Optimization

Charles J. Geyer
School of Statistics
University of Minnesota

Stat 8054 Lecture Notes

One-Dimensional Optimization

- Look at a graph.
- Grid search.

One-Dimensional Zero Finding

Zero finding means solving $g(x) = 0$ for x.

If $f(x)$ is objective function for optimization and $g(x) = f'(x)$, zero of g is stationary point of f — not necessarily optimum.

- Bisection method.
- Secant method.
- Newton's method.
- R function `uniroot` (on-line help).

Multi-Dimensional Optimization

Minimize $f : S \to \mathbb{R}$.

Global Minimum

Point x such that

$$f(y) \geq f(x), \qquad \text{for all } x \in S$$

Local Minimum

Point x such that

$$f(y) \geq f(x), \qquad \text{for all } x \in W$$

where W is neighborhood of x in S.

Convex Optimization

A function $f : \mathbb{R}^d \to (-\infty, +\infty]$ is *convex* if

$$f\Big(tx + (1-t)y\Big) \leq tf(x) + (1-t)f(y), \qquad x, y \in \mathbb{R}^d \text{ and } 0 < t < 1.$$

Theorem: Every local minimum of a convex function is a global minimum.

A convex function is *strictly convex* if

$$f(x) < \infty \text{ and } f(y) < \infty \text{ and } 0 < t < 1$$
$$\text{implies } f\Big(tx + (1-t)y\Big) < tf(x) + (1-t)f(y).$$

Theorem: A strictly convex function has at most one local minimum.

Convex Optimization (cont.)

Minimum need not be achieved. Consider

$$f(x) = e^x, \qquad x \in \mathbb{R}$$

$f(x) \downarrow 0$ as $x \to -\infty$ but $f(x) > 0$ for all x.

One-Dimensional Derivative Tests

Necessary and sufficient for convexity

(a) $f'(x) \geq f'(y)$ whenever $x > y$

(b) $f(y) \geq f(x) + f'(x)(y - x)$ whenever $x \neq y$

(c) $f''(x) \geq 0$ for all x

With $\geq$ replaced by $>$ (a) and (b) are necessary and sufficient for strict convexity and (c) is sufficient but not necessary.

Multi-Dimensional Derivative Tests

Necessary and sufficient for convexity

(a) $\langle y - x, \nabla f(y) - \nabla f(x) \rangle \geq 0$ whenever $x \neq y$

(b) $f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle$ whenever $x \neq y$

(c) $\nabla^2 f(x)$ positive semidefinite for all x

With $\geq$ replaced by $>$ (a) and (b) are necessary and sufficient for strict convexity. With positive semidefinite replaced by positive definite (c) is sufficient but not necessary.

Concave and Convex, Maximization and Minimization

f is *convex* if and only if $-f$ is *concave*.

In order to avoid lots of duplication, optimization theory only discusses minimization. Minimizing f maximizes $-f$.

Switch from minimization to maximization by standing on your head.

Global Optimization

Except for easy case of convexity. Global optimization is hard.

Deterministic algorithms only available for very special problems, e. g., difference of convex functions, and computer code not widely available.

Many adaptive random search algorithms advertised to do global optimization, but this is only a hope. They don't stop at the first local minimum they find, but ....

Adaptive Random Search

IMHO an area rife with charlatanry.

Widely used methods are justified by metaphors, not math

- simulated annealing
- genetic algorithms

Scientists are people too. They like stories.

Metaphors do not make optimization algorithms effective.

Charlie's Rules for Adaptive Random Search

When looking in a sequence of random places x_1, x_2, ..., x_n.

1. Keep track of the x_k that minimizes, $f(x_k) \leq f(x_i)$, $1 \leq i \leq n$.

2. Search intensively near the lowest point found so far.

3. Then search elsewhere. Don't stop.

Charlie's Rules (cont.)

You can do things you can call simulated annealing or genetic algorithms and follow these rules.

You can do things you can call simulated annealing or genetic algorithms that don't follow these rules.

