Optimisation Refreshment

Javier Zazo

Summary of Optimisation Techniques

Outline

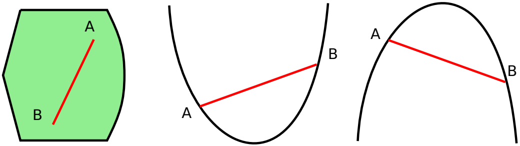
Motivation

Centralized Algorithms

Decentralized Algorithms

Convex Problem

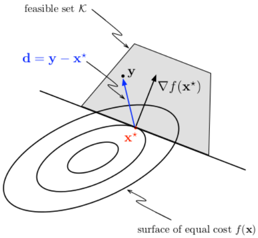
$$\min f(x)$$
$$s.t.\ g(x) \leq 0$$
$$h(x) = 0$$



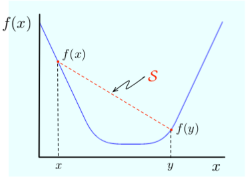
Existence of Solutions

- Local and global optima: any locally optimal point of a convex problem is globally optimal.

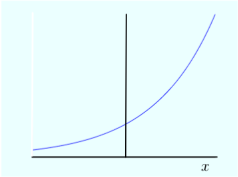
$$(\mathbf{y} - \mathbf{x}^\star)^T \nabla_x f(\mathbf{x}^\star) \geq 0, \quad \forall \mathbf{y} \in S$$



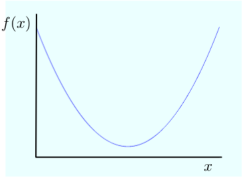
  - Convex $\Rightarrow$ Multiple Solutions (convex set)
  - Strictly Convex $\Rightarrow$ 1 solution (at most)
  - Strongly Convex $\Rightarrow$ Unique solution



(a)



(b)



(c)

Methods to solve CP

$$\min f(x)$$
$$s.t.\ g(x) \leq 0$$
$$h(x) = 0$$

1. Interior Point Methods $\Rightarrow$ Numeric
2. Projected Gradient Method $\Rightarrow$Reformulations
3. Lagrangian methods
   - 3.1 Dual Subgradient Method $\equiv$ Gauss-Seidel
   - 3.2 Primal-Dual Subgradient Method$\equiv$ Approximate Saddle Point

Unconstrained Convex

1. Descent Algorithms $\Rightarrow$ find a direction
    1.1 with Line search? $\Rightarrow$ Implies solving a one-dimensional optimization problem
        1.1.1 exactly $\equiv$ analytically
        1.1.2 backtracking
2. Steepest Descent $\Rightarrow$ Normalized Steepest Descent
    2.1 **Gradient descent** is particular to the Euclidean norm
3. Newton's Method

Constrained Convex

Only with Equality Constraints

1. Quadratic Problems are the simplest to solve
   1.1 KKT conditions form a set of linear equations
2. Newthon's Method $\Rightarrow$ Form a sequence of linear equality constrained QP

Also with Inequality Constraints $\Rightarrow$ Interior Point Methods

$$
\begin{aligned}
&\min f(x) &&\iff \min f(x) + I_{-}(g(x)) \\
&s.t.\ g(x) \leq 0 && \qquad s.t.\ h(x) = 0
\end{aligned}
$$

where $I_{-}(u) = \begin{cases} 0 & u \leq 0 \\ \infty & u > 0 \end{cases}$.

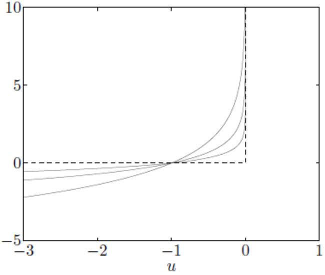
Interior Point Methods

We can approximate with:

$$\hat{I}_{-}(u) = -\frac{1}{t}\log(-u)$$

and solve for some t with Newton's method

$$\min f(x) - \frac{1}{t}\log(-g(x))$$
$$\text{s.t.}\, h(x) = 0$$



Algorithm 11.1 *Barrier method.*

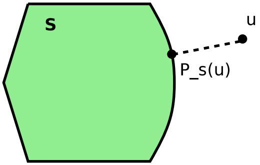
given strictly feasible x, $t := t^{(0)} > 0$, $\mu > 1$, tolerance $\epsilon > 0$.

repeat

1. *Centering step.*
   Compute $x^\star(t)$ by minimizing $tf_0 + \phi$, subject to $Ax =$
2. *Update.* $x := x^\star(t)$.
3. *Stopping criterion.* **quit if** $m/t < \epsilon$.
4. *Increase* t. $t := \mu t$.

