10 The Rayleigh Quotient and Inverse Iteration

From now on we will restrict the discussion to *real symmetric* matrices $A \in \mathbb{R}^{m\times m}$ whose *eigenvalues* $\lambda_1, \ldots, \lambda_m$ are guaranteed to be *real* and whose *eigenvectors* $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_m$ are *orthogonal*. Moreover, in this case Householder reduction will produce a *tridiagonal* matrix.

Definition 10.1 *Let $A \in \mathbb{R}^{m\times m}$, $\boldsymbol{x} \in \mathbb{R}^m$. The quantity*

$$r(\boldsymbol{x}) = \frac{\boldsymbol{x}^T A\boldsymbol{x}}{\boldsymbol{x}^T\boldsymbol{x}} \quad \in \mathbb{R}$$

is called a Rayleigh quotient.

Remark If $\boldsymbol{x}$ is an eigenvector of A, then $A\boldsymbol{x} = \lambda\boldsymbol{x}$ and

$$r(\boldsymbol{x}) = \frac{\lambda\boldsymbol{x}^T\boldsymbol{x}}{\boldsymbol{x}^T\boldsymbol{x}} = \lambda.$$

In general one can show that $\lambda_{\min} \le r(\boldsymbol{x}) \le \lambda_{\max}$.

Another fact about the Rayleigh quotient is that it forms the least squares best approximation to the eigenvalue corresponding to $\boldsymbol{x}$. Consider

$$\boldsymbol{x}\alpha \approx A\boldsymbol{x}.$$

This is similar to the standard least squares problem $A\boldsymbol{x} \approx \boldsymbol{b}$. However, now we can ask to find α such that

$$\|A\boldsymbol{x} - \boldsymbol{x}\alpha\|_2$$

is minimized. The associated normal equations (cf. $A^TA\boldsymbol{x} = A^T\boldsymbol{b}$) are given by

$$\boldsymbol{x}^T\boldsymbol{x}\alpha = \boldsymbol{x}^TA\boldsymbol{x} \quad \text{or} \quad \alpha = \frac{\boldsymbol{x}^TA\boldsymbol{x}}{\boldsymbol{x}^T\boldsymbol{x}} = r(\boldsymbol{x}).$$

Thus, the Rayleigh quotient provides an *optimal estimate for the eigenvalue*. In fact,

$$r(\boldsymbol{x}) - \underbrace{r(\boldsymbol{q}_J)}_{=\lambda_J} = \mathcal{O}(\|\boldsymbol{x} - \boldsymbol{q}_J\|_2^2) \quad \text{as } \boldsymbol{x} \to \boldsymbol{q}_J.$$

Here $\boldsymbol{q}_J$ is the eigenvector associated with λ_J, and the estimate shows that the convergence rate of $r(\boldsymbol{x})$ to the eigenvalue is *quadratic*.

In order to prove this fact we need the first-order (multivariate) Taylor expansion of r about $\boldsymbol{x} = \boldsymbol{q}_J$:

$$r(\boldsymbol{x}) = r(\boldsymbol{q}_J) + (\boldsymbol{x} - \boldsymbol{q}_J)^T\nabla r(\boldsymbol{q}_J) + \mathcal{O}(\|\boldsymbol{x} - \boldsymbol{q}_J\|_2^2).$$

If we can show that the linear term is not present, then this identity will imply the quadratic convergence result we wish to prove. We will show that $\nabla r(\boldsymbol{q}_J) = \boldsymbol{0}$. First,

$$\nabla r(\boldsymbol{x}) = \left[\frac{\partial r(\boldsymbol{x})}{\partial x_1}, \frac{\partial r(\boldsymbol{x})}{\partial x_2}, \ldots, \frac{\partial r(\boldsymbol{x})}{\partial x_m}\right].$$

One of these partials can be computed using the quotient rule as

$$\frac{\partial r(\boldsymbol{x})}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\frac{\boldsymbol{x}^T A\boldsymbol{x}}{\boldsymbol{x}^T\boldsymbol{x}}\right) = \frac{\frac{\partial}{\partial x_j}(\boldsymbol{x}^T A\boldsymbol{x})\boldsymbol{x}^T\boldsymbol{x} - \boldsymbol{x}^T A\boldsymbol{x}\frac{\partial}{\partial x_j}\boldsymbol{x}^T\boldsymbol{x}}{(\boldsymbol{x}^T\boldsymbol{x})^2}.$$

