14.1

Algorithms for NLP with nonlinear constraints

Katta G. Murty, IOE 611 Lecture slides

Penalty Function Methods

Consider : min $\theta(x)$ s. to $h_i(x) = 0$ $i = 1$ to m; $g_j(x) \geq 0$ $j = 1$ to ℓ.

Penalty function for equality constraints: $p(h(x)) = \Sigma_{i=1}^{m} |h_i(x)|^s$ typically, where $s = 1$ or 2. If $s = 2$, this function is continuously differentiable.

Penalty function for inequality constraints: $q(g(x)) = \Sigma_{j=1}^{\ell} [\max\{0, -g_j(x)\}]^r$, where $r = 1$ to 4. It is cont. diff. if $r = 2$, and twice cont. diff. if $r = 3$.

The overall penalty function for our NLP is: $\alpha(x) = p(h(x)) + q(g(x))$.

The **Exterior Penalty Method** solves the NLP by finding the unconstrained min of **auxiliary function** $f_\mu(x) = \theta(x) + \mu\alpha(x)$ where $\mu > 0$ is the **Penalty parameter**.

Example: min x_1 s. to $x_1 - 2 \geq 0$

Example: min $x_1^2 + x_2^2$ s. to $x_1 + x_2 - 1 = 0$.

The Penalty Problem: min $f_\mu(x)$ s. to $x \in X$. Here X may be R^n or the set of feasible sols. of other constraints not included in penalty function $\alpha(x)$ (usually these may be simple constraints like bounds on vars.).

Theorem: $X \neq \emptyset$, and suppose $x(\mu)$, the minimizer of the penalty problem exists $\forall \mu > 0$, and $\psi(\mu) = f_\mu(x(\mu))$. Then:

(i) Min obj value in original NLP $\geq \sup_{\mu>0} \psi(\mu)$.

(ii) $\theta(x(\mu)) \uparrow$, $\psi(\mu) \uparrow$, $\alpha(x(\mu)) \downarrow$ over $\mu > 0$.

Theorem: Under hypothesis of above theorem, and also if $\{x(\mu) : \mu > 0\}$ is contained in a compact subset of X, we have:

(i) min obj. value in original NLP $= \sup_{\mu>0} \psi(\mu) = \lim_{\mu\to\infty} \psi(\mu)$.

(ii) $\mu\alpha(x(\mu)) \to 0$ as $\mu \to \infty$, and the limit of any convergent subsequence of $\{x(\mu)\}$ is opt. to original NLP.

SUMT: Take an increasing sequence $\{\mu_1, \mu_2, \ldots\}$ diverging to ∞. Find $x(\mu_{t+1})$ using $x(\mu_t)$ as initial pt. by some unconstrained min algo. All $x(\mu)$ may be infeasible to original NLP, but as $\mu \to \infty$, $x(\mu) \to$ opt. of original NLP assuming it exists, hence method called **exterior penalty method**.

Lagrange Multiplier Estimates: Suppose $X = R^n$. Since $x(\mu)$ is unconstrained min of $f_\mu(x)$, we have $\nabla_x f_\mu(x(\mu)) = 0$ $\forall \mu$. From the coefficients in this eq. when μ is large, we can estimate the opt. Lagrange multiplier vector for original NLP.

Computational Difficulties: When μ very large computational difficulties are caused by ill-conditioning.

Also, when there are nonlinear equalities, movement along a direction d from a feasible pt. $\bar{x}$ leads to infeasible pts., so when μ is large, even if $\theta(x)$ decreases $f_\mu(\bar{x} + \lambda d)$ may be larger than $f_\mu(\bar{x})$ except when λ is very small. So, step lengths tend to be small resulting in slow convergence & premature termination.

Also, Hessian of auxiliary function tends to be highly ill-conditioned when μ is large.

For 2nd example above, Hessian of auxiliary func. is

$$\begin{pmatrix} 2(1+\mu) & 2\mu \\ 2\mu & 2(1+\mu) \end{pmatrix}.$$

It has eigenvalues 2 and $2(1+\mu)$, so its cond. no. tends to ∞.

That's why sequential implementations are used.

