CS 450: Numerical Anlaysis

Lecture 17

Chapter 6 Numerical Optimization

Constrained Optimization and Quadratic Programming

Edgar Solomonik

Department of Computer Science
University of Illinois at Urbana-Champaign

March 14, 2018

Constrained Optimization Problems

- We now return to the general case of *constrained* optimization problems:

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad \text{subject to} \quad \underbrace{\boldsymbol{g}(\boldsymbol{x}) = \mathbf{0}}_{\text{equality}} \quad \text{and} \quad \underbrace{\boldsymbol{h}(\boldsymbol{x}) \leq \mathbf{0}}_{\text{inequality}}$$

- Generally, we will seek to reduce constrained optimization problems to a series of unconstrained optimization problems:

sequential quadratic programming (SQP) – solve QP at each step, approximation to constrained nonlinear

duality / active sets – same idea of constraint "removal"

barrier / penalty

Lagrangian Duality

- The Lagrangian function with constraints $\boldsymbol{g}(\boldsymbol{x}) = \mathbf{0}$ and $\boldsymbol{h}(\boldsymbol{x}) \leq \mathbf{0}$ is

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \boldsymbol{\lambda}^T \begin{bmatrix} \boldsymbol{h}(\boldsymbol{x}) \\ \boldsymbol{g}(\boldsymbol{x}) \end{bmatrix} = f(x) + \lambda_1^T h(x) + \lambda_2^T g(x)$$

Lagrangian multipliers

- The Lagrangian dual problem is an unconstrained optimization problem:

$$\max_{\boldsymbol{\lambda}} q(\boldsymbol{\lambda}), \quad q(\boldsymbol{\lambda}) = \begin{cases} \min_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) & \text{if } \boldsymbol{\lambda} \geq \mathbf{0} \\ -\infty & \text{otherwise} \end{cases}$$

seeks activation of constraints $h(x)$ and $g(x)$ that maximally constrains (maximizes) the objective function $f(x)$

Constrained Optimality

- In equality-constrained optimization $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{0}$, minimizers $\boldsymbol{x}^*$ are on the border of the feasible region (set of points satisfying constraints), in which case we must ensure any direction of decrease of f from $\boldsymbol{x}^*$ leads to an infeasible point, which gives us the condition:

$$\exists \boldsymbol{\lambda} \in \mathbb{R}^n, \quad -\nabla f(\boldsymbol{x}^*) = \boldsymbol{J}_{\boldsymbol{g}}^T(\boldsymbol{x}^*)\boldsymbol{\lambda}$$

← cannot further improve

descent direction | g_i and λ_i

- Seek critical points in the Lagrangian function $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \boldsymbol{\lambda}^T \boldsymbol{g}(\boldsymbol{x})$, described by the nonlinear equation,

$$\nabla \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = \begin{bmatrix} \nabla f(\boldsymbol{x}) + \boldsymbol{J}_{\boldsymbol{g}}^T(\boldsymbol{x})\boldsymbol{\lambda} \\ \boldsymbol{g}(\boldsymbol{x}) \end{bmatrix} = \boldsymbol{0}$$

eliminate constraints

Optimize Lagr. over x subject to g(x)
to solve in terms of x, λ

Sequential Quadratic Programming

- *Sequential quadratic programming (SQP)* corresponds to using Newton's method to solve the nonlinear equations,

$$\nabla \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = \begin{bmatrix} \nabla f(\boldsymbol{x}) + \boldsymbol{J}_g^T(\boldsymbol{x})\boldsymbol{\lambda} \\ \boldsymbol{g}(\boldsymbol{x}) \end{bmatrix} = \mathbf{0}$$

$$\hat{x}_{k+1} = \hat{x}_k + \hat{s}_k$$

$s_k \to$ solution to lin. sys

$$H_{\mathcal{L}}(x_k, \lambda_k)\, s_k = \nabla \mathcal{L}(x_k, \lambda_k)$$

$$\begin{bmatrix} x_k \\ \lambda_k \end{bmatrix} = \hat{x}_k$$

$$H_{\mathcal{L}}(x_k, \lambda_k) = \begin{bmatrix} B(x_k, \lambda_k) & J_g^T(x) \\ J_g(x) & 0 \end{bmatrix}$$