For any particular problem, if you bother to really think about it, you can probably invent an adaptive random search algorithm particularly for it that beats the metaphors.

Newton's Method

Method for solving simultaneous non-linear equations $g(x) = 0$, where $g : \mathbb{R}^d \to \mathbb{R}^d$.

$$g(y) \approx g(x) + \nabla g(x)(y - x)$$

where $\nabla g(x)$ is linear operator represented by matrix of partial derivatives $\partial g_i(x)/\partial x_j$.

Set equal to zero and solve

$$y = x - \Big(\nabla g(x)\Big)^{-1} g(x)$$

Iterate and hope for convergence.

Newton's Method (cont.)

In optimization, Newton minimizes quadratic approximation to function

$$m(y) = f(x) + \langle \nabla f(x), y - x \rangle + \tfrac{1}{2} \langle \nabla^2 f(x)(y - x), y - x \rangle$$

if $\nabla^2 f(x)$ is positive definite.

Newton update is

$$x_{n+1} = x_n - \left(\nabla^2 f(x_n) \right)^{-1} \nabla f(x_n)$$

Big Oh and Little Oh

$a_n = O(b_n)$ means a_n/b_n is bounded.

$a_n = o(b_n)$ means a_n/b_n converges to zero.

Also used for continuous, e. g., definition of derivative

$$f(y) = f(x) + \langle \nabla f(x), y - x \rangle + \|y - x\| \cdot o(y - x)$$

Rates of Convergence

Let x_1, x_2, ... be iterates of optimization algorithm. Suppose $x_n \to x$, and let $\epsilon_n = \|x_n - x\|$.

Linear convergence if $\epsilon_{n+1} = O(\epsilon_n)$.

Superlinear convergence if $\epsilon_{n+1} = o(\epsilon_n)$.

Quadratic convergence if $\epsilon_{n+1} = O(\epsilon_n^2)$.

Newton's Method Theorem

Suppose $\nabla f(y)$ and $\nabla^2 f(y)$ are continuous in neighborhood of solution x and $\nabla^2 f(x)$ is positive definite, then Newton's method converges superlinearly (if it converges).

In addition suppose

$$\nabla f(y) = \nabla f(x) + \nabla^2 f(x)(y - x) + O(\|y - x\|^2)$$
$$\nabla^2 f(y) = \nabla^2 f(x) + O(\|y - x\|)$$

then Newton's method converges quadratically (if it converges).

These conditions hold when f has continuous third partial derivatives in a neighborhood of x.

Dennis-Moré Theorem

Let x_1, x_2, ... be iterates of optimization algorithm, and let

$$\Delta_n = -\Big(\nabla^2 f(x_n)\Big)^{-1} \nabla f(x_n)$$

be the Newton step at x_n.

If the algorithm converges superlinearly, then

$$x_{n+1} - x_n = \Delta_n + o(\|\Delta_n\|)$$

Dennis-Moré Theorem (cont.)

Every superlinearly convergent optimization algorithm is asymptotically equivalent to Newton's method. (Its steps are equal to Newton steps plus negligible amount.)

Every superlinearly convergent optimization algorithm is asymptotically equivalent to any other superlinearly convergent optimization algorithm.

To get quadratic convergence must have

$$x_{n+1} - x_n = \Delta_n + O(\|\Delta_n\|^2)$$

Quasi-Newton

Update is

$$x_{n+1} = x_n - B_n \nabla f(x_n) \tag{1}$$

where

$$B_n \approx \left(\nabla^2 f(x_n)\right)^{-1}$$

If the approximation (1) is good enough, then the algorithm will converge superlinearly (see Wikipedia page).

Why Newton Sucks

Although Newton is asymptotically most wonderful, it is not guaranteed to converge when started an appreciable distance from the solution.

The least a minimization routine can do is take downhill steps.

Newton isn't even guaranteed to do that (Rweb demo).

Newton or quasi-Newton needs modification to be practically useful. Such modification is called *safeguarding*.

