1 Lecture 8

First order necessary condition

For a continuously differentiable function $f : \mathbb{R}^d \to \mathbb{R}$, $\nabla f(x^*) = 0$ holds for the optimal solution x^*

Above is both necessary and sufficient for local and global optimality of x^*, if f is convex.

Second-Order necessary condition

For a twice continuously differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ in an optimization problem, the locally optimal solution x^* will satisfy $\nabla f(x^*) = \mathbf{0}$ and $H_f(x^*) \succeq 0$

When strictness applies for the second condition, we have a locally optimal solution (sufficient).

Lagrangian

Let the Lagrangian be defined as follows:

$$\ell(x, \lambda, \mu) := f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{j=1}^{p} \mu_j h_j(x).$$

where $g_i(x) \leq 0$ and $h_j(x) = 0$

Primal Objective wrt. Lagrangian

$\min_{x \in \mathbb{R}^d} \max_{\lambda \geq 0, \mu} \ell(x, \lambda, \mu)$

$$\max_{\lambda \geq 0, \mu} \ell(x, \lambda, \mu) = \begin{cases} f(x) & \text{if } x \text{ is feasible} \\ +\infty & \text{otherwise} \end{cases}.$$

For the dual problem we have the following: $\max_{\lambda \geq 0, \mu} \min_{x \in \mathbb{R}^d} \ell(x, \lambda, \mu)$

KKT- Necessary conditions

Let f, g_i, h_i all be continuously differentiable function. Then for an optimal solution x^*, following is satisfied

- Primal Feas.: $\forall i, j : g_i(x) \leq 0$ and $h_j(x) = 0$
- Dual Feasibility: $\forall i : \lambda_i \geq 0$
- Compl. Slackness: $\forall i : \lambda_i g_i(x) = 0$
- Stationarity: $\nabla f(x) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x) + \sum_{j=1}^{p} \mu_j \nabla h_j(x) = 0$

Minimax and maximum inequality

It always holds that: $\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} f(x, y) \geq \max_{y \in \mathcal{Y}} \min_{x \in \mathcal{X}} f(x, y)$.

We call this weak duality as well for optimization problems. **Strong duality** if it holds with equality.

2 Lecture 9

Strong duality holds for a convex optimization problem, if there exists a strictly feasible $x \in \mathbb{R}^d : h_j(x) = 0, g_i(x) < 0 \ \forall i, j$. This is called the Slater's condition, the weak slater's condition is with equality as well for the last constraint. If the problem is convex, and all functions affine, then strong duality also holds.

If we have a convex problem, KKT conditions are sufficient and always imply x^* is a globally optimal solution.

Dual function: $g(\lambda, \mu) = \min_{x \in \mathbb{R}^d} \ell(x, \lambda, \mu)$, which is ALWAYS concave. If we take the maximum over λ, μ we have a convex problem.

3 Lecture 10

Line Search and Descent methods

Line search is an interative algorithm with the following update for a computed search direction p_k and stepsize $a_k > 0$ such that: $x^{(k+1)} = x^{(k)} + a_k p_k$. For a descent method following holds: $f(x^{(k+1)}) < f(x^{(k)})$

For a convex continuously-differentiable function, p_k is a descent direction if $p_k^T \nabla f(x^{(k)}) < 0$

Useful characterization of p_k is through a PSD matrix: $p_k = -B_k \nabla f(x^{(k)})$

- Gradient Descent: $B_k = I_d$
- Newton's method: $B_k = H_f^{-1}(x^{(k)})$
- Quasi-Newton's methods: $B_k \approx H_f^{-1}(x^{(k)})$

Two methods to find stepsize a_k:

- Exact line search: $a_k = \underset{a>0}{argmin} f(x^{(k)} + \alpha p_k)$
- Backtracking line search: start from initial $s_k > 0$, repeat $s_k \leftarrow \beta s_k$ until following is achieved $f(\mathbf{x} + s\mathbf{d}) < f(\mathbf{x}) + \alpha s \nabla f(\mathbf{x})^T \mathbf{d}$, for some chosen $\alpha \in]0, 1[$ and $\beta \in]0, 1[$. Then s_k is our chosen step size

