgradient Descent Algorithm:**
  Initialization: Find starting point $x^{(0)}$
  - Let $z^{(k+1)}$ be a next gradient descent iterate for $f(x^k)$
  - Compute $prox_{\|x\|_1}\left(z^{(k+1)}\right) = x^{(k+1)} = \underset{x}{\textbf{argmin}} \frac{1}{2t}\|x - z^{(k+1)}\|_2^2 + \lambda\|x\|_1$ as follows:

Iterative Soft Thresholding Algorithm (Proximal Subgradient Descent) for Lasso

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

Tables for the Proximal Operator

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

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

Tables for the Proximal Operator

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

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

Calculus for the Proximal Operator: See

https://archive.siam.org/books/mo25/mo25_ch6.pdf

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

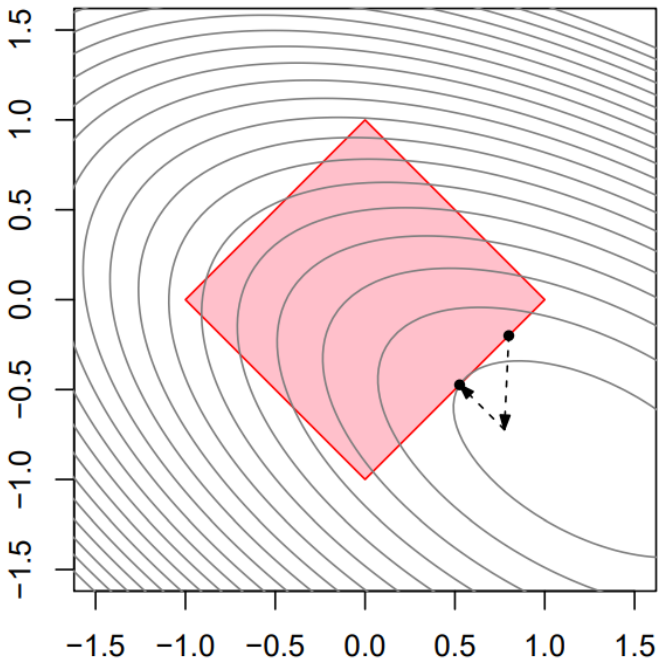
$c(x) =$	For $t = 1$, $prox_c(z) =$
Sum over components: $c(x) = \sum_{i=1}^{n} c_i(x_i)$ where $x = [x_1, x_2, \ldots, x_n]$	Product over components: $prox_c(z) = \prod_{i=1}^{n} prox_{c_i}(z_i)$ where $z = [z_1, z_2, \ldots, z_n]$
$c(\lambda x + a)$ where $\lambda \neq 0$ and c is proper	$\frac{1}{\lambda}\left[prox_{\lambda^2 c}(\lambda z + a) - a\right]$
$\lambda c\left(\frac{1}{\lambda}x\right)$ where $\lambda \neq 0$ and c is proper	$\lambda prox_{c/\lambda}\left(\frac{1}{\lambda}z\right)$
$c(x) + a^T x + \frac{\beta}{2}\|x\|^2 + \gamma$ where $\beta > 0$, $\gamma \in \Re$, c is proper	$prox_{\frac{1}{\beta+1}c}\left(\frac{z-a}{\gamma+1}\right)$
$c(Ax + b)$ where c is proper closed and convex, $b \in \Re^n$, $AA^T = \alpha I$, $\alpha > 0$	$z + \frac{1}{\alpha}A^T\left(prox_{\alpha c}\left(Az + b\right) - Az - b\right)$
$c(\|x\|)$ where $b \in \Re^n$, $AA^T = \alpha I$, $\alpha > 0$	$prox_c(\|z\|)\frac{z}{\|z\|}$ $\quad z \neq 0$ $\{u \mid \|u\| = prox_c(0)\}$ $\quad z = 0$

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)

Detour: 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 $h(x) = I(x \in \mathcal{C})$ where $I(.)$ is the Indicator function.
- Its easy to see that the $\text{prox}_h(x) = P_{\mathcal{C}}(x)$, 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!

Algorithm: Projected Gradient Descent (We use x_u^k instead of z^k)

Find a starting point $x_p^0 \in \mathcal{C}$.
Set $k = 1$
repeat

1. Choose a step size $t^k \propto 1/\sqrt{k}$.
2. Set $x_u^k = x_p^{k-1} - t^k \nabla f(x_p^{k-1})$.
3. Set $x_p^k = \underset{z \in \mathcal{C}}{\operatorname{argmin}} \, ||x_u^k - z||_2^2$.
4. Set $k = k + 1$.

until stopping criterion (such as $||x_p^k - x_p^{k-1}|| \leq \epsilon$ or $f(x_p^k) > f(x_p^{k-1})$ is satisfied[1]

Figure 1: The projected gradient descent algorithm.

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

Computing the Projection Operator

- Lets assume for simplicity that $\mathcal{C} = \{x | f(x) \leq c\}$
- Computing the projection step involves solving: $\min_z \{\frac{1}{2}||z - x||^2, \text{ such that } f(z) \leq c\}$.
- Use the idea of Lagrange multipliers!
- Define $g(z, \lambda) = \frac{1}{2}||z - x||^2 + \lambda(f(z) - c)$.
- Optimality conditions are: $\nabla_z g = 0$ and $\nabla_\lambda g = 0$!
- There are two options. Either $x \in \mathcal{C}$, in which case the constraints are not active, or x is outside $\mathcal{C}$ in which case we need $\nabla_z g = 0$ and $\nabla_\lambda g = 0$.
- The second case implies: $f(z) = c$ and $z - x + \lambda \nabla f(z) = 0$. If both these can be solved in closed form, we are done!

Computing the Projection Operator

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

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

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

Table of Orthogonal Projections: See

https://archive.siam.org/books/mo25/mo25_ch6.pdf

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

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