Operations Research Techniques and Algorithms (620-361)

Christina Burt

c.burt@ms.unimelb.edu.au
Telephone: 8344 1797
Office: 139 Barry St

Yao-ban Chan
y.chan@ms.unimelb.edu.au
Telephone: 8344 9073
Office: Room 198, Richard Berry Building

May 21, 2008

You are likely to achieve good marks if you can answer yes for the following questions for each method learned this semester:

- Do I know when it is appropriate to apply this algorithm?
- Can I implement the algorithm by hand for a few iterations?
- Do I know when to stop this algorithm?
- Can I explain how the algorithm works and what is happening?
- Can I list the pros and cons of this algorithm?
- Would I be able to compare this algorithm to another for a particular problem?

Operations research techniques and algorithms

Unconstrained 1−D methods
- Fibonacci search
- Golden section search
- False position search
- Newton's method

Unconstrained n−D methods
- General descent method
- Steepest descent method
- Newton's method
- BFGS method (Quasi−Newton)

Constrained methods
- Lagrange's method
- Karush−Kuhn−Tucker method
- l2 penalty method
- log−barrier method
- exact penalty method
- Lagrangian dual method
- Wolfe dual method
- Augmented Lagrangian method

Armijo-Goldstein and Wolff conditions

- The procedure finds a significant improvement, not the minimum in a descent direction.
- AG ensures that the stepsize is never too big $f(t) \leq f(0) + t\sigma f'(0)$.
- W ensures that the stepsize is never too small $f'(t) \geq \mu f'(0)$.
- $\mu \geq \sigma$ ensures that these two regions always overlap.

Directional derivative

- This is the one-dimensional derivative of the slice of f along direction d.

- The directional derivative is equivalent to $f'(x; d) = \nabla f(x)^T d$.

- That is, you multiply the gradient of the function by the descent direction.

Convexity

- A function is convex if $f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y)$.

- This means that if you draw a line between two points x and y, the rest of the function (between those points) lies below the line you drew.

- We know that convex functions must always have at least a positive semi-definite Hessian.

Matrix inverse

- A matrix is invertible if it is positive definite (although the converse is not true).

- If a matrix is symmetric, then it is also positive definite iff it is invertible and its inverse is positive definite.

- For 2x2 matrices we can use the following formula to find the inverse

$$A^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

First and second order conditions for a local minimum

The first order condition is (necessary but not sufficient):

the gradient at the optimum is zero.

The second order condition is (sufficient):

the Hessian is positive definite
and
the gradient at the optimum is zero.

(For a local **maximum**, the Hessian will be negative definite.)

Fibonacci search

■ We use this method to approximate the minimum of a unimodal function in 1-dimension when we start with an interval and can calculate the number of iterations precisely before we start the algorithm.

■ To calculate n, generate F_n until you find $F_n > \frac{(b-a)}{2\epsilon}$.

■ We reduce the interval by a factor of $\frac{F_{k-1}}{F_k}$ each time.

■ The size of the interval is related to the Fibonacci sequence:

$$\ldots, 8\epsilon, 5\epsilon, 3\epsilon, 2\epsilon, \epsilon$$

Golden section search

■ We use this method to approximate the minimum of a unimodal function in 1-dimension when we start with an interval and don't want to calculate the number of iterations precisely before we start the algorithm.

■ We take γ to be the limit of $\frac{F_{k-1}}{F_k}$, i.e. $\gamma = 0.618$.

■ We reduce the interval by a factor of γ each time.

■ Think about how this algorithm converges differently to the Fibonacci search, even though it looks very similar.

False position search

- We use this method to approximate the minimum of a unimodal function in 1-dimension when we can calculate the derivative and it is continuous and increasing on $[a, b]$.
- We study the derivative and find where the line between a and b intersects the x-axis.
- If the intersection yields $f'(x) < 0$ then x becomes our new a.
- If the intersection yields $f'(x) > 0$ then x becomes our new b.
- We stop when $f'(x) < tol$.
- We only need to calculate the derivative for this method.

Newton's method

- We use this method to approximate the minimum of a unimodal function in 1-dimension when we can calculate both the derivative and second derivative, both of which are continuous.

- Starting from an initial point, we calculate the second derivative and first derivative to draw a line that intersects the x-axis.

- The intersection provides a new x estimate.

- If an inflection point is very close to the minimum, this can cause $f''(x)$ to be zero and the algorithm fails (because we will try to divide by zero).

Revision

Steepest descent method

■ We use this method to approximate the minimum of a unimodal function in *n*-dimensions when we start with an interval and can calculate the gradient of the function at certain points.