Here

$$\begin{aligned}\frac{\partial}{\partial x_j}(\boldsymbol{x}^T A\boldsymbol{x}) &= \frac{\partial}{\partial x_j}(\boldsymbol{x}^T)A\boldsymbol{x} + \boldsymbol{x}^T\frac{\partial}{\partial x_j}(A\boldsymbol{x}) \\ &= \begin{bmatrix} 0 & \dots & 0 & 1 & 0 & \dots & 0 \end{bmatrix} A\boldsymbol{x} + \boldsymbol{x}^T A \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.\end{aligned}$$

Here the ones are in the j-th position so that the fact that we are dealing with scalar quantities (and therefore can transpose taking advantage of the symmetry of A) yields

$$\frac{\partial}{\partial x_j}(\boldsymbol{x}^T A\boldsymbol{x}) = 2(A\boldsymbol{x})_j.$$

Similarly, we can compute the other derivative as

$$\frac{\partial}{\partial x_j}(\boldsymbol{x}^T\boldsymbol{x}) = \begin{bmatrix} 0 & \dots & 0 & 1 & 0 & \dots & 0 \end{bmatrix}\boldsymbol{x} + \boldsymbol{x}^T \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = 2x_j.$$

Putting the pieces back together we get

$$\begin{aligned}\frac{\partial r(\boldsymbol{x})}{\partial x_j} &= \frac{2(A\boldsymbol{x})_j}{\boldsymbol{x}^T\boldsymbol{x}} - \frac{(\boldsymbol{x}^T A\boldsymbol{x})2x_j}{(\boldsymbol{x}^T\boldsymbol{x})^2} \\ &= \frac{2}{\boldsymbol{x}^T\boldsymbol{x}}\left((A\boldsymbol{x})_j - r(\boldsymbol{x})x_j\right),\end{aligned}$$

and we see that the gradient is given by

$$\nabla r(\boldsymbol{x}) = \frac{2}{\boldsymbol{x}^T\boldsymbol{x}}\left((A\boldsymbol{x}) - r(\boldsymbol{x})\boldsymbol{x}\right).$$

Finally, letting $\boldsymbol{x} = \boldsymbol{q}_J$ we have $r(\boldsymbol{q}_J) = \lambda_J$ and indeed $\nabla r(\boldsymbol{q}_J) = \boldsymbol{0}$ since $A\boldsymbol{q}_J = \lambda_j\boldsymbol{q}_J$.

10.1 Power Iteration

In order to take advantage of the Rayleigh quotient approximation of the eigenvalue we need a good approximation of an eigenvector. We begin with the basic form of the power iteration algorithm.

Algorithm (Power Iteration, simple form)

Initialize $\boldsymbol{v}^{(0)}$ with an arbitrary nonzero vector

for $k = 1, 2, \ldots$

$\boldsymbol{v}^{(k)} = A\boldsymbol{v}^{(k-1)}$

end

This algorithm generates a sequence of vectors

$$\boldsymbol{v}^{(0)}, A\boldsymbol{v}^{(0)}, A^2\boldsymbol{v}^{(0)}, \ldots$$

If we want to prove that this sequence converges to an eigenvector of A, the matrix needs to be such that it has a *unique largest eigenvalue* λ_1, i.e., $|\lambda_1| > |\lambda_2| \geq \ldots \geq |\lambda_m| \geq 0$. There is another technical assumption. The initial vector $\boldsymbol{v}^{(0)}$ needs to be chosen such that $\boldsymbol{q}_1^T\boldsymbol{v}^{(0)} \neq \boldsymbol{0}$. Otherwise, if $\boldsymbol{v}^{(0)}$ is completely perpendicular to the eigenvector $\boldsymbol{q}_1$, the algorithm will not converge. Usually, we do not worry about this assumption since a small roundoff error will already ensure it is satisfied.

We now show that the sequence produced by the power iteration algorithm converges to a multiple of $\boldsymbol{q}_1$, the eigenvector corresponding to λ_1.

