Operations Research Techniques and Algorithms (620-361)

Dr Yao-ban Chan

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

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

May 16, 2008

Wolfe duality also has weak and strong duality properties.

Theorem. Suppose (NLP) is a convex program.

1. (Weak Wolfe duality) For any feasible points x of (NLP) and (x', λ, η) of the Wolfe dual, we have $L(x', \lambda, \eta) \leq f(x)$.
2. (Strong Wolfe duality) A triple (x^*, λ^*, η^*) is a KKT point of the primal program if and only if x^* is primal feasible, (x^*, λ^*, η^*) is feasible for the Wolfe dual and $L(x^*, \lambda^*, \eta^*) = f(x^*)$. In this case, x^* is a global minimiser for the primal and (x^*, λ^*, η^*) is a global maximiser for the Wolfe dual.

Proof.

1. Because the original non-linear program is a convex program, we know that f and g are convex and h is affine. Since $\lambda \geq 0$ from feasibility of the dual point,

$$L(x, \lambda, \eta) = f(x) + \sum_i \lambda_i g_i(x) + \sum_j \lambda_j h_j(x)$$

is a convex function.

Because (x', λ, η) is feasible for the Wolfe dual,

$$\nabla_x L(x', \lambda, \eta) = 0.$$

Since L is convex, this implies that x' is a global minimiser for the Lagrangian considered as a function of x only.

Therefore

$$\begin{aligned} L(x', \lambda, \eta) &\leq L(x, \lambda, \eta) \\ &= f(x) + \sum_i \lambda_i g_i(x) + \sum_j \eta_j h_j(x) \\ &\leq f(x). \end{aligned}$$

The last inequality comes about because x is feasible for the primal, and so $g(x) \leq 0$ and $h(x) = 0$.

2. Let (x^*, λ^*, η^*) be a KKT point of the original program. KKTa gives us

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

and KKTb gives us

$$\lambda^* \geq 0,$$

so it is clear that (x^*, λ^*, η^*) is feasible for the Wolfe dual.

Now from KKTb, we know that $(\lambda^*)^T g(x) = 0$, and from KKTc, $(\eta^*)^T h(x) = 0$. Therefore

$$L(x^*, \lambda^*, \eta^*) = f(x^*) + (\lambda^*)^T g(x) + (\eta^*)^T h(x) = f(x^*).$$

But from weak duality, it is impossible for any feasible point of the Wolfe dual to have a higher objective function value than this. Therefore (x^*, λ^*, η^*) is a global maximum for the Wolfe dual with optimal function value $f(x^*)$. Since the original program is convex, any KKT point must be a global minimiser.

Now suppose that (x^*, λ^*, η^*) is a feasible point for the Wolfe dual which satisfies $L(x^*, \lambda^*, \eta^*) = f(x^*)$. From weak duality, no feasible point can have a higher Wolfe dual objective than $f(x^*)$, so (x^*, λ^*, η^*) is a global maximiser of the Wolfe dual.

Again from weak duality, no feasible point of the original program can have a smaller objective function than $L(x^*, \lambda^*, \eta^*)$, and therefore x^* is a global minimiser for the original program (and also a KKT point). This proves the theorem.

Note that for strong duality it is insufficient simply to have a (primal) feasible point x which satisfies $L(x, \lambda, \eta) = f(x)$ for some λ and η. For example, by setting $\lambda = \eta = 0$ we derive

$$L(x, \lambda, \eta) = f(x) + 0^T g(x) + 0^T h(x) = f(x)$$

for any feasible x. However not all feasible x are optimal points!

We also require that (x, λ, η) is feasible for the Wolfe dual (i.e. that it satisfies the KKTa condition).

Example. We return once more to the problem we have been using as an example:

$$\begin{aligned} \min \quad & x_1^2 + x_2^2 \\ \text{s.t.} \quad & -x_1 - x_2 + 4 \leq 0 \\ & x_1, x_2 \geq 0. \end{aligned}$$

We have shown previously that the Wolfe dual can be simplified to:

$$\begin{array}{ll} \max & -x_1^2 - x_2^2 + 4\lambda_1 \\ \text{s.t.} & \lambda_1 \geq 0 \\ & 2x_1 - \lambda_1 \geq 0 \\ & 2x_2 - \lambda_1 \geq 0. \end{array}$$

Weak duality gives us lower bounds on the optimal objective function of the primal (which we denoted by z).

For instance, $(x_1, x_2, \lambda_1) = (0, 0, 0)$, $(1, 1, 1)$ and $(2, 2, 4)$ are all feasible for the (reduced) Wolfe dual. Therefore

$$\begin{aligned} z &\geq -0^2 - 0^2 + 4 \times 0 = 0 \\ z &\geq -1^2 - 1^2 + 4 \times 1 = 2 \\ z &\geq -2^2 - 2^2 + 4 \times 2 = 8. \end{aligned}$$

Again we know beforehand that $z = 8$, so we can see that these inequalities are true.

