Dantzig Wolfe for the uninitiated

A map of how it works

ajaysa@smu.edu.sg / LARC

September 29, 2011

Structure

- 'Decompose' the problem to a smaller set of variables
  - $\rightarrow$ Uses special structure; sets of variables that only interact in the common constraint
  - $\rightarrow$ Uses convexity of feasible space; each extremity of feasible space represent one variable now
  - $\rightarrow$ Feasible space determined by the set of constraints for each group of variables $\implies$ we don't have to worry about those constraints any more!

Structure

- Would be easy if the decomposed problem only had a few variables - but this is not the case
- Simplex would take too long if the number of variables were in the millions
- Use ‘Revised Simplex’ method $\rightarrow$ needs understanding of simplex method, and how its properties are used

Simplex

- Describe Simplex Tableau
- Structure of the optimal tableau: basic/non-basic variables, etc.

$$\max z = \mathbf{c}_{BV}\mathbf{x}_{BV} + \mathbf{c}_{NBV}\mathbf{x}_{NBV} \quad (1)$$
$$\text{s.t.}\, B\mathbf{x}_{BV} + N\mathbf{x}_{NBV} = \mathbf{b} \quad (2)$$
$$\mathbf{x}_{BV}, \mathbf{x}_{NBV} \geq 0 \quad (3)$$

- Can be turned into :

$$z + (\mathbf{c}_{BV}B^{-1}N - \mathbf{c}_{NBV})\mathbf{x}_{NBV} = \mathbf{c}_{BV}B^{-1}\mathbf{b} \quad (4)$$
$$\text{s.t.} \quad \mathbf{x}_{BV} + B^{-1}N\mathbf{x}_{NBV} = B^{-1}\mathbf{b} \quad (5)$$

Properties of Optimal Simplex Tableau

- $(\mathbf{c}_{BV}B^{-1}N - \mathbf{c}_{NBV})$ is positive
  - if it is negative, then a non-basic variable can be made basic, increasing z
- $B^{-1}\mathbf{b}$ is positive
  - if it is negative, then solution is not feasible!
- These two are enough to gurarantee an optimal solution, irrespective of the rest of the table

Revised Simplex

- Revised Simplex uses this property - computes eqn (4) and finds most negative coefficient
- Uses eqn (5) to find maximum value increase for corresponding variable
- create new equations based on new B and recompute until no more negative coefficients
- does not recreate the whole table again!

Column Generation

- What if there are millions of variables (i.e. N is too large to compute eqn (4))
- Column generation frees us from having to do tedious work, finds the optimal column for us (column corresponds to variable/coefficient)

Column Generation

- Let us represent a column by $\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix}$
- Coefficient in the current tableau would be

$$\mathbf{c}_{BV} B_{curr}^{-1} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix} - c_{NBV} \tag{6}$$

- objective is to find minimum (most negative) of eqn (6)
- The above equation forms the objective of a subproblem!
- Note that c_{NBV} is not a vector $\implies$ coeff of all NBV is the same

Column Generation

- What about the constraints of the sub problem?
- What constrains the variables in the master? The original set of constraints! (make sure it's feasible)
- In D-W, the sub problem uses the original subset of constraints because they define the values each set of variables in the master can take

Back to Dantzig Wolfe

- Form a basic solution
- Solve sub problems to see which NBV variable best increases (in a max problem) the objective
- Enter the new NBV, form new equations
- Find next NBV to enter
- Rinse and repeat until no NBV increases the Objective

That's it!