Line Search

When at x_n with a search direction p_n that points downhill

$$\langle \nabla f(x_n), p_n \rangle \leq 0,$$

Find a local minimum s_n of the function

$$g(s) = f(x_n + sp_n)$$

and take step

$$x_{n+1} = x_n + s_n p_n$$

Sloppy solution s_n to line search subproblem is o. k. (see Wolfe conditions).

Line Search (cont.)

If search direction p_n converges to Newton direction, then steps will be nearly Newton for large n, and algorithm will have super-linear convergence.

However, this algorithm is safeguarded, guaranteed to take only downhill steps.

Line Search (cont.)

Convergence to a local minimum cannot be guaranteed.

In order for convergence to a stationary point to be guaranteed, search directions p_n must satisfy *angle criterion:* if

$$g_n = \nabla f(x_n)$$

then

$$\frac{\langle p_n, g_n \rangle}{\|p_n\| \cdot \|g_n\|}$$

(cosine of angle between p_n and g_n) must be bounded away from zero.

Trust Region Methods

Trust region methods work by repeatedly solving the *trust region subproblem*

$$\begin{aligned} &\text{minimize } m_n(p) = f_n + \langle g_n, p\rangle + \tfrac{1}{2}\langle B_k p, p\rangle \\ &\text{subject to } \|p\| \le \Delta_n \end{aligned}$$

where

$$\begin{aligned} f_n &= f(x_n) \\ g_n &= \nabla f(x_n) \\ B_n &= \nabla^2 f(x_n) \end{aligned}$$

Like Newton, we minimize the quadratic approximation (Taylor series up to quadratic terms), but we only trust the approximation in a ball of radius Δ_n.

Trust Region Methods (cont.)

Let p_n be the solution of the trust region subproblem (see design document for trust package for details).

Let

$$\rho_n = \frac{f(x_n) - f(x_n + p_n)}{m(0) - m(p_n)}$$

- Numerator is actual decrease in objective function if we take step p_n (positive if step is downhill).

- Denominator is decrease in objective function of trust region subproblem for step p_n (always positive).

Trust Region Methods (cont.)

Accept proposed step only if appreciable decrease in objective function.

If $\rho_n \geq 1/4$, then $x_{n+1} = x_n + p_n$, otherwise $x_{n+1} = x_n$.

Adjust trust region radius based on quality of proposed step.

If $\rho_k \leq 1/4$, then $\Delta_{n+1} = \|p_n\|/4$.

If $\rho_k \geq 3/4$ and $\|p_n\| = \Delta_n$, then $\Delta_{n+1} = \min(2\Delta_n, \Delta_{\max})$.

Otherwise $\Delta_{n+1} = \Delta_n$.

Trust Region Properties

Any cluster point x of sequence of iterates satisfies first and second order necessary conditions for optimality.

$$\nabla f(x) = 0$$
$$\nabla^2 f(x) \text{ is positive semidefinite}$$

Converges quadratically if solution satisfies first and second order sufficient conditions (same as above except positive definite).

Not bothered by restricted domain of objective function, so long as solution is in interior.

Just works (trust package web page).

Constrained Optimization

$$\begin{aligned} \text{minimize } & f(x) \\ \text{subject to } & g_i(x) = 0, \qquad i \in E \\ & g_i(x) \leq 0, \qquad i \in I \end{aligned} \tag{2}$$

where E and I are disjoint finite sets.

Say x is *feasible* if constraints hold.

Lagrange Multipliers

Lagrangian function

$$L(x, \lambda) = f(x) + \sum_{i \in E \cup I} \lambda_i g_i(x)$$

Theorem: If there exist x^* and λ such that

(a) x^* minimizes $x \mapsto L(x, \lambda)$,

(b) $g_i(x^*) = 0$, $i \in E$ and $g_i(x^*) \leq 0$, $i \in I$,

(c) $\lambda_i \geq 0$, $i \in I$, and

(d) $\lambda_i g_i(x^*) = 0$, $i \in I$.

Then x^* solves the constrained problem (2).

Lagrange Multipliers (cont.)