Now we know that $(2,2)$ is feasible for the primal program, $((2,2),(4,0,0))$ is feasible for the full Wolfe dual, and that

$$L((2,2),(4,0,0)) = 2^2 + 2^2 + 4(-2-2+4) + 0(-2) + 0(-2) = 8$$

and

$$f(2,2) = 2^2 + 2^2 = 8.$$

Strong duality now tells us that the original non-linear program has a global minimiser at $(2,2)$, with optimal KKT multipliers $(4,0,0)$. Furthermore the Wolfe dual has a global maximum at $((2,2),(4,0,0))$.

The augmented Lagrangian method

As one consequence of the work we have done on convex programming, we will take a brief look at the augmented Lagrangian method (also known as the *method of multipliers*). The framework for this method comes from what we have discussed with convex programs, but the method itself is not limited to convex programs.

We will look at the method in the case of equality constraints only:

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

Remember from convex programming that if we have the optimal multipliers η^*, we can minimise the Lagrangian $L(x, \eta^*)$ over all x to derive the optimal point x^*. We do not have the optimal multipliers, but maybe we can approximate them.

However, if we do this for approximate KKT multipliers, we have no guarantee that x will be feasible. To solve this we use a penalty approach. Specifically, we apply an l_2 penalty term to the Lagrangian.

This gives the augmented Lagrangian:

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

This is a combination of the Lagrangian and the l_2 penalty function. If we have the correct multipliers and minimise this function, then we will effective be applying a penalty function approach to minimising the Lagrangian.

Of course, we do not have the optimal KKT multipliers. Since this is a numerical method, we guess them at the start, and then successively update them every time we change the penalty parameter. But how do we do this?

Recall that in the original l_2 penalty method, we found that

$$\alpha_k h_j(x^k) \to \eta_j^* \text{ as } k \to \infty.$$

Rearranging this gives

$$h_j(x^k) \approx \frac{\eta_j^*}{\alpha_k}.$$

However, for x^k to be feasible, we will need $h_j(x^k) = 0$. Thus we are actually making the constraints invalid with the l_2 penalty approach. Although $\frac{\eta_j^*}{\alpha} \to 0$ as $\alpha \to \infty$, this is still undesirable.

The augmented Lagrangian function, however, has a minimum when the gradient is 0, i.e.

$$\nabla_x L_\alpha^A(x, \eta) = \nabla f(x) + \sum_j (\eta_j + \alpha h_j(x)) \nabla h_j(x) = 0.$$

As we let $\alpha \to \infty$, we can apply the same logic that we applied to the l_2 penalty method to find that, as $k \to \infty$,

$$\eta_j^k + \alpha_k h_j(x^k) \to \eta_j^*,$$

where η^k is our approximate multiplier at iteration k.

Rearranging this gives

$$h_j(x^k) \approx \frac{1}{\alpha_k}(\eta_j^* - \eta_j^k).$$

This shows us that if our approximate multipliers are very close to the actual multipliers, $h_j(x^k)$ will be very close to 0 — much closer than for the l_2 penalty method.

In fact, this line of reasoning also gives us a way to update our approximate multipliers η^k. We have

$$\eta_j^* \approx \eta_j^k + \alpha_k h_j(x^k)$$

and therefore the most logical update scheme is

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

Remember that we also update our penalty parameter α_k with each iteration.

The augmented Lagrangian algorithm

1. Select a penalty parameter α and initial approximate multiplier η^0. Set $k = 0$.
2. Find a (unconstrained) minimiser x^k of $L^A_{\alpha_k}(x, \eta^k)$.
3. If x^k satisfies some convergence test, stop.
4. Update $\eta_j^{k+1} = \eta_j^k + \alpha_k h_j(x^k)$.
5. Update $\alpha_{k+1} > \alpha_k$.
6. Update $k = k + 1$.
7. Return to step 2.

It is possible to adapt the augmented Lagrangian method to inequality-constrained problems, but it is somewhat more complicated.

Two results provide us with justification for this method.

Theorem. Let x^* be a local minimum of the non-linear program with optimal KKT multipliers λ^*, η^*. Then there exists a value α' such that for all $\alpha > \alpha'$, x^* is a local minimum of $L_\alpha^A(x, \lambda^*, \eta^*)$.

This tells us that if we have the exact multipliers, the augmented Lagrangian is actually an exact penalty function!

Theorem. Suppose that the augmented Lagrangian method gives us a series of iterates x^k. Then as $\alpha_k \to \infty$, x^k has a limit point x^* which is a local minimiser of the non-linear program. Furthermore, λ^k and η^k tend to λ^* and η^* respectively.

This tells us that the augmented Lagrangian method does in fact work.