Eg: Projected Gradient Descent

- Let

$$dist(x, C_i) = \min_{u \in C_i} \|x - u\|^2$$

- We define

$$c(x) = D(x) = \max_i dist(x, C_i)$$

  - If C_i is closed and convex, a unique minimizer $P_{C_i}(x)$ exists (projection of x on C_i)
  - $dist(x, C_i) = 0$ if $x \in C_i$

- Recall discussion on subgradient descent for this problem in class notes[4]

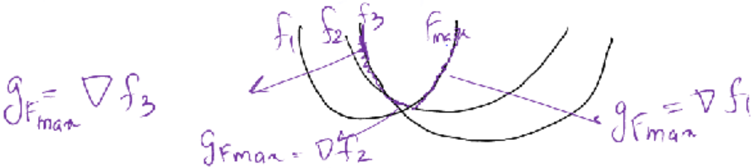
[4] http://www.cse.iitb.ac.in/~cs709/notes/enotes/lecture22a.pdf

- We get the subgradient of $D(x)$ as

$$g_D(x) = \nabla dist(x, C_i) \text{ if } D(x) = dist(x, C_i)$$

- For illustration, consider

$$g_{F_{max}}(x) = \nabla f_i(x) \text{ if } f_i(x) = \max_j f_j(x)$$



  - If f_i gives maximum value at a point, $g_{F_{max}}$ will be ∇f_i at that point
  - At the points of intersection of f_i and f_j, we will get some convex combination of ∇f_i and ∇f_j

Projection methods

- So far, we have dealt with simple projections during SMO and the general decomposition method
  - We considered $\alpha_i y_i + \alpha_j y_j = constant$, and solved a quadratic optimization problem for α_i and α_j
  - We then projected $(\alpha_i, \alpha_j) \to [0, C]^2$
- We will now 'scale up' these projections
- In active set methods, the working set changes slowly. Projection methods can solve bound constrained optimization problems with large changes in the working set at each iteration.

Overview

$x^k - t\nabla f(x^k)$

- We can find Δx as the change in x along some steepest descent direction of f without constraints
- Thus, let $x_u^{k+1} = x^k + \Delta x$ be the working set that reduces $f(x)$ without constraints (unbounded)
- To find the constrained working set, we project x_u^{k+1} onto Ω to get x^{k+1}

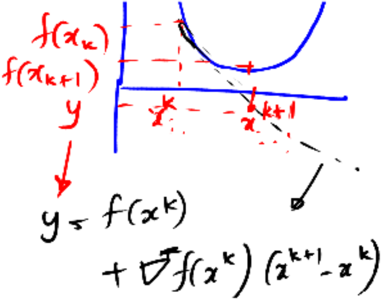
- To project x_u onto the non-empty closed convex set Ω to get the projected point x_p, we solve:

$$x_p = P_\Omega(x_u) = \underset{z \in \Omega}{\operatorname{argmin}} \|x_u - z\|_2^2$$

- That is, the projected point x_p is the point in Ω that is the closest to the unbounded optimal point x_u if Ω is a non-empty closed convex set

Descent direction for a convex function

- For a descent in a convex function f, we must have $f(x^{k+1}) \geq$ Value at x^{k+1} obtained by linear interpolation from x^k
- *ie.* $f(x^{k+1}) \geq f(x^k) + \nabla^\top f(x^k)(x^{k+1} - x^k)$



- Thus, for Δx^k to be a descent direction, it is necessary that $\nabla^\top f(x^k)\Delta x^k \leq 0$ (where $\Delta x^k = x^{k+1} - x^k$)

We want that the point obtained after the projection of x_u^{k+1} to be a descent direction from x^k for the function f

$$\nabla f(x^k) \cdot \Delta x_p \leq 0$$

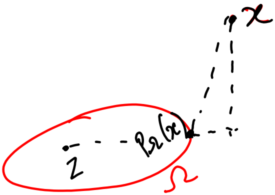
(where $\Delta x_p = P_\Omega(x_u^{k+1}) - x^k$)
You can prove this (necessary condition) for a convex $f(x)$ using the following result...

