Lecture 12, Oct 18, 2023

Inequality-Constrained Optimization

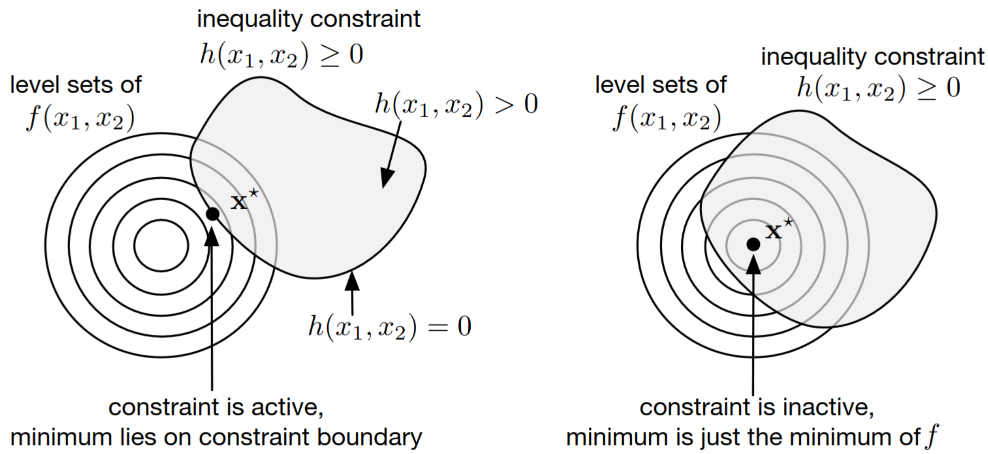

Figure 1: Active vs. inactive constraints.

- We will now consider the full optimization problem of minimizing $f(\boldsymbol{x})$ subject to $\boldsymbol{g}(\boldsymbol{x}) = \mathbf{0}$ and $\boldsymbol{h}(\boldsymbol{x}) \geq \mathbf{0}$
- There are 2 possible cases for each inequality constraint h_i; consider a minimum $\boldsymbol{x}$, then either $h_i(\boldsymbol{x}) = 0$ this point (active), or $h_i(\boldsymbol{x}) > 0$ (inactive)
  - In the first case, the inequality constraint is the same as an equality constraint and the optimum will be on the boundary of the inequality region
  - In the second case, the inequality constraint has effectively no impact on where the optimum is, since it lies fully within the inequality region
  - If there are multiple active inequality constraints, then the solution lies on the intersection of their boundaries
- If we assume that all inequality constraints are active, we can then treat them as equality constraints and use the Lagrange multiplier approach
  - $\Lambda(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\boldsymbol{x}) - \boldsymbol{\lambda}^T \boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{\mu}^T \boldsymbol{h}(\boldsymbol{x})$
  - Then for a critical point $\vec{\nabla} f(\boldsymbol{x}) - \sum_i \lambda_i \vec{\nabla} g_i(\boldsymbol{x}) - \sum_j \mu_j \vec{\nabla} h_j(\boldsymbol{x}) = \mathbf{0}$ and $g_i(\boldsymbol{x}) = h_j(\boldsymbol{x}) = \mathbf{0}, \forall i, j$
- If we let $\mu_j = 0$ whenever the inequality constraint h_j is inactive (i.e. $h_j(\boldsymbol{x}) \neq 0$), then the inactive constraints drop out and the above condition holds for the general case
  - Let $\forall j, \mu_j h_j(\boldsymbol{x}) = \mathbf{0}$, then the condition above holds in general
  - This condition is known as *complementary slackness*

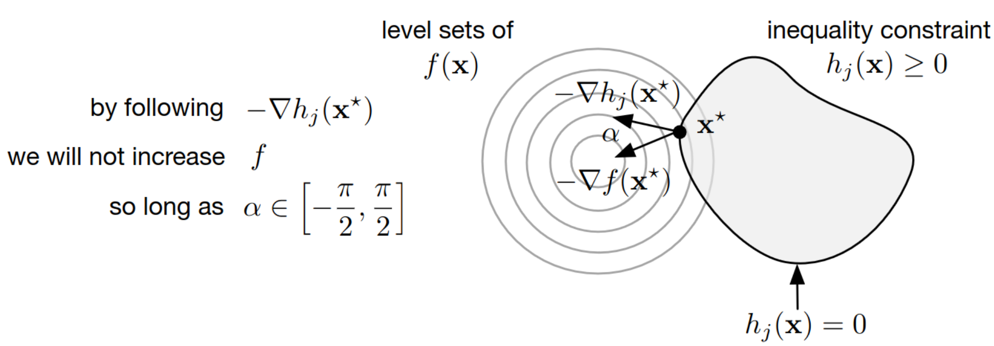

Figure 2: Constraints on Lagrange multipliers for inequality constraints.