Exact Penalty Functions

Exact penalty functions are penalty functions for which min of auxiliary func. is also opt. for original problem for finite positive values of penalty parameter. The *absolute value (L_1)* is an exact penalty function for : min $\theta(x)$ s. to

$$g_i(x) \begin{cases} = 0 & i = 1 \text{ to } m \\ \geq 0 & i = 1 \text{ to } m + \ell \end{cases}$$

The L_1 auxiliary function is:

$$f(x) = \theta(x) + \mu[\sum_{i=1}^{m} |g_i(x)| + \sum_{i=m+1}^{m+\ell} \max\{0, -g_i(x)\}]$$

Theorem: Suppose $\theta(x)$ convex, $g_i(x)$ affine for $i = 1$ to m, and concave for $i = m + 1$ to $m + \ell$. Let $(\bar{x}, \bar{\pi})$ be a KKT pair for original NLP. Then for $\mu \geq \max\{|\bar{\pi}_i|\}$, $\bar{x}$ also minimizes the auxiliary func. with the L_1 penalty function.

Augmented Lagrangian Penalty Function (ALAG), or Multiplier Penalty Function.

For siplicity consider eq. constraints only first. min $\theta(x)$ s. to $h_i(x) = 0$, $i = 1$ to m.

Let $\pi = (\pi_1, \ldots, \pi_m)$ be the Lagrange multiplier vector. The ALAG leads to the auxiliary function:

$$F_\mu(x, \pi) = \theta(x) + \sum_{i=1}^{m} \pi_i h_i(x) + \mu \sum_{i=1}^{m} (h_i(x))^2$$

The ALAG is another exact penalty func. If $(\bar{x}, \bar{\pi})$ satisfy 1st order opt. conds. for original NLP, then $\nabla_x F_\mu(\bar{x}, \bar{\pi}) = 0 \; \forall \mu > 0$.

Theorem: If $(\bar{x}, \bar{\pi})$ satisfies 2nd order suff. opt. conds., there exists a $\bar{\mu} > 0$ s. th. $\forall \mu \geq \bar{\mu}$, $F_\mu(x, \bar{\pi})$ has a strict local min at $\bar{x}$.

Method of Multipliers:

1 Initialization: Select (x^0, π^0) the initial pair, and $\mu = (\mu_1, \ldots, \mu_m)$ the penalty parameter values for different constraints. $w(x^0)$ $= \max \{|h_i(x^0)|\colon\ i = 1 \text{ to } m\}$ is initial infeasibility measure.

2 Penalty func. min.: Let (x^r, π^r, μ^r) be current vectors. Find min of $F_{\mu^r}(x, \pi^r) = \theta(x) + \pi^r h(x) + \Sigma_{i=1}^m \mu_i^r (h_i(x))^2$. Suppose it is x^{r+1}.

If $w(x^{r+1}) \leq \epsilon_1$, stop with x^{r+1} as (x^{r+1}, π^r) satisfy 1st order opt. conds.

If $\epsilon_1 < w(x^{r+1}) \leq \frac{1}{4} w(x^r)$, define $(\pi_i^{r+1}) = (\pi_i^r + 2\mu_i h_i(x^{r+1}))$, $\mu^{r+1} = \mu^r$. With $(x^{r+1}, \pi^{r+1}, \mu^{r+1})$ repeat this step.

If $w(x^{r+1}) > \frac{1}{4} w(x^r)$, for each i for which $|h_i(x^{r+1})| > \frac{1}{4} w(x^r)$, define $\mu_i^{r+1} = 10\mu_i^r$, and $\mu_i^{r+1} = \mu_i^r$ for all other i. Define $\pi^{r+1} = \pi^r$. With $(x^{r+1}, \pi^{r+1}, \mu^{r+1})$ repeat this step.

To handle inequalities: Write the constraint $g_i(x) \geq 0$ as $g_i(x) - s_i^2 = 0$ and apply the above method.

Barrier function methods

For inequality constraints only. If equality constraints exist you can include them in objective func. using penalty func. for equality constraints. So consider: min $\theta(x)$ s. to $g_j(x) \geq 0$, $j = 1$ to ℓ.

Barrier function for these inequalities is continuous in $\{x : g(x) > 0\}$, and it $\to \infty$ as point tends to the boundary from interior.

