Graphs and Network Flows
IE411

Lecture 22

Dr. Ted Ralphs

Lagrangian Relaxation

- Lagrangian Relaxation is a method of capitalizing on our ability to solve an underlying base model after adding side constraints.
- We remove the complicating constraints and instead assign a price associated with the resource.
- We can think of this price as a penalty for violation of the resource constraint.
- We set the prices, solve the underlying problem and then see if the resulting solution is feasible.

Example: Constrained Shortest Path

- Let us consider a shortest path problem in which each arc has both a length c_{ij} and a traversal time t_{ij} associated with it.
- We want to find the shortest path subject to the constraint that the total time is less than a certain limit T.
- Instead of imposing the limit directly, we assign a cost μ for each unit of time it takes to traverse the path.
- We then solve a regular shortest path problem with costs equal to $c_{ij} + \mu t_{ij}$.
- We can adjust μ if the time constraint is violated.

Bounding Principle

- Consider the modified cost of a feasible solution, i.e., a path that satisfies the time constraint.
- By definition, the additional cost imposed by the time penalty cannot be more than μT.
- Therefore, if we subtract μT from the modified cost, we will get a lower bound on the true cost.
- Another way of viewing this procedure is that we are adding $\mu(\sum_{(i,j)\in P} t_{ij} - T)$ to the cost of the path P.
- We can then think of μ as being a penalty on the slack in the time constraint.
- This is the general principle of Lagrangian Relaxation.

General Principle

- In general, the idea is to relax the constraints that we don't know how to deal with algorithmically.
- We have a vector of multiplier μ associated with all the constraints to be relaxed.
- The *Lagrangian subproblem* is to optimize over the relaxed problem with the costs adjusted by penalizing slack in the constraints.
- The value of the solution to the Lagrangian subproblem is denoted $L(\mu)$.
- Note that for inequalities, we must constrain the sign of the multiplier appropriately.
- The *Lagrangian dual* is to find the multipliers that maximize the lower bound.

Optimality Conditions

- When the constraints to be relaxed are equality constraints, if either

  1. For some solution x to the original problem and some choice of multipliers μ, we have $L(\mu) = cx$ or
  2. For some choice of multipliers, the solution x to the Lagrangian subproblem is feasible for the original problem,

  then x is optimal for the original problem.

- When some constraints are inequalities, we must also have complementary slackness, which says that the product of the multiplier and the slack for each constraint must be zero.

Solving the Lagrangian Dual

- Let us consider what $L(\mu)$ looks like as a function.
- We will consider the constrained shortest path problem as an example.
- Conceptually, one way we could compute $L(\mu)$ would be to enumerate all the paths and then taken the one that gave the smallest value.
- For a fixed path, the cost is linear in μ.
- Therefore, $L(\mu)$ as a function is the minimum of a finite number of linear functions.
- This means it is piecewise linear and concave.
- Thus, we need to maximize a concave function.
- This can be done with subgradient optimization.

Subgradient Optimization

- The idea is to compute $L(\mu)$ for some "guess" at the optimal multipliers.
- Then compute the gradient of the function and proceed in the direction indicated by the gradient (steepest ascent).
- We go in this direction for a certain fixed step size and this gives us a new guess.
- Fortunately, the gradient of the function $L(\mu)$ is easy to compute.

Implementation

For the constrained shortest path case, the basic loop for the equality constrained case is as follows:

- Pick an initial value for the multiplier μ^0 and $k \leftarrow 0$
- Main loop
  - Compute $L(\mu^k)$ by solving a shortest path problem with the time penalty μ^k to obtain path P^k.
  - If optimal, STOP.
  - Otherwise, $\mu^{k+1} \leftarrow [\mu^k + \theta_k(\sum_{(i,j)\in P^k} t_{ij} - T)]^+$.
- Note that the multipliers are never allowed to become negative.
- The value $\sum_{(i,j)\in P^k} t_{ij} - T$ is the gradient of $L(\mu)$ at μ^k.

Generalizing

The method is easy to generalize to other problem types. Here, we show the generalization for problem for which we have all inequalities.

- Pick initial multipliers μ^0
- Main loop
  - Compute $L(\mu^k)$ by solving the relaxation with Lagrangian objective to obtain x^k
  - If optimal, STOP.
  - Otherwise, $\mu^{k+1} \leftarrow [\mu^k + \theta_k s^k]^+$.

Here, s^k is the slack in the inequality constraints.

Convergence

- The main algorithmic choice in subgradient optimization is what step sizes to take (θ_k).
- Under mild conditions, the algorithm is guaranteed to converge to the optimal multipliers.
- Primarily the sequence of steps sizes must go to zero in the limit, but their infinite sum must go to ∞.
- In practice, choosing step sizes is something of an art.

Performing the Updates

- Suppose we have an estimate L^* of the optimal value.
- We can choose μ^{k+1} such that the Lagrangian objective of x^k is L^*.
- In other words, we want

$$cx^k + \mu^{k+1}s^k = L^*$$

- At the same time, we have that $\mu^{k+1} = \mu^k + \theta_k s^k$ (in the equality constrained case), so we have

$$cx^k + [\mu^k + \theta_k s^k]s^k = L^*$$

Performing the Updates (cont.)

- Finally, solving and putting it all together, we obtain

$$\theta_k = \frac{L^* - L(\mu^k)}{\|s^k\|^2}$$

- Since we do not usually know a good value for the new target, we can instead use the value of the best know solution.
- We also scale by a small factor that we reduce as the algorithm progresses.
- We then finally have

$$\theta_k = \frac{\lambda^k[UB - L(\mu^k)]}{\|s^k\|^2}$$

- Typically, we start with $\lambda^0 = 2$ and then reduced λ by half each time the Lagrangian objective does not improve for a specified number of iterations.
- Note that there is no convenient stopping criteria.