Ω is assumed to be convex

- **Claim:** $P_\Omega(x)$ is a projection of x, **iff**

$$\left(z - P_\Omega(x)\right)^\top \left(x - P_\Omega(x)\right) \leq 0, \; \forall z \in \Omega$$

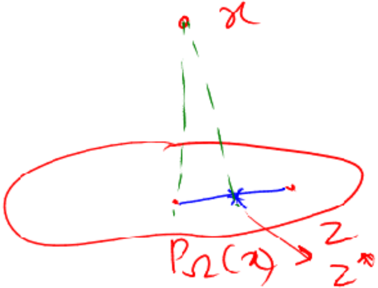
- That is, the angle between $\left(z - P_\Omega(x)\right)$ and $\left(x - P_\Omega(x)\right)$ is obtuse (or right-angled for the projected point), $\forall z \in \Omega$



If x lies "right above" Ω, then $(z - P_\Omega(x))^T (x - P_\Omega(x)) = 0$

Proof for $\langle z - P_\Omega(x), x - P_\Omega(x) \rangle \leq 0$

- To be more general, let us consider an inner product $\langle a, b \rangle$ instead of $a^\top b$
- Let $z^* = (1-\alpha)P_\Omega(x) + \alpha z$, for some $\alpha \in (0,1)$, and $z \in \Omega$
  $\implies z^* = P_\Omega(x) + \alpha(z - P_\Omega(x))$, $z^* \in \Omega$



- Since $P_\Omega(x) = \text{argmin}_{z\in\Omega} \|x - z\|_2^2$,
  $\left\|x - P_\Omega(x)\right\|^2 \leq \|x - z^*\|^2$

$$\begin{aligned}
&\|x - z^*\|^2 \\
&= \left\| x - \left(P_\Omega(x) + \alpha(z - P_\Omega(x))\right) \right\|^2 \\
&= \left\| x - P_\Omega(x) \right\|^2 + \alpha^2 \left\| z - P_\Omega(x) \right\|^2 - 2\alpha \left\langle x - P_\Omega(x), z - P_\Omega(x) \right\rangle \\
&\geq \left\| x - P_\Omega(x) \right\|^2
\end{aligned}$$

$$\implies \left\langle x - P_\Omega(x), z - P_\Omega(x) \right\rangle \leq \frac{\alpha}{2} \left\| z - P_\Omega(x) \right\|^2, \ \forall \alpha \in (0,1)$$

- Thus, the LHS can either be 0 or a negative value. Any positive value of the LHS will lead to a contradiction for some small $\alpha \to 0$
- Hence, we proved that $\left\langle z - P_\Omega(x), x - P_\Omega(x) \right\rangle \leq 0$

Proof of sufficiency:

- We can also prove that if $\langle x - x^*, z - x^* \rangle \leq 0, \forall z \in \Omega$ s.t. $z \neq x^*$, and $x^* \in \Omega$, then

$$x^* = P_\Omega(x) = \underset{\bar{z} \in \Omega}{\operatorname{argmin}} \|x - \bar{z}\|_2^2$$

- Consider $\|x - z\|^2 - \|x - x^*\|^2$
$= \left\|x - x^* + (x^* - z)\right\|^2 - \|x - x^*\|^2$
$= \|x - x^*\|^2 + \|z - x^*\|^2 - 2\langle x - x^*, z - x^* \rangle - \|x - x^*\|^2$
$= \|z - x^*\|^2 - 2\langle x - x^*, z - x^* \rangle$
> 0
- $\implies \|x - z\|^2 > \|x - x^*\|^2, \forall z \in \Omega$ s.t. $z \neq x^*$
- This proves that $x^* = P_\Omega(x)$

References

- Yu-Hong Dai, Roger Fletcher. New algorithms for singly linearly constrained quadratic programs subject to lower and upper bounds. http://link.springer.com/content/pdf/10.1007%2Fs10107-005-0595-2.pdf

Quadratic Optimization: Primal Active-Set Algorithm

I_k = index set of constraints active in in k^{th} iteration

$\forall i \in I_k \quad a_i^T x^k = b_i$

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{c}^T\mathbf{x} + \beta \\ \text{subject to} & A\mathbf{x} \geq \mathbf{b} \rightarrow \{a_i^T x \geq b_i\} \end{array} \tag{1}$$

where $Q \succ 0$.