Frisch's Log Barrier Function: $B(x) = \Sigma_{j=1}^{\ell} log_e(g_j(x))$.

Other barrier functions used are: $\Sigma_{j=1}^{\ell} \frac{1}{g_j(x)}$, $\Sigma_{j=1}^{\ell} \log[\min\{1, g_j(x)\}]$.

The auxiliary func. is: $f_\mu(x) = \theta(x) + \mu B(x)$. Let $\psi(\mu) = \min f_\mu(x)$ s. to $g(x) > 0$, and let $x(\mu)$ be the minimizing point.

Theorem: Inf $\{\theta(x) : g(x) \geq 0\} \leq \inf \{\psi(\mu) : \mu > 0\}$. Also, for $\mu > 0$, $\theta(x(\mu)) \uparrow \psi(\mu) \uparrow$ and $B(x(\mu)) \downarrow$.

Theorem: Suppose original problem has opt. at $\bar{x}$. Then

$$\min\{\theta(x) : g(x) \geq 0\} = \lim_{\mu \to 0^+} \psi(\mu) = \inf\{\psi(\mu) : \mu > 0\}$$

And the limit of any convergent subsequence of $\{x(\mu)\}$ is opt. sol. of original NLP, and $\mu B(x(\mu)) \to 0$ as $\mu \to 0$.

The Barrier Algo.

Initialization: Start with x^0 satisfying $g(x^0) > 0$, $\mu^0 > 0$ and $\beta \in (0, 1)$.

General step: Given x^r, μ_r use an unconstrained min algo to solve to $\quad \min\ \theta(x) + \mu_r B(x) \quad$ s. to $g(x) > 0$.

The constraints $g(x) > 0$ can be ignored as $B(x) \to \infty$ as x tends to satisfy $g_j(x) = 0$ for any j.

Let x^{r+1} be opt sol. If $\mu_r B(x^{r+1}) < \epsilon$, terminate with x^{r+1}. Otherwise let $\mu_{r+1} = \beta\mu_r$. With (x^{r+1}, μ_{r+1}) go to next step.

Recursive QP, or Merit Function Sequential QP (MSQP) Algorithms

Consider: min $\theta(x)$ s. to $g_i(x) \begin{cases} = 0 & i = 1 \text{ to } m \\ \geq 0 & i = m+1 \text{ to } m+p \end{cases}$

All functions assumed twice cont. differentiable.

Lagrangian $L(x, \pi) = \theta(x) - \Sigma_{i=1}^{m+p} \pi_i g_i(x)$.

A **merit function** $S(x)$, an absolute value penalty function (the L_1 penalty function) balancing the competing goals of decreasing $\theta(x)$, and reducing constraint violation, is used.

$$S(x) = \theta(x) + \sum_{i=1}^{m} \hat{\mu}_i |g_i(x)| + \sum_{i=m+1}^{m+p} \hat{\mu}_i |\min\{0, g_i(x)\}|$$

where the $\hat{\mu}_i$ are positive penalty parameters satisfying certain lower bound restrictions discussed later.

A QP employing a 2nd order approx to the Lagrangian minimized over a linear approx. to constraints, is solved in each step. Output of QP provides a descent direction for the merit func., and a line search is carried out in this direction.

Let $\bar{x}$ be current pt., & $d = x - \bar{x}$, and $\bar{\pi}$ satisfy $\bar{\pi} \geq 0 \forall i \in \{m+1, \ldots, m+p\}$. 2nd order Taylor approx. to Lagrangian obtained using a PD symmetric approx. to Hessian updated by BFGS QN update formula. It is $L(\bar{x}, \bar{\pi}) + \nabla_x L(\bar{x}, \bar{\pi})d + \frac{1}{2}d^T Bd$ where B is the current approx to the Hessian of the Lagrangian. Using the constraints, it can be seen that minimizing this s. to linearized constraints leads to QP:

$$\min \quad \nabla\theta(\bar{x})d + \frac{1}{2}d^T Bd$$

$$\text{s. to} \quad g_i(\bar{x}) + \nabla g_i(\bar{x})d \begin{cases} = 0 & i = 1 \text{ to } m \\ \geq 0 & i = m+1 \text{ to } m+p \end{cases}$$

Let $(\tilde{d}, \tilde{\pi})$ be the opt. pair for this QP.

