CS 450: Numerical Anlaysis

Lecture 18

Chapter 6 Numerical Optimization

Conjugate Gradient and Constrained Optimization

Edgar Solomonik

Department of Computer Science
University of Illinois at Urbana-Champaign

March 27, 2018

Sequential Quadratic Programming

- *Sequential quadratic programming (SQP)* solves a constrained quadratic program at the kth step:

  We approximate Lagrangian function $\mathcal{L}_f(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \boldsymbol{\lambda}^T \boldsymbol{g}(\boldsymbol{x})$ by taking its Taylor expansion in $\boldsymbol{x}$ and $\boldsymbol{\lambda}$:

$$\begin{aligned} q(\boldsymbol{x}_k + \boldsymbol{s}, \boldsymbol{\lambda}_k + \boldsymbol{\delta}) =& \mathcal{L}_f(\boldsymbol{x}_k, \boldsymbol{\lambda}_k) + \boldsymbol{s}^T(\nabla f(\boldsymbol{x}_k) + \boldsymbol{J}_{\boldsymbol{g}}^T(\boldsymbol{x}_k)\boldsymbol{\lambda}_k) + \frac{1}{2}\boldsymbol{s}^T \boldsymbol{B}(\boldsymbol{x}_k)\boldsymbol{s} \\ &+ \boldsymbol{\delta}(\boldsymbol{J}_{\boldsymbol{g}}(\boldsymbol{x}_k)\boldsymbol{s} + \boldsymbol{g}(\boldsymbol{x}_k)) \end{aligned}$$

  where we can minimize over $\boldsymbol{s}$ and $\boldsymbol{\delta}$ while ignoring the constant term $\mathcal{L}_f(\boldsymbol{x}_k, \boldsymbol{\lambda}_k)$. This unconstrained quadratic program corresponds to the Lagrangian form of the constrained quadratic program

$$\max_{\boldsymbol{s}} \boldsymbol{s}^T(\nabla f(\boldsymbol{x}_k) + \boldsymbol{J}_{\boldsymbol{g}}^T(\boldsymbol{x}_k)\boldsymbol{\lambda}_k) + \frac{1}{2}\boldsymbol{s}^T \boldsymbol{B}(\boldsymbol{x}_k)\boldsymbol{s}$$

  with constraint $\boldsymbol{J}_{\boldsymbol{g}}(\boldsymbol{x}_k)\boldsymbol{s} = -\boldsymbol{g}(\boldsymbol{x}_k)$.

Solving Quadratic Programs

- Newton's method for optimization can solve the quadratic program with a single step:

  Quadratic approximation is exact for an unconstrained QP. However, Newton's method forces us to form and invert $\boldsymbol{H}_f(\boldsymbol{x}_k)$, which require $O(n^3)$ cost and $O(n^2)$ storage.

- The *conjugate gradient method* provides an effective way of solving QPs iteratively:

  A parallel tangents implementation of the method proceeds as follows

  - *Perform a step of steepest descent to generate $\hat{\boldsymbol{x}}_k$ from $\boldsymbol{x}_k$*
  - *Generate $\boldsymbol{x}_{k+1}$ by minimizing over the line passing through $\boldsymbol{x}_{k-1}$ and $\hat{\boldsymbol{x}}_k$*

  Each conjugate direction is guaranteed to be $\boldsymbol{A}$-orthogonal to previous directions by a recurrence argument similar to that of Lanczos iteration.

Conjugate Gradient as a Krylov Subspace Method

- Generally, Krylov subspaces describe the information available from k matrix-vector products, and can be used to find an approximation $\boldsymbol{x}_k$ to the minima of $\boldsymbol{c}^T\boldsymbol{x} - \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}$:

  Recall the Krylov subspace is defined as $\mathcal{K}_r = span(\boldsymbol{x}_0, \boldsymbol{A}\boldsymbol{x}_0, \ldots, \boldsymbol{A}^{r-1}\boldsymbol{x}_0)$, so our goal is to select a good $\boldsymbol{x}_k \in \mathcal{K}_r$:
  - *there is insufficient information to minimize $||\boldsymbol{x}_k - \boldsymbol{x}^*||_2^2$*
  - *minimum-residual methods (MINRES/GMRES) minimize residual $||\boldsymbol{c}^T - \boldsymbol{A}\boldsymbol{x}_k||_2^2$*
  - *conjugate gradient methods minimize residual in $\boldsymbol{A}^{-1}$ norm $||\boldsymbol{c}^T - \boldsymbol{A}\boldsymbol{x}_k||_{\boldsymbol{A}^{-1}}^2 = (\boldsymbol{c}^T - \boldsymbol{A}\boldsymbol{x}_k)^T\boldsymbol{A}^{-1}(\boldsymbol{c}^T - \boldsymbol{A}\boldsymbol{x}_k)$*
- Conjugate gradient can be derived from vectors generated by the Lanczos algorithm for symmetric (positive-definite) $\boldsymbol{A}$, yielding

$$\boldsymbol{x}_k = \boldsymbol{Q}_k\boldsymbol{T}_k^{-1}\boldsymbol{e}_1||\boldsymbol{c}||_2 \quad \text{if} \quad \boldsymbol{x}_0 = \boldsymbol{c}$$

