CSCI 6676 - Numerical Optimization

Nilesh Jagnik

University of Colorado Boulder

April 25th, 2016

Calculating Derivatives - Finite Difference

Approximate derivative at a point x by observing change in function values in response to small perturbations of the unknown near x.

$$\frac{\partial f}{\partial x_i} \approx \frac{f(x+\epsilon e_i) - f(x+\epsilon e_i)}{2\epsilon}$$

Where ϵ is a small scalar and e_i is the ith unit vector.

Approximating The Gradient - Forward Difference

$$f(x + \epsilon e_i) = f(x) + \epsilon \frac{\partial f}{\partial x_i} + O(\epsilon^2)$$

$$\frac{\partial f}{\partial x_i}(x) = \frac{f(x+\epsilon e_i) - f(x)}{\epsilon} + O(\epsilon)$$

Approximating The Gradient - Central Difference

$$f(x + \epsilon e_i) = f(x) + \epsilon \frac{\partial f}{\partial x_i} + \frac{1}{2}\epsilon^2 \frac{\partial^2 f}{\partial^2 x_i} + O(\epsilon^3)$$
$$f(x - \epsilon e_i) = f(x) - \epsilon \frac{\partial f}{\partial x_i} + \frac{1}{2}\epsilon^2 \frac{\partial^2 f}{\partial^2 x_i} + O(\epsilon^3)$$

$$\frac{\partial f}{\partial x_i} \approx \frac{f(x+\epsilon e_i) - f(x+\epsilon e_i)}{2\epsilon} + O(\epsilon^2)$$

Accuracy And Cost

The error in central-difference method is $O(\epsilon^2)$ as compared to $O(\epsilon)$ in forward-difference method. This is desirable because numerical calculations do not work very well when ϵ is too small.

Central-difference method is costly and requires two additional function evaluations. In practice, there are errors in these function evaluations and the accuracy is not worth the additional cost.

Jacobian

The Jacobian of a vector function $r : \mathbb{R}^n \to \mathbb{R}^m$ is defined as :

$$\mathbf{J}(x) = \begin{pmatrix} \frac{\partial r_1}{\partial x_1} & \cdots & \frac{\partial r_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial r_m}{\partial x_1} & \cdots & \frac{\partial r_m}{\partial x_n} \end{pmatrix} = \begin{pmatrix} \nabla r_1(x)^T \\ \nabla r_2(x)^T \\ \vdots \\ \nabla r_m(x)^T \end{pmatrix}$$

Approximating Jacobian using FD

We use the finite difference method to derive the following estimate of the ith column:

$$\frac{\partial r}{\partial x_i}(x) = \frac{r(x+\epsilon e_i) - r(x)}{\epsilon}$$

A full Jacobian can be obtained at a cost of $n+1$ function evaluations.

Approximating Sparse Jacobian

However, when the matrix is sparse, we can often obtain the estimate at much lower cost.

The key is to chose points in a way that they may be used to estimate multiple columns.

For example, instead of chosing to perturb in direction $p = \epsilon e_i$, we chose $p = \epsilon(e_i + e_j)$ when x_i and x_j are *not present in the same component of* r.

Example

$$r(x) = \begin{pmatrix} 2(x_2^3 - x_1^2) \\ 3(x_2^3 - x_1^2) + 2(x_3^3 - x_2^2) \\ 3(x_3^3 - x_2^2) + 2(x_4^3 - x_3^2) \\ 3(x_4^3 - x_3^2) + 2(x_5^3 - x_4^2) \\ 3(x_5^3 - x_4^2) + 2(x_6^3 - x_5^2) \\ 3(x_6^3 - x_5^2) \end{pmatrix} \tag{8.13}$$

Jacobian Structure for $r(x)$

$$J(x) = \begin{pmatrix} \times & \times & & & & \\ \times & \times & \times & & & \\ & \times & \times & \times & & \\ & & \times & \times & \times & \\ & & & \times & \times & \times \\ & & & & \times & \times \end{pmatrix}$$

1st column and 4th column have no overlap. x_1 and x_4 can be perturbed by a single point.

