

Notes

1 Conjugate Gradient Method

1.1 Overview

The conjugate gradient method (cgm) is an algorithm used to solve a linear system of the form

$$Ax = b \quad (1)$$

Where A is a symmetric ($A^T = A$) positive definite ($x^T Ax > 0$) $n \times n$ matrix, x , b vectors.

The algorithm is iterative, starting from a guess solution x_0 and taking a step towards the solution at each cycle.

The search directions are calculated from the residual term, defined as $r_i = b - Ax_i$.

It is possible to prove that by choosing the step direction to be A-orthogonal to all the previous ones, the solution converges the fastest (i.e. the error term $\|e_i\| = \|x_i - x\|$ is minimized).

1.2 Steepest descent

A simpler algorithm is the steepest descent.

The idea is to take a step in the direction of the residual so that the quadratic form is minimized.

$$x_{i+1} = x_i + \alpha_i r_i \quad (2)$$

$$\alpha_i \text{ such that } \frac{df(x_{i+1})}{d\alpha_i} = 0 \implies \alpha_i = \frac{\|r_i\|}{\|r_i\|_A} \quad (3)$$

This method is inefficient as x_i often finds itself oscillating around the solution, since the search directions explore non-disjoint subspaces.

1.3 The algorithm

A better alternative is to set the search direction to be A-orthogonal to the error at the next iteration. If this is the case, it can be proven that an exact solution be found after n iterations.

$$d_i^T A e_{i+1} = 0 \implies \frac{df(x_{i+1})}{d\alpha_i} = -r_{i+1}^T d_i = 0 \quad (4)$$

$$\alpha_i = \frac{r_i^T d_i}{\|d_i\|_A} \quad (5)$$

By definition, the residual is orthogonal to the previous search directions, we also have $r_i^T r_j = \delta_{ij}$. Since

$$r_{i+1} = -A(e_{i+1}) = -A(e_i + \alpha_i d_i) = r_i - \alpha_i A d_i \quad (6)$$

Finire

1.4 Preconditioning

The rate of convergance of cgm depends on the conditioning of the matrix A , defined as $\kappa(A) = \frac{\max \lambda_i}{\min \lambda_i}$, where λ_i are the eigenvalues of the matrix.

The closer $\kappa(A)$ is to 1, the faster the convergence of the method.

Given a certain matrix M , symmetric, positive definite and easily invertible and such that $M^{-1}A$ has better conditioning than A , which is to say M well approximates A , we can hope to solve the problem

$$M^{-1}Ax = M^{-1}b \quad (7)$$

much faster than the original problem, where the two solutions will be the same.

The problem is that $M^{-1}A$ is not necessarily symmetric or positive definite.

The fact that $\exists E$ such that $M = EE^T$ and $E^{-1}AE^{-T}$ is symmetric and positive definite, we can solve the problem.

$$E^{-1}AE^{-T}x = E^{-1}b \quad (8)$$

By using some clever substitutions, we can go back to the original problem with the aid of the preconditioner, giving the following algorithm Finire

2 Finding the smallest eigenvalue

Finding the smallest eigenvalue/eigenvector pair of a matrix amounts to evaluating the minimum of the Reyleigh quotient

$$\lambda(x) = \frac{x^T Ax}{x^T x} \quad (9)$$

Or, more generally

$$Ax = B\omega x \implies \lambda(x) = \frac{x^T Ax}{x^T Bx} \quad (10)$$

λ is not a quadratic form, hence te cgm needs to be modified to use it.

2.1 Useful multivariable relations

Given $f(x) = x^T A x$ and taking the derivative of f in the direction of v

$$f(x + hv) = (x + hv)^T A(x + hv) = f(x) + hv^T A x + hx^T A v + o(h) \quad (11)$$

$$\frac{df}{dv} = \lim_{h \rightarrow 0} \frac{f(x + hv) - f(x)}{h} = v^T A x + x^T A v = v^T A x + v^T A^T x \quad (12)$$

We can now evaluate the gradient of f in x

$$\nabla_x f(x) = \frac{df}{dv} = (A + A^T)x \quad (13)$$

We can now take the gradient of the Rayleigh quotient

$$\nabla \lambda(x) = \frac{(A + A^T)xx^T Bx - (B + B^T)xx^T A x}{(x^T Bx)^2} \quad (14)$$

Using the fact that A and B are symemtric

$$\nabla \lambda(x) = 2 \frac{Axx^T Bx - Bxx^T A x}{(x^T Bx)^2} = 2 \frac{Ax - \lambda(x)Bx}{x^T Bx} \quad (15)$$

2.2 Non linear conjugate gradient

Using a non quadratic form as function to be minimized, the things that will change will be

- The step size α_i will be different, we may now have multiple zeros regarding the orthogonality of the gradient and search direction.
- The factor β to compute conjugated directions no longer has equivalent forms.
- The residual needs to be computed each time as $-\nabla f(x_i)$