261: 24th January

Solving matrix equations with iterative techniques

To date, we have looked at ways of solving the matrix equation $Ax = b$, for x. We have also looked at how errors in A and b affect our answer. The methods may have been time consuming, but with enough perseverence we could exactly solve matrix equations of any size.

However, it is not uncommon for us to need to solve matrix equations that are too large to use these methods. As the size of the matrix increases, so too does the time needed to solve the equation. For example, in the second tutorial, we saw that a 10,000*10,000 matrix would take up to half an hour to invert on a modern PC. So there comes a point when it is unreasonable to ask a computer to solve a matrix equation using these methods.

Instead, we can use an algorithm that starts with an approximation to the answer and refines it, repeatedly. If we refine it enough times, we will get very close to the exact answer. With this algorithm, the length of time it takes to solve the problem is determined by how accurate we want our answer to be. The more accuracy we want, the more times we have to repeat the algorithm.

This is an example of an *indirect* method. The methods that we studied prviously, where we calculated the exact answer, are known as *direct*.

An example of iteration

To illustrate the idea, we will look at the iterative equation, $x_{i+1} = x_i/2 + 1$, where x_0 is an initial guess at the solution and x_i represents the value of our the approximate solution after the i-th application of the equation.

If we try setting $x_0 = 2$, and apply the equation, then $x_1 = 2$. Similarly, $x_2 = 2$, $x_3 = 2$. In fact, every subsequent value of x_i will also be equal to 2. This means that 2 is the exact answer to the equation.

However, if we start by making a guess at the answer and then repeatedly apply the equation, we will find that the subsequent values of x_i converge on 2.

Question 1)

Calculate x_1, x_2, x_3, x_4, x_5, x_6 and x_7 for each of

a) $x_0 = 10$

b) $x_0 = -10$

With iterative techniques, the solution converges on the exact answer. They never get to the exact answer in finite number of iterations. Only after infinitely-many iterations would x_i have reached 2.

Iteration in matrix equations

As ever, we start with our matrix equation $Ax = b$ and introduce another matrix, B

$$Ax + Bx - Bx = b$$

$$Bx = (B - A)x + b$$
$$x = B^{-1}(B - A)x + B^{-1}b$$
$$x = (I - B^{-1}A)x + B^{-1}b$$

This looks like something that we could use for an iterative equation.

$$x^{i+1} = (I - B^{-1}A)x^i + B^{-1}b$$

The error in the value x^i is $x^i - x$, where x is the exact solution to $Ax = b$. This is labelled e^i. Subtracting the penultimate line from the last line, we get

$$x^{i+1} - x = (I - B^{-1}A)(x^i - x)$$

or

$$e^{i+1} = (I - B^{-1}A)e^i$$

For the iterations to converge on the solution, we need e^i to get closer and closer to zero as i increases. So we need $(I - B^{-1}A)^\infty = 0$.

Now if we were dealing with numbers, not matrices, we could say that if $a^\infty = 0$, then $|a|$ must be less than 1. In the case of matrices, we can reach a similar conclusion, if we use the norms. $\|(I - B^{-1}A)\|$ must be less than 1, where the $\|..\|$ denote any norm, for example the infinity norm.

The spectral radius

The spectral radius of a matrix is defined as being its largest eigenvalue.

One useful feature of spectral radii, is that the spectral radius is always less than or equal to the norm of a matrix (this is true regardless of which definition of the norm we use). ie, for a matrix M,

$$\rho(M) \leq \|M\|$$

So, if $\|(I - B^{-1}A)\| < 1$, then it follows that the spectral radius of $(I - B^{-1}A)$ must also be less than 1. This gives us the following rule of thumb, *if a matrix is to converge, its largest eigenvalue must be less than 1.*

Question 2)

Would the following matrices converge?

$$a) \begin{pmatrix} \frac{3}{10} & \frac{-1}{10} \\ \frac{-1}{10} & \frac{3}{10} \end{pmatrix} \quad b) \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{3}{4} & \frac{1}{2} \end{pmatrix} \quad c) \begin{pmatrix} \frac{9}{10} & \frac{1}{10} \\ \frac{1}{10} & \frac{3}{10} \end{pmatrix}$$

The method of steepest descent

So far, we've looked at how to take $Ax = b$ and, by choosing B carefully, create an iterative equation. However, we haven't been very specific about how to choose B so that the solution converges as quickly as possible.