Gradient descent for quadratic functions

For quadratic objectives: $f(x) = \frac{1}{2} x^T A x + b^T x + c$ for spd A with ev $0 < \lambda_1 \leq ... \leq \lambda_d$ it holds that $\left\| x^{(k+1)} - x^* \right\| \leq \frac{\lambda_d - \lambda_1}{\lambda_d + \lambda_1} \left\| x^{(k)} - x^* \right\|$ We have linear convergence, namely: $\frac{\left\| x^{(k+1)} - x^* \right\|}{\left\| x^{(k)} - x^* \right\|} \leq M$ for some $M < 1$. Define $\kappa = \frac{\lambda_d}{\lambda_1}$

Gradient descent for convex functions

For convex functions, we have the same, with the Hessian satisfying $\mu I_d \preceq H_f(x) \preceq \lambda I_d$, then we have: $\left\| x^{(k+1)} - x^* \right\| \leq (\frac{\lambda - \mu}{\lambda + \mu}) \left\| x^{(k)} - x^* \right\|$. Define $\kappa = \frac{\lambda}{\mu}$

If $\kappa \approx 1$, then we converge with satisfactory speed, if $\kappa >> 1$, then very slowly.

4 Lecture 11

For the Newton's method, we need the Taylor Approximation:

$$f(x) \approx f(x_0) + \nabla f(x_0)^T (x - x_0) + \frac{1}{2} (x - x_0)^T H_f(x_0)(x - x_0).$$

so we can just optimize over this function, where $x_0 = x^{(k)}$ over each iteration.

Affine invariant of Newton's method

We get the following sequence of updates when trying to minimize f(x)

$$x^{(0)}, x^{(1)}, x^{(2)}, ...$$

then for invertible $A \in \mathbb{R}^{d \times d}$, $f(Ax)$ and initialized at $y^{(0)} = A^{-1} x^{(0)}$ we get the following sequence of updates:

$$A^{-1} x^{(0)}, A^{-1} x^{(1)}, A^{-1} x^{(2)}, ...$$

Quad. convergence of Newton's method

Suppose f is twice differentiable and the Hessian is Lipschitz-continuous with constant L: $\forall x, x' : \left\| H_f(x) - H_f(x') \right\| \leq L \left\| x - x' \right\|$ then for Newton's method that is sufficiently close to the local optimum x^*: $\left\| x^{(k+1)} - x^* \right\| \leq \left\| \frac{L}{\lambda_{min}(H_f(x^*))} \right\| \left\| x^{(k)} - x^* \right\|^2$

Definition of operator norm

Given a matrix $A \in \mathbb{R}^{n \times d}$ and $\ell_p - norm$, the operator norm for matrices is defined as follows:

$$\left\| A \right\|_p = \max_{x \in \mathbb{R}^d : \left\| x \right\|_p \leq 1} \left\| Ax \right\|_p .$$

- $\left\| Ax \right\|_p \leq \left\| A \right\|_p \left\| x \right\|_p$ always holds
- $\left\| BA \right\|_p \leq \left\| B \right\|_p \left\| A \right\|_p$ holds for every matrix
- $\left\| A \right\|_2 = \max_{1 \leq i \leq d} |\lambda_i(A)|$, for symmetric matrix A

5 Lecture 12

Trust region methods

Given an unconstrained optimization problem, trust region methods aim to solve the following:

$$x^{(k+1)} = \underset{x \in S_k}{argmin}\ m_k(x).$$

where S_k is the trust region at iteration k.

1. Standard (quadratic function): $m_k(x^{(k)} + v) = f_k + g_k^T v + \frac{1}{2} v^T B_k v$ and trust region $S_k = \{x \in \mathbb{R}^d : \left\| x - x^{(k)} \right\| \leq \epsilon\}$
2. Affine-based: $m_k(x^{(k)} + v) = f(x^{(k)}) + \nabla f(x^{(k)})^T v$ with the trust region $S_k = \{x^{(k)} + v : \left\| v \right\| \leq \epsilon\}$

3. Quadratic-based: $m_k(x^{(k)}+v) = f(x^{(k)}) + \nabla f(x^{(k)})^T v + \frac{1}{2} v^T H_f(x^{(k)}) v$ with the trust region $S_k = \{x^{(k)}+v : \|v\| \leq \epsilon\}$

(2) implies that $x^{(k+1)} = \underset{v \in \mathbb{R}^d : \|v\| \leq \epsilon}{argmin} \nabla f(x^{(k)})^T v$

(3) implies that $x^{(k+1)} = \underset{v \in \mathbb{R}^d : \|v\| \leq \epsilon}{argmin} \nabla f(x^{(k)})^T v + \frac{1}{2} v^T H_f(x^{(k)}) v$.

