CHICAGO

9: The gradient projection method for nonlinear constrained optimization

CHICAGO

9.1 GRADIENT PROJECTIONS FOR QPS WITH BOUND CONSTRAINTS

Projection

- The problem:

$$\begin{aligned} \min_x \quad & q(x) = \tfrac{1}{2}x^T G x + x^T c \\ \text{subject to} \quad & l \le x \le u, \end{aligned}$$

- Like in the trust-region case, we look for a Cauchy point, based on a projection on the feasible set.
- G does not have to be psd (essential for AugLag)
- The projection operator:

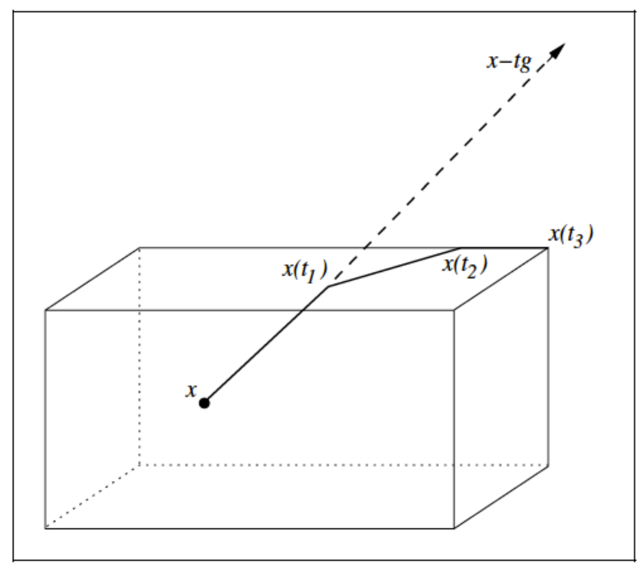
$$P(x, l, u)_i = \begin{cases} l_i & \text{if} \quad x_i < l_i, \\ x_i & \text{if} \quad x_i \in [l_i, u_i], \\ u_i & \text{if} \quad x_i > u_i. \end{cases}$$

The search path

- Create a piecewise linear path which is feasible (as opposed to the linear one in the unconstrained case) by projection of gradient.

$$x(t) = P(x - tg, l, u),$$

$$g = Gx + c;$$



CHICAGO

Computation of breakpoints

- Can be done on each component individually

$$\bar{t}_i = \begin{cases} (x_i - u_i)/g_i & \text{if } g_i < 0 \text{ and } u_i < +\infty, \\ (x_i - l_i)/g_i & \text{if } g_i > 0 \text{ and } l_i > -\infty, \\ \infty & \text{otherwise.} \end{cases}$$

- Then the search path becomes on each component:

$$x_i(t) = \begin{cases} x_i - t g_i & \text{if } t \leq \bar{t}_i, \\ x_i - \bar{t}_i g_i & \text{otherwise.} \end{cases}$$

Line Search along piecewise linear path

- Reorder the breakpoints eliminating duplicates and zero values to get

- The path: $0 < t_1 < t_2 < \ldots$

- $x(t) = x(t_{j-1}) + (\Delta t) p^{j-1}, \qquad \Delta t = t - t_{j-1} \in [0, t_j - t_{j-1}],$

$$p_i^{j-1} = \begin{cases} -g_i & \text{if } t_{j-1} < \bar{t}_i, \\ 0 & \text{otherwise.} \end{cases}$$

CHICAGO

Line Search (2)

- Along each piece, $\left[t_{j-1}, t_j\right]$ find the minimum of the quadratic
$$\frac{1}{2}x^T G x + c^T x$$
- This reduces to analyzing a one dimensional quadratic form of t on an interval.
- If the minimum is on the right end of interval, we continue.
- If not, we found the local minimum and the Cauchy point.

Subspace Minimization

- Active set of Cauchy Point

$$\mathcal{A}(x^c) = \{i \mid x_i^c = l_i \text{ or } x_i^c = u_i\}.$$

- Solve subspace minimization problem

$$\begin{aligned} \min_x \; & q(x) = \tfrac{1}{2} x^T G x + x^T c \\ \text{subject to} \quad & x_i = x_i^c, \; i \in \mathcal{A}(x^c), \\ & l_i \le x_i \le u_i, \; i \notin \mathcal{A}(x^c). \end{aligned}$$

- No need to solve exactly. For example truncated CG with termination if one inactive variable reaches bound.

Gradient Projection for QP

Algorithm 16.5 (Gradient Projection Method for QP).

Compute a feasible starting point x^0;

for $k = 0, 1, 2, \ldots$

  if x^k satisfies the KKT conditions for (16.68)

    stop with solution $x^* = x^k$;

  Set $x = x^k$ and find the Cauchy point x^c;

  Find an approximate solution x^+ of (16.74) such that $q(x^+) \leq q(x^c)$

    and x^+ is feasible;

  $x^{k+1} \leftarrow x^+$;

end (for)

Or, equivalently, if projection does not advance from 0.

Observations – Gradient Projection

- Note that the Projection – Active set solve loop must be iterated to optimality.
- What is the proper stopping criteria? How do we verify the KKT?
- Idea: When projection does not progress ! That is, on each component, either the gradient is 0, or the breakpoint is 0.

