Advanced Mathematical Programming
IE417

Lecture 12

Dr. Ted Ralphs

Reading for This Lecture

- Chapter 6, Section 4

Solving the Lagrangian Dual

Solving the Lagrangian Dual

- Since $\Theta(\mu, v)$ is concave, we can use a *line-search algorithm* to maximize it.
- If Θ is differentiable, then $\nabla\Theta(\mu^*, v^*)^T = (\mathbf{g}(x^*)^T, \mathbf{h}(x^*)^T)$ is an ascent direction.
- Move in this direction as far as is feasible.
- Move in a projected direction if $\mu_i = 0$ and $g_i(x^*) < 0$ for some index i.
- If Θ is not differentiable, then we have to work with subgradients.
- Finding the direction of steepest ascent in this case is an optimization problem.
- In practice, you may not want to move as far as possible each time.

Subgradient Algorithm for the Lagrangian Dual

- The idea of the subgradient algorithm is to first fix μ, v and solve the Lagrangian subproblem to get x.
- Then update μ, v by moving in an ascent direction for Θ.
- Here is a basic *subgradient algorithm* for solving the Lagrangian dual:
  1. Choose initial Lagrange multipliers $\mu^0 \geq 0$, v^0 and set $t = 0$.
  2. Solve the Lagrangian subproblem (evaluate $\Theta(\mu, v)$) to obtain x^t.
  3. Calculate an ascent direction d for Θ (usually by evaluating the constraints at x^t).
  4. Set $(\mu^{t+1}, v^{t+1}) \leftarrow (\mu^t, v^t) + \lambda^t \frac{d}{\|d\|}$ where λ^t is the chosen *step size*.
  5. Set $t \leftarrow t + 1$ and go to step 2.
- This algorithm is guaranteed to converge to the optimal solution as long as $\{\lambda^t\}_{t=0}^{\infty} \rightarrow 0$ and $\sum_{t=0}^{\infty} \lambda^t = \infty$
- In practice, one usually uses a geometric progression for the step sizes.
- Sometimes, it's difficult to know when the optimal solution has been reached.

Outer Linearization

- This method approximates the Lagrangian dual with a piecewise linear function.
- Recall the linear programming form of the Lagrangian dual:

$$\max_{z,\mu,v}\{z : z \leq \Phi(x^i, \mu, v), i \in 1, \ldots, T, \mu \geq 0\}$$

where $\{x^i\}_{i=1}^T$ are the members of the set X.

- In general, the set X may not be finite.
- We approximate Θ by using a subset of the members of X.
- Maximizing over the resulting piecewise approximation of the dual is a linear program.
- After maximizing, find a new constraint that "cuts off" the current optimum and continue.

Generating Primal Solutions

- Let (μ, v) be a given vector with $\mu \geq 0$ and consider the problem to minimize $f(x) + \mu^T \mathbf{g}(x) + v^T \mathbf{h}(x)$ subject to $x \in X$.
- If x^* is an optimal solution to this problem, then x^* is also an optimal solution to

$$\begin{aligned} \min \ & f(x) \\ \text{s.t. } g_i(x) &\leq g_i(x^*) \ \forall \, i \text{ such that } \mu_i > 0 \\ h_i(x) &= h_i(x^*) \ \forall \, i \in [1, l] \\ x &\in X \end{aligned}$$

- Hence, we get an approximate solution by evaluating Θ.

Generating Primal Solutions With Convexity

- In the convex case, things are easier.
- Given $\hat{x} \in X$, a feasible primal solution, and a sequence $x^i \in X$ of (possibly infeasible) primal solutions generated while solving the dual, we can obtain an improved primal feasible solution by solving

$$\begin{aligned} \min \ & \sum \lambda_j \, f(x_j) \\ \text{s.t. } & \sum \lambda_j \, g_i(x_j) \leq 0 \ \forall \ i \in [1, m] \\ & \sum \lambda_j \, h_i(x_j) = 0 \ \forall \ i \in [1, l] \\ & \sum \lambda_j = 1 \end{aligned}$$