Theorem for quadratic trust region subproblems

Given euclidean norm, feasible point v_k^* which is optimal if and only if for some $\lambda_k \geq 0$:

- $H_f(x^{(k)}) + \lambda_k I_d \succeq \mathbf{0}$
- $\lambda_k(\|v_k^*\| - \epsilon) = 0$
- $(H_f(x^{(k)}) + \lambda_k I_d) v_k^* = -\nabla f(x^{(k)})$

Newton's method for indefinite Hessians

- Add a matrix such that $B_k = H_f(x^{(k)}) + E_k$ is sufficiently positive semi-definite: $B_k \succeq \theta I_d$
- Eigenvalue modification: replace all the Hessians ev, less than a threshtold $\tau > 0$ with τ. Alternatively we can get the spectral decomposition and modify the diagonal
- Adding a multiple of the identity: $B_k = H_f(x^{(k)}) + \lambda I_d$, where $\lambda = \max\{0, \theta - \lambda_{min}(H_f(x^{(k)}))\}$

6 Lecture 13

$$\min_{x \in \mathbb{R}^d} f(x)$$
$$\text{subject to } Ax = b$$

Local optimality

For the problem above, every optimal solution will satisfy the following for some vector $\mu \in \mathbb{R}^p$:

$$\nabla f(x^*) + A^T \mu = 0, Ax^* = b.$$

It's necessary in general case, sufficient as well if f is convex.

Equivalent formulation

The equality constraint $Ax = b$ can be turned into $x = Fz + x_0$, where $\{x \in \mathbb{R}^d : Ax = b\} = Fz + x_0 : z \in \mathbb{R}^p$, where x_0 is one feasibly point $Ax_0 = b$ and $F \in \mathbb{R}^{d \times p}$ is a matrix whose range is equal to the null space of $A \in \mathbb{R}^{p \times d}$. We can then optimize over z on $f(Fz + x_0)$ and get rid of the equality constraints.

Another option is using the KKT conditions, and solving following for p_k in the following matrix: $\begin{bmatrix} H_f(x^{(k)}) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} p_k \\ \mu_k \end{bmatrix} = \begin{bmatrix} -\nabla f(x^{(k)}) \\ Ax^{(k)} - b \end{bmatrix}$. This can be derived from the optimality conditions, then our update is $x^{(k+1)} = x^{(k)} - a_k p_k$ for some a_k

$$\min_{x \in \mathbb{R}^d} f(x) - \theta \sum_{i=1}^m log(-g_-(x)) \quad (1)$$
$$g_i(x) \leq 0 \; \forall i \in 1, ..., m \quad (2)$$
$$Ax = b \quad (3)$$

Equivalent formulation using indicator function

$\min_{x \in \mathbb{R}^d} f(x) - \eta \sum_{i=1}^m log(-g_i(x))$
$Ax = b$, gives us an equivalent problem for $\eta > 0$ which is also convex, given the original problem is convex

Let $\phi(x) = -\sum_{i=1}^m log(-g_i(x))$, then we have:
$\nabla\phi(x) = \sum_{i=1}^m -\frac{1}{g_i(x)} \nabla g_i(x)$ and
$H_\phi(x) = \sum_{i=1}^m \left[\frac{1}{g_i^2(x)} \nabla g_i(x) \nabla g_i(x)^T - \frac{1}{g_i(x)} H_{g_i}(x) \right]$

Proposition (Interior Methods)

Every locally optimal solution $x^* \in \mathbb{R}^d$ of the interior point method's optimization problem with coefficient $\eta > 0$ satisfies the following conditions for vector $\lambda \in \mathbb{R}^m$ defined as $\lambda_i = \frac{-\eta}{g_i(x^*)}$ for every $1 \leq i \leq m$ and a vector $\mu \in \mathbb{R}^p$

- P. Feasibility: $g_i(x^*) \leq \forall i$ and $Ax^* = b$
- D. feasibility: $\lambda \geq 0$
- Approximate comp. sla.: $\lambda_i g_i(x^*) = -\eta \; \forall i$
- Stationarity: $\nabla f(\mathbf{x}^*) + \sum_{i=1}^m \lambda_i \nabla g_i(\mathbf{x}^*) + A^T \mu = 0$

Only difference in approximate complementary Slackness

Project Gradient Descent for constr. optimization

Consider an optimization problem with the feasible set S. Then the projection operator $\Pi_S : \mathbb{R}^d$ finds the closest point in set S to some input $x \in \mathbb{R}^d$: $\Pi_s(x) = \underset{y \in S}{argmin} \|y - x\|$.
Then projected gradietn descent is an iter. optimization method applying this projection to the optimization variable after every standard update of gradient descent.

