Delayed Column Generation
(aka variable generation)

Optimization - 10725

Carlos Guestrin
Carnegie Mellon University

February 13th, 2008

Why constraint generation converges

- LP with many many constraints:
- Solve with subset of constraints:
  - (also called "cutting planes")
- Relaxed problem, bound on objective:
- If solution x_Ω is feasible wrt all constraints:
- If solution x_Ω is infeasible wrt all constraints:

Constraint generation and duality

- Primal problem with many constraints:

$$\max_x \quad \sum_j \quad b_j x_j$$
$$s.t. \quad \sum_j \quad a_{ij} x_j \leq c_i, \quad \forall i \in \mathcal{I}$$

- Constraint generation: find most important constraints
- What's the dual equivalent?
- Dual:

Column generation (aka variable generation)

- Dual problem:

$$\min_y \quad \sum_{i \in \mathcal{I}} \quad c_i y_i$$
$$s.t. \quad \sum_{i \in \mathcal{I}} \quad a_{ij} y_i = b_j, \quad \forall j \in 1, \ldots, m$$
$$y_i \geq 0, \ \forall i \in \mathcal{I}$$

- Many many variables!!
- At optimal basic feasible solution
  - Most variables are zero
- Idea:
  - Set most variables to zero
  - Solve problem with other variables:
  - Incrementally increase sets of non-zero variables

Solving problem with subset of variables

- Solve problem with subset of variables

$$\min_y \sum_{i\in\Omega} c_i y_i$$
$$s.t. \quad \sum_{i\in\Omega} a_{ij} y_i = b_j, \quad \forall j \in 1,\dots,m$$
$$y_i \geq 0, \ \forall i \in \Omega$$

- Rest of variables set to zero

- Questions:
  - How do we decide what variables to use?
  - How do we decide when we are done?

What variables should we add?

$$\min_y \sum_{i\in\Omega} c_i y_i$$
$$s.t. \quad \sum_{i\in\Omega} a_{ij} y_i = b_j, \quad \forall j \in 1,\dots,m$$
$$y_i \geq 0, \ \forall i \in \Omega$$

- Same as simplex
  -
- Solve problem with variables Ω
  - At optimal basic feasible solution, set of basic variables B
- Find submatrix corresponding to basic variables A_B
  - Cost of these variables c_B
- Reduced cost for each potential new variable x_i, for $i \in I$:
  - If all are positive?
  - Otherwise:
- Guaranteed to converge to optimal solution

Column generation summary

- Dual of constraint generation
- Also useful for problems with infinitely many variables

- Some problems
  - Have efficient separation oracles
    - In these, constraint generation is useful
  - Have efficient variable generation oracles
    - In these, column generation is useful

- Both methods can be useful in polynomially large problems
  - E.g., when constraint matrix is too large to fit in memory
    - By incrementally solving the problem, bound amount of memory needed at each iteration

- If you have many many variables and constraints
  - Can use a combination of constraint and column generation