Now we are going to look at one of the more common algorithms and see how its parameters are chosen to give optimal convergence. The algorithm is called the *method of steepest descent*.

Using $Ax = b$, we can construct the function $f(x) = x^T Ax - bx + c$, where c is a constant number whose value doesn't matter. This function is a quadratic in x and, if we imagine x as having two components, then $f(x)$ is shaped like a bowl.

Now, if x is the solution to $Ax = b$, then $f(x) = x^T b - bx + c = c$. And if A is symmetric and positive definite, then this point is the bottom of the bowl shape.

An iterative approach to solving this, would be to pick a random starting vector, x_0, and modify x_0 in a direction that takes us down the surface; closer to the bottom and to the solution. After going some distance in that direction, we can stop, choose a new direction that will take us closer the bottom of the bowl and modify x in that direction.

We can keep doing this until either we hit the bottom of the bowl or we get close enough to meet our satisfation.

Let us look at the mathematical description of this. Our initial vector is x_0. The directon that will take us down the slope most steeply is $-\frac{df(x_0)}{dx_0}$, so we choose to modify x_0 in that direction.

$$x_1 = x_0 + \alpha_0(-\frac{df(x_0)}{dx_0})$$

where α_0 is a scalar. So what value does α_0 take? We can work this out by finding the minimum in $f(x_1)$. So we need to solve

$$\frac{d}{d\alpha} f(x_1) = 0$$

By the chain rule

$$\frac{d}{d\alpha} f(x_1) = \frac{df(x_1)}{dx_1}^T \frac{dx_1}{d\alpha}$$

But $\frac{dx_1}{d\alpha} = -\frac{df(x_0)}{dx_0}$. To make it neater, we will rename $-\frac{df(x_0)}{dx_0}$ as r_o. This gives us

$$\frac{df(x_1)}{dx_1}^T r_0 = 0$$

If we have introduced r_0 for $-\frac{df(x_0)}{dx_0}$, then we must introduce r_1 for $-\frac{df(x_1)}{dx_1}$.

$$r_1^T r_0 = 0$$

Now, if we look at the definition of $f(x)$ and try and differentiate it, we find that (after using the symmetry property of A)

$$\frac{df(x)}{dx} = Ax - b$$

so

$$\begin{aligned} r_1^T r_0 &= (b - Ax_1)^T r_0 \\ &= (b - A(x_0 + \alpha r_0))^T r_0 \end{aligned}$$

Because $r_1^T r_0 = 0$, this is still equal to zero. We can rearrange it to get

$$(b - Ax_0)^T r_0 = \alpha(Ar_0)^T r_0$$

The left hand side is equal to $r_0^T r_0$ and we can rearrange the right hand side, giving

$$r_0^T r_0 = \alpha r_0^T (A r_0)$$

Finally, this gives us for α,

$$\alpha = \frac{r_0^T r_0}{r_0^T A r_0}$$

Now we know the value of α, we have a complete algorithm for descending the surface to find the solution to $Ax = b$.

Starting at a guessed value for x_0, we work out the direction we need to travel in

$$r_0 = b - A x_0$$

We then calculate how far we need to go in that direction.

$$\alpha_0 = \frac{r_0^T r_0}{r_0^T A r_0}$$

and then we update the position of x.

$$x_1 = x_0 + \alpha_0 r_0$$

In summary, for the i-th iteration,

$$r_i = b - A x_i$$

$$\alpha_i = \frac{r_i^T r_i}{r_i^T A r_i}$$

$$x_{i+1} = x_i + \alpha_i r_i$$

A better explanation of the above can be found in the "conjugate gradient, pain-free paper" link, available on http://www.maths.tcd.ie/∼ watterss.

Question 3) Use the method of steepest descent to calculate x_1 and x_2, for,

$$A = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \quad b = \begin{pmatrix} 6 \\ 10 \end{pmatrix} \quad x_0 = \begin{pmatrix} 5 \\ 7 \end{pmatrix}$$