Advanced Mathematical Programming
IE417

Lecture 18

Dr. Ted Ralphs

Reading for This Lecture

- Sections 8.8-8.9

Methods for Large Problems

- *Conjugate gradient* methods are based on the same idea of deflecting the gradient to get conjugate directions.
- However, they use a much simpler scheme.
- These methods are generally not as robust and are less efficient than quasi-Newton methods.
- However, they are much more practical for large problems.
- Quasi-Newton methods are generally impractical for more than 100 variables.

Conjugate Gradient Methods

- Idea: Let the next search direction depend on the last one, i.e.

$$d_{j+1} = -\nabla f(y_{j+1}) + \alpha_j d_j$$

- As before, we require that directions produced be H-conjugate when f is quadratic.
- There are various choices for α_j, depending on the assumptions one makes.
- However, all choices coincide for quadratic functions when performing exact line search.

Fletcher-Reeves Method

- For the FR method, $\alpha_j = \|\nabla f(y_{j+1})\|^2 / \|\nabla f(y_j)\|^2$.
- Implementation
  - Start with $k = 1$, initial point $y_1 = x_1,\ and\ d_1 = -\nabla f(y_1)$.
  - For $j = 1\ to\ n$
    - If $\|\nabla f(y_j)\| < \varepsilon$, then STOP.
    - Otherwise, perform a line search in direction d_j to find y_{j+1}.
    - Set $d_{j+1} = -\nabla f(y_{j+1}) + \|\nabla f(y_{j+1})\|^2 / \|\nabla f(y_j)\|^2 d_j$.
  - Set $x_{k+1} = y_n, k = k + 1$.
- As before, if the inner loop exits after only $n' < n$ iterations, we have a *partial conjugate gradient method*.

Comments on Fletcher-Reeves

- If f is a quadratic function, then the directions produced are conjugate and are descent directions.
- These methods are also guaranteed to converge in n steps for quadratic functions.
- *Preconditioning* the first search direction by applying a well-chosen symmetric p.d. matrix can improve conditioning, i.e. $d_1 = -D\nabla f(y_1)$.
- These methods are equivalent to a memoryless version of the BFGS quasi- Newton method (assume $D_j = I$).

Convergence

- Under mild conditions, all methods based on conjugate directions are globally convergent.
- This essentially depends on the fact that the first step taken in each sequence of steps is $-D\nabla f(y_1)$ for some symmetric p.d. matrix D.
- For $m - step$ partial CG methods, the rate of convergence is independent of the first m eigenvalues of H.
- For full CG methods, $n - step$ superlinear convergence can be proven.
- Similar results apply to quasi-Newton methods.

Subgradient Methods

- Consider the problem $\min\{f(x) : x \in X\}$ where X is a nonempty, closed, convex subset of $\mathbb{R}^n$.
- We assume f is convex, but not necessarily differentiable.
- Instead of using the direction $-\nabla f(x)$, find a subgradient ξ and use $-\xi$ as the search direction.
- Note that $-\xi$ is not necessarily a descent direction and it may not even be feasible.
- Idea: Use projection!

Subgradient Algorithm

- Typical Algorithm
  - Begin with starting point x_1 and $k = 1$.
  - Iterate
    * Find a subgradient d_k at x_k.
    * Select a step size $\lambda_k > 0$.
    * Project $x_{k+1} = x_k + \lambda_k d_k$ into the feasible region.
- This method requires an efficient method of projecting onto the feasible region and also an efficient method of finding the subgradients.
- We also need an effective stopping criteria.

Choosing the Step Size

- If the step size is chosen as follows, these methods do converge.
  - $\{\lambda_k\} \to 0$
  - $\sum \lambda_k = \infty$
- This is essentially due to the fact that, although d_k may not be a descent direction, it leads to points closer in norm to the optimum for small enough step size.
- In practice, the step sizes must be chosen very carefully to achieve fast convergence.
- Always try to get "closer" to the optimum.

Subgradient Deflection Algorithms

- As with conjugate direction methods, here we may also deflect the subgradient.
- For instance, in the spirit of CG methods, choose

$$d_k = -\xi_k + \phi_k(x_k - x_{k-1})$$

- Alternatively, in the spirit of quasi-Newton methods, multiply the direction by some symmetric p.d. matrix.