■ We choose the direction with steepest descent and travel in this direction for some stepsize, t.

■ We choose $d = -\nabla f(x)$ because π is the solution to min $cos\theta||\nabla f(x)||, s.t.\theta \in \Re$.

■ Each descent direction is perpendicular to the previous directions only if we take the exact minimum for the stepsize.

Newton's method

■ We use this method to approximate the minimum of a unimodal function in n-dimensions when we can calculate the inverse of the Hessian matrix.

■ We minimise the quadratic approximation of the function at each iteration. $x_k \to x^*$ superlinearly ... and quadratically if f is C^3.

■ The descent direction is $d_k = -\nabla^2 f(x_k)^{-1} \nabla f(x_k)$.

■ It is not always guaranteed that $-\nabla^2 f(x_k)$ is invertible or that d is a descent direction.

■ We check both of these by checking that $\nabla^2 f(x_k)$ is positive definite at each iteration.

Quasi Newton methods

- We use this method to approximate the minimum of a function in n-dimensions when we cannot calculate the inverse of the Hessian at every iteration.

- We approximate the Hessian by H_k, to get the descent direction $d_k = -H_k \nabla f(x_k)$.

- If d_k and t_k satisfy the Wolff condition, then H_k is symmetric and positive definite.

- If f is convex and C^2, and $\nabla^2 f(x)$ is positive definite, then BFGS converges superlinearly.

Revision

Basic background

1-D unconstrained methods

n-D unconstrained methods

Constrained methods

Equality-constrained problems

- We have a problem of the form:

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & h(x) = 0 \end{array}$$

- We define the Lagrangian:

$$L(x, \eta) = f(x) + \sum_{j=1}^{q} \eta_j h_j(x).$$

- If f and h are C^1 and a constraint qualification holds at a local minimum x^*, then the following first-order conditions hold:

$$\nabla_x L(x^*, \eta^*) = 0$$
$$h(x^*) = 0.$$

Equality-constrained problems

■ The first condition tells us that at the minimum, the gradient of the objective function can be written as a linear combination of the gradients of the constraint functions (and they are parallel if there is only one constraint).

■ The second condition tells us that x^* must be feasible (which is obvious).

■ This only happens if a constraint qualification holds at x^*. Constraint qualifications are:

- h is an affine function (linear plus constant); or
- All gradients $\nabla h_j(x)$ are linearly independent at x^*.

We need these to ensure that the constraints are not 'pathological'.

Equality-constrained problems

■ We can sometimes solve for stationary x and η by algebraically manipulating the first-order conditions.

■ To be sure that a stationary point is indeed a local minimum, we need a sufficient condition:

- f is C^1 and convex, and h is affine (this gives a *global* minimum); or
- $\nabla^2_{xx} L(x^*, \eta^*)$ is positive definite on the nullspace of the transpose of the Jacobian $\nabla h(x^*)^T$.

■ The nullspace of $\nabla h(x^*)^T$ (a $q \times n$ matrix) is the set of all vectors d which satisfy

$$\nabla h(x^*)^T d = 0.$$

Equality-constrained problems

- Note that this isn't the same as requiring that the Lagrangian is convex; we only need it to be 'somewhat convex' at a certain point.

- Lagrange multipliers can be viewed as the rates of change of the optimal function as the level of constraint changes.

- If the right-hand side of a constraint j changes by an amount Δ, then the optimal function value will change by $-\eta_j^*\Delta$.

- Alternatively, $-\eta^*$ is the rate of change of optimal function value with respect to the change in the right-hand side of the constraints.

Solving an equality-constrained problem:

1. Re-arrange the problem to get it into standard form (notably, minimisation and $h(x) = 0$).
2. Write down the Lagrangian with one η_j for every constraint.
3. Find the gradient of the Lagrangian and set this to be zero. Add the constraint.
4. If possible, solve this system of equations.
5. Check that a constraint qualification holds at x^*.
6. Check that a sufficient condition holds at (x^*, η^*).

Inequality constrained problems

- We have a problem of the form:

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & g(x) \leq 0 \\ & h(x) = 0 \end{array}$$

- An inequality constraint is active at x if $g(x) = 0$. Equality constraints are always active.

- Our Lagrangian is:

$$L(x, \lambda, \eta) = f(x) + \sum_{i=1}^{p} \lambda_i g_i(x) + \sum_{j=1}^{q} \eta_j h_j(x).$$

Inequality constrained problems — KKT conditions

- If f, g and h are C^1 and a constraint qualification holds at a local minimum x^*, then the KKT conditions hold for some multipliers (λ^*, η^*):

KKTa	$\nabla_x L(x^*, \eta^*, \lambda^*) = 0$
KKTb	$g(x^*) \leq 0$
	$\lambda^* \geq 0$
	$\lambda_i^* g_i(x^*) = 0$
KKTc	$h(x^*) = 0$

Inequality constrained problems

■ KKTa tells us that the objective function gradient is a linear combination of active gradients (inactive constraints have zero multiplier).