CHICAGO

KKT conditions for Quadratic Programming with BC

9.2 AUGMENTED LAGRANGIAN

AUGLAG: Equality Constraints

- The augmented Lagrangian:

$$\mathcal{L}_A(x, \lambda; \mu) \stackrel{\text{def}}{=} f(x) - \sum_{i \in \mathcal{E}} \lambda_i c_i(x) + \frac{\mu}{2} \sum_{i \in \mathcal{E}} c_i^2(x),$$

- Observation: if

$$\lambda = \lambda^*; \mu \geq \mu_0 \Rightarrow \nabla_x \mathcal{L}_A\left(x^*, \lambda^*, \mu\right) = 0;$$

$$\nabla_{xx}^2 \mathcal{L}_A\left(x^*, \lambda^*, \mu\right) = \nabla_{xx}^2 \mathcal{L}\left(x^*, \lambda^*, \mu\right) + \mu\left(\nabla c\left(x^*\right)\right)^T \left(\nabla c\left(x^*\right)\right)$$

AUGLAG: SOC

- So x* is a stationary point for Auglag for exact multipliers … but is it a minimum?
- Yes, for mu sufficiently large.

$$\nabla_{xx}^2 \mathcal{L}_{\mathcal{A}}\left(x^*,\lambda^*,\mu\right) \sim \begin{bmatrix} Y & Z \end{bmatrix}^T \nabla_{xx}^2 \mathcal{L}_{\mathcal{A}}\left(x^*,\lambda^*,\mu\right) \begin{bmatrix} Y & Z \end{bmatrix} + \mu\left(\nabla c\left(x^*\right)Y\right)^T \left(\nabla c\left(x^*\right)Y\right) =$$

$$\begin{bmatrix} Z^T \nabla_{xx}^2 \mathcal{L}_{\mathcal{A}}\left(x^*,\lambda^*,\mu\right) Z & * \\ * & * + \mu\left(\nabla c\left(x^*\right)Y\right)^T \left(\nabla c\left(x^*\right)Y\right) \end{bmatrix} \succ 0 \quad \text{for } \mu \text{ suff large.}$$

- So it is *almost* as solving unconstrained problem … but how do I find multiplier estimates?

Multiplier Estimates Auglag

- At the current estimate, solve problem

$$0 \approx \nabla_x \mathcal{L}_A(x_k, \lambda^k; \mu_k) = \nabla f(x_k) - \sum_{i \in \mathcal{E}} [\lambda_i^k - \mu_k c_i(x_k)] \nabla c_i(x_k).$$

- The obvious choice:

$$\lambda_i^{k+1} = \lambda_i^k - \mu_k c_i(x_k), \quad \text{for all } i \in \mathcal{E}.$$

- What do I do if I converge lambda but x* is not feasible? Increase the penalty mu (it will have to end increasing eventually).

  –

The general case

- The bound constrained formulation. Slacks.

$$c_i(x) \geq 0, i \in \mathcal{I}, \quad \Longrightarrow \quad c_i(x) - s_i = 0, \quad s_i \geq 0, \quad \text{for all } i \in \mathcal{I}.$$

- The problem:

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{subject to} \quad c_i(x) = 0, \; i = 1, 2, \ldots, m, \quad l \leq x \leq u.$$

The augmented Lagrangian

- The new AugLag

$$\mathcal{L}_A(x, \lambda; \mu) = f(x) - \sum_{i=1}^{m} \lambda_i c_i(x) + \frac{\mu}{2} \sum_{i=1}^{m} c_i^2(x).$$

- The bound constrained optimization problem:

$$\min_x \mathcal{L}_A(x, \lambda; \mu) \quad \text{subject to } l \le x \le u.$$

- Same property: if Lagrange multiplier is the optimal one for eq cons and mu is large enough then x* is a solution !

  –

CHICAGO

Practical AugLag alg: LANCELOT

Main computation: Use bound constrained projection.

Forcing sequences

Algorithm 17.4 (Bound-Constrained Lagrangian Method).

Choose an initial point x_0 and initial multipliers λ^0;
Choose convergence tolerances η_* and ω_*;
Set $\mu_0 = 10$, $\omega_0 = 1/\mu_0$, and $\eta_0 = 1/\mu_0^{0.1}$;
for $k = 0, 1, 2, \ldots$
  Find an approximate solution x_k of the subproblem (17.50) such that

$$\left\| x_k - P\left(x_k - \nabla_x \mathcal{L}_A(x_k, \lambda^k; \mu_k), l, u\right)\right\| \le \omega_k;$$

  if $\|c(x_k)\| \le \eta_k$
    (* test for convergence *)
    if $\|c(x_k)\| \le \eta_*$ **and** $\left\| x_k - P\left(x_k - \nabla_x \mathcal{L}_A(x_k, \lambda^k; \mu_k), l, u\right)\right\| \le \omega_*$
      stop with approximate solution x_k;
    end (if)
    (* update multipliers, tighten tolerances *)
    $\lambda^{k+1} = \lambda^k - \mu_k c(x_k)$;
    $\mu_{k+1} = \mu_k$;
    $\eta_{k+1} = \eta_k / \mu_{k+1}^{0.9}$;
    $\omega_{k+1} = \omega_k / \mu_{k+1}$;
  else
    (* increase penalty parameter, tighten tolerances *)
    $\lambda^{k+1} = \lambda^k$;
    $\mu_{k+1} = 100\mu_k$;
    $\eta_{k+1} = 1/\mu_{k+1}^{0.1}$;
    $\omega_{k+1} = 1/\mu_{k+1}$;
  end (if)
end (for)

Solving the bound constrained subproblem

- It is an iterative bound constrained optimization algorithm with trust-region:

$$\min_{d} \tfrac{1}{2} d^T \left[\nabla^2_{xx}\mathcal{L}(x_k, \lambda^k) + \mu_k A_k^T A_k\right] d + \nabla_x \mathcal{L}_A(x_k, \lambda^k; \mu_k)^T d$$
$$\text{subject to } \; l \le x_k + d \le u, \qquad \|d\|_\infty \le \Delta,$$

- Each step solves a bound constrained QP (not necessarily PD), same as in your homework 4.
- The difference: after a subspace solve: compute the new derivative and update TR.