Minimization follows from the fact that we minimize the projection of the residual onto $\boldsymbol{Q}$, namely $\boldsymbol{Q}^T\boldsymbol{r}$, since

$$\boldsymbol{Q}^T(\boldsymbol{c} - \boldsymbol{A}\boldsymbol{x}) = \boldsymbol{e}_1||\boldsymbol{c}||_2 - \underbrace{\boldsymbol{Q}^T\boldsymbol{A}\boldsymbol{Q}}_{\boldsymbol{T}}\boldsymbol{T}^{-1}\boldsymbol{e}_1||\boldsymbol{c}||_2 = \boldsymbol{0}$$

Conjugate Gradient Properties

- Each iteration of conjugate gradient has cost proportional to a matrix-vector product:
  - *Lanczos generates each vector $\boldsymbol{q}_{k+1}$ by orthogonalizing $\boldsymbol{A}\boldsymbol{q}_k$ to $\boldsymbol{q}_{k-1}$ and $\boldsymbol{q}_k$ (three term recurrence)*
  - *Note that the directions of change $\boldsymbol{q}_{k+1}$ and $\boldsymbol{q}_{k-1}$ are $\boldsymbol{A}$-orthogonal since*

$$\boldsymbol{q}_{k+1}^T \boldsymbol{A}\boldsymbol{q}_{k-1} = \boldsymbol{q}_{k+1}^T(\alpha\boldsymbol{q}_k - \beta\boldsymbol{q}_{k-1} - \gamma\boldsymbol{q}_{k-2}) = 0$$

  - *Conjugate gradient method obtained by transforming recurrence for $\boldsymbol{q}_k$ into recurrence for $\boldsymbol{x}_k$, requiring storage of 4 vectors*

- Conjugate gradient is especially efficient when the matrix has a sparse or implicit representation:

  Requires only ability to perform matrix-vector product and store $O(1)$ vectors.

Active Set Methods

- To use SQP for an inequality constrained optimization problem, consider at each iteration an *active set* of constraints:

  A constraint is active if the current iterate is at a boundary satisfies the inequality constraint as an equality (i.e. current position is at boundary generated by constraint).

- The Karush-Kuhn-Tucker (KKT) optimality conditions given the generalized Lagrangian function $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\nu}) = f(\boldsymbol{x}) + \boldsymbol{\mu}^T \boldsymbol{g}(\boldsymbol{x}) + \boldsymbol{\nu}^T \boldsymbol{h}(\boldsymbol{x})$ are

$$\begin{aligned} \nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) &= \mathbf{0} \\ \boldsymbol{g}(\boldsymbol{x}) &= \mathbf{0} \\ \boldsymbol{h}(\boldsymbol{x}) &\leq \mathbf{0} \\ \boldsymbol{\nu} &\geq \mathbf{0} \\ \boldsymbol{\nu}^T \boldsymbol{h}(\boldsymbol{x}) &= \mathbf{0} \end{aligned}$$

  at an optimal point, we must have that for either the ith inequality constraint is active, so $h_i(\boldsymbol{x}) = \mathbf{0}$ or it is inactive, but its Lagrange multiplier $\nu_i = 0$.

Penalty Functions

- We can reduce constrained optimization problems to unconstrained ones by modifying the objective function. *Penalty* functions are effective for equality constraints $\boldsymbol{g}(\boldsymbol{x}) = 0$:

$$\phi_\rho(\boldsymbol{x}) = f(\boldsymbol{x}) + \frac{1}{2}\rho \boldsymbol{g}(\boldsymbol{x})^T \boldsymbol{g}(\boldsymbol{x})$$

is a simple merit function, and its solutions $\boldsymbol{x}_\rho^*$ satisfy $\lim_{\rho\to\infty} \boldsymbol{x}_\rho^* = \boldsymbol{x}^*$. However, the Hessian of ϕ_ρ becomes increasingly ill-conditioned for large ρ, leading to slow convergence.

- The augmented Lagrangian function provides a more numerically robust approach:

$$\mathcal{L}_\rho(\boldsymbol{x}) = f(\boldsymbol{x}) + \boldsymbol{\lambda}^T \boldsymbol{g}(\boldsymbol{x}) + \frac{1}{2}\rho \boldsymbol{g}(\boldsymbol{x})^T \boldsymbol{g}(\boldsymbol{x})$$

Barrier Functions

- A drawback of penalty function methods is that they can produce infeasible approximate solutions, which is problematic if the objective function is only defined in the feasible region:

  Moreover, the merit and augmented Lagrangian functions only make sense for equality constraints.

- Barrier functions provide an effective way (*interior point methods*) of working with inequality constraints $\boldsymbol{h}(\boldsymbol{x}) \leq \mathbf{0}$:

  Inverse barrier function:

$$\phi_\mu(\boldsymbol{x}) = f(\boldsymbol{x}) - \mu \sum_{i=1}^{m} \frac{1}{h_i(\boldsymbol{x})}$$

  Logarithmic barrier function:

$$\phi_\mu(\boldsymbol{x}) = f(\boldsymbol{x}) - \mu \sum_{i=1}^{m} \log(-h_i(\boldsymbol{x}))$$

  in theory with sufficiently small steps we have $\boldsymbol{x}_\mu^* \to \boldsymbol{x}^*$ *as* $\mu \to 0$