- How to evolve I_{k+1} from I_k?
- How to check whether to stop?
- How to initalize I_0? i.e. $a_i^T x^{(0)} = b_i \ \forall i \in I_0$
- Need to ensure that $\forall i \notin I_k, \ a_i^T x^{k+1} \geq b_i$ Else project!
-

Quadratic Optimization: Primal Active-Set Algorithm

Consider the quadratic optimization problem

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{c}^T\mathbf{x} + \beta \\ \text{subject to} & A\mathbf{x} \geq \mathbf{b} \end{array} \quad (1)$$

where $Q \succ 0$.

(1) Assume I_0 & $x^{(0)}$ obtained using interior point method (next)

(2) $x^{k+1} = \arg\min \frac{1}{2}x^TQx + c^Tx + \beta$ s.t. $a_i^T x = b_i \ \forall\, i \in I_k$ $\equiv (x^{k+1} - x^k) = d^k = \arg\min_d \frac{1}{2}d^TQd + g_k^Td$ s.t. $a_i^Td = 0 \ \forall i \in I_k$

$a_i^T x_{k+1} = b_i$ & $a_i^T x_k = b_i$

$g_k = Qx^k + c$ (just like in conjugate gradient)

unconstrained update (wrt to) inequalities of I_k^c

(3) Find α^k = step to take s.t. if α^k violates $a_i^T x^{k+1} \geq b_i$ for any $i \notin I_k$ then retract α^k s.t. all constraints are satisfied

$\alpha^k = \arg\min_\alpha |\alpha|$ s.t. $A(\alpha d^k) \geq 0$ → The projection step

(4) Convergence: in terms of KKT conditions!

Step 1
Input a feasible point, $\mathbf{x}^0$, identify the active set $\mathcal{I}^0$, form matrix $A_{\mathcal{I}^0}$, and set $k = 0$.
Step 2
Compute $\mathbf{g}^k = Q\mathbf{x}^k + \mathbf{c}$.
Check the rank condition $rank[A^T_{\mathcal{I}^k} \;\; \mathbf{g}^k] = rank[A^T_{\mathcal{I}^k}]$. If it does not hold, go to **Step 4**.

KKT conditions!

Step 3
Solve the system $A^T_{\mathcal{I}^k}\widehat{\lambda} = \mathbf{g}^k$. If $\widehat{\lambda} \geq \mathbf{0}$, output $\mathbf{x}^k$ as the solution and stop; otherwise, remove the index that is associated with the most negative Lagrange multiplier (some $\widehat{\lambda}_t$) from $\mathcal{I}^k$.
Step 4
Compute the value of $\mathbf{d}^k$:

$$\begin{aligned} \mathbf{d}^k = \quad & \underset{\mathbf{d}}{\text{argmin}} & & \tfrac{1}{2}\mathbf{d}^T Q\mathbf{d} + (\mathbf{g}^k)^T\mathbf{d} & & \\ & \text{subject to} & & \mathbf{a}_i^T\mathbf{d} = 0 & & \text{for } i \in \mathcal{I}^k \end{aligned} \tag{2}$$

Step 5
Compute α_k:

Projection step

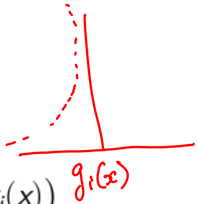
$$\alpha_k = \min\left\{1, \min_{\substack{j \notin \mathcal{I}^k \\ \mathbf{a}_j^T\mathbf{d}^k < 0}} \frac{\mathbf{a}_j^T\mathbf{x}^k - b_j}{-\mathbf{a}_j^T\mathbf{d}^k}\right\} \tag{3}$$

Set $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k\mathbf{d}^k$.
Step 6
If $\alpha_k < 1$, construct $\mathcal{I}^{k+1}$ by adding the index that yields the minimum value of α_k in (**??**). Otherwise, let $\mathcal{I}^{k+1} = \mathcal{I}^k$.
Step 7
Set $k = k + 1$ and repeat from **Step 2**.