7 EXTRA

Math Rules:

- Deriv. of $f : \mathbb{R}^n \to \mathbb{R}^m$: $df(x) = \begin{bmatrix} \frac{df_1}{dx_1} & \cdots & \frac{df_1}{dx_n} \\ \frac{df_2}{dx_1} & \cdots & \frac{df_2}{dx_n} \\ \cdots & \cdots & \cdots \\ \frac{df_m}{dx_1} & \cdots & \frac{df_m}{dx_n} \end{bmatrix} \in \mathbb{R}^{m \times n}$
- Gradient: we can calculate gradient for $f : \mathbb{R}^n \to \mathbb{R}$. It's defined as $\nabla f(x) = df(x)^T$
- Chain rule: $d(g \circ f)(x) = dg(f(x)) \cdot df(x)$
- Product rule: $g(x)h(x) = dg(x)h(x) + g(x)dh(x)$
- $f(x) = \|Ax - b\|^2, \nabla f(x) = 2A^T(Ax - b), Hess_f(x) = 2A^T A$
- $f(x) = \|Ax - b\|, \nabla f(x) = \frac{A^T(Ax-b)}{\|Ax-b\|}$, $Hess_f(x) = \frac{A^T A}{\|Ax-b\|} - \frac{A^T(Ax-b)((x^T A^T - b^T)A)}{\|Ax-b\|^3}$
- $d\mathbf{0} = 0, d(\alpha \mathbf{X}) = \alpha d\mathbf{X}, d(\mathbf{X^{-1}}) = -\mathbf{X^{-1}}(\mathbf{dX})X^{-1}$, $d\mathbf{X}^T = (d\mathbf{X})^T$, $\frac{dx^T a}{dx} = \frac{da^T x}{dx} = a$, $\frac{dx^T Ax}{dx} = (A + A^T)x$, $\frac{d}{ds}(x - As)^T W(x - As) = -2A^T W(x - As)$, $\frac{d}{dx}(x - As)^T W(x - As) = 2W(x - As)$ **Care:** some above are given as gradients, verify before.

7.1 Convexity

- f is convex if $\forall x, y$ and $\theta \in [0, 1]$: $f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta) f(y)$ holds
- A differentiable function f is convex iff its domain is convex and $\forall x, y \in dom(f) : f(y) \geq f(x) + \nabla f(x)^T (y - x)$
- Same as above, but this $\forall \mathbf{x} \in dom(f) : H_f(\mathbf{x}) \succeq 0$ (so PSD)

Dual Norm: Dual norm is defined as $\|x\|_* = \underset{y : \|y\| \leq 1}{max} y^T x$ Dual norm rules:

- Standard norm rules: positive definiteness, scalar product, triangle inequality
- Dual norm to Euclidean norm itself, by choosing $y = \frac{x}{\|x\|}$
- Dual norm to ℓ_∞ is ℓ_1, by choosing $y = sgn(x)$
- Dual norm to ℓ_1 is ℓ_∞, by choosing $y_i = 1$ where $max(x) = x_i$
- Dual norm to ℓ_p is ℓ_q, with $\frac{1}{p} + \frac{1}{q} = 1$

Pick an initial point $x^{(0)}$
for *k=0 to T* **do**
  Find a descent direction p_k
  Perform line search or backtracking to find step-size $a_k > 0$
  $x^{(k+1)} \leftarrow x^{(k)} + a_k \mathbf{p}_k$
end

Algorithm 1: Line search methods

Pick an initial point $x^{(0)}$
for *k = 0 to T* **do**
  Compute a model function m_k locally designed around $x^{(k)}$
  Assign a trust region S_k around $x^{(k)}$
  Update $x^{(k=1)} = \underset{x \in S_k}{argmin} m_k(x)$
end

Algorithm 2: Trust region methods

Projected Gradient Descent:
Same as normal gradient descent, but in the last step project $x^{(k)}$ onto the feasible set.