Projected Gradient Method

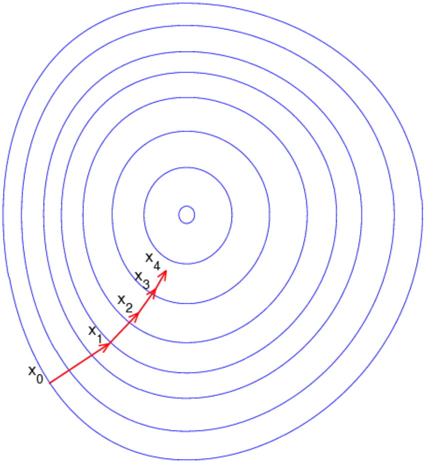
$$x_i^{k+1} = P_S\left(x_i^k - \alpha \frac{\partial f(x)}{\partial x_i}\right), \quad i = 1, \dots N$$

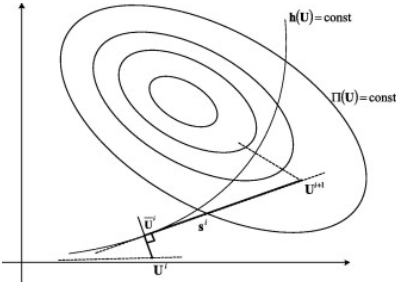


$$P_S(u) = \quad \min_x \|u - x\|_2^2 \quad \text{s.t. } g(x) \leq 0$$

$$P_S(u) \equiv \hat{x} = \arg\min_{x \in S} \|u - x\|_2^2$$

Projected Gradient





Lagrangian Methods

$$L(x,\mu) = f(x) + \mu g(x)$$

- Dual problem $\Rightarrow$ Solves the primal problem if strong duality holds

$$\max_{\lambda}\left[\min_{x} L(x,\mu)\right] = \min_{x} f(x)$$
$$\text{s.t.}\, g(x) \leq 0$$

- Slater condition $\Rightarrow$ sufficient (but not necessary) for strong duality

$$\exists \overline{x} \in \mathbb{R}^n | g(\overline{x}) < 0$$

Dual Subgradient

$$\max_{\mu \geq 0} \min_{x} f(x) + \mu g(x)$$

$$\max_{\mu \geq 0} q(\mu)$$

where $q(\mu) = \min_x f(x) + \mu g(x)$

- Due to concavity of $q(\mu)$ we can apply gradient ascent:

$$\mu_{k+1} = [\mu_k + \alpha g_k]^+$$

where

$$x_k = \arg\min_{x} f(x) + \mu_k g(x)$$

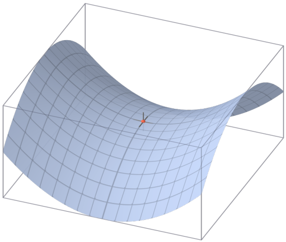
- Strict convexity of $q(\mu)$ is requirred for convergence

Lagrangian

$$L(x,\lambda) = f(x) + \mu g(x)$$

Dual problem $\Rightarrow$ Solves the primal problem if strong duality holds

$$\max_{\lambda}\left[\min_{x} L(x,\lambda)\right]$$



Gauss-Seidel Algorithm $\Rightarrow$ Game

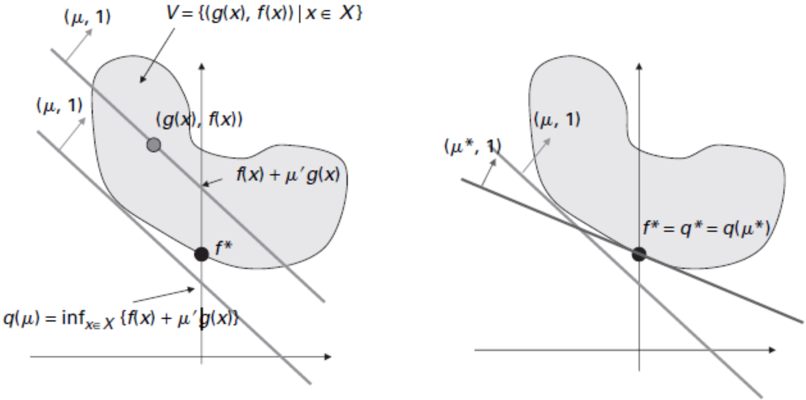
1. Initialize μ
2. Solve $q(\mu) = \min_x L(x,\mu)$
3. Update $\mu(k+1) = \mu(k) + \alpha g_k$
4. Repeat steps 2-3 till convergence

Drawback: Step 2 can be costly, and difficult to solve in a distributed fashion
Note: The Lagrangian can be augmented to solve non strictly convex problems (proximal methods).