Figure 1: Optimization for the quadratic problem in (**??**) using Primal Active-set Method.

Option 2: Log barrier function

min $f(x)$
s.t. $g_i(x) \le 0$ & $Ax = b$



- The log barrier function is defined as

$$B(x) = \phi_{g_i}(x) = -\frac{1}{t}\log\left(-g_i(x)\right)$$

- It looks like an approximation of $\sum I_{C_i}(x)$
- $f(x) + \sum_i \phi_{g_i}(x)$
  is convex if f and g_i are convex
- We've taken care of the inequality constraints, lets also consider an equality constraint $Ax = b$

- Our objective becomes

$$\min_x f(x) + \sum_i \left(-\frac{1}{t}\right) \log\left(-g_i(x)\right)$$

s.t. $Ax = b$

- At different values of t, we get different $x^*(t)$
- Let $\lambda_i^*(t) =$
- First-order necessary conditions for optimality (and strong duality) at $x^*(t), \lambda_i^*(t)$
  -
  -
  -
  -
  -
-

If $(x^*(t), \mu^*(t))$ was obtained by solving Barrier augmented problem without $g_i(x) \leq 0$ but with $Ax = b$

& if $\lambda_i^*(t) = \frac{1}{t g_i(x^*(t))}$ & $x^*(t)$ & $\mu^*(t)$ satisfy KKT conditions, we have converged!

- Our objective becomes

$$\min_x f(x) + \sum_i \left(-\frac{1}{t}\right) \log\left(-g_i(x)\right)$$

$$\text{s.t. } Ax = b$$

- At different values of t, we get different x^*
- Let $\lambda_i^*(t) = \frac{-1}{t\, g_i\left(x^*(t)\right)}$
- First-order necessary conditions for optimality (and strong duality) at $x^*(t), \lambda_i^*(t)$:
  - $g_i\left(x^*(t)\right) \leq 0$
  - $Ax^*(t) = b$
  - $\nabla f\left(x^*(t)\right) + \sum_{i=1}^m \lambda_i^*(t) \nabla g_i\left(x^*(t)\right) + \nu^*(t)^\top A = 0$
  - $\lambda_i^*(t) \geq 0$
    - Since $g_i\left(x^*(t)\right) \leq 0$ and $t \geq 0$
- $\left(\lambda_i^*(t), \nu^*(t)\right)$ is dual feasible

- If necessary conditions are satisfied and if f and g_i's are convex, and g_i's strictly feasible, they are also sufficient. Thus, $(x^*(t), \lambda_i^*(t), \nu^*(t))$ form a saddle point for the Lagrangian

$$L(x, \lambda, \nu) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \nu^\top (Ax - b)$$

- Lagrange dual function

$$L^*(\lambda, \nu) = \min_x L(x, \lambda, \nu)$$

$$L^*\left(\lambda^*(t), \nu^*(t)\right) = f\left(x^*(t)\right) + \sum_{i=1}^{m} \lambda_i^*(t) g_i\left(x^*(t)\right) + \nu^*(t)^\top \left(Ax^*(t) - b\right)$$
$$= f(x^*(t)) - m/t$$

  - m/t is the *duality gap*
  - As $t \to \infty$, duality gap $\to 0$

- If necessary conditions are satisfied and if f and g_i's are convex, and g_i's strictly feasible, they are also sufficient. Thus, $\left(x^*(t), \lambda_i^*(t), \nu^*(t)\right)$ form a saddle point for the Lagrangian

$$L(x, \lambda, \nu) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \nu^\top (Ax - b)$$