Write the initial vector as a linear combination of the eigenvectors of A (note that they form a basis for $\mathbb{R}^m$ since A is assumed to be real symmetric):

$$\boldsymbol{v}^{(0)} = \sum_{j=1}^{m} a_j \boldsymbol{q}_j.$$

Then

$$\begin{aligned} \boldsymbol{v}^{(k)} &= A^k\boldsymbol{v}^{(0)} = A^k \sum_{j=1}^{m} a_j \boldsymbol{q}_j \\ &= \sum_{j=1}^{m} a_j A^k \boldsymbol{q}_j = \sum_{j=1}^{m} a_j \lambda_j^k \boldsymbol{q}_j \end{aligned}$$

since the λ^k are eigenvalues of A^k. Now we factor out the largest eigenvalue, i.e.,

$$\boldsymbol{v}^{(k)} = \lambda_1^k \left(a_1\boldsymbol{q}_1 + \sum_{j=2}^{m} \left(\frac{\lambda_j}{\lambda_1}\right)^k a_j \boldsymbol{q}_j \right).$$

Since $\lambda_j/\lambda_1 < 1$ the sum goes to zero as $k \to \infty$, and $\boldsymbol{v}^{(k)}$ does indeed converge to a multiple of $\boldsymbol{q}_1$.

The previous algorithm is not particularly stable. We can improve it by ensure that $\boldsymbol{v}^{(k)}$ is always of unit length. This leads to

Algorithm (Power Iteration, improved form)

Initialize $\boldsymbol{v}^{(0)}$ with an arbitrary vector such that $\|\boldsymbol{v}^{(0)}\|_2 = 1$

for $k = 1, 2, \ldots$

$$\boldsymbol{w} = A\boldsymbol{v}^{(k-1)}$$
$$\boldsymbol{v}^{(k)} = \boldsymbol{w}/\|\boldsymbol{w}\|_2$$
$$\lambda^{(k)} = \left[\boldsymbol{v}^{(k)}\right]^T A\boldsymbol{v}^{(k)}$$

end

Note that we have also added the Rayleigh quotient estimate to get the largest eigenvalue of A.

It can be shown that convergence to the eigenvector is linear, while convergence to the eigenvalue is still quadratic. More precisely,

$$\begin{aligned} \|\boldsymbol{v}^{(k)} - (\pm\boldsymbol{q}_1)\|_2 &= \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right) \\ |\lambda^{(k)} - \lambda_1| &= \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right), \end{aligned}$$

which shows that the speed of convergence depends on the *gap* between the two largest eigenvalues of A. In particular, if the largest eigenvalue of A were complex (which it can't be for the real symmetric matrices we are considering), then $\lambda_2 = \overline{\lambda_1}$ and the algorithm would not converge at all.

In order to find more than one eigenvalue we can perform *simultaneous power iteration* (see Chapter 28 in [Trefethen/Bau] for more details):

Algorithm (Simultaneous Power Iteration)

Initialize $V^{(0)}$ with an arbitrary $m \times n$ matrix of rank n

for $k = 1, 2, \ldots$

$$V^{(k)} = AV^{(k-1)}$$

end

$$\hat{Q}^{(k)}\hat{R}^{(k)} = V^{(k)}$$

In the last step of the algorithm we perform a reduced QR factorization to obtain a well-behaved basis for the column space of $V^{(k)}$.

For this algorithm the speed of convergence will depend on the smallest gap among the first $n + 1$ eigenvalues.

The problem with this algorithm is that *all* of the columns of $V^{(k)}$ converge to a multiple of $\boldsymbol{q}_1$, and therefore the column of $Q^{(k)}$ will form an extremely ill-conditioned basis for range($V^{(k)}$). The fix is simple, but more expensive. We should orthonormalize at each step:

Algorithm (Orthogonal Simultaneous Iteration)

Initialize $\hat{Q}^{(0)}$ with an arbitrary $m \times n$ matrix with orthonormal columns

for $k = 1, 2, \ldots$

$Z = A\hat{Q}^{(k-1)}$

$\hat{Q}^{(k)}\hat{R}^{(k)} = Z$

end

We will see later that this algorithm is very useful. In fact, it is equivalent to the QR iteration algorithm we will study soon (*not* the QR factorization algorithm).

We now return to basic power iteration, and consider a few more modifications. If we want to find the smallest eigenvalue instead of the largest one, then we perform power iteration for A^{-1} (since the eigenvalues of A^{-1} are the reciprocals of the eigenvalues of A; see homework). Of course, we *do not want to compute* A^{-1}.