- Note that in the equality constraint case, our Lagrange multipliers can be positive or negative; but for inequality constraints, this is not the case
  - Consider the diagram above where $\boldsymbol{x}^*$ is a critical point and $\Delta\boldsymbol{x}$ is some perturbation; let constraint $h_j(\boldsymbol{x})$ be active
  - Note that $\vec{\nabla} h_i(\boldsymbol{x})$ is pointing inwards of the feasible set
  - Intuitively, if $\vec{\nabla} f(\boldsymbol{x})$ and $\vec{\nabla} h_i(\boldsymbol{x})$ have a negative dot product (i.e. they point at least somewhat in the opposite direction), then we can move into our feasible set while decreasing $f(\boldsymbol{x})$
    * But if $\vec{\nabla} f(\boldsymbol{x})$ and $\vec{\nabla} h_i(\boldsymbol{x})$ have a positive dot product, then we cannot both decrease $f(\boldsymbol{x})$ while keeping the inequality constraint satisfied
  - Let $\Delta\boldsymbol{x} = \vec{\nabla} h_i(\boldsymbol{x}^*)^T$, then
    * $h_i(\boldsymbol{x}^* + \Delta\boldsymbol{x}) \approx h_i(\boldsymbol{x}^*) + \vec{\nabla} h_i(\boldsymbol{x}^*)\Delta\boldsymbol{x} = h_i(\boldsymbol{x}^*) + \vec{\nabla} h(\boldsymbol{x}^*)\vec{\nabla} h(\boldsymbol{x}^*)$
    * Since both terms on the right are positive, we know that $h_i(\boldsymbol{x}^* + \Delta\boldsymbol{x}) \geq 0$, i.e. this perturbation is still feasible
    * $f(\boldsymbol{x}^* + \Delta x) \approx f(\boldsymbol{x}^*) + \vec{\nabla} f(\boldsymbol{x}^*)\Delta\boldsymbol{x} = f(\boldsymbol{x}^*) + \vec{\nabla} f(\boldsymbol{x}^*)\vec{\nabla} h_i(\boldsymbol{x}^*)^T$
    * If the second term is negative, then we know that the perturbation decreases f, which means $\boldsymbol{x}^*$ is no longer a critical point – contradiction
  - Therefore we must have that $\vec{\nabla} f(\boldsymbol{x}^*)\vec{\nabla} h_i(\boldsymbol{x}^*)^T \geq 0$ at a local minimum, hence $\vec{\nabla} f(\boldsymbol{x}^*) = \mu_i \vec{\nabla} h_i(\boldsymbol{x}^*)$ where $\mu_i \geq 0$ (strictly greater if h_i must be active)
  - This condition is known as *dual feasibility*
- Note that we can always take an equality constraint and rewrite it in terms of 2 inequality constraints, i.e. $g_i(\boldsymbol{x}) = 0 \iff g_i(\boldsymbol{x}) \geq 0, -g_i(\boldsymbol{x}) \geq 0$
  - This lets us write the critical point condition in terms of only inequality constraints

Theorem

Karush-Kuhn-Tucker (KTT) Conditions: The vector $\boldsymbol{x}^* \in \mathbb{R}^n$ is a critical point for minimizing $f(\boldsymbol{x})$ subject to $\boldsymbol{g}(\boldsymbol{x}) = 0$, $\boldsymbol{h}(\boldsymbol{x}) \geq \boldsymbol{0}$ when there exists $\boldsymbol{\lambda} \in \mathbb{R}^m$ and $\boldsymbol{\mu} \in \mathbb{R}^p$ such that:

1. Stationarity: $\vec{\nabla} f(\boldsymbol{x}^*) - \sum_i \lambda_i \vec{\nabla} g_i(\boldsymbol{x}^*) - \sum_j \mu_j \vec{\nabla} h_j(\boldsymbol{x}^*) = \boldsymbol{0}$
2. Primal feasibility: $\boldsymbol{g}(\boldsymbol{x}^*) = \boldsymbol{0}$ and $\boldsymbol{h}(\boldsymbol{x}^*) \geq \boldsymbol{0}$
3. Complementary slackness: $\forall j, \mu_j h_j(\boldsymbol{x}^*) = 0$
4. Dual feasibility: $\forall j, \mu_j \geq 0$