- Lagrange dual function

$$L^*(\lambda, \nu) = \min_x L(x, \lambda, \nu)$$

$$L^*\left(\lambda^*(t), \nu^*(t)\right) = f\left(x^*(t)\right) + \sum_{i=1}^{m} \lambda_i^*(t) g_i\left(x^*(t)\right) + \nu^*(t)^\top \left(Ax^*(t) - b\right)$$

$$= f\left(x^*(t)\right) - \frac{m}{t}$$

  - $\frac{m}{t}$ here is called the *duality gap*
  - As $t \to \infty$, duality gap $\to 0$

- At optimality, primal optimal = dual optimal
  i.e. $p^* = d^*$
- From weak duality,

$$f\left(x^*(t)\right) - \frac{m}{t} \leq p^*$$

$$\implies f\left(x^*(t)\right) - p^* \leq \frac{m}{t}$$

  - The duality gap is always $\leq \frac{m}{t}$
  - The more we increase t, the smaller will be the duality gap

Iterative algorithm

1. Start with $t = t^{(0)}$, $\mu > 1$, and consider ϵ tolerance
2. **Repeat**
    1. **Solve**

$$x^*(t) = \underset{x}{\operatorname{argmin}}\, f(x) + \sum_{i=1}^{m} \left(-\frac{1}{t}\right) \log\left(-g_i(x)\right)$$

$$\text{s.t.}\quad Ax = b$$

Dual ascent
ADMM
Boyd: Newton

    2. If $\frac{m}{t} < \epsilon$, **Quit**
       else, **set** $t = \mu t$

- In the process, we can also obtain $\lambda^*(t)$ and $\nu^*(t)$
- **Convergence of outer iterations:**
  We get ϵ accuracy after $\log\left(\frac{(m/\epsilon t^{(0)})}{\log(\mu)}\right)$ updates of t

- The inner optimization in the iterative algorithm using a barrier method,

$$x^*(t) = \underset{x}{\operatorname{argmin}}\, f(x) + \sum_i \left(-\frac{1}{t}\right) \log\left(-g_i(x)\right)$$

$$\text{s.t. } Ax = b$$

can be solved using (sub)gradient descent starting from older value of x from previous iteration

- We must start with a strictly feasible x, otherwise $-\log\left(-g_i(x)\right) \to \infty$

- We need not obtain $x^*(t)$ exactly at each outer iteration
- If not solving for $x^*(t)$ exactly, we will get ϵ accuracy after *more than* $\log\left(\frac{\left(m/\epsilon t^{(0)}\right)}{log(\mu)}\right)$ updates of t
    - However, solving the inner iteration exactly may take too much time
    - Fewer inner loop iterations correspond to more outer loop iterations

How to find a strictly feasible $x^{(0)}$?

Soln:

$$\min \Gamma$$
$$\text{s.t. } g_i(x) \leq \Gamma \quad \forall i$$
$$Ax = b$$

} Solve using Interior pt

for some $\Gamma < 0$

Eg: $\Gamma = \max_i g_i(x^{rand})$

How to find a strictly feasible $x^{(0)}$?

- *Basic Phase I method*

$$x^{(0)} = \underset{x}{\text{argmin}}\,\Gamma$$

$$\text{s.t. } g_i(x) \leq \Gamma$$

- We solve this using the barrier method, and thus will also need a strictly feasible starting $\hat{x}^{(0)}$
- Here,

$$\Gamma = \max_{i=1\ldots m} g_i(\hat{x}^{(0)}) + \delta$$

where, $\delta > 0$

  - *i.e.* Γ is slightly larger than the largest $g_i(\hat{x}^{(0)})$

- On solving this optimization for finding $x^{(0)}$,
  - If $\Gamma^* < 0$, $x^{(0)}$ is strictly feasible
  - If $\Gamma^* = 0$, $x^{(0)}$ is feasible (but not strictly)
  - If $\Gamma^* > 0$, $x^{(0)}$ is not feasible
- A slightly 'richer' problem can consider different Γ_i for each g_i, to improve numerical precision

$$x^{(0)} = \operatorname*{argmin}_x \Gamma_i$$

$$\text{s.t. } g_i(x) \leq \Gamma_i$$

Choice of a good $\hat{x}^{(0)}$ or $x^{(0)}$ depends on the nature/class of the problem, use domain knowledge to decide it