Instead we use the following

Algorithm (Inverse Iteration)

Initialize $\boldsymbol{v}^{(0)}$ with an arbitrary vector such that $\|\boldsymbol{v}^{(0)}\|_2 = 1$

for $k = 1, 2, \ldots$

Solve $A\boldsymbol{w} = \boldsymbol{v}^{(k-1)}$ for $\boldsymbol{w}$

$\boldsymbol{v}^{(k)} = \boldsymbol{w}/\|\boldsymbol{w}\|_2$

$\lambda^{(k)} = \left[\boldsymbol{v}^{(k)}\right]^T A\boldsymbol{v}^{(k)}$

end

Remark For this algorithm the matrix A needs to be factored *only once* (by Cholesky factorization for a symmetric A).

Another modification that can be applied to either inverse iteration (or basic power iteration) is a *shift* μ, i.e., we consider $A - \mu I$ instead of A. Then

1. The eigenvalues of $A - \mu I$ are $\lambda_j - \mu$, and
2. The eigenvectors of $A - \mu I$ are still the same as those of A.

This is clear since

$$(A - \mu I)\boldsymbol{x} = \underbrace{A\boldsymbol{x}}_{=\lambda\boldsymbol{x}} - \underbrace{\mu I\boldsymbol{x}}_{=\mu\boldsymbol{x}} = (\lambda - \mu)\boldsymbol{x}.$$

The resulting algorithm (for inverse iteration is)

Algorithm (Inverse Iteration with Shift)

Initialize $\boldsymbol{v}^{(0)}$ with an arbitrary vector such that $\|\boldsymbol{v}^{(0)}\|_2 = 1$

for $k = 1, 2, \ldots$

Solve $(A - \mu I)\boldsymbol{w} = \boldsymbol{v}^{(k-1)}$ for $\boldsymbol{w}$

$\boldsymbol{v}^{(k)} = \boldsymbol{w}/\|\boldsymbol{w}\|_2$

$\lambda^{(k)} = \left[\boldsymbol{v}^{(k)}\right]^T A\boldsymbol{v}^{(k)}$

end

This algorithm will yield the eigenvalue *closest to* μ. This means by picking appropriate shifts μ, *any one* eigenvalue of A can be found. The rate of convergence to the eigenvector is still linear, and that to the eigenvalue is quadratic.

Remark If $\mu = \lambda$, i.e., one runs the algorithm with a known eigenvalue, then one step of inverse iteration will produce the associated eigenvector.

Remark Shifter power iteration — while theoretically possible — is not very useful since it converges to the eigenvalue *farthest away* from μ.

10.2 A Cubically Convergent Improvement

If we update the estimate μ for the eigenvalue with the Rayleigh quotient at each iteration we can get a cubically convergent algorithm:

Algorithm (Rayleigh Quotient Iteration)

Initialize $\boldsymbol{v}^{(0)}$ with an arbitrary vector such that $\|\boldsymbol{v}^{(0)}\|_2 = 1$

Initialize $\lambda^{(0)} = \left[\boldsymbol{v}^{(0)}\right]^T A\boldsymbol{v}^{(0)}$

for $k = 1, 2, \ldots$

Solve $(A - \lambda^{(k-1)}I)\boldsymbol{w} = \boldsymbol{v}^{(k-1)}$ for $\boldsymbol{w}$
$\boldsymbol{v}^{(k)} = \boldsymbol{w}/\|\boldsymbol{w}\|_2$
$\lambda^{(k)} = \left[\boldsymbol{v}^{(k)}\right]^T A\boldsymbol{v}^{(k)}$

end

One can show that for almost all starting vectors we have

$$\begin{aligned} \|\boldsymbol{v}^{(k+1)} - (\pm\boldsymbol{q}_J)\|_2 &= \mathcal{O}\left(\|\boldsymbol{v}^{(k)} - (\pm\boldsymbol{q}_J)\|_2^3\right) \\ |\lambda^{(k+1)} - \lambda_J| &= \mathcal{O}\left(|\lambda^{(k)} - \lambda_J|^3\right), \end{aligned}$$

Since this is a cubically convergent algorithm one can expect the number of correct digits to *triple* in each iteration. This is illustrated in the MATLAB script `RayleighQuotient.m`. However, each iteration of the algorithm is fairly expensive since we need to solve a linear system with *different* system matrix in each iteration. Also, while the algorithm does usually converge, it need not converge to the eigenvalue closest to the initial shift $\lambda^{(0)}$.

Remark Cubically convergent algorithms are very rare in numerical algorithm (recall the the quadratically Newton iteration for finding roots of a nonlinear function is already considered a big deal).

Remark In order to find *all* eigenvalues and eigenvectors of A we can use *deflation* (which we will discuss a bit more in the next section in the context of QR iteration).