If $\tilde{d} = 0$, $(\bar{x}, \tilde{\pi})$ is a KKT pair for original NLP. Terminate.

If $\tilde{d} \neq 0$, it is a descent direction for merit func.

$$S(x) = \theta(x) + \sum_{i=1}^{m} \tilde{\mu}_i |g_i(x)| + \sum_{i=m+1}^{m+p} \tilde{\mu}_i |\min\{0, g_i(x)\}|$$

where weights $\tilde{\mu}_i$ satisfy $\tilde{\mu}_i > |\tilde{\pi}_i| \forall i$. These weights are usually choosen from:

$$\tilde{\mu}_i = \max\{|\tilde{\pi}_i|, \frac{1}{2}(\bar{\mu}_i + |\tilde{\pi}_i|)\} \quad \forall i$$

where $\bar{\mu}_i$ are weights used in previous step.

Do a line search for min $S(\bar{x} + \lambda\tilde{d})$: $\lambda \geq 0$. If $\bar{\lambda}$ is opt. step length, next pair is: $(\tilde{x} = \bar{x} + \bar{\lambda}\tilde{d}, \tilde{\pi})$. If it satisfies KKT conds. for original NLP reasonably closely, terminate with it. Otherwise go to next step with it.

Theorem: $\tilde{d}$ is a descent direction at $\bar{x}$ for the merit function $S(x)$.

A Difficulty: Even if original NLP feasible, the QP may be infeasible. For this replace QP by:

$$\min \quad \nabla\theta(\bar{x})d + \frac{1}{2}d^T Bd + \rho(\sum u_i + \sum v_i)$$

$$\text{s. to} \quad g_i(\bar{x}) + \nabla g_i(\bar{x})d + u_i - v_i = 0, \quad i \in \{1, \ldots, m\}$$

$$g_i(\bar{x}) + \nabla g_i(\bar{x})d + u_i \geq 0, \quad i \in \{m+1, \ldots, m+p\}$$

$$u_i, v_i \geq 0, \forall i$$

where ρ is a positive penalty parameter. This QP model always feasible since $d = 0$ is feasible for it. If $\tilde{d} \neq 0$ is opt for it, it will also be a descent direction for $S(x)$.

Example: $\min\ \theta(x) = x_1^3 + x_2^2$, s. to $x_1^2 + x_2^2 - 10 = 0$, $x_1 - 1 \geq 0$, $x_2 - 1 \geq 0$.

Theorem: Assume initial pt. x^0 sufficiently close to a KKT pt. $\bar{x}$ for NLP, and the pair $(\bar{x}, \bar{\pi})$ satisfies: $\{\nabla g_i(\bar{x}) : i$ s. th. $g_i(\bar{x}) = 0\}$ is l.i., and $\bar{\pi}_i > 0\ \forall i$ s. th. $i \in \{m+1, \ldots, m+p\} \cap \{i : g_i(\bar{x}) = 0\}$; and $y^T \nabla^2_{xx} L(\bar{x}, \bar{\pi}) y > 0$ for all $y \neq 0$ in $\{y : \nabla g_i(\bar{x}) y = 0, \forall i$ s. th. $g_i(\bar{x}) = 0\}$. Then the sequence of pairs generated by algo. converges to $(\bar{x}, \bar{\pi})$ superlinearly.

Successive (or Recursive, or Sequential) LP Approaches: Penalty SLP (PSLP)

Consider: min $\theta(x)$

$$\text{s. to} \quad g_i(x) \begin{cases} = 0, & i = 1 \text{ to } m \\ \geq 0, & i = m+1 \text{ to } m+p \end{cases}$$

$$x \in X = \{x : Ax \leq b\}$$

The L_1 exact penalty function for this problem is: $F_\mu(x) = \theta(x) + \mu[\Sigma_{i=1}^m |g_i(x)| + \Sigma_{i=m+1}^{m+p} \max\{0, g_i(x)\}]$.

The Penalty Problem is: min $F_\mu(x)$, s. to $x \in X$. This has a nonlinear obj. func., but linear constraints.