Proof: By (a)

$$L(x, \lambda) \geq L(x^*, \lambda) \tag{3}$$

so for feasible x

$$\begin{aligned} f(x) &\geq f(x^*) + \sum_{i \in E \cup I} \lambda_i g_i(x^*) - \sum_{i \in E \cup I} \lambda_i g_i(x) \\ &= f(x^*) - \sum_{i \in E \cup I} \lambda_i g_i(x) \\ &\geq f(x^*) \end{aligned}$$

top inequality is (3), equality is (d), bottom inequality is (c) and feasibility of x.

Kuhn-Tucker Conditions

(a) $\nabla f(x^*) + \sum_{i \in E \cup I} \lambda_i \nabla g_i(x^*)$,

(b) $g_i(x^*) = 0$, $i \in E$ and $g_i(x^*) \leq 0$, $i \in I$,

(c) $\lambda_i \geq 0$, $i \in I$, and

(d) $\lambda_i g_i(x^*) = 0$, $i \in I$.

Only (a) changed. Rest stay same.

Now no theorem. (a) too weak.

Existence of Lagrange Multipliers

Lagrange multipliers always exist such that theorem applies if

- all constraints affine,
- objective function convex, inequality constraint functions convex, and equality constraint functions affine, or
- the set

$$\{ \nabla g_i(x^*) : i \in E \cup I \text{ and } g_i(x^*) = 0 \}$$

  is linearly independent.

(May exist in other cases, but not guaranteed.)

R packages

Problem (2) called *linear programming* if all functions affine. R packages `linprog` (on-line help), `lpSolve` (on-line help), and `rcdd` (on-line help), do that.

Problem (2) called *quadratic programming* if objective function is quadratic and constraint functions affine. R package `quadprog` (on-line help), does that.

Problem (2) called *nonlinear programming* if general functions are allowed. R package `npsol` (not installed, not free software, obtain from me) does that.

Isotonic Regression

Suppose we observe pairs (y_i, x_i), $i = 1, \ldots, n$ and we wish to estimate the regression function $E(Y \mid X)$ subject to the condition that it is monotone. How?

Treat x_i as fixed (condition on them). Assume

$$Y_i \sim \text{Normal}(\mu_i, \sigma^2)$$

and

$$x_i \leq x_j \quad \text{implies} \quad \mu_i \leq \mu_j$$

What is the MLE of the vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)$?

Isotonic Regression (cont.)

Problem solved in 1955 independently by two different groups of statisticians.

Let z_1, ..., z_k be the unique x values in sorted order, and define

$$w_j = \sum_{\substack{i=1 \\ x_i = z_j}}^{n} 1$$

$$V_j = \frac{1}{w_j} \sum_{\substack{i=1 \\ x_i = z_j}}^{n} Y_i$$

Isotonic Regression (cont.)

Then equivalent weighted least squares problem is the following. Assume

$$V_j \sim \text{Normal}(\mu_j, \sigma^2/w_j)$$

and

$$\mu_1 \leq \mu_2 \leq \cdots \leq \mu_k$$

What is the MLE of the vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)$?

Isotonic Regression (cont.)

Lagrangian function is

$$\sum_{j=1}^{k} w_j (v_j - \mu_j)^2 + \sum_{i=1}^{k-1} \lambda_j (\mu_{j+1} - \mu_j)$$

Isotonic Regression (cont.)

Kuhn-Tucker conditions are

(a) $-2w_j(v_j - \mu_j) - \lambda_j + \lambda_{j-1} = 0,\ j = 1,\ \ldots,\ k.$

(b) $\mu_i \leq \mu_j,\ 1 \leq i \leq j \leq k.$

(c) $\lambda_j \geq 0,\ j = 1,\ \ldots,\ k,$ and

(d) $\lambda_j(\mu_{j+1} - \mu_j) = 0,\ j = 1,\ \ldots,\ k - 1.$

where to make (a) simple we define $\lambda_0 = \lambda_k = 0.$

Isotonic Regression (cont.)