Perturbation Vectors

We chose

$$p = \epsilon(e_1 + e_4)$$

and note that

$$r(x+p)_{1,2} = r(x+\epsilon(e_1+e_4))_{1,2} = r(x+\epsilon(e_1))_{1,2}$$

$$r(x+p)_{3,4,5} = r(x+\epsilon(e_1+e_4))_{3,4,5} = r(x+\epsilon(e_4))_{3,4,5}$$

Estimating Multiple Columns With Single Fn Evaluation

Using $p = \epsilon(e_1 + e_4)$, we can write:

For Column 1,

$$\begin{pmatrix} \frac{\partial r_1}{\partial x_1}(x) \\ \frac{\partial r_2}{\partial x_1}(x) \end{pmatrix} \approx \frac{r(x+p)_{1,2} - r(x)_{1,2}}{\epsilon}$$

For Column 4,

$$\begin{pmatrix} \frac{\partial r_4}{\partial x_3}(x) \\ \frac{\partial r_4}{\partial x_4}(x) \\ \frac{\partial r_4}{\partial x_5}(x) \end{pmatrix} \approx \frac{r(x+p)_{3,4,5} - r(x)_{3,4,5}}{\epsilon}$$

Reduction of Function Evaluations

Similarly, for columns 2 and 5:

$$p_2 = \epsilon(e_2 + e_5)$$

And for columns 3 and 6:

$$p_3 = \epsilon(e_3 + e_6)$$

Instead of using six function evaluations, we only use three.

Reduction to Graph Coloring

For any function $r : \mathbb{R}^n \to \mathbb{R}^m$, we can construct a column incident graph by drawing an edge between nodes i and k if there is some component of r that depends on both x_i and x_k.

We assign each node a color using the following rule: Two nodes can have the same color if there is no edge that connects them.

If nodes $i_1, i_2 .... i_l$ have the same color, the correspoding perturbation vector is $p = \epsilon(e_{i_1} + e_{i_2} + ... + e_{i_l})$.

Graph For $r(x)$

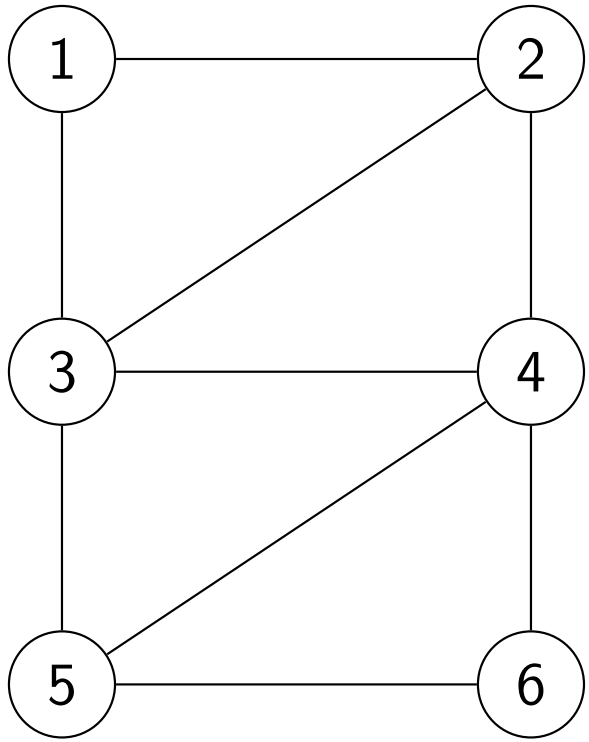

Figure : Column Index graph for $r(x)$ defined in **??**

Performance of Coloring Algorithms

Finding optimal coloring requires exponential time.

There are faster algorithms that find near-optimal solutions.

A greedy algorithm - Start by giving a color to node with highest degree. And eliminate possibilities.