- To find whether $\boldsymbol{x}^*$ is a local minimum (instead of a maximum or saddle point), we need to check the Hessian constrained to the subspace of $\mathbb{R}^n$ in which $\boldsymbol{x}$ can move without violating the constraints
- Example: optimal rectangle: same rectangle optimization constraint from last time, but we enforce $w \leq \bar{w}$
  - The inequality constraint can be expressed as $\bar{w} - w \geq 0$
  - The Lagrangian becomes $\Lambda(w, l, \lambda, \mu) = -wl - \lambda(2w + 2l - 1) - \mu(\bar{w} - w)$
  - Using the KTT optimality conditions:
    * Stationarity:
      - $\dfrac{\partial \Lambda}{\partial w} = -l^* - 2\lambda^* + \mu^* = 0$
      - $\dfrac{\partial \Lambda}{\partial l} = -w^* - 2\lambda^* = 0$
    * Primal feasibility:
      - $2w^* + 2l^* - 1 = 0$
      - $\bar{w} - \bar{w}^* \geq 0$
    * Complementary slackness: $\mu^*(\bar{w} - \bar{w}^*) = 0$
    * Dual feasibility: $\mu^* \geq 0$
  - We now need to consider 2 cases:
    * $\mu^* = 0$, where the inequality constraint is inactive, so we have $w^* = l^* = \dfrac{1}{4}$ as before
      - We can see that all our equations reduce to the same thing we had previously
    * $\mu^* \neq 0$, where the inequality constraint is active

        - We have $w^* = \bar{w}$ from complementary slackness
        - $\lambda^* = -\frac{\bar{w}}{2}, l^* = \frac{1}{2} - \bar{w}$
        - Substituting into the first stationary equation, $\mu^* = \frac{1}{2} - 2\bar{w}$
        - Since $\mu^* \geq 0$, this only holds if $\bar{w} \leq \frac{1}{4}$ – i.e. the constraint would only be active if $\bar{w} \leq \frac{1}{4}$ (above this value we just get the normal solution)
    - In summary: when $\bar{w} \leq \frac{1}{4}$, the constraint is active and we have $w^* = \bar{w}, l^* = \frac{1}{2} - \bar{w}$; when $\bar{w} \geq \frac{1}{4}$, then the inequality constraint is active and $w^* = l^* = \frac{1}{4}$
    - In general inequality constraints complicate the problem since now we need to consider multiple cases

Numerical Optimization Methods

- We will discuss two broad categories of algorithms:
    - *Sequential quadratic programming (SQP)*: iteratively solve a set of simpler, less constrained problems that approximate the fully constrained problem
    - *Barrier methods*: replace the constraints with penalties in the objective function, and then optimize the new function using unconstrained methods
- SQP approximates $f, \boldsymbol{g}, \boldsymbol{h}$ by simpler functions
    - f is replaced by a quadratic objective while the constraints are replaced by linear constraints
    - This is similar to Newton's method – taking a quadratic approximation, solving for the minimum, and then approximate again
    - The active set method checks which constraints are active
        * Given an active set, all constraints are treated as equality constraints and Lagrange multipliers are used to minimize
    - SQP only converges if we start with an initial guess in the neighbourhood of the true minimum, similar to Newton's method
    - Most model predictive controllers (MPC controllers) use SQP methods

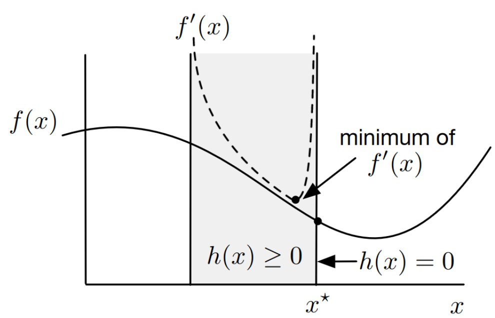

Figure 3: Illustration of barrier methods.