Graphical Interpretation (I)

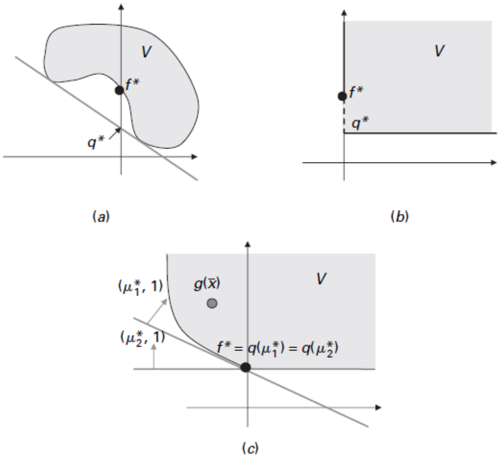
Constraint-cost function set:

$$V = \{(g(x), f(x) | x \in X)\}$$



$$f(x) - L(x, \mu) = \mu^{\top}(g(x) - g^{*}(x))$$

Graphical Interpretation (II)



Parts (a) and (b) provide two examples where there is a duality gap [due to lack of convexity in (a) and lack of "continuity around origin" in (b)]. Part (c) illustrates the role of the Slater condition in establishing no duality gap and boundedness of the dual-optimal solutions. Note that dual-optimal solutions correspond to the normal vectors of the (nonvertical) hyperplanes supporting set V from below at the point $(0, q^*)$.

Primal-Dual Subgradient Method

Again, we assume strong duality for

$$L(x,\mu) = f(x) + \mu g(x)$$

$$\max_{\lambda} \left[\min_{x} L(x,\mu) \right]$$

Algorithm:

$$x_{k+1} = P_x \left[x_k - \alpha \partial_x L(x_k, \mu_k) \right]$$
$$\mu_{k+1} = \left[\mu_k + \alpha g_k \right]^+$$

First proposed by Uzawa in 1958.
These TWO algorithms REQUIRE strict convexity for convergence

Variants of previous Algorithms

- We DROP the strict convexity assumption
- Nedic: compute the averages of previous algorithms
- Case 1

$$x_k = \arg\min_x f(x) + \mu_k g(x)$$
$$\mu_{k+1} = [\mu_k + \alpha g_k]^+$$

$$\hat{x}_k = \frac{1}{k}\sum_{i=0}^{k} x_i$$

- Case 2

$$x_{k+1} = P_x[x_k - \alpha \partial_x L(x_k, \mu_k)]$$
$$\mu_{k+1} = [\mu_k + \alpha g_k]^+$$

$$\hat{x}_k = \frac{1}{k}\sum_{i=0}^{k} x_i, \qquad \hat{\mu}_k = \frac{1}{k}\sum_{i=0}^{k} \mu_i$$

Variants of previous Algorithms

- Augmented Lagrangian

$$L(x,\mu) = f(x) + \mu g(x) + \psi(g(x))$$

where $\psi(u) = \begin{cases} \rho u^2 & u > 0 \\ 0 & u \leq 0 \end{cases}$

- Another version

$$L(x,\overline{x},\mu) = f(x) + \mu g(x) + \frac{\rho}{2}\|x - \overline{x}_k\|^2$$

$$x_{k+1} = P_x[x_k - \alpha\partial_x L(x_k,\mu_k) + \rho\alpha(x_k - \overline{x}_k)]$$
$$\overline{x}_{k+1} = \overline{x}_k + \alpha(x_k - \overline{x}_k)$$
$$\mu_{k+1} = [\mu_k + \alpha g]^+$$

Alternating Direction Method of Multipliers

Separable problem:

$$\min f(x) + g(z)$$
$$\text{s.t.} Ax + Bz = c$$

$$L_\rho(x, z, \mu) = f(x) + g(z) + \mu^\top (Ax + Bz - c) + \frac{\rho}{2} \|Ax + Bz - c\|^2$$

Solution:

$$x_{k+1} = \arg\min_x L_\rho(x, z_k, \mu_k)$$
$$z_{k+1} = \arg\min_z L_\rho(x_{k+1}, z, \mu_k)$$
$$\mu_{k+1} = \mu_k + \rho(Ax_{k+1} + Bz_{k+1} - c)$$

Separable Problems

Decentralized Dual Algorithms

$$\min_{x_i} \sum_i f_i(x_i)$$
$$\text{s.t.}\, g(x) \leq 0$$

and separable derivatives $\frac{\partial g}{\partial x_i} = h(x_i)$.