■ KKTb tells us that the inequality constraints hold, and that the multiplier corresponding to inactive constraints is 0.

■ KKTc tells us that the equality constraints hold.

■ To solve this system algebraically, (if no easy way presents itself) we try to make each possible set of constraints active and see if all equalities and inequalities hold for the solution.

Inequality constrained problems

■ The KKT conditions only work if a constraint qualification holds at the minimum x^*. Possible qualifications are:

- All active inequality constraints and equality constraints are affine;
- All active constraint gradients $\nabla g_i(x^*)$ and $\nabla h_j(x^*)$ are linearly independent at x^*;
- (Mangasarian-Fromovitz) The equality constraint gradients $\nabla h_j(x^*)$ are linearly independent at x^*, and there exists a d such that $\nabla h(x^*)^T d = 0$ and $\nabla g_i(x^*)^T d < 0$ for all active inequality constraints; or
- (Slater) All inequality constraints are convex, all equality constraints are affine, and there exists x' such that $h(x') = 0$ and $g_i(x') < 0$ for all inequality constraints.

Inequality constrained problems

■ Constraint qualifications imply that min $\Rightarrow$ KKT. For a KKT point to be a local minimum we need a sufficient condition. These include:

- The program is a convex program — f and g_i is convex, and h is affine; or
- $\nabla^2_{xx} L(x^*, \lambda^*, \eta^*)$ is positive definite on the critical cone.

■ The critical cone is defined as all $d \in \Re^n$ such that:

- $\langle \nabla g_i(x^*), d \rangle \leq 0$ if i is an active constraint with 0 corresponding multiplier;
- $\langle \nabla g_i(x^*), d \rangle = 0$ if $\lambda_i > 0$; and
- $\langle \nabla h_j(x^*), d \rangle = 0$ for all equality constraints.

■ The second condition is much easier to check if we restrict all active constraints to have strictly positive multipliers.

Solving an inequality constrained problem:

1. Re-arrange the problem so it is in standard form (notably $g(x) \leq 0$).
2. Write down the Lagrangian equation for the problem with one λ_i for each inequality constraint.
3. Write down the KKT conditions.
4. Solve the KKT system algebraically.
5. Check that a constraint qualification holds at your solution point x^*.
6. Check that a sufficient condition holds at your solution point x^*, λ^*, η^*.

Inequality constrained problems

■ KKT multipliers have the same interpretation as Lagrange multipliers for an equality-constrained problem.

■ If you change the right-hand side of a constraint i by Δ, the optimal function value changes by $-\lambda_i^* \Delta$.

■ Alternatively $-\lambda^*$ and $-\eta^*$ are the rates of change of the optimal function value with respect to the change in the right-hand side of the inequality and equality constraints respectively.

Penalty methods

- If we cannot solve the KKT condition algebraically, we use a numerical method such as penalty methods.

- These convert a constrained problem into an unconstrained problem by 'penalising' any violation of the constraints.

- The l_2 penalty function is:

$$P_\alpha(x) = f(x) + \frac{\alpha}{2}\left(\sum_{i=1}^{p}[g_i(x)_+]^2 + \sum_{j=1}^{q} h_j(x)^2\right).$$

- α is called the *penalty parameter*.

Penalty methods

■ If f, g and h are C^1, and we minimise $P_\alpha(x)$ for a succession of penalty parameters that tend to ∞, then the solutions will tend toward a feasible local minimum of the original problem.

■ Also $\alpha_k g(x^k)_+$ tends towards the optimal KKT multiplier λ^*.

■ And $\alpha_k h(x^k)$ tends towards the optimal KKT multiplier η^*.

■ The l_2 penalty function is C^1 but not C^2, so we can use steepest descent or BFGS to minimise it, but not Newton's method.

Penalty methods

■ The log-barrier penalty function is:

$$P_\alpha(x) = f(x) - \frac{1}{\alpha}\sum_{i=1}^{p}\ln(-g_i(x)) + \frac{\alpha}{2}\sum_{j=1}^{q}h_j(x)^2.$$

■ This is undefined for any point which violates an inequality constraint.

■ Again we minimise $P_\alpha(x)$ for a succession of penalty parameters that tend to ∞. The solutions will tend toward a feasible local minimum of the original problem.