- Barrier methods add penalties to the objective for violating constraints, then optimizes the new objective without constraints
    - e.g. define a new objective as $f'(x) = f(x) + \rho\frac{1}{h(x)}$ with ρ being some weight
    - If constraints are not satisfied closely enough, increase ρ to penalize the cost more
    - ρ is decreased iteratively if we are close enough to the constraints, which brings us closer to the true minimum

Convex Optimization

- We want to develop methods for identifying convex problems

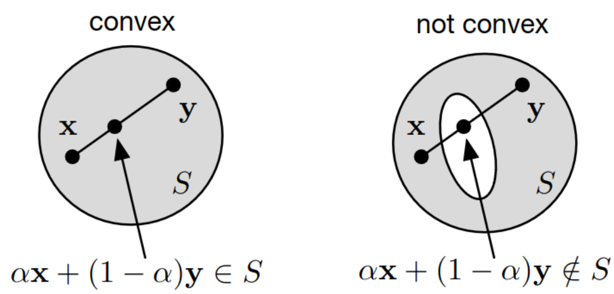

Figure 4: Convex and non-convex sets.

Definition

A set $S \subseteq \mathbb{R}^n$ is *convex* if

$$\forall \boldsymbol{x}, \boldsymbol{y} \in S, \alpha \in [0,1], \alpha\boldsymbol{x} + (1-\alpha)\boldsymbol{y} \in S$$

- Additional useful properties:
  - The intersection of convex sets is also a convex set
    - This means if we combine two convex constraints, we also get a convex constraint back
  - The sum and maximum of convex functions is also a convex function
  - If f is convex, then $\{ \boldsymbol{x} \mid f(\boldsymbol{x}) \leq c \}$ is a convex set for a fixed $c \in \mathbb{R}$
    - We can "chop off" a function below some line to get a convex set

Theorem

If the objective function f and feasible set of an optimization problem are both convex, then the optimization problem possesses a unique minimum.

- An optimization problem is convex if its objective function is convex and its feasibility region is also a convex set
  - When a problem is convex, we can make strong convergence guarantees
  - Intuition: if both $\boldsymbol{x}$ and $\boldsymbol{y}$ are in the feasible set, then any potential minimum that lies between them is also in the feasible set
- Example: consider the constrained least-squares problem of minimizing $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2$ subject to $\boldsymbol{x} \in \boldsymbol{x} \in \mathbb{R}^n | \boldsymbol{x} \geq \boldsymbol{0}$
  - The objective function is convex since it expands to $\boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} - 2b^T\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{b}$, which has a Hessian of $\boldsymbol{A}^T\boldsymbol{A}$, which is positive semi-definite for any $\boldsymbol{A}$
  - We can also show that the feasible set is convex since the positive weighted sum of any two positive numbers is also positive
  - Therefore this problem is convex
- Example: optimizing the 1-norm, $\|\boldsymbol{x}\|_1 = |x_1| + |x_2| + \cdots + |x_n|$
  - We can rewrite it as minimizing $\sum_i y_i$ with respect to $\boldsymbol{x}, \boldsymbol{y}$, subject to $\boldsymbol{x}, \boldsymbol{y} \in \{ \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n \mid y_i \geq x_i, y_i \geq -x_i \}$
  - We're forcing $|x_i| \geq y_i$ and trying to minimize the sum of y_i
  - This is now a convex problem: the objective function is convex since all y_i are convex, so their sum is convex; the feasible set is convex since $y_i - x_i \geq 0$ and $y_i + x_i \geq 0$ are both convex functions, so if we let $h_{i_1}(y, x) = y_i - x_i$, then $h_{i_1} \leq 0$ is also convex; the intersection of both convex constraints gives a convex feasible set
- Note that our optimal rectangle problem is not convex but we still found the global minimum
- Minimizing $\boldsymbol{c}^T\boldsymbol{x}$ subject to $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ is referred to as a *linear program*, which is a special convex problem that can be solved very quickly
- Second-order cone programs are $\boldsymbol{c}^T\boldsymbol{x}$ subject to $\|\boldsymbol{A}_i\boldsymbol{x} - \boldsymbol{b}_i\|_2 \leq d_i + \boldsymbol{c}^T\boldsymbol{x}$ which are also convex
- What about nonconvex problems?

    - We can try to approximate $f(\boldsymbol{x})$ with an easier (convex) problem
    - We can sample the space of feasible $\boldsymbol{x}$ as "seeds" for starting points of a local optimization, optimize all of them, and then picking the minimum
    - Randomized algorithms also exist (e.g. simulated annealing)
- Sometimes we want to do *online optimization*, where the objective function changes over time
    - A simple strategy is to use the old optimum $\boldsymbol{x}^*$ as the initial guess for the new problem