Solution must be step function. Have blocks of equal μ's. Let j_i, $i = 1, \ldots, m$ be the starting indices of the blocks, so

$$\mu_{j_l-1} < \mu_{j_l} = \cdots = \mu_{j_{l+1}-1} < \mu_{j_{l+1}}$$

where to make this simple we define $\mu_0 = -\infty$ and $\mu_{k+1} = +\infty$.

Complementary slackness implies $\lambda_{j_l-1} = \lambda_{j_{l+1}-1} = 0$. Hence

$$\begin{aligned} 0 &= \sum_{j=j_l}^{j_{l+1}-1} \Big[-2w_j(v_j - \mu_j) - \lambda_j + \lambda_{j-1} \Big] \\ &= -2 \sum_{j=j_l}^{j_{l+1}-1} w_j(v_j - \mu) \end{aligned}$$

where $\mu = \mu_{j_l} = \cdots = \mu_{j_{l+1}-1}$.

Isotonic Regression (cont.)

Hence

$$\mu = \frac{\sum_{j=j_l}^{j_{l+1}-1} w_j v_j}{\sum_{j=j_l}^{j_{l+1}-1} w_j}$$

is the value for this block that satisfies the first and fourth Kuhn-Tucker conditions. The same goes for every block. The estimate for the block is the average of the data values for the block.

So that gives us an algorithm: try all possible partitions into blocks. There is exactly one such partition that satisfies primal feasibility (makes an increasing function). And that is the solution.

Isotonic Regression (cont.)

The pool adjacent violators algorithm (PAVA) is much more efficient.

1. Initialize. Set $\boldsymbol{\mu} = \mathbf{v}$. Every point is a block by itself.

2. Test. If $\boldsymbol{\mu}$ satisfies primal feasibility stop. Have solution.

3. Pool. Since test fails, there exist adjacent blocks that violate primal feasibility. Pool them, setting the estimate for the new block to be the average of all data for it.

4. go to 2.

Exponential Families

One-parameter exponential family has log likelihood

$$l(\theta) = x\theta - c(\theta)$$

and derivatives

$$l'(\theta) = x - c'(\theta)$$
$$l''(\theta) = -c''(\theta)$$

The differentiation under the integral sign identities

$$E_\theta\{l'(\theta)\} = 0$$
$$\text{var}_\theta\{l'(\theta)\} = -E_\theta\{l''(\theta)\}$$

prove

$$E_\theta(X) = c'(\theta)$$
$$\text{var}_\theta(X) = c''(\theta)$$

Exponential Families (cont.)

From this it follows that the mean value parameter

$$\mu = E_\theta(X) = c'(\theta)$$

is a strictly increasing function of the natural parameter θ, because the derivative of the map $c' : \mu \to \theta$ is c'', which is positive, being a variance.

Hence if θ_i are the natural parameters and μ_i the mean value parameters, we have $\theta_i \leq \theta_j$ if and only if $\mu_i \leq \mu_j$.

Isotonic Regression and Exponential Families

Now we assume Y_i all have distributions in the same one-parameter exponential family, and θ_i are the natural parameters.

We do isotonic regression, maximum likelihood subject to

$$\theta_1 \le \theta_2 \cdots \le \theta_n$$

The Lagrangian is

$$\sum_{i=1}^{n} \Big[y_i \theta_i - c(\theta_i) \Big] + \sum_{i=1}^{n-1} \lambda_i (\theta_{i+1} - \theta_i)$$

The first Kuhn-Tucker condition is

$$\begin{aligned} 0 &= y_j - c'(\theta_j) - \lambda_j + \lambda_{j-1} \\ &= y_j - \mu_j - \lambda_j + \lambda_{j-1} \end{aligned}$$

Isotonic Regression and Exponential Families (cont.)

The rest of the Kuhn-Tucker conditions are the same as when we assumed the response was normal.

When we write $\mu_j = \theta_j$ as on the preceding slide, the Kuhn-Tucker conditions only involve the mean value parameters and *are the same regardless of the exponential family involved*.

Hence the same algorithm, PAVA, solves all exponential family isotonic regression problems!