Lecture 9: The conjugate gradient

Sourendu Gupta

TIFR Graduate School

Computational Physics 1
April 21, 2010

A generic function and a generic algorithm

A generic function admits a Taylor expansion as

$$f(\mathbf{x}+\mathbf{x}_0) = f(\mathbf{x}_0) + \nabla f|_{\mathbf{x}_0} \cdot \mathbf{x} + \frac{1}{2}\mathbf{x}\cdot A\cdot \mathbf{x} + \mathcal{O}(\mathbf{x}^3).$$

The matrix A is real-valued and symmetric, with matrix elements

$$A_{ij} = \left.\frac{\partial^2 f}{\partial x_i \partial x_j}\right|_{\mathbf{x}_0}.$$

If $\mathbf{x}$ is D-dimensional then A is a $D\times D$ matrix.
Sufficiently close to a minimum, therefore, one can neglect the cubic and higher order terms and write $f(\mathbf{x}) = c - \mathbf{b}\cdot\mathbf{x} + \mathbf{x}\cdot A\cdot\mathbf{x}/2$. The position of the minimum is given by the solution of the equation

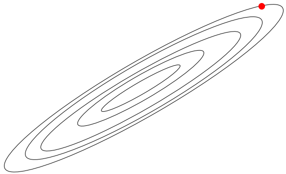
$$A\mathbf{x} = \mathbf{b}.$$

If A and $\mathbf{b}$ can be determined, then this could be the simplest way to determine the location of the minimum. This takes order D^2 time.

The parabolic approximation

If one performs a Taylor expansion around the point $\mathbf{x}_0'$, one again has a quadratic form of the function (provided that $\mathbf{x}_0' = \mathbf{x}_0 + \mathbf{y}$ is also close enough to the minimum for the higher order terms to be neglected). In that case one has the new coefficients

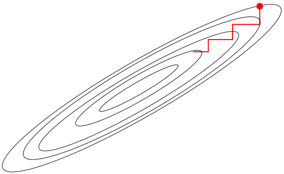
$$c' = c - \mathbf{b} \cdot \mathbf{y} + \mathbf{y} \cdot A \cdot \mathbf{y}, \quad b' = b - A \cdot \mathbf{y}, \quad A' = A.$$



In this approximation the **Hessian**, A, remains unchanged. Hence it can be evaluated at the minimum, where its eigenvalues are positive. The eigenvalues give the curvature of the function at its minimum. Contour lines are more closely spaced in the direction with the largest curvature.

A silly algorithm

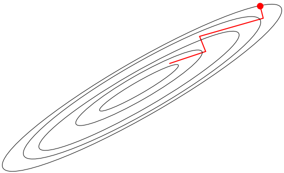
Choose a set of orthogonal directions $\mathbf{e}_i$ where $1 \leq i \leq D$. Starting from some point, $\mathbf{x}$, minimize along each of the directions in some fixed order. This does not, in general, reach the minimum of the function in D steps.



Problem 1: Prove or disprove the following assertion: if the eigenvectors of the Hessian are parallel to $\mathbf{e}_i$ then one can reach the minimum in D steps.

Steepest descent: another silly algorithm

Go along the direction of the gradient, $\mathbf{g}_0$, till you reach a minimum. Then go along the direction of the new gradient, $\mathbf{g}_1$, *etc.*. Again, this does not, in general, reach the minimum of the function in D steps.



Problem 2: Prove that $\mathbf{g}_i \cdot \mathbf{g}_{i+1} = 0$. Prove that the steepest descent algorithm does not terminate in D steps by showing that one can have $\mathbf{g}_i \cdot \mathbf{g}_j \neq 0$ when $i \neq j$.

All algorithms which ignore curvature information will turn out to be silly in the sense that they may not terminate in D steps.

What are conjugate directions?

When $f(\mathbf{x})$ can be treated in the parabolic approximation, the gradient is $\nabla f = A \cdot \mathbf{x} - \mathbf{b}$. In the i-th step of some minimization, if we move from $\mathbf{x}$ to $\mathbf{x} + \mathbf{h}_i$, then the change in the gradient is

$$\delta(\nabla f) = A \cdot \mathbf{h}_i.$$

Suppose we had previously moved along some direction $\mathbf{h}_{i-1}$ to minimize the function, and we want the gradient **after** the new move to remain orthogonal to $\mathbf{h}_{i-1}$, then we must have

$$\mathbf{h}_{i-1} \cdot \delta(\nabla f) = \mathbf{h}_{i-1} \cdot A \cdot \mathbf{h}_i = 0.$$

The two vectors are not orthogonal, but are said to be **conjugate**.

> **Problem 3**: How many mutually conjugate directions can one build in a real D-dimensional space using a symmetric positive definite A?