Penalty methods

■ Again, $\alpha_k h(x^k)$ tends towards the optimal KKT multiplier η^*.

■ Now $-\frac{1}{\alpha g_i(x^k)}$ tends towards the optimal KKT multiplier λ_i^*.

■ The log-barrier penalty function is as smooth as the original functions f, g and h, so we can (probably) apply Newton's method.

Penalty methods

- Exact penalty methods are designed so that they find the exact minimum of the original program for large but finite α.

- This is in contrast to l_2 and log-barrier methods which only find the minimum in the limit $\alpha \to \infty$.

- An exact penalty function is the l_1 function:

$$P_\alpha(x) = f(x) + \alpha \left(\sum_{i=1}^{p} |g_i(x)_+| + \sum_{j=1}^{q} |h_j(x)| \right).$$

- This function is not C^1 so we cannot use our regular methods to minimise it.

Augmented Lagrangian method

■ For equality-only problems we can penalise the Lagrangian instead of the objective function. This gives the *augmented Lagrangian*:

$$L_\alpha^A(x,\eta) = f(x) + \sum_{j=1}^{q} \eta_j h_j(x) + \frac{\alpha}{2} \sum_{j=1}^{q} h_j(x)^2.$$

■ We choose an initial penalty parameter and an initial guess for the KKT multipliers.

■ We then minimise the augmented Lagrangian with respect to x to get a new estimate x^k.

Augmented Lagrangian method

- We then update the penalty parameter, and the KKT multiplier estimate via:

$$\eta_j^{k+1} = \eta_j^k + \alpha_k h_j(x^k).$$

- If we know the optimal KKT multipliers, the augmented Lagrangian is an exact penalty function.

- However, we generally do not.

- Even if we do not, the iterates x^k will tend to the optimal solution to the original program as $\alpha_k \to \infty$.

- Also η^k will tend to η^*, the optimal KKT multipliers.

Convex programming

■ A program is a convex program if (and only if) f is convex, $g_i(x)$ is convex, and $h_j(x)$ is affine.

■ Remember that a positive linear combination of convex functions is convex. Also both an affine function and its negative are convex.

■ We already know that any KKT point of a convex program is a global minimiser.

■ It also minimises the Lagrangian.

Convex programming

- The triple (x^*, λ^*, η^*) is a KKT point of a convex program if and only if, for all $\lambda \geq 0$ and x, η,

$$L(x^*, \lambda, \eta) \leq L(x^*, \lambda^*, \eta^*) \leq L(x, \lambda^*, \eta^*).$$

- This tells us that x^* minimises the Lagrangian evaluated at λ^* and η^* and taken as a function of x only.

- Also λ^* and η^* maximise the Lagrangian evaluated at x^* and taken as a function of λ and η.

Duality

■ The penalty interpretation of the original problem is

$$\min_{x} \phi(x)$$

where

$$\phi(x) = \max_{\lambda \geq 0, \eta} L(x, \lambda, \eta).$$

■ This is an ideal penalty function because it equals $f(x)$ if x is feasible and ∞ if x is infeasible.

■ The Langrangian dual is

$$\max_{\lambda \geq 0, \eta} \psi(\lambda, \eta)$$

where

$$\psi(\lambda, \eta) = \min_{x} L(x, \lambda, \eta).$$

Duality

- Weak Lagrangian duality: If x is feasible and $\lambda \geq 0$, then $\psi(\lambda, \eta) \leq \phi(x)$.
- This also gives us a lower bound on the original objective function.
- Strong Lagrangian duality: (x^*, λ^*, η^*) is a KKT point if and only if $\lambda^* \geq 0$ and $\psi(\lambda^*, \eta^*) = \phi(x^*)$.
- This can be used to identify optimal points.

Duality

- By adding unconstrained first-order conditions for ψ we get the Wolfe dual:

$$\begin{array}{ll} \max_{x,\lambda,\eta} & L(x,\lambda,\eta) \\ \text{s.t.} & \lambda \geq 0 \\ & \nabla_x L(x,\lambda,\eta) = 0. \end{array}$$

- This optimises all 3 variable sets, x, λ and η.

Duality

- Weak Wolfe duality: If x is primal feasible and (x', λ, η) is Wolfe dual feasible, then $L(x', \lambda, \eta) \leq f(x)$.

- This again gives us a lower bound on the original objective function.

- Strong Wolfe duality: (x^*, λ^*, η^*) is a KKT point of the primal if and only if x^* is primal feasible, (x^*, λ^*, η^*) is Wolfe dual feasible, and $L(x^*, \lambda^*, \eta^*) = f(x^*)$.

- Again this identifies optimal points.