- With the Lagrangian formulation, it becomes a separable problem

$$L(x,\mu) = \sum_i f_i(x_i) + \mu g(x)$$

1st option

$$x_i(k+1) = \arg\min_{x_i} L(x, \mu(k)), \qquad \forall i$$
$$\mu(k+1) = \mu(k) + \alpha g(x_i(k+1))$$

2nd option

$$x_i(k+1) = x_i(k) - \alpha \partial_x L(x, \mu_i(k)), \qquad \forall i$$
$$\mu(k+1) = \mu(k) + \alpha g(x_i(k+1))$$

Non-separable Problems

Consensus / Diffusion

- Unconstrained

$$\min \sum_{i=1}^{M} f_i(x)$$
$$\text{s.t.} x \in \mathbb{R}^n$$

Solution:

$$x_i(k) = \underbrace{\sum_{j=1}^{M} a_{ij} x_j(k)}_{\text{consensus step}} - \underbrace{\alpha \partial_x f_i(x_i(k))}_{\text{subgradient step}}, \qquad \forall i$$

- Constrained

$$\min \sum_{i=1}^{M} f_i(x)$$
$$\text{s.t.} x \in \cap_i X_i$$

Solution

$$x_i(k) = P_{X_i}\left[\underbrace{\sum_{j=1}^{M} a_{ij} x_j(k)}_{\text{consensus step}} - \underbrace{\alpha \partial_x f_i(x_i(k))}_{\text{subgradient step}}\right], \qquad \forall i$$

- More variants with the averages, as seen before

Our Contribution

Applicable to Non-separable Problems with constraints $\Rightarrow$ Dual solutions

$$\min \sum_{i=1}^{M} f_i(x)$$
$$\text{s.t.} \, g(x) \leq 0$$

$$x_i(k+1) = \underbrace{\sum_j a_{ij} x_j(k)}_{\text{consensus step}} - \underbrace{\alpha \partial_x f_i(x_i(k))}_{\text{subgradient step}}, \qquad \forall i$$
$$\mu_i(k+1) = \sum_j a_{ij} \mu_j(k) + \alpha g(x_i(k))$$

Seems to work...
2nd Idea

$$y_i(k+1) = \arg\min_x f_i(x) + \mu_i(k) g_i(x)$$
$$x_i(k+1) = \sum_j a_{ij} y_j(k)$$
$$\mu_i(k+1) = \sum_j a_{ij} \mu_j(k) + \alpha g(x_i(k))$$

haven't checked if it works...

Algorithm	Strict Conv.	Centralized
$x_{k+1} = \arg\min_x f(x) + \mu g(x)$ $\mu_{k+1} = [\mu_k + \alpha g_k]^+$	O	O
$x_{k+1} = P_x[x_k - \alpha \partial_x L(x_k, \mu_k)]$ $\mu_{k+1} = [\mu_k + \alpha g_k]^+$	O	
$\hat{x} = \frac{1}{k} \sum x_i$ from 1	X	O
$\hat{x} = \frac{1}{k} \sum x_i, \; \hat{\mu} = \frac{1}{k} \sum \mu_i$	X	O
Augmented Lagrangian	X	O
ADMM $\min f(x) + g(z)$ $\text{s.t.} Ax + Bz = c$	X	X,O
Consensus $x_i(k) = P_{X_i}[\underbrace{\sum_{j=1}^{M} a_{ij} x_j(k)}_{\text{consensus step}} - \underbrace{\alpha \partial_x f_i(x_i(k))}_{\text{subgradient step}}]$	O	X
$x_i(k+1) = \underbrace{\sum_j a_{ij} x_j(k)}_{\text{consensus step}} - \underbrace{\alpha \partial_x f_i(x_i(k))}_{\text{subgradient step}}$ $\mu_i(k+1) = \sum_j a_{ij} \mu_j(k) + \alpha g(x_i(k))$	X	X