Finding conjugate directions

Starting from any vector $\mathbf{h}_0$, generate the sequence of matrix monomials acting on $\mathbf{h}_0$: $A\mathbf{h}_0$, $A^2\mathbf{h}_0$, $A^3\mathbf{h}_0$, *etc.*. Each of these can be made conjugate to all the previous ones by a process similar to the Gram-Schmidt process.

The first steps of this process are as follows. Let $\mathbf{h}_1 = A\mathbf{h}_0 + \alpha\mathbf{h}_0$ where we choose α so that the condition $\mathbf{h}_0 A\mathbf{h}_1 = 0$ is satisfied. Then we choose $\mathbf{h}_2 = A^2\mathbf{h}_0 + \beta\mathbf{h}_1 + \gamma\mathbf{h}_0$ and obtain β and γ by imposing the conditions $\mathbf{h}_0 A\mathbf{h}_2 = \mathbf{h}_1 A\mathbf{h}_2 = 0$. Proceeding in this way we can find all the conjugate directions. However, this requires that we store up to D vectors.

This process actually builds **matrix polynomials** acting on a fixed vector: $\mathbf{h}_0$, $[A+\alpha]\mathbf{h}_0$, $[A^2+\beta A+(\alpha\beta+\gamma)]\mathbf{h}_0$, *etc.*. As with many polynomials, it turns out that there is a two term recurrence relation for this: knowing only two of the polynomials (*i.e.*, the vectors), it is possible to construct the next. This non-trivial idea is incorportated into the computation of the conjugate gradient process.

Generating conjugate directions

We will generate two sequences of vectors $\mathbf{g}_i$ and $\mathbf{h}_i$ which enjoy the following properties—

$$\mathbf{g}_i \cdot \mathbf{g}_j = 0, \quad \mathbf{g}_i \cdot \mathbf{h}_j = 0, \quad \mathbf{h}_i \cdot A \cdot \mathbf{h}_j = 0, \quad \text{for} \quad i > j.$$

Start with some vector $\mathbf{g}_0$ and set $\mathbf{h}_0 = \mathbf{g}_0$. Then construct

$$\mathbf{g}_{i+1} = \mathbf{g}_i - \alpha_i A \cdot \mathbf{h}_i, \qquad \mathbf{h}_{i+1} = \mathbf{g}_{i+1} + \beta_i \mathbf{h}_i.$$

The orthogonality and conjugacy properties of $\mathbf{g}_i$ and $\mathbf{h}_i$ then allow us to solve for the unknowns—

$$\alpha_i = \frac{\mathbf{g}_i \cdot \mathbf{h}_i}{\mathbf{h}_i \cdot A \cdot \mathbf{h}_i} = \frac{\mathbf{g}_i \cdot \mathbf{g}_i}{\mathbf{h}_i \cdot A \cdot \mathbf{h}_i}, \qquad \beta_i = \frac{\mathbf{g}_{i+1} \cdot \mathbf{g}_{i+1}}{\mathbf{g}_i \cdot \mathbf{g}_i}.$$

Some problems

Problem 4: Find a set of vectors $\mathbf{y}_i$ such that one can set $\mathbf{g}_i = \mathbf{h}_i = \mathbf{y}_i$ and still satisfy the properties

$$\mathbf{g}_i \cdot \mathbf{g}_j = 0, \quad \mathbf{g}_i \cdot \mathbf{h}_j = 0, \quad \mathbf{h}_i \cdot A \cdot \mathbf{h}_j = 0, \quad \text{for} \quad i > j$$

for symmetric positive definite A.

Problem 5: Given a set of $\mathbf{g}_i$ and $\mathbf{h}_i$ satisfying the orthogonality and conjugacy relations above, is it possible to obtain another mutually conjugate set $\mathbf{h}'_i$ for which $\mathbf{g}_i \cdot \mathbf{h}'_j \neq 0$ for all i, j?

Problem 6: Obtain the expressions for α_i and β_i given above.

Problem 7: Find the time required to obtain the vectors $\mathbf{g}_{i+1}$ and $\mathbf{h}_{i+1}$ given the $\mathbf{g}_i$ and $\mathbf{h}_i$. Design an efficient parallelization of these computations.

The conjugate gradient algorithm

The algorithm is the following: initialize $\mathbf{h}_0 = \mathbf{g}_0$ to be the direction of the gradient (this choice does not really matter). Minimize along this direction. Then generate successive $\mathbf{h}_i$ and minimize along those directions. Terminate whenever the **residual**, *i.e.*, the change in the function value, is within a pre-specified tolerance. (See "Numerical Recipes" for a more detailed prescription.)

This completes the solution of the optimization problem. Since it is the same as solving a system of linear equations, it also solves the problem of the inversion of a matrix. The conjugate directions can be used to easily construct eigenvectors. Hence this algorithm also solves that problem. Linear differential equations (including partial differential equations) give rise to large and sparse matrix problems. Hence this algorithm can also be used to solve that class of problems.