The power method

The power method is an iterative algorithm designed to estimate the largest eigenvalue λ_1 of an n by n matrix A along with its associated eigenvector. The method starts with the following assumptions:

1. A has a complete set of real eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ and λ_1 is the largest eigenvalue in absolute value.
2. A has a complete set of associated eigenvectors $\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \ldots, \mathbf{v}^{(n)}$ and these eigenvectors are mutually perpendicular.

The method starts with the observation that any vector $\mathbf{x}$ can be written as a linear combination of the eigenvectors of A:

$$\mathbf{x} = \sum_{j=1}^{n} \beta_j \, \mathbf{v}^{(j)}$$

Multiplying both sides of this equation by A gives us

$$A\,\mathbf{x} = \sum_{j=1}^{n} \beta_j \, A\,\mathbf{v}^{(j)} = \sum_{j=1}^{n} \beta_j \, \lambda_j \, \mathbf{v}^{(j)}$$

Hitting both sides with A k times gives us

$$A^k\,\mathbf{x} = \sum_{j=1}^{n} \beta_j \, A^k\,\mathbf{v}^{(j)} = \sum_{j=1}^{n} \beta_j \, {\lambda_j}^k \, \mathbf{v}^{(j)}$$

Factoring out ${\lambda_1}^k$ from the last sum gives us

$$A^k\,\mathbf{x} = {\lambda_1}^k \sum_{j=1}^{n} \beta_j \, \frac{{\lambda_j}^k}{{\lambda_i}^k} \, \mathbf{v}^{(j)}$$

Since λ_1 is the largest eigenvalues, the ratios in the sum for $j > 1$ go to 0 in the limit as k gets very large:

$$\lim_{k\to\infty} A^k\,\mathbf{x} = \lim_{k\to\infty} \; {\lambda_1}^k \, \beta_1 \, \mathbf{v}^{(1)}$$

If we could find a way to cancel out the ${\lambda_1}^k \, \beta_1$ terms in this expression we would have a way to develop an estimate for $\mathbf{v}^{(1)}$. Once we have that estimate we can multiply that vector by A to get an estimate for the eigenvalue λ_1.

To implement this strategy I am going to compute a sequence of vectors $\mathbf{x}^{(i)}$. The first vector in the sequence, $\mathbf{x}^{(0)}$ is chosen arbitrarily. To get subsequent vectors in the sequence I do the following:

1. Let $\mathbf{x}^{(i+1)} = (A\ \mathbf{x}^{(i)})/\|A\ \mathbf{x}^{(i)}\|_2$

2. Let m be chosen that $\mathbf{x}^{(i)}{}_m$ is the largest component in $\mathbf{x}^{(i)}$ by absolute value. Let μ_i be the m^{th} component of $A\ \mathbf{x}^{(i)}$ divided by $\mathbf{x}^{(i)}{}_m$.

We repeat these basic steps until the difference between successive $\mathbf{x}^{(i)}$ vectors in the l_2 norm falls below some preset tolerance. The sequence of $\mathbf{x}^{(i)}$ vectors we generate will approach $\mathbf{v}^{(1)}$ in the limit as i gets large, and the sequence of μ_i values will approach λ_1 in the limit as i gets large.

We can get even better estimates for both λ_1 and $\mathbf{v}_1$ by the following two tricks:

1. The sequence of μ values generated by the algorithm converges linearly to λ_1. We can use Aitken's Δ^2 method to accelerate this convergence and get an even better estimate for λ_1.

2. Once we have an estimate for λ_1 we can find a vector $\mathbf{v}$ that solves the equation $(A - \lambda_1 I)\ \mathbf{v} = \mathbf{0}$ by Gauss elimination. That vector is an even better estimate for the first eigenvector $\mathbf{v}^{(1)}$.

Wielandt deflation

Now that we have a method that can estimate the largest eigenvalue of a matrix along with its associated eigenvector, what about the remaining eigenvalues and eigenvectors?

What we want is a way to "factor out" this first eigenvalue and eigenvector from the matrix to get a smaller matrix whose eigenvalues and eigenvectors are the remaining eigenvalues and eigenvectors of A. It turns out that this is actually possible: to do this select we introduce the new matrix

$$B = A - \lambda_1\ \mathbf{v}^{(1)}\ \mathbf{x}^t$$

where the vector $\mathbf{x}$ is chosen to satisfy $\mathbf{x}^t\ \mathbf{v}^{(1)} = 1$.

What this does for us is given by the following theorem:

Theorem Suppose that $\lambda_1, \lambda_2, \ldots, \lambda_n$ are eigenvalues of a matrix A with associated eigenvectors $\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \ldots, \mathbf{v}^{(n)}$ and that the vector $\mathbf{x}$ is chosen to satisfy $\mathbf{x}^t\ \mathbf{v}^{(1)} = 1$. Then the matrix

$$B = A - \lambda_1\ \mathbf{v}^{(1)}\ \mathbf{x}^t$$

has eigenvalues $0, \lambda_2, \ldots, \lambda_n$with associated eigenvectors $\mathbf{v}^{(1)}, \mathbf{w}^{(2)}, \ldots, \mathbf{w}^{(n)}$ with

$$\mathbf{v}^{(i)} = (\lambda_i - \lambda_1)\ \mathbf{w}^{(i)} + \lambda_1\ \left(\mathbf{x}^t\ \mathbf{w}^{(i)}\right)\ \mathbf{v}^{(1)}$$

There are many ways to choose an appropriate vector $\mathbf{x}$ to satisfy the $\mathbf{x}^t\ \mathbf{v}^{(1)} = 1$ condition. In the

Wielandt deflation method we choose $\mathbf{x}$ to be the i^{th} column of A divided by $\lambda_1 \mathbf{v}^{(1)}{}_i$ where i is chosen so that the component $\mathbf{v}^{(1)}{}_i$ of $\mathbf{v}^{(1)}$ is nonzero.

Typically we choose i to be 0. After forming the new matrix B we will find that the first row of B consists of all 0s: this is consistent with the theorem's prediction that one of the eigenvalues of B is 0. If we then remove the first row and first column of B to form a smaller matrix B' we can use the power method outlined above to find the largest eigenvalue of B' and its associated eigenvector $\mathbf{w}^{(2)'}$. We can then reinflate the vector $\mathbf{w}^{(2)'}$ into an eigenvector $\mathbf{w}^{(2)}$ of B by simply adding a 0 element to the front of the vector. Once we have computed $\mathbf{w}^{(2)}$ we can use the formula from the theorem to compute $\mathbf{v}^{(2)}$.

By repeatedly running the deflation process and the power method we can compute a full sequence of eigenvalues and eigenvectors for the original matrix A.

The only downside to this iterative procedure is that roundoff errors are going to degrade our estimates for the eigenvalues and eigenvectors as we move down the list. To avoid this problem we are going to introduce a completely new strategy in sections 9.4 and 9.5.