$$B(x_k, \lambda_k) = H_f(x_k) + \sum_{i=1}^{m'} \lambda_i H_{g_i}(x_k)$$

Quadratic Programming Problems

objective function is quadratic
constraints are linear

- An equality-constrained quadratic programming problem has the form

$g(x) = Ax - b$

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}), \quad f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^T\boldsymbol{x} \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \qquad g(x) = 0$$

equality constraint

$$\nabla f(x) = Qx + c$$

$$\nabla f(x) = 0 \implies Qx = -c \leftarrow \text{unconstrained optimality}$$

$$-\nabla f(x) = J_g^T(x)\lambda \leftarrow \text{constrained optimality}$$

$$-Qx + c = A\lambda$$

solve Lagrangian function

$$\begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} c \\ b \end{bmatrix}$$

Steepest Descent for Quadratic Programming

- Near the minima, all smooth nonlinear programming problems look like quadratic programming problems, where Q converges to the Hessian at the minima, $H_f(x^*)$:

$$Q_1 \quad \dots \quad Q_k \rightarrow H_f(x^*)$$

SQP

steps $1 \dots k$

constraint $Ax = b$

$$f(x) = x^T Q x + c^T x$$

- Consequently, we can analyze local convergence of methods by considering their convergence for a QP, e.g. for steepest descent: $A = 0$, $c = 0$, $b = 0$,

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k) = x_k - \alpha_k Q x_k$$

$$= (I - \alpha_k Q) x_k, \quad \text{so long as } \alpha_k < \frac{2}{\|Q\|_2}$$

Optimal steepest descent step size

$$\alpha_k = \alpha = \frac{2}{\lambda_{max}(Q) + \lambda_{min}(Q)}$$

$$\|e_{k+1}\| \leq C \|e_k\| \quad \text{where} \quad C = \frac{\kappa(Q) - 1}{\kappa(Q) + 1}$$

if Q is symmetric, real, $\lambda > 0$

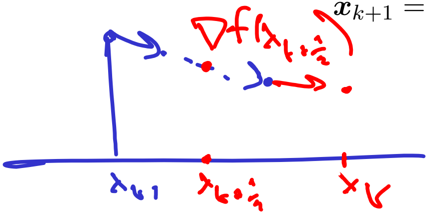
$$C = \frac{\lambda_{max}(Q)/\lambda_{min}(Q) - 1}{\lambda_{max}(Q)/\lambda_{min}(Q) + 1}$$

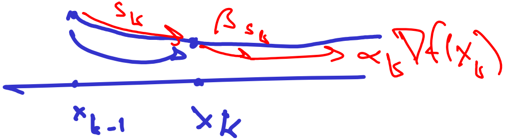
if $\kappa(Q) \gg 1$ (ill-conditioned) then $C \approx 1$ so convergence $\rightarrow$ slow

Gradient Methods with Extrapolation

- We can improve the constant in the linear rate of convergence of steepest descent, by leveraging extrapolation methods, which consider two previous iterates (using *momentum*):

$$\boldsymbol{x}_{k+1} = \underbrace{\boldsymbol{x}_k - \alpha_k \nabla f(\boldsymbol{x}_k)} + \beta_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$$





- The heavy ball method, which uses constant $\alpha_k = \alpha$ and $\beta_k = \beta$, achieves better convergence than steepest descent: constant α_k β_k

$$C = \frac{\sqrt{\kappa(Q)} - 1}{\sqrt{\kappa(Q)} + 1}$$

Conjugate Gradient Method

- The *conjugate gradient method* is capable of making the optimal choice of α_k and β_k at each iteration of an extrapolation method:

CG is best choice of step sizes α_k, β_k
(leading to minimum objective $f(x_{k+1})$)

- Generally conjugate gradient methods perform a sequence of line minimizations in n directions that are Q-orthogonal:

minimize in "different" directions each line

s_i and s_j are Q-orthogonal (convergence in n-steps)

$s_i^T Q s_j = 0$, each step is line search