Given any $x \in X$, define $(y_i), (z_i^+, z_i^-)$ associated with it by

$$z_i^+ = \max\{0, g_i(x)\}, \quad z_i^- = \max\{0, -g_i(x)\}, \forall i \in \{1, \ldots, m\}$$

$$y_i = \max\{0, g_i(x)\} \quad \forall i \in \{m+1, \ldots, m+p\}$$

So, for $i = 1$ to m, $z_i^+ + z_i^- = |g_i(x)|$.

PSLP attempts to solve the penalty problem using 1st order approx. & a trust region strategy. The 1st order approx. of $F_\mu(x)$ around current pt. $\bar{x}$, denoted by $FL_\mu(d)$ where $d = x - \bar{x}$ is:

$$FL_\mu(d) = \theta(\bar{x}) + \nabla\theta(\bar{x})d + \mu[\sum_{i=1}^{m} |g_i(\bar{x}) + \nabla g_i(\bar{x})d|$$

$$+ \sum_{i=m+1}^{m+p} \max\{0, g_i(\bar{x}) + \nabla g_i(\bar{x})d\}]$$

PLSP attemps to find d to min $FL_\mu(d)$ s. to $A(\bar{x} + d) \leq b$ and $-\alpha \leq d_j \leq \alpha$ $\forall j = 1$ to n, for some selected positive trust region tolerance α. This leads to following LP:

$$\min \nabla\theta(\bar{x})d + \mu[\sum_{i=1}^{m}(z_i^+ + z_i^-) + \sum_{i=m+1}^{m+p} y_i]$$

$$\text{s. to} \quad y_i \geq g_i(\bar{x}) + \nabla g_i(\bar{x})d, \quad i \in \{m+1, \ldots, m+p\}$$

$$z_i^+ - z_i^- = g_i(\bar{x}) + \nabla g_i(\bar{x})d, \quad i \in \{1, \ldots, m\}$$

$$A(\bar{x} + d) \leq b$$

$$-\alpha \leq d_j \leq \alpha, \quad j \in \{1, \ldots, n\}$$

$$y_i, z_i^+, z_i^- \geq 0, \quad \forall i$$

If $\bar{x} \in X$, then 0 is a feasible sol. to this LP. If $\bar{d}$ is an opt. sol. of this LP, define:

Actual change in exact penalty func. $= F_\mu(\bar{x}) - F_\mu(\bar{x} + \bar{d})$

Predicted Change by the linearized version $= FL_\mu(\bar{x}) - FL_\mu(\bar{x} + \bar{d})$

Theorem: $d = 0$ is an opt sol. for above LP iff $\bar{x}$ is a KKT sol. for penalty problem.

Also, since $d = 0$ is feasible to LP, the predicted change by linearized version is ≤ 0, and is 0 iff $\bar{d} = 0$ is opt. to LP.

The model PSLP Algorithm

Start with an $\bar{x} \in X$ as current point. Select trust region tolerance α, penalty parameter μ, and scalars $0 < \rho_0 < \rho_1 < \rho_2 < 1$ and tolerance adjustment factor $\beta \in (0, 1)$. Typically, $\rho_0 = 10^{-6}$, $\rho_1 = 0.25$, $\rho_2 = 0.75$, $\beta = 0.5$.

Solve LP corresponding to point $\bar{x}$. If $\bar{d} = 0$ is opt to LP, $\bar{x}$ satisfies necessary opt. conds. for penalty problem. In this case if $\bar{x}$ is close enough to being feasible to original NLP, it is a KKT point for it, terminate. If $\bar{x}$ is infeasible to original NLP, increase penalty parameter μ and repeat.

If opt. to LP, $\bar{d} \neq 0$, compute the actual and predicted changes. By theorem, predicted change > 0, compute $R = \frac{\text{Actual Change}}{\text{Predicted change}}$.

If $R < \rho_0$, penalty function either worsened or improvement in it is insufficient; keep $\bar{x}$ as current sol., shrink α to $\beta\alpha$ and go to next step. After several such reductions if needed, a new pt. will be choosen.

If $R > \rho_0$, accept $\bar{x}+\bar{d}$ as the new current sol. If $R < \rho_1$ shrink α to $\beta\alpha$ as the penalty function has not improved sufficiently. If $\rho_1 \leq R \leq \rho_2$ retain α at its present value. If $R > \rho_2$ amplify trust region by setting α to α/β. Go to next step.

The Generalized Reduced Gradient (GRG) Method

Write the constraints as eqs. by introducing squared slack variables for inequality constraints, if any.

Consider problem in form: min $\theta(x)$ s. to $h(x) = 0$, $\ell \le x \le u$ where $h(x) = (h_1(x), \ldots, h_m(x))^T$.

Start with a feasible sol. If none available, let x^0 be a good pt. Modify problem to: min $\theta(x)+\alpha x_{n+1}$, s. to $h(x)-x_{n+1}h(x^0) = 0$, $\ell \le x \le u$, $0 \le x_{n+1} \le 1$. where x_{n+1} is an artificial variable with a large positive cost coeff of α in obj. func.

Clearly, for modified system $(x^0, x_{n+1} = 1)^T$ is a feasible sol. And modified system in same form as the original.

We continue to discuss the original problem. Let $\bar{x}$ be current feasible sol.

Assume $\nabla h(\bar{x})$ of order $m \times n$ has rank m. Partition the variables into (x_B, x_D) where x_B is a vector of m basic variables satifying: $\nabla_{x_B}(h(\bar{x}))$ of order $m \times m$ is nonsingular; and x_D is the vector of remaining $n - m$ nonbasic variables.

The **reduced gradient at** $\bar{x}$ in the space of nonbasic variables x_D is:

$$\bar{c}_D = (\bar{c}_j) = \frac{\partial\theta(\bar{x})}{\partial x_D} - \frac{\partial\theta(\bar{x})}{\partial x_B}(\frac{\partial h(\bar{x})}{\partial x_B})^{-1}\frac{\partial h(\bar{x})}{\partial x_D}$$

In the space of nonbasic variables x_D define the search direction $y_D = (y_j)$ by:

$$\bar{y}_j = \begin{cases} -\bar{c}_j & \text{if either } \bar{c}_j < 0 \;\&\; \bar{x}_j < u_j\text{; or } \bar{c}_j > 0 \;\&\; \bar{x}_j > \ell_j \\ 0 & \text{if above conds. not met} \end{cases}$$

If $\bar{y}_D = 0$, $\bar{x}$ is a KKT pt., terminate.

If $\bar{y}_D \neq 0$, $\bar{c}_D\bar{y}_D < 0$, so $\bar{y}_D$ is a descent direction at $\bar{x}_D$ in the space of nonbasic variables, it is the negative reduced gradient direction.

Take a positive step length, λ say, from $\bar{x}_D$ in the space of nonbasic variables to the pt. $\bar{x}_D + \lambda y_D$.

The corresponding values of basic variables $x_B(\lambda)$ are to be determined uniquely from the square system of nonlinear eqs.

$$h(x_B(\lambda), \bar{x}_D + \lambda y_D) = 0$$

Newton's method is used to find $x_B(\lambda)$. Denote the vector x_B by ξ to avoid confusion. Beginning with $\xi_0 = \bar{x}_B$, Newton's method generates the sequence of iterates $\{\xi^s\}$ by the iteration

$$\xi^{r+1} = \xi^r - (\nabla_\xi h(\xi^r, \bar{x}_D + \lambda y_D))^{-1} h(\xi^r, \bar{x}_D + \lambda y_D)$$

For some r if (i) $||h(\xi^r, \bar{x}_D + \lambda y_D)|| < \epsilon =$ tolerance, and (ii) $\ell_B \leq \xi^r \leq u_B$, and (iii) $\theta(\xi^r, \bar{x}_D + \lambda y_D) < \theta(\bar{x}_B, \bar{x}_D)$, then fix $(x_B = \xi^r, x_D = \bar{x}_D + \lambda y_D)$ as the new feasible sol. and go to next iteration.

If (i) holds, but not (ii) or (iii) , go to *Step length reduction*.

If a preselected upper bound on Newton steps is reached and still (i) is not satisfied, go to *Step length reduction*.

Step length reduction: Replace λ by $\lambda/2$ and do the Newton iterations again from the beginning.